Addis Ababa University

College of Natural and Computational Sciences

Design of a Generic Multi-Channel Clearance Management System

Ermias Getachew Assfea

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia
Jun, 2020

Addis Ababa University

College of Natural and Computational Sciences

Ermias Getachew Assfea

Advisor: *Fekade Getahun (PhD)*

This is to certify that the thesis prepared by *Ermias Getachew Assfea*, titled: *Design of a Generic Multi-Channel Clearance Management System* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Name	Signature	Date
Advisor: Dr, Fekade Getahun		
Examiner: Dr, Dagmawi Lemma		
Examiner: Dr. Minale Ashagrie		

DEDICATED TO:

My mom *(Tiruwork)*

Acknowledgments

Most of all, I would like to thank God, who makes everything possible, for helping me pass all those hard times that I will never forget in my life.

I owe my deepest gratitude to my advisor Dr. Fekade Getahun for his time, patience, and undeniably helping comments through this study. an inspiration for me to proceed whenever I face difficulties and he is easily approachable. This thesis would not have been possible without his constructive comments on every aspect of the study.

I like to pass my sincere gratitude to my wonderful classmates for creating such an exciting batch. I will always cherish the time we have spent together. Finally, I want to thank all the people who have contributed in one way or another on this thesis work.

Table of Contents

List of Tables

List of Figures

List of Algorithm

List of Annexes

Acronyms

AAU	Addis Ababa University
API	Application Program Interface
ASP	Application Service Provider
CDMA	Division Multiple Access
CDN	Content Delivery Network
EAV	Entity Attribute Value
EMI	Electromagnetic Interference
ETSI	European Telecommunication Standard Institute
GCMS	Generic clearance management system
GPM	Generic Process Model
GSM	Global System for Mobile
HCI	Human Computer Interaction
HEI	Higher Education Institutions
JSON	Java Script Object Notation
ODBC	Open Database Connectivity
Rest	Representational State Transfer
SOA	Service Oriented Architecture
SQA	Software Assurance Point
SSADM	Structured System Analysis and Design Methodology
TDMA	Time-division multiple access
URI	Uniform Resource Identifier
3GPP	Third Generation Partnership Project

Abstract

A clearance is a certificate permitting to disengage from an institution. It is very essential and must be implemented taking into account the different enterprise and organizational assets into account. The current known approaches to clear an entity is user collects the Clearance Request Form physically and fill the required information. A user appears in person in the office of stakeholder, look for the person responsible to verify for a sign. After properly completing such forms and getting them signed by the appropriate offices, the user submits it to the office. The main drawback of the existing approach the integrity of information is not fully guaranteed, Lack of data security in Unauthorized access and Accidental loss, it needs a physical presence in each office which are disintegrated, the process is time-consuming, loss of some files and document.

This research aims to design and develop a generic clearance system that will apply for any organization and any entity to clear to reduce delays in the manual process and create a central repository for entities to be cleared. The designed system will serve as a more reliable and effective means of undertaking user clearance, remove all forms of delay and stress as well as enable the user to understand procedures involved, as well as how to do clearance.

We proposed a generic clearance management system that can clear any entity and use for any company at any time. we designed a generic system that accepts variabilities related to entities, organizations, and their business flow which are not currently available in the existing clearance platform. The proposed system enables to register a material for the participant, borrow some material, return a borrowed item, request clearance, approve the request, and give certification at any time. Based on the proposed system, a clearance management that shows all the features of the system are developed. In this work, the implementation of the computer-based system using PHP, HTML, CSS, and MySQL at the backend. Finally, a usability experiment was carried out, and, 96% of participants have agreed that the generic clearance management system has higher usability than the previous manual method.

Keywords: *clearance system, generic, distributed, EAV, clearance management*

Chapter 1 – Introduction

1.1 Overview

A clearance is a certificate permitting to disengage an entity from an institution [1]. The term "clearance" is also used in organizations that have a formal method to check employees for access to sensitive information [2]. A clearance system is very essential and it must be implemented in all organizations.

For example, in Higher Education Institutions (HEI), at the end or in the middle of an academic year a student undergoes a clearance process as a precondition to disengage from the HEI. The process of cleaning involves the participation of different stakeholders. The stakeholders are dispersed in a different geographical area; they may use different/ heterogeneous systems that support the clearance process.

In different organizations they have a clearance procedure for user and item to assure all user have paid all their fees, have returned all material, and have no outstanding equipment yet to be returned, the return of all property and items. Users must complete this process before they will be allowed to take their certificate.

The idea of the clearance process is the same for any organization, it is clearing from the resources associated with the user to clear from the company. The participants use a different mechanism to check the user some office use their own system same use manual methods and there is no comment model or framework to facilitate and integrate between this resources and use full system. The resources are distributed in a different location and use a heterogeneous system for the clearance process and they need to integrate each other by using a distributed system concept.

A distributed system is a collection of independent computers that appears to its users as a single coherent system [3]. This definition has several important aspects. The first one is that a distributed system consists of components (i.e., computers) that are autonomous. This means that one way or the other the autonomous components need to collaborate [3]. In the clearance process, the responsible stakeholders for the clearance process need to integrate and communicate with each other for finalizing the user clearance request.

This research aims to design and develop a generic clearance system that will reduce delays in the manual process and create a central repository for entities to be cleared.

1.2 Motivation

One motivation for conducting this work is the absence of a generic clearance management system that facilitates the clearance process. Considering the Addis Ababa University, for instance, there is a lack of integration among the different offices that work in the clearance process. In the current practice, most of the offices do not use a system that can be consulted before signing for the student/ staff clearance. A student/ staff should appear in person in the office of stakeholder, look for the person responsible to verify for compliance and sign.

The main challenges associated with the clearance process are:

1. Lack of data security related to clearance,
2. It needs a physical presence in each office which are disintegrated,
3. The process is time-consuming and it takes at least 2 or 3 working days to clear,
4. Loss of some files and document.

There are some offices that use their own system to check the student status before signing the clearance form. Thus, the new system needs to get relevant data from those already existing systems.

The following is a hypothetical futuristic scenario that guides the new system. A student in the College of Natural and Computational Sciences has to clear at the end of the academic year. He fills in the clearance request form. In the case of AAU, the clearance process starts once the batch advisor of the student sign on the application form. The system sends an email to the batch advisor automatically stating the clearance request, when it was filled. The advisor logs into the web application and gets the list of applications already made, he/ she assesses the request and approves or rejects.

If the request is rejected or pended for some reason, the system sends a notification to the requester through email. If approved stakeholders involved in the clearance process will be notified. The notification to the stakeholders depends on priority, task dependency order, or other criterions configured in the business process of the clearance.

If a specific stakeholder is not responding as per the maximum wait period stated, the system automatically escalates the request. After all the stakeholder's sign on the request, the system sends a notification about the completion,

The design of an efficient, distributed information system that uses delivery channels will help an actor to file clearance requests and empower responsible personnel to get service request messages, check, validate and approve the request. When the clearance is done, it sends automatically an email message to the service requester.

1.3 Statement of the problem

In any organization, there is a clearance process that disengages an entity/ requester from an organization. This process is done mainly manually checking the record of the requester and commitment and linked resources and service in the information system. The lack of integration from the responsible office that makes the process difficult inefficient and time-consuming.

The current clearance practice in the different organization shows the following problems:

- The integrity of information is not fully guaranteed,
- Too much time is wasted
- The existing approach is only capable of attending to a limited number of users a day,
- The user should roam from office to office.

Recalling the motivation scenario stated in the previous section- motivation, the main issues to be considered in this research are.

- How to represent the basic feature of entity to be cleared? (Entity could be person, object, student, employee, etc.)? And the purpose of the clearance? (such as disengage the entity or as a security clearance,)
- How to model the basic clearance business process of an organization generically?
- How to model the system architecture of generic clearance process?

The purpose of this work is to study the current system and develop a generic clearance management system that will apply for any organization and any entity. The system facilitates charging and discharging of clearance and keeping records of eligible users.

1.4 Objectives of the Study

General objectives

The general objective of this work is to develop and implement a Generic multi-channel clearance management system.

Specific objectives

To realize the above general objective, the following specific objectives are identified:

- Understand the current clearance business process of different institutions (AAU is just one case)
- Develop a mechanism to integrate the work process (using web service, Web API or Database)
- Develop system model for clearance. The model shall be capable of configuring the business process, rules, and dependency between activities.
- Design the generic clearance management system
- Build the prototype

1.5 Significant of the study

The work will help in a good number of ways to ease the queuing system in the organization, creates accountability, transparency, and increase the efficient use of resources. Clear advantages of a new generic system of clearance processing over those of traditional manual systems are higher yields. A new system allows users to check their clearance status.

There are many other advantages of a new generic clearance system and some of them are listed below.

- It saves a lot of time to finish the clearance
- The requester can see clearance statues and can get clearance certificate online
- It avoids device limitations by giving mobile service and web-based service at any time
- Information processing is very fast and delays can be avoided
- It also helps to reducing costs such as labor and stationary

1.6 Method of the study

To design the new system, a service-oriented software design approach will be applied.

A. System Analysis

The systems analysis methodology for developing and implementing the generic clearance system is presented below.

Service-oriented modeling is the discipline of modeling business and software systems, to design and specify service-oriented business systems within a variety of architectural styles and paradigms, such as application architecture, service-oriented architecture, microservices, and cloud computing [7].

Any service-oriented modeling method includes a modeling language that can be employed by both the 'problem domain organization' (the Business), and 'solution domain organization' (the Information Technology Department), whose unique perspectives typically influence the service development life-cycle strategy and the projects implemented using that strategy [8].

Service-oriented modeling strives to create models that provide a comprehensive view of the analysis, design, and architecture of all 'Software Entities' in an organization, which can be understood by individuals with diverse levels of business and technical understanding [7].

B. Develop system architecture.

The system architecture provides a road map for the systems building process. It puts the system components into perspective, specifies the system functionalities, and defines the structural relationships and dynamic interactions among system components. It must identify the constraints imposed by the environment, state the objectives of the development efforts, and define the functionalities of the resulting system to achieve the stated objectives.

C. System design

A research project's requirements may be driven by new functionalities envisioned by the researcher or may be determined partially by the research sponsor's requests. Design, one of the most important parts of a system development process, is rooted in engineering [7]. It involves the understanding of the studied domain, the application of relevant scientific and technical knowledge, the creation of various alternatives, and the synthesis and evaluation of proposed alternative solutions.

The system must work in a form of mobile service and online web applications. Records deleted in the system shall be kept in the database.

D. System Implementation

"Building a prototype system is an engineering concept" [8]. Researchers in systems development often conduct their research by building a prototype system. Implementation of a system is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project [10].

The process of implementing a working system can provide researchers with insights into the advantages and disadvantages of the concepts, the frameworks, and the chosen design alterative. This new system is structured to work with the standard software development procedure to achieve effective design goals and objectives effectively distribute the clearance system by developing in PHP and MYSQL. The system must not contain errors making some system functionalities unavailable or errors disturbing the user while working with the system [12].

E. System testing

Test performance and usability have impacts on individuals, groups, or organizations. The test results should be interpreted and evaluated based on the conceptual framework and the requirements of the system defined in the earlier stages [10].

- Program Testing: After coding, program testing is needed to ensure that the program is working by the set objectives
- Evaluation: Once the system comes into action, it is necessary to be monitored to ensure that set objectives are being achieved. The system development life cycle is reviewed accordingly. This type of testing is called Usability Testing - mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives done by AAU ICT office and registrar
- Maintenance: The life span of an application can be significantly extended through proper maintenance. Periodic maintenance should be scheduled because preventive maintenance is better than corrective type. Preventive maintenance is daily maintenance which improved system reliability.

Delivery channels:

The clearance system is delivered by different approach like web-based and mobile app

Web-based: the administrators and all office use a web-based platform to add edit and update the data. The user may use this service to check their states and to send clearance requests and other services.

Mobile service: this service is used by the user to use by their phone to send a request and to see their states and to gate the clearance certificate.

Email service: it is mail service which is a delivery notification by admin to the user email address

1.7 Scope and Limitations

Scope:

This work is scoped to developing a generic clearance management system. The system will be developed using PHP and MYSQL to manage both the database and the interface of the system. The developed system will be tested for clearing AAU students and the same logic can be used to clear other entities such as employees.

Limitations:

The HCI will be only in English and hence local language will not be considered

1.8 Application of the Study

Apart from being research to fulfill the requirement of the Master's program, the result of this study is believed to be used either as an input for other researches or can be put into use in different fields. The possible applications of the system which is built with this research are;

- ✓ To be used in any clearance type for any organization
- ✓ To be used as input to integrated to other systems
- ✓ Also, the study can open a way for further researches in the area clearance management.

1.9 Thesis Organization

The rest of the chapters in this report are organized as follows. Chapter two deals with the literature reviewed and related work so far. It includes Concepts Related to the clearance system and other generic based applications and the existing methods for clearance management and other related research will be addressed. Chapter three describes the Design and Implementation of the proposed generic clearance system framework. The Experiment and Evaluation of the system are discussed in Chapter four. Finally, conclusions are drawn from the thesis result, the contributions of this research work, and recommendations on possible future works related to this research are given in Chapter five.

Chapter 2 - Literature Review and Related work

In this Chapter, extensive reviews of concepts in generic clearance management system, known approaches, methods and related literature are reviewed. The extensive literature has been reviewed to understand the problem associated with the realm of this thesis and also to identify the appropriate solution. The last subsection details related works.

