

## Research Article

**Development of a Web-Based Complaint Management Platform for a University Community****Victoria Oguntosi<sup>1,\*</sup>, Mejabi Oluwadurotimi<sup>1</sup>, Anthony Adoghe<sup>1</sup>, Ademola Abdulkareem<sup>1</sup> and Gideon Adeyemi<sup>2</sup>**<sup>1</sup>Department of Electrical & Information Engineering, Covenant University, Ota, Ogun State, Nigeria<sup>2</sup>Department of Civil Engineering, Covenant University, Ota, Ogun State, Nigeria

Received 10 December 2020; Accepted 18 February 2021

**Abstract**

Complaint management has served as a unique and efficient method of assessing student and staff satisfaction in any university community. It has helped foster improvement and change in the way operations are carried out and ensure that the students and staff are served to the best of the University's ability. Although there have been massive improvements using complaint management, there is still a lack of efficiency when it comes to manual complaints management. This work is aimed at developing a web-based complaint management system for a University community with the use of JavaScript as the programming language and MongoDB server as the database, in order to improve the way complaints are handled and analyzed in the University. It was discovered from this research that students were genuinely interested in an online platform for submitting complaints as it helped to simplify the process. The web-based complaint management system is, however, subject to various improvements and developments as technology advances.

*Keywords:* Complaint Management; JavaScript; MongoDB database; University Community; Web-based Platform.

**1. Introduction**

Complaint management is crucial and essential to any organization that intends to ensure maximum customer satisfaction. The methods of complaint management may be traditional (paper-based) or digital (electronic). The primary purpose of complaint management is to ensure that reported complaints in an organization are tracked efficiently and that there is a quick response to whatever problems need to be tackled. Complaints are complicated to track and monitor, and having to manage every complaint manually is going to be both inefficient and exhaustive. Many universities do not currently have an automated system for tracking and handling complaints. The conventional method of making complaints have proven to be quite stressful on the students as it requires multiple trips to the appropriate offices to ensure that complaints are going to be resolved.

Every university community must make complaint management systems one of the critical strategies needed for the essential running of the community [1]. The importance of a customer complaint management system in the operation of universities cannot be overemphasized. Student and staff complaints are inescapable in any university community, regardless of how reputable such an institution might be. Some complaints serve as a feedback mechanism for the university community. Members of such community might need to communicate their sentiments and impressions on issues of concerns. Complaints become helpful when such grumbings pull in a fast and compelling reaction from the university management. University management are

expected to hold their staff (teaching and non-teaching) and students in high esteem since they are the lifeblood of any university community [2].

In a world where speed and efficiency are continually being emphasized daily, having an online system that allows people to register any complaints quickly they might have and get a resolution is becoming more and more in demand. If complaints cannot be handled properly, effectively, and efficiently, organizations would fall behind their competition. Several complaint management systems have been developed based on different approaches for both private and public communities. Complaint management has to do with endeavors to encourage the voicing of issues through emails, toll-free numbers, or online surveys and then initiatives relating to quality improvement based on the voiced complaints. For instance, a reputable hotel lodging network utilizes dissatisfaction about individual outlets' quality control to improve its services [3]. The essential objective of complaint management is to enhance client care. From a broad perspective, complaint management is the procedure by which organizations handle clients' grievances. Complaints can enable a business to develop and progress in their objectives. It permits them to comprehend their customers better, gather information and input, increment client dependability, and reinforce their public image [3]. Previously, customers had to fill their complaints physically at a branch of the company. The problem with this method was that it was hard to organize all the protests, which meant that some complaints never get any attention. It was also hard to track complaints and avoid duplicate submissions. For this reason, the introduction of electronic channels became more favorable.

Complaint management systems are products that collect client input about dissatisfaction for a company. Successful

\*E-mail address: victoria.oguntosi@covenantuniversity.edu.ng  
ISSN: 1791-2377 © 2021 School of Science, IHU. All rights reserved.  
doi:10.25103/jestr.141.17

companies think about complaint issues as a way to open doors for development. They comprehend the connection between complaint resolution and client dedication and make a substantial effort to act promptly on items that can be immediately settled [4]. Nasr et al. [5] proposed a generic web-based complaint management system. The framework gave an electronic method of resolving people's issues in general by saving time and eliminating corruption. The system aimed to make matters simpler to gather, arrange, monitor, fix, and give the organization a helpful device to recognize and target problem territories, monitor issue handling execution, and make business upgrades. Abd El-Aziz et al [6] suggested a framework for a customer complaint management system using SOA. They set up the complaint management system using an online application that the residents would utilize to submit requests about their disappointment with the offered services. Their framework had the option to deal with issues by recording and providing input to each raised case. Rokhmawati et al [7] proposed an E-complaint system at GraPARI Telkomsel Malang, Indonesia. They were interested in making the e-complaint process to be more efficient in handling customer complaints. Acknowledgment of client dissatisfaction through the My GraPARI machine was considered not to address the clients' issues, given the set number of My GraPARI gadgets accessible. Concerning the issues that required further attention, the call center was considered unable to deal with all the problems maximally because each process of complaint handling must be communicated directly to the client. They gave a solution by suggesting that an information system be developed as one form of an application to aid Customer Relationship Management (CRM) operations, and it was planned to get complaints from clients. The strategy they utilized for framework advancement was the waterfall model, which incorporated a few phases of development and implementation, including requirements investigation, structure, execution, and testing. Meik, et al [8] point out that customer complaints analysis is an under-explored field of client incorporation into administration advancement processes.

