PROJECT PROPOSAL DOCUMENT

JHCSC GUIDANCE: Digital Students Information Records Management System

Submitted by:

DMA PROTOCOL

Carlowe Deala

Dave Jay Madrazo

Marnel Asutilla

Submitted to:

Jaafar Omar

Course & Section:

BSIT – 3rd Year, Section A

Date:

08/10/2025

1. Introduction

1.1 Purpose

The purpose of this project is to develop a web-based College Guidance File Management System that centralizes student guidance records from first year to fourth year. The system aims to allow students to fill out and submit guidance-related forms online and enable guidance personnel to easily search, retrieve, and update student files. This will minimize manual processes, reduce administrative delays, and improve accessibility for both students and staff.

1.2 Intended Audience

- College Students (1st–4th Year): Submit and view guidance-related forms and personal records.
- Guidance Personnel: Manage, track, and search student records.

1.3 Product Scope

The system will:

- Provide an online platform for students to submit forms without visiting the guidance office.
- Allow guidance staff to search and retrieve records by name, ID, or batch/year.
- Organize records chronologically for tracking student progress.

The initial version will not include:

- Mobile application development.
- Integration with external Student Information Systems (SIS).
- Automated counseling appointment scheduling.

1.4 Definitions, Acronyms, and Abbreviations

- SIS – Student Information System
- DBMS – Database Management System
- GUI – Graphical User Interface
- Role-Based Access – Restricting system access depending on the user's role (student, staff, admin)

2. Overall Description

2.1 User Characteristics

- Students: Basic computer literacy, can navigate a website to fill and submit forms.
- Guidance Personnel: Familiar with record-keeping and counseling workflows, moderate computer skills.
- Administrators: Higher-level access, responsible for reviewing overall reports and ensuring compliance.

2.2 Constraints

- The system must be accessible via web browsers (Chrome, Edge, Firefox).
- Must comply with Data Privacy Act of 2012 (Philippines) for handling student records.
- Limited by college network infrastructure (internet speed, server storage).
- Only available to authorized users with proper login credentials.

2.3 Assumptions and Dependencies

- Students will have reliable access to the internet to use the system.
- Guidance office staff will be trained to use the platform.
- The system will rely on a stable database server (e.g., MySQL, PostgreSQL).
- Security updates and maintenance will be regularly performed by IT staff.
- Future expansion may include integration with SIS and mobile application development.

3. Requirements Specifications

3.1 Functional Requirements

The system shall:

- Allow students to log in securely and submit guidance-related forms online.
- Enable guidance personnel to search student records by name, ID number, or batch/year.
- Store and organize records chronologically for each student.
- Allow authorized staff to update, edit, and retrieve student records.
- Provide role-based access control (student, guidance staff, administrator). Support data backup and restore functionality.
- Generate simple reports for administrators (e.g., number of cases, student progress tracking).

3.2 Non-Functional Requirements

- Performance: The system should handle at least 200 concurrent users without performance degradation.
- Usability: The interface should be user-friendly and require minimal training.

- Security: Must comply with the Data Privacy Act of 2012, ensuring that student data is protected and only accessible to authorized personnel.
- Reliability: The system must be available 99% of the time during school hours.
- Maintainability: The codebase should be modular to allow future integration with SIS and mobile apps.
- Portability: The web app should run on major browsers (Chrome, Edge, Firefox).

3.3 External Interface Requirements

User Interface (UI):

- Students: A web-based portal to submit forms and view their own records.
- Guidance Personnel: A dashboard for searching, updating, and managing records.
- Administrators: Access to reports and overall system monitoring.

Hardware Interface:

- Standard desktop or laptop computer with internet connection.

Software Interface:

- Web browser (Chrome, Edge, Firefox).
- Database server (MySQL or PostgreSQL).

Communication Interface:

- Secure HTTPS protocol for all data transmissions.

3.4 System Models

Context Diagram: Shows the interaction between students, guidance personnel, administrators, and the database.