2.1 Concepts Related to Generic system

A generic system is one that is representative of all systems, be they physical, sentient, or more abstract [15]. One can define a boundary anywhere; the system it defines will acknowledge the same procedure.

2.1.1 Generic process framework

A framework is a standard way to build and deploy applications [15]. a process framework establishes the foundation for a complete software process by identifying a small number of framework activities that are applicable to all software projects, regardless of size or complexity. The software process framework includes all set of umbrella activities. It also includes a number of framework activities that are applicable to all software projects [20].

According to [15] A generic process framework encompasses five activities which are given below one by one:

1. **Communication:**

This framework activity involves heavy communication and collaboration with the customer (and other stakeholders) and encompasses requirements gathering and other related activities.

2. **Planning:**

This activity establishes a plan for software engineering work that follows. It describes the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.

3. **Modeling:**

Modeling is about building representations of things in the 'real world'. It consists of complete requirement analysis and the design of the project like algorithm, flowchart, etc. The algorithm is the step-by-step solution to the problem and the flow chart shows a complete flow diagram of a

program. In modeling activity, a product's model is created in order to better understanding and requirements.

4. **Construction:**

In software engineering, construction is the application of a set of procedures that are needed to assemble the product. In this activity, it consists of code generation and tests the product in order to make a better product. The coding part implements the design details using an appropriate programming language. Testing is to check whether the flow of coding is correct or not. Testing also checks that the program provides the desired output.

5. **Deployment:**

In this activity, complete or non-complete products or software are represented to the customers to evaluate and give feedback. If the customer wants some corrections or demands for the additional capabilities, then the change is required for improvement in the quality of the software.

2.1.2 Generic process model

The Generic Process Model (GPM) is a formal framework for process analysis [22]. It uses concepts from Bunge's ontology and extends this ontology with process-related concepts. The GPM-based process analysis methods are language-independent and can be used for process models at different modeling languages mapped to GPM [22].

The main concepts employed by GPM are domain, state, event, goal, transition, and law [22]. In GPM an enacted process is represented as a sequence of state transitions in the process domain. Transitions result either from transformations within the domain (reflecting its dynamics) or from actions of the environment on the domain (external events). A process ends when the domain reaches a desired ("goal") state where no more changes can occur due to domain dynamics.

The GPM framework has served as a basis for a variety of process analysis tasks, such as assessing process validity, analyzing it in terms of goals and soft-goals, scoping process change, analyzing process robustness to data deficiencies, analyzing inter-organizational processes, learning process model, and more [21].

2.2 Service-oriented modeling and architecture

Service-Oriented Architecture

Service-Oriented Architecture *(SOA)* is an architectural approach to viewing and creating a business solution as a network of modular components, each component implementing a discrete business function [25]. These components are called *services* and can be distributed across geography, enterprises, and disparate IT systems and can be reconfigured into new business processes as needed. These services are built on open standards and loosely coupled, allowing them to be easily combined both within and across enterprises to create new business processes.

2.2.1 Services Characteristics

According to [25] The following are key characteristics of services:

- **Coarse-grained and business-oriented:** Because there is always some latency due to network communication, a service should hide a substantial body of the application logic, giving significant business value to the consumer in one call.
- **Interface-based design:** A service implements a defined set of interfaces.
- **Invoked and discoverable:** A service and providers of the service can be discovered by other systems and invoked by those other systems.
- **Distributed:** Services composing an application can be distributed over different systems within an organization or even distributed across organizations.
- **Loosely coupled:** Services are connected through loosely coupled interfaces, modifying an interface possible without breaking the other ones.

Service-Oriented Architecture: A conceptual model

This concept is based on an architectural style that defines an interaction model between three primary parties: the service provider, who publishes a service description and provides the implementation for the service, a service consumer, who can either use the uniform resource identifier (URI) for the service description directly or can find the service description in a service registry and bind and invoke the service [26]. The service broker provides and maintains the service registry.

2.2.2 The architectural style and principles

The architecture style-defining SOA describes a set of patterns and guidelines for creating loosely coupled, business-aligned services that, because of the separation of concerns between description, implementation, and binding, provide unprecedented flexibility in responsiveness to new business threats and opportunities.

SOA is an enterprise-scale IT architecture for linking resources on-demand [25]. In SOA, resources are made available to participants in a value net, enterprise, and line of business (typically spanning multiple applications within an enterprise or across multiple enterprises). It consists of a set of business-aligned IT services that collectively fulfill organization business processes and goals. You can choreograph these services into composite applications and invoke them through standard protocols.

A service is a software resource (discoverable) with an externalized service description [25]. This service description is available for searching, binding, and invocation by a service consumer. The service provider realizes the service description implementation and also delivers the quality of service requirements to the service consumer. Services should ideally be governed by declarative policies and thus support a dynamically re-configurable architectural style.

Business agility is gained by IT systems that are flexible, primarily by separation of interface, implementation, and binding (protocols) offered by SOA, allowing the deferral of the choice of which service provider to opt for at a given point in time based on new business requirements, functional and non-functional (for example, performance, security, scalability, and so forth requirements).

You can reuse the services across internal business units or the value chains among business partners in a fractal realization pattern. Fractal realization refers to the ability of an architectural style to apply its patterns and the roles associated with the participants in its interaction model in a composite manner [23]. You can apply it to one tier in architecture and multiple tiers across the enterprise architecture. Among projects, it can be between business units and business partners within a value chain in a uniform and conceptually scalable manner.

2.2.3 The Service-Oriented Architecture Paradigm

Modern IT infrastructures are inherently heterogeneous across application infrastructures [25]. The push toward business automation and the requirement for more reliable executions has generated the need for integrating various available applications. However, enterprises are challenged with integrating silos of information and products from multiple vendors and across different platforms. This proves very difficult but enterprises cannot afford to simply take a single-vendor approach to IT, as application suites and supporting infrastructure would be very inflexible for organization-specific customizations.

Business organizations need to think about interoperability. They are under increasing pressure to respond to business changes with agility, align business processes with the current market, and respond quickly to competitive pressures. Globalization, for instance, is an important factor, as it leads to fierce competition.

To achieve these goals, organizations need to make their business processes automated, more open, and interoperable. SOA is a business-centric, IT architectural approach proposed to address the above goals, through designing business solutions as linked services based on open standards.

2.3 Distributed system

As mentioned earlier [3] a distributed system is a collection of autonomous computing elements that appears to its users as a single coherent system. A distributed system contains multiple nodes that are physically separate but linked together using the network. All the nodes in this system communicate with each other and handle processes in tandem. Each of these nodes contains a small part of the distributed operating system software. In our case, there are heterogeneous independent computers or stakeholders that are responsible for the clearance process and located in a different area by using a different system. There must be communication and coordination in order to appear as a single coherent system to the end-user.

According to [26] some advantages of Distributed Systems are as follows:

Horizontal Scalability—Since computing happens independently on each node, it is easy and generally inexpensive to add additional nodes and functionality as necessary.

Reliability—Most distributed systems are fault-tolerant as they can be made up of hundreds of nodes that work together. The system generally doesn't experience any disruptions if a single machine fails.

Performance—Distributed systems are extremely efficient because workloads can be broken up and sent to multiple machine failure of one node does not lead to the failure of the entire distributed system. Other nodes can still communicate with each other.

A distributed database system exhibits two main characteristics [26]:

Characteristic 1: Collection of autonomous computing elements

A distributed system is a collection of computing elements each being able to behave independently of each other [26]. A computing element, which we will generally refer to as a node, can be either a hardware device or a software process. Modern distributed systems can, and often will consist of all kinds of nodes, ranging from very big high-performance computers to small plug computers or even smaller devices.

Characteristic 2: Single coherent system

A distributed system should appear as a single coherent system [3]. In some cases, researchers have even gone so far as to say that there should be a single-system view, meaning that end-users should not even notice that they are dealing with the fact that processes, data, and control are dispersed across a computer network. More specifically, in a single coherent system, the collection of nodes as a whole operates the same, no matter where, when, and how the interaction between a user and the system takes place.

In principle, even within a single system, they could range from high-performance mainframe computers to small devices in sensor networks. Likewise, we make no assumptions concerning the way that nodes are interconnected [28].

2.3.1 Architectures of a distributed system

The notion of an architectural style is important. It is formulated in terms of components, the way the components are connected to each other, the data exchanged between components, and finally how these elements are jointly configured into a system [39]. A component of a modular unit with well-defined required and provided interfaces that are replaceable within its environment. That a component can be replaced, and, in particular, while a system continues to operate, is important [39]. This is due to the fact that in many cases, it is not an option to shut down a system for maintenance. At best, only parts of it may be put temporarily out of order. Replacing a component can be done only if its interfaces remain untouched.

Using components and connectors, we can come to various configurations which, in turn, have been classified into architectural styles. Several styles have by now been identified, of which the most important ones for distribute systems are [39]:

a. Layered architectures
b. Object-based architectures
c. Resource-centered architectures
d. Event-based architectures

a. **Layered architectures**

The basic idea for the layered style is simple: components are organized in a layered fashion where a component at layer j can make a down call to a component at a lower-level layer i (with $i < j$) and generally expects a response. Only in exceptional cases will an up call be made to a higher-level component [39].

A well-known example of this is the OSI model that incorporates a layered architecture when interacting with each of the components. Each interaction is sequential where a layer will contact the adjacent layer and this process continues until the request is been catered to. But in certain cases, the implementation can be made so that some layers will be skipped, which is called cross-layer coordination.

The layers on the bottom provide a service to the layers on the top. The request flows from top to bottom, whereas the response is sent from bottom to top. The advantage of using this approach is that, the calls always follow a predefined path, and that each layer can be easily replaced or modified without affecting the entire architecture.

b. **Object-based and service-oriented architectures**

In essence, each object corresponds to what we have defined as a component, and these components are connected through a procedure call mechanism. In the case of distributed systems, a procedure call can also take place over a network, that is, the calling object need not be executed on the same machine as the called object [39].

Object-based architectures are attractive because they provide a natural way of encapsulating data (called an object's state) and the operations that can be performed on that data (which are referred to as an object's methods) into a single entity. The interface offered by an object conceals implementation details, essentially meaning, in principle, can consider an object completely independent of its environment. As with components, this also means that if the interface is clearly

defined and left otherwise untouched, an object should be replaceable with one having exactly the same interface.

c. **Resource-based architectures**

As an increasing number of services became available over the Web and the development of distributed systems through service composition became more important, researchers started to rethink the architecture of mostly Web-based distributed systems. One of the problems with service composition is that connecting various components can easily turn into an integration nightmare.

As an alternative, one can also view a distributed system as a huge collection of resources that are individually managed by components. Resources may be added or removed by (remote) applications, and likewise can be retrieved or modified. This approach has now been widely adopted for the Web and is known as Representational State Transfer (REST) [39].

. There are four key characteristics of what is known as a REST full architectures**:**

- ✓ Resources are identified through a single naming scheme
- ✓ All services offer the same interface, consisting of at most four operations
- ✓ Messages sent to or from service are fully self-described
- ✓ After executing an operation at a service, that component forgets everything about the caller.

d. Event-Based Architecture

The entire communication in this kind of a system happens through events. When an event is generated, it will be sent to the bus system. With this, everyone else will be notified telling that such an event has occurred. So, if anyone is interested, that node can pull the event from the bus and use it. Sometimes these events could be data or even URLs to resources. So the receiver can access whatever the information is given in the event and process accordingly. processes communicate through the propagation of events.

These events occasionally carry data. An advantage in this architectural style is that components are loosely coupled. So it is easy to add, remove, and modify components in the system. Some examples are the publisher-subscriber system, Enterprise Services Bus (ESB). One major advantage is that these heterogeneous components can contact the bus, through any communication protocol. But ESB or a specific bus has the capability to handle any type of incoming request and process accordingly.

2.4 System architecture

Deciding on software components, their interaction, and their placement leads to an instance of software architecture, also known as a system architecture [39]. We use these two kinds of services in our day to day lives, but the difference between these two is often misinterpreted. The main architectural styles are centralized, decentralized, or hybrid [39].

A. Centralized organizations

The client-server architecture has two major components. The client and the server. The Server is where all the processing, computing, and data handling is happening, whereas the Client is where the user can access the services and resources given by the Server (Remote Server). The clients can make requests from the Server, and the Server will respond accordingly. Generally, there is only one server that handles the remote side. But to be on the safe side, we do use multiple servers will load balancing techniques [39].

As one common design feature, the Client-Server architecture has a centralized security database. This database contains security details like credentials and access details. Users can't log in to a server, without the security credentials. So, it makes this architecture a bit more stable and secure than Peer to Peer. The stability comes where the security database can allow resource usage in a much more meaningful way. But on the other hand, the system might get low, as the server only can handle a limited amount of workload at a given time.

B. Decentralized organizations: peer-to-peer systems

Multi-tiered client-server architectures are a direct consequence of dividing distributed applications into a user interface, processing components, and data-management components [39]. In many business environments, distributed processing is equivalent to organizing a client-server application as a multi-tiered architecture. We refer to this type of distribution as vertical distribution. The characteristic feature of the vertical distribution is that it is achieved by placing logically different components on different machines. The term is related to the concept of vertical fragmentation as used in distributed relational databases, where it means that tables are split column-wise and subsequently distributed across multiple machines [39].

