

Adventist University of Central Africa

**PARENTS TEACHERS SYSTEM**

**CASE STUDY: Kigali Greater Heights School (KGHS)**

A final year project Presented in partial fulfillment of the Requirements for the  
degree of BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

Major in

**NETWORKING AND COMMUNICATION SYSTEMS**

By

NSHUTI Philemon

November, 2023

# **ABSTRACT**

Research Project for Bachelor Degree in Information Technology

Emphasis in Networking and Communication Systems

Adventist University of Central Africa

**TITLE: PARENTS TEACHERS SYSTEM**

Name of researcher: NSHUTI Philemon

Name of supervisor: Dr. Jason

Date Completed: November 2023

The central objective of this research is to develop and implement an innovative "Parents Teachers System" for Kigali Greater Heights School (KGHS), fostering enhanced communication between parents and teachers. Renamed the "Parent-Teacher Communication Platform," this system aims to elevate parental involvement in their children's academic journey and school life.

The design process involves gathering requirements through interviews, surveys, and focus groups, tailoring the system to meet the unique needs of KGHS. The collected data will be translated into detailed system specifications using UML, providing a clear roadmap for development.

During the construction phase, cutting-edge technologies will be employed to ensure the platform's resilience. Rigorous testing, including unit, integration, and user acceptance tests, will be conducted to guarantee precision and effectiveness. Post-development activities include continuous monitoring, regular updates, and comprehensive user training, ensuring seamless alignment with evolving educational requirements.

Integration of data analytics into the system will offer valuable insights into communication patterns, facilitating optimized interaction between parents and teachers. Robust security measures will be implemented to safeguard sensitive information, ensuring compliance with data privacy regulations and instilling confidence in users.

The roles within the system include Parents, Teachers, Headteacher, and School Director, each with distinct responsibilities. Parents can send queries, concerns, and appointment requests, while receiving announcements from teachers and the administration. Teachers respond to parent queries, make announcements, and inquire with the school administration. The Headteacher handles queries involving the school administration, sends out announcements, and collaborates with the School Director. The School Director responds to inquiries from the Headteacher and accepts appointments from parents for deeper discussions.

In conclusion, the implementation of the "Parent-Teacher Communication Platform" is poised to significantly enhance communication and collaboration at KGHS. The user-friendly interface and advanced functionalities will set a new standard for effective parent-teacher engagement, aligning with the school's commitment to holistic education. Adhering to best practices in monitoring, training, data analytics, and security will enable KGHS to fully leverage the platform's potential, fostering lasting benefits for parents, teachers, and learners.

## **DECLARATION**

I, **NSHUTI Philemon** hereby declare that the project report **PARENTS TEACHERS SYSTEM** based on my own work carried out during the course of my study under the supervision of Dr. Jason. I submit the statements made and conclusions drawn are an outcome of my research work, and it has not been previously submitted anywhere.

**Signature** : .....

**Date** :...../...../.....

## **APPROVAL**

I, Dr. **JASON** hereby certify that this project report has been done under my supervision and submitted with my approval.

**Signature**.....

**Date**...../...../.....

## **DEDICATION**

I dedicate this book

To my lovely family

To all my friends and relatives

and To my supervisor for his

guidance.

## TABLE OF CONTENTS

ABSTRACT.....	i
DECLARATION .....	iii
APPROVAL .....	iv
DEDICATION.....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES .....	ix
LIST OF TABLES.....	x
LIST OF ABBREVIATIONS .....	xi
ACKNOWLEDGEMENT .....	xii
CHAPTER I.....	1
GENERAL INTRODUCTION.....	1
Introduction .....	1
Background of the Study.....	1
Problem Statement .....	2
Choice and Motivation in the Study.....	2
Objectives of the Study .....	2
General Objective .....	2
Specific objectives .....	2
Scope of the Study.....	3
Method and Techniques Used in the Study.....	3
Documentation.....	3
Interview .....	3
Expected Results .....	3

Organization of Report.....	4
CHAPTER 2 .....	5
ANALYSIS OF THE EXISTING SYSTEM.....	5
Introduction .....	5
Description of the Existing System Environment .....	5
Historical Background .....	5
Mission .....	8
Vision.....	8
Description of the Existing System.....	8
Analysis of the Existing System .....	9
Modelling of the Current System .....	9
Problem of the Current System .....	9
Proposed Solutions.....	10
System Requirements.....	11
Functional Requirements .....	11
Non-Functional Requirements.....	12
CHAPTER 3 .....	13
REQUIREMENTS ANALYSIS AND DESIGN OF THE NEW SYSTEM.....	13
Introduction .....	13
Unified Modeling Language (UML).....	14
Design of the New System - Diagrams .....	19
Use Case Diagram .....	19
Class Diagram.....	24
Sequence diagram.....	25
Activity Diagram .....	28



Database Design .....	28
Data Dictionary.....	29
System Architecture Design .....	32
CHAPTER 4 .....	33
IMPLEMENTATION OF THE NEW SYSTEM.....	33
Introduction .....	33
Technologies used .....	33
Presentation of the New System (screenshots and briefings).....	34
Hardware and Software Requirements .....	42
CHAPTER 5 .....	44
CONCLUSION AND RECOMMENDATIONS .....	44
Conclusion.....	44
Recommendations .....	46
REFERENCES .....	47
Books .....	47
Websites.....	47
APPENDICES .....	49
Curriculum Vitae.....	50

## LIST OF FIGURES

Figure 0: Modeling of the Current System .....	9
Figure 1: Use case Diagram.....	21
Figure 2: Class Diagram .....	25
Figure 3: Sequence diagram for sending queries .....	26
Figure 4: Sequence diagram for Login .....	27
Figure 5: <i>Sequence diagram for Teacher Making Announcements</i> .....	27
Figure 6: <i>Sequence diagram for Parent Requesting Appointment</i> .....	27
Figure 7: Activity Diagram.....	28
Figure 8: Database Diagram .....	29
Figure 8: Data Dictionary Diagram .....	31
Figure 9: Data Dictionary Diagram .....	30
Figure 10: Architecture Diagram .....	32
Figure 11: Login Page.....	34
Figure 12: Creating User accounts with parent role .....	35
Figure 13: School director Dashboard .....	35
Figure 14: Head Teacher Dashboard .....	36
Figure 15: Parent and Teachers Dashboards.....	37
Figure 16-17: <i>chat screen between Teacher and parent</i> .....	37-38
Figure 18: Manage Users .....	38
Figure 19: View Announcement.....	39
Figure 20: User Setting Scheduler .....	39
Figure 21-23: requesting and Notications of Appoitments.....	40

## LIST OF TABLES

Table 1: Use-case diagram elements.....	20
Table 2: Use case description for Parent Requests Appointment .....	22
Table 3: Use case description of Teacher Approves Appointment.....	23
Table 4: Use case description for Head Teacher Sends Announcement .....	23
Table 5: Use case description for School Director Manages Users.....	24

## **LIST OF ABBREVIATIONS**

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
AUCA	Adventist University of Central Africa
KGHS	Kigali Greater Heights School
CSS	Cascading Style Sheet
DOM	Document Object Model
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model-View-Controller
PHP	Hypertext Preprocessor
REST	Representational State Transfer
SQL	Structured Query Language
UI	User Interface
UX	User Experience

## **ACKNOWLEDGEMENT**

Firstly I thank God for all He has been with me in this journey of my studies and even till now. There are many students who dropout due to different reasons but by His grace I have been at university till the end of all classes.

I can't forget that as a student I needed the basic knowledge to be where I am now with skills I have. My appreciation goes to AUCA administration for the academic program and positive impact on students. I have noticed the lifestyle of student after joining AUCA. Thank you for changing the student's mindset.

I would like to thank my parents, who helped me in their prayers, advices, and encouragement. They virtually and financially worked along with me.

I am extremely grateful to the departments staff members who always there for students to be successful. I am deeply thankful to Dr. Jason as my supervisor, who guided me in the accomplishment of this work, his professional assistance and ideas.

I am grateful for my colleagues who played a crucial role in my studies, their team working and support gave me a lift and experience.

I am thankful to my friends in general for being always there for me where things seemed to be impossible.

Finally, thanks a lot to everyone who has been involved in this work directly or indirectly for their help and contribution.

God Bless you all!

**NSHUTI Philemon**

# **CHAPTER I**

## **GENERAL INTRODUCTION**

### **Introduction**

Today's generation heavily relies on information technology across various sectors, transforming the way companies manage their daily operations. Businesses globally have adopted IT to enhance efficiency, communication, and productivity. However, in some regions, such as Rwanda, organizations face technical challenges that hinder optimal operations. This study focuses on improving information technology in Rwanda, specifically within KGHS (Kigali Greater Heights School). The proposed solution, the Parents Teachers System, aims to facilitate communication between parents, teachers, and school administrators, fostering greater parental involvement in their children's education and school life.

### **Background of the Study**

The initial phase of the Parents Teachers System (PTS) project at KGHS necessitates the formation of a PTS Implementation Team, comprising members from diverse roles such as Parents, Teachers, the Head Teacher, and the School Director. This collaborative team is tasked with initiating and overseeing the implementation of the PTS to facilitate effective communication between parents and teachers. Once the PTS framework is established, notifications are disseminated internally, enabling users to familiarize themselves with the system's functionalities. The PTS is then activated and maintained within the school's communication infrastructure, ensuring seamless interaction to enhance parental involvement in their children's studies and school life.

## **Problem Statement**

The existing communication system at KGHS encounters issues, including a lack of streamlined interaction, manual processes, delays in addressing parental concerns, and challenges in effective appointment scheduling. These issues affect the overall communication flow and lead to operational inefficiencies within the school community.

## **Choice and Motivation in the Study**

The inspiration for the Kigali Greater Heights School (KGHS) System arises from the challenge parents face in effectively communicating concerns and scheduling appointments within the existing framework. With logistical challenges such as transportation costs and time constraints hindering parents and teachers from physically meeting, there is a compelling need for an automated solution.

## **Objectives of the Study**

### **General Objective**

The general objective is to implement a computerized system, the Parents Teachers System, to enhance communication and collaboration between parents, teachers, and school administrators at KGHS.

### **Specific objectives**

Include facilitating the ordering process, improving data management, enhancing data security, and creating a digital repository of parent and student information.

## **Scope of the Study**

This project focuses on enhancing communication and collaboration within KGHS through the implementation of the Parents Teachers System. The primary goal is to streamline communication channels, boost efficiency, and enable timely interaction between parents, teachers, and school administrators.

