**Full Stack Job Portal with Strapi and Next.js using REST API and Postgres**

**Software Requirements Specification (SRS)**

Version 1.0



**Group Id: S240298ED1**

**Supervisor Name : Amjad Iqbal Khan**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 20/05/2024 | 1.0 | Full Stack Jobs Portal is a web-based project designed for job seekers looking for employment and companies that are hiring. In this project, we will leverage the strengths of Next.js, Strapi, and Postgres. By following industry best practices, we aim to create an application that is both robust and efficient. | BC200414326 |

**Table of Contents**

* [Scope (of the project)](#scope)
* [Functional Requirements Non Functional requirements](#FRNFR)
* [Use Case Diagram](#UCD)
* [Usage Scenarios](#UCS)
* [Adopted Methodology](#Adopted)
* [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

**Scope of Project:**

Full Stack Job Portal project aims to develop a robust web application using Next.js, Strapi (with Postgres), and REST API for job seekers, companies, and administrators. The key features and scope of this project are as follows:

**1. User Roles and Authentication:**

* Implement user authentication and authorization using Simple JWT.
* Differentiate between user roles: Admin, Content Editor/Manager, and Frontend Users (registered companies and job seekers).

**2. Job Management:**

* Registered companies can add, update, and manage job postings.
* Specify job details including company name, address, expected salary, job type, experience required, and application deadline.
* Jobs are categorized by type (Permanent, Contractual, Full Time, Part Time).

**3. Job Search and Filtering:**

* Users can search jobs based on keywords and location.
* Filter jobs by type, education, experience, and salary range using checkboxes.

**4. Profile Management:**

* Registered users can update their profiles, upload CVs, and view the list of applied jobs.
* Companies can view and manage their posted jobs and see the list of candidates who applied.

**5. Content Management:**

* Content Editors/Managers can review, approve, and publish jobs posted by companies.
* Manage media assets associated with jobs (images).

**Functional and Non Functional Requirements:**

**Functional Requirements**

**1. Super Admin / Administrator**

|  |  |
| --- | --- |
| **Sr.** | **Functional Requirements** |
| **FR1** | Admin can **login**. |
| **FR2** | Admin can **logout**. |
| **FR3** | Admin can **manage users, managers and companies accounts.** |
| **FR4** | Admins can **grant or revoke access permissions.** |
| **FR5** | Admin can **view overall site statistics, including user activity and job trends.** |
| **FR6** | Admin can **generate reports on job statistics.** |

**2. Content Editor / Manager**

|  |  |
| --- | --- |
| **Sr.** | **Functional Requirements** |
| **FR1** | Manager can **register.** |
| **FR2** | Manager can **login**. |
| **FR3** | Manager can **logout**. |
| **FR4** | Manager can **access job postings for moderation.** |
| **FR5** | Manager can **approve job postings.** |
| **FR6** | Manager can **upload images/media related to job postings.** |
| **FR7** | Manager can **access and manage uploaded CVs by applicants.** |

**3. Companies (Frontend Users)**

|  |  |
| --- | --- |
| **Sr.** | **Functional Requirements** |
| **FR1** | Companies can **register.** |
| **FR2** | Companies can **login**. |
| **FR3** | Companies can **logout**. |
| **FR4** | Companies can **manage their profiles**. |
| **FR5** | Companies can **create and publish job postings with required details** |
| **FR6** | Companies can **upload images related to job postings.** |
| **FR7** | Companies can **set expiry date for job postings to prevent applications after the deadline.** |
| **FR8** | Companies can **view and filter applicants based on job postings.** |
| **FR9** | Companies can **monitor user engagement with job postings.** |