In this network, any new node has to first join the network. After joining in, they can either request a service or provide a service. The initiation phase of a node (Joining of a node), can vary according to the implementation of a network. There are two ways in how a new node can get to know, what other nodes are providing.

i. **Centralized Lookup Server** - The new node has to register with the centralized lookup server a mention the services it will be providing, on the network. So, whenever you want to have a service, you simply have to contact the centralized lookup server and it will direct you to the relevant service provider.

ii. **Decentralized System** - A node desiring for specific services must broadcast and ask every other node in the network so that whoever is providing the service will respond.

C. Hybrid Architectures

So far, we have focused on client-server architectures and several peer-to-peer's architectures. Many distributed systems combine architectural features, as we already came across in super-peer networks. In hybrid architectures at some specific classes of distributed systems in which client-server solutions are combined with decentralized architectures.

2.5 Related work

This section briefly discusses major relevant works related to our proposed work but the idea is unique and new we cannot get enough thesis with related to our work.

2.5.1 Generic system related to a clearance management

In the higher institution of learning, final year students that have satisfied the academic requirements to graduate must undergo a clearance process before they disengage from the institution as define by Agbo [1]. This paper aims to design and develop an online student's clearance system following a traditional System Development Life Cycle (SDLC) and using PHP and MySQL The proposed approach can manage student's clearance process across all the departments and units and it eliminated the weakness of the manual process which mainly lack centralize repository for the clearance process.

The followings are the major drawbacks of this work.

- This research work has a lack of integrity in all office related to clearance.
- It is specific to only for graduation time it is limited to a specific period.
- The research does not support any delivery channel to communicate in the clearance process.
- Another problem is the system is designed only for students not for other entities like employee and item.…

Similarly, Umezinwa [13] used Structured System Analysis and Design Methodology (SSADM) and developed a web-based student clearance system using PHP, JAVASCRIPT, CSS, APACHE, and MySQL for the database. The system was able to process data with great speed and also replaces the error-prone manual clearance system. The limitation of this work is it work for only one entity.

Osman [33] a project on the development of an online clearance system using ASP.Net to create interfaces of the system. The system which was implemented at Quest University contains a database that can store all the required information of students for a clearance certificate via web pages. And another researcher Osman "*Design Online Clearance System*" for Tertiary Institutions [8] designed software serves as a more reliable and effective means of undertaking students clearance, remove all forms of delay and stress as well as enable to understand the procedure involved, as well as how to do clearance online. This project makes use of data collection from the university, materials, and journals from various authors, and the software was developed to effectively achieve the aim of the project. In this project, the implementation was carried out using PHP language, Html, CSS, and MySQL.

The main drawback of all the above studies are.

- The first one is dependence it focuses only on student clearance it must be flexible with other entities like an employee.
- The system must work at any time when the clearance is necessary
- It only works with a specific time (graduate time).
- Their main recommendation and future work are maintaining a central database for accessing all information relating to students. And integrate different office related to the clearance system
- Also, the developed approach did not consider different delivery channels such as email, the level of integration is not clear.

In conclusion, our work is fundamentally different from the other works since we have proposed a new generic clearance which is the capability of accepting any type of clearance entity like student, employee and item for any organization which has many features of the delivery channel and it works for any time based on the system setup and understandable for users.

Summarization of all paper and their gap related to our generic clearance management system

Research gaps	Deficiencies	References
The system is designed only for students not for other entities like employee and item	It has a lack of integration in all resources related to clearance. It is specific to only for graduation time it is limited to a specific period.	Agbo-Ajala, O. E Makinde. *"A Web-Based Database-Driven Students' Clearance System"*. Department of Physical Sciences, Ajayi Crowther University Oyo, Oyo State Nigeria. 2013 p.60.
It is not address all paper objectives	Some of the objectives of this paper were not actualized The research does not support any delivery channel to communicate in the clearance process.	Umezinwa Chukwuebuka, "*Design and Implementation of Online Clearance System"*: A Case Study of Imo State University European Journal of Applied Sciences, 2001, PP. 25-31.
The paper not cover all responsible resources and have no central database	The paper covers only clearance for various departments by the graduating student	Osman, O.L., **"***Design and Development of Online Clearance System* for Tertiary Institutions:" A Case Study of Tai Solarin University of Education... March 2016, page 655-664.
The effectiveness and efficiency of this new system provide room for further improvement	It is not Maintaining a central database for accessing information relating to student. Automation of student academic record to enable the management to have access to student academic performance.	Awuzie, S.C. "*Design and implementation of online Clearance system"*: a case study of Caritas University, 2003, PP. 21-32.

Table 1.1 summery of research gaps

Chapter 3- Design and Implementation of a generic clearance management system

3.1. Overview

In this chapter, the design and implementation of the proposed generic clearance system are presented in detail. Before designing and implementing the generic clearance system, the researchers conduct a high-level requirement definition and identified design goals

The following are the list of requirements identified from the group discussion along with different experts working in the design of generic clearance management system.

- The system shall be capable to clear any entity;
- It shall allow defining business process of any organization related to clearance;
- It shall be accessed by without device limitation;
- It shall be easy to integrate with existing systems.

The proposed system architecture is composed of two major modules: a setup and main module. Setup is to model generically the clearance stakeholders including organization, the entity to be cleared, resources associated with the entity and the business process to be defined to facilitate the clearance process.

The stakeholder server is a dedicated machine, which is designed to handle the clearance requests coming from the user. When users want to clear, communication between the server and the client is established through the network. Users (clearance requesters) can check the status of their request; and stakeholders involved in clearance can carry out their tasks from anywhere. The requesters can also get pdf certificates when the clearance is completed.

3.2 System Architecture

This section presents and illustrates the components of the generic clearance system as well as the interactions between the components. The proposed system operates on client-server based three-tier architecture and the communication between the client and the server takes place over a network.

The architecture of a system is one of the major issues that should be given paramount importance [39]. The architecture plays a pivotal role in modularizing the different components of a system. Moreover, the architecture of the system can be implemented in a tiered approach.

Designing the architecture in tiers is useful as it supports layering, promotes scalability, and allows easy maintenance of the application.

Considering this fact into account, this system is designed in a three-tier architecture. The architecture is composed of the User tier, Web application tier, and Data tier. The architecture of the system is depicted in Figure 3.1. The system is composed of different resources distributed in the three layers. The user layer contains all the client-side sub-systems and functionalities that contain the major resources: the user interface, the client-side security subsystem, and the setup sub-system.

The user interface is composed of the user application that permits the user to interact with the other system module. The user wants the clearance process on the client tier of the architectural model, which functions on a smartphone and web. the user registration records, the clearance records, and the transaction log are managed on the data tier. The user tier or user tier is the front-end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end-user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls. The presentation tier displays information related to such services as browsing all borrowed material, user clearance requests, and user clearance status. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

The web application tier is responsible for establishing a connection with the data tier as well as providing services to the user tier. The logic tier is pulled out from the presentation tier and, as its layer; controls an application's functionality by performing detailed processing. Logic tier is where mission-critical business problems are solved. The resources that make up this layer can exist on a server machine, to assist in resource sharing.

The application tier provides process management services (such as process development, process enactment, process monitoring, and process resourcing) that are shared by multiple applications. These application can be used to enforce business rules, such as business algorithms and data rules, which are designed to keep the data structures consistent.

Because these middle-tier components are not tied to a specific client, they can be used by all applications and can be moved to different locations, as response time and other rules require. For example, simple edits can be placed on the client-side to minimize network round-trips, or data rules can be placed in stored procedures.

The application tier server improves performance, flexibility, maintainability, reusability, and scalability by centralizing process logic. Centralized process logic makes administration and change management easier by localizing system functionality so that changes must only be written once and placed on the middle tier server to be available throughout the systems. it also manages distributed database integrity by the two phase commit process. It provides access to resources based on names instead of locations, and thereby improves scalability and flexibility as system components are added or move.

Data Tier is responsible for establishing a connection between the user tier and the data center. This tier consists of database servers. It can be accessed through the business services layer and on occasion by the user services layer. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its tier also improves scalability and performance.

Data needed by the application logic layer are retrieved from the database, then the computation results produced by the application logic layer are stored back in the database. Since data are one of the most complex aspects of many existing Information systems, it is essential in structuring the system. Both the facts and rules captured during data modeling and processing are important to ensure the data integrity.

This layer consists of data access to aid in resource sharing and to allow clients to be configured without installing the DBMS libraries and ODBC drivers on each client. An example would be a computer hosting a database management system (DBMS), such as a MySQL Server database.

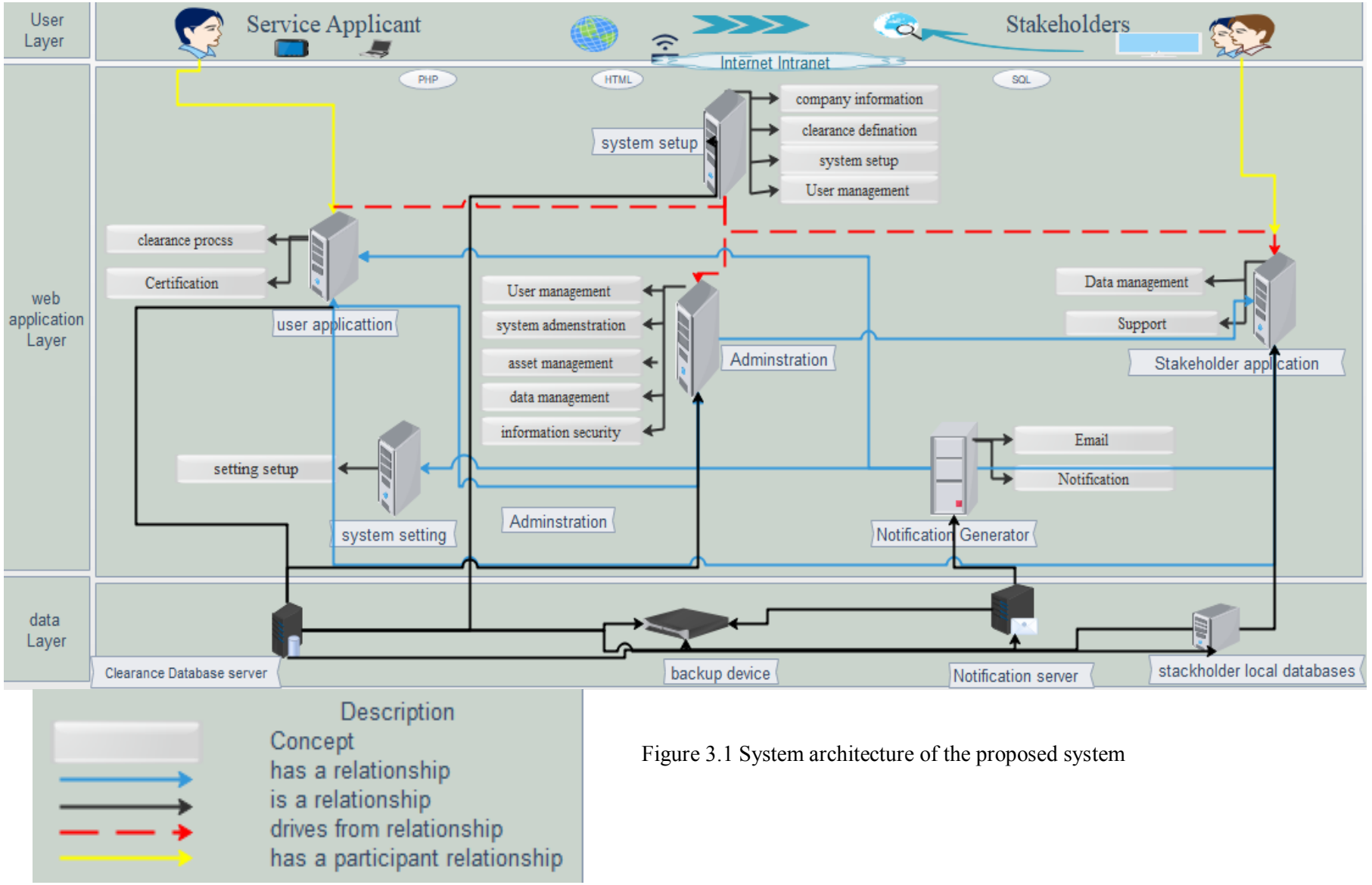

Figure 3.1 System architecture of the proposed system

3.3 Database Design

This section presents and shows logical and physical designs models of the proposed database system for the generic Clearance management system.

Overview

Our database was developed with the MySQL server management studio software, one of the requirements for our prototype was to make it generic, i.e. be able to store arbitrary data. This meant a little bit of struggling in terms of designing the database to meet the requirement. Several suggestions were presented and partly made before we looked into a database schema called EAV [37], short for the entity, attribute, value.

When new data types are introduced or existing types are modified in a conventional relational database system, the physical design of the database must be changed accordingly. For this reason, it is desirable that a clearance database be flexible and allow for modifications and for the addition of new types of data without having to change the physical database schema. The ideal clearance database would, therefore, implement a highly-detailed logical database schema in a completely-generic physical schema that stores the wide variety of data in a small and constant number of tables

3.3.1 EAV Model

In common sense, the Entity Attribute Value model is a data model to describe entities where the number of attributes (properties and parameters) that can be used to describe them is potentially vast, but the number of attributes that will actually apply to a given entity is relatively modest. In mathematics, this model is known as a sparse matrix [38]. EAV is also known as an object–attribute–value model, vertical database model, and open schema.