## **Method and Techniques Used in the Study**

### **Documentation**

The study utilizes documentation analysis, referencing relevant educational materials and reports on KGHS.

### **Interview**

Furthermore, discussions with key participants, including the Project Participant and Academic Program Manager, offer valuable perspectives into the operational aspects of KGHS School Management system functions and initiatives.

## **Expected Results**

The Parents Teachers System will be an online web application facilitating seamless communication between parents, teachers, and school administrators. The system aims to store data securely, reduce waiting times for appointments, eliminate the need for physical meetings, and provide real-time updates on student-related information.



## **Organization of Report**

This study comprises five chapters, maintaining the structure outlined in the provided content. Each chapter contributes to a comprehensive exploration of the Parents Teachers System for KGHS, from introduction to conclusion and recommendations.

## **CHAPTER 2**

### **ANALYSIS OF THE EXISTING SYSTEM**

#### **Introduction**

In the Kigali Greater Heights School (KGHS) system project, success is not solely dependent on hard work and learning from past mistakes. Instead, a thorough exploration of the existing system is essential to understand its shortcomings and develop better solutions. Before commencing work on a new system, it's crucial to analyze the current system thoroughly. This process helps us determine if a new system is needed and enables us to identify areas for improvement in KGHS's communication practices between parents, teachers, and school administration.

#### **Description of the Existing System Environment**

##### **Historical Background**

KGHS's journey began in 2014, growing into an International School serving around 600 learners from six different countries. Located in Masoro cell, Ndera sector, Gasabo district in Kigali city, the school boasts appealing buildings on a one-hectare land with a green environment and modern facilities. With divisions for Nursery, Primary, and upcoming Secondary, the school is accredited to offer national and Cambridge International Assessment programs.

The analysis of KGHS's existing communication system reveals several weaknesses and challenges that need to be addressed. The following is an overview of the key aspects analyzed:

- **Limited Parent-Teacher Communication:** The current system lacks a structured platform for effective communication between parents and teachers. This hinders the involvement of parents in their children's studies and school life.
- **Inadequate Announcement Mechanism:** The teacher struggles to make announcements about school-related functions and programs in a timely and organized manner. This impacts the dissemination of crucial information to parents.
- **Absence of Appointment Requests:** The current system lacks a feature for parents to request appointments with the head teacher for discussions on deeper matters. This hampers effective communication on important issues.
- **Limited Access Levels:** The existing system does not provide differentiated access levels for users, restricting the ability to tailor communication privileges based on roles within the school.

Based on this analysis, it is clear that the existing communication system at KGHS suffers from inefficiencies, inadequate features, and restricted access levels. These issues highlight the need for a more robust system that enhances parent-teacher communication, facilitates timely announcements, incorporates appointment request features, and allows differentiated access levels based on roles within the school.

The existing system environment of KGHS's communication system encompasses various aspects, including technology infrastructure, software applications, data management, and communication systems. Here is a description of the existing system environment:

#### **Technology Infrastructure:**

**Hardware:** KGHS maintains a network of servers, computer workstations, laptops, and mobile devices to support its operations. These devices are used by school staff across different departments for their day-to-day tasks.

**Network:** The school has a local area network (LAN) connecting its offices, classrooms, and administrative facilities. It also utilizes wide area network (WAN) connections to enable communication between geographically dispersed locations.

**Internet Connectivity:** Kigali Greater Heights School relies on high-speed internet connections to facilitate online operations, communication, and data exchange.

### **Software Applications:**

**Communication System:** The school employs a communication system to manage interactions between parents, teachers, and the administration. This system includes features for sending queries, making announcements, and requesting appointments.

**Announcement Platform:** KGHS uses an announcement platform to disseminate information about school-related functions and programs to parents and teachers.

**Appointment Request Feature:** The system includes a feature for parents to request appointments with the head teacher for discussions on deeper matters.

**Access Control System:** The school utilizes an access control system to manage different levels of access for users, including parents, teachers, head teacher, and school director.

### **Data Management:**

**Databases:** KGHS maintains databases to store and manage various types of data, such as parent information, student details, communication records, and appointment requests.

**Data Security:** The school implements security measures to protect sensitive data from unauthorized access, including firewalls, encryption, access controls, and regular data backups.

**Data Analysis and Reporting:** KGHS employs tools for data analysis and reporting to extract insights from its communication and operational data, enabling informed decision-making.

## **Mission**

The Kigali Greater Heights School mission is “to inspire students to become independent, God-fearing lifelong learners and productive citizens of a global society.”

## **Vision**

Kigali Greater Heights School envisions producing "respecting, pro-active, and knowledgeable learners built on a firm foundation for higher academic, technical, ethical, and moral values.”

## **Description of the Existing System**

The Function/Department Where the Problem Exists

Currently, the Parent-Teacher Communication and Announcement System face challenges. Parents struggle to effectively communicate concerns to teachers, and the announcement mechanism lacks efficiency. The absence of an appointment request feature further hampers discussions on important matters.

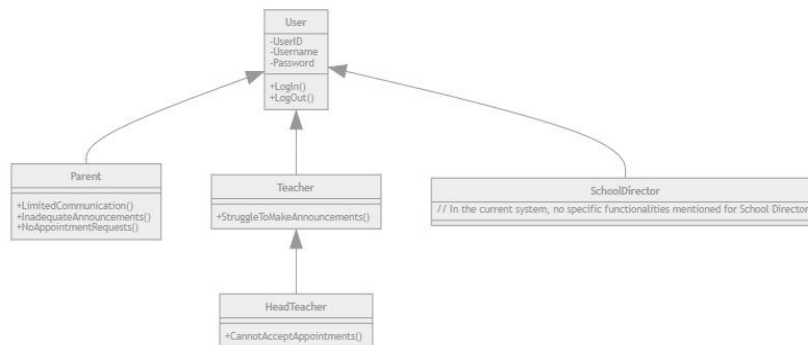
Actors in the Department Roles and Responsibilities

Roles:

- Parent: Sends queries and concerns, requests appointments from the head of school, and receives announcements from teachers or administration.
- Teacher: Replies to queries from parents, makes announcements to parents, and communicates with the school administration where necessary.
- Head teacher: Replies to queries from parents involving the school administration, sends out announcements regarding school programs, and communicates with the school director where necessary.
- School Director: Replies to inquiries from the head teacher and accepts appointments from parents for physical meetings to discuss important matters.

## Analysis of the Existing System

### Modelling of the Current System



*Figure 0: Modeling of the Current System*

### Problem of the Current System

The Kigali Greater Heights School (KGHS) System currently faces several challenges that hinder efficient communication and collaboration among parents, teachers, and school administration. These problems include:

- **Limited Parent-Teacher Communication:** The system lacks a structured platform for effective communication between parents and teachers, hindering parental involvement in their children's studies and school life.
- **Inadequate Announcement Mechanism:** Teachers struggle to make timely and organized announcements about school-related functions and programs, impacting the dissemination of crucial information to parents.
- **Absence of Appointment Requests:** The system does not have a feature for parents to request appointments with the head teacher for discussions on deeper matters, hampering effective communication on important issues.
- **Limited Access Levels:** The existing system does not provide differentiated access levels for users, restricting the ability to tailor communication privileges based on roles within the school.

## **Proposed Solutions**

As a solution, I am introducing a new system named the KGHS System, designed to facilitate seamless communication between parents, teachers, and school administration. This system aims to address the identified challenges and enhance overall communication effectiveness.

The new system will provide the following improvements:

- **Structured Parent-Teacher Communication:** The system will feature a dedicated platform for effective communication between parents and teachers, fostering increased parental involvement in their children's studies and school activities.

- **Timely and Organized Announcements:** Teachers will have a streamlined mechanism for making timely and organized announcements about school-related functions and programs, ensuring efficient information dissemination to parents.
- **Appointment Request Feature:** The system will include a feature for parents to request appointments with the head teacher for discussions on deeper matters, facilitating effective communication on important issues.
- **Differentiated Access Levels:** The new system will offer differentiated access levels for users, allowing tailored communication privileges based on roles within the school, ensuring secure and efficient communication channels.

#### Requirement Specifications: Functional and Non-functional Requirements

The requirement specification activity includes a set of features that the system must have to be used. A functional requirement is a specification of a function that the system must support, while non-functional requirements are constraints on the operation of the system that are not related directly to a function of the system.

## **System Requirements**

### **Functional Requirements**

- The application should be a web-based application.
- The system should allow parents to create user accounts.
- Parents should be able to send queries and concerns through the system.
- Teachers should be able to reply to queries and make announcements through the system.



- The system should allow parents to request appointments with the head teacher.
- The system should generate reports on communication activities for administrators.
- The system should allow users to log in and log out.

## **Non-Functional Requirements**

### **Security**

- The system must encrypt user passwords.
- The system should provide the right privileges according to user types.
- Any user has to log in to perform a certain action.

### **Maintainability**

- The system should be easily maintainable when needed.

### **Performance**

- The system should be available 24/7.
- The system must perform user requests within 1-3 seconds.

### **Accessibility**

- The system should be accessible on laptops, tablets, and mobile phones.
- The system should be accessible within local network infrastructures, web-based.
- The system should be online for whoever needs it.
- The system should be accessible to authorized personnel.

### **Privacy**

- The system shall be able to protect users' privacy.

# **CHAPTER 3**

## **REQUIREMENTS ANALYSIS AND DESIGN OF THE NEW SYSTEM**

### **Introduction**

After a thorough examination of the existing system in chapter 2, our next phase is to delve into system enhancement. The aim is to create a robust and successful system that fully meets user needs. To address the challenges of the current system, a meticulous analysis is crucial, leading to a well-defined design that serves as a solid foundation for effective solutions.

System enhancement involves studying and understanding the current system to identify requirements, limitations, and areas for improvement. This includes analyzing components, processes, and interactions to understand its current state and shortcomings.

Moving forward, system design is the process of creating a blueprint for a new or improved system based on the findings of the analysis. This includes defining the system's structure, architecture, interfaces, and behavior to address identified requirements and enhance overall performance and functionality.

Together, system analysis and system design form a comprehensive approach to evaluate, optimize, and create systems that meet specific objectives and deliver desired outcomes. The focus of system design is on achieving the system's objectives.

## **Analysis and Design Methodology**

### **Object Oriented Methodology (OOM)**

Providing a structured framework for system development, OOM begins with an analysis phase, gaining a thorough understanding of user requirements. The subsequent design stage formulates an application model with intricate details for a comprehensive representation.

The main building blocks in Object Oriented Methodology are Classes and Objects, where an Object is an instance of a class, representing individual entities with defined boundaries in the software system. In contrast, a class serves as a blueprint or description of a group of objects with similar attributes and behaviors.