This work involves the specific design and implementation of a web-based online platform that would allow members of a university community to lodge whatever complaints they might have regarding matters that need appropriate attention. This work is aimed at developing a web-based complaint management system for a University community with the use of JavaScript as the programming language and MongoDB server as the database, in order to improve the way complaints are handled and analyzed in the University. The basic structure of this paper is as follows: Section one introduces the work. Section two contains a literature review concerning past works that have been undertaken in this research area. Section three involves the system design of this project. It describes how the project was designed with detailed descriptions of the methodologies involved and steps taken. In Section Four, the results are presented. Section five concludes this paper with possible recommendations for enhancing the system.

## **2. Literature Review**

Razali et al. performed a recent examination concerning client complaints [9]. They developed a complaint management system from scratch called (e-Aduan) to allow UiTM Pahang's clients to gripe and remark about the

services and facilities provided by the College. In any case, concerning the setting of this task, the scientist discovered that the most suitable to the project theme of taking care of client complaints utilizing Service Oriented Architecture (SOA) was the work done by Najar et al. [10]. In their work, they attempted to improve the connection between the residents and the administration by bringing in another framework dependent on Service Oriented Architecture (SOA). Using the given framework in Government associations, from a perspective, governments would be able to limit residents' disappointment. It could then urge residents to take an interest in controlling Government bodies, for example, the administrations' staff and associations.

Cho et al [11] pointed out that online complaint management included dealing with client disappointment, accompanying web customer complaint management, and e-CRM. Nasr et al [7] indicated that a complaint management framework should enable issues to be simpler to arrange, track, and resolve. It should give an organization a valuable tool to distinguish target issue regions and monitor objections. Liu et al. [12] suggested that based on the connections between customer loyalty, repeat purchases, client dependability, and complaint voicing, companies would do well to ensure that a sufficient system is put in place to allow their customers voice their complaints and have those complaints attended to on time.

Manuhutu et al. [13] proposed a prototype model to design online students' complaints management using three steps: listening to customer needs, building/revising mock-up, and performing customer test drives. Their model made a connection between the framework designer and the framework client. They did this to avoid a clash, and it was majorly a trial-and-error process that worked iteratively. The object of this exercise was a data arrangement of students' complaints at Victory University, Sorong. Regularly, the staff take all the cards and evaluate them together with all stakeholders. By this manual system, the organization spent a lot of its budget on buying paper and wasted a lot of time considering the complaints even when the staff had other main jobs. In their system, Manuhutu et al. [13] used the Unified Modeling Language (UML) as the work system method. Lee [14] expressed that UML is perceived as a standard demonstrating language in the industry with extensive graphical documentation and complete arrangement of outlines. It is used to indicate, envision, alter, develop, and report the antiquities of an object-oriented situated programming framework being worked on. In describing the interaction or scenario of actor and use case, they used a use-case diagram that involved the student and the administrator. These two actors were required to login before moving on to the next process [13]. This research has limitations in data security, so for further enhancement, a security system needs to be developed and implemented into the design.

El-Sayed et al. [15] attempted to establish a model for electronic complaints, which could address dissatisfaction from inadequate assistance quality. The bit of leeway for this model was the primary technique for taking care of the resident's cases about what disappoints them. El-Sayed et al. disagreed with Najar et al. [16] whose proposed framework was intended to expel the focal complaint handling department. The outcome built small complaint handling segments in every division whose purpose was to give complaint services to dissatisfied residents. Because every section has its complaint handling procedure, every

department gave explicit complaint handling strategies as per its area of expertise. The specialist couldn't help contradicting the past model worried about expelling the centralization of complaint handling. This was against the focus of driving e-Complaints, which is the coordination and shared perspective on the clients' procedures. They proposed the underneath structure for the e-Complaint framework dependent on SOA. They upgraded the past structure by updating and unifying the complaint management framework. They separated the proposed structure into three levels that comprised of: A database level, which would contain information about system clients and their profiles, resident's data, accessible data, accessible assets, and social affiliation profiles; A business-level which comprised of the center of the framework—for example, complaint handling and criticism segments; A presentation level consisting of a web-based user interface. The proposed framework of El-Sayed et al. could, by and large, be applied to any electronic e-complaint system.

Fornell et al. [17] researched the relationship between customer complaints and customer satisfaction. They prompted firms to urge disappointed clients to complain and that the organizations ought to put resources into encouraging grievances, and that they ought to reward liberally. They proposed that this would lead to higher customer retention, and it would help the firms rank higher than the competition. Complaint management is a form of defensive marketing instead of offensive marketing strategies such as product quality, advertising, or pricing.

Shahin [18] considered customer complaints management (CCM) just as the related primary difficulties as fundamentals for accomplishing client maintenance and steadfastness. He discovered that managers manage various issues, for example, differing social problems in overseeing CCM frameworks and utilizing quality administration approaches in such frameworks. Culture is a significant issue that could influence consumer loyalty, complaint procedures, and worker mentalities. Altogether, it brings about company profitability. In light of the reviewed models, it was also underlined that a CCM framework might have various internal and outer levels, in which different processes are examined for organizational strategies. A comprehensive system of CCM was additionally introduced, which included multiple advances. Besides, a group working together was tended to as an essential achievement factor in overseeing client complaints. Other than the upsides of a group working together, it appears to be essential to give training to service staff to deal emphatically and compassionately with upsetting situations. It is viewed as a fundamental advantage in urging trust in staff to be enabled. A very much prepared and enabled staff member can help transform a disappointed client into an advocate of the company.