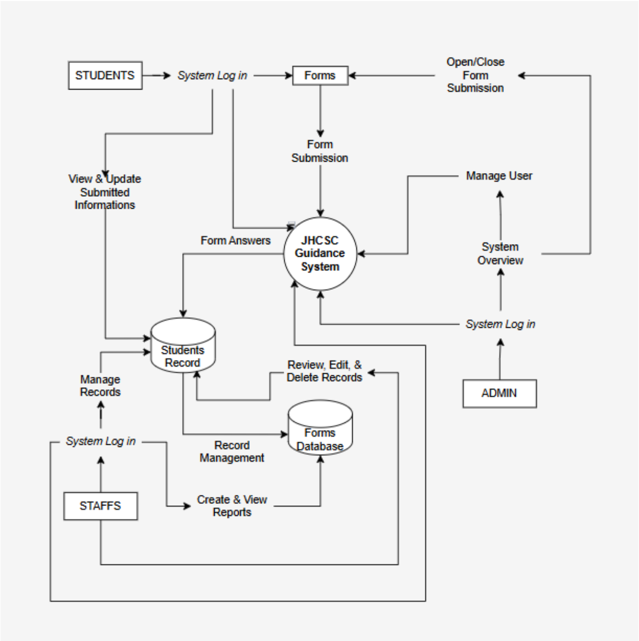

Figure 1

Use Case Diagram: Represents major functions such as submitting forms, searching records, updating records, generating reports, and performing backup/restore.

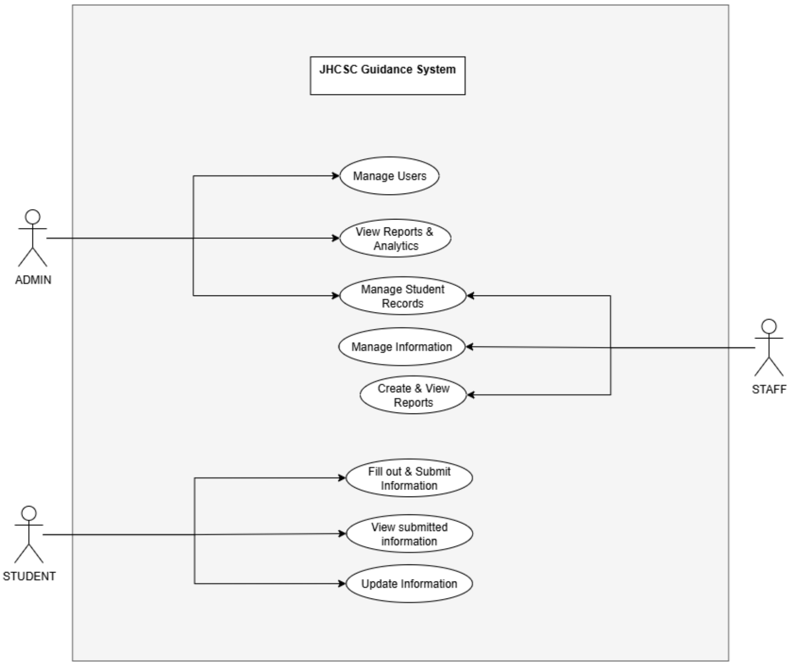

Figure 2

3.5 Database Design

Entity-Relationship Diagram (ERD): Defines the relationship between student profiles, guidance forms, records, and administrative reports.