### **Unified Modeling Language (UML)**

A general-purpose modeling language in software engineering, UML offers graphic notation techniques for visual models of object-oriented software-intensive systems. It provides a standard way to visualize a system's architectural blueprints, including activities, actors, business processes, database schemas, and reusable software components.

The Unified Modeling Language (UML) has a standard way to visualize a system's architectural blueprints, including elements like:

- Activities
- Actors
- Business Processes
- Database schemas
- Reusable software components

## **Object-oriented analysis and design (OOAD)**

OOAD is a popular technique for analyzing and designing applications, systems, or businesses. It employs the object-oriented paradigm and visual modeling throughout the development life cycle, facilitating effective communication with stakeholders and ensuring high product quality. UML, the Unified Modeling Language, is the standard notation used for representing these models.

### **Object-Oriented Analysis:**

An approach examining how data interacts within a system, based on the object-oriented programming paradigm. It defines and analyzes requirements for system success.

### **Object-Oriented Design:**

Involves planning a system of interacting objects to address a software problem systematically. Building upon identified candidate objects, it extends their definitions significantly, emphasizing how the system achieves functionality. OOD expands upon the analysis model to generate implementation specifications, focusing on how the system achieves functionality.

## **UML Concepts**

<b>Name of class</b>
<b>Attributes</b>
<b>Operations()</b>

- The upper part holds the name of the class.
- The middle part contains the attribute of the class.
- The last part gives the method or operation the class can take or undertake.

**An attribute** is named property of a class that describes a range of values that instances of the property may hold.

**A method** is the implementation of a service that can be requested from any object to the class to affect behavior.

Relationships: A relationship is a connection among things.

They are most important relationships which are association, generalization and dependency

- **Association**

Use cases and actors are connected through association relationships, indicating the interactions between actors and the corresponding use cases. A line between an actor and a use case represents the association. This association signifies two-way communication between the actor and the use case.

- **Generalization**

Generalization: Generalization represents a relationship between a general entity (superclass or parent) and a more specific entity (subclass or child). It is denoted by an arrow pointing from the subclass to the superclass. This relationship signifies that the subclass inherits attributes, behavior, and relationships from the superclass.

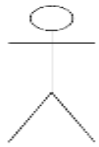
- **Dependency**

Dependence is a weaker form of relationship which indicates that one class depends on another because it uses it at some point of time.

- **Extend**

The extension relationship indicates that the behavior of the extension use case can be included in the extended use case under certain conditions. It is represented by a dashed arrow from the extension use case to the extended use case, labeled as "extended." Notes or constraints can be associated with this relationship to specify the conditions under which this behavior will be executed.

**Actor:** An actor represents a role that can be performed by a user or another system when interacting with the system. It is not tied to a specific individual but rather signifies a particular function or responsibility. In the case of system interactions, the actor can be depicted as a stick figure labeled with the actor's name.



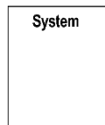
**Use Case:** A use case is a significant system process that provides a benefit to one or more actors. It is labeled with a descriptive verb-noun phrase and represents a significant piece of system functionality.



### **System boundary boxes (optional)**

**System Boundary:** A rectangle, known as the system boundary box, is drawn around the use cases to define the scope of the system. Anything contained within the box represents the

functionality that is within the system's scope, while anything outside the box is considered out of scope.



In this project we will focus only on Actors, Use Case and System boundary as UML notations.

### **Software development process**

In culinary arts, the cooking process involves dividing the culinary work into distinct phases or stages, each consisting of activities aimed at effective planning and management. It is often considered a subset of the culinary creation life cycle. The process may include predefined recipes and techniques used by a cooking team to prepare or enhance a dish, ensuring it meets the diner's expectations.

**Flavorful Approaches:** Flavorful approaches to cooking can utilize various traditional methods such as baking, grilling, frying, boiling, and sautéing. However, sautéing using a high-heat methodology is commonly associated with flavorful approaches.

**Difference from Conventional Approaches:** The primary distinction between a conventional approach, like traditional cooking, and a flavorful approach lies in how a dish is composed. In conventional approaches, the dish composition process is either ingredient-centric or technique-centric. However, flavorful approaches recognize the close relationship between ingredients and techniques, making it challenging to prioritize one over the other as the primary focus.

## **Unified Process (UP)**

Unified Process (UP): The Unified Process is an architecture-centric, use case-driven, iterative, and incremental development process, also known as the unified software development process. It aims to incorporate the favorable aspects of traditional software process models while embracing the principles of agile software development.

The UP divides its phases (Inception, Elaboration, Construction, and Transition) into a series of time-boxed iterations. Each iteration delivers an increment, which enhances the system by adding or improving its functionality compared to the previous release.

### **Analysis of the new system**

Analysis Phase: The analysis phase addresses essential questions about the system, including its intended users, functionality, and usage locations. It involves studying the system, analyzing it, and selecting logical alternatives to model and design an information system. Use cases, which describe interactions between users and the system, are a highly effective tool in the analysis process.

## **Design of the New System - Diagrams**





### **Use Case Diagram**

Use Case: In system analysis, a use case is a methodology employed to identify, clarify, and organize system requirements. It comprises a set of potential interaction sequences between users and the system within a specific environment, all aimed at achieving a particular goal. The use case should encompass all system activities that hold significance for the users. It can be visualized as a collection of possible scenarios associated with a specific objective.



**Use Case Diagram:** A use case diagram is a UML diagram that depicts a business or software system, its external users (referred to as actors), and a series of actions (referred to as use cases) that users can or should perform while utilizing the system. Use case diagrams are utilized to illustrate the system's functionality from the perspective of an external user.

**The symbols below are used in use case diagram:**

Description	Shape
<b>An actor:</b> Is a Person/System that derives benefit from and is external to the subject.	
<b>A use case:</b> Represents a major piece of system functionality.	
<b>A Field relationship:</b> <ul style="list-style-type: none"> <li>Links an actor with the use case(s) with which it interacts.</li> </ul>	
<b>A boundary:</b> It is a box drawn around the use case to denote the edge or boundary of the system being modeled.	

*Table 1: Use-case diagram elements*

The figure below describes the operations of the new system and the stakeholders through the use case diagram.

## Use-Case Diagram

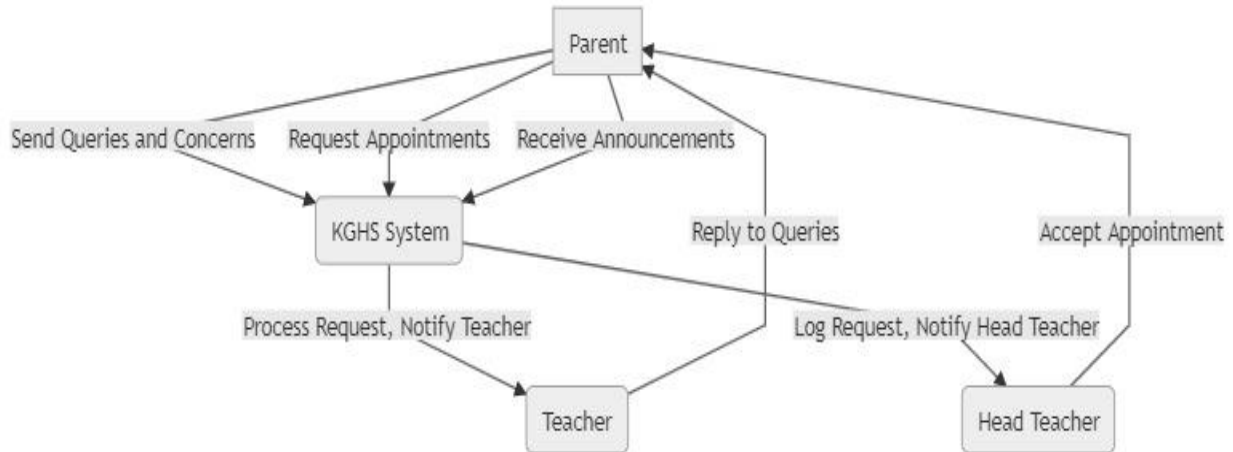


Figure 1: Use case Diagram

### Description of the application case

Use Case description details what a use case uses, and what it requests in order to be well executed. Each use case looks like this:

- **Name:** A name of a use case.
- **Description:** what a system intends to do
- **Actor:** The actor involved in the use case
- **Pre-condition:** The system state before the use case can begin
- **Post-condition:** The system state when the use case is over
- **Normal flow:** The actual steps of the use case
- **Alternative flow:** Steps which may happen in case a normal flow fails.

### Use case description for Parent Requests Appointment

<b>Name:</b> Parent Requests Appointment
<b>Actor:</b> Parent
<b>Description:</b> A parent initiates a request for an appointment with either the Head of School or a Teacher.
<b>Pre-condition:</b> Parent is logged in.
<b>Post-condition:</b> Appointment request is submitted.
<b>Normal flow:</b> <ol style="list-style-type: none"><li>1. Parent logs in.</li><li>2. Navigates to the appointment section.</li><li>3. Selects the recipient (Head of School or Teacher).</li><li>4. Chooses a suitable time.</li><li>5. Submits the request.</li></ol>
<b>Alternative flow:</b> <ul style="list-style-type: none"><li>- Parent cancels the appointment request.</li></ul>

*Table 2: Use case description for Parent Requests Appointment*

### Use case description for Teacher Approves Appointment

<b>Name:</b> Teacher Approves Appointment
<b>Actor:</b> Teacher
<b>Description:</b> A teacher reviews and approves an appointment request from a parent.
<b>Pre-condition:</b> Teacher is logged in.
<b>Post-condition:</b> Appointment is confirmed
<b>Normal flow:</b> <ol style="list-style-type: none"><li>1. Teacher logs in.</li><li>2. Accesses the appointment requests.</li><li>3. Reviews details.</li></ol>

4. Approves the appointment.
<b>Alternative flow:</b> - Teacher reschedules the appointment.

*Table 3: Use case description of Teacher Approves Appointment*

#### **Use case description for Head Teacher Sends Announcement**

<b>Name:</b> Head Teacher Sends Announcement
<b>Actor:</b> Head Teacher
<b>Description:</b> The Head Teacher posts announcements regarding school programs or events.
<b>Pre-condition:</b> Head Teacher is logged in.
<b>Post-condition:</b> Announcement is published.
<b>Normal flow:</b> 1. Head Teacher logs in. 2. Navigates to the announcement section. 3. Composes and posts the announcement.
<b>Alternative flow:</b> - Head Teacher edits the announcement before posting

*Table 4: Use case description for Head Teacher Sends Announcement*

#### **Use case description School Director Manages Users**

<b>Name:</b> School Director Manages Users
<b>Actor:</b> School Director
<b>Description:</b> The School Director performs CRUD operations to manage user accounts for Head Teachers and other Teachers.
<b>Pre-condition:</b> School Director is logged in.
<b>Post-condition:</b> User accounts are created, updated, or deleted as needed.

