**CHAPTER ONE**

**INTRODUCTION**

1. **INTRODUCTION**

Students are the main stakeholders in educational institutions, and the success of the academic environment as a whole greatly depends on how satisfied they are. However, students frequently encounter a variety of difficulties that can impede their learning process, from academic and administrative worries to facility-related obstacles. Educational institutions must handle student complaints in a methodical and open manner in order to properly address these issues. To make sure that students feel heard and supported, a complaint management system (CMS) created especially for them can be an effective tool for streamlining the reporting, monitoring, and addressing of complaints. Complaint Management system may be a platform independent application, so this web application are often accessed anywhere within the system. this is often also developed for reduces the communication cost between the staffs and to supply the efficient service to their staffs. The main objective of this Complaint Management system is to specialize in the problems associated with internal system.

**1.1 BACKGROUND OF THE SYSTEM**

In globally, A complaint is a disclosure of dissatisfaction by an individual or organization to a particular institution or unit related to the results of the provided products or services (Iso, n.d.) . The submitted complaint is expected to be responded to by the relevant institution or unit. Complaints can be a very important source of information to measure the performance of an institution, The complaint allows the institution to fix problems in its performance immediately. That way, the institution can improve the efficiency of its performance and service quality (Setiadi et al., 2022). Good service quality from an institution or unit can provide opportunities to turn dissatisfied customers into satisfied customers and can even become regular customers. Complaint management encompasses a systematic approach to handling, managing, responding to, and reporting customer grievances. It necessitates a well-structured framework and continuous monitoring to expedite resolutions promptly. CMS presents itself as a step-by-step methodology for receiving, recording, processing, responding to, and reporting on complaints (Singh Rana, n.d.).

In Africa, a complaint is a statement in which you express your dissatisfaction with a particular situation (Ifeanyi Alex, 2023) . Although a lot of systems that were manual in the past have gone digital, but the issue of complaint management still remains manual. Many students, especially fresher, have many issues, which they are dissatisfied with, but they keep these issues bottled down

Complaints are challenging to track and monitor, and manually managing each one would be inefficient and time-consuming.The current process of filing complaints has proven to be highly stressful for students and staff, especially in universities in developing nations since it necessitates repeated journeys to the proper departments to guarantee that problems are processed (Okokpujie et al., 2023) .

In Somalia, The effectiveness of student complaint management

helps ensure a comfortable atmosphere to the university users,

motivates learners, and upholds the reputation of the organization. Traditionally, universities have employed manual systems such as suggesting boxes and in person complaint submission to cater to student issues (Mohamed et al., 2025).

Today, complaint management systems leverage web-based platforms and mobile apps. These systems offer real-time tracking, automated responses, and predictive insights, making the process more efficient and user-friendly. The transition to web-based systems has been driven by several factors, including the growing demand for transparency, efficiency, and accessibility in handling student complaints.

The online development of web-based students complaint management system this system is Proposed to help the faculties at Salaam University automate the tasks associated with the management of the complaints project.

**1.2 PROBLEM STATEMENT**

The information about complaints and issues is currently recorded in spreadsheets. Due to several challenges, this data may be lost or mismanaged by the university. To address this problem, the faculties at Salaam University have identified the need for a computerized complaint management system to efficiently handle and track student and staff complaints. In this work, the complaint management process was analyzed in detail, and the following issues were identified: time-consuming manual tracking of complaints, lack of centralized storage and backup, and poor search and retrieval capabilities. A web-based complaint management system could help salaam university staff efficiently manage and resolve complaints. The system would ensure that complaints are properly logged, tracked, and resolved in a timely manner, avoiding duplication or loss of information. Additionally, it would provide a centralized platform for storing and retrieving complaint records, improving transparency and accountability. The system could also serve as a reference for future improvements, allowing the university to identify recurring issues and implement proactive measures to address them.

**1.2.1 PROVIDING SOLUTION FOR THE PROBLEM**

The problem statement that we came back to solve is what we did in this project to address the needs of the students.

**1.3 PURPOSE OF PROJECT**

Complaint Management System provides an online way of solving the problems faced by the salaam university students. the main purpose or the aim of this project is saving time and reducing corruption , And The ability of providing many of the reports on the system , and add to improve the process of submitting a complaint.

**1.4 PROJECT OBJECTIVES**

The objectives of this study are the goals we hope to achieve when the new system is up and working.

The objectives are:

1) To plan easy complaint management system for students to apply the complaint

2) To analyze the complaint made by students

2) To design, implement and test a complaint management system

**1.5 PROJECT SCOPE**

Complaint management in universities is increasingly adopting computerized systems. However, this research focuses exclusively on Salaam University, a private higher education institution in Mogadishu, Somalia. The university faces various issues raised by students, including academic concerns, administrative delays, and facility-related complaints.

The final product of this research will be a fully functional web‑based Complaint Management System designed to enhance efficiency, transparency, and student satisfaction. The system will be developed using PHP for the front‑end and MySQL for the back‑end.

**1.5.1 CONTENT SCOPE**

This project focuses on developing a complaint management system that streamlines the process of reporting, tracking, and resolving complaints. The system will include the following main features:

1. Student login – students can login
2. Complaint submission and tracking – Student can submit complaints and monitor their progress.
3. Complaint categorization – complaints can be classified by type
4. Admin dashboard – a management panel for staff to view, update, and resolve complaints efficiently.

**1.5.2 GEOGRAPHICAL SCOPE**

This research and system development are limited to Salaam University in Mogadishu, Somalia. The project aims to optimize the process of handling complaints for students and staff within this institution.

**1.5.3 TIME SCOPE**

The research and system development were carried out over a period of six months, starting from February and ending in July.

**1.6 SIGNIFICANCE OF THE OBJECTIVES**

The significance of this system is to serve better than the existing system which is highly manual and therefore difficult in terms of monitoring the complaint in the University, improve database and enhance effectiveness, efficiency and security of the system. The system is expected to be easy as user can browse their complaint anytime Generally this system provides the following significances to the system users

* To make the compliance process to be easy
* Any student can send his/her complaint from anywhere and anytime using his smart phone.
* Reduce the paper work
* To reduce the unwanted communication from customers
* Customers use time effectively.
* To students satisfaction.
* Faster decision making due to the report generation functionality.
* Creates Job satisfaction to the employees.

# 1.7 REPORT ORGANIZATION

In this research, chapters organized as follows:

**CHAPTER ONE:** is an introduction and background of web-based ance evaluacomplain management system. Furthermore, this chapter will cover the objectives, scope as well as significance of the study and how the project was organized.

**CHAPTER TWO:** This chapter also discusses the introduction of relevant literature of design and implementation of an complaint management system with listing concepts and opinions from experts or authors. This chapter deals with previous work or existing systems

**CHAPTER THREE:** Defines how the data collected, analyzed, how the interview has been done and the finding of the analyzed data and also the chapter will discuss how the system was analyzed using UML (use case diagram).

**CHAPTER FOUR:** Proposes the design and implementation of an complaint management system in details

**CHAPTER FIVE:** is the last chapter of the project and it describes the implementation of this project and testing it to produce a complete functioning system.

**CHAPTER SIX:** Finally, this chapter provides conclusion and recommendations by describing the strengths and achievements of the projects. This chapter also gives the weaknesses of the design and implementation of an complaint management system.

**CHAPTER TWO**

# LITERATURE REVIEW

## 2.1 INTRODUCTION

Universities and schools continue to make incremental steps into the digital era, For higher education institutions, it’s critical that they must be able to arrange their information storage for easier, more effective access.

A complaint management system (CMS) is a crucial component in any organization, particularly in academic institutions where students require an efficient platform to express their concerns. A web-based complaint management system (WCMS) facilitates real-time complaint submission, tracking, and resolution. This chapter reviews relevant literature on CMS, covering theoretical and conceptual frameworks, case studies, comparisons with existing systems, and a summary of key findings.

## 2.2 THEORETICAL AND CONCEPTUAL DEVELOPMENT

In traditional complaint management systems, the process of handling and resolving complaints is typically carried out through manual or paper-based methods. In the traditional approach, complainants typically submit their complaints through physical complaint forms, suggestion boxes, email, or direct communication with relevant personnel.

manual submission process may involve filling out paper forms or sending written correspondence (Devaraj, S., & Kohli, R. 2003). Upon receiving a complaint, the traditional approach involves documenting the details manually on paper. Complaints may be logged into physical registers or filed in paper folders. This method can be time-consuming and prone to errors in information management (Parasuraman, A., Zeithaml, V.A., & Berry, L. L. 1985). In a traditional system, complaints are manually routed to the relevant personnel or departments responsible for addressing the specific issue raised in the complaint. This process often relies on manual coordination, which can introduce delays and potential miscommunication (Bitner, M. J., Booms, B. H., & Tetreault, M. S. 1990). Complaints are manually tracked using spreadsheets, logbooks, or other physical records. This method requires diligent record-keeping and manual updates to track the status of complaints.

Follow-up actions and resolution progress are communicated through direct interaction or traditional communication channels (Johnston, R., & Clark, G. 2005).

Complaint management ensures that all reported complaints in an organization are adequately followed up and that any issues are immediately resolved. Complaints are challenging to track and monitor, and manually managing each one would be inefficient and time-consuming.The complaint management system recognizes the link between the goal of resolving complaints and customer loyalty, and it make a generous effort to act quickly on issues that can be fixed right away[9].

A complaint is defined herein as an issue arising when an incident occurs, and there is an absence of willingness or initiative to address the matter at hand. Failure to articulate concerns when something goes awry not only deprives students of a platform for expression but also hinders the institution's potential for improvement.

## 2.3 CASE STUDY OF THE RESEARCH

Salaam University (SU) is one of the most popular Private University which is non-profit academic institution in Somalia. At Salaam University, there is currently no dedicated digital platform to manage and resolve student complaints. Students often face academic, administrative, and facility-related issues but lack an efficient system to raise their concerns. Therefore, there is a critical need to design and implement a web-based complaint management system that streamlines the process, ensures transparency, and improves student satisfaction. The credit for the idea of establishing this university goes to Somali Scholars felt the need for a university under private initiative, and discussed the concept of establishing this university with a group of intellectuals, researchers and educationists, at home and abroad. By the Grace of Allah, Salaam University (SU) was founded in the year 2009.

**VISION**

SU aspires to be university of choice in nurturing innovation and talent in science, technology and development

**MISSION**

SU mission is to preserve, create, and disseminate knowledge, conserve and develop scientific, technological and cultural heritage through quality teaching and research.

## 2.4 COMPARISON BETWEEN EXISTING SYSTEMS

With that on different institutions employ various CMS solutions. A comparative analysis of existing systems highlights their strengths and weaknesses.