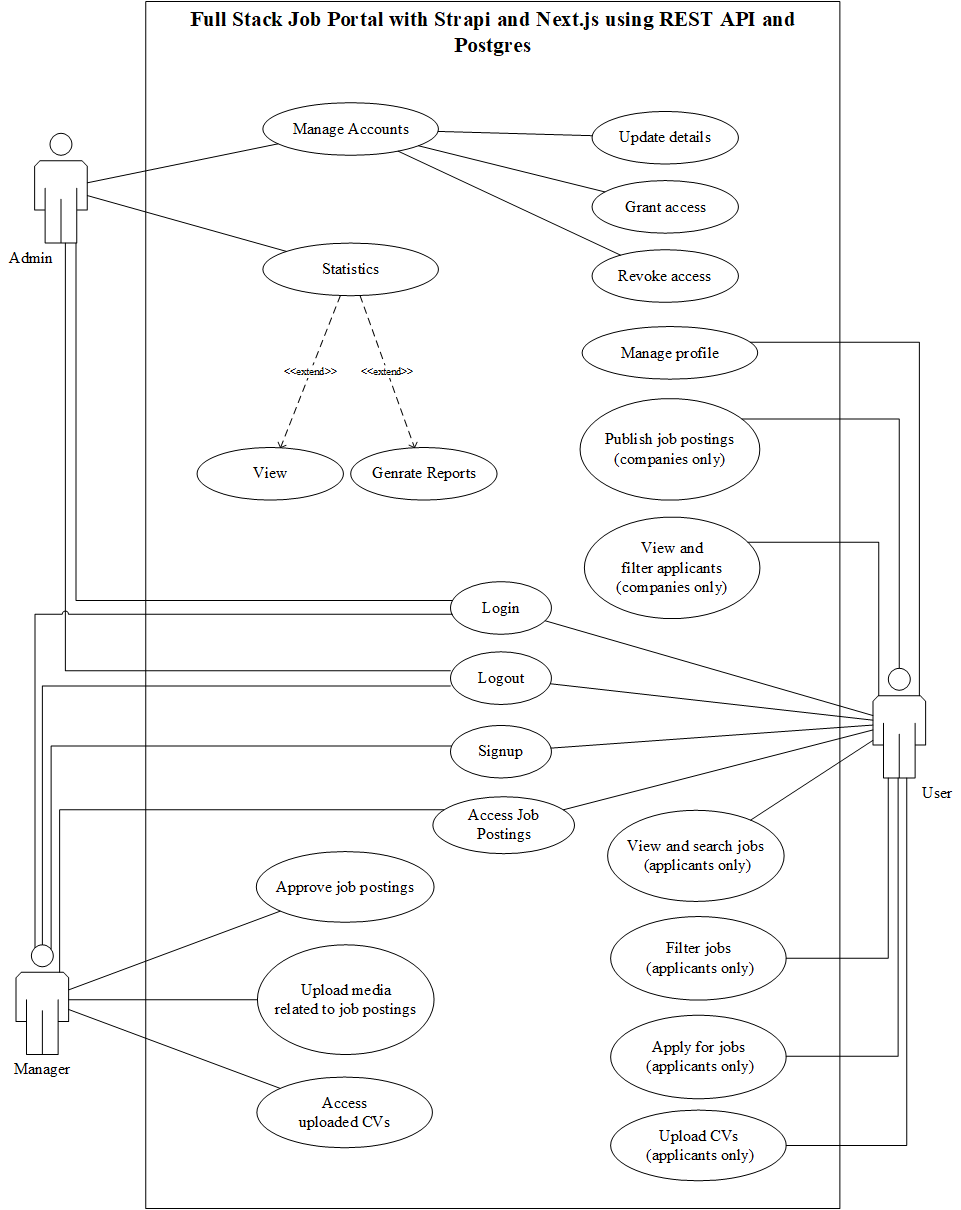
**4. Applicants (Frontend Users)**

|  |  |
| --- | --- |
| **Sr.** | **Functional Requirements** |
| **FR1** | Applicants can **register.** |
| **FR2** | Applicants can **login**. |
| **FR3** | Applicants can **logout**. |
| **FR4** | Applicants can **manage their profiles**. |
| **FR5** | Applicants can **view and search jobs based on keywords and location.** |
| **FR6** | Applicants can **filter jobs based on type, education, experience, and salary range.** |
| **FR7** | Applicants can **apply for jobs.** |
| **FR8** | Applicants can **upload CVs while applying for job.** |