**Normal flow:**

1. School Director logs in.
2. Navigates to the user management section.
3. Adds, updates, or deletes user accounts.

**Alternative flow:**

- School Director cancels user account creation.

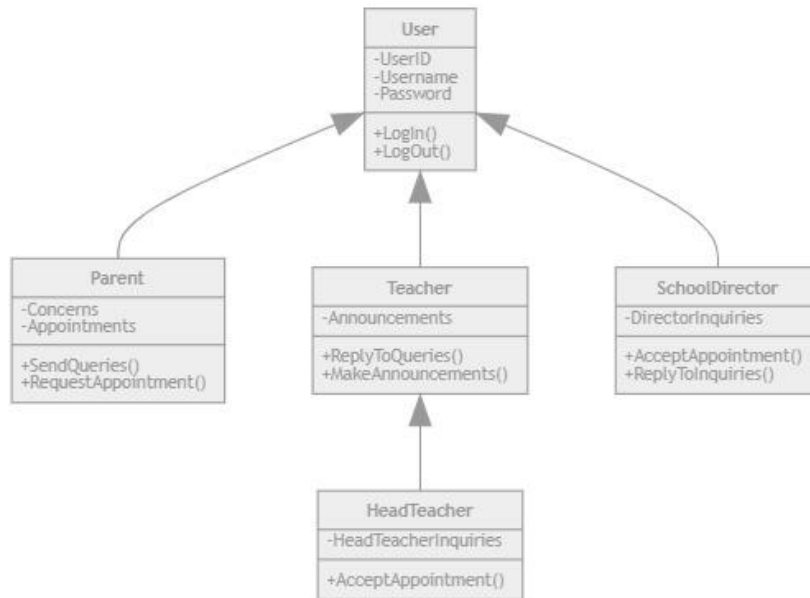
*Table 5: Use case description for School Director Manages Users*

**Class Diagram**

A class diagram visualizes the system's structure by illustrating classes, their attributes, methods, and the associations between classes.

**Importance of class diagram**

1. Class diagrams enable effective communication, serving as a blueprint for software development.
2. They identify key entities, attributes, and relationships, aiding in system analysis and design.
3. Class diagrams promote modular and reusable design, improving system quality and maintainability.



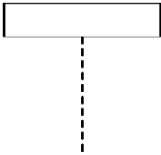
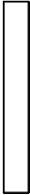
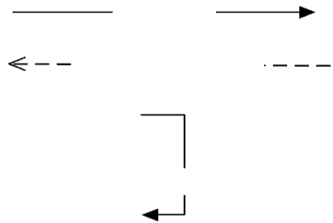
*Figure 2: Class Diagram*

## Sequence diagram

A sequence diagram is a form of interaction diagram which shows objects as lifelines running down the page, with their interactions over time represented as messages drawn as arrows from the source lifeline to the target lifeline. A sequence diagram shows object interactions arranged in time sequence. (Kraemer, 2009)

The notations and their definitions that are used in sequence diagram:

Term and definition	Symbol
<p>An actor:</p> <ul style="list-style-type: none"> <li>It can be a person or system that derives benefit from and is external to the system.</li> <li>It participates in a sequence by sending and/or receiving messages.</li> <li>It is placed across the top of the diagram.</li> </ul>	

<p>An object lifeline:</p> <p>UML object symbol illustrates class role, but doesn't list object attributes. It is placed across the top of the diagram.</p>	
<p>An activation:</p> <p>Activation boxes represent the time an object needs to complete a task. It denotes when an object is sending or receiving messages.</p>	
<p>A message:</p> <ul style="list-style-type: none"> <li>• It conveys information from one object to another.</li> <li>• An operation call is labeled with the message being sent and a solid arrow, whereas a return is labeled with the value being returned and shown as a dashed arrow.</li> </ul>	

### Sequence diagram for creating a layer of the organization structure

As the sequence diagram graphically displays the interaction between actors and the system, the diagram below clearly describes how the user interacts with the system.

### Sequence diagram for creating a Parent sending queries.



Figure 3: Sequence diagram for sending queries

### Sequence diagram for Login.

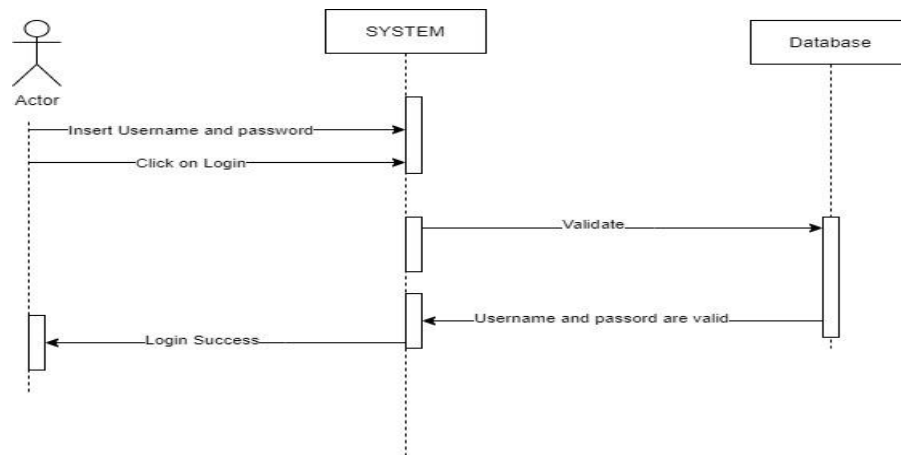


Figure 4: Sequence diagram for Login

### Sequence diagram for Teacher Making Announcements.



Figure 5: Sequence diagram for Teacher Making Announcements

### Sequence diagram for Parent Requesting Appointment.



Figure 6: Sequence diagram for Parent Requesting Appointment



## Activity Diagram

Activity diagram is a visual representation of the Parents Teachers System processes at Kigali Greater Heights School (KGHS), illustrating activities, decisions, and the flow of communication. It depicts how parents, teachers, head teachers, and the school director interact within the system, facilitating effective communication and collaboration. (Reference: KGHS System Documentation)

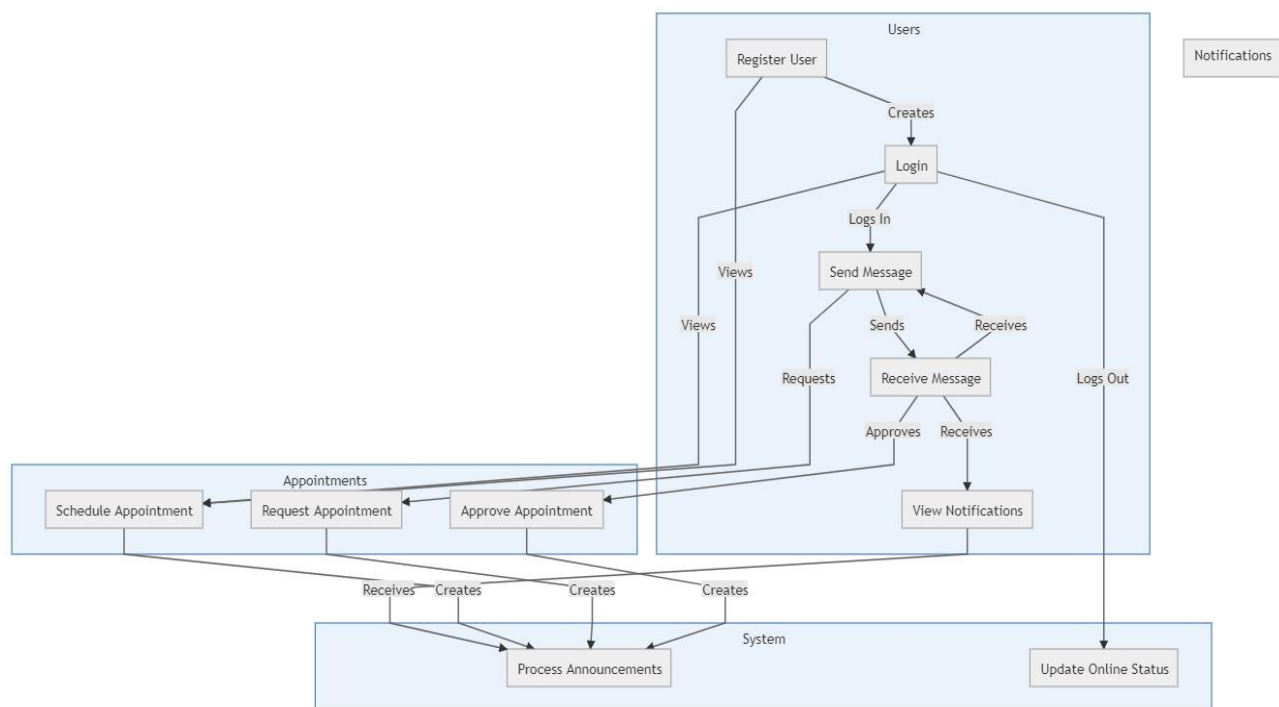


Figure 7: Activity Diagram

## Database Design

Database modeling is the systematic approach of creating a comprehensive data model that encompasses logical and physical design decisions. It includes essential parameters for storage and enables the generation of a data definition language design, ultimately facilitating the creation of a database. (Pty, 2008)

## DATABASE DIAGRAM

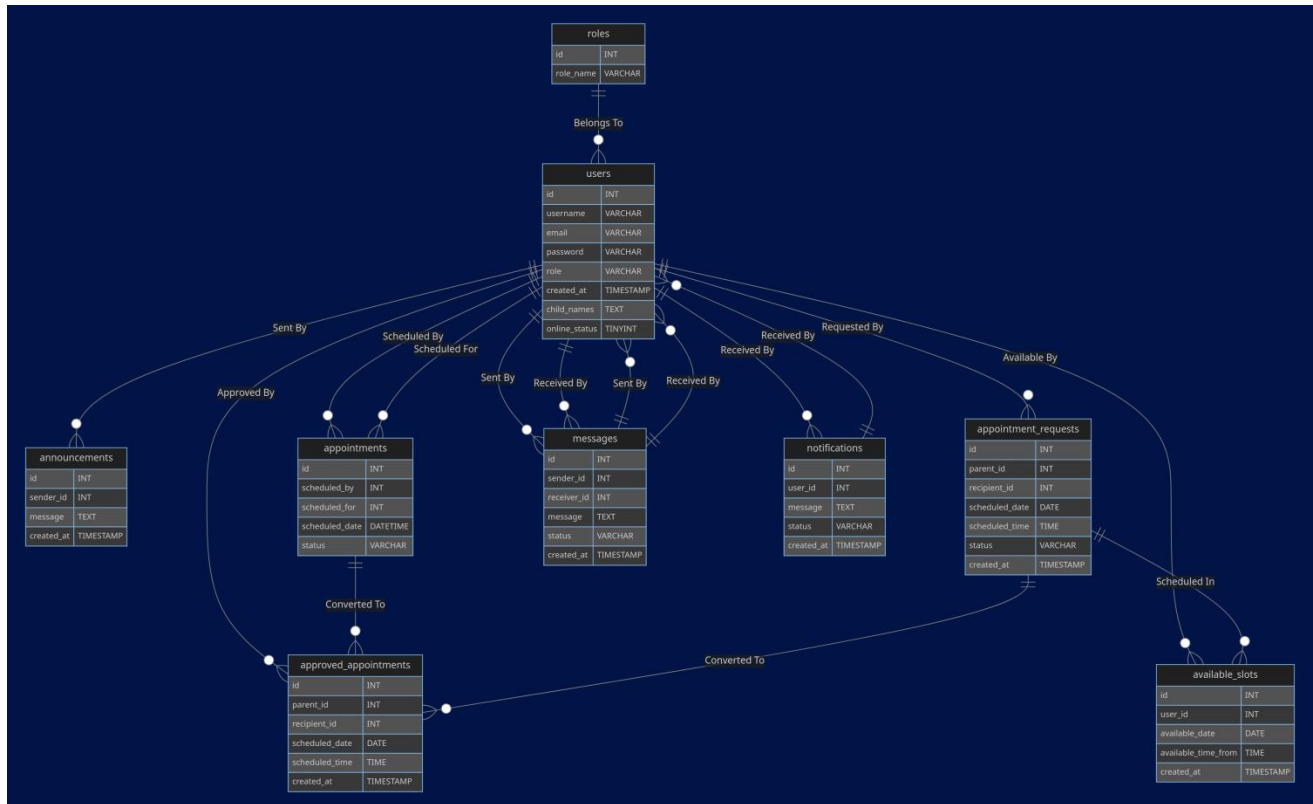


Figure 8: Database Diagram

## Data Dictionary

A data dictionary diagram is a visual representation of a system's data elements, mapping their attributes and relationships. It showcases key data components as interconnected blocks, outlining their functions and associations. The diagram aids in understanding data structures and enables efficient communication between software and hardware architecture, fostering effective human interaction.

### Users Table

Column Name	Data Type	Description
id	INT	User ID (Primary Key)
username	VARCHAR	User's username
email	VARCHAR	User's email
password	VARCHAR	User's password
role	VARCHAR	User's role
created_at	TIMESTAMP	Timestamp of creation
child_names	TEXT	Names of user's children (if applicable)
online_status	TINYINT	Online status indicator

### Roles Table

Column Name	Data Type	Description
id	INT	Role ID (Primary Key)
role_name	VARCHAR	Name of the role

### Announcements Table

Column Name	Data Type	Description
id	INT	Announcement ID (Primary Key)
sender_id	INT	User ID of the sender
message	TEXT	Announcement message
created_at	TIMESTAMP	Timestamp of announcement

### Appointments Table

Column Name	Data Type	Description
id	INT	Appointment ID (Primary Key)
scheduled_by	INT	User ID who scheduled the appointment
scheduled_for	INT	User ID for whom the appointment is scheduled (nullable)
scheduled_date	DATETIME	Date and time of the appointment
status	VARCHAR	Status of the appointment

### Appointment Requests Table

Column Name	Data Type	Description
id	INT	Request ID (Primary Key)
parent_id	INT	User ID of the parent making the request
recipient_id	INT	User ID of the recipient (school, teacher, etc.)
scheduled_date	DATE	Date of the requested appointment
scheduled_time	TIME	Time of the requested appointment
status	VARCHAR	Status of the request (pending, approved, declined)
created_at	TIMESTAMP	Timestamp of request creation

### Approved Appointments Table

Column Name	Data Type	Description
id	INT	Approved Appointment ID (Primary Key)
parent_id	INT	User ID of the parent
recipient_id	INT	User ID of the recipient (school, teacher, etc.)
scheduled_date	DATE	Date of the approved appointment
scheduled_time	TIME	Time of the approved appointment
created_at	TIMESTAMP	Timestamp of approval

### Available Slots Table

Column Name	Data Type	Description
id	INT	Available Slot ID (Primary Key)
user_id	INT	User ID offering the available slot
available_date	DATE	Date of the available slot
available_time_from	TIME	Start time of the available slot
created_at	TIMESTAMP	Timestamp of slot availability

### Messages Table

Column Name	Data Type	Description
id	INT	Message ID (Primary Key)
sender_id	INT	User ID of the sender
receiver_id	INT	User ID of the receiver
message	TEXT	Message content
status	VARCHAR	Status of the message (pending, read)
created_at	TIMESTAMP	Timestamp of message creation

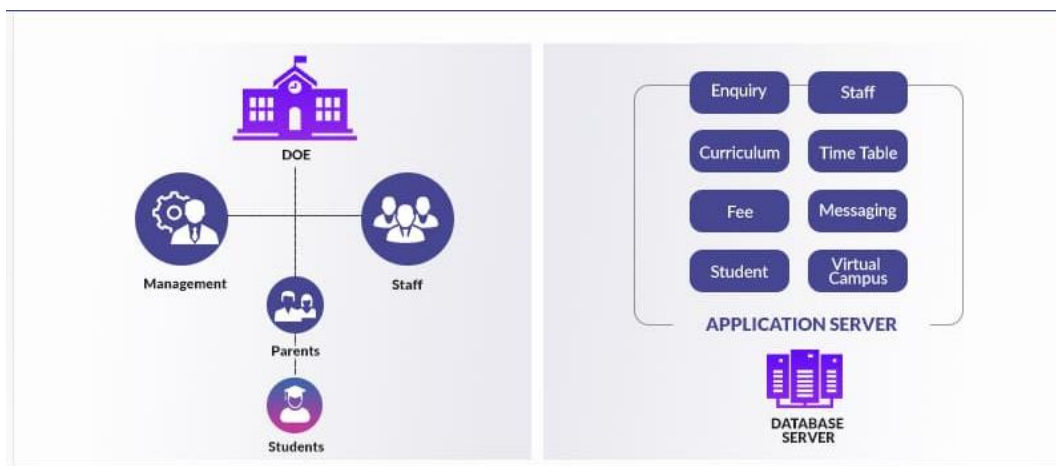
### Notifications Table

Column Name	Data Type	Description
id	INT	Notification ID (Primary Key)
user_id	INT	User ID receiving the notification
message	TEXT	Notification message
status	VARCHAR	Status of the notification (unread, read)
created_at	TIMESTAMP	Timestamp of notification creation

Figure 9: Data Dictionary Diagram

## System Architecture Design

System architecture is a visual depiction of a system that maps functionalities onto hardware and software components, establishes a relationship between software and hardware architecture, and incorporates human interaction. It is a diagram illustrating the system's key components represented by interconnected blocks, showcasing their relationships and functions. (Edrawsoft, 2022)



*Figure 10: Architecture Diagram*

Adopting the Model-View-Controller (MVC) design pattern in the KGHS Parents Teachers System, a PHP-based communication and appointment management platform, proves to be highly advantageous. The implementation of MVC ensures a well-structured and modular codebase, fostering code reuse and elevating the system's maintainability. MVC enforces a distinct separation between data (Model), user interface (View), and application logic (Controller), facilitating streamlined development, debugging, and scalability of the KGHS Parents Teachers System.

# **CHAPTER 4**

## **IMPLEMENTATION OF THE NEW SYSTEM**

### **Introduction**

In this section, we delve into the evolution of the "KGHS School Project," shedding light on the project's technological landscape, functionality, and execution. The narrative unfolds through a meticulous portrayal of the project, complemented by illustrative snapshots and an in-depth exploration of conducted assessments. Through this segment, we aim to provide a nuanced understanding of the KGHS School Project's features, interface, and dependability, ensuring proficient management and monitoring of the school's various aspects.

### **Technologies used**

#### **Frontend Technologies:**

- **HTML:** Used to create the structure of web pages, defining the elements and layout.
- **CSS:** Used to style and format the appearance of web pages, controlling the visual aspects like colors, fonts, and layout.
- **JavaScript:** Used for client-side scripting, enabling interaction with users, controlling the browser, and altering the displayed content dynamically.

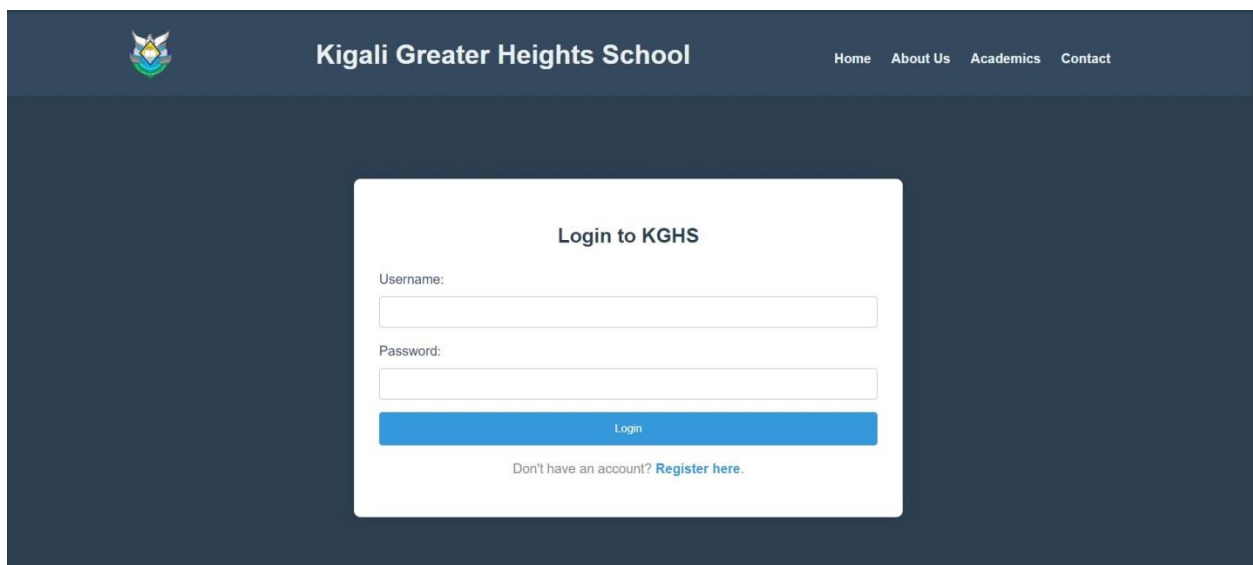
#### **Backend Technologies:**

- **MySQLi:** Utilized for creating and managing the database. It handles storing and retrieving data from the database.
- **PHP:** Used as the server-side scripting language for web development. PHP enables dynamic content generation and interacts with the database to fetch or modify data as needed.
- **XAMPP:** A comprehensive web server solution stack package that combines Apache HTTP Server and MariaDB (a fork of MySQL) as the database server. It provides the necessary environment for running PHP scripts and handling database operations on the server-side.

To summarize, the Frontend technologies (responsible for the user interface and user interactions) consist of HTML, CSS, and JavaScript. On the other hand, the Backend technologies (responsible for server-side operations, data storage, and retrieval) include MySQLi, PHP, and the XAMPP stack (Apache HTTP Server + MariaDB). These technologies work together to create a fully functional web application with both frontend and backend capabilities.

## **Presentation of the New System (screenshots and briefings)**

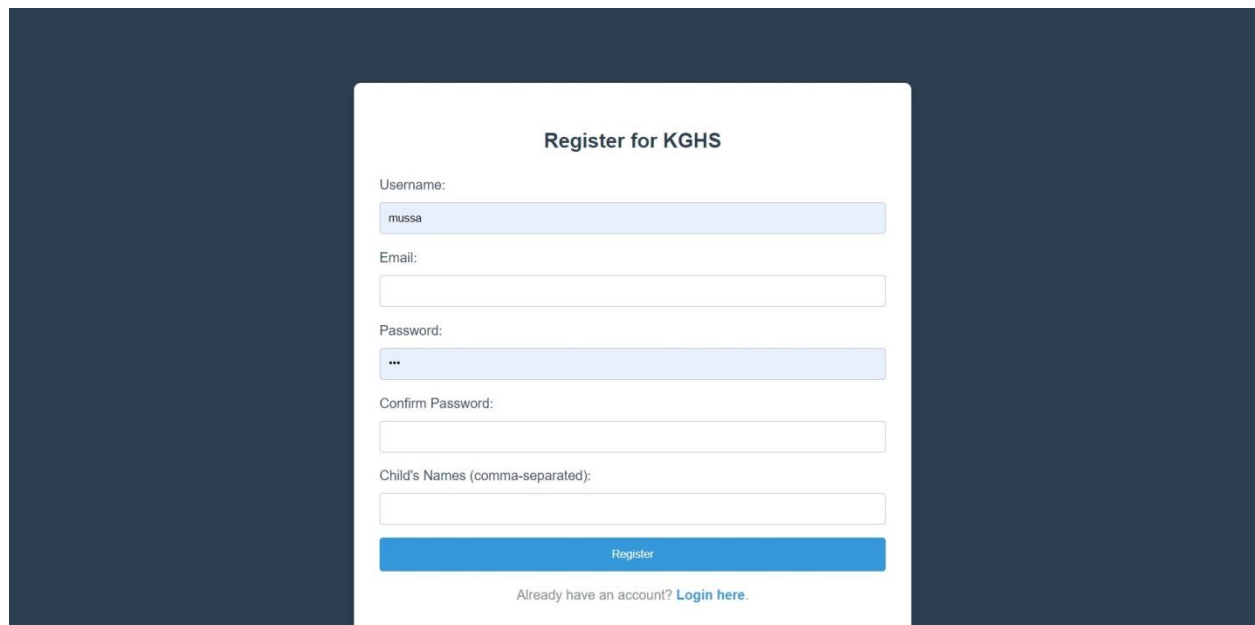
### **Login Page**



The screenshot displays the login interface for Kigali Greater Heights School. The header is dark blue and contains the school's logo, name, and navigation menu. The login form is a white box with a title, input fields for username and password, a login button, and a registration link.

*Figure 11: Login Page*

## Creating Accounts



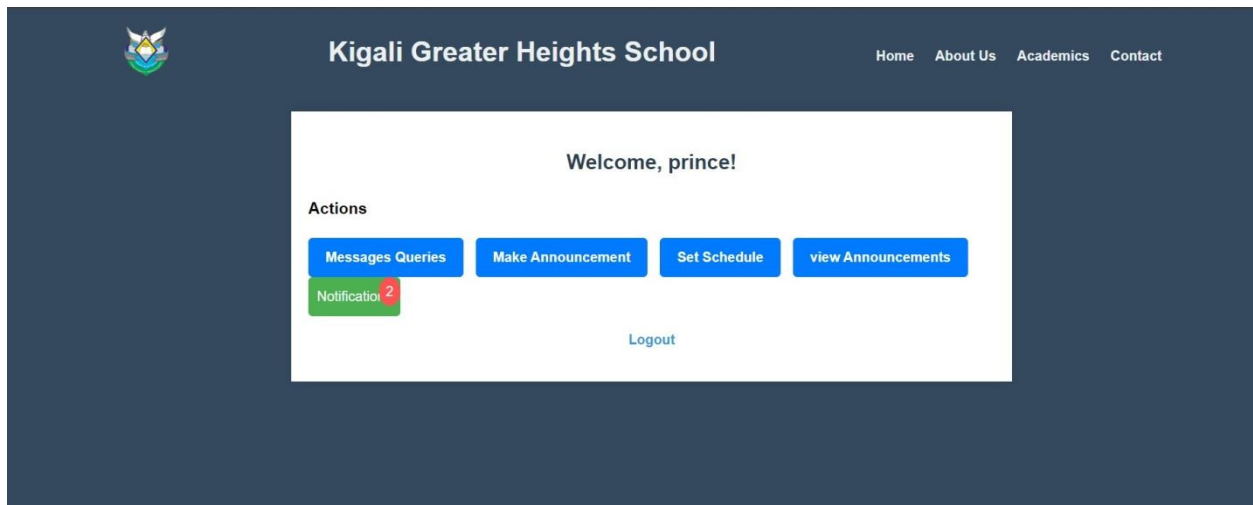
The image shows a registration form titled "Register for KGHS" on a dark blue background. The form is white and contains the following fields: Username (with "mussa" entered), Email, Password (with three dots indicating a masked password), Confirm Password, and Child's Names (comma-separated). A blue "Register" button is at the bottom of the form. Below the button, there is a link: "Already have an account? [Login here.](#)"

Figure 12: Creating User accounts with parent role

## Staff Dashboards

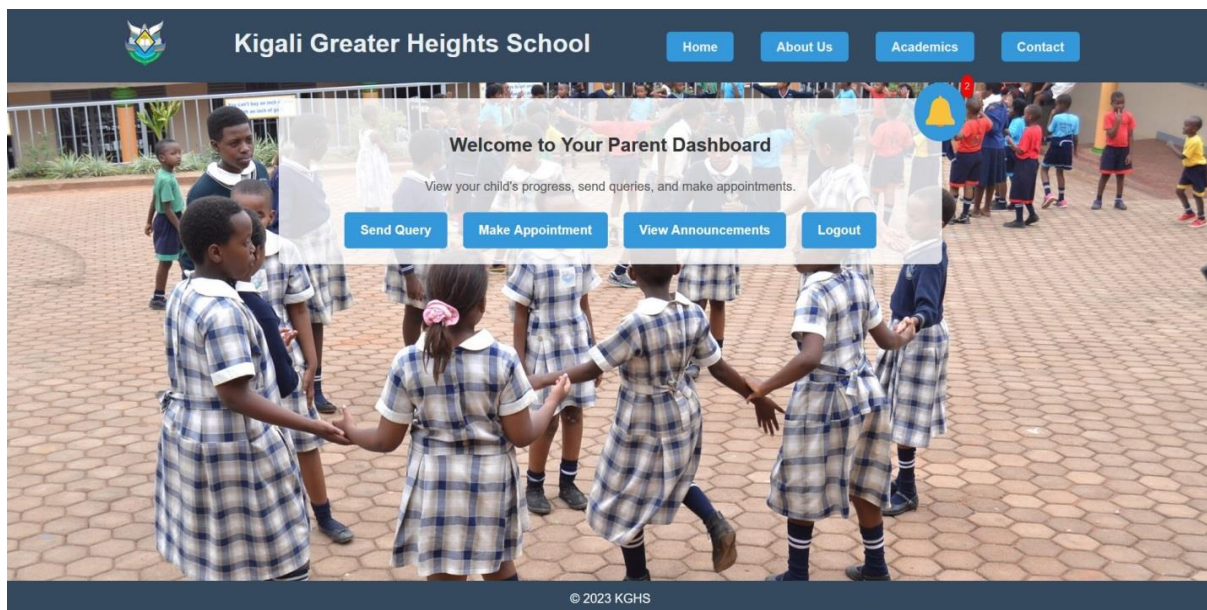


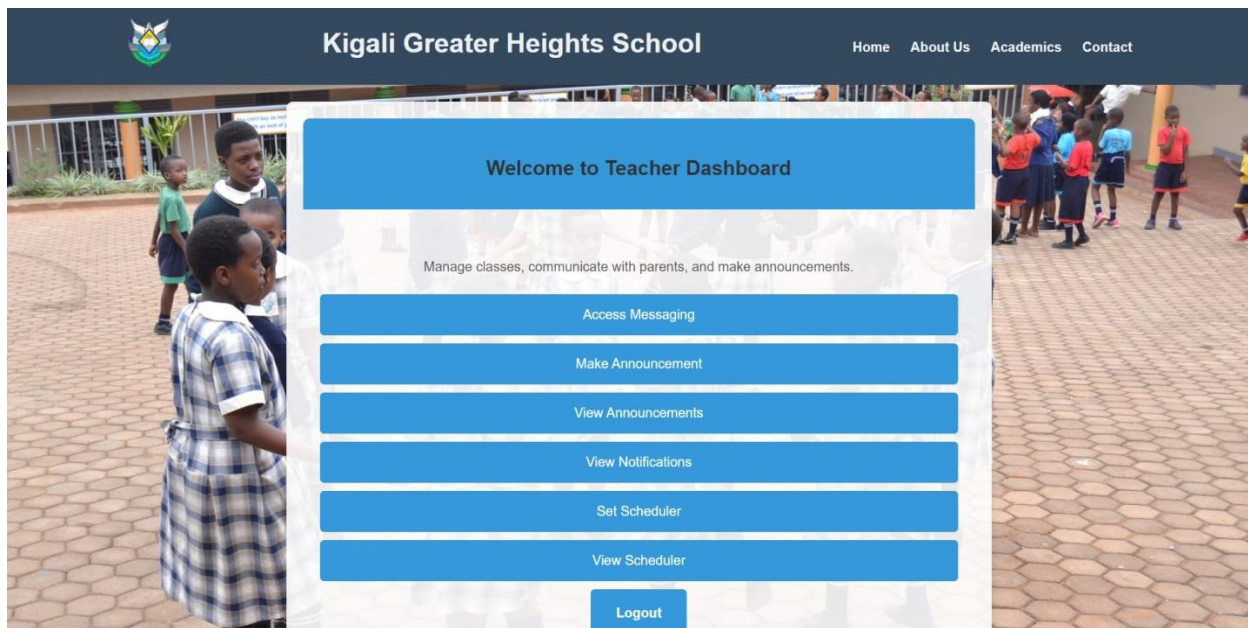




*Figure 13-14: School director & Head Teacher Dashboards*

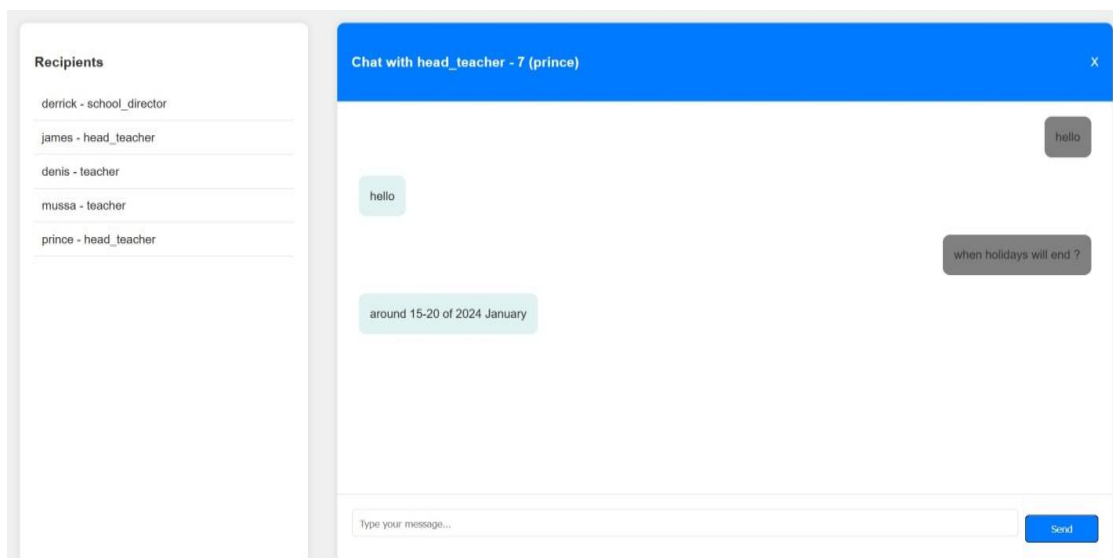
## Parent and Teachers Dashboard





*Figure 15: Parent and Teachers Dashboards*

## Chat Screen Between Teacher and Parent



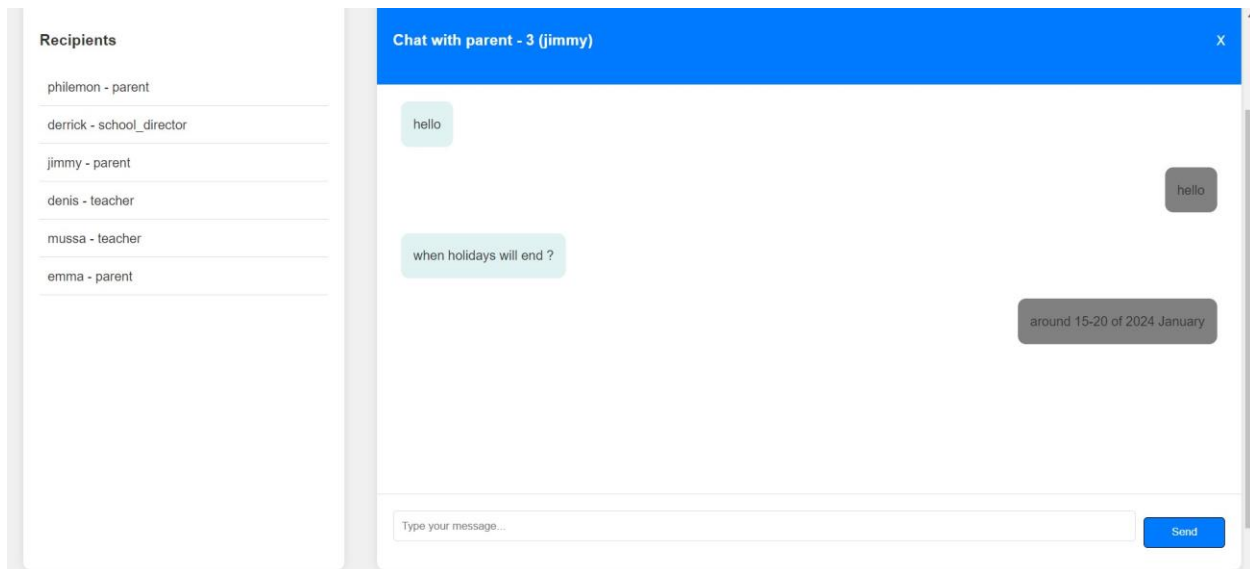


Figure 16-17: chat screen between Teacher and parent

## Manage Users

7	prince	prince@gmail.com	head_teacher		<a href="#">Edit</a>	<a href="#">Delete</a>
8	emma	emma@gmail.com	parent	emmy,diane,erric,james	<a href="#">Edit</a>	<a href="#">Delete</a>

### Add New User

Username:

Email:

Password:

Role:

Child Names (for Parent):

[Add User](#)