There are certain cases where an EAV schematic is an optimal approach to data modeling for a problem domain. However, in many cases where data can be modeled in statically relational terms, an EAV based approach is an anti-pattern that can lead to longer development times, poor use of database resources, and more complex queries when compared to a relationally-modeled data schema.

For EAV let's look into each part and try to understand them a little better.

- **Entity:** The entity represents any clearance event or items such as student, employee, customers, and orders. Each entity (student, employee, etc.) will have its own entity record in the database.
- **Attribute**: The attributes represent data items that belong to an entity. For example, the student entity has attributes such as name, age, department and many more
- **Value:** The value is the simplest to understand as it is simply a value linked to an attribute. For better understanding, let's consider the student entity. Each entity will have a series of attributes; one of them is the name attribute. Each employee will then have a value for the name attribute (and all other attributes) this can be seen in Figure 3.2.

3.3.2 Entity Table structure

This data representation is analogous to space-efficient methods of storing a sparse matrix, where only non-empty values are stored. In an EAV data model, each attribute-value pair is a fact describing an entity, and a row in an EAV table stores a single fact. EAV tables are often described as "long and skinny": "long" refers to the number of rows, "skinny" refers to the few columns.

(entity) student
- (attribute) name
    * (value) Ermias
- (attribute) age
    * (value) 27

Figure 3.2 Illustration of the EAV structure

The EAV model is mapped into a table having three columns:

- **The entity**: The item to be cleared.
- **The attribute or parameter**: A foreign key into a table of attribute definitions or metadata table for describing the datatype detail. At the very least, the attribute definitions table would contain the following columns: an attribute ID, attribute name, description, data type, and columns assisting input validation, e.g., maximum string length and regular expression, set of permissible values, etc.
- **The value of the attribute**.

In conventional database design, each parameter of interest is represented in a separate column in a table. As new kinds of data need to be managed, the number of columns and/or tables needs to grow. To add a new attribute for clearance description (e.g., phone number) to conventional relational database design (Table 3.1), another column has to be added to the table.

userid	Name	Date of Birth
1	Ermias Getachew	1984-Aug-14
2	Selam Tesfaye	1982-Aug-14

Table 3.2 Conventional relational database design

In EAV design, however, data may be stored in a single table with (conceptually) 3 columns: the first for column for entity identification, the second for the attribute, and the thired for the value of the attribute (Table 3.2).

Userid	Attribute	Value
1	Name	Ermias Getachew
1	Date of Birth	1984-Aug-14
2	Name	Selam Tesfaye
2	Date of Birth	1982-Aug-14

Table 3.3 EAV (Entity-Attribute-Value) database design

EAV (Entity-Attribute-Value) database design

To add a phone number attribute in the EAV table (Table 3.2), all that is required is to define a new code for the phone number to be stored in the attribute column. No change to the table schema is needed. Theoretically, most of the facts that are stored in a database can be stored in a single EAV table.

3.3.3 Database design

During the designing of the database for our prototype, we apply the **EAV (Entity, Attribute, Value)** model. This pattern is also known under several alternative names including the 'object-attribute-value' model and 'open schema'., as seen in Figure 3.3, our database design uses an EAV table, to group entities and attributes together. The tables consist of columns as shown in the next E-R diagram:

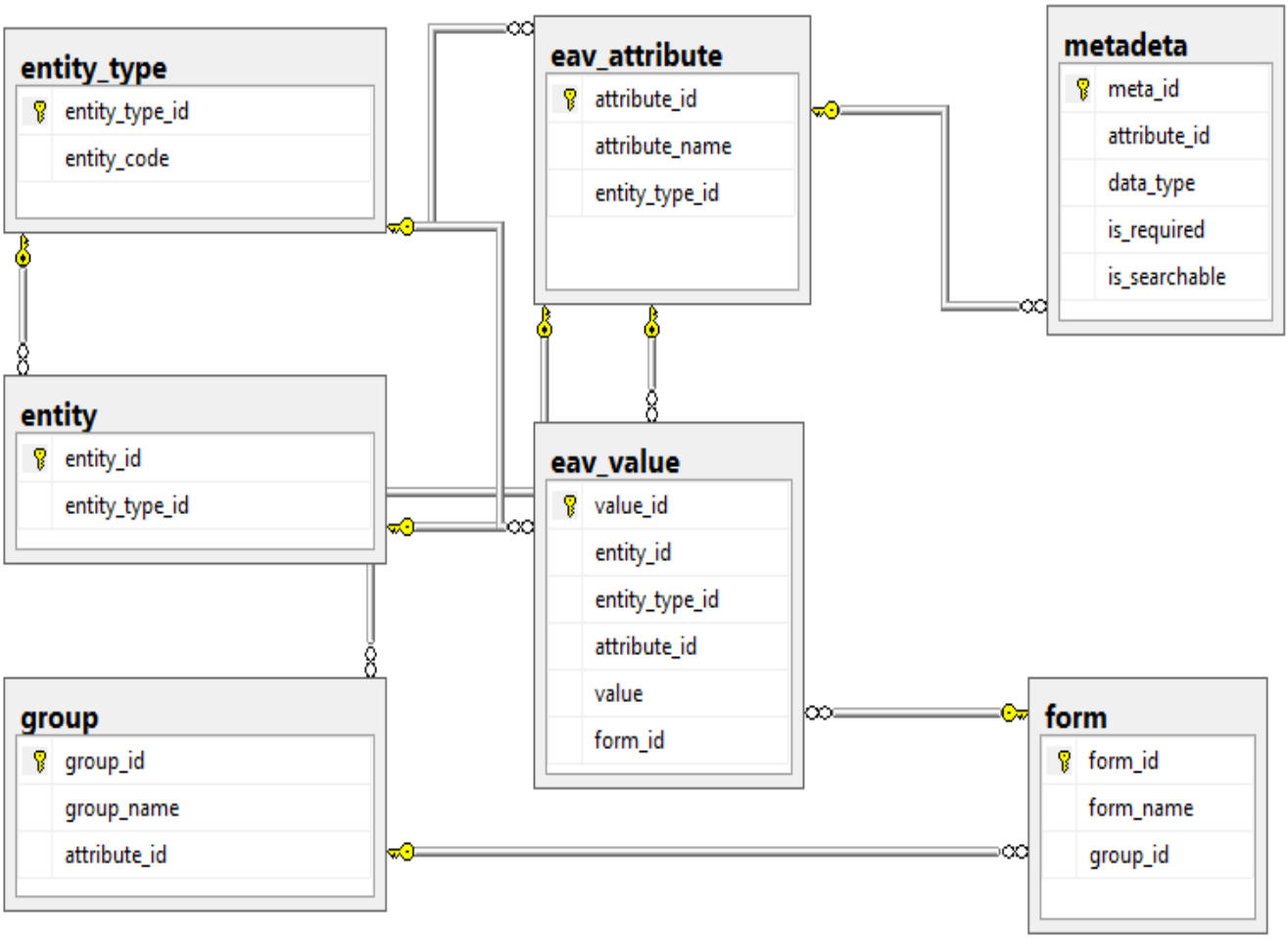

Figure 3.3 E-R diagram representing database design using EAV model

Description of all table

dbo.entity_type

- the name of entity type. Type varchar.
- stores the entity type including the information of model for the entity or the default attribute set.

dbo.entity

- The name of the entity. Type: Varchar
- It contains an eav_entity table storing objects of a certain entity type

dbo.attribute

- The name of the attribute. Type: Varchar
- Attributes are divided into groups (one group may have a lot of attributes and one attribute may be in a lot of groups). An attribute set includes the number of groups. An object has an attribute set.

dbo.value

- The value of the attribute. Type: string

dbo.metadata

- Name of the table where the value is stored. Type: Varchar
- This table describes which data tables certain types should reside in.

dbo.group

- the name of a set of attributes. Type: varchar

dbo.form

- the name of related groups. Type: varchar

Working with EAV Model

Entity-Attribute-Value (EAV) model is designed for the physical layout of data tables in a clearance database. In Entity-Attribute-Value design all data can be stored in a single table with conceptually 3 columns: the first for the entity (e.g., clearance type identification), second for the attribute (e.g., name), and third for value (e.g., "student"). To add more descriptive fields to the entity class, all that is necessary is to add attribute values to be stored in the attribute field.

EAV data storage

The EAV data storage in this mode seems quite complicated with the separated data for each store [42]. In databases, an entity is a single person, place, or thing (e.g., patient or diagnostic test) about which data can be stored. In conventional relational database design, each entity is mapped to one or more tables using values of one or more rows to uniquely identify each record.

This strategy works well for most databases even if the number of concepts involved in a domain may be high. As long as the domain of interest remains relatively unchanged, the table layout (i.e., the physical schema) should work well for many years. In a conventional database (that is, in a conventional relational database), new tables must be created to record new concepts. To give users access to the new tables, new forms must be designed and links to these forms must be provided in the user interface. If a table that is already in the database needs to be modified care must be taken not to destroy existing data and not to break any constraints. Accordingly, user-interface forms must be redesigned to reflect changes (e.g., fields that have been added or removed) in existing tables.

That means that for each entity there exists at least one table. there are some data storage tables:

EAV entity type: stores the entity type including the information of model for the entity or the default attribute set.

EAV entity: Contains an EAV entity table storing objects of a certain entity type. An entity can be any item; so far we have seen examples where an entity was an event of clearance, a student, or an employee or item. Entities in EAV are managed via an Object table that captures common facts about each item, such as name, description, and so on. The Objects table must have a unique identifier for each entity. The identifier is then used across the database as a foreign key. Using EAV modeling does not stop us from using conventional tables to capture additional details for individual objects. It is common to employ traditional relational database modeling and EAV modeling approaches within the same database schema.

EAV entity attribute: Attributes are stored in a dedicated attributes table. The primary key of this table is used as a reference across the database. Attributes are divided into groups (one group may have a lot of attributes and one attribute may be in a lot of groups). An attribute set includes the number of groups. An object has an attribute set.

EAV entity value: The simplest solution to represent data in an EAV model is to store it as a string. Values that are empty or do not apply to this entity are not stored in the EAV model. In

some circumstances, however, there is a need to record reasons for missing values. In such cases, the solution is to add a missing value code column to a table, which is non-null only when the value column is null. This code is then used to look up a list of textual explanations.

Grouping related attributes for display purposes may be accomplished in several ways. One or more descriptive columns may be added to the "entity part" of the Data table, or the metadata schema may be enhanced. A group table and a form table have been added to the metadata schema. Attributes may now be grouped and attribute groups may be part of forms. To the entity part of the Data table a new field, form ID, has been added telling the application to which form a data record belongs. Now any event recorded in the Data table belongs to a form and then maybe displayed together with all the other attributes on that form. Furthermore, this design facilitates the reuse of attribute groups in different forms. Depending on the domain being modeled and the requirements of the users, other metadata schemas may be suitable.

Metadata

EAV design is a way of simplifying the physical schema of the database, making it domain-independent [45]. Regardless of the physical schema, the user naturally perceives the data as conventionally structured in tables and columns. The logical schema of the database reflects the user's perception of the data. In an EAV database, the logical schema differs greatly from the physical schema. In a conventional database, the two are similar.

Therefore, an EAV system must have some means of translating the physical schema into a logical schema that reflects the user's understanding of data. This is achieved through metadata (or dictionary) tables whose content defines the semantics of the domain being modeled.

The metadata model may be enhanced considerably by adding more descriptive attributes to the metadata table. These attributes may have several purposes definition of an attribute's data type, constraints, or display layout (text field, select box, etc.).

EAV data access process

Exporting EAV data requires querying continuously from many tables, so the model carries out mapping data on many database tables.

```
1. Input: data from entity table
2. output: To read data from database to object
3. Start
4. Read data from main table or entity table
5. Define set of object attributes
6. Read out values of attributes for objects
7. Change values of attributes
8. Map data in object
9. End
```

Algorithm 3.1 read data from database to entity

```
1. Input: data from entity table
2. output: Burning EAV data to database
3. Start
4. Get data mapped in the objects
5. Change values of attributes
6. write data in entity table
7. write values in attributes table
8. End
```

Algorithm 3.2 Burning EAV data to database

According to [37] The EAV design has several advantages:

Flexibility: There are no limits to the number of attributes per entity. The logical database schema can grow without affecting the physical schema.

Storage: In a clearance database thousands of parameters are available while only a few may be recorded for each user. In a conventional design, this may lead to empty (NULL) fields. The EAV design does not need to reserve space for attributes with NULL values.

Efficient entity-centered queries: If, for example, all information for a single user is needed, it is necessary to query all data tables looking for information about this user. In a conventional database, this may be a time-consuming task that requires looking through hundreds of tables each of which may or may not have information for this user. As the number of tables and columns grow, the query must be reprogrammed. In an EAV database, only one table needs to be queried, no joins are necessary, and no change of code is required as the domain evolves.

A simple physical data format with partially self-describing data. This is important for cookies and registries. For example, Windows developers are encouraged to use the registry instead of proprietary formats to store program settings.

EAV design is potentially attractive for databases with complex and constantly changing schemas that reflect rapidly evolving knowledge in scientific domains. Here, when a conventional design is used, the tables (and the code that manipulates them, plus the user interface) need a continuous redesign with each schema revision. EAV design, by simplifying the schema, may provide relative insulation against such consequences of change. Along with simplification comes the potential for domain independence. With proper design, none of the tables in the system (and only a modest proportion of the code) are domain-specific. This architecture should therefore be portable across scientific domains.