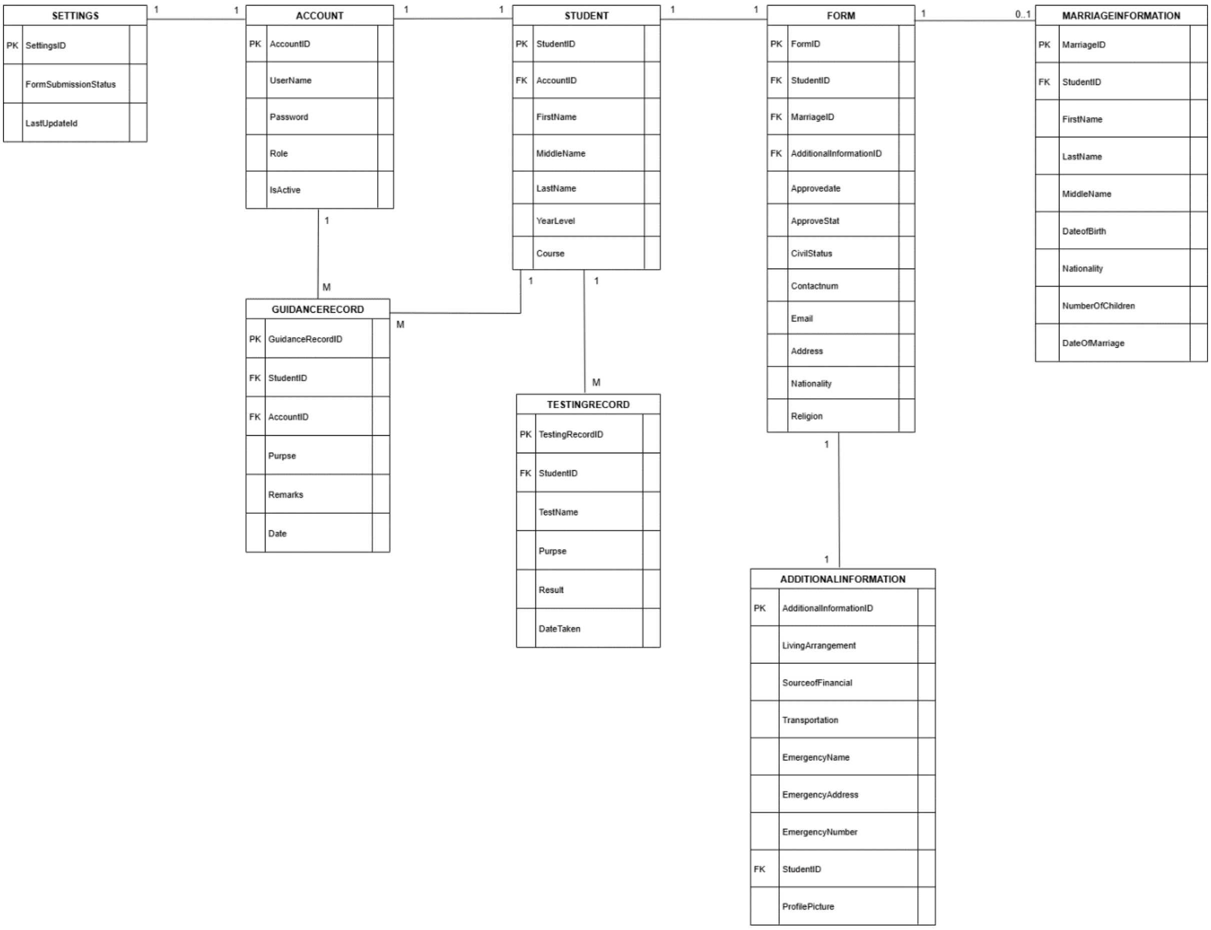

Figure 3

3.6 Implementation

Development Environment

The JHCSC Guidance System will be developed using a web development stack optimized for the system's requirements, ensuring scalability, security, and compliance with the Data Privacy Act of 2012 (Philippines). The development environment includes:

- **Operating System**: Windows for both development and production environments, aligning with the use of XAMPP for local testing and deployment.

- **Integrated Development Environment (IDE)**: Visual Studio Code for coding, debugging, and file management, with PHP and MySQL extensions for efficient development.
- **Version Control**: Local version control using manual backups and folder versioning on developer machines to track code changes and facilitate collaboration.
- **Database Server**: MySQL 8.0, configured locally for development and on a Windows-based server for testing and production.
- **Local Development Server**: XAMPP for local testing, providing Apache and MySQL to simulate the production environment.

