# ASHESI UNIVERSITY

## A FACILITIES MANAGEMENT SYSTEM UTILIZING THE APPLICATION OF THREE-DIMENSIONAL MODELS, FOR ASHESI UNIVERSITY.

**Applied Project**

B.Sc. Management Information Systems

**Samuel Kofi Ntaama Armo-Himbson**

**2020**

**ASHESI UNIVERSITY**

# A FACILITIES MANAGEMENT SYSTEM UTILIZING THE APPLICATION OF THREE-DIMENSIONAL MODELS, FOR ASHESI UNIVERSITY.

**APPLIED PROJECT**

Applied project submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in Management Information Systems

**Samuel Kofi Ntaama Armo-Himbson**

**April 2020**

# DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

………………………………………………………………………………………………

Candidate's Name:

………………………………………………………………………………………………

Date: ………………………………………………………………………………………………

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied project laid down by Ashesi University College.

Supervisor's Signature:

………………………………………………………………………………………………

Supervisor's Name:

………………………………………………………………………………………………

Date: ………………………………………………………………………………………………

## Acknowledgement

Making it this far has is a blessing that I would like to thank the Almighty Lord for that. The journey has been extremely difficult, tiresome, and weary. However, He has given me success.

I also want to thank my supervisor, Mr. David Samper for his dedication and guidance towards this project. I was doubtful about my ability to complete this project, but with some words of encouragement he spoke and probably was not aware of, I got the motivation to make the project a reality.

Appreciation also goes to my parents for making provision and supporting me throughout for my four yearlong stay at Ashesi.

Finally, I thank everyone including my lecturers, my friends and loved ones for their support.

# Abstract

The act of maintenance is a critical factor in ensuring that the lifespan and efficiency of an object or a structure are maintained. Likewise, it is of much importance that organizations continuously pay close attention to the conditions of the environment in which their operations run. To attain maximum efficiency, every equipment that comprises a given environment must be appropriately maintained or taken care of. The field of facility management is a very well established one and utilized in various levels of organizations ranging from recreation centers to office complexes and even academic institutions. Facility Managers ensure that resources and assets acquired are appropriately looked after and periodically maintained.

This paper gives a walkthrough of a facility management system for Ashesi University. The system is designed to be a digital implementation of processes involved in managing facilities from the stage of reporting to assigning duties to respective parties for reparations. The adoption of the use of a three-dimensional modeling tool enables users to have a realistic feel of the school environment hence making it easier to make adequate reports consideration the exact location and type of fault. The main tools utilized in the development of the system include HTML, PHP, CSS, JavaScript, MySQL, and three.js, which is a JavaScript library for SD modeling. This project is intending to revolutionize the field of facility management, making it more inclined towards a digital representation.

# Table of Contents

# Chapter 1: Introduction

## 1.1 Introduction

The act of maintenance is vital in ensuring that facilities or structures serve their purpose for a relatively prolonged period [1]. These facilities or structures include buildings and equipment such as computers, cars, furniture, and many others. The availability of these facilities however may differ from place to place considering the purpose or function of the organization. For example, a hospital will have medical equipment that are tailored towards the sole purpose of diagnosing and monitoring issues regarding the human health, while that of an educational institution will have equipment that are needed in fostering education. As a result, it is important to take in measures to ensure that these structures and equipment work properly in order to bring enough value. One way to ensure relatively high efficiency in these facilities is through the practice of Facilities Management [2].

## 1.2 Background

Facility management encompasses all the activities or services that is performed to keep assets up and running in order to serve the grand purpose of a built environment [3]. Its importance is significant because the efficiency of an organization is linked to the physical environment in which it operates and how that environment can be improved to increase efficiency [4]. Findings describe facility management as a diverse field in terms of the activities involved in the process [4]. This may be as a result of the vast variety of organizations and the kind of processes that they undertake. Studies have shown that the field of facilities management is gradually shifting from the idea of cost reduction to value addition [5]. Studies conducted in various industries concluded that the role of facility management was essential to the industries gaining more value. Hence the same applies to its application in educational institutions. Research suggests that the success of an educational institution has facility management as one of its key influencers [4]. Thus, regarding facility management as an important component of the institution can have an impact on the performance of the students.

## 1.3 Problem

Ashesi University has a department that handles the complaints regarding issues that arise on the school campus. Complaints are brought in by students, staff and faculty and are addressed by the various departments that are equipped to handle those issues. Meetings with personnel from the logistics department have indicated their readiness to solve these problems. However, without any knowledge of the problem, no solution can be implemented. Thus, there is still the presence of some faulty equipment on campus. A common example captured from users through face to face interviews on campus, was the presence of faulty plugs all around campus. This hindered them from being able to plug in their electrical devices at those specific points to charge them for use. This problem brings about disruptions relating to their academic work. This problem affects mostly the students on campus and can occur anywhere on the school campus. The ability to attend to problems will help facilitate the provision of educational services which is the aim of the school. The solution proposed to solve this problem is to build a system that allows users of the school's facilities to have access to a platform that enables the reporting of faulty equipment on campus.  This will be of benefit in the long run to students, staff, and faculty as it will provide a conducive environment for teaching and learning to take place in the university [4].

## 1.4 Related Work

### 1.41 Design of user-driven facility management system

A project conducted by Grace Lorraine D. Intal, Rex Aurelius C. Robielos and Alysia Georgia B sought to create an inclusive user-driven facility management system for a university [4]. A survey was conducted in two universities in the Philippines to determine the adequate requirements for a high performing facility management system. Pertaining to the university that the system was built for, the areas of focus were reservation and monitoring of the university facilities, mainly the availability of its labs and lab equipment. The system that was implemented, served two purposes. The first was to provide the identified users with a platform purposely for monitoring lab equipment and the facilities of the university.

The survey of the system above was conducted with the participation of 398 respondents and 354 respondents each from two universities. Findings from the survey indicated some activities or processes that will have to be considered when building the system. These processes include the borrowing of lab equipment, the reservation of facilities, receiving borrowed lab equipment, accounting for the loss and destruction of equipment and tools, and the reservation of laboratory facilities and equipment outside of class hours.

Results from the testing of the system showed that the users picked performance as their highest satisfying factor while that of responsiveness and competence of the system were ranked the lowest. Upon comparing the incumbent system and new system deployed, users attested to the fact that there was a rise in the performance of the new system hence its acceptability.