Creates opportunities for re-use: Using the EAV model can create opportunities for reusing code. All those user-defined attributes are maintained by the same code.

Can make for very elegant and compact code: Well thought-out EAV code can be compact and elegant.

Can make the data very compact: Using an EAV model can reduce the number of tables you need and the space the data takes up

3.4 The Proposed system for generic clearance management

In this section we will discuss about the main modules related to our generic clearance management system. The setup module has the configuration part and the admin first customize this module according to the business rule and regulation of the organization. The next one is the clearance process module it manages and access each activity and event start from request a clearance to getting the certificate.

3.4.1 Setup module

The setup module the clearance stakeholders and associated business processes. Before starting the system, the admin must configure the setup based on the organization requirement and procedure. The first step is defining the company profile and the next step is creating clearance definition and stakeholder creation. After modeling the process and association the system automatically generates the new system based on the configuration.

And it has 6 steps as shown below

Step 1: create or register an organization/ company**.**

Step 2: create an entity and define user such as student, employee, item…. And they attribute to describe the clearance.

Step 3: create a resources associated with the entity and their attribute for all: After creating clearance we must define those responsible participant and register attribute.

Step 4: model their process: To make which one is the first step and which one is next to process the clearance request and the final place to clear.

Step 5: define their associated forms: In the main system form is generated based on the configuration so all attribute makes a group and those group make a form.

Step 6: finalized by creating users: The final phase is to create password user for all role types to authenticate and to use the main page.

After configuring the setup, it creates the main clearance system page to process the user clearance.

3.4.2 proposed generic clearance management system

The proposed generic clearance management system is designed in a client-server architecture. It has three elements: the server, stakeholder or participant and the end-user (Mobile app or web app)

The server is responsible for delivering the relevant data after collecting the necessary information. It is also responsible for processing and managing user clearance requests, and storing and handling user's and clearance context data. Furthermore, it is also responsible to maintain a central repository for all transactions related to clearance activity. The server delivers the relevant information to the user after analyzing the request. In addition to this, the server determines the status of the user clearance request.

The participants are responsible to provide specific data based on the role of the office. This data includes registers' new borrowed items, insert a new item, modify an item, store user information when something is borrowed, etc. These collected data will be sent to the server to be stored in the central repository for later retrieval and reasoning to deliver the relevant information to the user according to the different contexts.

End user is responsible for the interaction of the user with the server to get (access) the proposed service in either pull or push mode. That is, it allows a user to query or receive personalized clearance information from the server. The user data is acquired from the user's mobile phone explicitly through the provided user interface and implicitly through accessing the properties (features) of the mobile phones (devices). The data that is collected explicitly from the user's mobile phone includes his preferences (or request) for the type of service during the pull scenario, and his profile data (such as identity, name, address, phone No., e-mail address). The user explicitly provides these data on the provided interface by any device.

All communications between the client and the server take place through the currently existing cellular phone infrastructure over the network. To handle the communication between these three major elements, we have developed a module that manages the interaction. The two clients communicate with the server via TCP/IP sockets through a network gateway.

The general overview of the architecture of the proposed system is depicted in Figure 3.4. It describes the three main elements in some detail and the overall interaction between them. Furthermore, it is also shown how to interact with the external services like the notification.

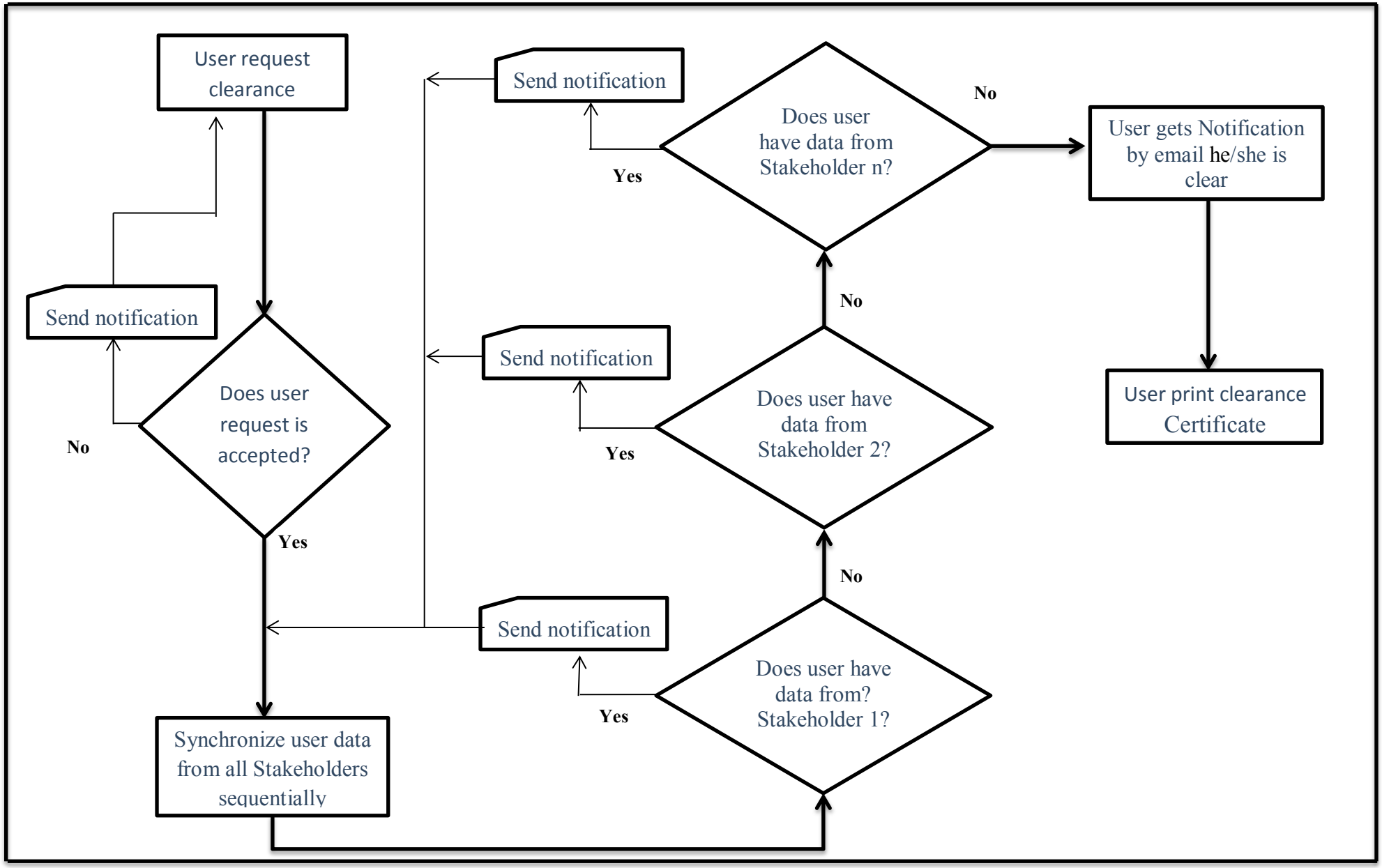

Figure 3.4 The proposed generic clearance management system flowchart

Chapter 4- Experimentation and Evaluation

This chapter briefly presents each of the proposed system components implemented prototype. This Chapter also briefly lists and explains the tools and technologies that have been used to implement the proposed system prototype and why they were selected. Moreover, experimental procedures, requirements of testing environment, and performance of the proposed system from different perspectives are also presented and discussed in detail.

4.1 Tools and Technologies used for Development

Appropriate and advanced technologies are used to develop a clearance management system. The followings are major technologies with their brief and precise description used for development.

Java Programming Language

Java programming language has been used to write computer instructions for the client-side and server-side applications [46]. Java is one of the most popular programming languages used to create web applications and platforms. It was designed for flexibility, allowing developers to write code that would run on any machine, regardless of architecture or platform. According to the Java home page, more than one billion computers and three billion mobile phones worldwide run Java. So java is everywhere, that is, it's on desktop, mobile and almost everywhere. Moreover, Java18 is object-oriented, multi-threaded, interpreted, distributed, dynamic, robust, secure, architecture-neutral, portable, and executes with high performance. That is why Java is a preferred programming language for implementing the proposed system.

Java Script Object Notation (JSON)

JSON33 is a lightweight data-interchange format that is easy for machines to parse and generate. It is a text format that is completely language independent. This property makes JSON an ideal data-interchange language.

MySQL

MySQL is an open-source relational database engine used as a back end to manage data for server-side development. MySQL 5.7 has been used as a server-side relational database engine in the proposed system.

4.2 Proposed system Components, Implementation Details, and Demonstration

This section presents the demonstration and implementation details of each component of the generic clearance management system.

4.2.1 Generic clearance management system user interface

When an admin first initiates GCMS by accessing the website on the web than the system displays the login window as shown in Figure 4.1. There is a default username and password for admin to configure the setup first before creating the main page after inserting the default login parameter the system displays the home page of admin which have the setup link and other default pages.

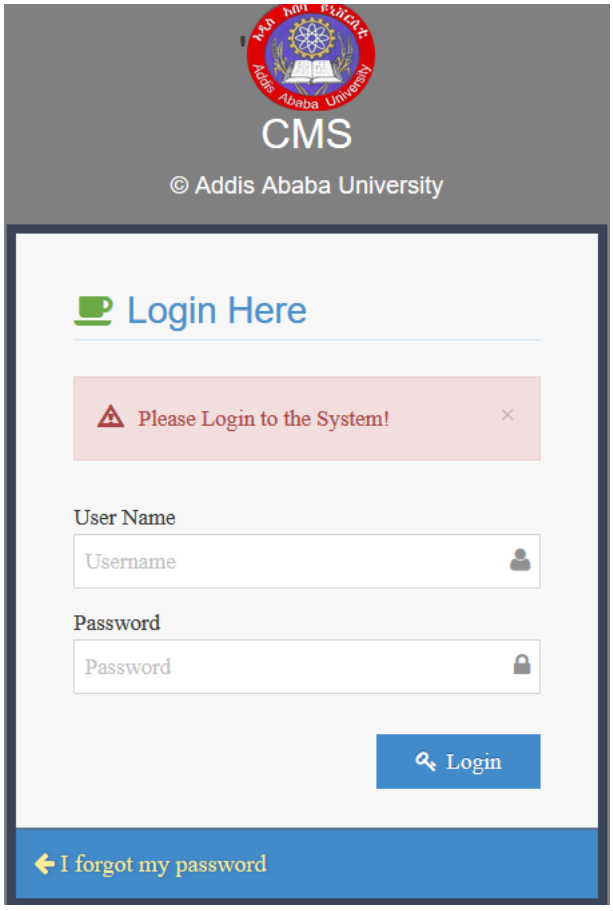

Figure 4.1 Snapshot of GCMS user login interface

4.2.2 GCMS company information and address wizard

To begin the setup, the admin must click on the setup button to get the wizard. The first wizard part is the company profile phase that helps the user to insert company profile and address and it is flexible the user can add any attribute which describes for his organization as shown in Figure 4.2.

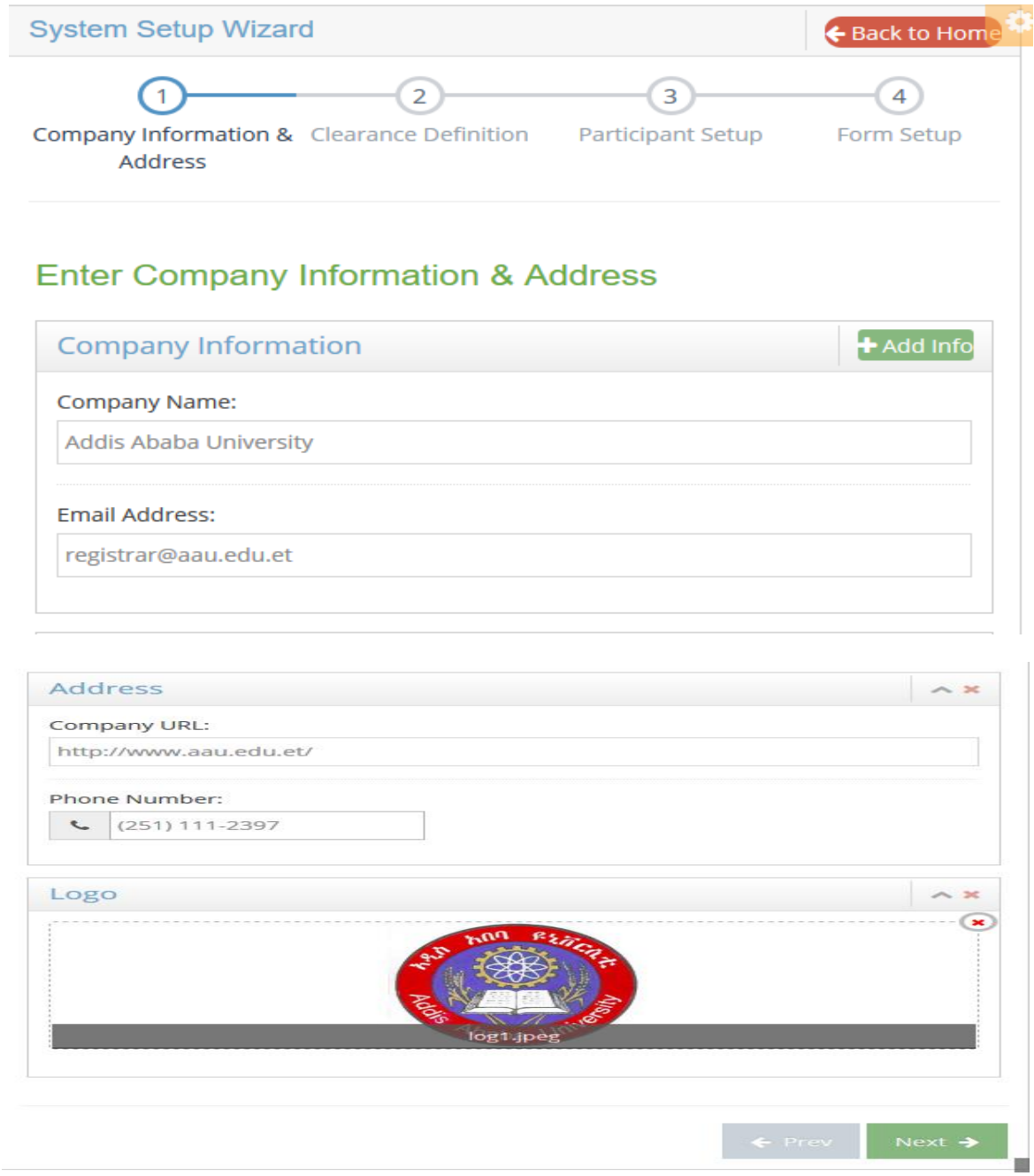

Figure 4.2 Snapshot of GCMS to add company information and address

4.2.3 GCMS clearance definition wizard

The next phase is to create a clearance entity which is the company wants to be cleared like student, employee, and item whatever the company wants and they can add one entity at a time as shown in Figure 4.3 and Figure 4.4.