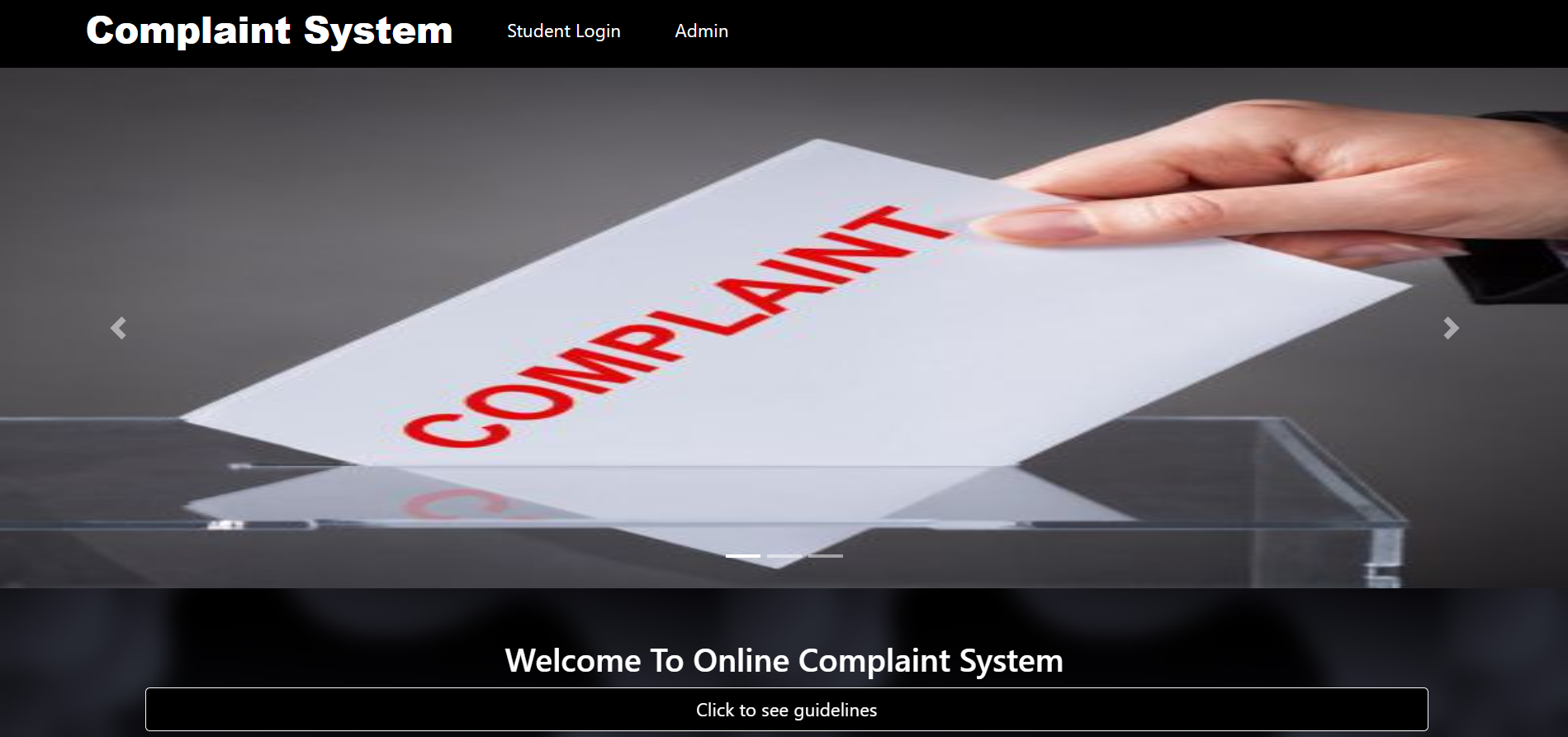
This is some comparison on existing systems and ower one

**ONE OF THE EXISTING SYSTEMS**



*Figure 2.1 one of the existing systems*

**OUR CMS**

****

*Figure 2.2 our system*

|  |  |  |
| --- | --- | --- |
| FEATURE | OTHER CMS | OUR CMS |
|  |  |  |
| Accessibility | Office-based | 24/7 Online Access |
|  |  |  |
| Response Time | Slow | Fast (Automated Processing) |
|  |  |  |
| Transparency | Low | High (Real-time Tracking) |
|  |  |  |
| User-Friendliness | Moderate | High (User-centered Design) |
|  |  |  |
| Cost | Low | Low (Digital & Automated) |

From the comparison, it is evident that a web-based CMS provides significant improvements in accessibility, transparency, and efficiency.

## 2.5 CHAPTER SUMMARY

This chapter explored the theoretical and conceptual foundations of WCMS, reviewed case studies demonstrating successful implementations, and compared existing systems. The findings indicate that web-based complaint management systems significantly enhance efficiency, accessibility, and user satisfaction. The next chapter will discuss the research methodology adopted for developing the WCMS for university students.

**CHAPTER THREE**

**SOFTWARE PLANNING AND ANALYZING**

# 3.0 INTRODUCTION

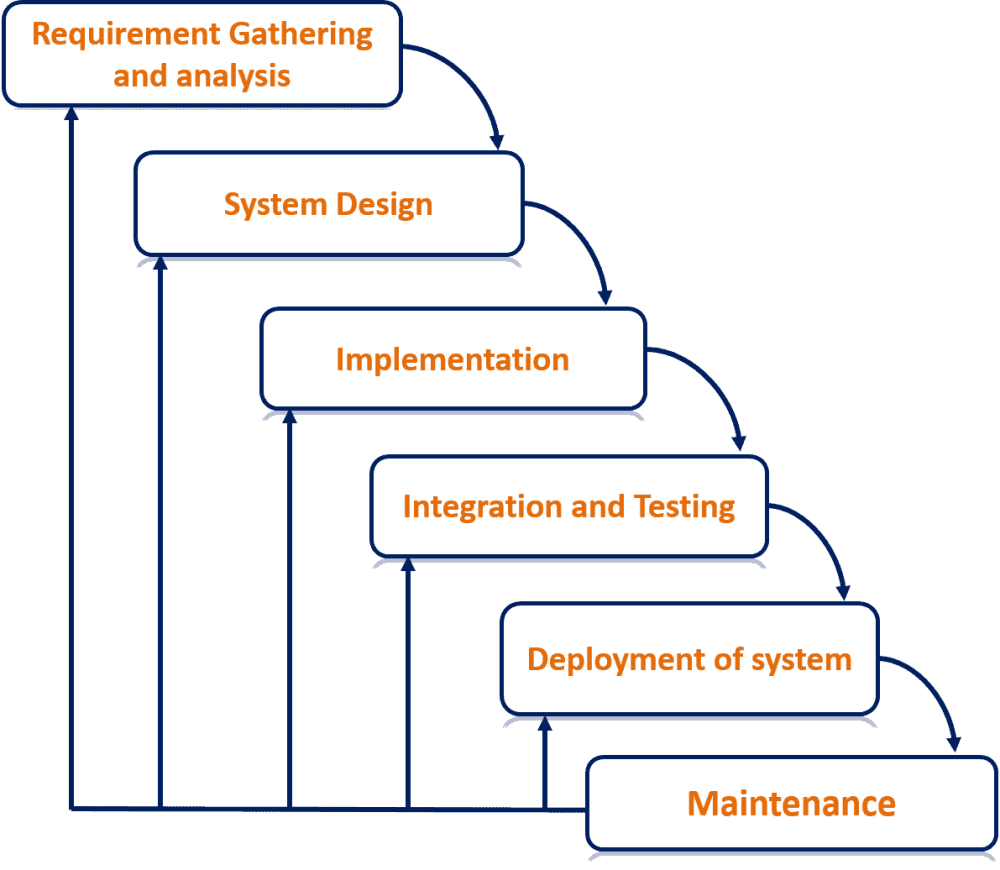
**Software planning:** Planning is an objective of each and every activity, where we want to discover things that belong to the project. An important task in creating a software program is extracting the requirements or requirements analysis. Customers typically have an abstract idea of what they want as an end result but do not know what software should do. Skilled and experienced software engineers recognize incomplete, ambiguous, or even contradictory requirements at this point. Frequently demonstrating live code may help reduce the risk that the requirements are incorrect. Once the general requirements are gathered from the client, an analysis of the scope of the development should be determined and clearly stated. This is often called a scope document. (Ralph & Wand, 2009)

**Software analyzing:** In computer science, program analysis is the process of automatically analyzing the behavior of computer programs regarding a property such as correctness, robustness, safety and liveness. Program analysis focuses on two major areas: program optimization and program correctness. The first focuses on improving the program’s performance while reducing the resource usage while the latter focuses on ensuring that the program does what it is supposed to do. Program analysis can be performed without executing the program (static program analysis), during runtime (dynamic program analysis) or in a combination of both. (Agrawal & Horgan, 2020)

This chapter will include the following sections: Introduction, Operational Framework, Work Break down structure (WBS), System Requirement, Problem Analysis Identification, Requirements Gathering Techniques, Process Modeling, Data Modeling, Suitable solution Strategies of the proposed system, System Feasibility and Chapter Summary.

# 3.1 OPERATIONAL FRAMEWORK

The Operational Framework starts planning, proposal, chapter one introduction, chapter two literature review, chapter three software planning & analyzing chapter four software design, chapter five system implementation and finally chapter six conclusions & enhancement and we gonna use waterfall model



*Figure 3.1 Operational Framework*

# 3.2 WORK BREAK DOWN STRUCTURE (WBS)

Work break down structure (WBS) is a vehicle for breaking an engineering project down into subproject, tasks, subtasks, work packages, and so on. It is an important planning tool which links objectives with resources and activities in a logical framework. It becomes an important status monitor during the actual implementation as the completions of subtasks are measured against the project plan.

**Specification**

**Development**

**Implementation**

**Validation**

*Figure 3. 2 Work Break down Structure (WBS) Process*

# 3.3 SYSTEM REQUIREMENT

System requirement is a characteristic or feature that must be include in any information system to satisfy users. Since the Administrator and the users are the main target collection of our software, I will only concern about some important functions for the admin and the user. the system needs Applications like web browsers safari, Mozilla Firefox, Google chrome and internet connection. they include the following interfaces:

1. User Interfaces: The keyboard, mouse, menus of a computer system. The user interface allows the user to communicate with the operating system.
2. Software Interfaces: The languages and codes that the applications use to communicate with each other and with the hardware.
3. Hardware Interfaces: The wires, plugs and sockets that hardware devices use to communicate with each other, computer systems, or any other medium of communication. A physical interface is the interconnection between two items of hardware or machinery.