1.42 IKEA

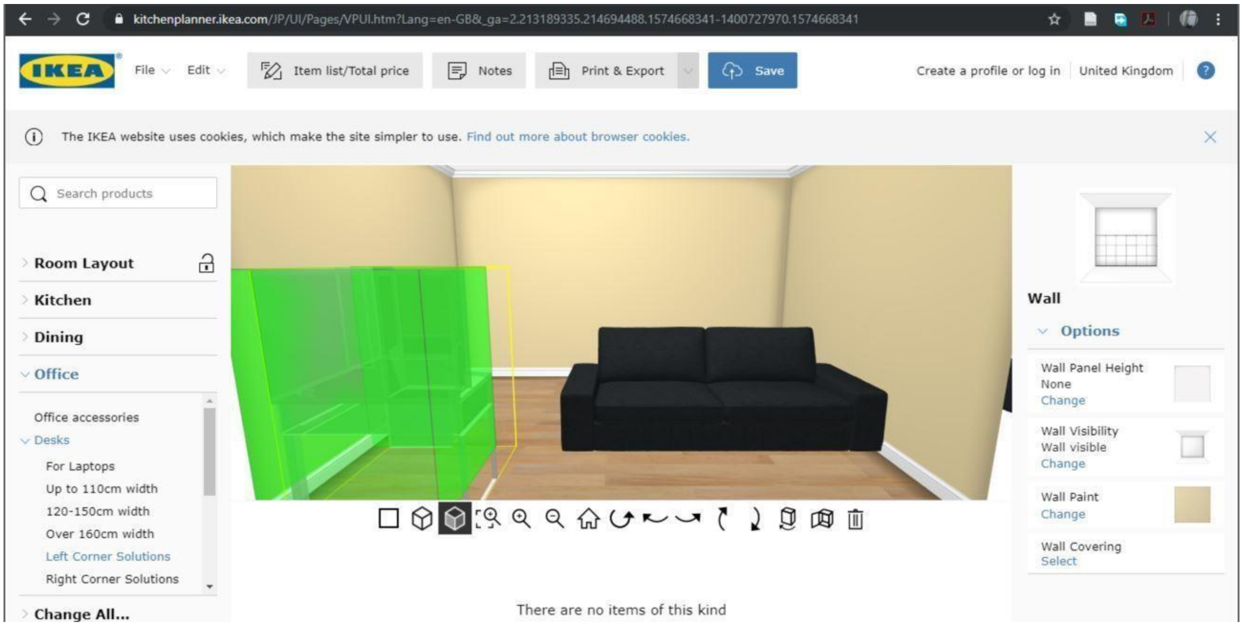


*Figure 1.1*

IKEA is a company with a focus on providing room furniture to its customers. It applies a modern technique to its online customers by making their online shopping experience seem as though it is a real-life event. The implementation of the service comes through a web-based application that incorporates the use of 3d technology into its website. This is done using 3d objects that represent the type of furniture that the customer is looking for.

The online customer begins by selecting the room type using pre-generated shapes or by inputting dimensions of a specific rom which will then be used by the system to generate the virtual room. Once this process has been successfully completed, a display of the room is shown on the user interface. The user can drag objects into the room from a category of items on the sidebar. Other functionalities apart from dragging and dropping objects include rotating the objects, repositioning them and changing the color of the object selected[6].

The relevance of this related work is its approach to delivering its 3d environment to the class of users IKEA identified. It also takes into consideration the target users by allowing more precision to the user's desired room size.

1.43 Ashesi University

Another related work is that Ashesi University. The University receives complaints using emails. After the emails have been received, the issues are analyzed and then forwarded to the required department to fix the issue. This project will involve the building of a web application which will serve the purpose of providing users with an avenue for receiving records concerning equipment that seem to be malfunctioning.

# Chapter 2: Requirement Analysis

## 2.1 Data Collection: Interview

In considering the stakeholders for the web application, two classes of users were identified. The first class of stakeholders was the students from Ashesi University and the second was the staff of Ashesi University. Face to face interviews were conducted with students and staff of the IT department of the University. From the interviews, basic requirements of the application were established.

The meeting with the logistics department of Ashesi University was to find out whether issues on faulty equipment get reported, how these reports are handled, and a suitable structure for a potential facilities management system to be implement. Findings from this meeting revealed that issues on faulty equipment around campus were indeed reported to the department. The meeting also revealed that there was an existing solution/system similar to this project scope, which uses email services and a web application. However, this project seeks to adopt the use of web-based application which will provide a broader range of functionalities as compared to the already existing system. There was also the exchange of information on how the system works. The structure of the available system works in a way that students or staff send reports in the form of emails to the Ashesi support center. These mails are then forwarded to the required department responsible for handling the issues.

Interviews with students were undertaken to find out whether they reported issues on faulty equipment on campus and how they went about it if they did. The responses received indicated that there were some students who reported the issues and others who did not report issues on faulty equipment. Those who did either reported them via emails to the support center or made it known to facility managers around the school campus. The students who did not reports issues were either unconcerned, unwilling, or simply just did not know where or how to make these issues known to the appropriate personnel. This brought about the basic requirements of the web application.

## 2.2 General requirements of the system

1. Display virtual halls on campus with three dimensional models
2. Collect and display data on issues reported by users
3. Administrator portal to access issue logs
4. Generate analytics ie. Pie charts

## 2.3 Operating Environment

The programming languages that will be adopted include HTML, CSS, PHP, JavaScript, and possibly a 3d rendering software such as AutoCAD for the development of 3d objects.

## 2.4 Functional Requirements

This section contains the functionalities that will be available for the different types of users.

### 2.41 User 1: General User

General users encompass anyone who wants to make a complaint concerning any faulty equipment on campus. These users may include students, staff, faculty, and anyone with access to the school's intranet.

These users can;

1. Report issues by clicking on corresponding 3d object

2. Report issues by typing text into text area if object is not available

2.42 User 2: Administrator

The administrator is any user that has the authority to oversee the complaints that the system will be receiving from users. The user will have a special case to log in to be able to assume the roles of the system administrator.

The administrator will be able to;

1. Have a user account
2. Log in and log of account
3. View issue logs for specific periods of time
4. Report issues
5. Assign issues to workers to handle

2.5 Non-functional requirements

2.51 Usability Requirement

The users of the system must find it easy to navigate. There will be a user interface that users will be able to interact with making it possible to operate the system.

2.52 Efficiency Requirement

Efficiency should be considered when analyzing how the application is going to function. It should be able operate without any setbacks to the user’s devices or applications. An example of a setback is the system displaying errors or sudden crashes.

2.53 Availability Requirement

This requires that as long it is hosted on the server, the web application should be available to its users all the time.

## 2.6 Use Cases

| Use Case: Login | Primary Actor: Administrator ID: 001 Priority: High |
|---|---|
| Stakeholders: | Staff |
| Description: | This use case describes the ability of the administrator to login to the system. |
| Goal: | The goal is for the administrator to be able to fully access his/her account via login details. |
| Success Measurement: | Success on login into the system and operating it from the administrator side. |
| Precondition: | Administrator must know login details and system should have it stored in database |
| Trigger: | The admin clicks on the login button |
| Relationship: | |

| | |
|---|---|
| Event flow: | 1. Access the web app with a browser<br>2. Locate login button and click on it<br>3. Enter username and password<br>4. Click the submit button<br>5. The system compares the details with that of the database<br>6. If correct, admin is then logged in |

| | |
|---|---|
| Use Case: Report Issue via 3d object | Primary Actor: User   ID: 002   Priority: High |
| Stakeholders: | Staff, Faculty, Student |
| Description: | This use case describes the ability of the user to report issues using the 3d aspect of the system. |
| Goal: | The goal is for the user to be able to report issues to the required department. |
| Success Measurement: | Success on having the report transferred to the support system |
| Precondition: | User must be able to access site |
| Trigger: | The admin clicks on the 3D object |
| Event flow: | 1. Access the web app with a browser<br>2. Scroll through locations to find your choice<br>3. Click on location<br>4. The virtual location is generated<br>5. Click on the 3d object that corresponds to the faulty equipment<br>6. System then generates details from the object clicked<br>7. Issue is logged<br>8. Issue if forwarded to support center<br>9. Saved into database |

| | |
|---|---|
| Use Case: Report Issue via text box | Primary Actor: User   ID: 003   Priority: High |

| | |
|---|---|
| Stakeholders: | Staff, Faculty, Student |
| Description: | This use case describes the ability of the user to report issues by regular text |
| Goal: | The goal is for the user to be able to report issues to the required department. |
| Success Measurement: | Success on having the report transferred to the support system |
| Precondition: | User must be able to access site |
| Trigger: | The admin clicks on a button |
| Event flow: | 1. Access the web app with a browser<br>2. Click on the report issue button<br>3. Text box is generated<br>4. User types in the details<br>5. Issue is logged<br>6. Issue if forwarded to support center<br>7. Saved into database |

| | |
|---|---|
| Use Case: View Issue log | Primary Actor: Administrator ID: 004 Priority: High |
| Stakeholders: | Staff |
| Description: | This use case describes the ability of the administrator to view the log of issues reported. |

| | |
|---|---|
| Goal: | The goal is for the administrator to be able to fully access the log of issues and their details. |
| Success Measurement: | Success on accessing the log of issues. |
| Precondition: | Administrator must login to system |
| Trigger: | The admin clicks on a button |
| Event flow: | 1. Access the web app with a browser<br>2. Locate login button and click on it<br>3. Enter username and password<br>4. Click the submit button<br>5. The system compares the details with that of the database<br>6. If correct, admin is then logged in<br>7. Admin clicks on the issue log button<br>8. System retrieves information from database<br>9. Information is displayed |