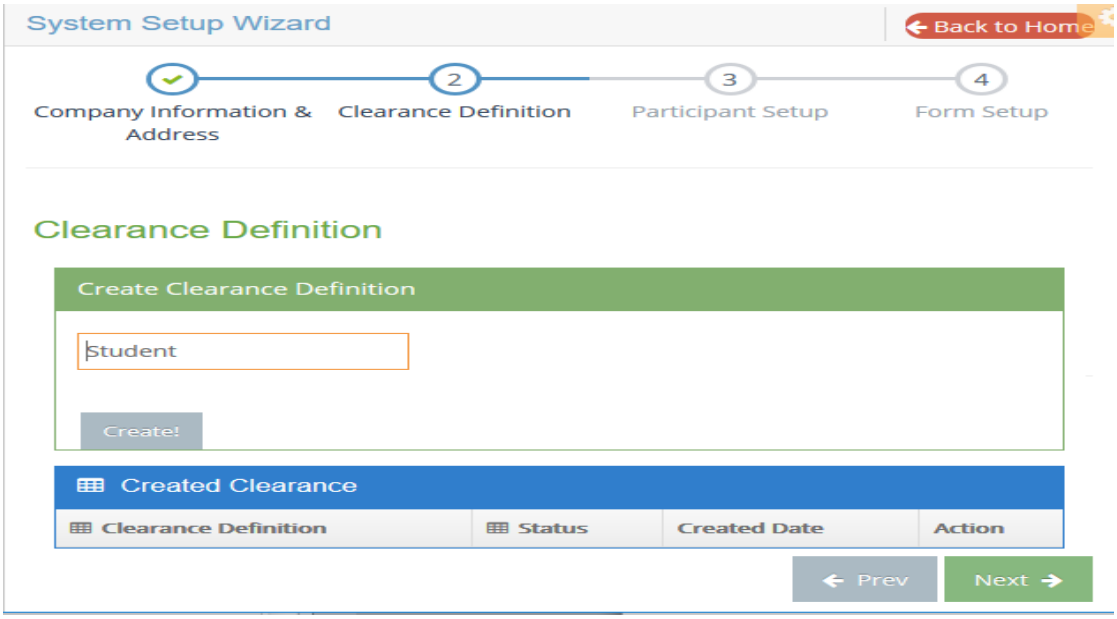

Figure 4.3 Snapshot of GCMS creating clearance definition

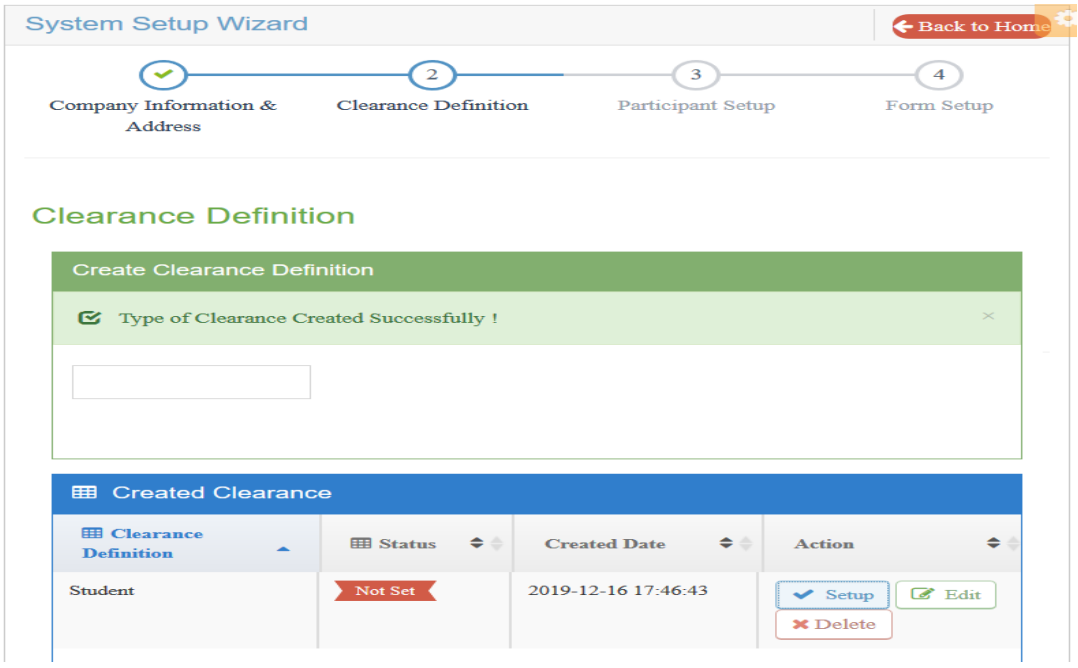

Figure 4.4 Snapshot of GCMS clearance definition setup

The entity must be described by associating attributes that describe the entity as user input having attribute name and validation of the attribute which required or not.

4.2.4 GCMS clearance participant setup wizard

After creating the entity, we need to register the main responsible office or participant who has relation with the entity to clear the clearance as shown in Figure 4.5.

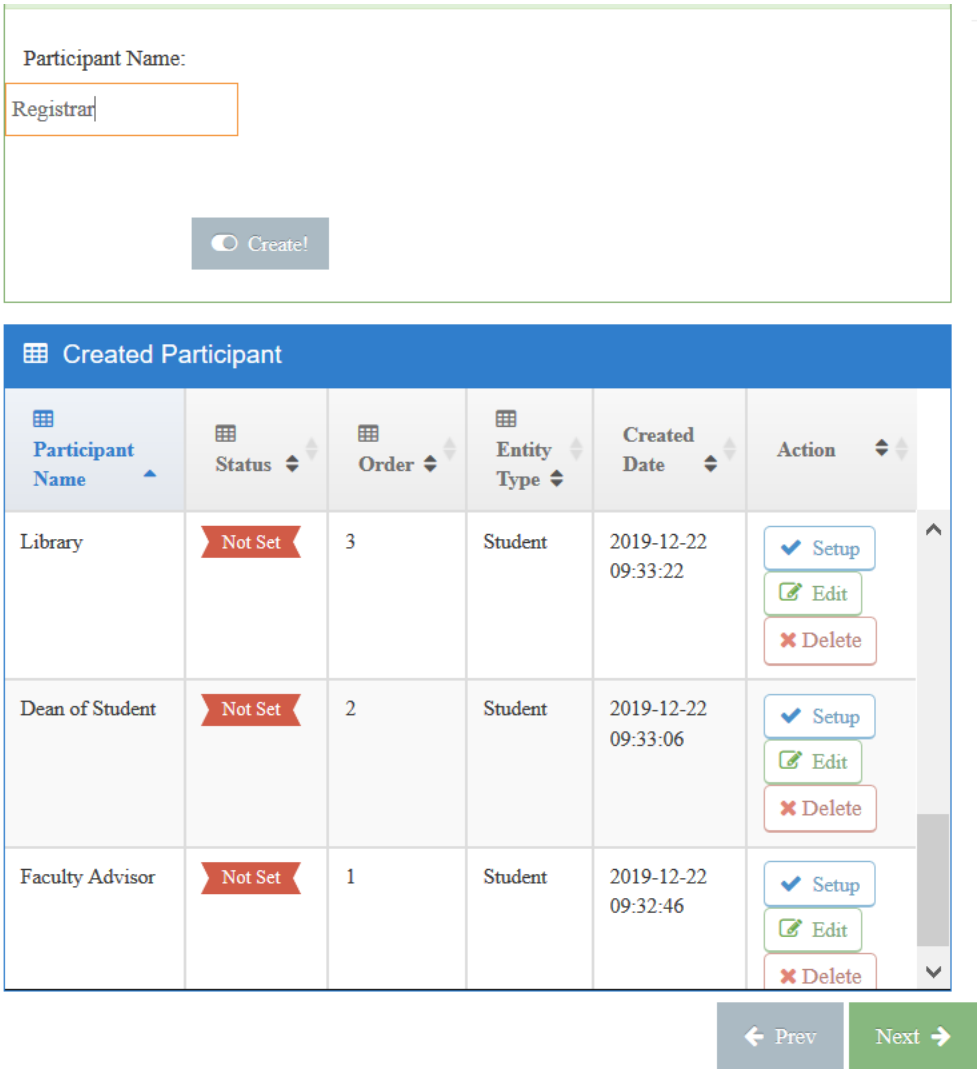

Figure 4.5 Snapshot of GCMS to create participant

The participant must describe by creating the attribute from user input with a column of attribute name and validation of the attribute which required or not to create the form for register the item to borrow them for the user as shown in Figure 4.6.

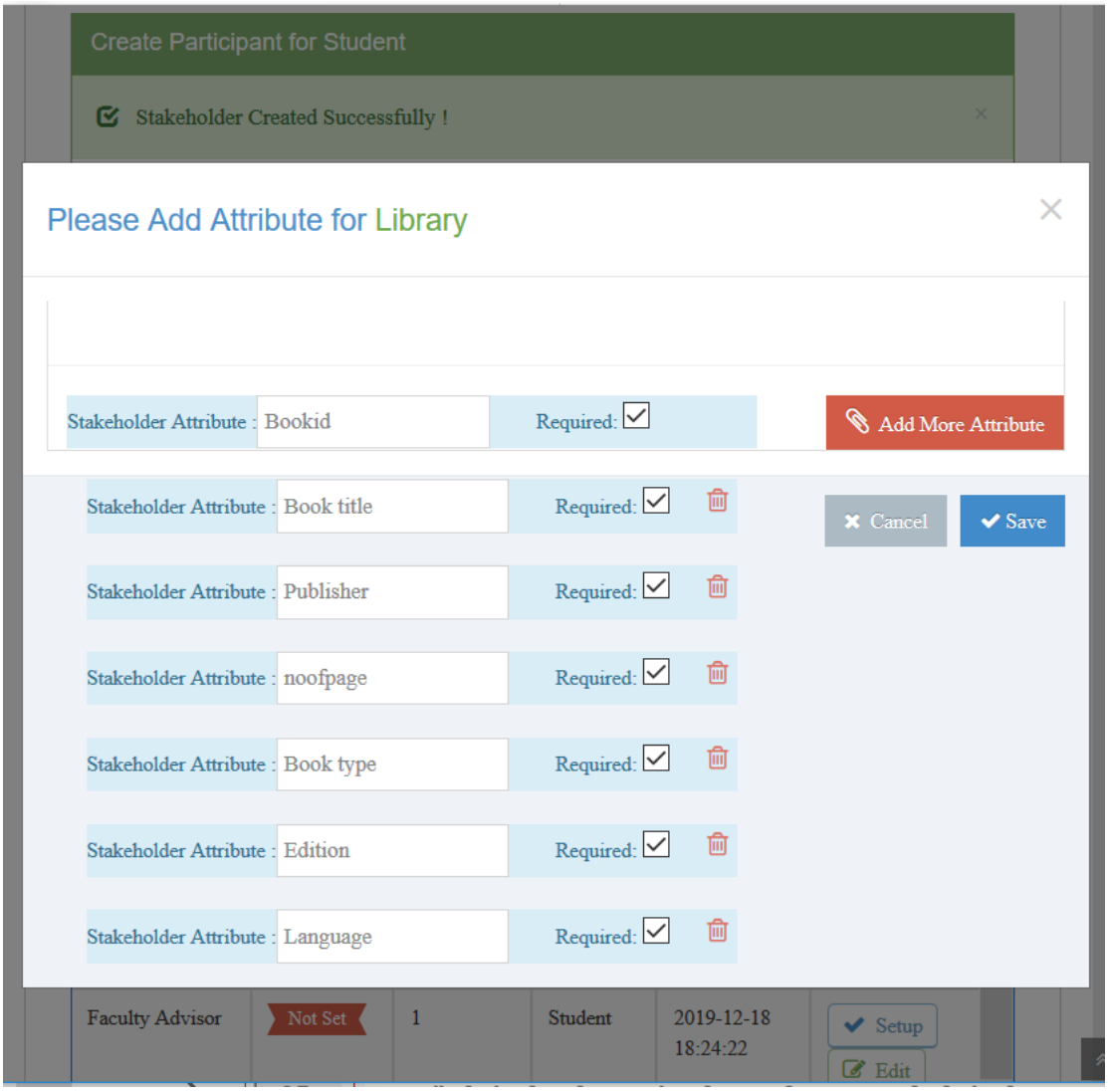

Figure 4.6 Snapshot of GCMS to add attribute for participant

4.2.5 GCMS clearance form setup wizard

The last step is to create a form for all participants to display on the user page by their role to record the item data and to make any transaction regardless of the participant as shown in Figure 4.7 and Figure 4.8.

