FACULTY OF COMPUTING AND INFORMATION MANAGEMENT

BACHELOR OF BUSINESS INFORMATION TECHNOLOGY

UNIT: KCA 398

PROGRAMMING PROJECT

TITLE: SOSA ATTACHMENT MANAGEMENT SYSTEM

SYSTEM DESIGN SPECIFICATION DOCUMENT

BY: JASON NABOTH OMONDI

Reg No – 19/03354

SUPERVISOR: MR ISAAC OKOLA

MAY – DEC 2022

System Design Specification Submitted To The Faculty Of Computing And Information Management In Partial Fulfillment Of Bachelors In Information Technology For KCA University.

Table of Contents

1.0 Introduction

In this chapter, we'll go over how the system works in general. The document describes and illustrates the data, architectural, interface, and component-level designs of the software. This document will serve as a guide for the developer, explaining how the program is divided into modules, how the modules interact with one another, and how users perceive the program. The proposed system is The Sosa Internship Management System, which was created for the placement cells of a variety of educational institutions, and this software requirements design defines all of its features and limitations.

1.1 Objective

The primary goal of this document is to serve as a reference for those who implement the suggested system. It includes the architecture of each application, as well as any associated interfaces and database designs.

1.2 System Scope

The scope of this document is to show the design constrains of the system (Physical as well as the database design).

The product will use a local Apache web server to serve the user interface, which will be accessed through the Google Chrome web browser.

The front end is designed using HTML, CSS, and JavaScript, while the back end is handled by PHP.

There are three types of users in the web interface: administrators, students, and staff. Each user type will be granted different levels of system access.

1.3 System Overview

Sosa Internship management system is a web-based project that, after logging in, displays a dashboard that informs students about companies that have open and upcoming opportunities in the coming weeks, as well as brief information about the company and details about the job profile, package, and so on. Specifics about a particular company will be provided later or in another section.

2.0 System Architecture

2.1 Architectural and component-level design

The software interface is designed for use with Windows or Linux operating systems (Windows 10, Linux 4.0). The web server is Apache 5.2.4, and the database is MySQL. The scripting languages used are PHP (Hypertext Preprocessor) and JavaScript.

The product will run on the XAMPP server, with the user interface accessible through the Google Chrome web browser.

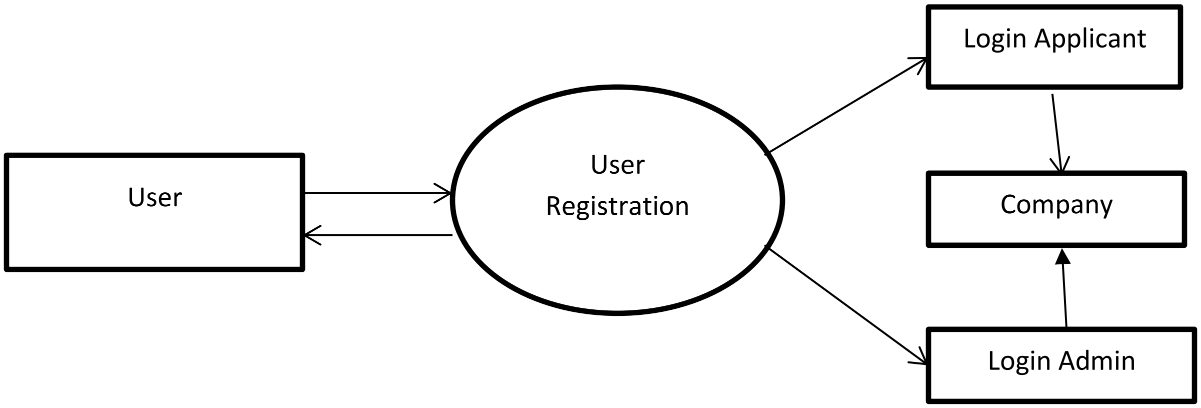
Front-end languages include HTML, CSS, and JavaScript. They will determine image placement, size, and overall layout, as well as background and border colors and text display. All images are jpeg or png in size.