## 3.4 SOFTWARE REQUIREMENT SPECIFICATION

One of the critical success factors in the development of high-quality software product or website is the deep understanding of the user’s real requirement as opposite to their perceived requirements. This will make the product very easy to develop, maintain, and to use the software properly by the end users.

User requirement is the process by which user desires, needs, and expectations are gathered in order to establish what the users will actually use the software for, and recorded in a way that will be meaningful both to the users and developments.

**Software Requirements**

*Table 3. 1 Software Requirements*

|  |  |  |
| --- | --- | --- |
| Operating System | : | Macintosh, Linux, Windows Xp,7,8,8.1,10,11 |
| Front End | **:** | PHP , HTML, CSS, JAVASCRIPT |
| Back End | **:** | MySQL |

**Hardware Requirement**

*Table 3. 2 Hardware Requirements*

|  |
| --- |
| 1 Computer HP/Dell/Lenovo |
| 2 Corei5 or Corei7 |
| 3 4GB or 8GB RAM |
| 4 500 or 1 TB Hard Disk or 500 GB Solid State Drive (SSD) |

**3.5 USER REQUIREMENTS DEFINITION**

A document that defines what a proposed system must be capable of doing to solve the problems of a defined set of potential users of such a system. The user requirements specification should be completely independent of any solution-oriented bias and must use terminology from the problem domain of the users. It must be understandable by the intended users who must “buy in” to it. Therefore, it is most unlikely to be created using a conventional requirements-analysis method, since these introduce solution bias, representations, and concepts that are rarely understood by (and are irrelevant to) the users (user-requirements, 2017).

**3.6 PROBLEM ANALYSIS IDENTIFICATION**

Problem analysis is focused on identifying cause and effect. It can be very difficult to determine what is cause and what is effect. For example, a problem that initially looks like a human error may be a latent human error that is the result of a poorly designed user interface, system or process.

the basic problem that we must analyse is time that consumed by Traditional complaint management system mechanism and other web–based Ones.

To avoid this problem, we need a computerized complaint management (CMS) system to manage the students’ complaints, in this work the CMS Project process analyzed in detail, and the following problems identified time consuming, manual tracking

**3.7 REQUIREMENTS GATHERING TECHNIQUES**

Requirements gathering techniques are a process of collecting data from different sources. Data or (datum in singular) is valuable pieces of information collected during the research. In order to develop a system you have to search and gather all the data and information you need to make the system because the more we get useful information about what we doing the more accurate it going to be and more suitable which is what we looking for. Data can be divided in to two categories: primary data and secondary data. In this research, the primary data comes from observation to the respondents. First, the observation will be carefully evaluated and to ensure its validity and reliability by pilot test. There are various methodologies for gathering data as the following:

* **Interviews**
* **Observations**

## 3.7.1 INTERVIEW

An interview is a formal face-to-face meeting, especially, one arranged for the assessment of the qualifications of an applicant, as for employment or admission. A conversation, as one conducted by a reporter, in which facts or statements are elicited from another, the interview is the primary technique for information gathering during the systems analysis phases of a development project. It is a skill which must be mastered by every analyst. The interviewing skills of the analyst determine what information is gathered, and the quality and depth of that information.

Interviewing, observation, and research are the primary tools of the analyst.

The interview is a specific form of meeting or conference, and is usually limited to two persons, the interviewer and the interviewee. In special situations there may be more than one interviewer or more than one interviewee in attendance

**ADVANTAGES OF INTERVIEW**

The main advantage of interviews is that the researcher (interviewer) can adapt the questions as necessary, clarify doubt and ensure that the responses are properly understood, by repeating or rephrasing the questions. The researcher/interviewer can also pick up nonverbal cues from the respondent. Any discomfort, stress and problems that the respondent experiences can be detected through frowns, nervous taping and other body language, unconsciously exhibited by any person. This would be impossible to detect in a telephone interview. So face-to-face helps the interviewee to get the desired results and help them the expression of the person to whom they are interviewing. By reading the facial expression of the respondent the interviewer can easily understand what the respondent wants to tell them about anything (answers-yahoo, 2017)

**DISADVANTAGES OF INTERVIEW**

The main disadvantages of face-to-face interviews are the geographically limitations they may impose on the surveys and the vast resources needed if such surveys need to be done nationally or internationally. The costs of training interviewers to minimize interviewer's biases for example differences in questioning methods, interpretation of response are also high. Another drawback is that respondents might feel uneasy about the anonymity of their responses when they interact face to face interviews.

**3.7.2 OBSERVATION**

An observation is a data collection method used to gather detailed information about a situation or event. Observation data is used to describe the setting, activities, participants, and the meaning of the observations from the observer's perspective. Observation in this project will be based on facts which are both accurate and detailed, but left any irrelevant or slight information that makes the description difficult to understand. This data allows the reader to fully understand the situation.

**ADVANTAGES OF OBSERVATION**

1. Allows you to directly see what people do practically rather than relying on what people say they did.
2. Observer’s document program activity takes time and cost gathering providing documentation, rather than relying on recollections of the events.

**DISADVANTAGES OF OBSERVATIONS**

1. Observer bias may occur. (Observer may only notice what interests him or her).
2. With participant’s observation, a skilled facilitator is needed to help participants present a critical evaluation.

**3.8 PROCESS MODELING**

A process model is a formal way of representing how a computer system operates. It illustrates the processes or activities that are per-formed and how data move among them. A process model can be used to document the current system (i.e. as-is system) or the new system being developed (i.e. to-be system). Whether computerized. Process modeling is the process and analysis of data and objects that are used in business or other context and identification of the relationship among these data objects. Data modeling is a first step in doing object-oriented Programming and involves professional data modelers working closely with business stakeholders, as well as potential users of information system.

**3.8.1 DATA FLOW DIAGRAM (DFD)**

Data Flow Diagrams show information transfers and process steps of a system. The general concept is an approach of a depicting how occurs input in a system, further processes and what runs out. The aim of DFD is in accomplishing of understanding between developers and users. Data flow diagrams are maintained with other methods of structured systems analysis.

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

* Graphical representation of the "flow" of data through an information system;
* Modeling process aspects;
* An overview of the system;
* For the visualization of data processing (structured design);
* What kinds of information will be input to and output from the system;
* Where the data will come from and go to;
* Where the data will be stored.(conceptdraw, 2017).

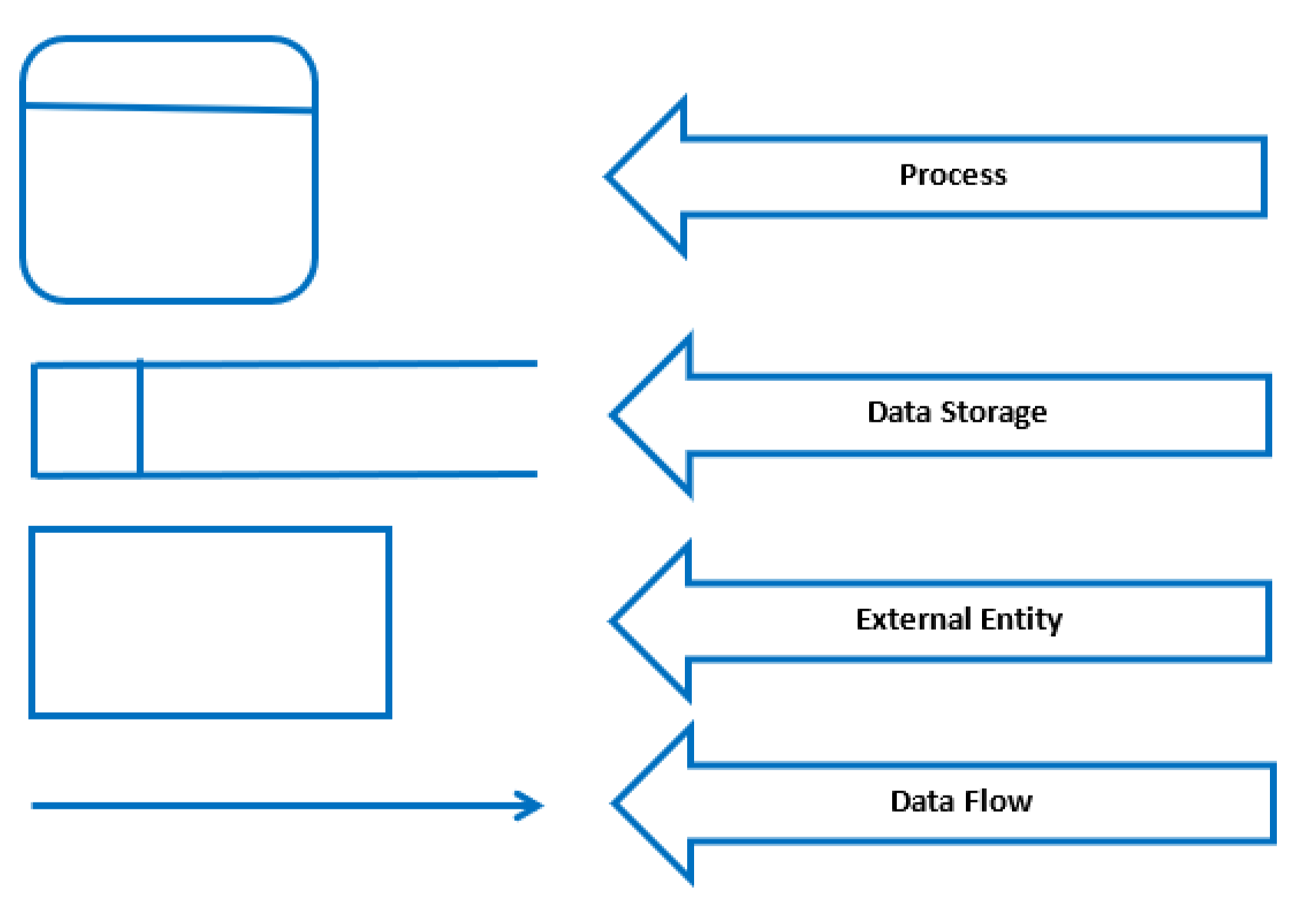


*Figure 3. 3 UCD*

Data Flow Diagrams notations:

The Data Flow Diagrams solution from the Software Development area of ConceptDraw Solution Park provides three vector stencils libraries for drawing DFD using the ConceptDraw PRO diagramming and vector drawing software.

The design elements library DFD, Yourdon and Coad notation contains 22 data flow diagram symbols of Yourdon/DeMarco DFD notation(conceptdraw, 2017)



*Figure 3. 4 UCD*

**3.8.2 UNIFIED MODELING LANGUAGE (UML)**

UML stands for Unified Modeling Language; UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects.