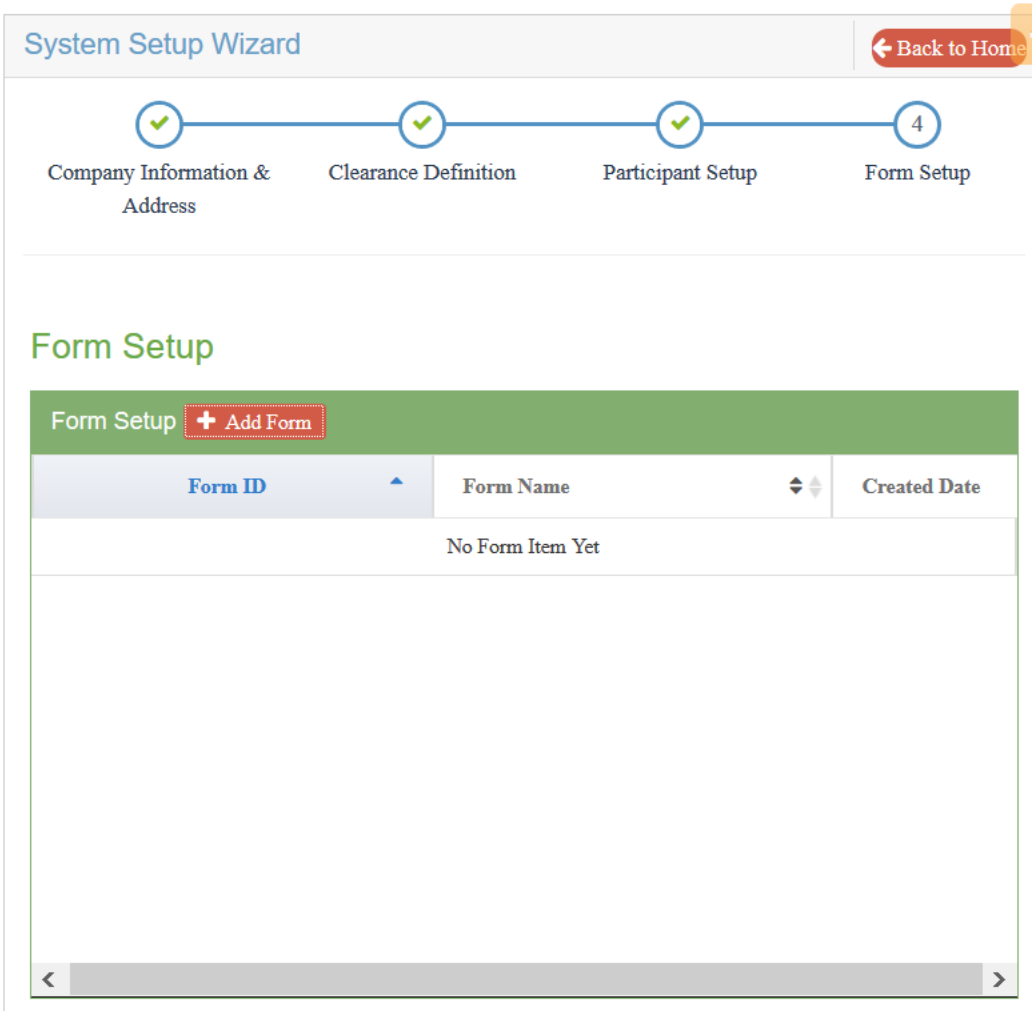

Figure 4.7 Snapshot of GCMS to add form

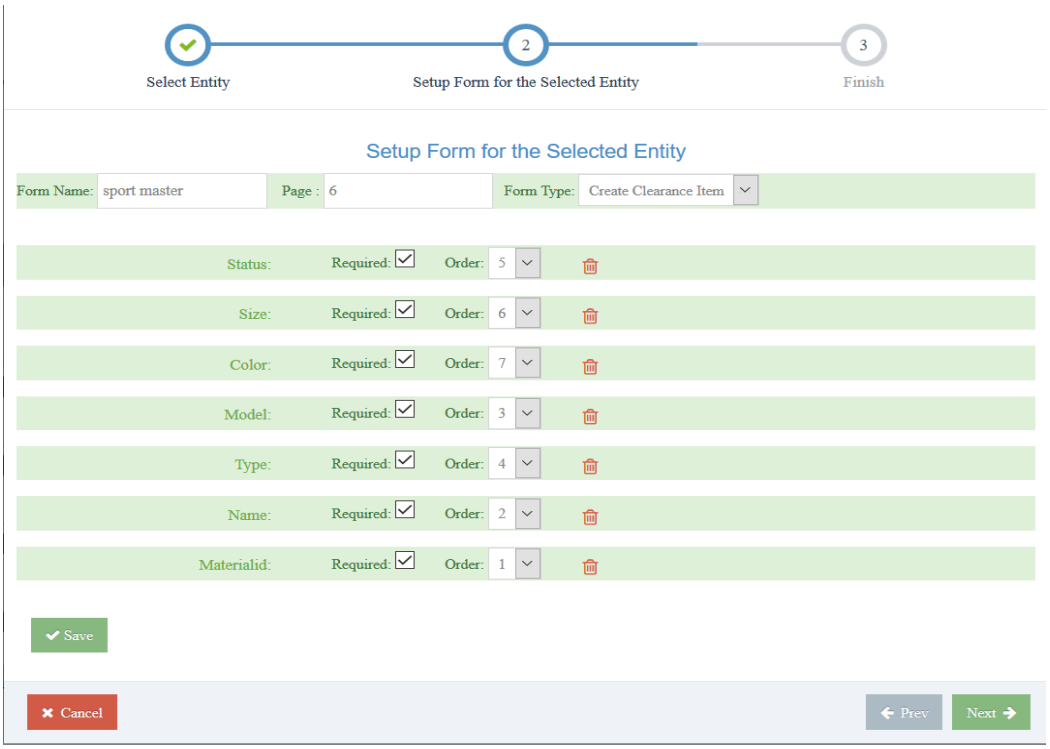

Figure 4.8 Snapshot of GCMS to view all created form

4.3 The Prototype Usability Testing

In this section, the usability of the generic clearance model from a different perspective is evaluated from users' perceptions. The model needs to be highly usable by a large cross-section of the population to be feasible. Usability testing refers to evaluating a product or service by testing it with representative users [49].

The goal is to identify any usability problems, collect qualitative and quantitative data, and determine the participant's satisfaction with the product. the prototype is developed for a student clearance process AAU and the total participants of 20.

·A mix of genders

· A mix of educational background

· A mix of the ages:

According to [50] usability metrics should include:

- **Simplicity:** A generic clearance model needs to be easy to use by a broader set of people of different ages and technical competency and literacy.
- **Accuracy:** The generic clearance model needs to be accurate enough for any kind of transaction with various amounts and types of clearance requests.
- **Security:** A generic clearance model needs to be secured for any transaction.
- **Speed:** The generic clearance model needs to be fast.

To evaluate users' perception about the simplicity, accuracy, security, and speed of clearance management system transactions, participants are selected purposely from clearance requester and the participant. As a result, selected participants were composed of current Students from AAU (7 participants), employees who are working on the clearance process from AAU (6 participants), other users currently working on computer and programming user (4 participants), and former student (3 participants) Each participant was asked to complete a short demographics survey to determine their technical competency. The demographic survey questions prepared and provided for participants are shown in Annex A - Questionnaire.

For the experiment, all categories of participants are ordered to make clearance requests, to borrow an item, to return an item, and to approve the request from the participant. For experimenting with clearance transactions, an experimenter played the role of the other person involved in the clearance transaction. As a result, all categories of participant's users performed all the transactions successfully.

After the completion of all experiments, perceptions are measured through a simple questionnaire provided in Annex A - Questionnaire considering four major criteria such as simplicity, accuracy, security, and speed of generic clearance management system of any transactions. Figure 4.9 shows the experiment results.

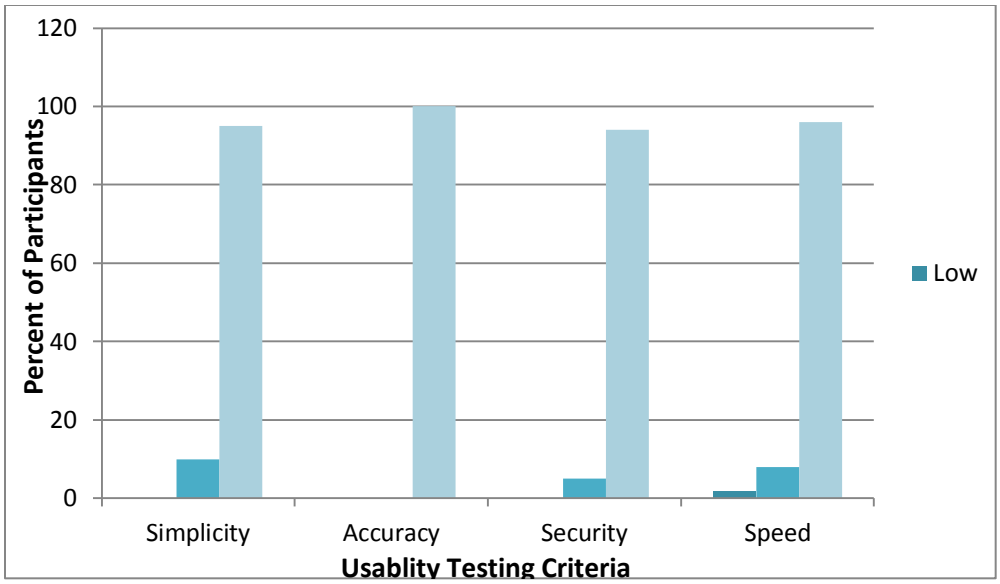

Figure 4.9 Usability Testing Experiment Result

According to the experiment result, 95% of participants have agreed that clearance system has higher simplicity; 98% of participants have agreed that clearance system has higher accuracy; 96% of participants have agreed that clearance system has higher security, and 96% of participants have agreed that clearance system has higher speed than the current approach. To sum up, approximately 96% of 20 participants have agreed that the clearance system has higher usability than the current approach.

4.4 Summary

In this chapter, appropriate tools and technologies used to develop the proposed system are briefly presented and described. Next to that, the demonstration and implementation details of major components of the proposed system are illustrated and presented.

As the prototype shows, an easy-to-use and interactive user interface are designed that can be easily learned and adapted by users. Several features are implemented in the system in order to make it usable. In the usability testing experiment result, 96% of 20 participants have agreed that the new clearance system has more usability than existing approaches. In the experiment, it is observed that all users can equally perform clearance system transactions easily in general, the new clearance system has achieved simplicity with accuracy and end-to-end security. Several features are implemented in the system in order to make it usable.

Chapter 5- Conclusion and Future Work

This chapter presents the summary of various works presented in this thesis and the vital contribution of the research work in a generic clearance management system that will apply for any organization and any entity to clear. Moreover, the future works of the research are presented in this chapter.

5.1 Conclusion

This work proposed to design and develop a generic clearance system that will apply for any organization and any entity to clear that will reduce delays in the existing process and create a central repository for entities to be cleared. The designed framework will serve as a more reliable and effective means of undertaking user clearance, remove all forms of delay and stress as well as enable you to understand the procedure involved, as well as how to do your clearance.

In this work, we have accomplished the following tasks:

The proposed system architecture is composed of two major components: setup module and the main system. The setup module part is composing of configuring the setup process to define the clearance, stakeholder, and their form. After finalizing the configuration part based on the user input the main system is generated. The components of the proposed system with their functionalities and interactions are identified and explained.

Finally, in the usability experiment, 96% of participants have agreed that a generic clearance management system has higher usability than the existing approaches. As a result, the clearance management system is properly modeled while preserving strong security.

5.2 Contributions

The followings are the main contribution of this thesis:

1. A novel generic system for a clearance management system is proposed.
2. Secured communication technique is employed.
3. The user to see clearance statues and to get clearance certificate online method is proposed
4. A new approach to clear the entity without appearing physically to the participant, office, or stakeholder unlike the existing clearance management systems is proposed.
5. A new approach of generic setup configuration which has the capability of configuring any entity clearance system for any organization is proposed.

5.3 Future Works

In the future, we plan to extend the existing work to address the following issues:

1. Delivery channels are some communication methods that every computer system should improve and achieve. Due to this fact, in the future, researchers would like to extend the research to incorporate and use others.
2. A better security mechanism towards communication between the participant and the entity can also be further studied.
3. Securing log data sources from intruders' corruption.