2.2 Dataflow Overall System

A registered user can access the system as an administrator or as a candidate.

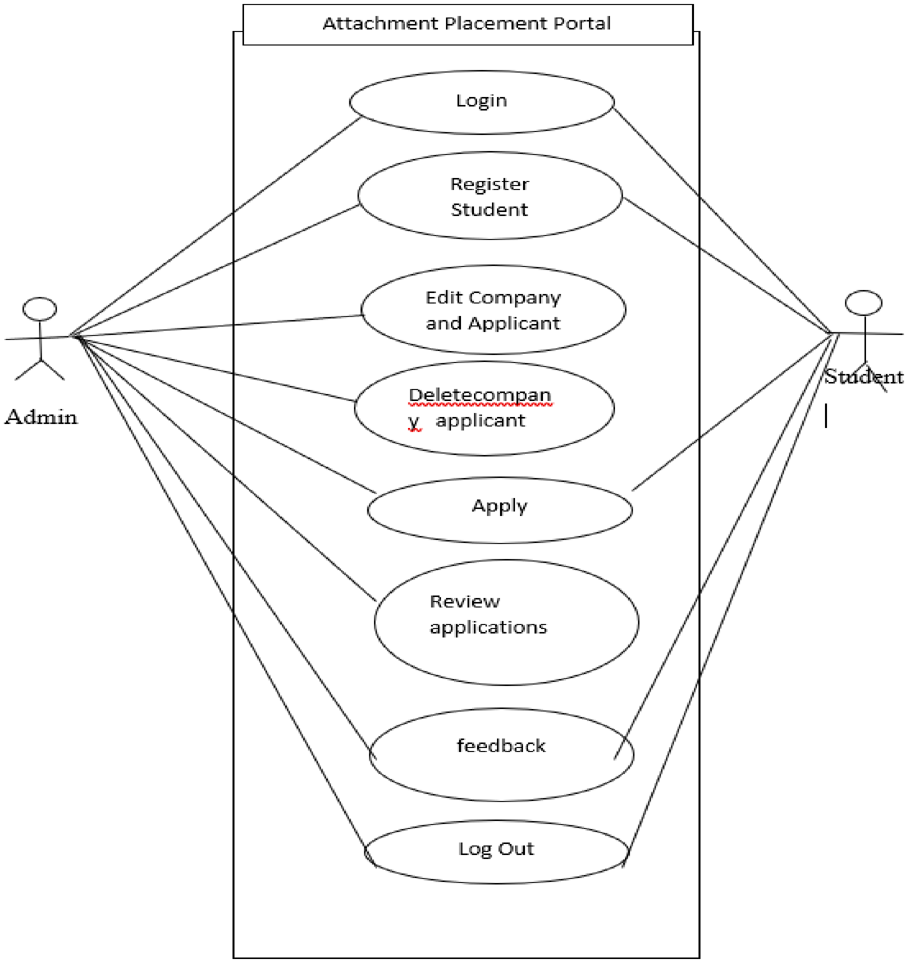
An administrator has access to information about the company and the applicants. In comparison to other users, the admin module gives the system administrators absolute power. The administrator can add more companies to the system and track the progress of applications in the admin module.

On a type-and-search basis, applicants can access internship opportunities from companies. They can apply for jobs by filtering different categories based on their field of study.



2.3 System structure

The user(student/admin) logs in by entering their user id and password. After the authenticating system validates the user id and password, it redirects to a PHP page that displays the main page. Furthermore, the user information is saved in the session for future use, and the session is terminated when the browser is closed. Depending on the user's privileges, the system will display different menu items.