## 2.7 Use case diagram

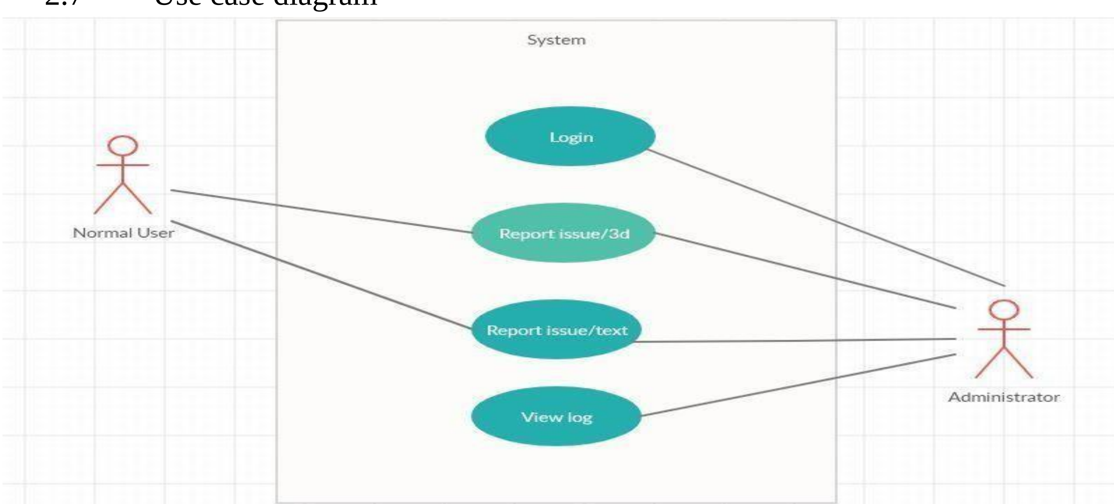


*Figure 2.1*

# Chapter 3: System Architecture

## 3.1 Overview

The main architecture that the system will adopt is the Model-View-Controller framework. This framework mainly used for building websites uses an organized approach by separating its logical components. It is also adopted by developers with the aim of building systems that require the development of a Graphic User Interface (GUI) [7]. Research also stresses on one of its key benefits such as view synchronization [8]. The components, as the name suggests, include the model, the view, and the controller. The three components help in the overall organization of code and the entire system. This is important because in the long run, maintenance or debugging is made less complex as the developer has his or her codes developed in a structure. It is also very flexible and has the design capability to support multiple platforms [9]. Based on these reasons, this architecture was considered the best option.

## 3.2 Model

The model component of the system MVC framework handles all instances that has data involved. More specifically, the model is responsible for managing data that is required by the software. This includes the housing and the manipulation of the data[7Hansen]. This data is generated by user interaction in instances where it is required by the application for further processing.

3.3 View

The view component of the MVC framework is used to represent the user interface of the application. This refers to the front-end of the application where users can make interactions. Some features likely to be noticed with this component are text boxes, buttons, text, and many others depending on the functionalities of the application.

3.4 Controller

The controller acts as the bridge between the model and the view components. This includes more specific functionalities that control how data is manipulated, how the system is required to handle all the various sorts of interactions and events and make translations in the required application logic [10].

3.5 MVC Diagram

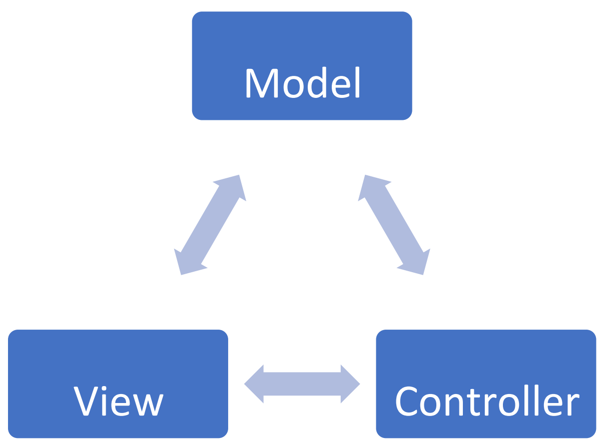


*Figure 3.1*

3.6 Frameworks

For the purpose of this project, the frameworks that are going to be used include three.js, Ajax and bootstrap. Three.js is a JavaScript library for creating three-dimensional objects, with its implementation using WebGL [11]. Three.js will be used to develop the 3dimensional features of the system that will be displayed on the website. The programming languages to be used include HTML, CSS, and JavaScript. There is also going to be the use of MYSQL for the creation and management of the database that is required for the application. There will also be the use of xampp, which will set up local server to run the system on during development.

Summary

| | |
|---|---|
| Front End | HTML, CSS, Bootstrap, Three.js |
| Processing | JavaScript, PHP, Ajax |
| Back End | MySQL, Xampp |

3.7 Database Design

The web-based application is supported by a database that stores that stores the relevant information needed for the site. This information includes administrator details, specifically the set username and password. It will also store the various categories of issues that can be reported using the application and specific details concerning those issues such as the time and date as of when the report was generated.

Based on that information, the tables that the database will have include;

3.71 Issue log table

This table stores anything that has to do with reports. This includes the location of the issue, the kind of issue reported, the name and contact of the user who reported it, a column reserved for who the job has been assigned to, a column for the status of the issue and the unique token assigned to that issue reported. The primary for this key for this table is the issue id since it is unique for all the entries.

3.72 Employee Table

This table keeps records of the contact information of stakeholders who will be appointed to handle a faulty equipment report. The table contains the columns for the worker id, worker name and the worker's email address. This table has the primary key to be the id of the worker. The Id is a autogenerated number assigned to every new entry into the table. Hence it can be used to uniquely identify the workers.

3.73 Job Assign Table

This table keeps track of the various issues that have been assigned to specific workers. The workers and issues at hand will be populated from the existing issue log and employee tables. This table also contains the date of assignment and date of completion. Figure 4.1 shows the database schema.

