**NCHIRE: DEVELOPING A DIGITAL RECRUITMENT PLATFORM FOR NORZAGARAY COLLEGE**

**SOFTWARE REQUIREMENTS SPECIFICATION**

**September 2025**

**Version 3.5**

**PREPARED BY:**

**Lorenzana, John Kennedy B.**

**Manansala, John Paul**

**Medico, Kyla Mae J.**

**Orolfo, Monica S.**

**I.INTRODUCTION--------- --- -----------------------------------------------------**

This Software Requirements Specification (SRS) describes the NCHire Online Recruitment Platform for Norzagaray College, designed to provide a structured approach to managing the institution’s hiring process. The document presents an overview of the platform, including its objectives, scope, and requirements, while also identifying its primary users and the environment in which it will operate. It serves as a reference for developers, administrators, and stakeholders to ensure a clear understanding of the system’s features and functions. The purpose of this SRS is to collect, organize, and define the requirements necessary for the development of the platform, emphasizing user accessibility, organized applicant management, and dependable communication between applicants and the college administration.

1.1 **Purpose**

The purpose of this document is to specify the requirements for the ***NCHire*** online recruitment platform, which will function as a centralized system for managing recruitment activities at Norzagaray College. It serves as a framework for developers in building the platform, a guide for administrators in its use, and a reference for stakeholders in understanding the project’s objectives. The platform will support applicant account registration, online submission of application forms, uploading of required documents, and resubmission of updated files when necessary. It will also provide administrators with tools for posting job vacancies, organizing applicant records, tracking applications, and sending notifications to applicants. By documenting both functional and non-functional requirements, this SRS provides the foundation for software design, testing, and evaluation. In addition, it anticipates future enhancements, ensuring that the platform remains adaptable and scalable to meet the evolving needs of the institution.

**1.2 Objectives**

The main objective of this system is to develop and implement a digital recruitment platform that enhances the hiring process of Norzagaray College. It aims to digitizemanual recruitment activities, reduce errors, and improve the efficiency of managing job postings, applicant information, and communication. This development supports the institution’s goal of transitioning to a paperless and more sustainable process, ensuring faster, more accurate, and organized recruitment operations. Through its digital features, the platform provides a transparent and user-friendly experience for both applicants and administrators.

In addition, this SRS document outlines the overall objectives, functions, and technical requirements essential for the development of the ***NCHire*** platform. It specifies the intended users including administrators and applicants, as well as the system’s interface, hardware, and software specifications. The document also guides designers and developers throughout the system development phases to ensure that the platform meets Norzagaray College’s needs for a reliable, efficient, and sustainable recruitment system.

**1.3 Scope**

The scope of this study focuses on the development of **NCHire: Online Recruitment Platform for Norzagaray College**, which aims to simplify and organize the institution’s hiring process:

* In Scope

The NCHire platform focuses on improving the recruitment process at Norzagaray College by providing a centralized, digital system for both administrators and applicants. The system enables the dean to create and manage job postings, schedule interviews, track applicant submissions, update and maintain records, monitor recruitment progress, and send notifications to applicants. For users, the platform allows applicants to create accounts, browse available job postings, submit applications and required documents, receive updates on their application status, and resubmit updated files when necessary. All applicant records and documents are securely stored and managed, ensuring organized access for authorized users.

* Out of Scope

The platform does not cover certain recruitment-related processes outside the scope of Norzagaray College. It does not provide automated applicant evaluation, ranking, or skill-matching tools, and final hiring decisions remain the responsibility of the dean. Document verification for authenticity is performed manually, and functions such as salary negotiation, onboarding, payroll management, or advanced communication like live chat or video calls are not included. While the system supports web browser access on desktop and mobile devices, a dedicated mobile application is not part of the current implementation. Additionally, the platform assumes that applicants have a stable internet connection and basic digital literacy to navigate the system effectively.

* 1. **Definitions, Acronyms, and Abbreviations**

This section provides a glossary of terms, acronyms, and abbreviations used in the document to ensure a common understanding among developers, administrators, and stakeholders.