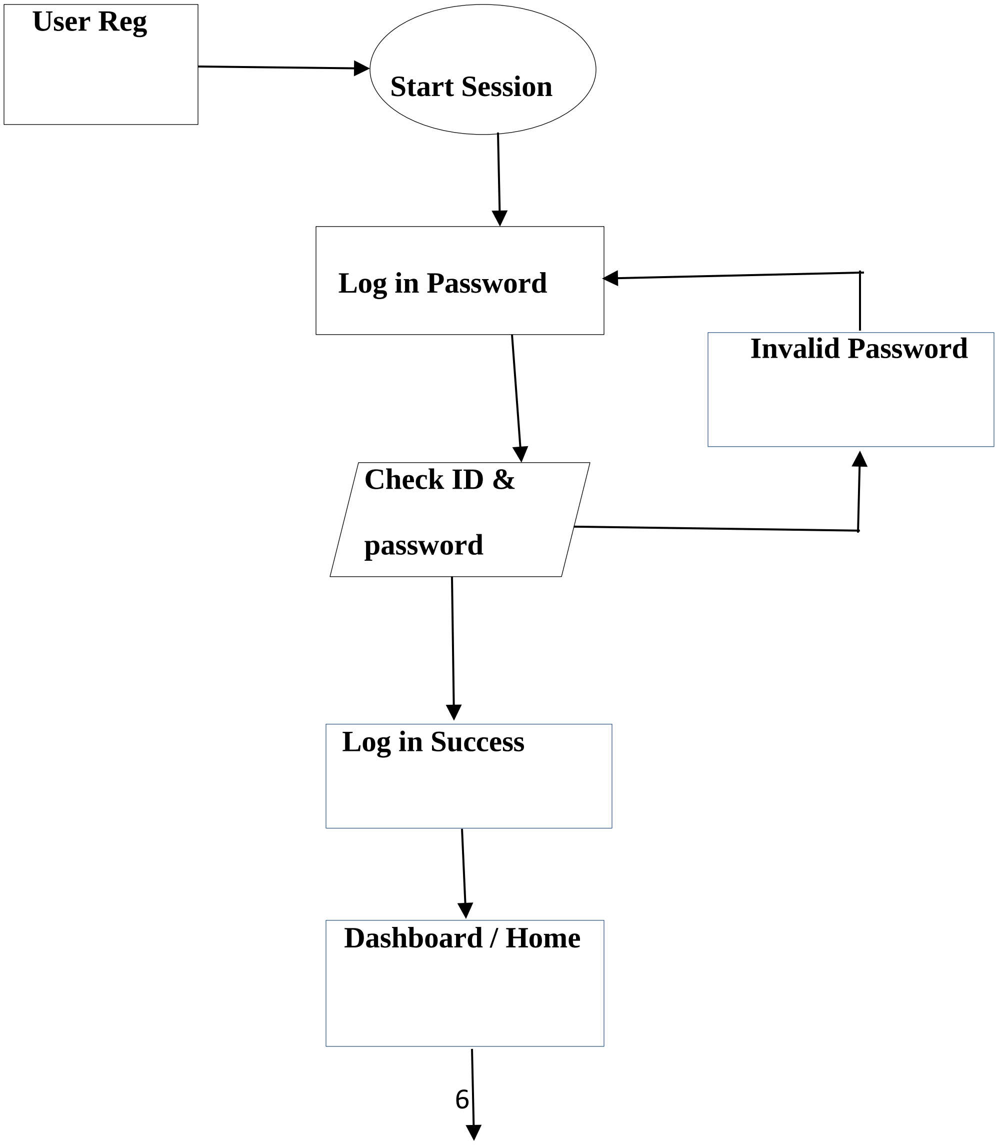
2.4 Logical Design

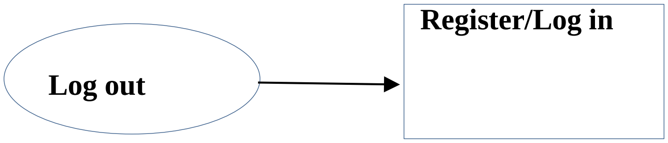
The method used to create logical design modules is to divide the system into modules so that the operations performed within each module are highly related to one another but not so much to other operations performed within other modules. Taking this into account, the main services of the internship management system project are as follows:

- Student Module
- Admin Module
- User Authentication Module

This diagram represents the logical design of the system.