Today, UML is accepted by the Object Management Group (OMG) as the standard for modeling software development(smartdraw, 2017)

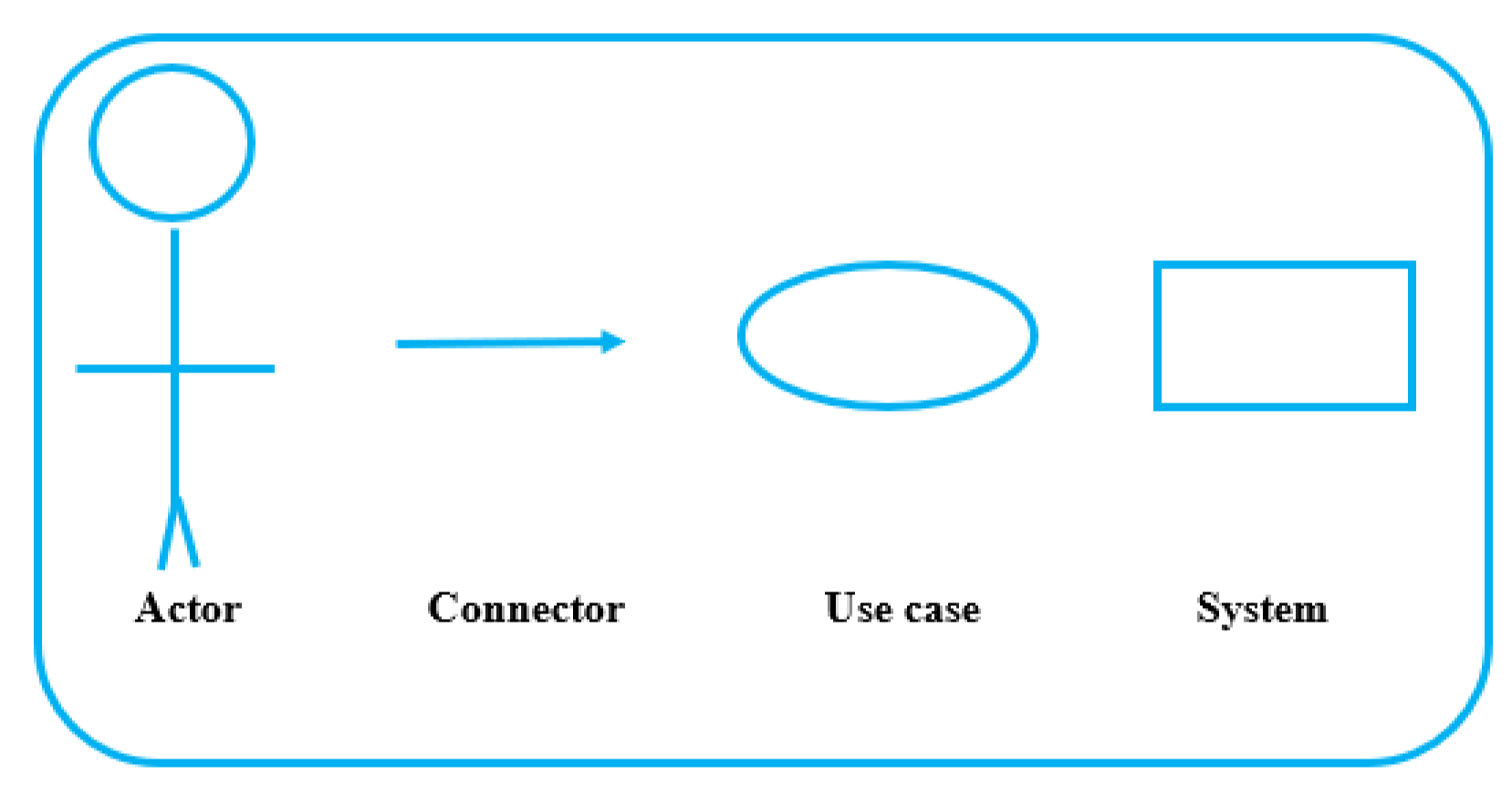
**TYPES OF UML DIAGRAMS**

The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing, and deployment.

**3.8.2.1 USE CASE DIAGRAM (UCD)**

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined(techtarget, 2017).

**Here are the symbols of UCD:-**



*Figure 3. 5 UCD*

*Figure 3. 6 Use Case Diagram*



Admin login

Manage complaint

Student registration

Edit Personal info

Student login

Manage students

Make complaint

Users

Admin

Logout

**3.9 DATA MODELING**

Data modeling is the analysis of data [objects](https://www.webopedia.com/TERM/O/object.html) and their relationships to other data objects. Data modeling is often the first step in [database](https://www.webopedia.com/TERM/D/database.html) design and [object-](https://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html)oriented programmin[g](https://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html) as the designers first create a conceptual model of how data items relate to each other. Data modeling involves a progression from conceptual model to logical model to physical schema(webo, 2017).

Data modeling is the process and analysis of data and objects that are used in business or other context and identification of the relationship among these data objects. Data modeling is a first step in doing object-oriented programming and involves professional data modelers working closely with business stakeholders, as well as potential users of information system.

**3.9.1 ENTITY RELATIONSHIP DIAGRAM**

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ERD show entities in a database and relationships between tables within that database. It is essential to have one of these if you want to create a good database design. The patterns help focus on how the database actually works with all of the interactions and data flows, although another useful tool is a Data Flow Diagram (DFD) which more directly described ERD is a detailed, logical representation of the entities, associations and data elements for an organization or business. ERD is a graphical modeling tool to standardize ER modeling; the modeling can be carried out with the help of pictorial representation of entities, attributes and relationships(lucidchart, 2107).

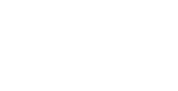
The basic building blocks of ERD are Entity, Attributes, Relationship and lines entity is an object that exists and is distinguishable from other object in other words Entity types or Entity set is a collection of similar entities; an entity may belong to more than one entity type. A relationship is an association of entities where the association includes one entity from other particular types is meaningful association before entity types. Attributes are properties of entity types in other words; entities are described in a data base by a set of attributes (Peter Pin-Shan Chen, 1976)

**3.9.1.1 ERD SYMBOLS**

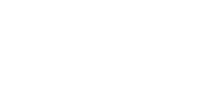
The ERD is used to represent database schema, here are some symbols used to represent ERD Symbols:

A rectangle represents an **Entity,** An Ellipse represents an **attribute**, A diamond represents a **relationship** Lines represent **linking** of attributes to entity sets & of entity sets to relationship sets.

*Figure 3.7 ERD*



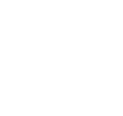
Entity



Relationship



Attribute



Line

**3.9.1.2 TYPES OF RELATIONSHIP**

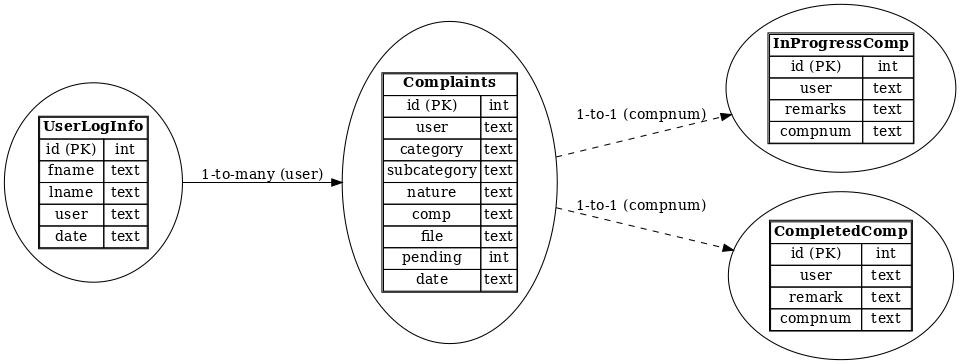
Relationship is an association among one or more entities. This relationship can be broadly classified into one-to-one relation, one-to-many relation, many-to-many relation.

* One to many Relationship Type: Abbreviated **1:M,** the relationship that associate one entity to more than one entity is called one to many relationships: - Example is country having states for one country there can be more than one states hence is an example one to many relationships.
* One to one Relationship Type: Abbreviated **1:1,** one to one relationship is a special case of one to many relationships. True one to one relationship is rare. The relationship between the president and Country is an example is one to one relationship.
* Many to Many Relationship Type: Abbreviated **M: M,** the relationship between EMPLOYEE entity and PROJECT entity is an example of many to many relationships. Many employees will be working in many projects hence the relationship between employee and project is many to many relationships.
* Many to One Relationship Type: Abbreviated **M: 1**, the relationship between EMPLOYEE and DEPARTMENT is an example of many to one relationship, there may be EMPLOYEE working in one DEPARTMENT. Hence relationship between EMPLOYEE and DEPARTMENT is many to one relationship (Aplim, 2010).

*Table 3. 3 ERD Proposed System*

|  |
| --- |
| Relation Type Representation |
| One-to-one |
| One-to-many |
| Many-to-many |
| Many-to-one |

**3.9.1.3 ERD OF THE PROPOSED SYSTEM**



*Figure 3. 8 ERD of the system*

**3.10 SUITABLE SOLUTION STRATEGIES OF THE PROPOSED SYSTEM**

As we are reaching the development phase, this section would contemplate on other ways project’s objectives can be achieved. As you may know, the system will be developed in . programming language and it is the most used or popular.

**3.11 SYSTEM FEASIBILITY**

A feasibility study for an information system project is an in-depth look at the project in order to determine whether or not an organization should proceed with its implementation. Feasibility studies provide project managers with an overview of the primary issues related to the project, as well as insights about the outcomes of the project, before the company invests too much time and money(reference, 2017).

A feasibility study could be used to test a new working system, which could be used because the Feasibility study is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. It can decide whether a process be carried out by a new system more efficiently than the existing one.

**Feasibility study is divided into four types: -**

1. Technical Feasibility.
2. Operational Feasibility.
3. Economic Feasibility.
4. Schedule Feasibility.

## 3.11.1 TECHNICAL FEASIBILITY

Technical Feasibility is defining as the feasibility that is concerned with specifying equipment and software that will successfully satisfy the requirement. It compasses the technical needs of the system Configuration of the system is given a huge importance than the actual make of hardware while examining technical feasibility.

This can be counted in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform sufficiently or not. Technological feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personal and expertise, to handle the completion of the project.