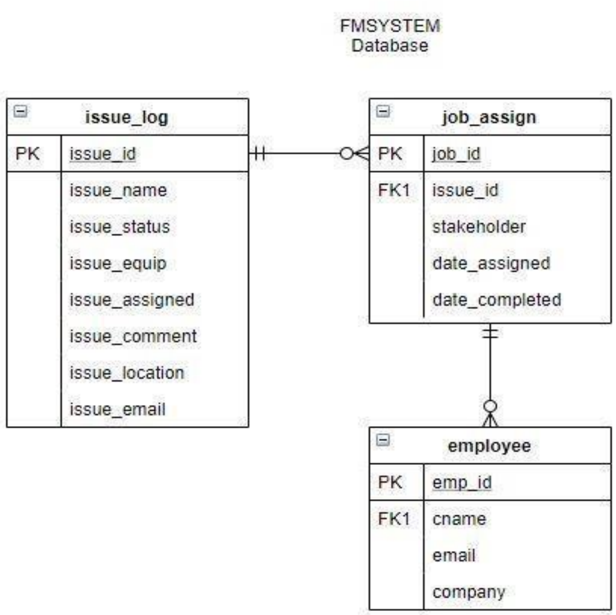


*Figure 3.2*

## 3.8 Activity Diagrams

### 3.81 General User Activity Diagram

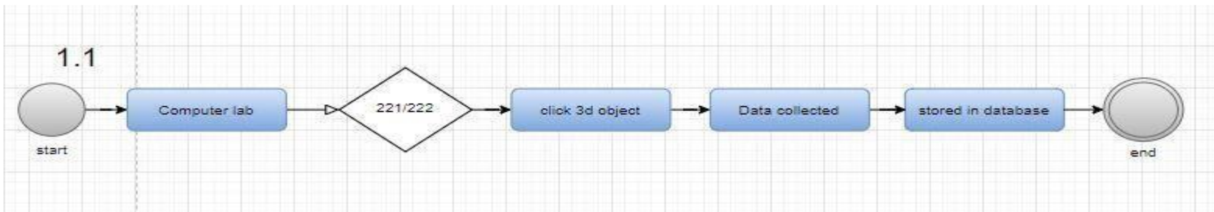


*Figure 3.3*

In figure 5.1, the user wants to report an issue on the computer lab. The user then logs onto the website and clicks on computer lab on the index page. The user then selects which lab he or she wants to report the issue on by clicking one of two options. The 3d model is loaded, the issue is reported, and data is sent to the database.

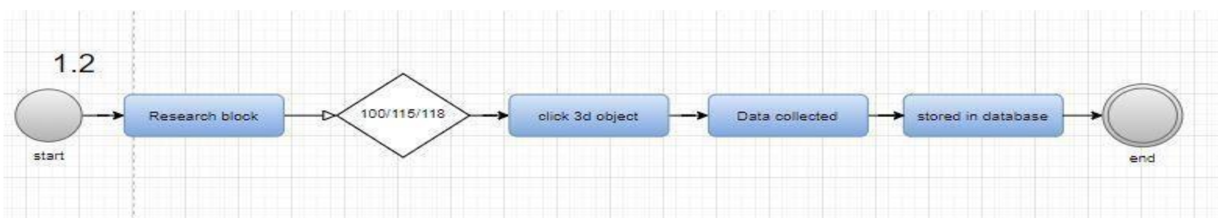


*Figure 3.4*

In figure 6.1, the user wants to report an issue on the research block. The user then logs onto the website and clicks on research on the index page. The user then selects which room he or she wants to report the issue on by clicking one of three options. The 3d model is loaded, the issue is reported, and data is sent to the database.

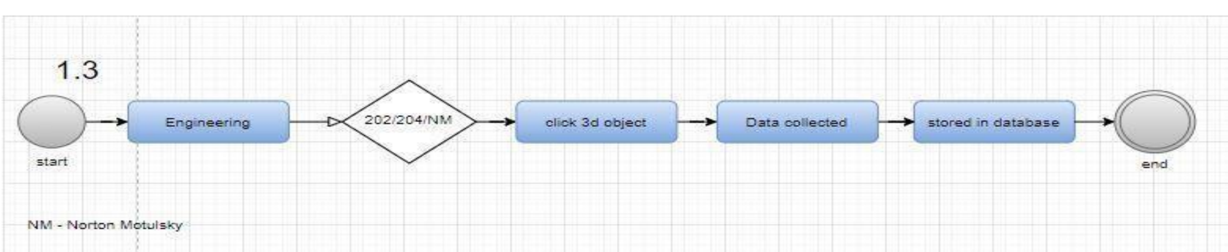

*Figure 3.5*

In figure 7.1, the user wants to report an issue on the engineering building. The user then logs onto the website and clicks on engineering on the index page. The user then selects which room he or she wants to report the issue on by clicking one of three options. The 3d model is loaded, the issue is reported, and data is sent to the database.

3.82Administrator Activity Diagram

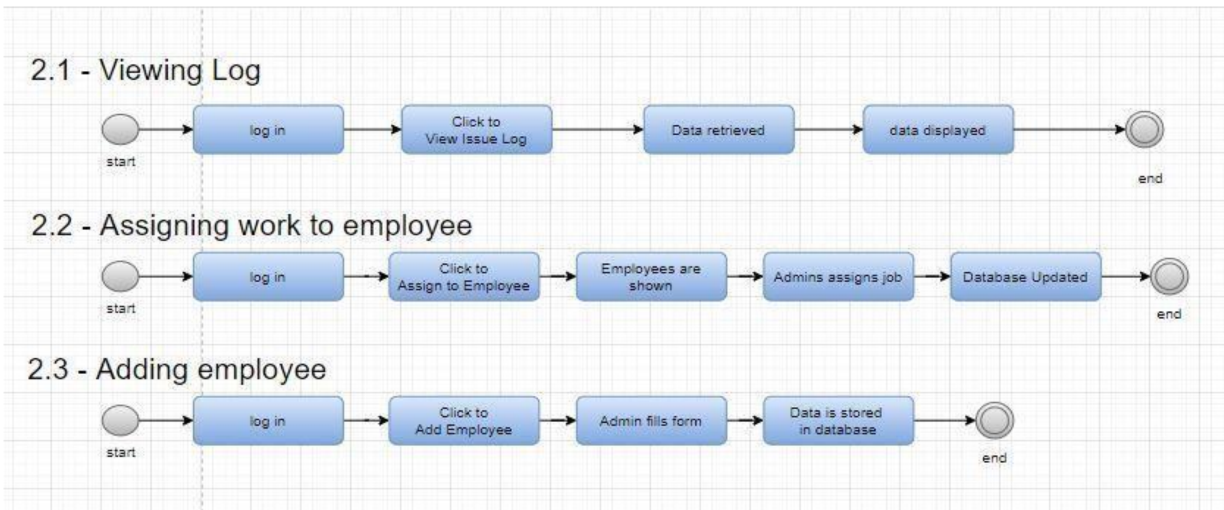


*Figure 3.6*

In figure 8.1, a system administrator is trying to view the issue logs. The administrator visits the website and logs in. After a successful login, the administrator navigates to the view issue log button, clicks and the system will have the log displayed.

In figure 2.2, a system administrator is trying to assign a pending issue to an appropriate stakeholder. After a successful login, the administrator navigates to the view issue log button. A form is then displayed beside the log. The administrator then assigns a job by inputting the required details in the form.

In figure 2.3, a system administrator is trying to add an employee to the database. After a successful login, the administrator navigates to the add employee button and inputs the required details in a form.

### 3.9 Other Information

Visual Studio code will be the main source code editor. This application was preferred due to its flexibility and friendly working environment. Browsers that have the capacity to support the system include Google Chrome, Safari, Firefox and Opera.

# Chapter 4: Implementation

## 4.1 Overview

This chapter provides an in-depth description of the various processes that were involved in transferring the system from the design stage to a functional working unit. It involves the processes that have been undertaken to transform the requirements discussed in chapter 2 into a practical application.

In order to begin implementation, it was required to install Node.js as this would be an important requirement in order to run live JavaScript codes on the web browser. The three.js JavaScript library was also downloaded from the official website which provides unlimited access and tutorials for developers. It was also required the folders to be created and organized to fit the Model-View-Controller structure. Figure 9.1 shows a snapshot of the arrangement.