**Non-Functional Requirements**

| **Sr.** | **Non Functional Requirements** |
| --- | --- |
| **NFR1** | Implement **JWT-based authentication** for user login and access control. |
| **NFR2** | Ensure **role-based access** to features and data. |
| **NFR3** | Implement **custom error handling** for better user experience. |
| **NFR4** | Provide **meaningful error messages** and **appropriate status codes**. |
| **NFR5** | Optimize **server-side rendering** with Next.js for faster page loads. |
| **NFR6** | Ensure **efficient database queries** and **data caching**. |
| **NFR7** | Implement **data encryption** for sensitive user information. |
| **NFR8** | **Prevent unauthorized access** and data breaches. |
| **NFR9** | **Deploy application** on Railway and Vercel. |
| **NFR10** | Integrate Cloudinary for **secure and efficient file uploads** (CVs, images). |
| **NFR11** | Ensure proper configuration and **management of uploaded files**. |

**Use Case Diagram:**



**Usage Scenarios:**

**1. Login**

| **Use Case Title** | **Login** |
| --- | --- |
| Use Case ID | 01 |
| Actors | Admin, Manager, User |
| Description | Users can log into the system. |
| Actions | 1. User navigates to the login page.  2. User enters credentials.  3. User submits the login form.  4. System verifies credentials and grants access. |
| Alternative Paths | If credentials are incorrect, the user is prompted to re-enter them. |
| Pre/Post Conditions | Pre: User must have an account.  Post: User is logged in and redirected to the dashboard. |
| Exceptions | If the login fails, an error message is displayed. |
| Author | BC200414326 |

**2. Logout**

| **Use Case Title** | **Logout** |
| --- | --- |
| Use Case ID | 02 |
| Actors | Admin, Manager, User |
| Description | Users can log out of the system. |
| Actions | 1. User clicks on the logout button.  2. System logs out the user.  3. User is redirected to the login page. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User must be logged in.  Post: User is logged out and session is ended. |
| Exceptions | If the logout process fails, an error message is displayed. |
| Author | BC200414326 |

**3. Signup**

| **Use Case Title** | **Signup** |
| --- | --- |
| Use Case ID | 03 |
| Actors | User (Applicants/Companies) |
| Description | New users can create an account. |
| Actions | 1. User navigates to the signup page.  2. Select role (Applicant/Company).  3. User enters required details.  4. User submits the signup form.  5. System creates a new account and redirects to the login page. |
| Alternative Paths | If any details are incorrect, the user is prompted to correct them. |
| Pre/Post Conditions | Pre: User must not have an existing account.  Post: New account is created and user is prompted to log in. |
| Exceptions | If the signup fails, an error message is displayed. |
| Author | BC200414326 |

**4. Manage Accounts**

| **Use Case Title** | **Manage Accounts** |
| --- | --- |
| Use Case ID | 04 |
| Actors | Admin |
| Description | Admin can manage users, managers, and companies accounts (update details, grant access, revoke access). |
| Actions | 1. Admin logs into the system.  2. Admin navigates to the “Manage Accounts” section.  3. Admin selects a user account.  4. Admin updates details, grants access, or revokes access.  5. Admin saves changes. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: Admin must be logged in.  Post: Account details are updated, access is granted or revoked as per the admin’s actions. |
| Exceptions | If the account update fails, an error message is displayed. |
| Author | BC200414326 |

**5. Access Statistics**

| **Use Case Title** | **Access Statistics** |
| --- | --- |
| Use Case ID | 05 |
| Actors | Admin |
| Description | Admin can view and generate statistical reports. |
| Actions | 1. Admin logs into the system.  2. Admin navigates to the “Statistics” section.  3. Admin views existing statistics.  4. Admin selects parameters for generating a report.  5. Admin generates and views the report. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: Admin must be logged in.  Post: Statistical reports are generated and viewed. |
| Exceptions | If the report generation fails, an error message is displayed. |
| Author | BC200414326 |

**6. Access Job Postings**

| **Use Case Title** | **Access Job Postings** |
| --- | --- |
| Use Case ID | 06 |
| Actors | User |
| Description | Users can access and view job postings. |
| Actions | 1. User logs into the system.  2. User navigates to the job postings section.  3. User views the list of available job postings. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User must be logged in.  Post: Job postings are displayed to the user. |
| Exceptions | If job postings cannot be retrieved, an error message is displayed. |
| Author | BC200414326 |