Cetin [19] expressed that it is necessary to increase customers' loyalty who buy goods or services from them and ensure that they operate as voluntary salespeople of the company. However, businesses do not always make their customers smile. They might cause customers to be dissatisfied with their products, services, or employees. If the company does not take any measures about the mentioned negativities, it will be exposed to massive customer migration to competitors. As a result of this, the company can considerably reduce customer losses by taking into account current and new customers' complaints and ensuring that the problem is solved. Listening carefully to complaints, recording them, and preventing the occurrence

of the same negative situation can provide a significant increase in the business's profitability. It would also strengthen its image because the trust it gives to its customers will also increase. A company that can conduct proper complaint management is always in contact with the customers. Because customers are accessible to the business, customers do not waste time reporting positive or negative opinions. If issues are resolved quickly, then the customer would be satisfied and be more loyal to the business. Today, in which the competition is very intense, companies can use customer complaints as a great advantage and create a strong bond with the customer [19]. Najjar et al. [16] tried to find a way to improve the already built-up complaint management system in which the method of complaining was that the clients had to go to the office building for any sort of help. They said that the current framework had a lot of notoriety and was not easy to use. They designed a new system that would allow complaint submission to be simpler to organize, monitor, resolve and provide the company with a useful means to decipher and acknowledge issue areas and control duplication of complaints. The system designed would allow the user to post complaints and will enable the employee to reply to the customer based on that complaint.

Alve et al. [20] concentrated on the most proficient method to track issues. In their proposed framework, they suggested a web application for registering complaints brought up in a company. Concerning the nature of the complaint, the related authority could make a move and attempt to settle an issue ahead of schedule, as could be expected under the circumstances. Due to such administrations, the time factor and desk work engaged with putting down the issues and maintaining records in a complaints book is eliminated. The physical labour required for issue tracking and resolution is minimized. They stated that in the previous framework, issues were written in a book, and it was consistently a tedious procedure. They said that sometimes there were problems that did not get resolved because the people responsible did not attend to the issue, and the higher authorities had no thought regarding it. Their proposed framework beat the drawbacks of the previous structure by giving a simple method to register, track, and monitor issues. It likewise lessened the resolution time, improved client administration, and developed the company's standard. The proposed system was more user-friendly, allowed complaint tracking functionality, and allowed users to monitor the system's condition. The administrator could rapidly recognize the problematic territories in the company. It was also easier to maintain. The project design of [20] contained two modules, one for the user and one for the admin. It had a client-server relationship. The users could post their objections, and the moderator would assess that issue and take the essential steps necessary for resolution. The page administrator can see the number of clients, specialists, pending and complete cases in the admin dashboard. On the full complaint page, the user can see which complaints have been resolved successfully.

In this section, the various complaint management systems have been explained with articles justifying each of their importance and success. This review also shows that web-based complaint management systems have been adopted in order to bypass the traditional methods of complaint management. However, some web-based frameworks need to be improved upon in areas such as security and centralization. This work aims to design and develop an online application platform where the students

and staff of any University can register their complaints about various problems and issues in the University environment.

### 3. Material and method

This section gives a detailed description of the methodology involved in the design of the complaint management system. The system was divided into several modules to allow for easy reusability, which was the basis for the design. The platform's system requirements, including the users, the user interface, the database models, and the structure, are explained. This system is targeted at a university setting but can also be used in other organizations that want to have a stable platform to handle complaints. The method developed in this research allows its users to submit whatever complaint they might have regarding the university community on the developed complaint management platform. The considered users are the students of the university, the staff of the university and the system administrators.

The primary functions of this system are data entry, data retrieval, and data management. The system has an authentication feature that will enable users to sign in to perform particular actions. The system allows authorized users to submit complaints, delete complaints, and take action on the filed complaints. The system also has an authentication feature that will enable users to log in to perform particular actions.

The system design is fully described in this section, including the system architecture, system modeling, and database designs which are explained in detail. The interaction of various components to achieve the purpose of the system is discussed extensively, including block diagrams, process flow diagrams, and UML diagrams.

#### 3.1 System Analysis

This system is to be used by the students and staff within a university community. The system will allow students and staff to register complaints and dissatisfactions about the university community. The system's database allows for quick entry and retrieval of data. This system is to be facilitated on the internet, thereby allowing all community members and authorized users to operate the system via the internet. During the process of developing the software, requirements gathering was carried out. The system needed a foundation upon which to base its database design and flow of operations. The requirements collection was done through the observation method. The usual process of submitting complaints in the university involves either writing a mail to the Dean of Student's Affairs explaining the issues or going physically to the office of the Dean to make the complaint. With this manual method, it is hard to track the problems. Most complaints never got attended to because of poor information management. This flow of operations is the traditional way of submitting complaints, and the method of resolving the complaints was the basis for the design of this system. After analysis of the data obtained during requirements gathering, there were several system requirements identified. They belong to two groups, namely; functional and non-functional requirements. The system requirements for the platform include:

- (1) The system should allow users to submit complaints easily.

- (2) The system should store user records and complaints.
- (3) The system should manage user data, which involves read and write operations like creating, reading, updating, deleting.