To access the dashboard and homepage modules, registered users must enter their usernames and passwords. Users can also log out, which will restart the authentication module, as shown in the figure below:





2.4.1 User Authentication Module

This is the user authentication module where user identify him/herself. Its purpose is to enables users register to system and log in, also this module provides changing password and help user if they forgot their password.

All Users have access to this module in order for them to be logged into their respective accounts.

User authentication module contains 3 subordinate which its sign in, sign up and the change password module in which under it you can change password or forget password. The three subordinate modules are shown below

2.4.1.1 Sign in

Identification: Sign in

Type: Process

Purpose: Provides user to get their authorization level which is recorded in corresponding cell of the user table.

Function: A user enters their required information (username, password) to the web form. If those information match with a row of the user table, user gets their authorization level.

2.4.1.2 Sign Up

Identification: Sign up

Type: Process

Purpose: Provides a registration process to user, if user information is not recorded to the user table.

Function: A visitor fills the registration form and click on the Complete Registration button. If every information given is correct, user's information is recorded to user table and user get the customer status by default. For other employee is defined by the administrator in their panel manually

2.4.1.3 Password

Under password module user can either change or forget password

2.4.1.3.1 Change Password

Identification: Change Password

Type: Process

Purpose: To make user able to change their passwords.

Funtion: User enters their old password and new password. New password replaces with the old password if it is correct.

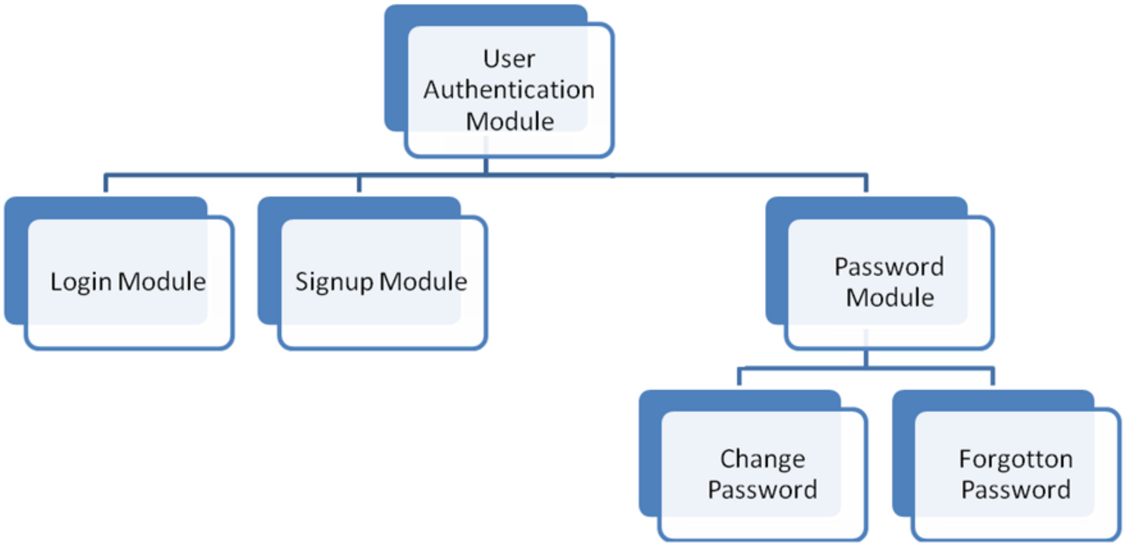
2.4.1.3.2 Forgotten Password

Identification: Forgotten Password

Type: Process

Purpose: By this module, user can get their fogotten password.

Funtion: User sends a request to the system that denotes they have forgotten their password. System sends them an e-mail including their password.