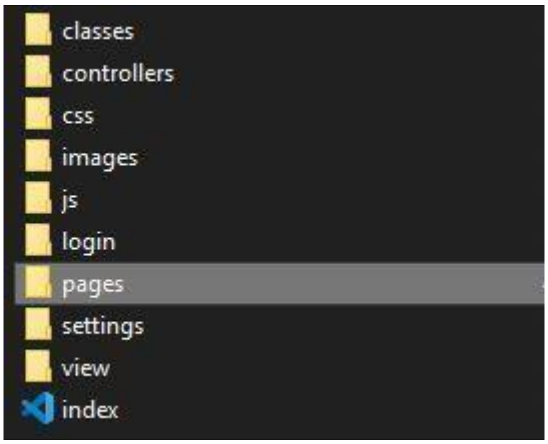


*Figure 4.1*

4.2 System Features

4.21. Interaction with 3d Models the feature displays some venues in the university as 3d models that allow users to navigate through and interact with it to report damaged equipment. Users can rotate, zoom in and zoom out with the use of their mouse. It also processes onclick, and mouse hover events. The snapshot of code below shows the code implementation for an onclick event of a three-dimensional object model on an air-condition.

```
//click A/C 1
var domEvents = new THREEx.DomEvents(camera, renderer.domElement);
domEvents.addEventListener(cubeac1, 'click', event =>{
    var txt;
    var person = prompt("Please enter your name:", "Harry Potter");
    var personb = prompt("Clicked on A/C2.Please enter comment:", "Fault type");
    if (person2 == null || person2 == "" || personb == null || personb == "") {
        txt = "User cancelled the prompt.";
    } else {
        ac1data.username = person;
        ac2data.comment = personb;
        var data = JSON.stringify(ac1data);
        console.log(data);
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function(){
        if (this.readyState == 4 && this.status == 200) {
            console.log(this.response);
        }
    };
    xhttp.open("GET", "output.php?data="+data, true);
    xhttp.send();;
    }
});
```

*Figure 4.2*

4.22Issue Log forms for unavailable locations

The log form is an available feature that allows users to fill a form instead of reporting issues using the 3d models. This feature also serves as back-up to allow users to report on venues that do not have corresponding 3d models. It also provides a platform for users to be more specific in the issues that they want to report. This feature was implemented using html and php. HTML was used to handle the outright look of the form and PHP was used to retrieve the data from it. Below is a snapshot of code used to develop the form and the form itself.

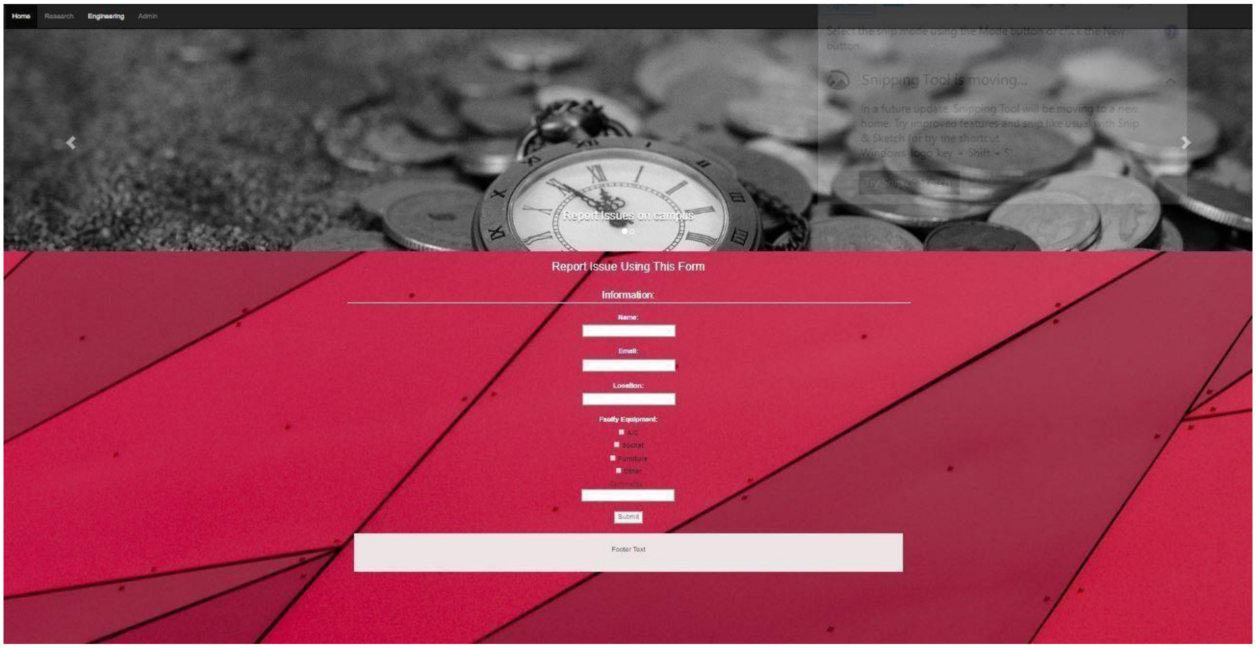


*Figure 4.3*

LogForm.html ×

LogForm.html > html > body > div#myCarousel.carousel.slide > div.carousel-inner > div.item > img

```html
<div class="container text-center">
  <h3 style ="color: white;">Report Issue Using This Form</h3><br>
  <div class="row">
    <form action="pages/insertFunction.php">
      <fieldset>
        <legend style ="color: white;" >Information:</legend>
        <label style ="color: white;" for="fname">Name:</label><br>
        <input type="text" id="loc" name="name" value=""><br><br>
        <label style ="color: white;" for="mail">Email:</label><br>
        <input type="text" id="loc" name="email" value=""><br><br>
        <label style ="color: white;" for="fname">Location:</label><br>
        <input type="text" id="loc" name="loc" value=""><br><br>

        <label style ="color: white;" for="Equipment">Faulty Equipment:</label><br>
        <input type="checkbox" id="type1" name="type" value="A/C">
        <label for="vehicle1"> A/C</label><br>
        <input style ="color: white;" type="checkbox" id="type2" name="type" value="Socket">
        <label for="vehicle2"> Socket</label><br>
        <input style ="color: white;" type="checkbox" id="type3" name="type" value="Furniture">
        <label for="vehicle3"> Furniture</label><br>
        <input style ="color: white;" type="checkbox" id="type3" name="type" value="Other">
        <label for="vehicle3"> Other</label><br>

        <tr>
          <td> Comments : </td><br><td><input type="text" name="comment"></td>
        </tr><br><br>
        <input type="submit" value="Submit">
```

*Figure 4.4*

## 4.23 Login feature

The login feature is available for system administrators. Logging in grants administrators access to an administrator's page where they have access to functionalities that the system restricts normal users from accessing.

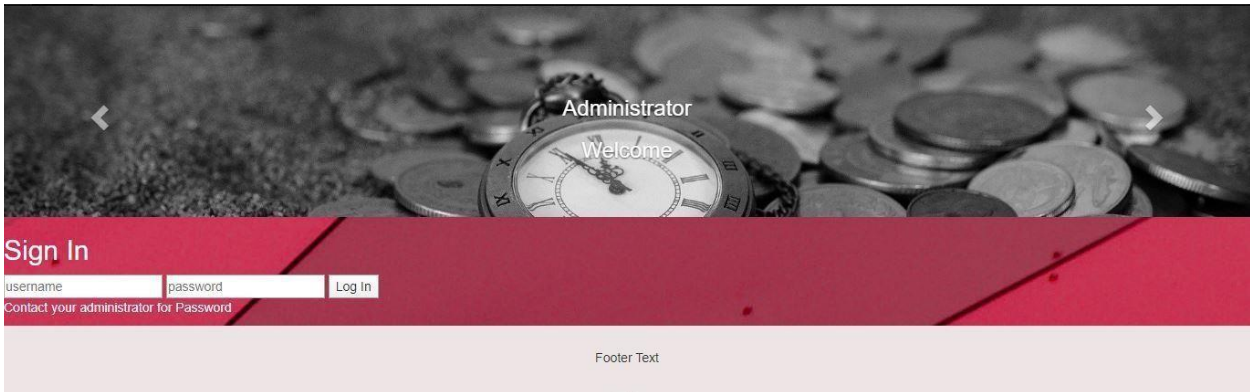


*Figure 4.5*

4.24 Display data

The type of data displayed here is the issue log. The issue log contains specific details about issues reported, such as the location of the problem, the device or equipment at fault, and the time of its report. The image below is a snapshot that shows data being displayed from the database.