#### 3.1.1 Functional Requirements

The system developer implements the functional requirements to ensure the users can carry out the scheduled tasks [21]. At the end of the requirements gathering, the following functional requirements were identified:

- (1) The system should allow users to log in to access the functionality of the system permitted to them.
- (2) The system should ensure that ONLY authorized users manipulate data in the database.
- (3) The system should allow for users of different levels which are the administrators and regular users (Students and staff)
- (4) The system should allow anyone, whether authenticated or not, to see the registered complaints.
- (5) The system should enable the administrators to delete a complaint if they deem it unfit for registration.
- (6) The system should allow users to vote up a complaint if they consider the claim relevant to them.
- (7) The system should keep a user logged in for one hour for security reasons.
- (8) The system should store all the submitted data in a database for easy access and historical reference.

#### 3.1.2 Non-Functional Requirements

The non-functional requirements highlight the systems' quality and performance attributes [22]. The developed system must behave in some specific ways. It must exhibit:

- (1) Usability: Usability has to do with the ease of use of the system to carry out tasks. The decision was made to ensure that this requirement was met to employ the "user-centered design" concept. This concept involves ensuring that the system is used as naturally and intuitively as possible by the user.
- (2) Scalability: The system needs to expand data as more users inevitably begin to use the system.
- (3) Security: Security was of utmost importance because anyone with access to the internet would view the database's information. Only authorized users should be allowed access to specific data. Because of this, some design decisions were made to ensure that this requirement was met. These include: Authentication of users which ensures that the user is who they say they are through the required use of passwords; Authorization is given to the administrators to enable them to have total control over the data that is being passed into the system; Validation is carried out on input to ensure that all malicious attempts at database intrusion or injection do not succeed.
- (4) Reliability/Availability: The system should be readily available to the user anytime they need to access it.
- (5) Modularity: This requirement ensured that the content of the system could be used in another application. What this ensures is that the system also has more tendency to be reusable. Some design decisions towards this requirement are the use of functions for

processes that are used repeatedly and separation of the user interface, backend server, and database

- (6) Data Integrity: The data stored in the system should be timely and valid

### 3.1.3 Minimum Hardware Requirements

These requirements are the physical resources needed for the system to function. They include an Intel Pentium processor, 32GB hard disk space, and 256MB RAM.

### 3.1.4 Software Requirements

Software is not a small aspect of today's systems. It allows systems to be fully functional and capable of handling tasks while giving them flexibility and control [23]. The specified software requirements are specified as: operating system (Linux, Mac OS, or Windows OS); JavaScript programming language; Development tools (VS Code, Git, and Postman); and a MongoDB Database Management System (DBMS).

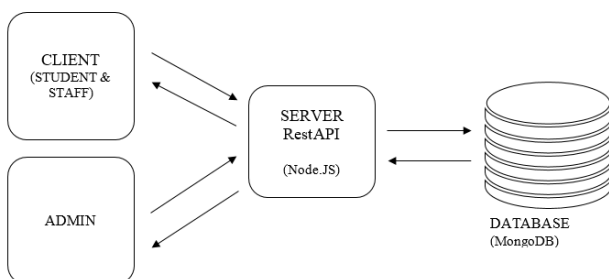


Fig. 1. Base Architecture of the Complaint Management System.

## 3.2 System Architecture

Figure 1 shows the system design of the complaint management system. The system architecture involves defining how software components would be combined to achieve the proposed output [24]. The architectural design style used are the User-Client and Client-Server Architectural Style.

### 3.2.1 Client-Server Architecture

This type of architecture involves sharing loads of client-server. It includes a central system hosting the server to be consumed by the client. It is a service-oriented architecture that allows for continual, uninterrupted client service. It responds to clients' requests by subduing network traffic rather than sending a full file [25]. There are separate client and server systems, both connected by a network. The backend server used is the Node.JS HTTP server. The database management system is MongoDB. The clients are the various users of the system. The network that connects the clients with the DBMS is the internet.

### 3.2.2 User-Client Architecture

The client initiate requests via the Graphical User Interface (GUI), waits for the reply and processes this reply. The server authorizes users and then generates the result from processes carried out. The complaint management system comprises of essential modules, namely:

- a) The login/logout management module handles the data validation involved in the login process and authentication necessary to access the system. It also handles updating the audit trail for security measures.
- b) Complaint submission module: This module would handle a complaint's entry into the system by an

authenticated user. Random users would not be allowed to submit a complaint.

- c) Complaint management module: This module would give the administrator access to manage complaints, such as moving the complaint from pending to complete. It also includes deleting submissions that are inappropriate such as the use of foul language or if the complaint has already been resolved.
- d) Complaint viewing module: This module would allow anybody who gains access to the website to view all the registered complaints in the system without the need to log in.

These modules are accessed using different interfaces to interact with users allowing them viewing and editing privileges where necessary. A regular user can access the login/logout management module, complaint submission module, and complaint retrieval module. At the same time, an administrator has access to all the modules and the complaint management module.

## 3.3 Dynamic System Modelling

A formal model of the Complaint Management System is developed using Unified Modeling Language (UML). The UML is the standard language for specifying, visualizing, constructing, and documenting all the software systems' artifacts. It allows the designer to use diagrams to provide different perspective views of the system parts. The system's behaviors are modeled using the dynamic UML diagrams. The diagrams used are; Activity diagram and use-case diagram. The interactions are modeled separately using the appropriate diagrams and grouped under Interaction UML diagrams. Significantly, use cases are indicated at various abstraction degrees due to the different partners in a project. This helps advance better incorporation, communication, and understanding amongst the involved stakeholders [26]. A use-case diagram is used to capture the system functionality as seen by its users. In other words, it is used for the visualization of the system's interaction with the outside world. The systems functionality has been grouped into four modules which are the Login/Logout Management Module; the Complaint Submission Module; the Complaint Retrieval Module; and Complaint Update Module. Their use-case narratives are represented in the following subsections.