* **Applicant** – A job seeker who uses the system to apply for a position at Norzagaray College.
* **CRUD** – Create, Read, Update, Delete; basic operations performed in database management.
* **CSS** – **C**ascading **S**tyle **S**heets; used to design and style the visual layout of the NCHire platform.
* **Database** – An organized collection of data stored electronically, allowing easy access, retrieval, and management of applicant records.
* **Dean (Administrator)** – The primary user responsible for managing recruitment activities within the platform, such as posting job vacancies, reviewing applications, and scheduling interviews.
* **Digital Platform** – An online environment where recruitment activities such as application submission, document management and evaluation take place without requiring physical presence.
* **GDPR** – **Ge**neral**D**ata**P**rotection**R**egulation
* **HTML** – **H**yper**t**ext **M**arkup **L**anguage; the standard language used to create and structure web pages for the platform.
* **IEEE** – **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
* **ISTQB** – **I**nternational **S**oftware **T**esting **Q**ualifications **B**oard
* **JavaScript** – The programming language used to add interactivity and dynamic features to the NCHire platform.
* **KMP** – Knuth-Morris-Pratt algorithm
* **Manual Process** – A paper-based and manual system for processing and storing records.
* **MySQL** – A relational database management system used to store and manage applicant and administrator data.
* **NCHire** – The recruitment platform developed for Norzagaray College.
* **Onboarding** – The process of integrating a new hire into the organization after they have been selected.
* **SHA-256** – Secure Hash Algorithm 256-bit, a cryptographic function for securing user passwords and sensitive data.
* **SRS** – Software Requirements Specification; this document that outlines the system's requirements.
* **UI** – User Interface; the visual layout and design of the system that users interact with.
* **UX** – User Experience; the overall experience of a user while interacting with the system.
* **Web Browser** - is the software application such as Google Chrome or Microsoft Edge that enables users to access and interact with the NCHire over the internet.
* **XAMPP** – A local server solution used for development, combining Apache, MySQL, PHP, and Perl.
  1. **References**

The references are:

* **Software engineering standards (IEEE 830, IEEE 29148)**
* **Web accessibility standards (WCAG Compliance)**
* **Security standards (GDPR, OWASP)**
* **Testing standards (ISTQB)**

**II. REQUIREMENTS SPECIFICATION--------- --------------------------------------------**

**2.1 Functionality**

**2.1.1 *User Account Management***

Both applicants and administrators can create and manage accounts safely. Applicants can only see their own information, while administrators can handle job postings and all applications.

**2.1.2 *Job Posting and Management***

Administrators can add new job openings, update existing ones, or remove old listings. They can also schedule interviews and track who has applied for each job.

**2.1.3 *Application Submission and Tracking***

Applicants can submit their forms and documents online and update them if needed. They can see the status of their applications at any time, keeping everything clear and transparent.

**2.1.4 *Notifications***

The platform lets applicants know about updates, like when an interview is scheduled or a document needs resubmission.

**2.2 Usability**

**2.2.1 *User Interface***

The platform is designed with a simple and intuitive interface for both desktop and mobile web browsers. Dashboards, forms, and notifications are easily navigable, minimizing the learning curve for both applicants and administrators.

**2.2.2 *Ease of Use***

All features are organized logically. Clear labels, descriptive headings, and visual cues guide users efficiently through the system.

**2.3 Accessibility**

**2.3.1 *Visual Accessibility***

The platform supports high-contrast modes and readable fonts to assist users with visual difficulties.

**2.3.2 *Navigation and Interface Consistency***

Menus, buttons, and icons are consistent across the platform to help users quickly locate and access features.

**2.3.3 *Browser Compatibility***

The system works on modern web browsers and adapts to different screen sizes, although it does not include a dedicated mobile app.

**2.4 Reliability and Availability**

**2.4.*1 Data Protection and Backup***

All applicant and administrative data is stored securely, with regular backups to prevent loss in case of system failure.

**2.4.2 *System Access***

The platform is designed to be accessible at all times, with minimal downtime for maintenance, allowing uninterrupted use by applicants and administrators.

**2.5** **Performance**

**2.5.1 *Fast Processing***

Data retrieval and updates are optimized to provide immediate feedback on applications and administrative actions.

**2.5.2 *Lightweight Interface***

The user interface is designed to load quickly, using only essential elements to reduce waiting times even for users with slower internet connections.

**2.6 Security**

**2.6.*1 Secure Login and Access***

The system enforces strong passwords and secure authentication. Users’ access is limited according to their role to prevent unauthorized actions.

**2.6.2 *Data Privacy***

Sensitive information, including personal and educational data, is encrypted and only accessible to authorized users.

**2.6.3 *Protection Against Unauthorized Actions***

Administrative functions are restricted to authorized accounts, preventing accidental or malicious data alterations.

**2.7 Supportability and Maintainability**

**2.7.1 *Flexible System Structure***

NCHire is built in separate, independent modules. This means that if one part needs improvement or repair, it can be changed without disrupting the rest of the platform.

**2.7.2 *User and Admin Guides***

Step-by-step instructions and helpful tips are included for both applicants and administrators, so anyone can use the system confidently and complete tasks smoothly.

