KARAMAY: Digitalizing Human Rights Education and Enhancing Institutional Effectiveness through a Comprehensive Digital Platform for the Center of Human Rights Education at Camarines Sur Polytechnic Colleges

A Capstone Project
presented to the Faculty of
College of Computer Studies
Camarines Sur Polytechnic Colleges

In Partial Fulfillment of the Requirements
for the degree Bachelor of Science in Information Technology

By
Maybelle R. Agonos
James M. Balane
Feann Faith J. Digay

CHAPTER 3
TECHNICAL BACKGROUND

This chapter represents the detailed discussion of the technical components essential to the development and deployment of the proposed system. It includes the technologies employed, hardware and software requirements, system architecture, programming environment, and the testing methodologies that will ensure the system's quality and reliability. By defining these technical foundations, the chapter demonstrates the project's feasibility and establishes its operational structure.

3.1 Overview of Current Technologies to be Used in the System

The development of the KARAMAY will adopt modern technologies to ensure accessibility, reliability, and efficiency. By utilizing updated technologies, the system can deliver a seamless user experience while meeting the administrative requirements of the CHRE Office.

Codeigniter will be used as the main framework for developing the KARAMAY. CodeIgniter is a light, fast, flexible, and secure PHP full-stack web framework, a toolkit for building websites, that minimizes code needed for tasks. CodeIgniter is compatible for building KARAMAY because it will fit the system's requirements. Its MVC (Model-View-Controller) structure keeps the system organized, making it easier to manage different modules like complaints, appointments, and events.

HTML5 will be used to define the structure of all web-pages within the system, such as the complaint submission form, appointment request page, log in screen, admin dashboard. By providing semantic tags and cross browser compatibility, HTML5 ensures

that the system will be accessible to both students and administrators, whether they use a laptop or smartphone. It will also improve maintainability and searchability of the system, making future updates easier to manage.

CSS will be utilized to style all user interface components such as buttons, tables, and forms within the system. It will organize the visual layout pages like the complaint form, appointment scheduler and admin dashboard. During development, CSS will allow the developers to apply consistent formatting across modules, quickly adjust design elements and ensure the system looks clean and accessible to both desktop and mobile devices. By controlling visual presentation separately from content, it enables faster iteration and easier updates to the interface.

Javascript will be used to add interactivity to the system's user interface. It will handle form validations, real time error messages, dynamic content updates (e.g., loading appointments without page reload), and user notifications. In development, Javascript will streamline front end behavior, allowing smoother experiences and reducing reliance on server calls for basic interactions.

Bootstrap will be integrated to speed up UI development using pre-built responsive components such as navigation bars, alerts, form controls. It will help the system adapt seamlessly across mobile and desktop devices. Bootstrap will accelerate the layout process, ensure visual consistency, and reduce the need to build components from scratch.

PHP will be responsible for handling core logic such as user authentication, complaint tracking, appointment scheduling, and communication between the client side

interface and database. PHP can be used to create dynamic content and server-side processing, which is necessary for handling form data processing, session management, and creating real-time feedback to users.

MySQL will serve as the database where complaints, appointment records, logs, user credentials, and system configurations are stored. It will be used to define relational tables and handle CRUD (Create, Read, Update, Delete) operations. In development, MySQL enables structured data management and easy integration with PHP, making data retrieval and storage efficient.

Through the strategic use of these technologies, the system will provide a secure, efficient, and user-friendly platform that addresses the institution's need for an improved grievance handling and appointment scheduling system.

3.2 Resources

To ensure the successful development, deployment, and maintenance of the system, both hardware and software resources will be carefully selected. These resources will support a robust and efficient system capable of addressing the requirements of the CHRE Office, including online complaint filing, appointment scheduling, and complaint monitoring.

3.2.1 Hardware Specifications

In this part it outlines the essential hardware components required for the development and deployment of the KARAMAY. These components ensure the system

operates smoothly, providing high performance, stability, and compatibility with institutional infrastructure.

Table 1 presents a detailed list of the necessary hardware tools for both the system's development and operational deployment. The listed components, including Laptop/PC for developer and admin, web server, network devices, and mobile devices for users, the components work together to support the seamless functioning of the web-based application and to accommodate multiple users simultaneously.