*Table 3. 4 Technical Feasibility*

|  |  |  |  |
| --- | --- | --- | --- |
| No Item Description | Quantity | Cost Per Unit | Amount |
| 1. laptop or Desktop Core i7 | 1 | $530.00 | $530.00 |
| Total |  |  | $530.00 |

*Table 3. 5 Items*

|  |  |  |  |
| --- | --- | --- | --- |
| No Item Description | Quantity | Cost Per Unit | Amount |
| **1.** PHP and MYSQL | 1 | Free |  |
| **Total** |  |  | Free |

## 3.11.2 OPERATIONAL FEASIBILITY

Operational feasibility is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operates or uses the system. To be operationally feasible, the system must fulfill a need required by the business. The following table will show the training cost of the user of the system.

*Table 3. 6 Operational Feasibility*

|  |  |  |
| --- | --- | --- |
| Activities | Duration | Expected Cost |
| **Training Current**  **Employee** | 1 Week | $350.00 |
| **Training New Employee** | 2 Week | $250.00 |
| **Total** |  | $600.00 |

## 3.11.3 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used evaluating the effectiveness of proposed system, more commonly known as benefit analysis.

The benefit analysis is to determine benefits and saving which are expected from candidate system and compare them with cost. If the benefits are more than the cost, the decision is made to design and implement the system. The cost and benefits may be direct or indirect and tangible or intangible.

*Table 3. 7 Economic Feasibility*

|  |  |  |
| --- | --- | --- |
| Types of feasibility | Description | Expect Cost |
| **Technical feasibility** | Hardware & software | $690.00 |
| **Operational feasibility** | Training with best option | $600.00 |
| **Total** |  | $1290.00 |

## 3.11.4 SCHEDULE FEASIBILITY

Typically, this means estimating how long the system will take to develop, and if it can recomplete in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is.

This involves questions such as how much time is available to build the new system, when it can be built, whether it interferes with normal business operation, number of resources required, dependencies, can the system be developed in time etc. The project duration schedule is explained in the following.

*Table 3. 8 Schedule Feasibility*

|  |  |  |
| --- | --- | --- |
| Level | Phase Name | Duration |
| **1** | Analysis phase | 4 weeks |
| **2** | Design phase | 2 weeks |
| **3** | Coding | 6 weeks |
| **4** | Implementation and Testing | 2 weeks |
| **5** | Operation and maintenance phase | 2 week |
| **Total** |  | **16 weeks** |

# 3.12 CHAPTER SUMMARY

This chapter presented the following sections: Introduction, Operational Framework, Work Break down structure (WBS), System Requirement, Problem Analysis Identification, Requirements Gathering Techniques, Process Modeling, Data Modeling, Suitable solution Strategies of the proposed system, System Feasibility and Chapter Summary.

**CHAPTER FOUR**

**SOFTWARE DESIGN**

# 4.0 INTRODUCTION

Software design is the process of implementing software solutions to one or more sets of problems. One of the main components of software design is the software requirements analysis (SRA). SRA is a part of the software development process that lists specifications used in software engineering. If the software is "semi-automated" or user centered, software design may involve user experience design yielding a storyboard to help determine those specifications. If the software is completely automated (meaning no user or user interface), a software design may be as simple as a flow chart or text describing a planned sequence of events. (Ralph &wand,2009)

There are also semi-standard methods like Unified Modeling Language and Fundamental modeling concepts. In either case, some documentation of the plan is usually the product of the design. Furthermore, a software design may be platform-independent or platform-specific, depending upon the availability of the technology used for the design. Software design usually involves problem solving and planning a software solution. This includes both a low-level component and algorithm design and a high-level, architecture design. (Ralph & Wand, 2009)

This chapter will include Architectural Design, User Interface Design, Database Storage Design, Database Design, Designing Forms and Reports and Chapter Summary.

**4.1 ARCHITECTURAL DESIGN**

The software architecture of a program or computing system is the structure or structures of the system which compromise

* The software components
* The externally visible properties of these components
* The relationships among the components

Software architectural design represents the structure of data and program components that are required to build a computer-based system

An architectural design model is transferable

- It can be applied to the design of other systems

- It can represent a set of abstractions that enable software engineers to describe architecture in predictable ways. (Preeti, 2014)

This figure will show the architectural design of the proposed system:

*Figure 4.1 ER diagram*

**4.2 USER INTERFACE DESIGN**

User interface (UI) design is the design of user interfaces for software or machines, such as the look of a mobile app, with a focus on ease of use and pleasure ability for the user. UI design usually refers to the design of graphical user interfaces—but can also refer to others, such as natural and voice user interfaces.

Since software is intangible, the only way a user can control or interact with it is through a designed usera interface. A well-designed user interface creates a user experience that the designer intended and/or a user experience that the user appreciates. Many user interfaces are designed with a focus on usability and efficiency. (ui-design, 2016)

**4.3 DATABASE STORAGE DESIGN**

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. (Lightstone, Teorey, & Nadeau, 2007)

The proposed system will use MySQL as Database Management System(DBMS).

**4.4 DATABASE DESIGN**

Database design process integrates relevant data in such a manner that is can be processed through a mechanism for recording the facts. A database of an organization is an information repository that represents facts about the organization. The database design is a complex process. The complexity arises mainly because of the identification of relationships among individual components and their representation for maintaining correct functionality are highly involved. The complexity increases if there are many-to-many relationships among individual components.

To design the system’s database, it is required to normalize the data.

**4.4.1 DATABASE NORMALIZATION**

Normalization is the process of organizing data in a database. This includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating two factors: redundancy and inconsistent dependency. Redundant data wastes disk space and creates maintenance problems. Most popular Normalization stages include.

First Normal Form(1NF), Second Normal Form(2NF) and Third Normal Form(3NF)

**4.4.1.1** **FIRST NORMAL FORM (1NF)**

A relation is in first normal form (1NF) if the following two constraints both apply: There are no repeating groups in the relation (thus, there is a single fact at the intersection of each row and column of the table) and A primary key has been defined, which uniquely identifies each row in the relation.

A database is in first normal form if it satisfies the following conditions:

* Contains only atomic values
* There are no repeating groups

**Admin**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Username (*unique*) | varchar(50) | No |
| Password\_hash | Varchar(255) | No |

**Complaints**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User | text | No |
| Category | text | No |
| Subcategory | text | No |
| Nature | text | No |
| Comp | text | No |
| File | text | No |
| Pending | Int(11) | Yes |
| Date | text | No |

**Completedcomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Inprogresscomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Userloginfo**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Fname | Text | No |
| Lname | Text | No |
| User | Text | No |
| Date | text | no |

**Userregistration**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| username | Text | No |
| Fname | Text | No |
| Lname | Text | No |
| Email | Text | No |
| Phone | Text | No |
| Gender | Text | No |
| Pass | Text | No |
| Date | text | no |

**4.4.1.2 SECOND NORMAL FORM (2NF)**

A relation is in second normal form (2NF) if it is in first normal form and contains no partial functional dependencies. A *partial functional dependency* exists when a non-key attribute is functionally dependent on part (but not all) of the primary key. To convert a relation with partial dependencies to second normal form, the following steps are required: Create a new relation for each primary key attribute (or combination of attributes) that is a determinant in a partial dependency. That attribute is the primary key in the new relation. Move the non-key attributes that are dependent on this primary key attribute (or attributes) from the old relation to the new relation.

A database is in second normal form if it satisfies the following conditions:

* It is in first normal form
* All non-key attributes are fully functional dependent on the primary key

**Admin**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Username (*unique*) | varchar(50) | No |
| Password\_hash | Varchar(255) | No |

**Complaints**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User | text | No |
| Category | text | No |
| Subcategory | text | No |
| Nature | text | No |
| Comp | text | No |
| File | text | No |
| Pending | Int(11) | Yes |
| Date | text | No |

**Completedcomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Inprogresscomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Userloginfo**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Fname | Text | No |
| Lname | Text | No |
| User | Text | No |
| Date | text | no |

**Userregistration**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| username | Text | No |
| Fname | Text | No |
| Lname | Text | No |
| Email | Text | No |
| Phone | Text | No |
| Gender | Text | No |
| Pass | Text | No |
| Date | text | no |

**4.4.1.3 THIRD NORMAL FORM (3NF)**

A relation is in third normal form (3NF) if it is in second normal form and no transitive dependencies exist. A ***transitive dependency*** in a relation is a functional dependency between the primary key and one or more non-key attributes that are dependent on the primary key via another non-key attribute. You can easily remove transitive dependencies from a relation by means of a three-step procedure: 1) For each non-key attribute (or set of attributes) that is a determinant in a relation, create a new relation. That attribute (or set of attributes) becomes the primary key of the new relation. 2) Move all of the attributes that are functionally dependent on the primary key of the new relation from the old to the new relation. 3) Leave the attribute that serves as a primary key in the new relation in the old relation to serve as a foreign key that allows you to associate the two relations. table and stays as foreign key in the old table.

A database is in third normal form if it satisfies the following conditions:

* It is in second normal form
* There is no transitive functional dependency

**Admin**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Username (*unique*) | varchar(50) | No |
| Password\_hash | Varchar(255) | No |

**Complaints**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User | text | No |
| Category | text | No |
| Subcategory | text | No |
| Nature | text | No |
| Comp | text | No |
| File | text | No |
| Pending | Int(11) | Yes |
| Date | text | No |

**Completedcomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Inprogresscomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Userloginfo**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Fname | Text | No |
| Lname | Text | No |
| User | Text | No |
| Date | text | no |

**Userregistration**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| username | Text | No |
| Fname | Text | No |
| Lname | Text | No |
| Email | Text | No |
| Phone | Text | No |
| Gender | Text | No |
| Pass | Text | No |
| Date | text | no |

**4.4.1.4 NORMALIZED TABLES OF THE PROPOSED SYSTEM**

Normalization of Database can decrease redundancy, increase efficiency and reduce anomalies by implementing three of seven different levels of normalization called Normal Forms. The first three NF’s are usually sufficient for most small to medium size applications.