**7. View and Search Jobs**

| **Use Case Title** | **View and Search Jobs** |
| --- | --- |
| Use Case ID | 07 |
| Actors | User |
| Description | Users can view and search for jobs. |
| Actions | 1. User logs into the system.  2. User navigates to the job search page.  3. User enters search criteria.  4. User views search results. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User must be logged in.  Post: Job search results are displayed. |
| Exceptions | If the search fails, an error message is displayed. |
| Author | BC200414326 |

**8. Filter Jobs**

| **Use Case Title** | **Filter Jobs** |
| --- | --- |
| Use Case ID | 08 |
| Actors | User |
| Description | Users can apply filters to job searches. |
| Actions | 1. User logs into the system.  2. User navigates to the job search page.  3. User applies filters to narrow down search results.  4. User views filtered results. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User must be logged in.  Post: Filtered job search results are displayed. |
| Exceptions | If the filter application fails, an error message is displayed. |
| Author | BC200414326 |

**9. Apply for Jobs**

| **Use Case Title** | **Apply for Jobs** |
| --- | --- |
| Use Case ID | 09 |
| Actors | User |
| Description | Users can apply for jobs. |
| Actions | 1. User logs into the system.  2. User navigates to a job posting.  3. User clicks on the apply button.  4. User submits application details.  5. System confirms the application. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User must be logged in and have a completed profile.  Post: Job application is submitted. |
| Exceptions | If the application submission fails, an error message is displayed. |
| Author | BC200414326 |

**10. Upload CVs**

| **Use Case Title** | **Upload CVs** |
| --- | --- |
| Use Case ID | 10 |
| Actors | User (Applicants) |
| Description | Users can upload their CVs. |
| Actions | 1. User logs into the system.  2. User opens specific job posting.  3. User uploads their CV while applying for job.  4. System stores the CV. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User must be logged in.  Post: CV is uploaded and stored. |
| Exceptions | If the CV upload fails, an error message is displayed. |
| Author | BC200414326 |

**11. Publish Job Postings**

| **Use Case Title** | **Publish Job Postings** |
| --- | --- |
| Use Case ID | 11 |
| Actors | User (Companies) |
| Description | User (Companies) can publish job postings. |
| Actions | 1. Companies logs into the system.  2. Companies navigates to the job postings section.  3. Companies creates a new job posting.  4. Companies publishes the job posting.  5. Job posting is available to users. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User (Companies) must be logged in.  Post: Job posting is published and visible to users. |
| Exceptions | If the job posting fails to publish, an error message is displayed. |
| Author | BC200414326 |

**12. View and Filter Applicants**

| **Use Case Title** | **View and Filter Applicants** |
| --- | --- |
| Use Case ID | 12 |
| Actors | User (Company) |
| Description | Companies can view and filter job applicants. |
| Actions | 1. User (Company) logs into the system.  2. User (Company) navigates to the applicants section.  3. User (Company) can apply filters to view specific applicants.  4. User (Company) views filtered list of applicants. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: User (Company) must be logged in.  Post: Filtered list of applicants is displayed. |
| Exceptions | If the filtering fails, an error message is displayed. |
| Author | BC200414326 |

**13. Approve Job Postings**

| **Use Case Title** | **Approve Job Postings** |
| --- | --- |
| Use Case ID | 13 |
| Actors | Manager |
| Description | Managers can approve job postings. |
| Actions | 1. Manager logs into the system.  2. Manager navigates to the job approval section.  3. Manager reviews job postings.  4. Manager approves or rejects job postings. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: Manager must be logged in.  Post: Job postings are approved or rejected. |
| Exceptions | If the approval process fails, an error message is displayed. |
| Author | BC200414326 |

**14. Upload Media Related to Job Postings**

| **Use Case Title** | **Upload Media Related to Job Postings** |
| --- | --- |
| Use Case ID | 14 |
| Actors | Manager, User (Company) |
| Description | Actors can upload media related to job postings. |
| Actions | 1. Manager, User (Company) logs into the system.  2. Manager, User (Company) navigates to a specific job posting.  3. Manager, User (Company) uploads media (images, videos, etc.).  4. System stores the uploaded media. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: Manager, User (Company) must be logged in.  Post: Media is uploaded and linked to the job posting. |
| Exceptions | If the media upload fails, an error message is displayed. |
| Author | BC200414326 |