| Id Number | User Name | User Email | Location | Equipment | Comment | Assigned | Status |
|---|---|---|---|---|---|---|---|
| 1 | Kofi | @ | Lab 222 | Projector | Fault type | | |
| 2 | Sam Himbs | sntaama@gmail.co | Lab 222 | Projector | Poor display | Samuel Quansah | Pending |
| 3 | | | Lab 222 | Table r2 | | | |
| 4 | Samule Qwerty | sam@gmail.com | Lab 222 | Table r3 | broken leg | | |
| 5 | Aaba | aba@hotmail.com | Lab 222 | A/C 2 | Leakage | Samuel Quansah | |
| 6 | Harry Kingston | hk@gmail.com | Lab 222 | Table r3 | broken leg | | |
| 7 | Elsa Stone | estone@ymail.com | Lab 222 | A/C 6 | Blowing warm air | Peter Arhin | |
| 8 | Merry Boakye | m.boakye@ashesi.ed.gh | Room 202 | Projector | Faulty dsiplay | Samuel Quansah | Completed |
| 9 | Merry Boakye | m.boakye@ashesi.ed.gh | Room 202 | Projector | Faulty dsiplay | Fred Dompreh | Pending |
| 10 | Samuel Himbson | samuel.himbson@ashesi.edu.gh | Lab 221 | Table r2 | Broken leg | | |
| 11 | Kevin Deyoungster | kafui@gmail.com | Lab 221 | A/C 3 | Leaking | | |
| 12 | Harry Potter | m@gmail.co | Lab 221 | Table r3 | Fault type | | |
| 13 | Mike Okyere | youngthello@gmail.com | Lab 221 | Stand | Damaged socket | Fred Dompreh | Pending |
| 14 | Joy Iusedtoknow | byejoy@6feet.com | Room 202 | A/C 1 | Its gone | | |
| 15 | Esi Parker | ayparker@ashesi.edu.gh | Room 202 | Back Door | Broken Handles | Fred Dompreh | Completed |
| 16 | Derek Rona | rona.derek@ashesi.edu.gh | Room 202 | Front Door | Broken Glass | Fred Dompreh | Pending |
| 17 | Victor Stone | stone@ashesi.edu.gh | Room 202 | Back Door | screeching noise | Samuel Quansah | Pending |

*Figure 4.6*

4.25Assign to worker

The feature allows the system administrator to assign issues to relevant stakeholders to work on them.

```
function assignworker(){
    echo '<select>
    <option>Select</option>';

    $sqli = "SELECT * FROM employees";
    $result = mysqli_query($con, $sqli);
    while ($row = mysqli_fetch_array($result)) {
    echo '<option>'.$row['Employee'].'</option>';
    }

    echo '</select>';
}
```

*Figure 4.7*

4.26View progress on work

The feature allows the system administrator to view the progress that stakeholders are making considering fixing the problem they have been assigned.

4.27Charts

The charts section of the system that provides some analytics based on the data recorded in the system. Information the charts display includes the number of issues reported from the various rooms, the number of issues reported on various specific equipment generally and for specific rooms available. This functionality is only available for administrators.

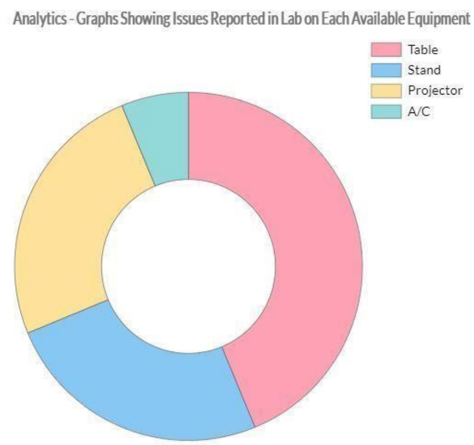


*Figure 4.8*

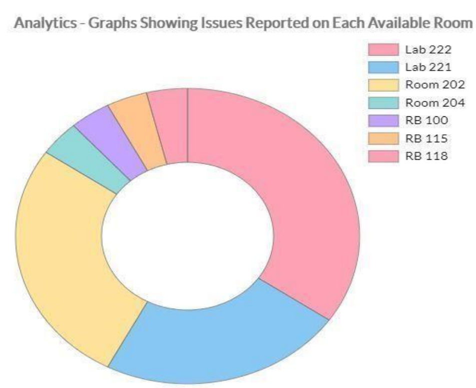


*Figure 4.9*

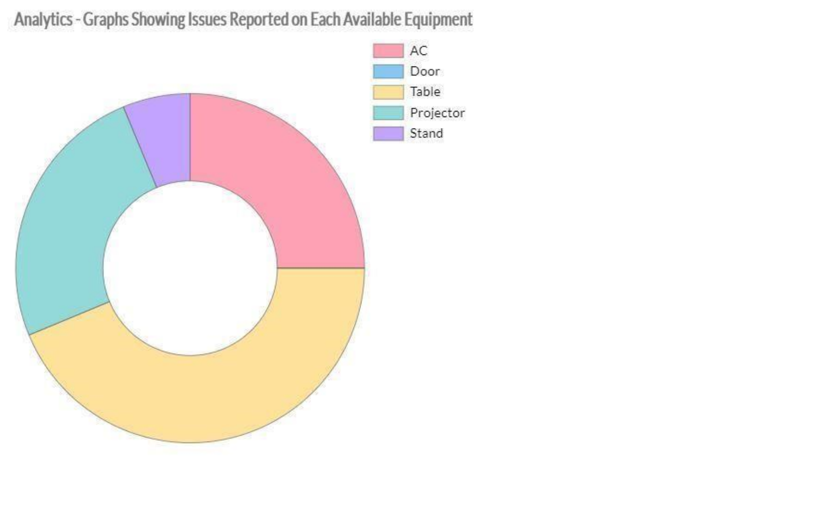


*Figure 4.10*

## 4.3 Front End

The front-end view is designed to keep the appearance of the website a little bit identical to that of the University's main website. This was to ensure consistency, allowing the users to have a feeling of familiarity when using the web application. It comes with a standard design that runs throughout all the pages of the web application. The use of colors such; red, white, and black was adopted because of the evenness these colors produce. Tools and libraries employed for this purpose are HTML, CSS, and Bootstrap. Also, Three.js is a library utilized for the creation of the 3d objects. 3D objects were also provided with color codes in order to make the identification of objects much easier. The color code for the air-conditioner and projector is dark-grey and that of the furniture including the stands and tables is brown.