**Admin**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Username (*unique*) | varchar(50) | No |
| Password\_hash | Varchar(255) | No |

**Complaints**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User | text | No |
| Category | text | No |
| Subcategory | text | No |
| Nature | text | No |
| Comp | text | No |
| File | text | No |
| Pending | Int(11) | Yes |
| Date | text | No |

**Completedcomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Inprogresscomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

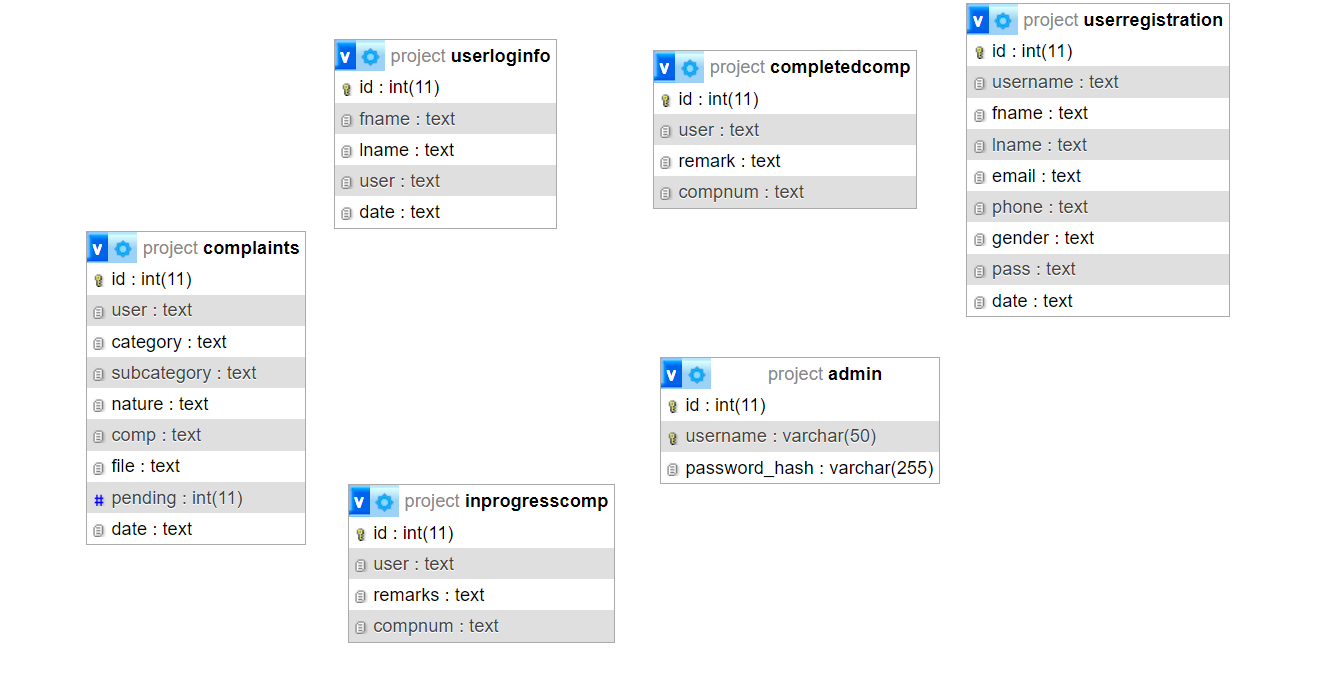
**Userloginfo**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Fname | Text | No |
| Lname | Text | No |
| User | Text | No |
| Date | text | no |

**Userregistration**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| username | Text | No |
| Fname | Text | No |
| Lname | Text | No |
| Email | Text | No |
| Phone | Text | No |
| Gender | Text | No |
| Pass | Text | No |
| Date | text | no |

**4.4.2 TRANSFORMING E-R DIAGRAMS INTO RELATIONS**

****

*Figure 4. 2 ER diagram*

**4.4.3 Data Dictionaries**

**Admin**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Username (*unique*) | varchar(50) | No |
| Password\_hash | Varchar(255) | No |

**Complaints**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User | text | No |
| Category | text | No |
| Subcategory | text | No |
| Nature | text | No |
| Comp | text | No |
| File | text | No |
| Pending | Int(11) | Yes |
| Date | text | No |

**Completedcomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Inprogresscomp**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| User (*unique*) | Text | No |
| Remark | Text | No |
| Compnum | Text | No |

**Userloginfo**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| Fname | Text | No |
| Lname | Text | No |
| User | Text | No |
| Date | text | no |

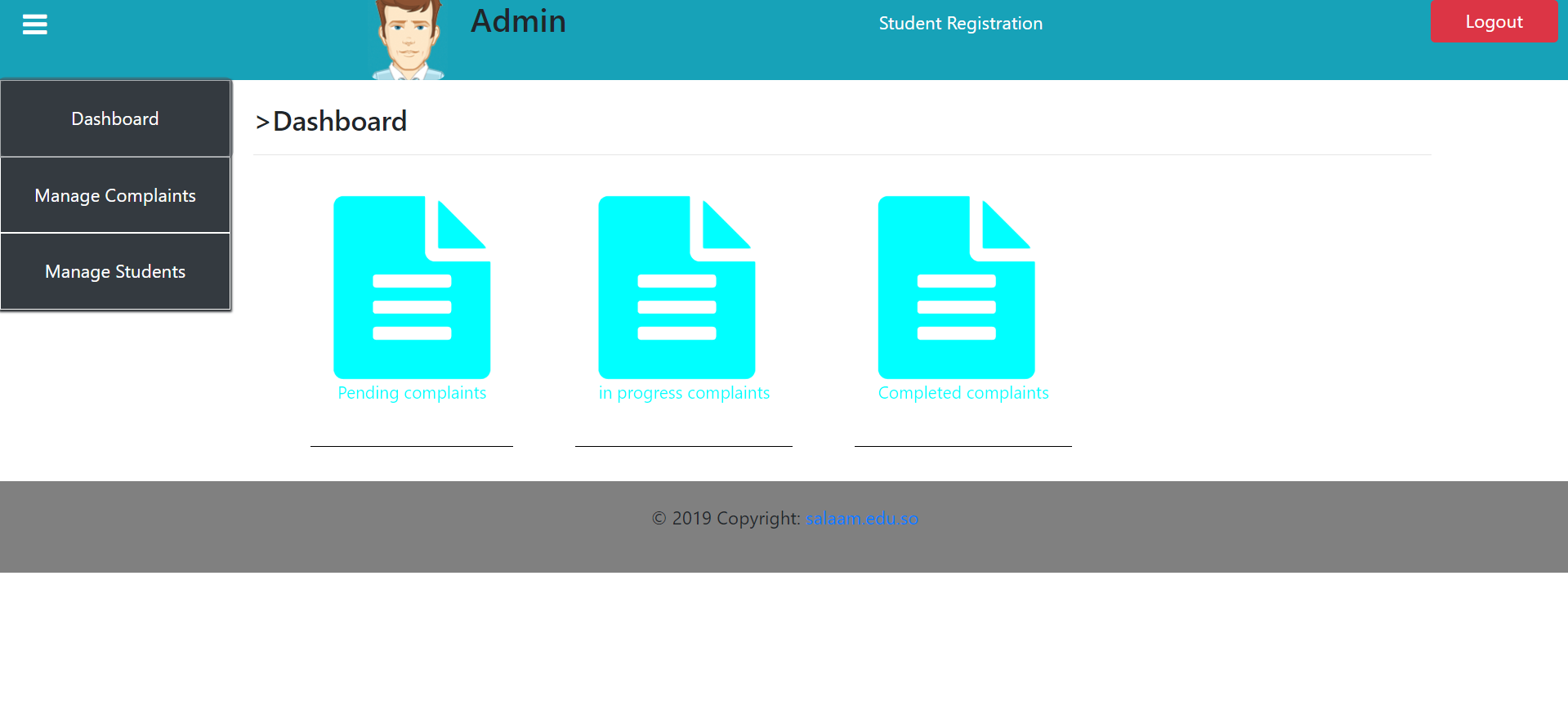
**Userregistration**

|  |  |  |
| --- | --- | --- |
| Column | Type | Null |
| ID *(Primary Key)* | int(11) | No |
| username | Text | No |
| Fname | Text | No |
| Lname | Text | No |
| Email | Text | No |
| Phone | Text | No |
| Gender | Text | No |
| Pass | Text | No |
| Date | text | no |

**4.5 DESIGNING FORMS AND REPORTS**

****

*Figure 4. 3 Student Login*



*Figure 4. 4 Admin*

****

*Figure 4. 5 Student dshboard*

**4.6 CHAPTER SUMMARY**

This chapter explained the software design process for the proposed system. It covered the overall **architecture**, showing how system components and their relationships are structured. It discussed **user interface design**, focusing on usability and efficiency, and detailed **database design and normalization** (1NF, 2NF, 3NF) to ensure data is well‑organized and consistent. The chapter also included **E‑R diagrams**, **data dictionaries**, and examples of **forms and reports** such as the student login interface. Overall, it highlighted how proper design ensures the system is efficient and user‑friendly.

**CHAPTER FIVE**

**SYSTEM IMPLEMENTATION**

**5.0 INTRODUCTION**

System implementation is a process of ensuring that the information system is operational. Implementation allows the users to take over its operation for use and evaluation. It involves training the users to handle the system and plan for a smooth conversion. (implementation, 2018)

This chapter will include Coding phase, Test system implementation, development testing, release testing, user testing, developing user manuals and chapter summary.

**5.1 CODING PHASE**

The coding phase of the software life-cycle is concerned with the development of code that will implement the design. This code is written is a formal language called a programming language. Programming languages have evolved over time from sequences of ones and zeros directly interpretable by a computer, through symbolic machine code, assembly languages, and finally to higher-level languages that are more understandable to humans. (INN, 2009)

This Phase is devoted to providing access to most of the computer programs that the researcher used to prepare the data and apply the programming is techniques, instructions on how to construct this application project using the software available on the attached compact disk (CD).

We used PHP and MySQL to develop the proposed system.

**5.2 TEST SYSTEM IMPLEMENTATION**

The testing phase involves some modification to the pervious design phase and system testing has been done to minimize the programming errors.

Testing the system is a very important stage to ensure that all system requirements have been developed without errors. System testing can be done through some stages. These include.

1. Developing test
2. Release test
3. User test

**5.2.1 DEVELOPMENT TESTING**

Development testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. Development testing is always done by the developer who is responsible programming stage.

This testing shows as if wrong username and password is entered can’t allowed to access the system.

**5.2.2 RELEASE TESTING**

Prior to making a software system available for public consumption, a series of tests should be conducted to ensure the software runs as intended. Flaws are fixed and inefficiencies removed to come up with a program that runs seamlessly. A battery of tests is conducted with a view towards identifying and fixing bugs, which are a precursor for system failure. It is an essential part of the Software Development Life Cycle (SDLC), and may perhaps be considered the most important aspect, aside from design and coding. Indeed, the quality of the product plays a huge part in setting the stage for the acceptance and success of subsequent ones in the market. With this in mind testing hitherto release of software should not be taken for granted. (Brown, 2016)

The ability to evaluate and ensure the quality of in-process and/or final product based on process data, which typically include a valid combination of measured material attributes and process controls. A release is the distribution of the final version of an application. A software release may be either public or private and generally constitutes the initial generation of a new or upgraded application.