**2.7.3 *Tracking Changes and Updates***

The platform uses version control to keep a record of all changes. This makes it easier for developers to work together, fix mistakes, or revert to a previous version if needed.

**2.8 Scalability**

**2.8.1 *Handling More Applicants***

The platform is built to support a growing number of applicants. As more users register and submit applications, the system will maintain fast response times and organized data management.

**2.8.2 *Expanding Job Postings***

Administrators can add additional job vacancies without affecting the performance of existing postings or user interactions. The system can store and display an increasing number of job listings efficiently.

**2.8.3 *Adding Administrators***

More administrative accounts can be created as needed, allowing multiple staff members to manage recruitment tasks simultaneously without disrupting ongoing operations.

**2.8.4 *System Growth***

The overall architecture is designed to accommodate future updates and expansions, including potential new features or integrations, ensuring the platform remains functional and adaptable over time.

**2.9 Design Constraints**

**2.9.1 *Technology Stack Constraints***

The platform will use **HTML, CSS, JavaScript, PHP, and React** for the front end, with MySQL for database management. Tailwind CSS will handle styling, and XAMPP will serve as the local development environment.

**2.9.2 *Algorithm and Security Constraints***

Essential algorithms such as Binary Search, QuickSort, and CRUD operations will be used for efficient data handling. Passwords will be encrypted using SHA-256, and session management will secure user logins.

**2.9.3 *Hardware and Software Constraints***

The system is designed to run on standard laptops with Windows 10. Visual Studio Code will be used for coding and testing, ensuring smooth development and compatibility with the chosen tools.

**2.9.4 *Scalability Constraints***

The platform must support an increasing number of applicants, job postings, and administrative accounts without needing major redesigns, allowing future growth and feature additions.

**2.10 Applicable Standards**

**2.10.1 *Coding and Development Practices***

The platform will follow structured development practices to keep the code organized and maintainable. Developers will write clear, consistent code and document their work so that updates or improvements can be done efficiently.

**2.10.2 *Accessibility Guidelines***

NCHire will prioritize ease of use for all users. The platform will feature clear labels, simple navigation, and readable layouts to help applicants and administrators complete tasks without confusion.

**2.10.3 *Data Protection Standards***

The software shall implement strong security measures following the General DataProtection Regulation (GDPR) and OWASP security guidelines. This includes the following:  
**Encrypted Data Storage**. All sensitive data, including personal user information and passwords, shall be encrypted during transmission and while stored in the database.

**Secure Authentication**. User login shall be protected through strong password rules and secure session management to prevent unauthorized access.

**Controlled Data Access**. A role-based access system shall be used to ensure users and administrators can only access the data and functions allowed for their role.

**2.10.4 *Testing and Verification Standards***  
 The software shall undergo complete testing based on ISTQB (International Software Testing Qualifications Board) standards to ensure it works correctly and reliably. This includes the following:

**Unit Testing**. Each individual part of the software shall be tested on its own to verify it works as intended.

**Integration Testing**. The connections between different software parts shall be tested to make sure they work together properly.

**User Acceptance Testing (UAT)**. End-users shall test the final system to confirm it meets their needs and is ready for use.

**III.OVERALL DESCRIPTION--- -- ----------------------------------------------**

**3.1 Product Features**

**3.1.1 Summary**

NCHire: Developing a Digital Recruitment Platform for Norzagaray College is designed to replace the current manual and paper-based hiring process with a centralized online system. The platform supports applicant submissions, organized record management, and clear communication between applicants and administrators. It addresses the problems identified in the institution’s traditional recruitment practices such as delays, repetitive tasks, and limited accessibility by providing a faster, more transparent, and user-friendly digital solution.

**3.1.2 Use Case Scenarios**

The content of each use case scenario in the NCHire digital recruitment platform includes several essential elements that define the interaction between the system and its users. These components describe how each function of the platform operates under different conditions to ensure smooth and efficient recruitment processes.

**Use Case: User Sign Up**  
**Actor:**

* + Applicant

**Preconditions:**

* + - The user has access to the internet and the NCHire platform.
    - The user does not yet have an existing account.

**Trigger:**

* The applicant selects the “Sign Up” option on the login page.

**Main Success Scenario (Primary Flow):**

1. The applicant clicks “Sign Up.”
2. The system displays the registration form.
3. The applicant enters the required information such as Name, Email, and Password.
4. The applicant submits the form.
5. The system verifies the input and saves the account to the database.
6. The applicant is redirected to the “Sign In” page to log in using the new account.