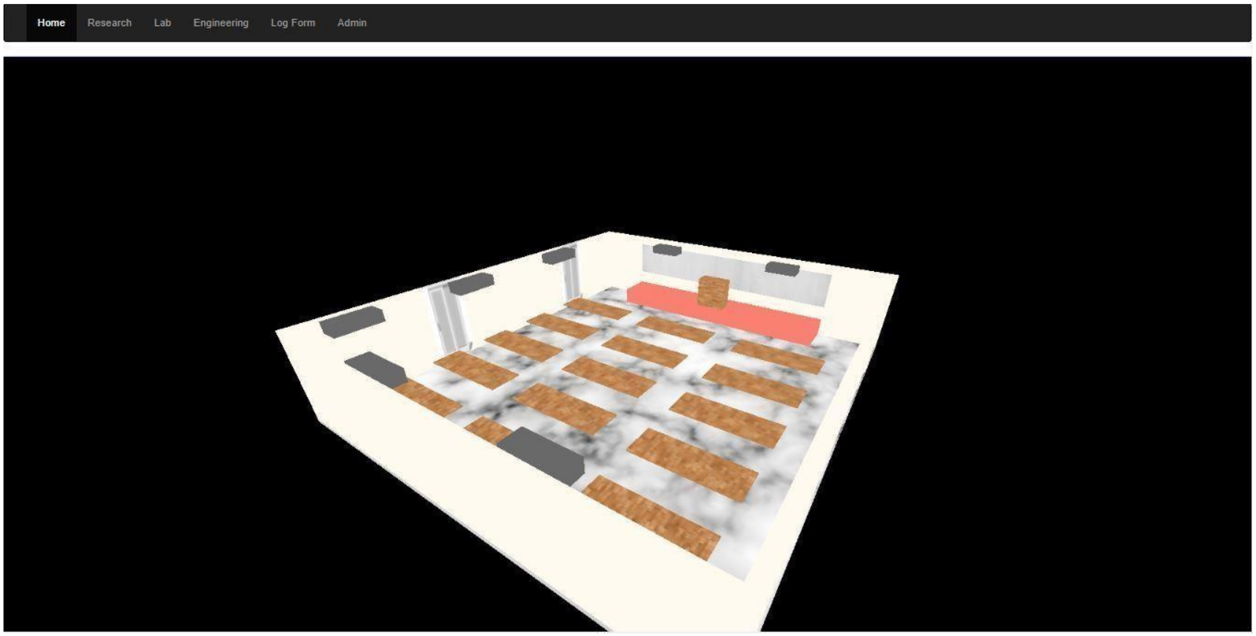


*Figure 4.11*

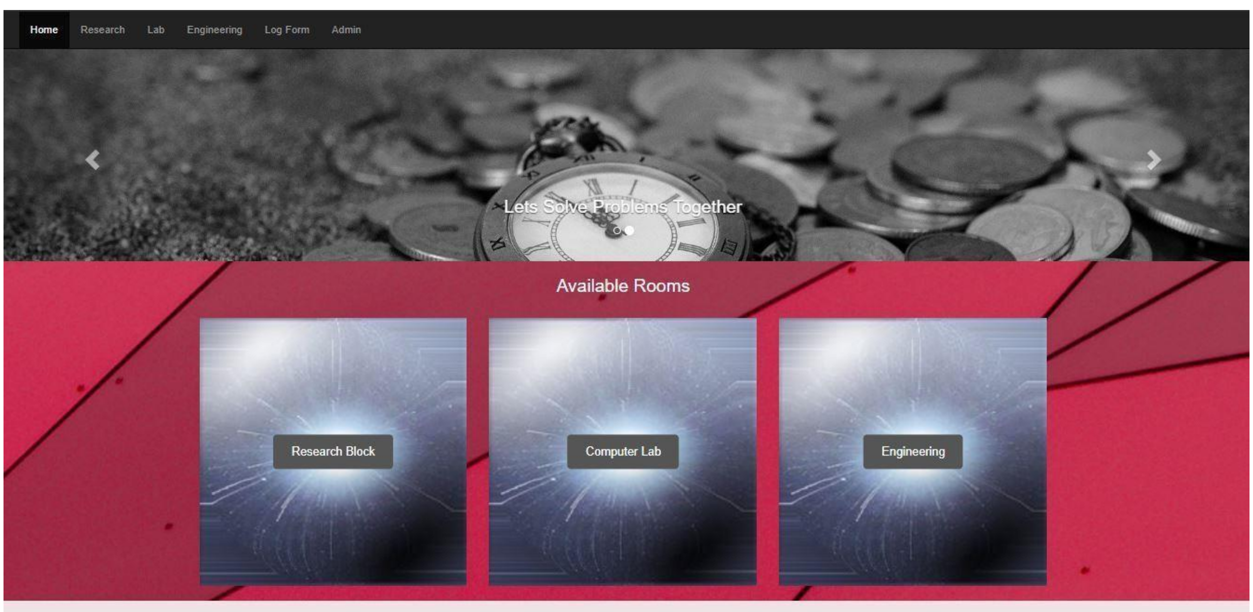


*Figure 4.12*

## 4.4 Processing Section

Implementation of the web application involved the use of data processing and the movement of data. Examples include the collection of data from the forms and 3d objects to be transferred to the database. The data being referred to includes user's full name, room numbers, the equipment at fault and the type of fault the user reported. Also, there were other processes, such as navigation through pages and the rendering of 3d objects. Tools and libraries applied for this purpose included JavaScript, PHP, and Ajax.

## 4.5 Back End

The leading player of the back end is the database of the system. This was created for recording data that the system generates from its operations. The primary tool used for the section is the MySQL database. Xampp was also applied during the implementation of the database to allow the local machine to act as a local server for the application. In applying Xampp, the modules for Apache and MySQL had to be run in order to establish the local server.

# Chapter 5: Testing

## 5.1 Overview

The chapter talks about the final stage of the development of the system which is testing. The process of testing involves the evaluation of a system or program to ensure that it meets the specified results and specifications [12]. In regard to this project, the following forms of testing were undertaken, development testing, release testing and user acceptance testing.

## 5.2 Development Testing

Development testing was mainly conducted during the implementation of the system. This required the reviewing of the code in order to fix bugs or make sure that it is performing a task as it should. This stage of testing is further broken down into three separate phases. They are unit testing, component testing and system testing.

### 5.21 Unit Testing

In this phase of testing, it involved the basic units of the system such as the functions that were developed, the classes used and the database queries that the system deployed.

### 5.22 Component Testing

Component testing was done afterwards, and this involved testing a cluster of units that function together to make up a module. The modules were tested to ensure that they conform seamlessly to the functionalities that they are meant to do.

| **Scenario** | **Testing** | **Actual Result** | **Pass/ Fail** |
| --- | --- | --- | --- |

| | | | |
|---|---|---|---|
| Viewing/Interacting with rooms in 3D. | Loading 3d objects unto webpage. | 3d models display when room is clicked | Pass |
| | A response after clicking on 3d models | A form is opened when a 3d object is clicked on | Pass |
| | Ability to rotate 3d models | Allows movement using mouse | Pass |
| | Viewing all available rooms | Room links display the appropriate room. | Pass |
| Feeding in data | Textboxes can take input from users. | Users can type in data | Pass |
| | Check boxes can take user input | Users can select option from check boxes. | Pass |

| | | | |
|---|---|---|---|
| | Appropriate system generated data. | Data is generated for | Pass |
| | | respective 3d objects | |
| Retrieving data | Database successfully links with webpage | Yes | Pass |
| | Upon form submission, data is reflected in the database. | When the submit button is clicked, the data is seen in the database. | Pass |
| | Data is stored in appropriate tables in database | No data is mixed up. | Pass |
| Administrator functionalities | View issue log | Successfully displays at the click of a button. | Pass |

| | | | |
|---|---|---|---|
| | Login | Successful login upon entering credentials. | Pass |
| | Assign a job. | Positive | Pass |
| | Check job status. | Positive | Pass |

5.23 System Testing

The last phase of development testing was system testing. This involved the testing of the full system as it stood at the time of testing. This required that that the various parts of the system were integrated and tested with real inputs to validate the performance of the system.

| **Functional Requirement** | **Meets Expectation** | **Comments** |
|---|---|---|
| Report issues using the system. | Yes | Works perfectly. |
| Administrator log in and log of account | Yes | Works. |

| | | |
|---|---|---|
| Make a record of reported issues available. | Yes | System generates records automatically |
| Administrator access to issue logs | Yes | Has permission to view issues reported. |

5.3 Release Testing

Release testing is conducted to measure the systems performance against required specifications. The release test phase is expected to be tested in real environments as that is what the system will be operating in. However, due to some major occurrences, this phase could not be conducted to its fullest potential. However due to social distancing measures, testing was conducted with available individuals who could be potential users. Feedback received from this testing phase revealed key factors to note. These factors included to need to make to make the system more user friendly by providing measures to direct first time users and the need to increase resemblance to actual objects such as the A/Cs.