Programming Languages

- **PHP (v8.2)**: For server-side logic, handling form submissions, user authentication, and database interactions, as implemented in studform.php, student.php, Staff.php, and Admin-Dashboard.php.
- **HTML5**: For structuring web pages, ensuring semantic markup and accessibility, as seen in student.php, studland.php, and login.php.
- **CSS3**: For styling, with responsive design principles applied in studform.css, studview.css, dash.css, and login.css to ensure compatibility across devices.
- **JavaScript (ES6+)**: For client-side interactivity, such as dynamic form field toggling in studform.php and student filtering in Staff.php.
- **SQL**: For querying and managing the MySQL database, as implemented in the database interactions across PHP files.

Frameworks and Libraries

- **Bootstrap 5**: For responsive and consistent UI components, enhancing user experience across browsers (Chrome, Edge, Firefox), as utilized in studform.css and dash.css.
- **ch**: For simplified DOM manipulation and AJAX requests, used in studform.php for dynamic field visibility and in Staff.php for interactive elements.
- **Chart.js**: For generating visual reports (e.g., cases by semester, students by year) in the admin dashboard, as referenced in Admin-Dashboard.php and Staff.php.

Tools

- **Composer**: For managing PHP dependencies, ensuring libraries are up-to-date.
- **npm**: For managing front-end dependencies like Bootstrap and jQuery, as utilized in the project's asset compilation.
- **MySQL Workbench**: For designing and managing the database schema, including the revised ERD with entities like ACCOUNT, STUDENT, FORM, ADDITIONAL_INFORMATION, etc.
- **Postman**: For testing API endpoints, particularly for potential future RESTful API integration with external systems.
- **phpMyAdmin**: For manual database administration during development, as implied in the database connection setups in studform.php and login.php.

Development Approach

- **Agile Methodology**: Development will follow an iterative approach with two-week sprints, delivering modules like user authentication (login.php), student form submission (studform.php), and admin dashboard (Admin-Dashboard.php) incrementally.
- **Modular Design**: The codebase is organized into reusable components (e.g., authentication in login.php, form handling in studform.php, record retrieval in student.php) to facilitate maintenance and future integration with a Student Information System (SIS) or mobile app.
- **Security Measures**:
  - Passwords are hashed using PHP's password_hash() function, as implemented in login.php.
  - Prepared statements are used in MySQL queries (e.g., studform.php, Staff.php, Admin-Dashboard.php) to prevent SQL injection.
  - File uploads (e.g., profile pictures in studform.php) are validated for type (JPEG, PNG) and size (2MB limit) to ensure security.
- **Testing**:
  - Unit testing for PHP backend logic to validate form processing and database interactions.
  - Integration testing to ensure seamless interaction between frontend (studform.css, studview.css) and backend (student.php, Staff.php).
  - Browser compatibility testing for Chrome, Edge, and Firefox, ensuring portability as specified in the non-functional requirements.
- **Deployment**: The system will be deployed on an Apache server with MySQL, hosted on a Windows-based server. Regular backups will be configured using Windows Task Scheduler to meet the data backup requirement. The Uploads/ directory for profile pictures will be secured with proper file permissions.

Implementation Details Specific to the System

- **User Authentication**: Implemented in login.php using session management and role-based redirection to Admin-Dashboard.php, Staff.php, or studland.php based on the Role field in the ACCOUNT table.
- **Student Form Submission**: studform.php handles form submissions with conditional fields for living arrangements and marriage information, storing data in the FORM, ADDITIONAL_INFORMATION, and MARRIAGE_INFORMATION tables, with dynamic field visibility controlled by JavaScript.
- **Record Management**: student.php and Staff.php provide search and retrieval functionality by name, ID, or year, fetching data from the STUDENT and FORM tables using PDO or MySQLi prepared statements.
- **Admin Dashboard**: Admin-Dashboard.php and Staff.php include report generation (e.g., student counts, cases by semester) using Chart.js, with role-based access for admins and staff, leveraging queries on guidancerecord and student tables.

- **Responsive Design**: CSS files (studform.css, dash.css, login.css) use Bootstrap and media queries to ensure usability on various devices, meeting the portability requirement for Chrome, Edge, and Firefox.
- **Database Interactions**: All PHP files use PDO or MySQLi with prepared statements for secure database access, supporting the revised ERD structure with entities like TESTING_RECORD, GUIDANCE_RECORD, and SETTINGS.