The User authentication module is a security check designed to confirm users into the system. If a user exists in the database, regardless of who they are, they should be able to access the next module by entering their username and password.

If a user forgets their password, there is an optional submodule where they can reset it.

After logging into the system, users may be able to change their passwords.

2.4.1.4 Student Dashboard Module

This module is responsible for viewing and applying internship opportunities when an applicant has found the internship of their choice. Here the applicant can view the internship requirements, upload resume or cover letter or internship letter and apply for the best opportunities that suit them.

This module is accessible only to registered users looking for internship positions.

Applicants can also view the progress of their application and will receive feedback via email and on the module too.

It contains sub panels such as;

- The student search result page will only display a list of preferred attachment opportunities for particular category search. The name of the attachment company along with the available positions is displayed. This page provides a link to the view student progress page where student can the apply for attachments of their choice and upload their resume.
- Students can only view the company information. The search result is sorted alphabetically and contains the name, address, phone number and. A brief description of the company.

2.4.1.5 Admin Module

The administrator has absolute power of creating users and giving permissions, update company and staff data, track the progress of internship applications and many more.

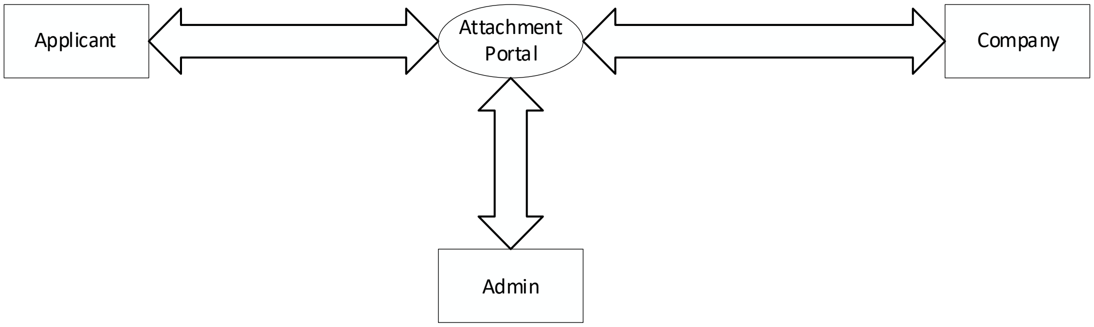
This module has four sub-systems within them namely;

- Company – new companies can be added on the company panel. The admin can also view the company data such as address company name, contact number, delete a company and also edit their details.
- Internships
- Staff Members
- Student Applications and Applicants – The administrator can view and delete student applications.
- Internship Categories – The admin can view internship categories and their respective number of students who've applied to them.
- Users – This module allows the administrator to update and delete other users' data, add new companies to the Company database table and schedule

3.0 Interface design

The respective interface of the new system design to be human friendly and function are explained below. The use interface is developed is a browses specific environment to have distributed architecture. PHP will be used to generate background and border colors, as well as text display. To control client-side interactions, JavaScript will be used (i.e., sound output when image is clicked). The images and sounds that will be displayed at the output will be stored in the MySQL database.

Below is a design diagram of the interface flow once you access the system as a user.