# Chapter 6: Conclusion and Recommendation

## 6.1 Overview

As the final chapter of this report, this section seeks to point out some challenges encountered and future works and recommendations.

## 6.2 Challenges

1. Time constraints: Time constraints prevented the development of more three-dimensional rooms/halls for the facilities management system. Working with the time available, it only permitted the development of a few lecture halls amongst the vast number of halls available on campus.
2. Code Implementations: This challenge proved to be very daring as it required very close attention and focus to overcome. It ranged to the omission of keywords to constructing ways of implementing certain features that would otherwise be implemented in a rather known format.

## 6.3 Future work

1. More functionalities: There can be improvements made to the system in terms of the introduction of new functionalities that will be of benefit to its users. For instance, further

works can include booking functionalities to cater for rooms/halls on campus that require booking before they can be accessed.

2. Updated Locations: The 3d component of the system includes a limited number of locations. As such, developments can be geared towards updating the available rooms to grant users a wider variety of choices. This will improve the accuracy and consistency that comes with accounting for faulty equipment on campus.  For instance, there are offices on campus staff and faculty, but the current system does not include 3d model of any office. Updating venues by including offices of staff and faculty will increase the number of users which will in turn make stakeholders constantly updated on the conditions of the school environment.
3. Mobile Application: The system is currently limited to a web application. Considering the diverse means by which various people use technological devices, it would be in the developer's best interest to make provision for users with all various types of technology. The system in its current state does not have a mobile implementation and as such it will be a great contribution to make. Also, it will be more convenient to some users who would rather prefer to interact with a mobile application rather than a web application because of the frequency at which they engage with their mobile devices. It will also cater for users who lack devices such as a laptop.
4. Addition of more objects: As this project came with the aim of modelling select halls of Ashesi University to match as close as possible. This could not be possible due to time constraints, as mentioned earlier and other factors. Hence there is to need to keep on building upon what is already done in the form of new three-dimensional models to represent equipment and devices which have not been added to the system. Some examples

include sockets, lights, ac controllers, computers in labs and library, smoke detectors, and more.

## 6.4 Conclusion

For this project, a web-application for a facility management system was built for Ashesi University. This was to help stakeholders of the university track the condition of the equipment and facilities on campus by allowing users such as students, staff, and faculty to have a platform to report issues on damaged equipment. It incorporated the use of three-dimensional objects to create a virtual environment for its users.

## References

[1] Lavy, S. (2008), "Facility management practices in higher education buildings: A case study", *Journal of Facilities Management*, Vol. 6 No. 4, pp. 303-315. https://doi.org/10.1108/14725960810908163

[2] Ibm.com. 2018. *What Is Facilities Management?*. Retrieved February 6, 2020 from https://www.ibm.com/topics/facilities-management.

[3] Grace Lorraine D. Intal, Rex Aurelius C. Robielos, and Alysia Georgia B. Ortega. 2018. Design of user driven facility management system for universities. In Proceedings of the 6th International Conference on Information and Education Technology (ICIET ’18). *Association for Computing Machinery, New York, NY, USA*, 170–176. DOI:https://doi.org/10.1145/3178158.3178211

[4] Jensen, Per & Sarasoja, Anna-Liisa & Van der Voordt, Theo & Coenen, Christian. (2013). How Can Facilities Management Add Value To Organizations As Well As To Society?

[5] Salonen, A. and Deleryd, M. (2011), "Cost of poor maintenance: A concept for maintenance performance improvement", *Journal of Quality in Maintenance Engineering*, Vol. 17 No. 1, pp. 63-73. https://doi.org/10.1108/13552511111116259

[6]*IKEA International sales*. (n.d.). Retrieved April 20, 2020, from

https://about.ikea.com/en/contact/international-sales

[7] Stuart Hansen and Timothy V. Fossum. 2005. Refactoring model-view-controller. J. *Comput. Sci*. Coll. 21, 1 (October 2005), 120–129.

[8] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal.Pattern-Oriented Software Architecture, *A System of Patterns*, Vol 1. Wiley, Chichester, UK, 1996.

[9] E. Albassam and H. Gomaa. Applying software product lines tomultiplatform video games. In Proceedings of the 3rd International Workshop on Games and Software Engineering: *Engineering Computer Games to Enable Positive, Progressive Change, GAS* ’13, pages 1–7, Piscataway, NJ, USA, 2013. IEEE Press.

[10] Matthias Veit and Stephan Herrmann. 2003. Model-view-controller and object teams: a perfect match of paradigms. In Proceedings of the 2nd international conference on Aspectoriented software development (AOSD ’03). *Association for Computing Machinery, New York, NY, USA*, 140–149. DOI:https://doi.org/10.1145/643603.643618

[11] ED Angel, Eric Haines, and Dave Shreiner. 2018. Getting started with webGL and three.js. In ACM SIGGRAPH 2018 Courses (SIGGRAPH ’18*). Association for Computing Machinery, New York, NY, USA*, Article 2, 1–82. DOI:https://doi.org/10.1145/3214834.3214861

[12] Pankaj Mudholkar, Megha Mudholkar, and Snehal Kulkarni. 2010. Software testing. In Proceedings of the International Conference and Workshop on Emerging Trends in

Technology (ICWET ’10). *Association for Computing Machinery, New York, NY, USA*, 1024. DOI:https://doi.org/10.1145/1741906.1742242

# Appendix

Appendix A: Interview Guide

1. How does the school track its equipment such as furniture and electronic appliances such as projectors and computers?
2. Does the department receive reports when this equipment develops faults?
3. How often do these reports come in?
4. By whom is the report made?
5. Is there a system available for tracking these issues?
6. How does the system operate?

Appendix B: Issue Log Table from Database

Table structure | Relation view

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | issue_id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | issue_name | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 3 | issue_email | varchar(100) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | issue_location | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 5 | issue_equip | varchar(200) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 6 | issue_comment | varchar(250) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 7 | issue_assigned | varchar(100) | latin1_swedish_ci | | No | None | | | Change Drop More |
| 8 | issue_status | varchar(20) | latin5_turkish_ci | | No | None | | | Change Drop More |

Check all With selected: Browse Change Drop Primary Unique Index Fulltext Add to c

Appendix C: Job Assign Table from database

Table structure | Relation view

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | issue_name | int(11) | | | No | *None* | | | Change Drop More |
| 2 | stakeholder | varchar(100) | latin1_swedish_ci | | No | *None* | | | Change Drop More |
| 3 | date assigned | varchar(50) | latin1_swedish_ci | | No | *None* | | | Change Drop More |
| 4 | date completed | varchar(50) | latin1_swedish_ci | | No | *None* | | | Change Drop More |
| 5 | progress | varchar(50) | latin1_swedish_ci | | No | *None* | | | Change Drop More |

Appendix D: Employee table from database

Table structure | Relation view

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|
| 1 | emp_id | int(11) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| 2 | cname | varchar(100) | latin1_swedish_ci | | No | *None* | | | Change Drop More |
| 3 | email | varchar(100) | latin1_swedish_ci | | No | *None* | | | Change Drop More |
| 4 | company | varchar(200) | latin1_swedish_ci | | No | *None* | | | Change Drop More |

Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext