SmartCustodiX

A Web-Based Property Management System for Lamhako Integrated School”

RICKY V. CATALUÑA

MOBILE COMPUTING

Platform Used: HTML, CSS, Bootstrap, JavaScript, Json, PHP, MYSQL (Apache)

MASTER IN INFORMATION TECHNOLOGY

July 2025

Chapter I

INTRODUCTION

1. Introduction

1.1 Background of the Study

Lamhako Integrated School, situated in the rural municipality of Tboli, South Cotabato, has served as a vital center for basic education since 1979. As it transitioned into an integrated school in 2011 and began offering Junior and Senior High School programs in 2016, the institution experienced steady growth in its population and operational complexity. This development introduced increased logistical challenges, particularly in managing school properties, instructional materials, and equipment.

The specific dilemma lies in the continued reliance on manual systems—spreadsheets, printed forms, and file cabinets—for recording, approving, and monitoring property requisitions and inventory. This outdated system has led to inefficiencies such as inaccurate records, delayed approvals, inconsistent tracking of borrowed items, and difficulty in identifying low stock levels. Despite the critical role of resource availability in sustaining daily school operations, the problem remains unresolved due to limited access to modern information systems, lack of technical expertise, and infrastructure constraints typical of rural public schools.

To resolve this issue, this research introduces SmartCustodix, a role-based, online web-powered custodial management system. It aims to automate the requisition workflow, provide real-time inventory monitoring, generate alert notifications for low stock, and manage borrowing and returns of school properties efficiently. By utilizing SDLC with an Agile methodology, this project ensures incremental development and continuous user feedback, making the system adaptable and aligned with actual school processes.

The hypothesis of this study is that implementing a customized, online web-based custodial management system will significantly improve the efficiency, accuracy, and accountability of property management at Lamhako Integrated School, compared to the existing manual system.

1.2 Problem Statement

Lamhako Integrated School continues to rely on manual processes—such as Excel spreadsheets and physical documents—for handling property custodial tasks like requisition, approval, inventory tracking, and borrowing of equipment. This traditional approach has resulted in inefficiencies, such as delayed approvals, inaccurate inventory records, untracked borrowed items, and poor stock monitoring. As the school population grows and demands on resources increase, these issues become more critical, leading to operational delays and resource mismanagement. Despite these challenges, no centralized, automated, and accessible system is currently in place to support efficient property management in the school.

1.3 Objectives

1.3.1 General Objective

To develop and implement an online, role-based custodial management system that automates and streamlines the requisition, approval, inventory monitoring, and borrowing processes at Lamhako Integrated School to improve efficiency, accuracy, and accountability in property management.

1.3.2 Specific Objectives

1. To design a user-friendly online interface for teaching and non-teaching staff to submit and monitor requisition requests.
2. To develop an approval module for the School Head that allows secure, real-time decision-making based on budget and availability.
3. To implement a centralized inventory monitoring system accessible via the internet for the Property Custodian.
4. To include an automated alert feature that notifies the custodian when stock levels reach critical thresholds.
5. To provide a borrowing and return tracking module that ensures accountability and prevents item loss.
6. To generate comprehensive reports on inventory, requisitions, and borrowed items to support decision-making and transparency.

1.4 Scope and Limitations

1.4.1 Scope

This study focuses on the design, development, and implementation of SmartCustodix, an online, role-based property custodial management system for Lamhako Integrated School. The system is developed to digitize and automate the following key custodial processes:

1. **Requisition Submission and Approval** – Teaching and non-teaching staff can submit requests online, and the School Head can approve or reject based on available resources and budget.
2. **Inventory Monitoring** – The Property Custodian can manage and update inventory in real time, with system-generated alerts for low stock levels.
3. **Borrowing and Return Management** – Users can borrow equipment, and the system tracks borrow and return dates to ensure accountability.

4. **Reporting** – The system generates reports related to requisitions, inventory status, borrowing logs, and stock usage trends.
5. **User Roles and Access** – The system has three roles: (1) Requesters (teachers and non-teaching staff), (2) Approver (School Head), and (3) Custodian (inventory manager).

The system is developed using SDLC with Agile methodology, hosted on a web server, and accessible via internet-connected devices. It is intended for internal use within the school community, excluding students.

1.4.2 Limitations

1. **No Student Access** – The system is exclusively available to teaching and non-teaching personnel. Students cannot access or make requisitions.
2. **Limited to Property Custodianship** – The system focuses solely on managing supplies and equipment; it does not cover other school administrative systems such as HR, finance, or academic records.
3. **Initial Infrastructure Dependence** – The system assumes availability of stable internet and devices, which may pose challenges in case of technical issues or limited connectivity.
4. **No External Integration** – The system is not integrated with government platforms, accounting software, or other school management systems during its initial phase.
5. **Basic Financial Control** – While it considers budget limits in the approval process, it does not include full financial accounting or procurement modules.

1.5 Significance of the Study

This study is significant for the following stakeholders:

1. **School Administration**: Enables better planning, resource allocation, and operational efficiency through automated reports and real-time data access.
2. **Property Custodian:** Reduces manual workload, improves stock visibility, and streamlines borrowing processes, enhancing accuracy and accountability.
3. **Teaching and Non-Teaching Staff**: Simplifies requisition procedures and provides transparency in request status, improving satisfaction and productivity.
4. **Department of Education and Policymakers**: Demonstrates a scalable and replicable digital solution for school property management, especially useful in rural or underserved areas.

5. **Future Researchers**: Serves as a practical reference for studies involving web-based administrative systems in education settings.

Chapter II

Review of Related System/Literature

2. Review of Related System/Literature

2.1 Brief Discussion of the Current System

At Lamhako Integrated School, property custodianship is still handled through manual means such as Excel spreadsheets, printed forms, and handwritten logs. Requisition requests are manually filed by teachers and non-teaching staff, approved through physical signatures by the School Head, and tracked by the Property Custodian using static records. This process is time-consuming, prone to delays, and often leads to discrepancies in stock levels and unrecorded borrowing. Additionally, the absence of an automated alert mechanism for low supplies makes it difficult for the school to replenish inventory in a timely manner. As the school's operations continue to grow, the inefficiencies of this system hinder administrative productivity and accountability.Several studies affirm that organizations relying on manual inventory processes face challenges in accuracy, data tracking, and real-time monitoring. According to Oladele et al. (2021), traditional inventory systems lack automation and fail to support decision-making in fast-paced environments. Their study highlighted the benefits of integrating rule-based engines and pattern detection in inventory systems to predict supply needs and streamline restocking.

2.2 Comparison with the Proposed System (SmartCustodix)

The proposed system, SmartCustodix, addresses the limitations of the manual system by offering an online, role-based web application for custodial management. Unlike traditional manual systems, SmartCustodix enables users (i.e., teaching and non-teaching staff, School Head, and Property Custodian) to access the platform from any internet-connected device. It supports secure login, online requisition requests, approval routing, real-time inventory tracking, borrowing and return logs, and system-generated low-stock alerts.Compared to traditional LAN-based inventory systems which are often limited to internal networks, SmartCustodix offers enhanced accessibility and flexibility. This aligns with findings by Utomo et al. (2020), who demonstrated that web-based inventory platforms provide broader access, centralized data, and easier collaboration among users. Their study emphasized how online systems outperform local applications in ensuring data availability, especially in multi-user settings.Furthermore, SmartCustodix was developed using the Software Development Life Cycle (SDLC) in combination with Agile methodology, allowing for iterative prototyping, stakeholder feedback, and continuous refinement. This structured yet flexible development approach ensures the system remains user-centered and adaptable to the evolving needs of the school community, consistent with recommendations by Pressman (2014) and Schwalbe (2015) on best practices in software engineering.In summary, while the current manual system is limited in speed, accuracy, and transparency, SmartCustodix offers a comprehensive, scalable, and accessible solution to modernize custodial tasks in schools. It automates core processes, enhances recordkeeping, and strengthens accountability, making it a relevant and practical response to the administrative challenges in rural public-school settings like Lamhako Integrated School.

Chapter III

Methodology

3. Methodology

3.1 Tools and Technologies Used

To develop the SmartCustodix web-based property custodial management system, the following programming tools, languages, and environments were utilized:

- PHP – Used as the primary server-side scripting language for developing dynamic web pages and connecting to the backend database.
- JavaScript – Used for enhancing user interaction, input validation, and client-side scripting.
- JSON (JavaScript Object Notation) – Enabled efficient data exchange between client and server, particularly for asynchronous operations and dynamic UI updates.
- HTML (Hypertext Markup Language) – Provided the structural framework for all web interfaces.
- CSS (Cascading Style Sheets) – Used for designing the system's user interface to ensure a responsive and intuitive user experience.
- Bootstrap – A front-end framework that accelerated UI development through pre-designed responsive components.
- Visual Studio Code (VS Code) – The integrated development environment (IDE) used for writing and managing the project's codebase.
- Navicat – A database administration tool used for managing, designing, and monitoring the MySQL database.
- MySQL – Served as the relational database management system (RDBMS) for storing, retrieving, and managing all custodial data.
- Apache – Acted as the web server for hosting and running the application locally and online.
- XAMPP – Used as the local development environment during testing stages.
- Draw.io / Lucidchart– Used to create visual documentation such as DFDs, ERDs, and UML diagrams.
-

3.2 System Architecture and Design Documentation

The system architecture of SmartCustodix is based on a client-server model hosted on a web server. It consists of three layers:

- Presentation Layer – The user interface (UI) built using HTML, CSS, and JavaScript accessed via a web browser.
- Application Layer – Business logic written in PHP, handling user inputs, processing transactions, and communicating with the database.
- Data Layer – A MySQL database that stores all records including user accounts, inventory items, requisitions, borrowing logs, and system logs.

Design Artifacts Included in the Study:

- Context Diagram – Shows the entire system's interaction with its external users: requesters, School Head, and Property Custodian.

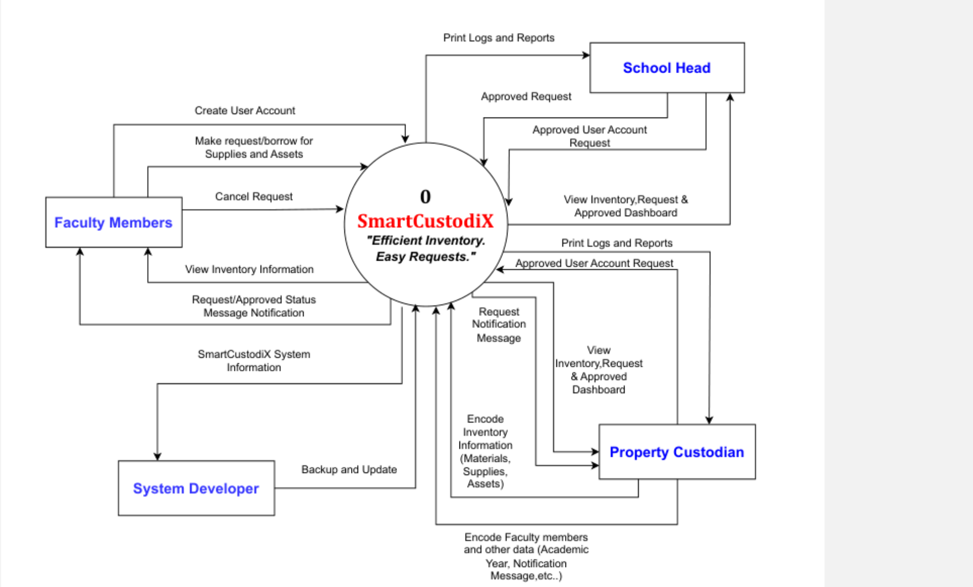

Print Logs and Reports
School Head
Approved Request
Create User Account
Approved User Account Request
Make request/borrow for Supplies and Assets
0
SmartCustodiX
"Efficient Inventory. Easy Requests."
Cancel Request
Faculty Members
View Inventory,Request & Approved Dashboard
Print Logs and Reports
Approved User Account Request
View Inventory Information
Request/Approved Status Message Notification
Request Notification Message
SmartCustodiX System Information
View Inventory,Request & Approved Dashboard
Encode Inventory Information (Materials, Supplies, Assets)
Property Custodian
System Developer
Backup and Update
Encode Faculty members and other data (Academic Year, Notification Message,etc..)

Figure 1. Context Diagram of SmartCustodiX

- Data Flow Diagram (DFD) – Represents the flow of information and processes within the system from Level 0 to Level 2.

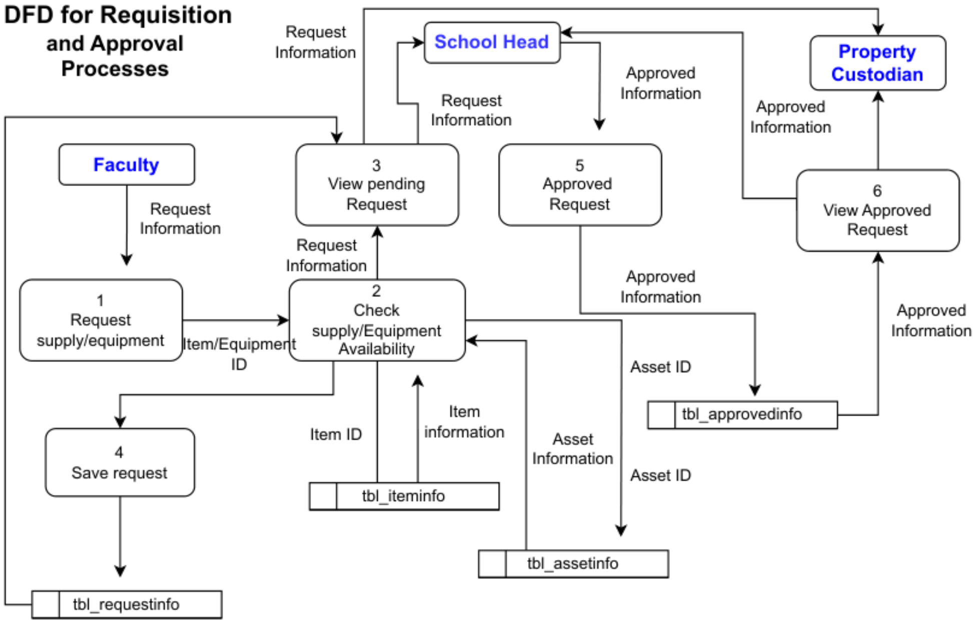


Figure 2. Data Flow Diagram for Requisition and Approval Processes

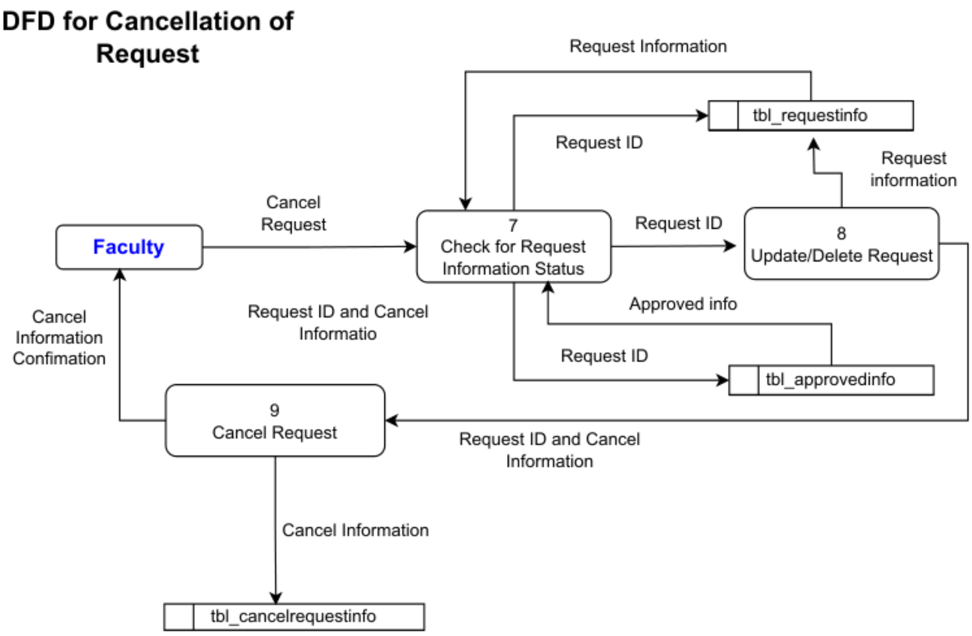


Figure 3. Data Flow Diagram for Cancellation of Request

- Entity Relationship Diagram (ERD) – Defines the database schema, showing relationships among users, requisitions, inventory, and borrowing records.

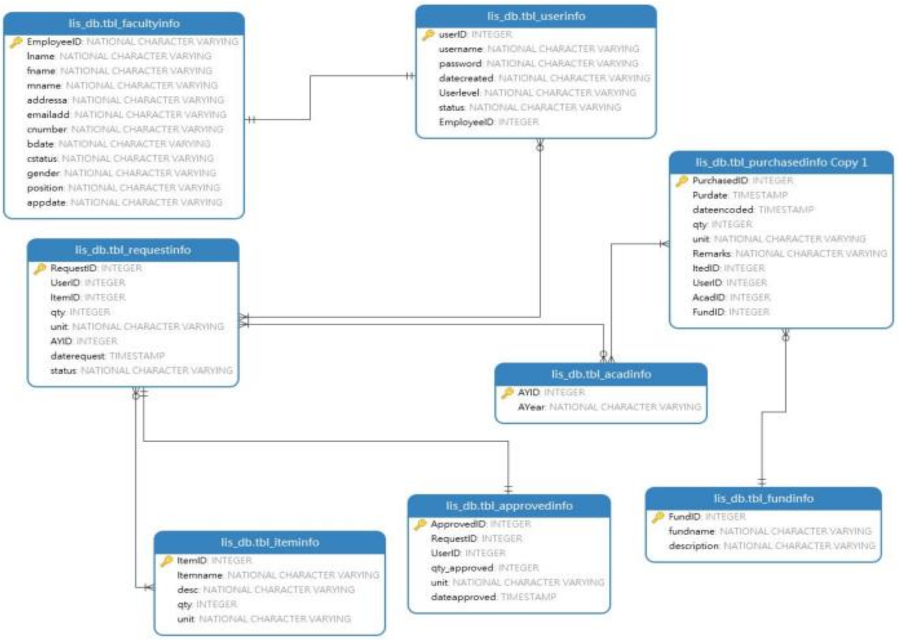

lis_db.tbl_facultyinfo
EmployeeID: NATIONAL CHARACTER VARYING
lname: NATIONAL CHARACTER VARYING
fname: NATIONAL CHARACTER VARYING
mname: NATIONAL CHARACTER VARYING
addressa: NATIONAL CHARACTER VARYING
emailadd: NATIONAL CHARACTER VARYING
cnumber: NATIONAL CHARACTER VARYING
bdate: NATIONAL CHARACTER VARYING
cstatus: NATIONAL CHARACTER VARYING
gender: NATIONAL CHARACTER VARYING
position: NATIONAL CHARACTER VARYING
appdate: NATIONAL CHARACTER VARYING
lis_db.tbl_userinfo
userID: INTEGER
username: NATIONAL CHARACTER VARYING
password: NATIONAL CHARACTER VARYING
datecreated: NATIONAL CHARACTER VARYING
Userlevel: NATIONAL CHARACTER VARYING
status: NATIONAL CHARACTER VARYING
EmployeeID: INTEGER
lis_db.tbl_purchasedinfo Copy 1
PurchasedID: INTEGER
Purdate: TIMESTAMP
dateencoded: TIMESTAMP
qty: INTEGER
unit: NATIONAL CHARACTER VARYING
Remarks: NATIONAL CHARACTER VARYING
ItedID: INTEGER
UserID: INTEGER
AcadID: INTEGER
FundID: INTEGER
lis_db.tbl_requestinfo
RequestID: INTEGER
UserID: INTEGER
ItemID: INTEGER
qty: INTEGER
unit: NATIONAL CHARACTER VARYING
AYID: INTEGER
daterequest: TIMESTAMP
status: NATIONAL CHARACTER VARYING
lis_db.tbl_acadinfo
AYID: INTEGER
AYear: NATIONAL CHARACTER VARYING
lis_db.tbl_approvedinfo
ApprovedID: INTEGER
RequestID: INTEGER
UserID: INTEGER
qty_approved: INTEGER
unit: NATIONAL CHARACTER VARYING
dateapproved: TIMESTAMP
lis_db.tbl_fundinfo
FundID: INTEGER
fundname: NATIONAL CHARACTER VARYING
description: NATIONAL CHARACTER VARYING
lis_db.tbl_iteminfo
ItemID: INTEGER
Itemname: NATIONAL CHARACTER VARYING
desc: NATIONAL CHARACTER VARYING
qty: INTEGER
unit: NATIONAL CHARACTER VARYING

Figure 5. Entity Relationship Diagram

- Unified Modeling Language (UML) Diagrams – Include Use Case, Class Diagram, and Sequence Diagrams that detail system functionality and interactions.

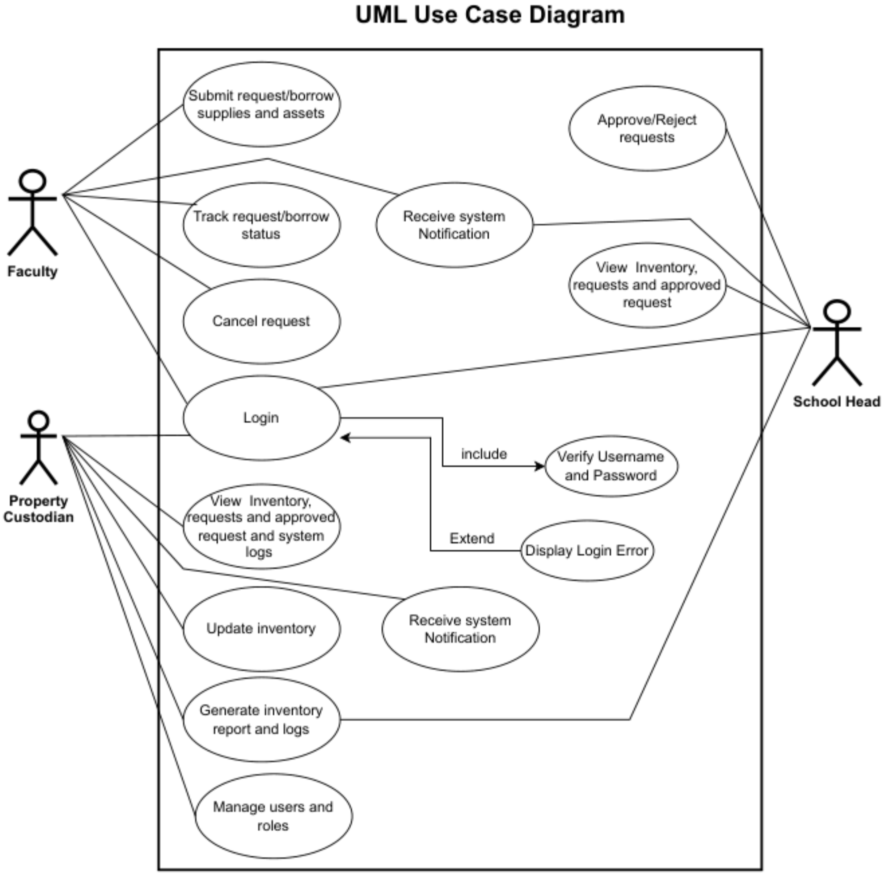

UML Use Case Diagram
Faculty
Property Custodian
School Head
Submit request/borrow supplies and assets
Track request/borrow status
Cancel request
Login
View Inventory, requests and approved request and system logs
Update inventory
Generate inventory report and logs
Manage users and roles
Receive system Notification
Approve/Reject requests
View Inventory, requests and approved request
include
Verify Username and Password
Extend
Display Login Error
Receive system Notification

Figure 5. UML (Use Case Diagram)

3.3 System Flowchart and Development Process

3.3.1 System Flowchart

The system flowchart illustrates the operational flow, including:

- Login process and user validation
- Requisition submission by staff
- Approval process by the School Head
- Inventory update and tracking by the Property Custodian
- Borrowing and returning of items
- Alert generation for low-stock thresholds
- Each user role follows a distinct path through the system, ensuring task segregation and accountability.

3.3.2 Development Process

The system development followed a combination of the Software Development Life Cycle (SDLC) using the Waterfall Model and Agile methodology for flexibility and continuous improvement.

- Planning Phase (Waterfall): Identified the scope, roles, and system objectives through stakeholder interviews.

- Analysis Phase (Waterfall): Gathered functional and non-functional requirements; documented current problems.
- Design Phase (Waterfall): Designed system architecture and created visual models (DFD, ERD, UML).
- Development Phase (Agile): Followed iterative sprints to develop core modules such as login, inventory management, requisition workflows, and notifications. Feedback from users was regularly integrated.
- Testing Phase (Agile): Performed unit testing, integration testing, and user acceptance testing (UAT) with real school data.
- Deployment Phase (Waterfall): Hosted the system online for real-time access by school personnel.
- Maintenance Phase (Agile): Continues to collect user feedback and apply updates to improve usability and performance.

This hybrid approach allowed structured progression while maintaining flexibility for feature enhancements during actual use.

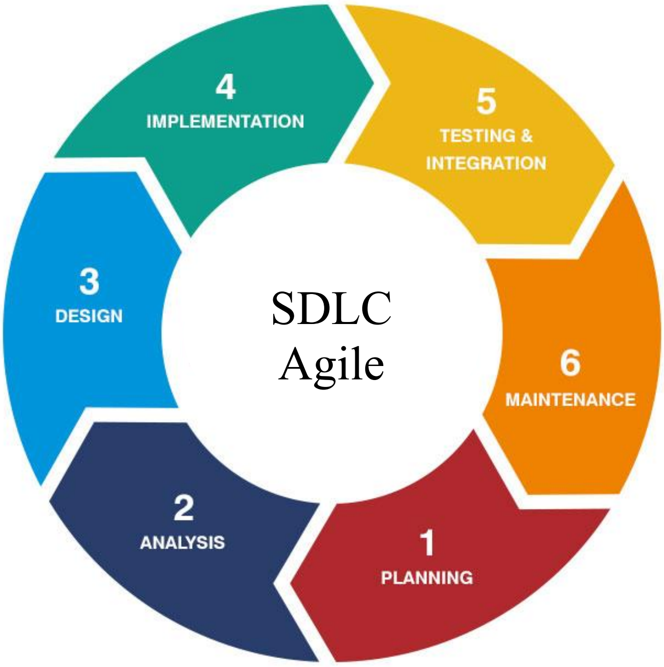


Figure 6. System Development Life Cycle

Chapter IV

System Design and Implementation

4. Design and Implementation

4.1 Screenshots of User Interface

The implemented system, SmartCustodix, delivered the following functionalities:

1. **Requisition Module:** Enabled users to submit requests via web forms, significantly reducing paperwork and approval delays.
2. **Approval Workflow:** Allowed the School Head to review, approve, or reject requests instantly.
3. **Inventory Module:** Provided real-time stock monitoring, item history, and automated stock alerts.
4. **Borrowing Tracker:** Logged borrowed items with return dates, improving accountability.
5. **Notification System:** Sent alerts for critical stock levels.
6. **Reports Dashboard:** Generated downloadable reports for requisitions, inventory, and borrowing.

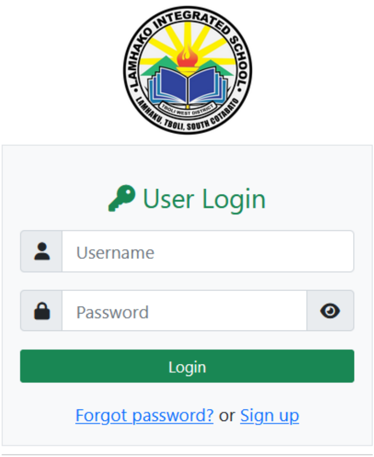


Figure 6. User Login

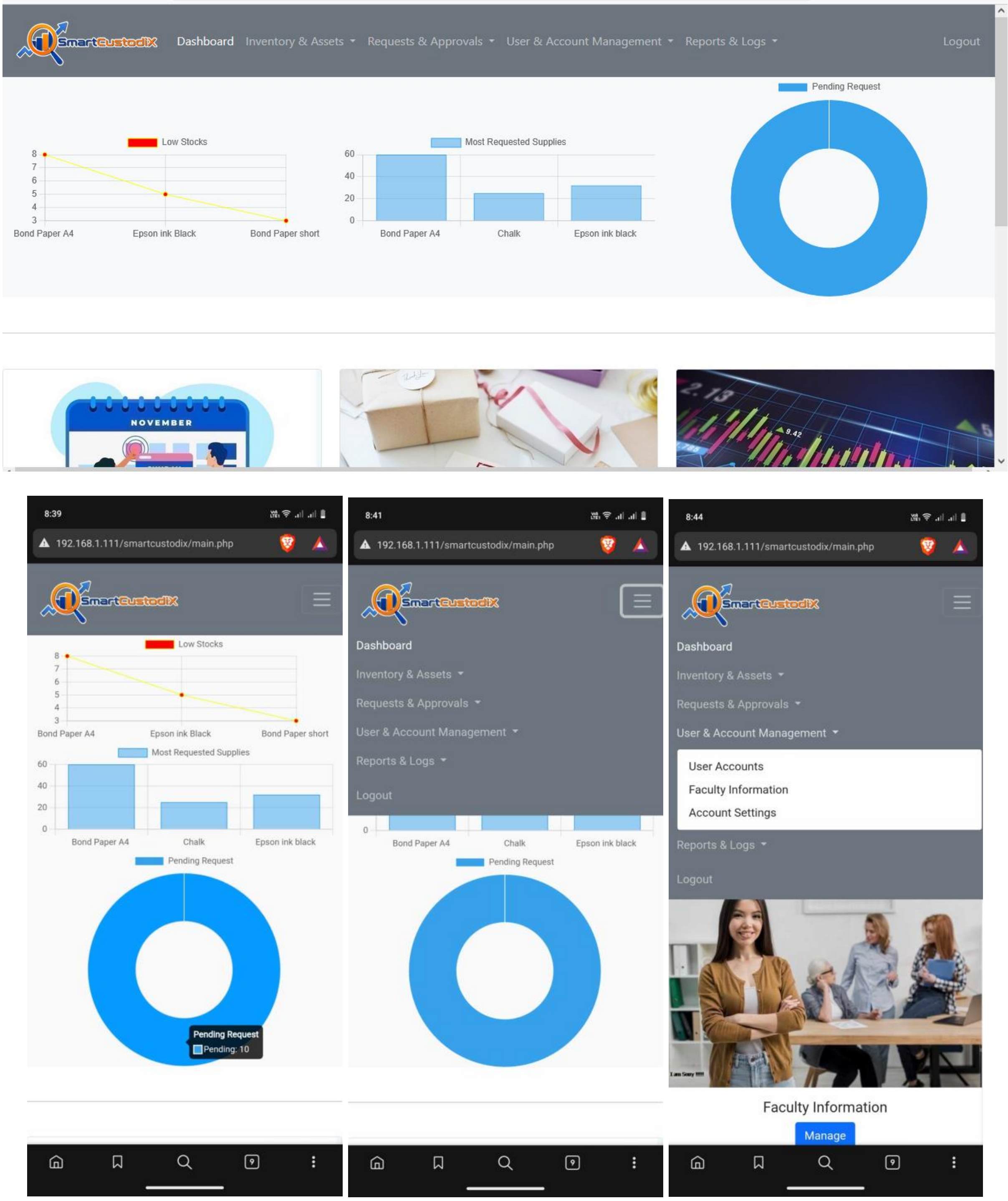
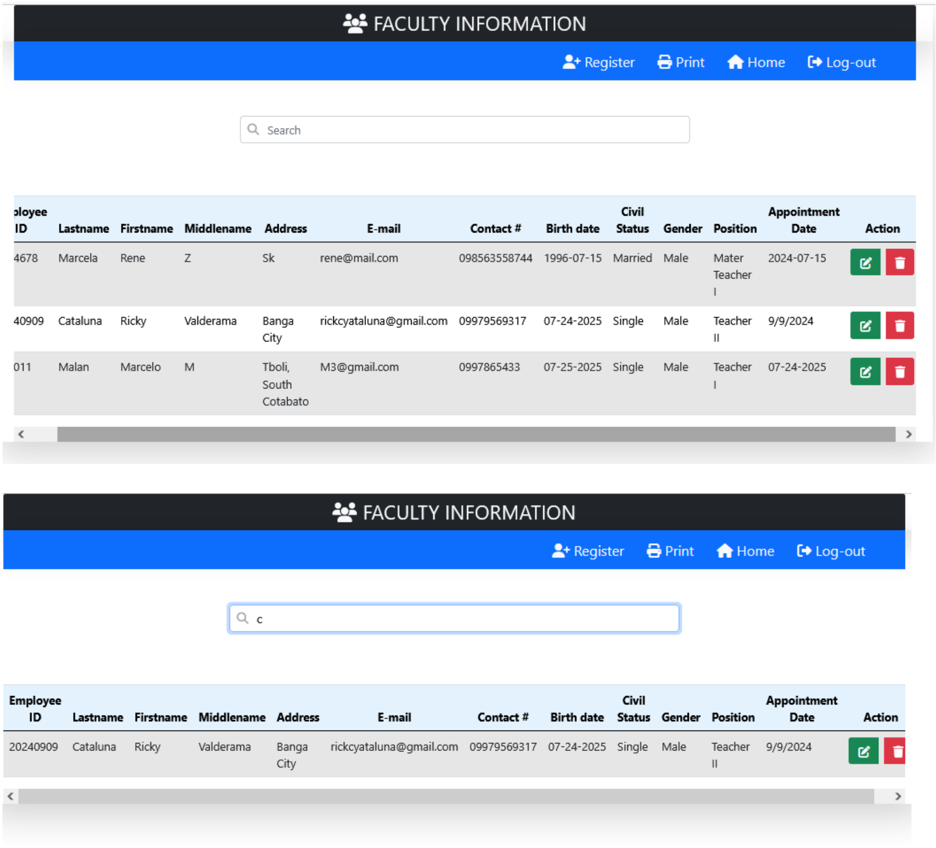


Figure 7. Property Custodian Panel (Dashboard and Navigation Bar)



ployee ID	Lastname	Firstname	Middlename	Address	E-mail	Contact #	Birth date	Civil Status	Gender	Position	Appointment Date	Action
4678	Marcela	Rene	Z	Sk	rene@mail.com	098563558744	1996-07-15	Married	Male	Mater Teacher I	2024-07-15	
40909	Cataluna	Ricky	Valderama	Banga City	rickcyataluna@gmail.com	09979569317	07-24-2025	Single	Male	Teacher II	9/9/2024	
011	Malan	Marcelo	M	Tboli, South Cotabato	M3@gmail.com	0997865433	07-25-2025	Single	Male	Teacher I	07-24-2025	

Employee ID	Lastname	Firstname	Middlename	Address	E-mail	Contact #	Birth date	Civil Status	Gender	Position	Appointment Date	Action
20240909	Cataluna	Ricky	Valderama	Banga City	rickcyataluna@gmail.com	09979569317	07-24-2025	Single	Male	Teacher II	9/9/2024	

Figure 8. Faculty Information

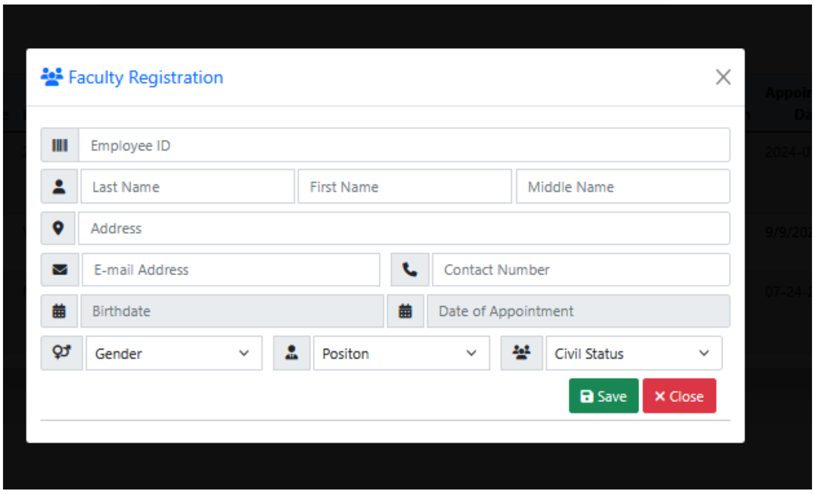
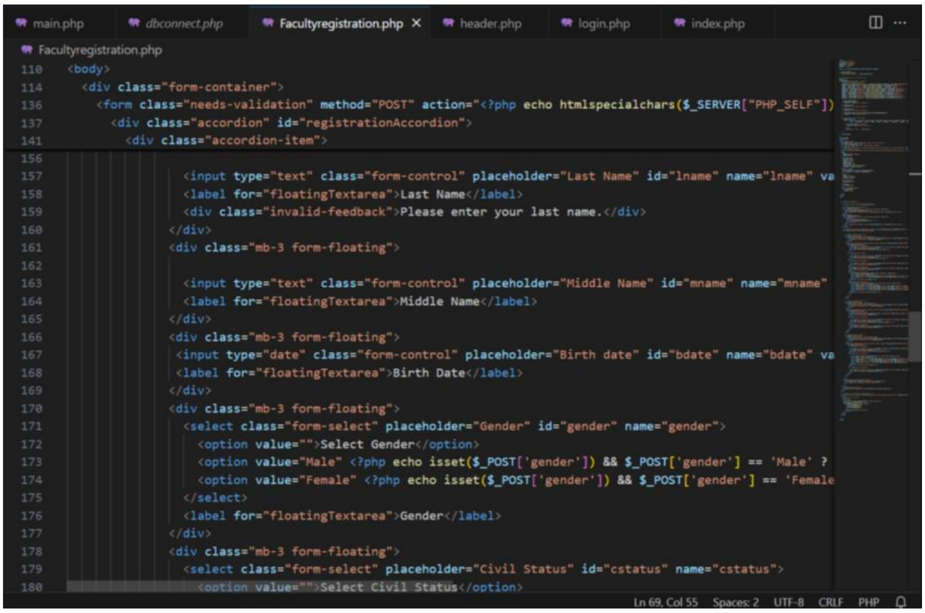
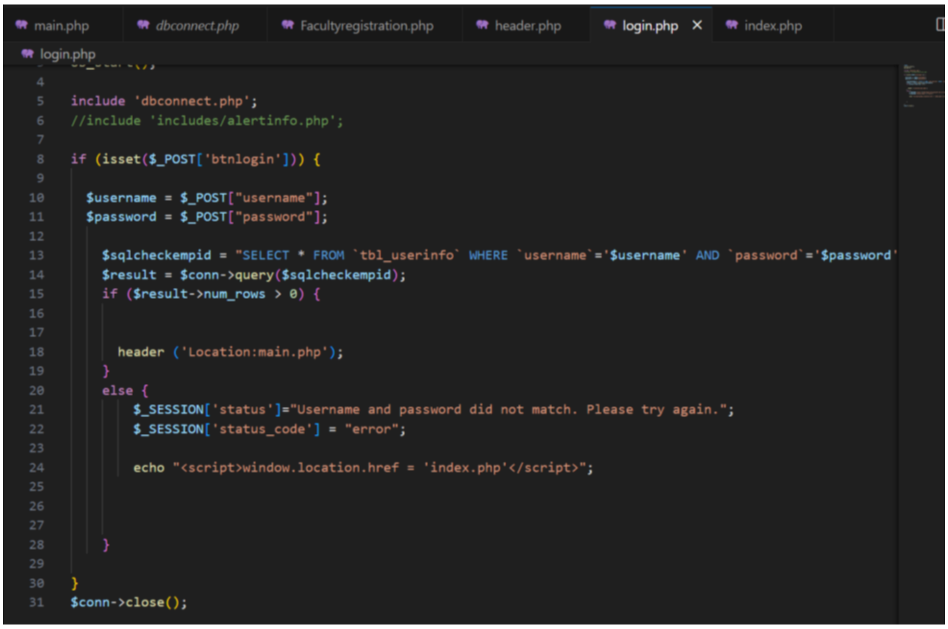


Figure 9. Faculty Registration Form

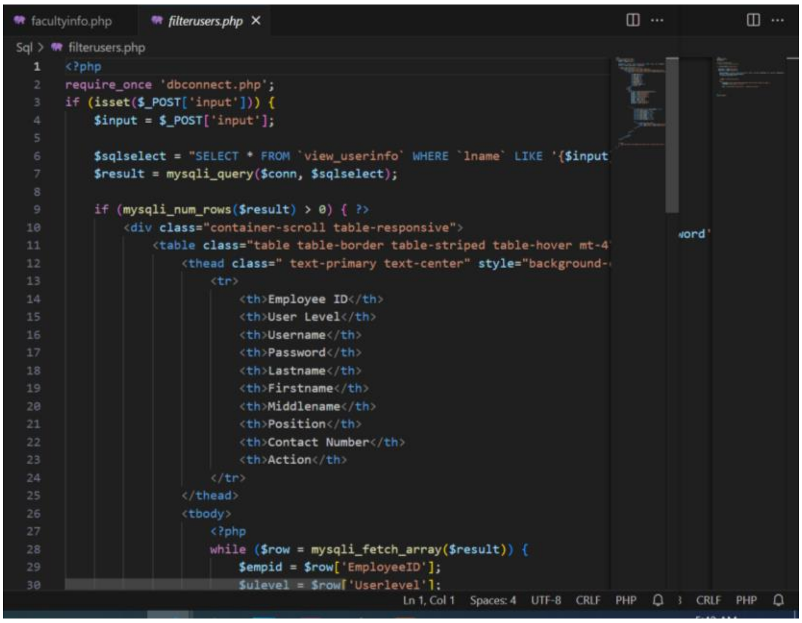
4.2 Code Snippets



```
<body>
  <div class="form-container">
    <form class="needs-validation" method="POST" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"])
      <div class="accordion" id="registrationAccordion">
        <div class="accordion-item">

                <input type="text" class="form-control" placeholder="Last Name" id="lname" name="lname" va
                <label for="floatingTextarea">Last Name</label>
                <div class="invalid-feedback">Please enter your last name.</div>
              </div>
              <div class="mb-3 form-floating">

                <input type="text" class="form-control" placeholder="Middle Name" id="mname" name="mname"
                <label for="floatingTextarea">Middle Name</label>
              </div>
              <div class="mb-3 form-floating">
               <input type="date" class="form-control" placeholder="Birth date" id="bdate" name="bdate" va
               <label for="floatingTextarea">Birth Date</label>
              </div>
              <div class="mb-3 form-floating">
                <select class="form-select" placeholder="Gender" id="gender" name="gender">
                  <option value="">Select Gender</option>
                  <option value="Male" <?php echo isset($_POST['gender']) && $_POST['gender'] == 'Male' ?
                  <option value="Female" <?php echo isset($_POST['gender']) && $_POST['gender'] == 'Female
                </select>
                <label for="floatingTextarea">Gender</label>
              </div>
              <div class="mb-3 form-floating">
                <select class="form-select" placeholder="Civil Status" id="cstatus" name="cstatus">
                  <option value="">Select Civil Status</option>
```

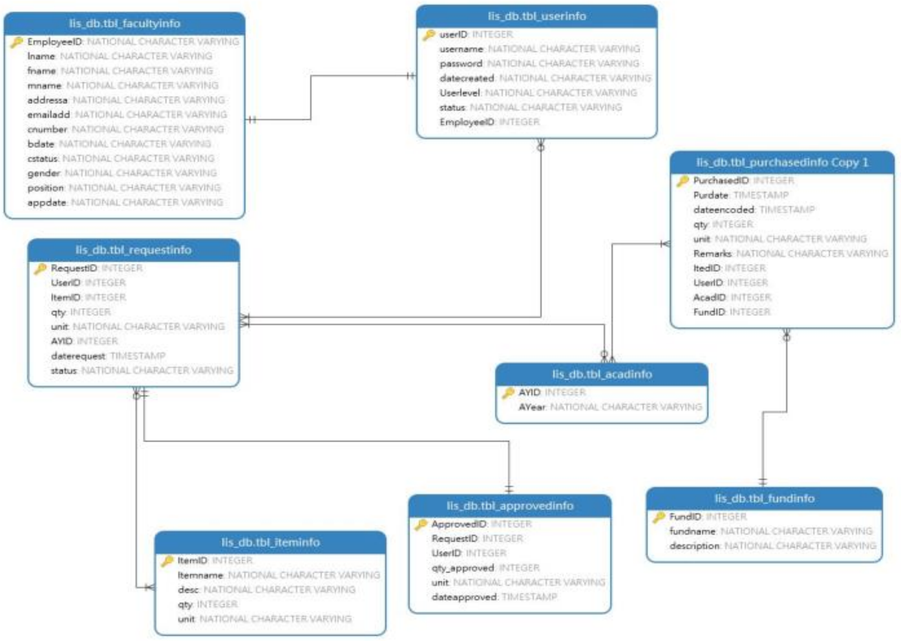


```
ob_start();

include 'dbconnect.php';
//include 'includes/alertinfo.php';

if (isset($_POST['btnlogin'])) {

  $username = $_POST["username"];
  $password = $_POST["password"];

    $sqlcheckempid = "SELECT * FROM `tbl_userinfo` WHERE `username`='$username' AND `password`='$password'
    $result = $conn->query($sqlcheckempid);
    if ($result->num_rows > 0) {


      header ('Location:main.php');
    }
    else {
        $_SESSION['status']="Username and password did not match. Please try again.";
        $_SESSION['status_code'] = "error";

        echo "<script>window.location.href = 'index.php'</script>";



    }

}
$conn->close();
```



```
<?php
require_once 'dbconnect.php';
if (isset($_POST['input'])) {
    $input = $_POST['input'];

    $sqlselect = "SELECT * FROM `view_userinfo` WHERE `lname` LIKE '{$input

    $result = mysqli_query($conn, $sqlselect);

    if (mysqli_num_rows($result) > 0) { ?>
        <div class="container-scroll table-responsive">
            <table class="table table-border table-striped table-hover mt-4
                <thead class=" text-primary text-center" style="background-
                    <tr>
                        <th>Employee ID</th>
                        <th>User Level</th>
                        <th>Username</th>
                        <th>Password</th>
                        <th>Lastname</th>
                        <th>Firstname</th>
                        <th>Middlename</th>
                        <th>Position</th>
                        <th>Contact Number</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    while ($row = mysqli_fetch_array($result)) {
                        $empid = $row['EmployeeID'];
                        $ulevel = $row['Userlevel'];
```

4.3 Database Schema/ERD



lis_db.tbl_facultyinfo
EmployeeID: NATIONAL CHARACTER VARYING
lname: NATIONAL CHARACTER VARYING
fname: NATIONAL CHARACTER VARYING
mname: NATIONAL CHARACTER VARYING
addressa: NATIONAL CHARACTER VARYING
emailadd: NATIONAL CHARACTER VARYING
cnumber: NATIONAL CHARACTER VARYING
bdate: NATIONAL CHARACTER VARYING
cstatus: NATIONAL CHARACTER VARYING
gender: NATIONAL CHARACTER VARYING
position: NATIONAL CHARACTER VARYING
appdate: NATIONAL CHARACTER VARYING

lis_db.tbl_userinfo
userID: INTEGER
username: NATIONAL CHARACTER VARYING
password: NATIONAL CHARACTER VARYING
datecreated: NATIONAL CHARACTER VARYING
Userlevel: NATIONAL CHARACTER VARYING
status: NATIONAL CHARACTER VARYING
EmployeeID: INTEGER

lis_db.tbl_purchasedinfo Copy 1
PurchasedID: INTEGER
Purdate: TIMESTAMP
dateencoded: TIMESTAMP
qty: INTEGER
unit: NATIONAL CHARACTER VARYING
Remarks: NATIONAL CHARACTER VARYING
ItedID: INTEGER
UserID: INTEGER
AcadID: INTEGER
FundID: INTEGER

lis_db.tbl_requestinfo
RequestID: INTEGER
UserID: INTEGER
ItemID: INTEGER
qty: INTEGER
unit: NATIONAL CHARACTER VARYING
AYID: INTEGER
daterequest: TIMESTAMP
status: NATIONAL CHARACTER VARYING

lis_db.tbl_acadinfo
AYID: INTEGER
AYear: NATIONAL CHARACTER VARYING

lis_db.tbl_approvedinfo
ApprovedID: INTEGER
RequestID: INTEGER
UserID: INTEGER
qty_approved: INTEGER
unit: NATIONAL CHARACTER VARYING
dateapproved: TIMESTAMP

lis_db.tbl_fundinfo
FundID: INTEGER
fundname: NATIONAL CHARACTER VARYING
description: NATIONAL CHARACTER VARYING

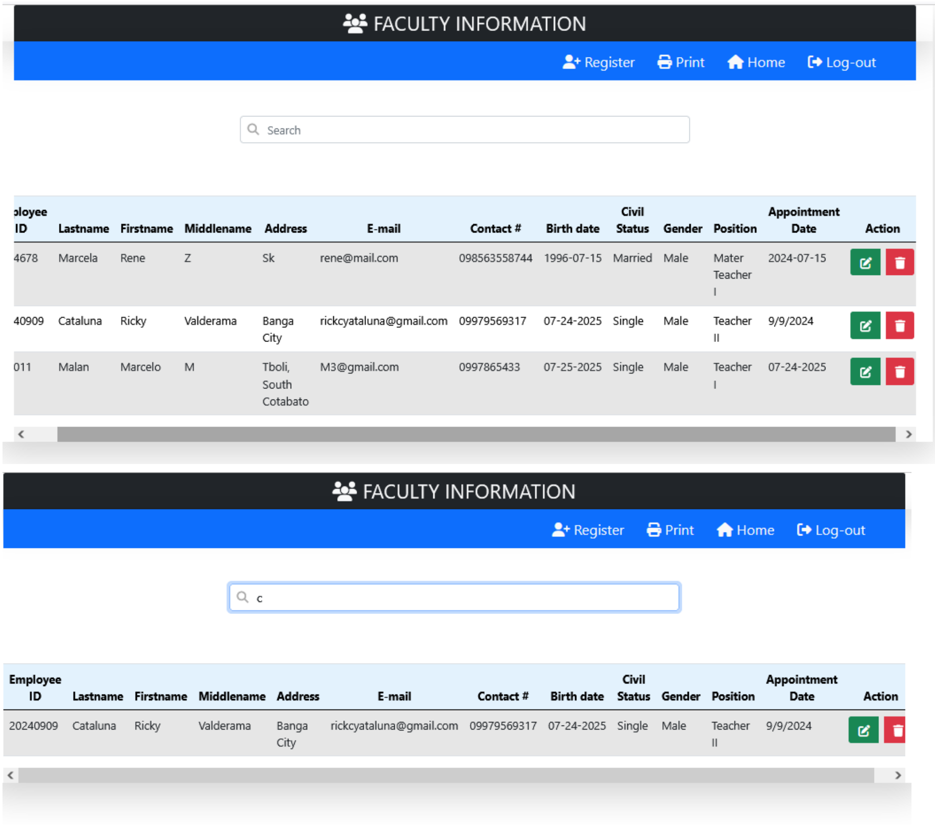
lis_db.tbl_iteminfo
ItemID: INTEGER
Itemname: NATIONAL CHARACTER VARYING
desc: NATIONAL CHARACTER VARYING
qty: INTEGER
unit: NATIONAL CHARACTER VARYING

Chapter V

Result and Testing

5. Result and Testing

5.1 Test Cases

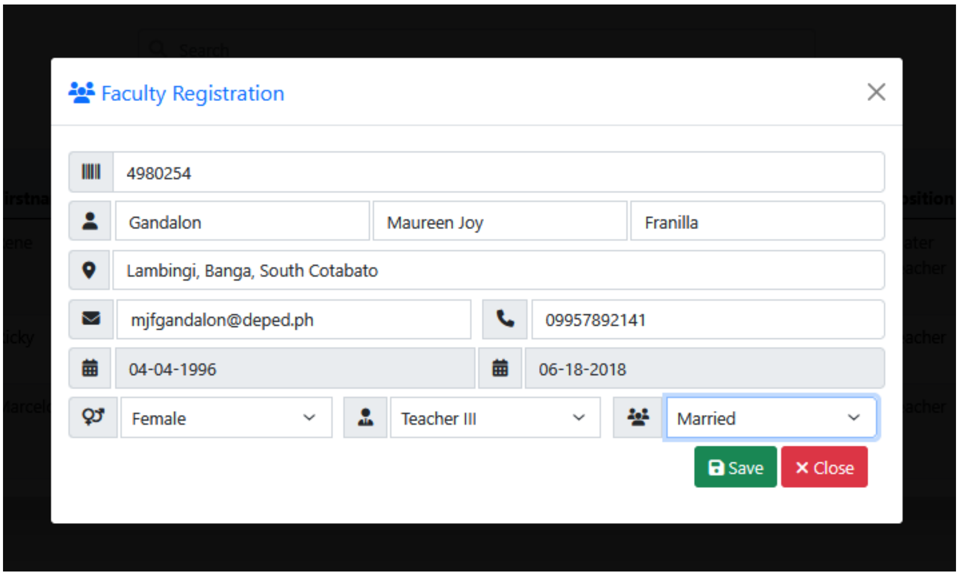


ployee ID	Lastname	Firstname	Middlename	Address	E-mail	Contact #	Birth date	Civil Status	Gender	Position	Appointment Date	Action
4678	Marcela	Rene	Z	Sk	rene@mail.com	098563558744	1996-07-15	Married	Male	Mater Teacher I	2024-07-15	
40909	Cataluna	Ricky	Valderama	Banga City	rickcyataluna@gmail.com	09979569317	07-24-2025	Single	Male	Teacher II	9/9/2024	
011	Malan	Marcelo	M	Tboli, South Cotabato	M3@gmail.com	0997865433	07-25-2025	Single	Male	Teacher I	07-24-2025	

Employee ID	Lastname	Firstname	Middlename	Address	E-mail	Contact #	Birth date	Civil Status	Gender	Position	Appointment Date	Action
20240909	Cataluna	Ricky	Valderama	Banga City	rickcyataluna@gmail.com	09979569317	07-24-2025	Single	Male	Teacher II	9/9/2024	

Figure 10. Search Result

5.2 Screenshots of submitted entries



4980254

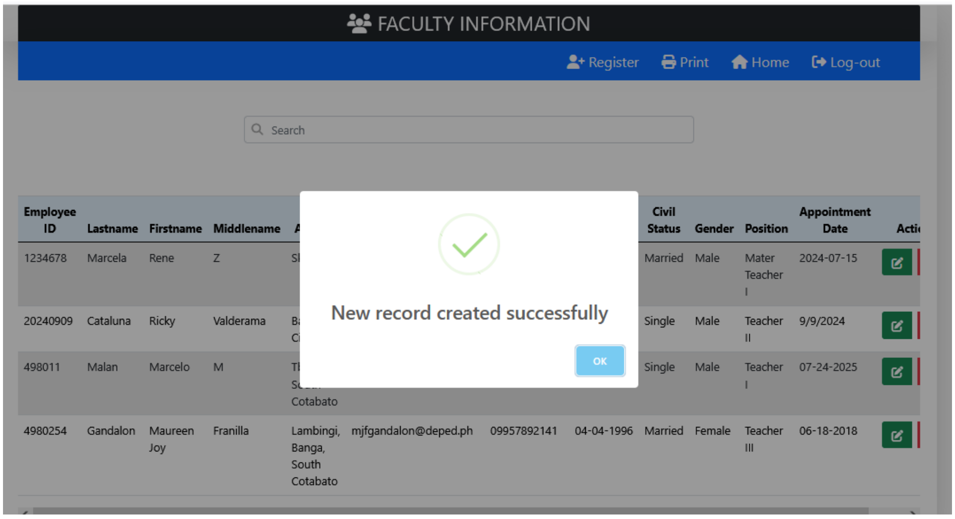
Gandalon | Maureen Joy | Franilla

Lambingi, Banga, South Cotabato

mjfgandalon@deped.ph | 09957892141

04-04-1996 | 06-18-2018

Female | Teacher III | Married



FACULTY INFORMATION

New record created successfully

Employee ID	Lastname	Firstname	Middlename	A				Civil Status	Gender	Position	Appointment Date	Acti
1234678	Marcela	Rene	Z	Sl				Married	Male	Mater Teacher I	2024-07-15	
20240909	Cataluna	Ricky	Valderama	Ba Ci				Single	Male	Teacher II	9/9/2024	
498011	Malan	Marcelo	M	Tl Sc Cotabato				Single	Male	Teacher I	07-24-2025	
4980254	Gandalon	Maureen Joy	Franilla	Lambingi, Banga, South Cotabato	mjfgandalon@deped.ph	09957892141	04-04-1996	Married	Female	Teacher III	06-18-2018	

5.3 Common bugs and how to fix them

Bug Encountered	Description	Fix Applied
Inventory not updating after return	Borrowed items did not reflect back in inventory	Fixed SQL update query in return function
Misaligned UI in mobile view	Buttons overlapped in small screens	Applied Bootstrap grid classes and media queries

5.4 Summary of what worked

The web-based nature of the system allowed real-time access from any device with internet, a major improvement over Excel files.

- The role-based system design enforced accountability and secured access by user type.
- Using Agile allowed for quick turnaround on bugs and user feedback.
- The alert system greatly helped in maintaining supply levels and enforcing return deadlines.
- Teachers and staff found the interface intuitive and easier to use than expected, especially with the integration of Bootstrap for responsiveness.

Chapter VI

Conclusion and Recommendation

6. Conclusion and Recommendations

6.1 Conclusion

The implementation of SmartCustodix, a web-based property custodial management system, successfully addressed the inefficiencies found in the manual custodial processes at Lamhako Integrated School. Through the use of PHP, MySQL, Bootstrap, and other web technologies, the system enabled seamless online requisition, inventory tracking, borrowing management, and automated notifications. It provided role-based access to teaching and non-teaching staff, the School Head, and the Property Custodian, enhancing accountability and streamlining workflows.

The study demonstrated that shifting from a manual to a digital platform using SDLC and Agile methodologies led to improved operational transparency, reduced processing time, and better responsiveness in managing school properties. The system's online accessibility further increased convenience and allowed real-time interaction, which is crucial for schools in geographically dispersed or rural locations.

6.2 Recommendation

Based on the results and feedback from users, the following recommendations are made:

- Wider Implementation – It is recommended that SmartCustodix be adopted by other public schools, especially those with growing inventory management demands and limited staff.
- System Expansion – Future upgrades could include integration with procurement modules, budget tracking, and automated inventory forecasting to further support school-based financial management.
- Training and Orientation – Conduct regular orientation sessions and hands-on training for system users to ensure proper utilization and sustainability of the platform.
- Data Backup and Security – Implement automated backups and strengthen user authentication mechanisms to safeguard data integrity and protect sensitive information.

The continued development and refinement of SmartCustodix will ensure that schools can adapt to the increasing demand for digital transformation and resource accountability.

References:

Department of Education (DepEd). (2016). K to 12 Basic Education Program. Retrieved from https://www.deped.gov.ph/k-to-12/

Schwalbe, K. (2015). Information Technology Project Management (8th ed.). Boston: Cengage Learning.

Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

Sommerville, I. (2016). Software Engineering (10th ed.). Pearson Education.

W3Schools. (n.d.). PHP Tutorial. Retrieved from https://www.w3schools.com/php/

W3Schools. (n.d.). MySQL Tutorial. Retrieved from https://www.w3schools.com/sql/

Bootstrap. (n.d.). *Bootstrap: The world's most popular framework for building responsive, mobile-first sites*. https://getbootstrap.com/

Draw.io. (n.d.). *draw.io – Free online diagram software*. https://www.draw.io/

Lucidchart. (n.d.). *Lucidchart: Online diagramming application*. https://www.lucidchart.com/

Atlassian. (n.d.). What is Agile methodology? Retrieved from https://www.atlassian.com/agile

Bourque, P., & Fairley, R. E. (Eds.). (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK V3.0). IEEE Computer Society.

Open Web Application Security Project (OWASP). (2023). OWASP Secure Coding Practices. Retrieved from https://owasp.org/

Oladele, M. A., Adepoju, S. A., & Babalola, O. A. (2021). *Development of an Inventory Management System using Association Rule*. Journal of Applied Computing and Informatics.

Utomo, A. B., Saputra, E., & Nugroho, Y. (2020). *Development of a Web-Based Inventory Information System*. International Journal of Scientific and Technology Research, 9(2), 3184–3187.

Appendices:

https://github.com/RickyC-glitch/SmartCustodix.git