Table 1

Hardware Specifications

Hardware Tool	Specification
Developer Laptop/PC	Intel Core i5/i7, 8GB–16GB RAM, 256GB SSD
Printer/Scanner (Admin Use)	All-in-one printer/scanner
Admin Desktop PC	Intel Core i3/i5, 8GB RAM
Wi-Fi Router	Dual-band, Wi-Fi 5 or 6
Mobile Devices (User Access)	Android/iOS smartphone or tablet

The table presents the necessary hardware tools required to support the development, deployment, and user interaction of the system. These components are selected to ensure stable performance and reliability across different stages of the system's use.

To support the development environment, it is recommended to use a laptop or PC equipped with at least an Intel Core i5 or i7 processor, 8GB to 16GB of RAM, and a 256GB SSD. It offers high performance through multiple cores and threads, enabling

developers to compile code, debug, and run virtual environments with reduced latency and improved multitasking [6]. A printer is recommended for generating hard copies of reports, appointment confirmations, or complaint records, which are useful for documentation, filing, and administrative use within the CHRE office. Administrative operations may be conducted using a desktop computer with a minimum of an Intel Core i3 or i5 processor and 8GB of RAM, which is sufficient for handling routine management tasks. Reliable network connectivity should be ensured through a dual-band Wi-Fi router supporting Wi-Fi 5 or Wi-Fi 6 standards. Students and other users can access the system conveniently via Android or iOS smartphones and tablets, promoting accessibility and mobility. The users can also use a laptop to access the system if it is more convenient to them.

These hardware specifications are selected to align with the system's functional requirements and to support optimal performance, stability, and user experience throughout development, deployment, and usage phases.

3.2.2 Software Specifications

This section outlines the essential software tools and platforms used in the development and deployment of the system. These tools support the system's frontend and backend development, database management, and testing processes.

Table 2

Software Specification

Software Tool	Specification
Operating system	Windows 10/11 or higher

Integrated Development Environment (IDE)	Visual Studio Code version 1.95 or 1.98
Programming Language	PHP 8.2.12 or higher (Back-End), HTML5, CSS3, JavaScript (ES6) (Front-End)
Database management	MySQL Workbench 8.0 or phpMyAdmin 5.2.1
Local Server Environment	XAMPP Version 8.2.12 (includes Apache 2.4, PHP 8.2, MySQL 8.0)
Front-End Technologies	HTML5, CSS3, JavaScript, Bootstrap 5
Browser	Google Chrome, Mozilla, Firefox, Microsoft Edge
Domain (for deployment)	Free domain via Freenom or purchased via Namecheap or Godaddy

This table presents the software tools that will be used in the development. The system is built to be compatible with Windows 10 and newer versions to ensure stability, security and compatibility with the latest software features [1]. PHP 8.2.12 is selected as the server-side scripting language due to its enhanced performance, improved type safety and strengthened security features, making it a reliable backend solution [2]. Visual Studio Code is the primary development environment because of its lightweight design, rich extension library and cross platform support , which boosts developer productivity [5]. The front-end of the system will be developed using a combination of HTML5, CSS3, JavaScript, and Bootstrap 5, allowing for a responsive and user-friendly interface across various devices and screen sizes. For database management, MySQL will be utilized through MySQL Workbench or phpMyAdmin, providing efficient handling of structured data such as user accounts, appointment records, and complaint logs. The local

development environment was powered by XAMPP, an open-source cross-platform web server solution that supports Apache, MySQL, and PHP integration, making it ideal for testing and debugging the application before deployment. For deployment purposes, a free domain from Freenom or a registered domain from providers like Namecheap or GoDaddy was considered to make the system accessible online. Modern browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge were used during testing to ensure compatibility and consistent performance across different user environments.

Together, these software tools form a reliable stack that supports the efficient development, testing, deployment, and maintenance of the system.

3.2.3 Program Specifications

This part describes all the software components and the technical architecture that will be used to build the system; how this system will process data, how user authentication will be enforced and how secure communications between users and the server will be maintained.

The system prescribes a set of functional and non-functional requirements that characterize its expected behavior, characteristics, and general quality standards. The requirements ensure that the system can adequately accommodate the needs of both CHRE staff and students through efficient handling of complaints and appointment scheduling.

Functional requirements

The Functional Requirement defines what a product must do and what its features and functions are [3]. The system's functional requirements ensure it delivers essential features expected by users.

Module	Functional Requirement	Description
Complaint Management Module	Complaint Submission	Users can file complaints using an online web form.
Notification Module	Email Notification	Complainants receive updates about the status of their complaints via email.
Complaint Management Module	Real-time Monitoring	Complaints can be monitored in real-time through a dedicated dashboard.
Privacy & Anonymity Module	Anonymity & Privacy	System supports anonymous complaints, data privacy, and non-disclosure agreements.
Complaint Management Module	Staff Processing	Authorized CHRE staff can process and update the status of complaints via the admin panel.
Appointment Management Module	Appointment Booking	Users can book appointments with CHRE staff online.
Appointment Management Module	Appointment Editing	Appointments can be edited and updates are reflected on the dashboard.
Appointment Management Module	Rescheduling & Cancellation	CHRE staff can reschedule or cancel appointments through the admin panel.
Event Management	Event Creation	CHRE staff can schedule

Module		and publish events related to human rights education.
Document Upload Module	Uploading of documents	Complainants, CHRE Staff can upload necessary documents
OPCR Monitoring Module	OPCR Monitoring	Staff can view and monitor OPCR-related activities and metrics.
Reporting Module	Report Generation	Staff can generate complaint summary reports and other reports for documentation or analysis.

Non-Functional Requirements

Non-Functional requirements are the criteria that define how a system should behave, rather than what it is supposed to do [4].

Module	Non-Functional Requirement	Description
Performance Optimization Module	Fast Response Time	The system should load pages and respond to user actions within three seconds under normal conditions.
Concurrency Management Module	Support Multiple Users	The system must be able to handle at least fifty concurrent users without performance degradation.
Authentication & Authorization Module	User Authentication	All users (students, staff, admin) must log in to access appropriate system features, ensuring data confidentiality.

Input Validation & Sanitization Module	Data Validation	All user inputs must be properly validated to prevent errors and protect against security threats like SQL injection and cross-site scripting.
System Reliability Module	High Availability	The system should maintain 99% uptime during operational hours to ensure continuous service delivery.
UI Compatibility Module	Cross-Browser Compatibility	The system must function correctly on major web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
Privacy & Compliance Module	Data Privacy Compliance	The system must adhere to the Data Privacy Act of the Philippines (RA 10173) to ensure the confidentiality and proper use of personal data.

3.2.3 Programming Environment

This section describes the programming environment used in the development of the system. It includes the tools, frameworks, and languages utilized for both the front end and back end of the system, ensuring seamless functionality, responsiveness, and data processing throughout the application.

Front End

The front-end development of the system will be carried out using a combination of HTML5, CSS3, and JavaScript, which are standard web technologies that ensure semantic structure, responsive design, and interactive user experiences. To enhance the

visual layout and responsiveness of the interface across devices, the Bootstrap 5 framework was used. Bootstrap offers a wide range of pre-designed UI components, grid systems, and utilities that accelerate front-end development. These technologies are chosen for their browser compatibility, lightweight performance, and strong community support.

Back End

For the back end, the system will utilize PHP as the primary server-side scripting language. PHP is selected due to its robustness, improved performance in recent versions, and seamless integration with relational databases like MySQL. The application logic, user authentication, data validation, and server-side processing were implemented in PHP. Additionally, MySQL will be used as the database management system, interfaced through phpMyAdmin and MySQL Workbench for easy administration and structured data handling. The development and testing environment will be powered by XAMPP, which integrates Apache, MySQL, and PHP to simulate a local server setup.

3.3 Testing Plan

This section outlines the testing strategy to ensure that the system will perform according to its functional requirements. The system will undergo Black Box Testing, a method that focuses on input-output behavior without delving into internal code structure.

Black box testing, a form of testing that is performed with no knowledge of a system's internals, can be carried out to evaluate the functionality, security, performance, and other aspects of an application [7].

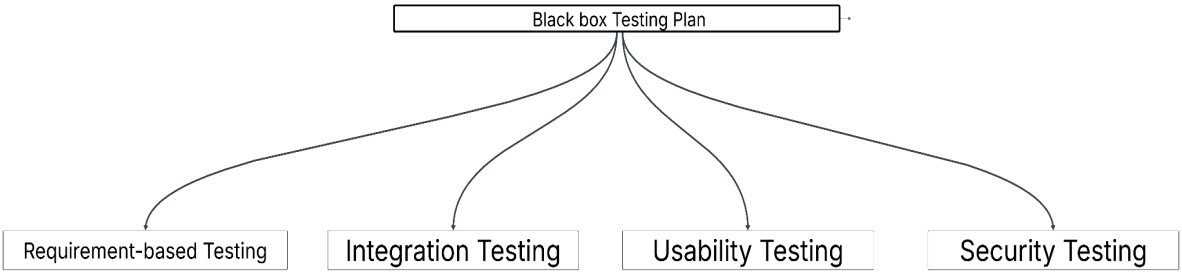

Figure 1. Black box Testing Plan

Figure 1 illustrates the black box testing strategies that will be employed for the proposed system, which include Requirement-Based Testing to verify functional and non-functional requirements, Usability Testing to evaluate user-friendliness and interface effectiveness, Integration Testing to ensure that modules work together as intended, and Security Testing to evaluate the system's ability to protect data and resist unauthorized access.

3.3.1 Types of Testing

In evaluating the system, Black Box Testing will be employed as it is about evaluating the system's functionality through user interactions without looking at the internal structure of the code. This is ideal for a web-based system in which the main objective is to determine whether the user interface, submission of complaints, scheduling of appointments, and administrative tasks function properly. Three main types of Black Box Testing will be employed: requirement-based testing, usability testing and deployment testing.

Requirement-based Testing. Requirement-based testing ensures that the software is developed as per the given requirements, and its working is aligned as per the expected results [9]. Requirements-based testing is crucial for ensuring that the system does not

only work from a user's point of view, but also aligns precisely with the agreed-upon specifications, rules, and expectations of the stakeholders in this case, the CHRE office and the institution. This includes features like secure authentication, proper data validation, web responsiveness, appointment update, and real-time complaint monitoring. It ensures that every specified requirement is fulfilled and that the system functions in alignment with the CHRE office's operational goals.

Usability Testing. Usability testing usually involves observing users as they attempt to complete tasks and can be done for different types of designs [8]. In this study, usability testing will be used to ensure that students, staff, and administrators can easily navigate and interact with the system without confusion or errors. It will help confirm that users can efficiently file complaints and perform related tasks with minimal assistance. Since the system is intended to be used by individuals with varying levels of technical skill, usability testing is essential to guarantee that the interface is intuitive, accessible, and aligns with the goal of delivering fast, user-friendly, and reliable support services for the CHRE office.

Integration Testing. Integration Testing is a software testing technique that focuses on verifying the interactions and data exchange between different components or modules of a Software Application [13]. In this study, integration testing will be used to ensure that the modules for complaint filing, appointment scheduling, event management, and OPCR monitoring function cohesively as a unified system. It will help confirm that data flows correctly between modules for example, when a complaint is filed and needs to be tracked across both the complaint and administrative dashboards. This testing will

also validate that user roles such as student, staff, and administrator can interact with the system consistently without data loss or miscommunication between functions. Ensuring smooth integration is essential for maintaining accuracy, efficiency, and user trust in the overall system.

Security Testing. Security Testing is a type of software testing that uncovers vulnerabilities in the system and determines that the data and resources of the system are protected from possible intruders [14]. In this study, security testing will be used to verify that only authorized users can access restricted areas and sensitive data within the system. It will help ensure the confidentiality and integrity of complaint records, user information, and administrative actions.

3.3.2 Testing Tools and Framework

To ensure that the system meets its functional requirements and delivers a user-friendly experience, a structured combination of testing tools and methodologies will be employed. The evaluation will focus on black-box testing, which verifies system behavior based on inputs and expected outputs without accessing the internal source code.

Black-box testing will be utilized to validate the system's external behavior, particularly for key modules such as complaint filing, appointment scheduling, and OPCR monitoring. By focusing on input-output validation, this technique ensures that the system responds correctly to user actions, meeting defined functional requirements without needing to inspect internal logic.

Manual testing will simulate real-world interactions by navigating through pages, submitting forms, and observing outputs. This method will help detect interface issues, broken links, and layout inconsistencies that automated tools may miss. Manual testing is especially important during early development when functionality and design are still being refined.

Usability testing will be conducted to involve end-users—primarily students and CHRE staff who will perform typical tasks such as logging in, submitting complaints, or reviewing OPCR data. Their feedback will inform improvements in system navigation, content clarity, and overall user experience, directly supporting Objective 3.3 on usability testing [11].

Browser compatibility testing will be conducted to ensure that the system will function reliably across multiple web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. This will help guarantee that the interface, styles, and functionalities will render consistently and operate correctly for all users, regardless of browser type or version. Web applications that fail to pass compatibility testing risk alienating users on less common platforms [12].

Cross Device Compatibility testing will be executed to meet the needs of users accessing the system on different devices, the platform will be tested on desktops, tablets, and mobile devices running Android and iOS. This ensures a seamless and responsive experience, which supports broader inclusivity and system adoption [10].

Security testing will focus on protecting user data and preventing unauthorized access. Input fields will be tested for vulnerabilities like SQL injection, sessions will be reviewed for proper management, and data will be checked for proper encryption. These tests are essential for fulfilling Objective 3.4 on securing the platform.

Performance testing will be conducted to evaluate how the system will behave under various levels of load. This will include testing the response time, throughput, and stability when multiple users are simultaneously accessing the platform. The system will be expected to support users without any performance degradation, as defined in the non-functional requirements. Load testing helps prevent latency and failure issues in production environments.

Notes

[1] Anon. 2025. Check Windows 10 System Requirements & Specs: Microsoft. Retrieved April 30, 2025 from https://www.microsoft.com/en-us/windows/windows-10-specifications

[2] PHP Group. 2023. PHP 8.2.12 Release Announcement. Retrieved April 30, 2025 from https://www.php.net/releases/8.2/en.php

[3] AltexSoft. 2023. Functional and Nonfunctional Requirements Specification. (November 2023). Retrieved April 30, 2025 from https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/

[4] Anon. 2024. Non-functional Requirements: What They Do, Examples, and Best Practices. (August 2024). Retrieved April 30, 2025 from https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples

[5] Microsoft. 2021. Documentation for Visual Studio Code. (November 2021). Retrieved May 1, 2025 from https://code.visualstudio.com/docs

[6] Anon. 2025. Intel® Core™ Processors – View Latest Generation Core Processors. Retrieved May 1, 2025 from https://www.intel.com/content/www/us/en/products/details/processors/core.html

[7] Michali. 2022. What is Black Box Testing? (February 2022). Retrieved May 1, 2025 from https://www.checkpoint.com/cyber-hub/cyber-security/what-is-penetration-testing/what-is-black-box-testing/

[8] Anon. 2024. What is Usability Testing? (November 2024). Retrieved May 23, 2025 from https://www.interaction-design.org/literature/topics/usability-testing?srsltid=AfmBOoptXf76QdM-8JcdgyjhDsavqFLhbz-cwg-EBDci1MQNxy6QmkW5

[9] Anon. 2025. Requirements Based Testing. Retrieved May 1, 2025 from https://www.tutorialspoint.com/software_testing_dictionary/requirements_based_testing.htm

[10] Miracool. 2023. A Comprehensive Guide to Cross-Browser Testing. DEV Community. Retrieved April 30, 2025 from https://dev.to/miracool/a-comprehensive-guide-to-cross-browser-testing-4p8k

[11] J. Nielsen. 1994. 10 Usability Heuristics for User Interface Design. Nielsen Norman Group. Retrieved April 30, 2025 from https://www.nngroup.com/articles/ten-usability-heuristics/

[12] A. Mesbah and M. R. Prasad. 2011. Automated Cross-Browser Compatibility Testing. In *Proceedings of the 33rd International Conference on Software Engineering (ICSE '11)*. Retrieved April 30, 2025 from https://www.cs.columbia.edu/~junfeng/12fa-e6121/papers/browser-compat.pdf

[13] Anon. 2025. Integration Testing - Software Engineering - GeeksforGeeks. Retrieved June 13, 2025 from https://www.geeksforgeeks.org/software-engineering-integration-testing/

[14] Anon. 2025. Security Testing - Software Testing - GeeksforGeeks. Retrieved June 13, 2025 from https://www.geeksforgeeks.org/security-testing/