References

[1] Agbo-Ajala, O. E Makinde. *"A Web-Based Database-Driven Students' Clearance System"*. Department of Physical Sciences, Ajayi Crowther University Oyo, Oyo State Nigeria. 2013 p.60.

[2] Donald, L and Newland, Dan "*Online Clearance"*. (3rd Ed.). Pearson Publishing. & Glossary, 2000, p. 416.

[3] M. Tamer Özsu. Patrick Valduriez. "*Principles of Distributed Database Systems*". 2011, pp. 62-152.

[4] Gämmêy Shiferaw. "*cost sharing documentation on*". *HU Staff Clearance System Documentation* Jan 23, 2019. p 2.

[5] IEEE explorer. *"web service approach to secure data access"*. by john Haber department of information technology, Uganda Christian University, 2009 pp. 231-250.

[6] Awuzie, S.C. "*Design and implementation of online Clearance system"*: a case study of Caritas University, 2003, PP. 21-32.

[7] "Service-oriented modeling "internet: https://en.wikipedia.org/wiki/Service-oriented_modeling**,** 17 January 2020 [Accessed on Feb. 2020]**.**

[8] jay f. Jnamaker." *Systems Development in Information Systems Research*",1991, pp. 45-89.

[9] Salehi Nejad, Alireza." *Fundamentals of System Analysis & Design*". 2014, pp. 38-45.

[10] Charles S. Wasson." *System Analysis, Design, and Development*". John Wiley & Sons, Inc. 2006, pp. 265-289.

[11] Whitten, Bentley. "Systems Design" in *System Analysis and design methods*. McGraw-HilJ Companies., 2007, pp. 444-510.

[12] Alan d, Roberta m." architecture design "in *system analysis and design*. John Wiley & Sons, Inc. Fourth Edition, 269-292.

[13] Umezinwa Chukwuebuka, "*Design and Implementation of Online Clearance System"*: A Case Study of Imo State University European Journal of Applied Sciences, 2001, PP. 25-31.

[14] Goland, Y. Y. "*The Promise of Portable Business Processes*", Web Services Journal, 2005, pp.15-38.

[15] "*Software Process Framework*" internet: https://www.geeksforgeeks.org/software-engineering-software-process-framework/ [Accessed on Nov. 23 2020].

[16] Y. Zorian, E.J. Marinissen, S. Dey, "*Testing embedded-core based system chips*", *Proc. IEEE ITC, 1998*, pp. 130-143.

[17] Paspallis, N., Raspopoulos, "*Systems Development in Information Systems Research*", Cyprus, September 6 - 8, 2017. pp. 45=63.

[18] Pinaki Mazumde. "*Genetic algorithms for VLSI design layout & test automation*", 1999. pp. 233-290.

[19] Williams, S. "*Business Process Modeling Improves Administrative Control*," In *Automation*. Dec 1967, pp. 44 - 50.

[20] Yongchareon, *"A View Framework for Modeling and Change Validation of Artifact-Centric Inter-Organizational Business Processes"*. Information Systems. Sira (2015). pp. 72-80.

[21] JAY F. JNAMAKER." *Systems Development in Information Systems Research*".1991 pp. 121-200.

[22]. Soffer P. and Wand Y." Goal-driven Analysis of Process Model Validity". *Advanced Information Systems Engineering. (CAiSE'04)* (LNCS 3084), 2004, pp. 521-535

[23] INCOSE. '*Systems Engineering Handbook*: A Guide for System Life Cycle Processes and Activities', version 4.0. Hoboken, NJ, USA: John Wiley and Sons, Inc., ISBN: 978-1-118-99940-0. 2015. pp. 11-134.

[24] Claudius Ptolemaist "Heterogeneous Modeling *"in System Design, Modeling, and Simulation.* First Edition, Version 1.02 Ptolemy.org, 2014. Pp 13-20.

[25] Ali Arsanjani, "Service-Oriented Architecture: A conceptual model". *Service-oriented modeling and architecture.* IBM Global Services 09 Nov 2004 pp 2-15.

[26] M. Tamer Özsu. Patrick Valduriez "*Distributed Systems*" Third Edition 2017. Pp. 87-210.

[27] Arno Puder, Kay Römer." distributed system" in *distributed systems architecture a middleware approach*. Elsevier Inc. 2006, pp.7-14.

[28] Mohamed e "distributed database architecture" in *distributed databases.* 2012, pp. 9-15.

[29] M. Tamer Özsu "Distributed Database Design" *Principles of Distributed Database Systems* Fourth Edition Springer Media, LLC 2011, pp.13-28.

[30] Saeed k, frank s. "DBE Architectural Concepts" *distributed database management systems a practical approach.* John Wiley & Sons, *2010. 5-11.*

[31] Ross J. Anderson" Distributed Systems", *Security Engineering: A Guide to Building Dependable Distributed Systems.* Second edition, Wiley, 2008, pp.115-189.

[32] Arno Puder, Kay Römer, Frank Pilhofer." Basic Concepts" in *Distributed Systems Architecture*. 1st Edition, Wiley, 18th April 2011, pp 56-69.

[33] Osman, O.L., **"***Design and Development of Online Clearance System* for Tertiary Institutions:" A Case Study of Tai Solarin University of Education... March 2016, page 655-664.

[34] James Aspnes. *"*model *"in Notes on Theory of Distributed Systems.* Creative

Commons Attribution-Share, 2020-01-13, pp.113-118.

[35] George Coulouris, Jean Dollimore. "designing distributed systems: google case study" in *distributed systems concepts and design*, fifth edition, Pearson Education, Inc. 2012, pp.915-956.

[36] M. Bauer, N. Coburn. *A Distributed System Architecture for a Distributed Application Environment.* Article in IBM Systems Journal · January 1994, pp.401-422.

[37] Embedded Analytics Market Study. *3-Tier Architecture: A Complete Overview.* Internet: https://www.jinfonet.com/resources/bi-defined/3-tier-architecture-complete-overview/, [Accessed on Jan. 2020]

[38] Microsoft. *using a Three-Tier Architecture Model.* Internet https://docs.microsoft.com/en-us/windows/win32/cossdk/using-a-three-tier-architecture-model, [Accessed on Dec. 1 2020]

[39] Geeks for geeks "*Software Process Framework*" internet:

https://www.geeksforgeeks.org/software-engineering-software-process-framework/, [Accessed on Nov. 23 2020].

[37] Bruce "Entity Attribute Value in Magento" Internet:*https://blog.magestore.com/entity-attribute-value-in-magento/,* Sep. 6, 2017[Accessed on Dec. 21 2019].

[38] Robert Raszczynski. "Understanding the *EAV data model*" Internet: *https://inviqa.com/blog/understanding-eav-data-model-and-when-use-it,* Oct. 22, 2010[Accessed on Oct. 4, 2019].

[39] keet. Malin. "*Distributed System Architectures and Architectural Styles*". Internet:https://keetmalin.wixsite.com/keetmalin/post/2017/09/27/distributed-system-architectures-and-architectural-styles, Sep. 30, 2017 [Accessed on Jun. 29, 2019].

[40] Ajay D, Mukesh Singhal "A model of distributed computations" *distributed computing principles, algorithms, and systems,* Cambridge University Press, 2008, pp.39-48.

[41] Valentin Dinua, Prakash Nadkarni. *Guidelines for the Effective Use of Entity-Attribute-Value Modeling for Biomedical Databases.* Int J Med Inform. 2007, pp 4-15.

[42] Prakash m. Nadkarni, *Organization of Heterogeneous Scientific Data Using the EAV/CR Representation.* Journal of the American Medical Informatics Association, Volume 6, Number 6, Dec 1999, PP.478-492

[43] Roland s. Chen, Prakash Nadkarni. *exploring performance issues for a clinical database organized using an entity-attribute-value representation.,* journal of the American medical informatics association, volume 7, number 5, Oct. 2000. pp.475-486.

[44] Cynthia m. Brandt, *Automatic Metadata-driven Generation of Web Interfaces to Entity-Attribute-Value Databases.* Journal of the American Medical Informatics Association Volume 7 Number 4 Jul / Aug 2000, PP. 344-355.

[45] Shivani Batra, Shelly Sachdeva.” *Entity Attribute Value Style Modeling Approach for Archetype Based Data*”. 29 December 2017. 8-20.

[46] Kate Bruscke. Why Is Java Important? Internet: https://smallbusiness.chron.com/java-important-30466.html, [Accessed on Oct. 12 2019].

[46] Allen, R. "*Workflow: An Introduction*". Fischer L. (ed.), 2001, pp. 15-38.

[47] Christopher A. *“*model translation*” Simulation modeling and book a practical approach.* Chung CRC press, 2004, pp.231-311.

[48] Umuagwo Ohaji, *“Design and Implementation of Online Clearance System”*: A Case Study of Imo State University European Journal of Applied Sciences.2003, pp. 25-31.

[49] U.S. Department of Health & Human Services. **“***Usability Testing”. Internet:* https://www.usability.gov/how-to-and-tools/methods/usability-testing.html, Digital Communications Division, July 10, 2017 [Accessed on Aug. 23, 2019].

[50] Justin Mifsud.” usability metrics” *Usability Metrics – A Guide to Quantify the Usability of Any System*. 2011.

[51] Velte, Anthony T. “Cloud Computing: A Practical Approach.”. McGraw Hill. (2010). *pp.150-184.*

Annex A Questionnaires part 1

Information Sheet

Questionnaire No _____

Title of the Research: - To design and implement generic clearance management system for any organization to clear any entity at any time.

Name of Investigator: - Ermias Getachew

This study will be conducted by a postgraduate student at Addis Ababa University computer science programmer.

The aim of this study is to evaluate a system developed for clearance management. The system facilitates charging and discharging of clearance and keeping records of eligible users.

Benefits and Harms

There is no harm by participating in this study. But the any sector who wants clearance management as a whole may benefit from the result of the study.

Confidentiality

The answers and comments you provide will be kept strictly confidential and will be used for the research purpose only. No personal identification is required and therefore you can freely express your opinion. You do not have to answer any question if you do not want. You have right to stop filling this questionnaire at any time. However, the information that you will provide is very useful to achieve the objective of this study.

Considering this assurance, I would greatly appreciate your co-operation in responding to this questionnaire and the time that we need is about 20-25 minutes to fill it.

Contact Person

Ermias Getachew A Postgraduate student,

Program of computer science, Addis Ababa University

Email: - egetachew5@gmail.com

Thank you for your willingness to participate in this study.

Do you agree to participate in this research? (Indicate by making a "_" mark)

Yes _____ No _____

Annex B Questionnaires part 2

Background Information about the user

Instruction: - In this section please circle the numbers (s) in front of the choices that represent(s) your appropriate answer out of the list else write on specify parts.

Section I. Demographic characteristics	
1	What is your age? 1. 18-30 2. 30-45 3 45-60 4. greater than 60
2	What is your gender? 1. Male 2. Female
3	Educational level 1. Elementary 2. High school 3. Diploma 4. Degree 5. Masters 6. other

Annex C Questionnaires for Usability Testing

Questionnaires for Usability Testing

Generic clearance management system usability testing questionnaires which helps to know user satisfaction about the services provided.

Prototype

The following items relate the different features of generic clearance management system. Please indicate the extent to which you agree or disagree with each of the following statements by ticking "√" from 1 to 5 that best represents your level of agreement with the statement.

Note: 1- Strongly Disagree, 2- Disagree, 3- Neutral, 4- Agree, 5- Strongly Agree

No	**Questions**	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
Section I. **Simplicity:**						
1	Are you satisfied by the user interface of the application?					
2	Do you found the system unnecessarily complex?					
3	Do you think that I would need the support of a technical person to be able to use this system?					
4	Do you agree with the idea that the new system reduces unnecessary physical appearances to different office for signature?					
5	Are Content, Color, Icons, Images used are aesthetically pleasing?					
Section II. **Accuracy::**						
6	Does the system is compatible to your mobile?					
7	Do you get too much inconsistency in this system?					

8	Do you think the system can produce data correctly without errors for each clearance item according to the given category?					
9	Does the system totally avoid the burdens and challenges you faced To gate the signature of different office from manual journals to your clearance?					
10	Does the system give enough description when the error occurs?					
11	Do you think the menu items and links allowed to you to access are Consistently located and work without failure?					
12	Is the system useful and adds value to you?					
Section III. **Security(confidential, integrity and availability):**						
13	Do you felt very confident using the system?					
14	Do you think the security of user is maintained properly?					
15	Do you think the system is secure enough (in terms of privacy, data integrity)?					
Section V: **Speed::**						
16	Do you think response time for most operations is fast enough?					
17	Do you found the system very cumbersome to use?					

18	Do would imagine that most people would learn to use this system very quickly?					
19	Do you think the developed system can save your time to get your clearance?					
20	Are you satisfied by the clearance system application service?					

Please write any other comment about the generic clearance management System:

_____

_____

_____

Declaration

I, the undersigned, declare that this research is my original work and has not been presented for Degree in any other university, and that all sources of materials used for the research have been Acknowledged.

Declared by:

Name: **Ermias Getachew Assfea**

Signature: 

Date: _____18/6/2020_____

Confirmed by advisor:

Name: **Dr Fekade Getahun**

Signature: _____

Date: _____

Place and date of submission: Addis Ababa University, Jun, 2020.