### 3.3.1 Dynamic modeling of the Login/Logout management module

The Login/logout management module's dynamic modeling involves the use-case narratives for login (Table 1) and logout (Table 2) as indicated in Figure 2.

### 3.3.2 Dynamic modeling of the complaint update module

The update management module's dynamic modeling involves the use-case narratives for update (Table 3) and delete (Table 4).

## 3.4 Interface design

There are several elements of the systems interface design. The Menus are used to allow for smooth navigation of the system by the user. There is the permanent "top menu," which allows for navigation at the web application's global level. This means that all users have access to all the web pages to which this menu provides links to. The Reports are the means via which results are displayed. The Dialogue or screen design are what the user interacts with. The user can

edit information (majorly via forms) and view the system's output via the screen design.

**Table 1.** Use-Case narratives for login management

<b>USE CASE 1</b>		<b>LOG-IN</b>	
Goal in content	Allow users access into the system and to save login session data.		
Level	This is a primary Login use case.		
Parameters	In: User's username and password. Out: User's Home Page GUI.		
Preconditions	The user is a genuine legitimate operator of the system.		
Post-Conditions (success end)	Access is granted into the system.		
Post-conditions (failed end)	Access is denied into the system.		
Actor	Student, Staff, Administrator		
Trigger	The user requests for access to submit a new complaint into the system, and permission to carry out operations		
Description (event flow)	Actor action	System respond	Affected data object with the operation
	1. Requests for access into the system by inputting a valid username and password.	2. Validates input, authenticates user, saves login details, and displays home page via rich Interface for the logged-in user.	Searches database to check for user and validity of login credentials. Updates login audit log with username and time of login.

**Table 2.** Use-Case narratives for logout management

<b>USE CASE 2</b>		<b>LOG-OUT</b>	
Goal in content	To destroy the user's session and update login audit log session data.		
Level	This is a primary logout use case		
Parameters	In: Logged in session details Out: Log-in session details		
Preconditions	The user is a genuine legitimate operator of the system and is logged in.		
Post-conditions (success end)	The login session ends successfully.		
Post-conditions (failed end)	The login session does not end successfully.		
Actor	User, Administrator		
Trigger	The user requests to log out of the system		
Description (event flow)	Actor action	System respond	Affected data object with the operation
	1. User clicks on the "Logout" link.	2. Logs out the user and redirects to the home page of the system.	The login audit log is updated with logout time and date, and all session variables are destroyed.

### 3.5 Database Design

A database is similar to a warehouse where data is stored. A database serves as the repository for the system's information. The design of the database was one of the primary design considerations of this system. Since the system's significant function is to store data in a data repository, how the tables relate is of utmost importance. The non-Relational Database Management System used was MongoDB. It allows for the management of the data involved in this system, and it is separate from the system's

development platform. It also provides tools that help in the design of the database, as well as its implementation.

In order to ensure cohesion throughout the system, data models were designed to guide how a user was created and how a complaint would be stored. The model consisted of various fields and properties that each user and complaint would have. The two main models designed for this system were the user and complaint models. The user model describes the properties that each user that creates an account should have. The table contains four main fields; user\_id, email, password, and admin as shown in Figure 3.

#### 4. Complaint Management System Implementation

This section provides insight into the implementation of the Complaints Management System. The required systems specifications necessary to support the developed system and the tools used to develop it are provided. This is followed by a presentation of the various modules that make up the system. The following sub-sections describe the complaint management system implementation of the Home, Login, Signup, Add Complaint, View Complaint, and Delete Pages.

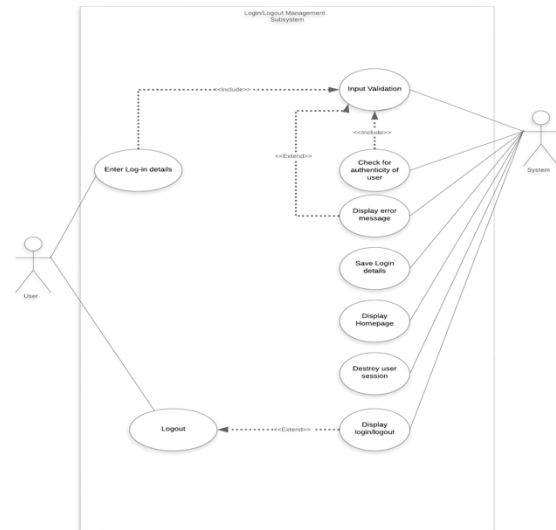


Fig. 2. Use-case diagram for login/logout management.

Table 3. Use case narratives for complaint updates

USE CASE 3	REGISTER A NEW COMPLAINT		
Goal in content	To add a new complaint into the database		
Level	This is a primary Register new complaint use case		
Parameters	In: Complaint title and description Out: Unique complaint identification number		
Preconditions	The user must be an authenticated user of the system; the user must be logged in.		
Post-conditions (success end)	The complaint is successfully registered.		
Post-conditions (failed end)	The complaint is not successfully registered.		
Actor	User		
Trigger	The user requests to add a new complaint to the database.		
Description (event flow)	Actor action	System respond	Affected data object with the operation
	1. Requests to add a new complaint. 3. Fills the complaint form with complaint details	2. Confirms that a logged-in user makes the request. 6. Validates input.	Reads session data displays add complaint page. Updates the database with the complaint details, displays new complaints on the home page.

