**WEB-BASED STUDENT-TO-SUPERVISOR ALLOCATION AND ASSESSMENT SYSTEM**

**PREPARED BY:**

**RICHARD EGHENAYARHIORE EMMANUEL**

**CST20HND0558**

**SUPERVISED BY:**

**MR. ADEOYE, BAMIDELE ADEDAYO**

**NOVEMBER 2022**

**CHAPTER THREE**

**Methodology and Design**

**3.1 Introduction**

A methodology is a process of rigorous study or inquiry, particularly to unearth new facts or information; hence, research methodology should be good enough to enable the achievement of the specified objectives which are achievable using specific components, such as data collection and design procedures and system modeling (use case, activity, and class diagrams). This chapter contains the input/output specifications, and system requirements for the development of a student-to-supervisor allocation and assessment system.

**3.2 Methods of Data Collection**

Before developing any system, collecting data and facts about the existing system is critical to understand what is going on. This research was carried out using three methods.

i. Observation of the Work Environment

ii. Documentation

iii. Interview

**3.2.1 Observation of the Work Environment**

This method was employed to acquire information and data for this study by monitoring how the manual system worked. The most evident flaws in the existing system were discovered via detailed inspection. Using the observational approach, the context in which the observation is made can be modified in a variety of ways.

**3.2.2 Documentation**

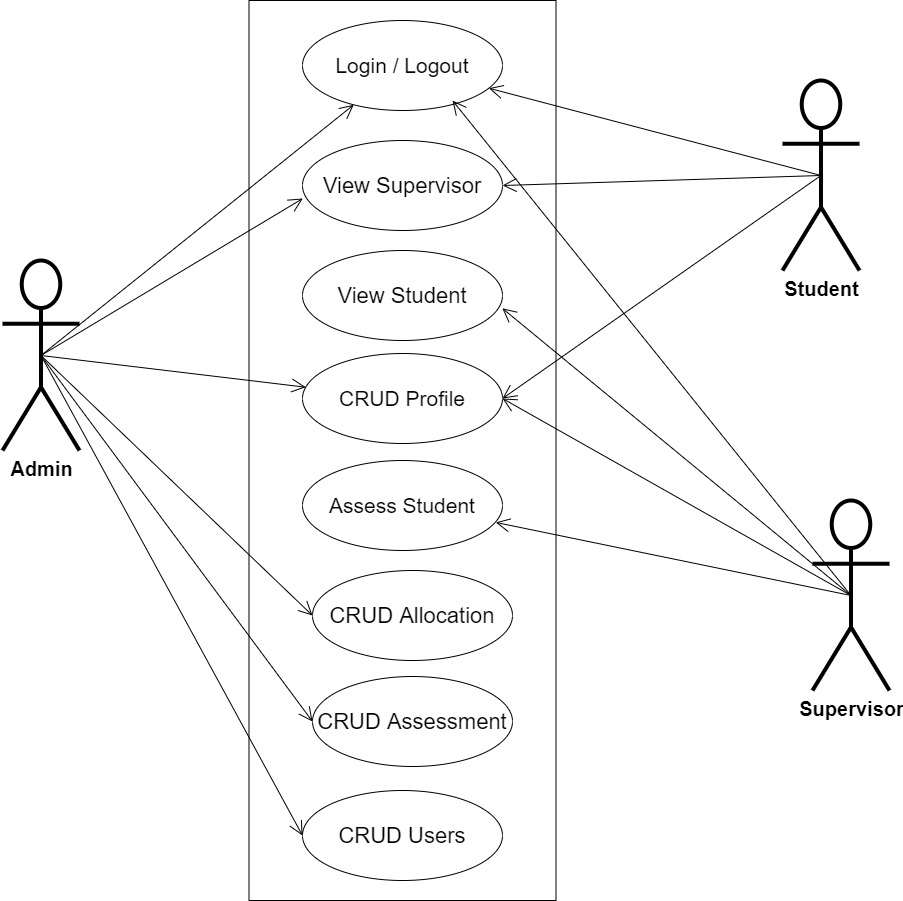
Documentation is a type of secondary data collection. This method makes use of journals, manuals, past work, publications, and other sources. This method of data collection is used because it allows for comparison with past studies. This includes the internet, which is a data collection tool. The internet was used to find information on difficult or ambiguous issues.

**3.2.3 Interview**

The major purpose of using interviews as a data collection approach is to collect data thoroughly and rigorously. The researcher met with the department's project coordinators and got reliable information based on the researcher's queries.

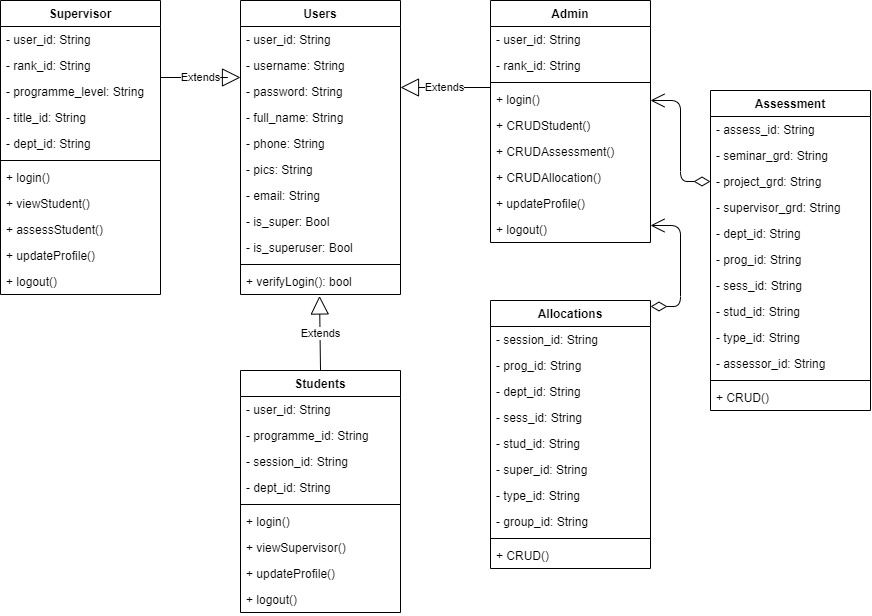
**3.3 System Modeling**

A system model is a conceptual model which is a result of system modeling that describes and represents a system. It is an interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the development of this modern system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

**Fig 3.1 System Use Case Diagram**

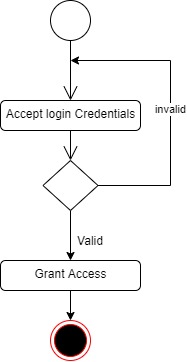
**3.3.2 Class Diagrams**



**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

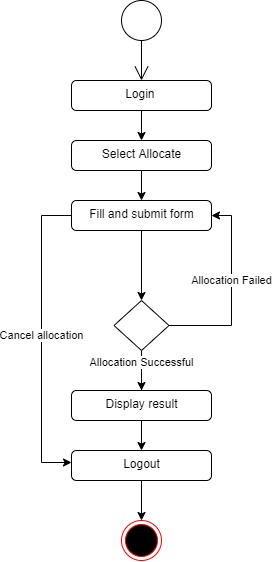
**Login**

The process for gaining access to the system is depicted in the diagram below; the email address and password must be accurate to gain access.

**Fig 3.3 System Login Activity Diagram**

**Create Allocation**

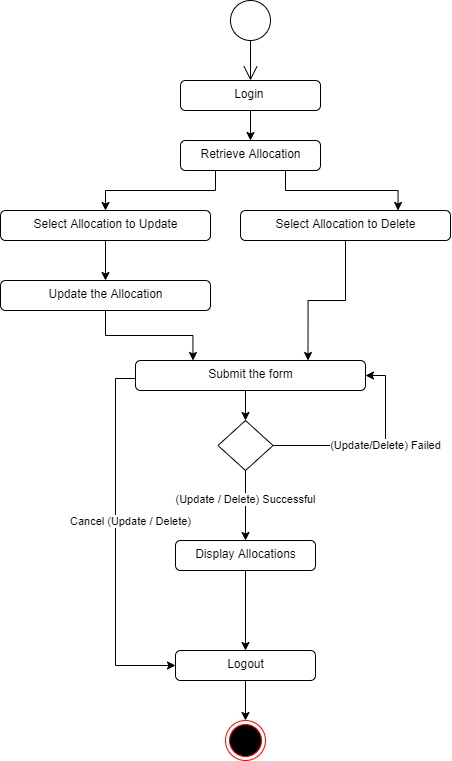
The process for automatically allocating students to their supervisors is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the allocation.