**5.2.3 USER TESTING**

User testing refers to a technique used in the design process to evaluate a product, feature or prototype with real users. There are several reasons why you might want to undergo usability testing, the most common is that it allows the design team to identify friction in a user experience they are designing, so that it can be addressed before being built or deployed. Identifying any issues early reduces the long-term cost. (user-testing, 2018)

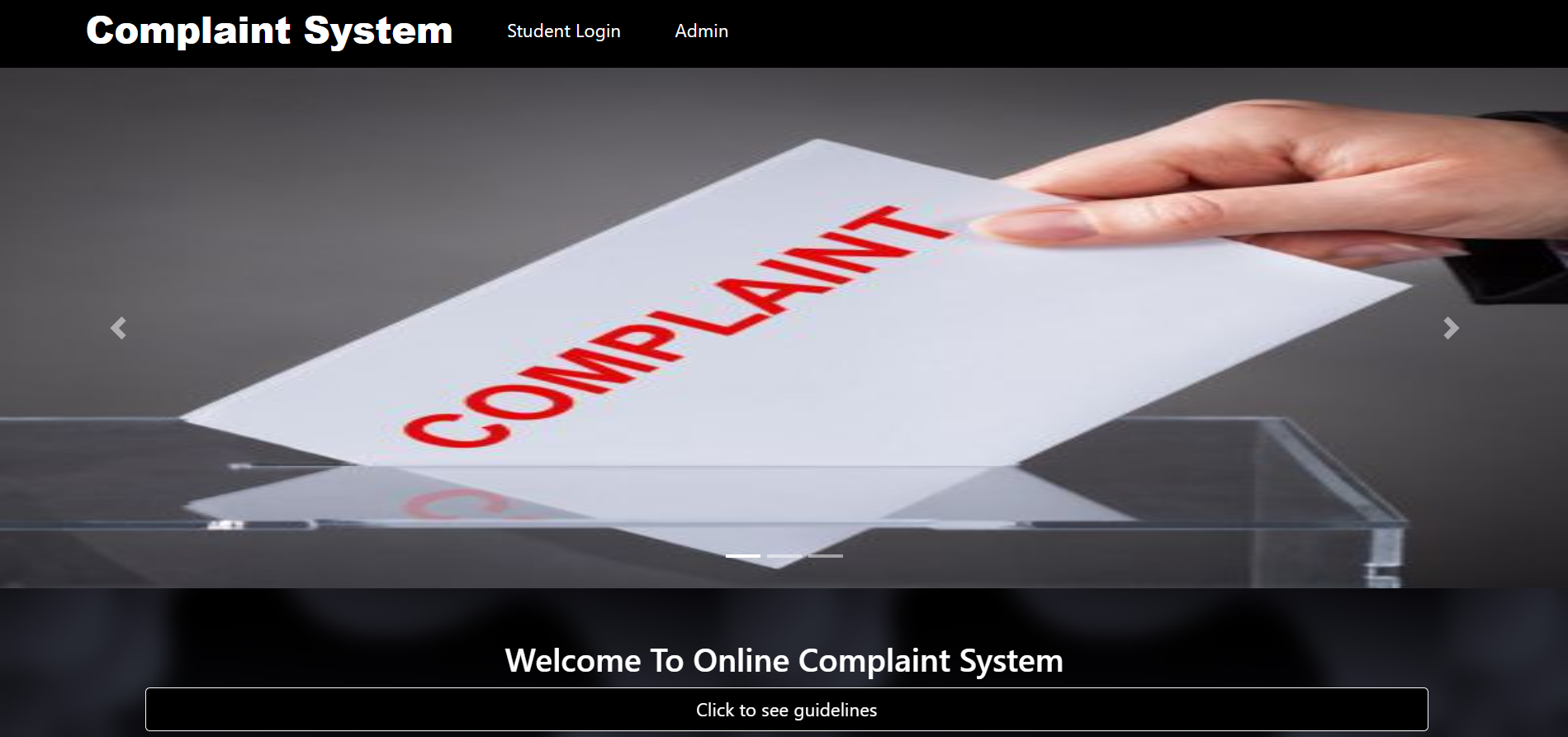
This system has been tested by the user (alpha testing), the developer and the user (beta testing), and the user only in his work location (Acceptance testing).

**Alpha Test**: An alpha test is a preliminary software field test carried out by a team of users in order to find bugs that were not found previously through other tests. The main purpose of alpha testing is to refine the software product by finding (and fixing) the bugs that were not discovered through previous tests. (alpha-test, 2017)

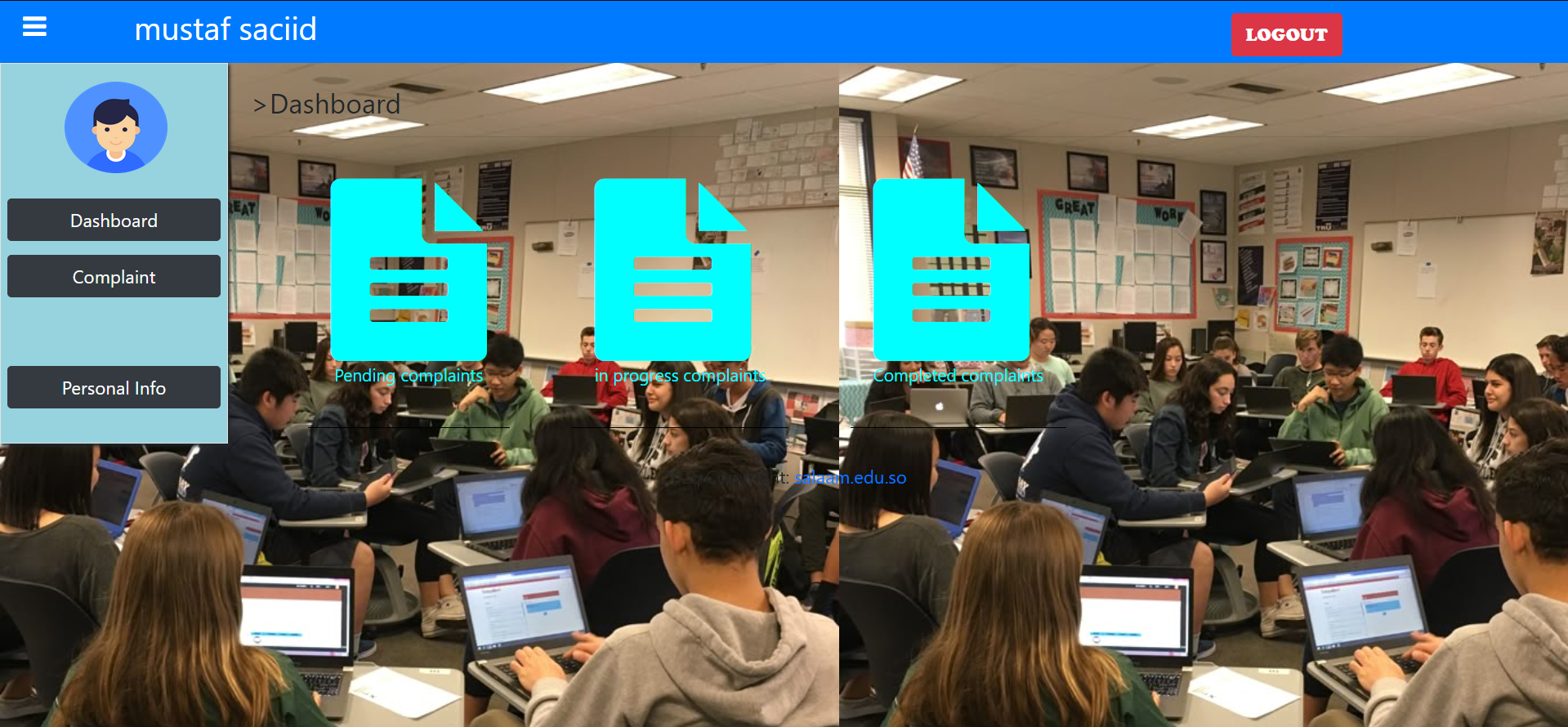
**Beta Test**: Beta testing is also sometimes referred to as user acceptance testing (UAT) or end user testing. In this phase of software development, applications are subjected to real world testing by the intended audience for the software. The experiences of the early users are forwarded back to the developers who make final changes before releasing the software commercially. (Rouse, 2015)

**5.3 DEVELOPING USER MANUALS**

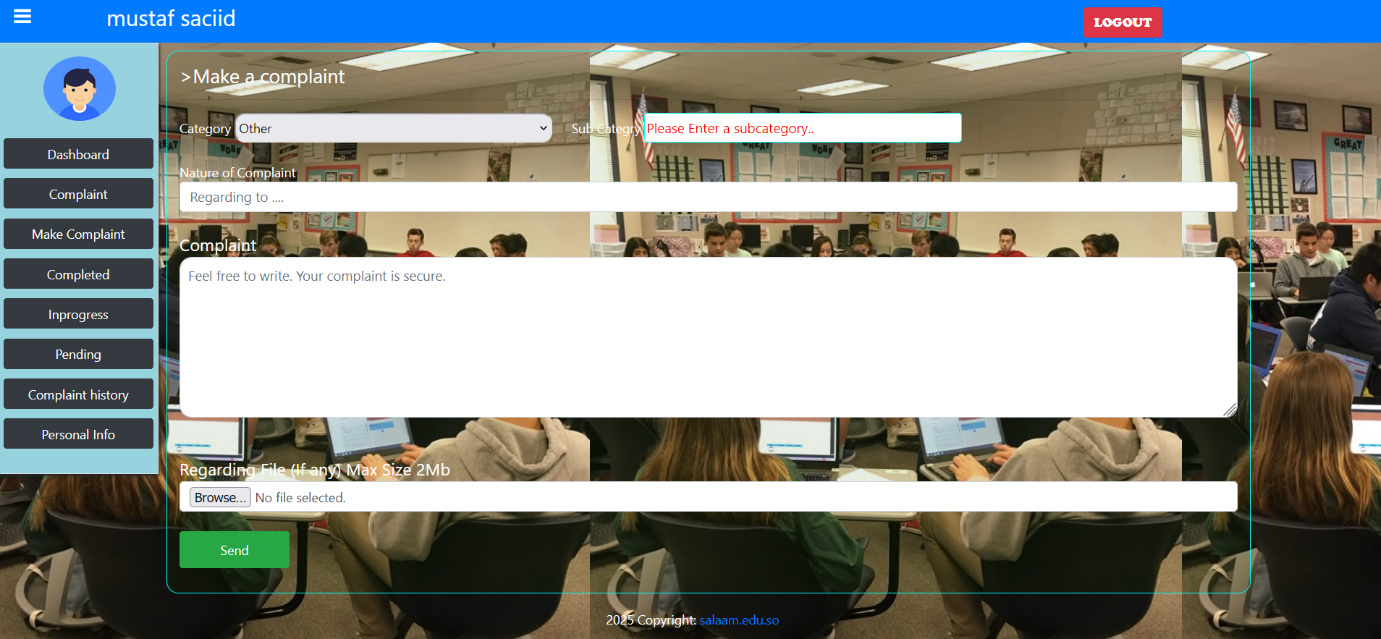
In computer software product development, documentation is the information that describes the product to its users. It consists of the product technical manuals and online information (including online versions of the technical manuals and help facility descriptions). The term is also sometimes used to mean the source information about the product contained in design documents, detailed code comments, white papers, and blackboard session notes. The term is derived from the idea that engineers and programmers. Also, is a general term for a multiplicity of documents in a chosen mix of media and with a certain collection. Purpose of documentation is the use to support a tool or a process Classical documentation is a set of documents printed on paper. The researcher presents here some of the main forms of the proposed system with some explanation.