**Alternate Flow:**  
Empty Fields:  
3a. If required fields are empty, the system displays an error message.  
Duplicate Email:  
4a. If the email is already registered, the system shows a “User already exists” warning.  
Database Error:  
5a. If there is a network or database issue, registration fails and an error message appears.

**Postconditions:**

* + A new applicant account is created and stored in the database.

**Use Case: User Sign In**  
**Actors:**

* + Admin ( Dean )
  + Applicant

**Precondition:**

* The user has a registered account in the system.

**Trigger:**

* The user selects “Sign In” and enters their credentials.

**Main Success Scenario (Primary Flow):**

1. The user opens the Sign In page.
2. The user enters their Email and Password.
3. The system verifies the credentials.
4. If valid, the user is successfully logged in.
5. The system redirects the user to the appropriate dashboard based on their role:
   * Admin Dashboard: View Applications, Manage Users, Create/Edit/Delete Job Posts, View Applicants
   * Applicant Dashboard: View Available Job Posts, Track Application Status.
6. The dashboard displays content such as Application Overview, Recent Activity, Recent Job Posts, and Recent Applicants.

**Alternate Flow:**  
Invalid Credentials:  
3a. If the credentials are invalid, the system displays “Invalid Email or Password.”  
Empty Fields:  
2a. If required fields are empty, the system prompts completion.  
Server Error:  
4a. If there is a connection or server issue, an error message is displayed.

**Postconditions:**

* The user is logged into their respective dashboard with role-based access.

**Use Case: Manage Job Posts**  
**Actor:**

* Admin ( Dean )

**Preconditions:**

* The Dean is logged into the system.

**Trigger:**

* The Dean selects the “Job Post” section to create a new job posting.

**Main Success Scenario (Primary Flow):**

1. The Dean clicks “Add Job Post.”
2. The system displays a form requiring job details.
3. The Dean fills out the following fields:
   * Job Title
   * Department
   * Employment Type (Full-Time or Part-Time)
   * Location
   * Salary Range
   * Deadline of Submission
   * Minimum Qualifications
   * Description
4. The Dean submits the form.
5. The system saves the post and displays it under “Recent Job Posts.”
6. The system sends notifications to users about the new post.

**Alternate Flow:**  
Missing Fields:  
3a. If required fields are missing, the system prompts completion.  
Database Error:  
5a. If a system or network error occurs, the system displays an error message.

**Postconditions:**

* The new job post is saved and viewable by all authorized users.

**Use Case: Edit or Delete Job Post**  
**Actor:**

* Admin ( Dean )

**Preconditions:**

* The Dean is logged in and has access to existing job posts.

**Trigger:**

* The Dean selects a job post to modify or remove.

**Main Success Scenario (Primary Flow):**

1. The Dean opens the Job Post section.
2. The Dean selects a post to edit or delete.
3. If “Edit,” the Dean updates details and clicks “Save Changes.”
4. If “Delete,” the Dean confirms deletion.
5. The system updates or removes the post.
6. A notification is sent to users about the update or removal.

**Alternate Flow:**  
Incomplete Edits:  
3a. If edited details are incomplete, the system rejects the update.  
Deletion Canceled:  
4a. If deletion is canceled, the system returns without changes.

**Postconditions:**

* The job post is successfully updated or removed from the database.

**Use Case: Apply for Job**  
**Actor(s):**

* Applicant

**Preconditions:**

* The applicant is logged into the system.
* At least one active job post exists.

**Trigger:**

* The applicant selects a job post and clicks “Apply.”

**Main Success Scenario (Primary Flow):**

1. The applicant views the list of available job posts.
2. The applicant selects a specific job post to read full details.
3. The applicant clicks “Apply.”
4. The system displays an application form.
5. The applicant fills in the form and uploads required documents (e.g., résumé, transcript).
6. The applicant submits the application.
7. The system saves the submission and marks the status as “Under Review.”
8. A notification is sent to the Admin or Dean about the new application.

**Alternate Flow:**  
Missing Document:  
5a. If a required document is missing, the system prompts completion.  
Submission Error:  
6a. If submission fails, an error message is shown.

**Postconditions:**

* The applicant’s application and uploaded documents are stored and visible to authorized personnel.

**Use Case: View Applications**  
**Actor:**

* Admin ( Dean )

**Preconditions:**

* The Admin is logged in.
* Applicants have submitted their applications.

**Trigger:**

* The Admin opens the “Applications” section on the dashboard.