Figure 18: Manage Users

## View announcement

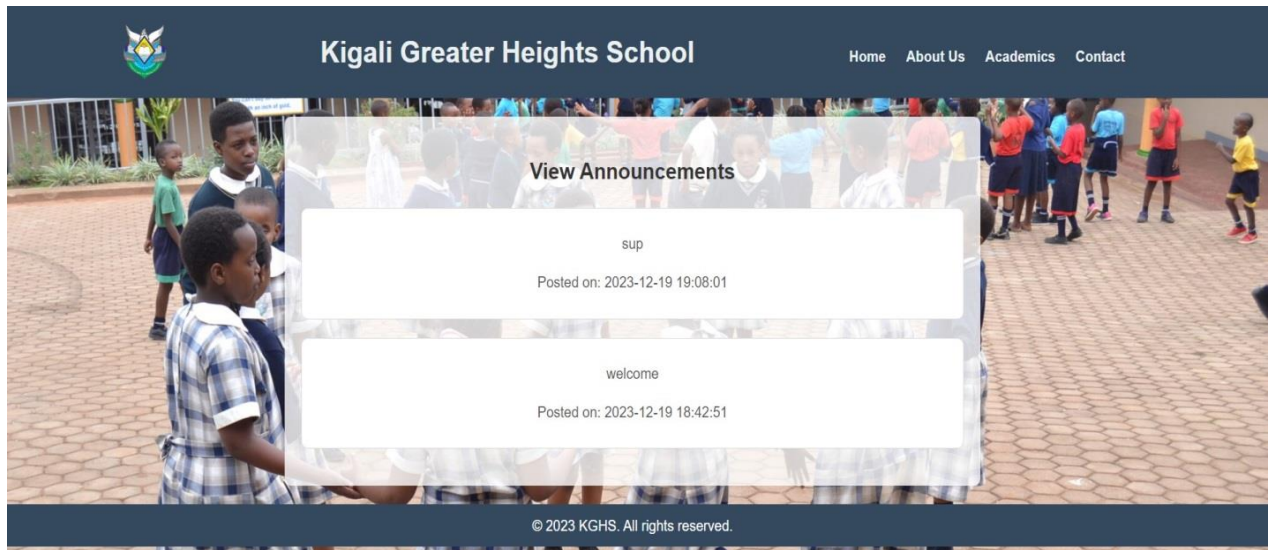


Figure 19: View Announcement

## User Setting there own scheduler



Figure 20: User Setting Scheduler

## requesting and Notifications of Appointments

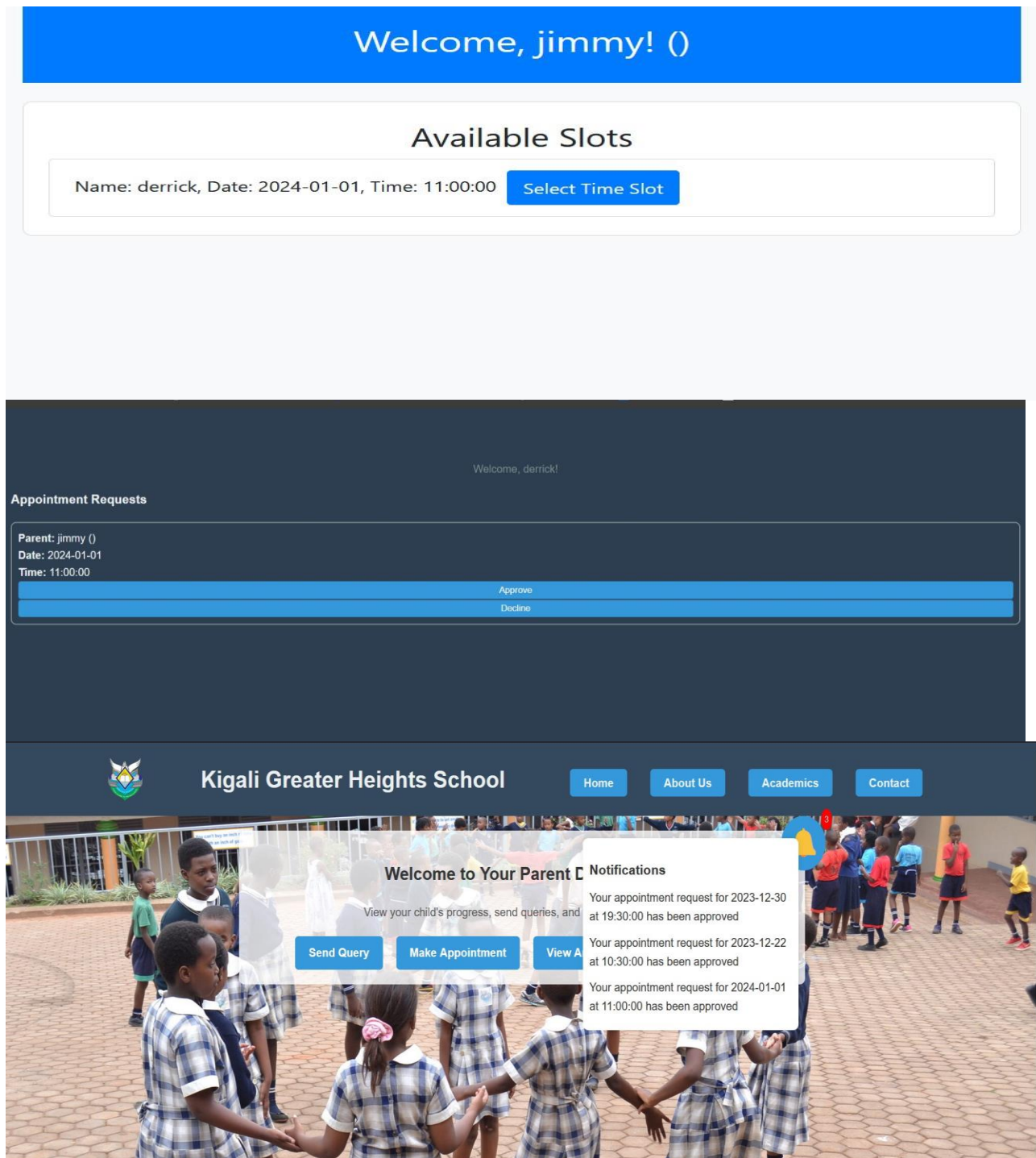


Figure 21-23: requesting and Notifications of Appointments



**Software testing** is an essential practice that involves systematically assessing and validating the functionality of software. Its primary objective is to ensure that the software performs as intended. Testing serves multiple purposes, including the early detection and prevention of bugs, minimizing development expenses, and enhancing overall performance.

They are some key points to check in software testing

- ❖ **Compliance with Requirements:** Ensure that the software meets the specified requirements and objectives that guided its design and development. This involves validating that the software functions as intended and delivers the desired outcomes.
- ❖ **Functional Validity:** Verify that the application works as expected and meets the intended behavior. This includes testing all the functionalities and features of the software to ensure they operate correctly and produce accurate results.
- ❖ **Performance Testing:** Measure and optimize the software's performance by assessing factors such as response time, scalability, and resource usage. This involves evaluating how the software performs under different workloads and ensuring it meets performance expectations, providing a smooth and efficient user experience.

## **Unit Testing**

Unit testing is a crucial aspect of software development that concentrates on validating the smallest testable components of an application, known as units, in isolation to ensure their correct functioning.

Unit testing can be performed manually; however, it is commonly automated to streamline the process and enhance efficiency.

## **Integration testing**

Integration testing is a critical stage in software testing, where individual software modules are integrated and tested collectively as a unified system. This testing phase ensures the correct assembly of various software components and verifies that they function harmoniously. The integration testing process involves a series of tests that progressively combine software and hardware components until the entire system is thoroughly tested. By first conducting individual tests on software components and subsequently integrating them, we can ensure that they seamlessly work together as intended.

## **Validation testing**

The process of assessing software during development or upon completion to ascertain its adherence to specified business requirements is known as validation testing. This type of testing aims to verify that the product effectively fulfills the client's needs. Another definition of validation testing is to demonstrate that the product successfully serves its intended purpose when deployed in the appropriate environment.

## **Hardware and Software Requirements**

### **Client Side Software Requirements**

- Web Browser: The latest versions of widely used browsers such as Mozilla Firefox, Google Chrome, Safari, or Microsoft Edge.

### **Server Side Software Requirements**

- Operating System: Compatible with major operating systems, including Windows 10 and 11, macOS, and popular Linux distributions (e.g., Ubuntu, Fedora).
- PHP Version: Compatible with the latest stable version of PHP (e.g., PHP 5 or higher).

- Database: Support for popular databases like MySQL or PostgreSQL.
- Web Server: Compatibility with web servers such as Apache or Nginx.
- Security Measures: Implementation of appropriate security measures, including secure coding practices, encryption of sensitive data, and protection against common web vulnerabilities (e.g., SQL injection, cross-site scripting).
- Backup and Recovery: Regular backup mechanisms and procedures in place to ensure data integrity and facilitate recovery in case of system failures or data loss.

#### **Client Side Hardware Requirements**

- Microprocessor: Intel Core i3 or higher processor.
- Memory (RAM): Minimum 2 Gigabytes (GB) of RAM.
- Storage Space: At least 1 Gigabyte (GB) of free disk space.

#### **Server Side Hardware Requirements**

- Web Server: Apache HTTP Server or Nginx, compatible with PHP and capable of handling high traffic and concurrent connections efficiently.
- Database: MySQL or MariaDB database server, with a version of at least 5.7.
- Network Infrastructure: A high-speed network card with a minimum bandwidth of 1 Gigabit per second (Gbps).
- RAM: Minimum 4 Gigabytes (GB) of RAM.
- Storage Space: At least 5 Gigabytes (GB) of available hard disk space.



## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATIONS**

#### **Conclusion**

The primary goal of the KGHS School Project was to create and deploy a comprehensive monitoring and management system tailored for KGHS, facilitating efficient oversight and management of various aspects within the school. The system aimed to tackle challenges related to monitoring different facets of the school environment, ensuring timely responses, and avoiding operational shortcomings.

The implemented KGHS School Project boasts the following key features:

1. **Real-time Activity Monitoring:** The system empowers KGHS to monitor various activities in real-time, enabling proactive management and preventing operational hiccups.
2. **Automated Notifications:** Automated alerts and notifications are incorporated, triggering when specific thresholds are reached, guaranteeing timely responses and actions to enhance school operations.
3. **Comprehensive Tracking:** The system effectively tracks various activities within the school premises, providing KGHS with a holistic view of the flow of events and ensuring streamlined management.
4. **Data Analysis for Decision-Making:** The system generates insightful reports and analytics on different aspects of school activities, aiding KGHS in making informed decisions regarding school strategies.

5. Collaborative Communication: The system facilitates seamless communication and collaboration among different stakeholders within the school, enhancing overall coordination.

6. Role-based Access Control: Distinct user roles, including administrators, teachers, and staff, are established with specific access permissions to uphold data security and integrity.

The implementation of this system is anticipated to yield numerous benefits for KGHS, including enhanced oversight, operational efficiency, streamlined communication, and informed decision-making based on real-time data.

The development process involved a thorough exploration of KGHS's operational challenges through methods like observation, interviews, and documentation analysis. Leveraging Unified Modeling Language (UML) for analysis and design ensured a tailored and effective solution aligned with KGHS's specific requirements.

In conclusion, the KGHS School Project was successfully implemented, addressing identified challenges and providing a robust solution for efficient school management. The system offers an intuitive interface, streamlines workflows, extracts meaningful insights from the database, and presents results in a visually engaging manner.

## **Recommendations**

Recommendations for optimizing KGHS School Project's management system:

1. **Implementation:** KGHS should implement the school project management system to address operational challenges and enhance overall management efficiency.
2. **Continuous Enhancement:** Regularly update and improve the system based on user feedback to ensure its continued relevance and effectiveness.
3. **Collaboration:** Foster collaboration with other educational institutions to establish shared standards and best practices for school project management.
4. **Data Analysis:** Leverage the system's analytical capabilities for insightful reporting and data analysis to inform decisions related to school strategies and operations.
5. **Training and Support:** Offer comprehensive training and support to users of the system within the school community, ensuring a seamless integration and maximizing its benefits.

By adhering to these recommendations, KGHS can effectively manage various aspects of school operations, streamline processes, and contribute to the overall success of the educational institution.

## REFERENCES

### Books

- Alan, D. (2012). *System Analysis and Design*. Indiana: John Wiley & Sons, Inc.
- ARIADNE TRAINING CORPORATE. (2001). *UML Applied-Object oriented analysis and design using the UML*. Somerset UK: Ariadne: ARIADNE TRAINING CORPORATE.
- Bell&Douglas. (2005). *Software Engineering for Students, a Programming Approach, Fourth Edition*. USA: Bell.: Bell&Douglas.
- Haley, W. (2011). *System Analysis and Design, Fifth Edition*. USA: Don Fowley: Wixom&Barbara Haley.
- Kraemer. (2009). Automated encapsulation of UML activities for incremental development and verification. In *UML*. USA.
- Rumbaugh, J. (2006). *Unified Modeling Language*. USA: Rational software Library.

### Websites

- dotnetfreakblog. (2019, march 20). *Concept of Dependency, Generalization, Association, Aggregation, Composition in Object Oriented Programming*. Retrieved from dotnetfreakblog.wordpress: <https://dotnetfreakblog.wordpress.com/2014/01/11/concept-of-dependency-generalization-association-aggregation-composition-in-object-oriented-programming>
- EDRAWSOFT. (2020). *DATA DICTIONARY SYMBOLS*. Retrieved from EDRAWSOFT: <HTTPS://WWW.EDRAWSOFT.COM/DATA-DICTIONARY-SYMBOLS.HTML>
- Investopedia. (2019, april 26). *Investopedia*. Retrieved from Investopedia: <https://www.investopedia.com/terms/a/atm.asp>
- Joel, M. (2015). *MYSQL 2nd Edition*. Retrieved from pdfdrive: <https://www.pdfdrive.com/murachs-mysql-d176103509.html>

Kraemer. (2009). Automated encapsulation of UML activities for incremental development and verification. In *UML*. USA.

Patrick, G. H. (2005). *uml-20-in-action-a-project-based tutorial a detailed andpractical walk through showing how to apply uml to real world development*. Retrieved from pdfdrive: <https://www.pdfdrive.com/uml-20-in-action-a-project-based-tutorial-a-detailed-andpractical-walk-through-showing-how-to-apply-uml-to-real-world-development-projectsdl58626574.html>

techopedia. (2019, july 2). *Integration Testing*. Retrieved from techopedia: <https://www.techopedia.com/definition/7751/integration-testing>

## **APPENDICES**

## **Curriculum Vitae**