****

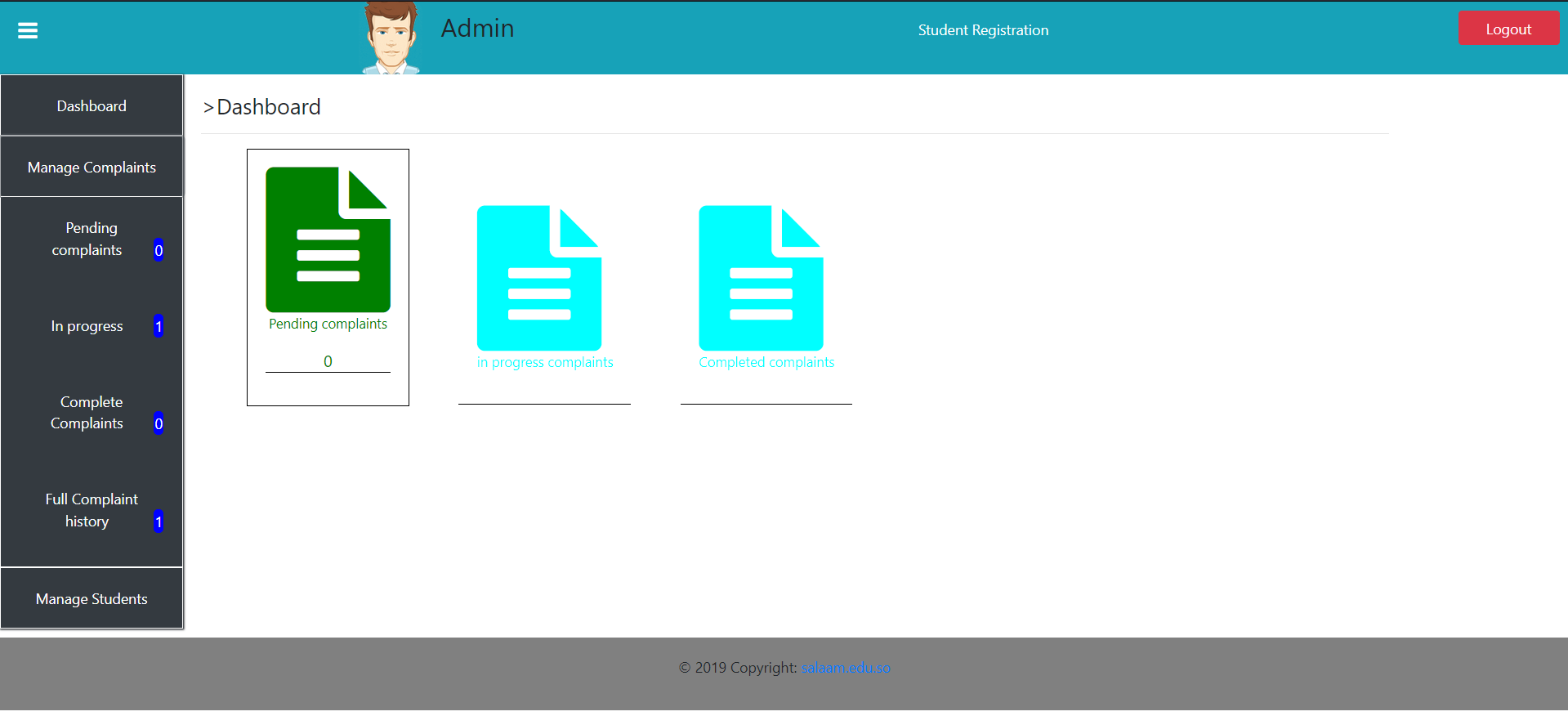
*Figure 5.1 interface*

****

*Figure 5.2 Student Dashboard*

****

*Figure 5.3 Student Complaint*

****

*Figure 5.4 Admin Manage Complaint*

**5.4 CHAPTER SUMMARY**

This chapter explained the implementation and testing of the system. Developing the system requires a server, host to access the portal. The purpose of the Testing is to check the errors and analyse the problem in order to develop a successful System that meets the student’s requirements.

**CHAPTER SIX**

**CONCLUSION & ENHANCEMENT**

**6.0 INTRODUCTION**

This chapter outlines the objectives successfully achieved throughout the course of the project and illustrates how the various components were implemented. It also reflects on the experiences gained during the development process. Furthermore, it offers recommendations for future enhancements that could improve the system’s functionality and expand its features.

**6.1 OBJECTIVE ACHIEVEMENT**

The key objective for developing this project is to develop web-based students complaint management system and the project will be applicable to the other organizations like this. Furthermore, here are the details of the project objectives that have been accomplished.

1. **To plan an easy complaint management system for students to apply complaints**  
   A user-friendly system was carefully planned and structured to ensure students can easily submit their complaints through a simplified process.
2. **To analyze the complaints made by students**  
   The system provides administrative tools to view, categorize, and monitor complaints, enabling better understanding and faster resolution of issues.
3. **To design, implement, and test a complaint management system**  
   A fully functional web-based complaint management system was successfully designed, developed, and tested, meeting the project requirements and demonstrating reliable performance.

All the stated objectives have been successfully achieved, and the user’s needs and challenges have been effectively addressed through the implementation of the project.

**6.2 WEAKNESSES AND PROBLEMS OF SYSTEM**

Although the proposed Complaint Management System (CMS) significantly improves the process of submitting and tracking complaints, several weaknesses and problems were identified during and after its implementation:

1. Limited Functional Features

The current system focuses primarily on complaint submission, storage, and basic tracking. Advanced features such as automatic escalation of unresolved issues, AI‑based categorization, and feedback analysis are not yet integrated, limiting the overall functionality.

1. Scalability Challenges

The system was designed for use within Salaam University and may face performance challenges if the number of users, complaints, or departments grows beyond the initial design capacity. Future scaling would require redesigning parts of the database and optimizing queries.

1. Dependency on Internet Connectivity

As a web‑based application, the system relies completely on stable internet access. In areas with poor connectivity, students and staff may experience delays in submitting or retrieving complaints.

**6.3 STRENGTH OF SYSTEM**

The Complaint Management System has several good points that make it very helpful for the university.

It gives students and staff one clear place to send, follow, and handle complaints, which makes things much more organized than using papers or spreadsheets.

Because everything is online, students can send their complaints anytime and from anywhere, without having to visit offices in person. This makes the process faster and less stressful for both students and administrators.

The system also makes it easier for staff to keep track of complaints and solve them on time, while reports can be created to see how things are going and what needs to be improved. Overall, the system makes communication smoother, saves time, and helps the university respond to student issues in a better and more reliable way.

**6.4 FUTURE WORK**

Although the Complaint Management System has achieved its main objectives, there are still opportunities to enhance its performance and extend its capabilities in the future. By adding new features and improving existing ones, the system can become more efficient, secure, and user‑friendly. The following points highlight the areas that can be considered for future development:

* Develop a mobile application or re-design the interface to be fully responsive, allowing students to access the system easily on smartphones and tablets.
* Implement automatic notifications, such as email or SMS alerts, to update users whenever the status of their complaint changes.
* Integrate artificial intelligence tools to automatically categorize and prioritize complaints, improving response time.
* Strengthen security measures by adding two‑step verification and encrypting sensitive information to protect user data.
* Expand the system’s usage beyond Salaam University, making it adaptable for other institutions or departments.
* Add advanced reporting and analytics, including dashboards to monitor trends and support better decision-making.

**6.5 CHAPTER SUMMARY**

In this chapter, the main aspects of the system were discussed, including its strengths, weaknesses, and possible future improvements. The strengths of the system show how it has made the complaint process faster, more organized, and easier for students. Overall, this chapter reflects on what the system has achieved so far and how it can continue to grow and improve in the future.

**REFERENCES**

*Ifeanyi Alex, Dr. A. O. (2023). An Automated and Robust Tertiary Institution Web-Based Student Complaint Management System. International Research Journal of Innovations in Engineering and Technology, 07(12), 110–124.* [*https://doi.org/10.47001/irjiet/2023.712016*](https://doi.org/10.47001/irjiet/2023.712016)

*Iso. (n.d.). General requirements for the competence of testing and calibration laboratories Exigences générales concernant la compétence des laboratoires d’étalonnages et d’essais COPYRIGHT PROTECTED DOCUMENT. www.iso.org*

Mohamed, H. M., Mohamed, A. H., Hashi, S. O., Hassan, H. N., Barre, K. A., & Ahmed, B. A. (2025). Automated University Complaint Management System by Leveraging Machine Learning and Natural Language Processing for Enhanced Efficiency and Accuracy. *Forum for Linguistic Studies*, *7*(2). <https://doi.org/10.30564/fls.v7i2.8162>

Okokpujie, K., Daramola, D., Okokpujie, I. P., Daramola1, D., & Student, R. (2023). Issue 11 IJSDR2311082 www.ijsdr.org. *International Journal of Scientific Development and Research*, *8*, 555. www.ijsdr.org

Setiadi, R., Setyowati, R., Iskandar, K., Syaifulloh, M., Abadiyah, A., Yulianto, A., & Ikhwan, S. (2022). The Effect of Perceived Price and Service Quality on Consumer Satisfaction of Healthy Baby Food Counters. *Food Science and Technology (United States)*, *10*(2), 17–22. <https://doi.org/10.13189/fst.2022.100201>

Singh Rana, R. (n.d.). Streamlining Complaint Management: A Comprehensive Project Analysis. *Journal of Pharmaceutical Negative Results ¦*, *14*, 2023. <https://doi.org/10.47750/pnr.2023.14.S02.346>