**Fig 3.4 Creating Allocation Activity Diagram**

**Updating / Deleting Allocation**

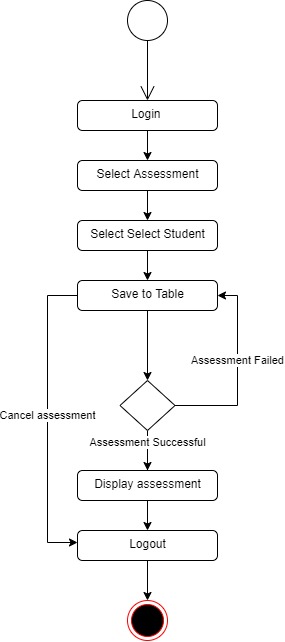
The process for updating/deleting allocation is manually done, as depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the allocation.



**Fig 3.5 Updating / Deleting Allocation Activity Diagram**

**Creating Assessment**

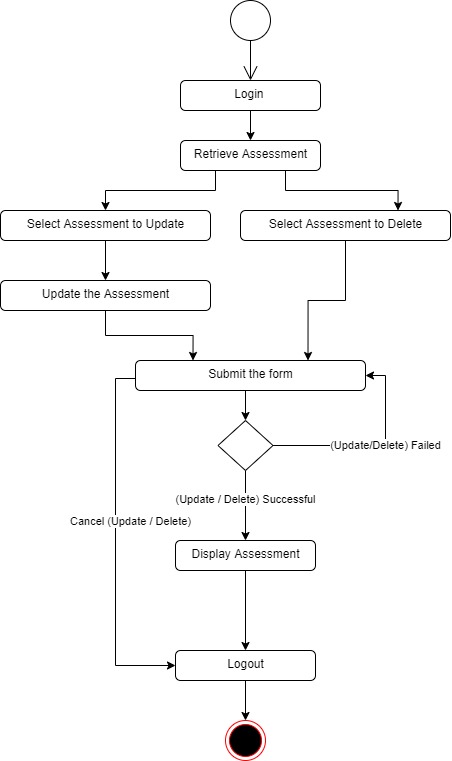
The process for assessing seminar and project defense for students is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the assessment.



**Fig 3.6 Creating Assessment Activity Diagram**

**Updating / Deleting Assessment**

The process for updating / deleting student seminar and project defense assessment is depicted in the diagram below; The system ensures that the user is authenticated and authorized to perform the assessment.



**Fig 3.7 Updating / Deleting Assessment Activity Diagram**

**3.4 Database Design**

The following are some of the input specifications used in this project work.

1. Users Table: contains the generic information of all system users.
2. Allocation Table: contains every system student-to-supervisor allocation information.
3. Assessment Table: contains every student assessment information

**Table 3.1 Users Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| user\_id | Varchar | No | PK | 32 | Unique string for identifying users |
| username | Varchar | No |  | 20 | Student regNo/Supervisor fileNo |
| password | Varchar | No |  | 128 | User Password |
| full\_name | Varchar | No |  | 60 | User full name |
| phone | Varchar | Yes |  | 11 | User phone number |
| pics | Varchar | Yes |  | 100 | User profile picture |
| email | Varchar | Yes |  | 100 | User email address |

**Table 3.2 Allocation Input Specification Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| allocation\_id | Varchar | No | PK | 32 | Unique string for identifying allocation |
| dept\_id | BigInt | No |  | 10 | Student department reference |
| prog\_id | BigInt | No |  | 10 | Student programme (ND/HND) reference |
| sess\_id | Varchar | No |  | 32 | student academic session reference |
| stud\_id | Varchar | No |  | 32 | Student registration number reference |
| super\_id | Varchar | No |  | 32 | Supervisor file number reference |
| type\_id | BigInt | No |  | 10 | Student type (Reg/Eve) reference |
| group\_id | BigInt | No |  | 10 | Student group number reference |

**Table 3.2 Assessment Input Specification table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Null** | **Key** | **Length** | **Description** |
| assess\_id | Varchar | No | PK | 32 | Unique string for identifying assessment |
| seminar\_grd | Int | No |  | 10 | Student seminar grade |
| project\_grd | Int | No |  | 10 | Student project grade |
| supervisor\_grd | Int | No |  | 10 | Student supervisor grade |
| dept\_id | Varchar | No |  | 32 | Student department reference |
| prog\_id | BigInt | No |  | 10 | Student programme (ND/HND) reference |
| sess\_id | BigInt | No |  | 10 | student academic session reference |
| stud\_id | Varchar | No |  | 32 | Student registration number reference |
| type\_id | BigInt | No |  | 10 | Student type (Reg/Eve) reference |
| assessor\_id | Varchar | No |  | 32 | Assessor file number reference |

**3.5 Output Design**

This declares and displays the outcome of the given input. This automated system's output is dependent on its input. The output specification is listed below.

**Table 3.3 Users** **output design table**

**PROJECT MANAGER**

**List of the System Registered Users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User\_id** | **Username** | **Full\_name** | **Phone** | **Email** | **Date\_registered** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.4 Allocation** **output design table**

**PROJECT MANAGER**

**List of Student to Supervisor Allocation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **allocation\_id** | **dept\_id** | **prog\_id** | **sess\_id** | **stud\_id** | **super\_id** | **type\_id** | **group\_id** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.5 Assessment** **output design table**

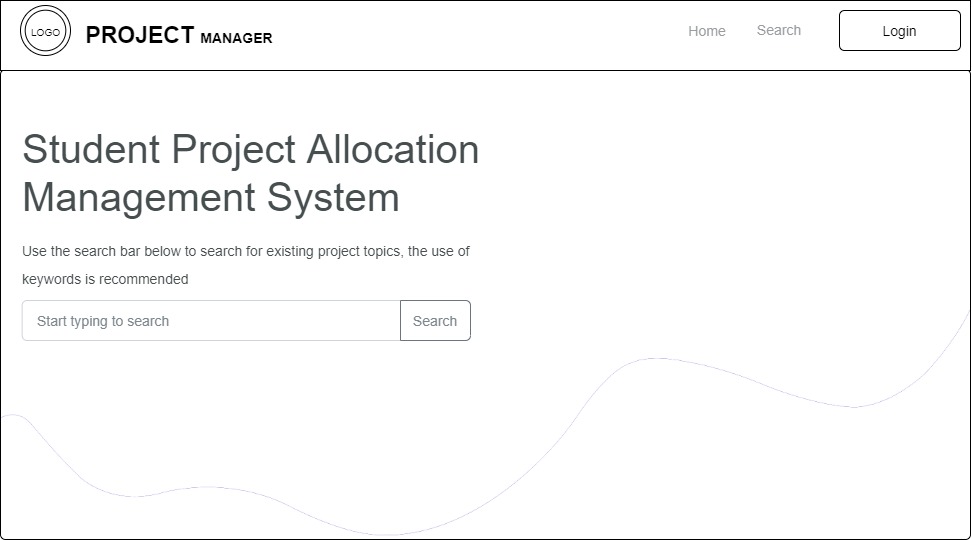
**PROJECT MANAGER**

**List of Student Defense Assessment**

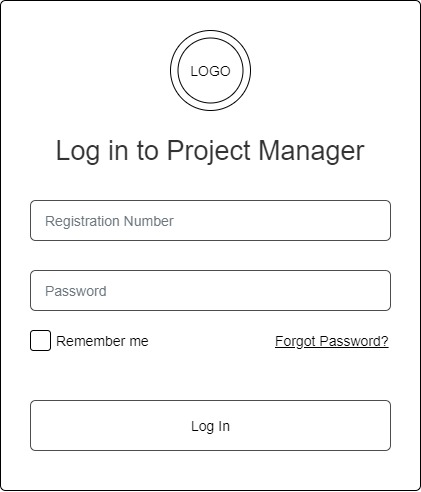
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **assess\_id** | **dept\_id** | **prog\_id** | **sess\_id** | **stud\_id** | **project\_grd** | **supervisor\_grd** | **seminar\_grd** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**3.6 Input & User Interface Design**

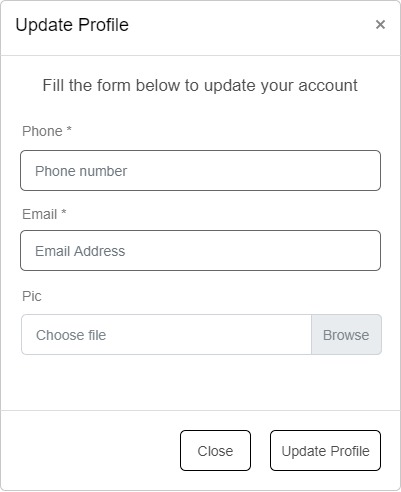
This shows a visual representation of the system interface; it will be made to be intuitive to use, quick to respond to, and visually appealing. Additionally, it will be properly protected, so authorization will be necessary to view some levels of the contents. A mid-fidelity wireframing application named Draw.io is used to assist with the designs.



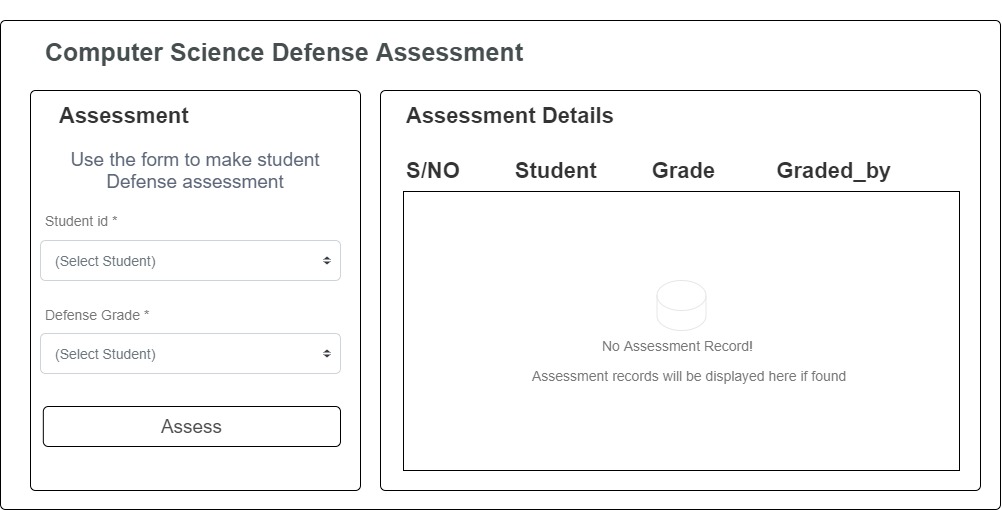
**Fig 3.8 Home Page**



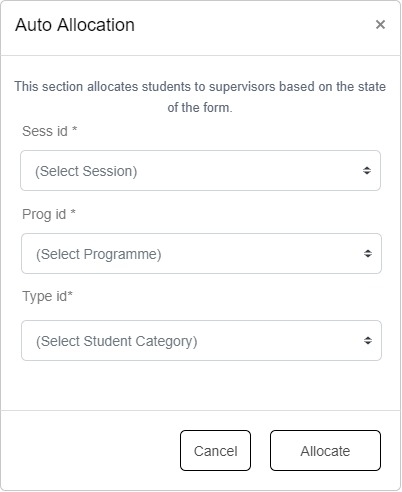
**Fig 3.9 Login Form**



**Fig 3.9.1 Update Profile Form**



**Fig 3.9.2 Defense Assessment Screen**



**Fig 3.9.3 student to supervisor Allocation Form**

**3.7 System Requirement**

For optimal performance, every software system developed has a specified system requirement on which it has been intended to run. The system requirements, on the other hand, are the minimum hardware and software requirements for the system's smooth functioning.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 2 GB of RAM (Random Access Memory).
2. Minimum of Intel Dual core processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 7 OS (Operating System).
2. Vs. Code IDE installation.
3. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This research work will be a web-based application and will be implemented on a relational database system (SQLite). HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript for the frontend development while Django (Python) will be employed for the backend programming.