**Main Success Scenario (Primary Flow):**

1. The Admin selects the “Applications” section.
2. The system displays all submitted applications.
3. The Admin views applicant details and attached requirements.
4. The Admin can monitor and manage application statuses.

**Alternate Flow:**  
No Applications:  
2a. If there are no applications, the system displays “No applications available.”  
Database Error:  
3a. If the database fails to load, an error message is displayed.

**Postconditions:**

* The Admin successfully reviews and monitors all submitted applications.

**Use Case: Manage Users**  
**Actor(s):**

* Admin ( Dean )

**Precondition:**

* The Admin is logged in.

**Trigger:**

* The Admin selects the “Users” section.

**Main Success Scenario (Primary Flow):**

1. The Admin clicks “Add User.”
2. The system displays a user registration form.
3. The Admin enters details (Name, Email, Role – Dean).
4. The Admin submits the form.
5. The system saves the new user in the database.

**Alternate Flow:**  
Empty Fields:  
3a. If a required field is empty, the system prompts for completion.  
Duplicate User:  
4a. If the user already exists, the system shows a warning.

**Postconditions:**

* A new Dean user is successfully added and stored in the system.

**Use Case: View Dashboard**  
**Actor(s):**

* Admin (Dean)
* Applicant

**Preconditions:**

* The user is logged in.

**Trigger:**

* The user accesses the dashboard page.

**Main Success Scenario (Primary Flow):**

1. The system loads the dashboard interface.
2. The dashboard displays the following sections:

* Application Overview
* Recent Activity
* Recent Job Posts
* Recent Applicants

1. The user can click on each section for more details.
2. The dashboard updates automatically as new data is added.

**Alternate Flow:**  
No Data:  
2a. If data cannot be retrieved, the system displays “No data available.”

**Postconditions:**

* The dashboard displays updated and accurate information to the user.

**Use Case: Notification System**  
**Actor(s):**

* System (Automated)
* Admin ( Dean )
* Applicant

**Preconditions:**

* The system is active and connected to the database.
* Events such as job posting, editing, or new applications occur.

**Trigger:**

* A trigger event happens (e.g., new job post, updated post, or new application).

**Main Success Scenario (Primary Flow):**

1. The system detects a trigger event.
2. The system sends a notification to the affected users.
3. Users receive and view notifications on their dashboards.

**Alternate Flow:**  
Offline User:  
2a. If the user is offline, notifications are queued and shown later.  
Notification Error:  
2b. If notification sending fails, the system logs the error.

**Postconditions:**

* Notifications are delivered and visible to the intended users.

**3.2 User Characteristics**

**NCHire** is designed for two types of users: Administrator and Applicant. Both user types are expected to have stable internet access and basic knowledge of using web browsers. The platform is designed to be user-friendly and accessible on devices such as desktop computers, laptops, or tablets that support modern web browsers.

**3.2.1 Administrators**

Administrators are responsible for managing the recruitment process within the NCHire platform. They are expected to have knowledge of basic web application operations, such as creating and posting job openings, monitoring applications, and updating applicant statuses. Familiarity with the college’s hiring procedures and evaluation criteria is essential for effective use of the system.

**3.2.2 Applicants**

Applicants are individuals seeking employment opportunities through the NCHire platform. They should be able to register for an account, log in, and complete their personal and educational profiles. Applicants must also know how to upload required documents, apply for job postings, and track their application status. Basic computer literacy and the ability to navigate web applications are required to ensure smooth interaction with the system.

Both user types should be comfortable using online platforms and have access to a device with a web browser to efficiently use the features of the NCHire recruitment system.

* 1. **Operating Environment**

**3.3.1 Hardware Requirements**

* **Processor**:
  + Intel i3 or higher
* **RAM:** 4 GB RAM or higher
* **Browser:** Google Chrome or Microsoft Edge (latest versions recommended); compatible with other modern browsers such as Mozilla Firefox and Safari.
* **OS**: Windows 10 or higher (Recommended), compatible with macOS and Linux operating systems that support modern web browsers.

**3.3.2 Software Requirements**

**Operating System (OS) Requirements**

* Works on any OS that supports web development environments:  
  o Windows 10 / 11   
  o Ubuntu 20.04+ (common for servers)  
  o macOS 10.15+

**Development Environment**

* **Visual Studio Code** – as the primary source-code editor for building and maintaining the NCHire platform.
* **XAMPP** – used to create a local development environment with Apache and PHP for backend integration and local testing.
* **MySQL** – serves as the database management system to store applicant data, job listings, and recruitment records.

**Frontend Framework**

* **Tailwind CSS** – a utility-first CSS framework used for designing a responsive and modern user interface for the NCHire platform.