**15. Access Uploaded CVs**

| **Use Case Title** | **Access Uploaded CVs** |
| --- | --- |
| Use Case ID | 15 |
| Actors | Manager, User (Company) |
| Description | Manager, User (Company) can access uploaded CVs of applicants. |
| Actions | 1. Manager, User (Company) logs into the system.  2. Manager, User (Company) navigates to the applicants section.  3. Manager, User (Company) views uploaded CVs of applicants. |
| Alternative Paths | None |
| Pre/Post Conditions | Pre: Manager must be logged in.  Post: Uploaded CVs are displayed for viewing. |
| Exceptions | If the CV access fails, an error message is displayed. |
| Author | BC200414326 |

**Adopted Methodology:**

For the successful execution of our software development project, we adopted the VU Process Model, which is a hybrid approach combining elements from both the Waterfall and Spiral models. This model was chosen to leverage the structured nature of the Waterfall model while incorporating the iterative, risk-driven features of the Spiral model, thereby enhancing both the discipline and flexibility of our development process.

### Overview of VU Process Model

The VU Process Model integrates the sequential phases of the Waterfall model with the iterative cycles of the Spiral model. It is particularly well-suited for projects requiring a blend of structured development and iterative refinement to manage risks effectively.

**Methodology Breakdown**

**1. Requirements Gathering and Analysis**

**Phase: Waterfall**

* **Objective:** Collect and analyze detailed requirements from stakeholders.
* **Activities:** Conduct interviews, surveys, and workshops with stakeholders to gather requirements. Document these requirements clearly.
* **Output:** Software Requirements Specification (SRS).

**2. System Design**

**Phase: Waterfall**

* **Objective:** Develop a comprehensive system design based on the SRS.
* **Activities:** Create high-level architecture designs, detailed system and component designs, and user interface designs.
* **Output:** Design Document.

**3. Prototyping**

**Phase: Spiral**

* **Objective:** Develop prototypes to explore and validate critical aspects of the system.
* **Activities:** Implement prototypes focusing on high-risk areas such as new technologies, performance bottlenecks, or complex functionalities.
* **Output:** Prototypes and feedback for refining requirements and designs.

**4. Implementation**

**Phase: Waterfall**

* **Objective:** Develop the detailed design and begin implementation.
* **Activities:** Use insights gained from prototypes to refine system designs. Implement the system according to the detailed design.
* **Output:** Source code and detailed design documents.

**5. Iterative Refinement**

**Phase: Spiral**

* **Objective:** Iteratively refine the system based on feedback from testing and stakeholder review.
* **Activities:** Conduct iterative cycles where the system is developed, tested, and reviewed. Each cycle focuses on improving and expanding system functionality.
* **Output:** Updated versions of the system, testing results, and refined requirements.

**6. Integration and System Testing**

**Phase: Waterfall**

* **Objective:** Integrate system components and perform comprehensive testing.
* **Activities:** Combine all system components and conduct system-level tests to ensure they work together seamlessly.
* **Output:** Tested and integrated system.

**7. Deployment and Maintenance**

**Phase: Waterfall**

* **Objective:** Deploy the system to the production environment and provide ongoing maintenance.
* **Activities:** Prepare deployment plans, perform user training, and establish a maintenance plan for post-deployment support.
* **Output:** Deployed system and maintenance documentation.

**Benefits of Using the VU Process Model**

1. **Structured Approach:** The Waterfall phases ensure a clear, structured approach to development, which is essential for maintaining order and clarity in project execution.
2. **Risk Management:** The iterative Spiral phases allow for continuous risk assessment and mitigation, reducing the likelihood of project failure due to unforeseen issues.
3. **Flexibility:** The model allows for iterative refinements, making it adaptable to changing requirements and stakeholder feedback.
4. **Comprehensive Testing:** By integrating the Waterfall approach's thorough testing phases, the model ensures the system's robustness and reliability.

**Work Plan (Use MS Project to create Schedule/Work Plan)**