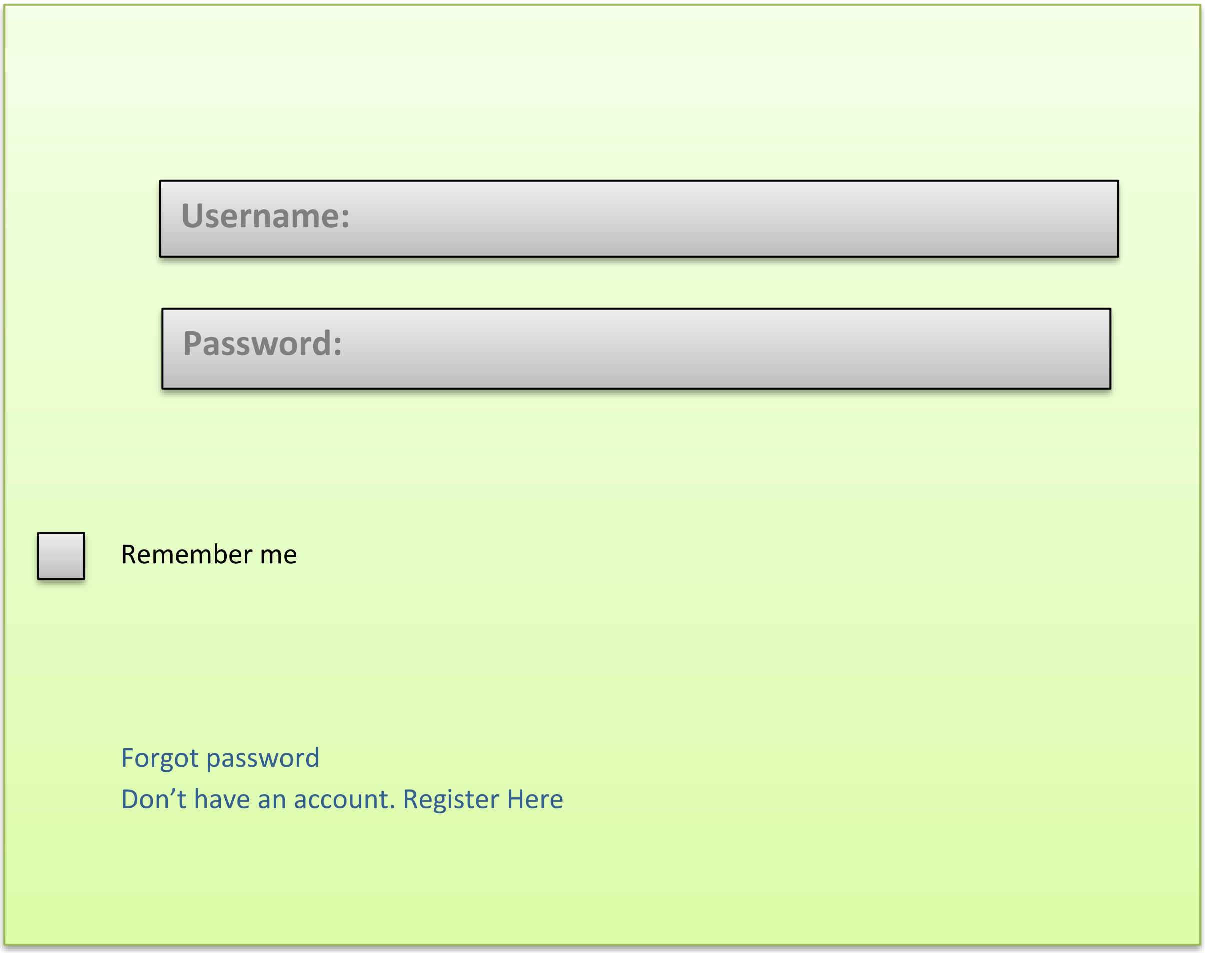


Students must register as new users in the system. The system interface includes a registration form that collects new user information and creates an account in the database.

The administrator can add companies to the system, and companies can receive and respond to student applications via email.

3.1 Login

Registered Users can access the system and update data based on the level of access granted to the user.

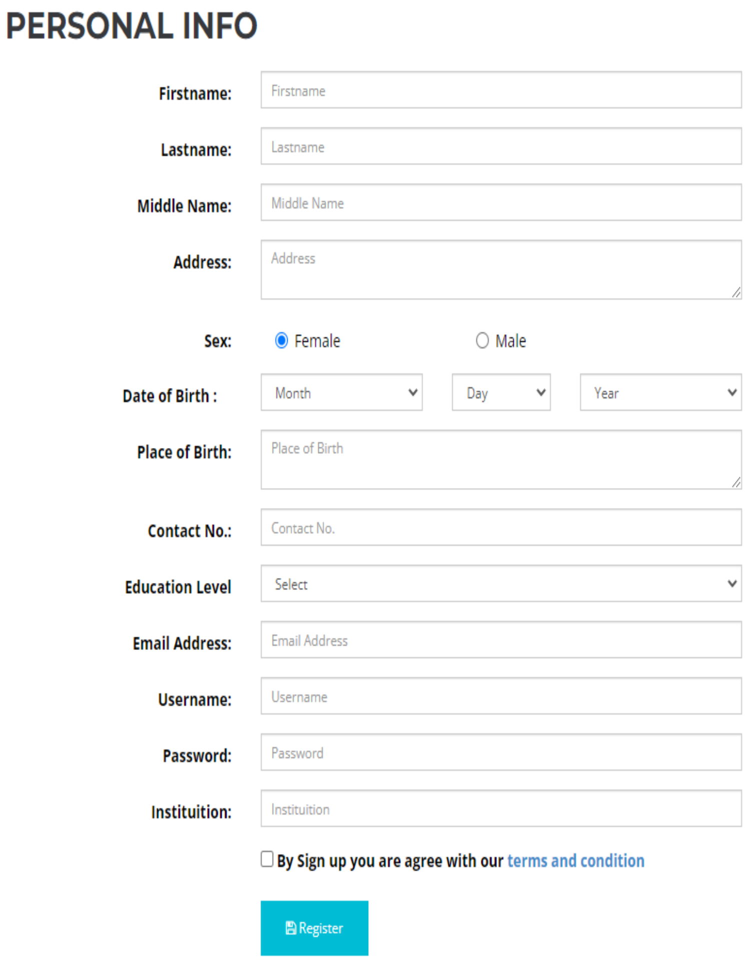


3.2 Registration

Students and businesses must be registered with all of their documentation, which the Administrator will verify. The student and the company will be allowed to proceed only after extensive verification. These documents would also be kept by the Administrator for future reference. The documents should be kept at least until the placement activities of the company or students are completed.

Students should provide complete information, such as academic or professional credentials, as well as personal information, when registering for the company.

On successfully signing into the system, the student login page displays all the privileges a student has in this system.



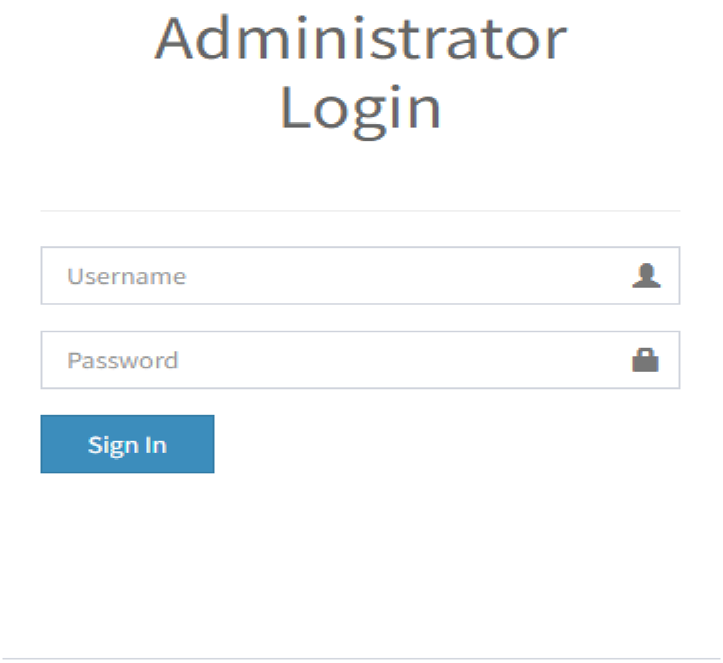
PERSONAL INFO

Firstname:	Firstname
Lastname:	Lastname
Middle Name:	Middle Name
Address:	Address
Sex:	● Female ○ Male
Date of Birth :	Month Day Year
Place of Birth:	Place of Birth
Contact No.:	Contact No.
Education Level	Select
Email Address:	Email Address
Username:	Username
Password:	Password
Instituition:	Instituition

□ By Sign up you are agree with our terms and condition

3.3 Administrator

The admin login interface is shown below.



The admin has absolute power over all other users of the system.
To gain access to the admin page, the user must first log in with the administrator credentials.

4.0 Database Design

The system is made up of three tables: the applicant table, the company table, and the admin table. The applicant table contains all of the information about the students, whereas the company table contains information about the companies that have enrolled in the system. The admin table stores the company administrators.

Database Schema

- ***Table Applicants***

Field	**Data Type**	**Relationship**
Applicant_ID	Int	Primary Key
First_Name	Varchar	Not null
S_name	Varchar	Not null
M_Name	Varchar	Not null
Sex	Varchar	Not null
Age	Int	Not null

Education_Level	Varchar	Not null
Birth_date	Varchar	Not null
Username	Varchar	Foreign Key
Password	Varchar	Not null
Email	Varchar	Not null
Photo	Varchar	Not null
National_ID	Varchar	Not null

- **Table attachment files**

Field	Data Type	Relationship
ID	Int	Primary Key
loc_attach_files	Varchar	Null
File_ID	Varchar	Not null
attachment_ID	Int	Not null
User_ID	Int	Foreign Key

- **Table category**

Field	Data Type	Description
Category_ID	Int	Primary Key
category	Str	Foreign Key

- **Table company**

Field	Data Type	Description
Company_id	Int	Primary Key
Name	Varchar	Not null
Address	Varchar	Not null
Contact	Varchar	Not null
status	Varchar	null

- **Table Users**

Field	Data Type	Description
User_Id	int	Primary Key
Fullname	Char	Not null
Username	Char	Foreign Key
Password	Varchar	Not null
Role	Char	Not Null

- **Table Attachments**

Field	Data Type	Description
Attachment Id	Int	Primary key
Company Id	Int	Foreign key
Company name	Varchar	Not null
Category	varchar	Not null
Title	varchar	Not null
Salary	Int	Null
Duration	Varchar	Not null
Qualification	Varchar	Not null
Description	Varchar	Not null
Sex	Varchar	Null
Date_posted	Varchar	varchar

5.0 Testing

5.1 Development environment

The system will be developed on top of a PC with windows operating system using PHP programming language and a MYSQL Database.

The development pc has the following specs

-8GB RAM

-2.60GHZ duo

-500GB HDD

-8 –CPU’s

For the PHP programming environment

--Atom text editor.

5.2 SYSTEM COMPONENTS

The following will be the components of this system.

Hardware: -

Laptop - the application will run on top of this.

Software: - XAMPP server- MYSQL database will run on top of this server

Operating system – The system can run on top of windows Operating system.

Human: - The system has two types of users’ company admins and students. Each type of user has different levels of privileges and roles they each perform.

5.3 Test Plan

Tests were performed to ensure that the system's objectives were met and that it functioned properly. The project's objectives were the testing points of the tests.

The primary testing strategy was unit testing, in which each project unit was tested individually. When the units performed as expected, the tests were approved.

5.4 Unit Tests

System tests	Results
1. Install xampp web server	Pass
2. Start Up all servers; Apache Web Server, MySQL	Pass
3. Run tests by using real data on forms and reports	Pass

6.0 References

Converse, T., Park, J., & Morgan, C. (2004). PHP5 and MySQL bible (Vol. 147). John Wiley & Sons.

Gardner, B. S. (2011). Responsive web design: Enriching the user experience. Sigma Journal: Inside the Digital Ecosystem, 11(1), 13-19.

Hesterberg, T. (2011). Bootstrap. Wiley Interdisciplinary Reviews: Computational Statistics, 3(6), 497-526.