Table 4. Use-Case narratives to delete data

USE CASE 4	DELETE DATA		
Goal in content	To delete data from the database.		
Level	This is a basic delete data use case		
Parameters	In: The complaint to be deleted Out: Confirmation message and success message		
Preconditions	The user must be an authorized administrator of the system, and the admin must be logged in.		
Post-conditions (success end)	The delete operation is successful		
Post-conditions (failed end)	The delete operation is not successful, or the administrator clicks the "cancel" button in the confirmation message box to discontinue the transaction.		
Actor	Administrator		
Trigger	The administrator requests to delete a complaint.		
Description (event flow)	Actor action	System respond	Affected data object with the operation

	<ol style="list-style-type: none"> <li>1. Requests to view Delete Page.</li> <li>3. Clicks the "Delete" button.</li> <li>5. Clicks "OK" to continue.</li> </ol>	<ol style="list-style-type: none"> <li>2. Confirms that an authorized administrator makes the request.</li> <li>4. Pops up a message box asking the admin to confirm the delete action.</li> <li>6. Deletes the complaint from the database and displays a success message.</li> </ol>	<p>Reads session data and displays "Delete" page.</p> <p>On clicking "Delete," the confirmation message is triggered.</p> <p>Deletes the complaint from the database.</p>
--	---	--	---

```

4 const usersSchema = mongoose.Schema({
5   _id: mongoose.Types.ObjectId,
6   email: { type: String, required: true, unique: true },
7   password: { type: String, required: true },
8   admin: { type: Boolean, default: false },
9 });

```

Fig. 3. User Model.

```

4 const complaintsSchema = mongoose.Schema({
5   _id: mongoose.Types.ObjectId,
6   date: { type: Date, default: Date.now() },
7   title: { type: String, required: true },
8   description: { type: String, required: true },
9   likes: { type: Array },
10  numberOfLikes: { type: Number, default: 1 },
11 });

```

Fig. 4. Complaint Model.

#### 4.1 The Home Page

The home page shows all the registered complaints in the system. It also includes a navigation bar that allows the user to access different sites like logging in and out. The next section of the home page is the complaints. Here, the user can view all the complaints, and based on whether there are complaints registered already in the database, the user would be notified. While the data is being fetched from the database, a loading spinner would be shown to signify progress. The data is fetched from the database through the backend code. The backend code acts as an intermediary between the front-end user interface and the database. After the page has finished loading, the resulting page is shown in Figure 5. The Home Page shows different complaints posted by members of the university community which are not yet resolved.

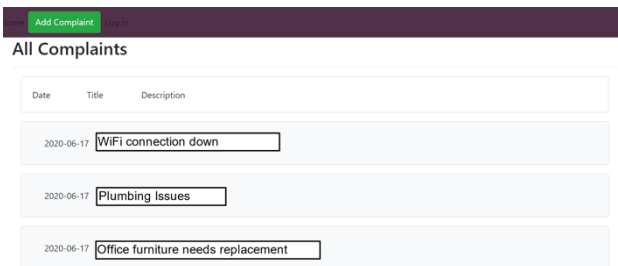


Fig. 5. The Home Page.

#### 4.2 The Login Page

The login page (Figure 6) is available to anyone who has access to the system. It requires the entry of the correct email and password to get authentication. If a user has not been successfully logged in, there is a link to login on to the navigation bar. If a user also tries to add a complaint without

being logged in, they are also redirected to the login page for the necessary authentication. To prevent a user from accessing the add a new complaint page, a check is done to ensure that the user is logged in. If the test returns false, the user is automatically redirected to the login page. On the other hand, if a user is already logged in, they are allowed access, and if they try to access the login page, it redirects them to the home page. When the user navigates to the login page, they are met with the login interface. Anytime the user inputs their details into the input boxes, a validation check is done to ensure that neither the email nor password input boxes are empty and that the password is at least six characters long. If the validation check is successful, the system will send the data to the backend to verify that the user is registered in the database. At the backend, the system checks with the database to confirm that the user is present.

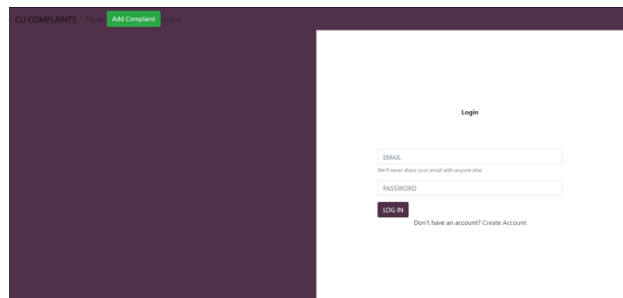


Fig. 6. The Login Page.

#### 4.3 The Signup Page

If a user cannot log in because they do not have an account, they can create a new account in the system. When the user inputs their details, the system checks to ensure that none of the input fields is empty and that the passwords both matches more than six characters. If the validation checks are successful, then the details are sent to the backend. A check is made to ensure that the user does not already have an account at the backend as shown in Figure 7.

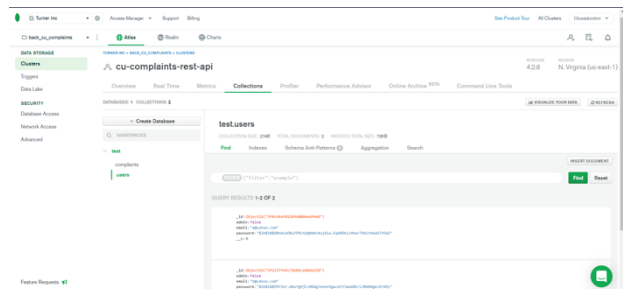


Fig. 7. Registered users in the MongoDB database.

If the account is not found, then a new user is created (Figure 8). A user model is used to specify the



characteristics each user should have (Figure 3). The model is then used to map a new user in the database.

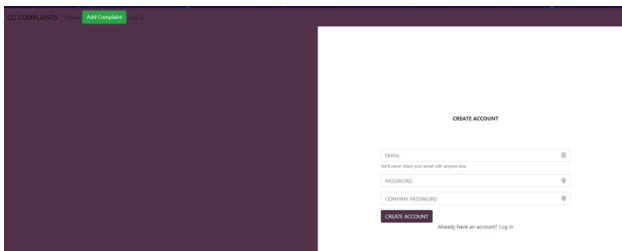


Fig. 8. Create new account page.

#### 4.4 The Add Complaint Page

The user must be authenticated to add a new complaint to the system. If they are, they can gain access to this page; if not, they would be redirected to the login page. To save a new complaint into the system, the front-end system communicates with the backend to save it into the database as shown in Figure 9.

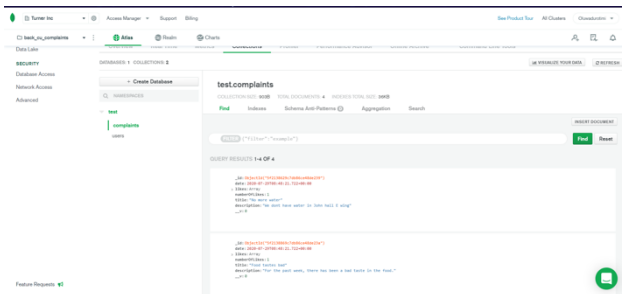


Fig. 9. Saved complaints in the database.

At the backend, a model is created for each complaint (Figure 4). After the model is created, the complaint is saved into the MongoDB cloud. The user interface is shown in Figure 10.

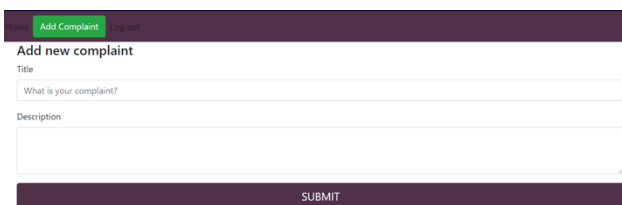


Fig. 10. Saved complaints in the database.

#### 4.5 The View Complaint Page

This page (Figure 11) allows anyone to view a complaint and its corresponding details. A user does not need to be logged in to see a complaint, but they need to be logged in to like or dislike a complaint. While the page is loading, the system checks to see if the user has liked a complaint previously or not. When a user decides to like or dislike a complaint, the original complaint data needs to be updated in the database.

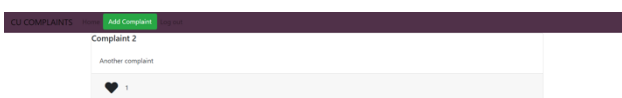


Fig. 11. View complaints page.

#### 4.6 The Delete Page

This page is only available to an authorized administrator. It allows the administrator to delete data from the database.

Regular users cannot see a delete button. There is a tag of admin in the database, which is given to every user when they sign up, and this is defaulted to false. To provide user admin privileges, the system owner needs to manually change the tag from false to true in the database (Figure 12 and 13).

The interaction between the system and the users has been explained and described by showing the various interfaces and how they all work together to form a complaint management system. As indicated through the different page design at both frontend and backend, members of the university community are able to lodge their complaints onto the web app. After the complaint has been resolved, it is removed from the forum and marked as resolved.

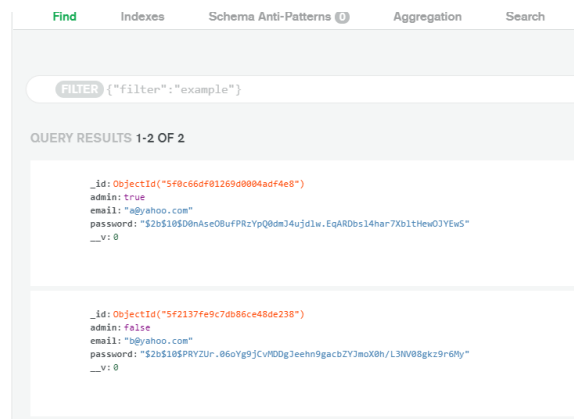


Fig. 12. Admin privileges.

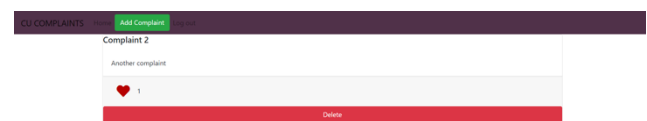


Fig. 13. View complaints page with a delete button.

## 5. Conclusions

This research has presented the analysis, design, implementation, and testing of the complaint management system. This designed system is expected to improve the complaint management and resolution in place of manual information management and provide a secure way of expressing dissatisfaction and urgent issues. The inclusion of this web-based complaints management system to any university community will aid in improving the school management's operations in terms of processing and attending to complaints. The system's development has been done using several concepts, tools, and programming languages for the backend and frontend logic, with the feature of a database storage service. Future research direction includes introducing extra features to allow greater ease of use, these features include complaint tracking and a private messaging module. The system can also be adapted for mobile operating systems and can be used on Android and iOS devices. This complaint management system can also be adapted for use in other organizations that depend on customer satisfaction to thrive.

## Acknowledgements

The authors acknowledge the financial support offered by Covenant University in the actualization of this research work for publication.



## References

1. F. Claes and W. Birger, "A model for customer complaint Management" *Marketing Science*. Vol. 7 No.3 pp. 287-298, (1998)
2. E.O. Elijah and U. Abdullahi, "Imperatives of Customer Relationship Management in Nigeria Banking Industry" Kuwait chapter of *Arabian Journal of Business and Management Review*. Vol. 2 No.1 pp 60.62. (2012)
3. B. Wernerfelt, C. Fornell, "A model for customer complaint management" (1988).
4. E.A. Afify, "A model for customer complaint management system using SOA" (2011).
5. Osman Nasr, Enayat Alkhider, "Online Complaint Management System", *International Journal of Innovative Science, Engineering & Technology*, Vol. 2, pp. 305-307 (2015).
6. Esraa Abd El-Aziz, Abd El-Fatah, Mona Ahmed Kadry. In the Proceedings of the 15th WSEAS international conference on computers, USA, pp. 291-296, (2011).
7. R. Rokhmawati, and F. Pradana, Pengembangan Sistem Informasi E-Complaint (Studi Kasus GraPARI Telkomsel Malang). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 1, No. 12, pp. 1589-1598, (2017).
8. Julia Meik, Markus Blut and Christian Brock, "Complaining Customers as Innovation Contributors", *SRII Global Conference* Pp.125132, (2014)
9. R. Razali, K. N. Abd Halim, and K. Jusoff, "Quality Improvement of Services in Unversiti Teknologi Mara Pahang from a Management Perspective." *Management Science & Engineering* Vol.5, No.1, pp. 71-80, (2011).
10. A. S. Najjar, H. A. Al-Sukhni, and N. Aghakhani, "The Application of Service-Oriented Architecture in E-complaint System", *International Conference on Communication Software and Networks*, (2010, 26-28 Feb. (2010).
11. Yooncheong Cho and Roxanne Hiltz Rutgers, "An Analysis of Online Customer Complaints: Implications for Web Complaint Management", *35th Hawaii International Conference on System Sciences*, (2002)
12. Jin-Lan Liu, Jian Kang, Yin Bai, Xin Zhang, "The Study of Customer Complaints Management Based On System Dynamics: Modeling And Simulation", *Proceedings of the Fifth International Conference on Machine Learning and Cybernetics*, pp. 2041-2046, pp.2040-2046, (2006)
13. M. Manuhutu and L. Uktoiseja, "Design and Implementation of Online Students' Complaint (Case Study of English Study Program at Victory University, Sorong), (2018).
14. Sunguk Lee, Unified Modeling Language (UML) for Database Systems and Computer Applications, *International Journal of Database Theory and Application*, Vol. 5, No. 1, pp. 157-164, (2012).
15. Mona Ahmed Kadry El-Sayed & Abd El-Fatah A. Hegazy, "A model for Customer Complaint Management System using SOA".
16. A. S. Najjar, H. A. Al-Sukhni, and N. Aghakhani, "The Application of Service-Oriented Architecture in E-complaint System." Paper presented at (ICCSN '10) the Second International Conference on Communication Software and Networks, (2010).
17. Claes Fornell and Birger Wernerfelt, "A model for Customer Complaint Management", (1988).
18. Arash Shahin, "Customer Complaints Management and Empowerment".
19. Filliz Asian Cetin, "Customer Complaint Management, A conceptual View", (2018).
20. Sneha Alve, Vishakha Babardesai, Sneha Bhosale, Siddhi Kapadi and Atul. B. Yadav, "Web Application for Complaint Tracking and Resolving", (2017).
21. AltexSoft, "Functional and Non-functional Requirements: Specification and Types." <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/> (Accessed Jul. 11, 2020).
22. Guru99, "What is Non-Functional Requirement? Types and Examples." <https://www.guru99.com/non-functional-requirement-type-example.html> (Accessed Jul. 20, 2020).
23. L. Laird, "Strengthening the 'engineering' in software engineering education: A software engineering bachelor of engineering program for the 21st century," *Proc. - 2016 IEEE 29th Conf. Softw. Eng. Educ. Training, CSEEandT*, pp. 128–131, 2016, (2016).
24. K. E. Harper and A. Dagnino, "Agile software architecture in advanced data analytics," *Proc. - Work. IEEE/IFIP Conf. Softw. Archit*, pp. 243–246, (2014).
25. Apachebooster, "What is client-server architecture and what are its types?" <https://apachebooster.com/blog/what-is-client-server-architecture-and-what-are-its-types/> (Accessed Jul. 11, 2020).
26. D. Savić et al., "Use case specification at different levels of abstraction," *Proc. - 2012 8th Int. Conf. Qual. Inf. Commun. Technol.* pp. 187–192, (2012).