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

**Software Requirements Specification**

Version 1.0



**\**

**Group Id: S2402B4CFD**

**Supervisor Name: Amjad Iqbal Khan**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| **05/05/2024** | **1.0** | SRS document includes scope of the project, Functional Requirements, Non-Functional Requirements, Use Case Diagram, Usage Scenarios, Adopted Methodology, Work Plan/ Gannt Chart.  The Full Stack Job Portal aims to provide a comprehensive platform for users to search, apply, and manage job listings. Utilizing Next.js, Strapi, and Postgres, the software ensures robustness and scalability. With a REST API architecture, users can easily interact with the system to find suitable job opportunities and manage their profiles efficiently. This document outlines the key requirements, functionalities, and roles associated with the development and deployment of the Full Stack Job Portal. | **BC210200939** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

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

**SRS Document**

1. **Scope of Project:**

The scope of the project includes the development of a Full Stack Job Portal using Next.js, Strapi, and Postgres. It encompasses features such as user registration, job posting, job search and filtering, profile management, application submission, administrative roles, authentication, and deployment. Additionally, the project involves integrating with external services such as Cloudinary for file uploads and deploying the application on platforms like Railway and Vercel. The project aims to deliver a robust, scalable, and user-friendly platform for job seekers and recruiters.

**Functional and non-Functional Requirements:**

1. **Functional Requirements:**

* Users can register with the portal by providing the necessary information like username, email, password for applying for job.
* User can register his/her company for job posting (For Job posting user must have a firm or company if not then user cannot post jobs). User should provide the following details for posting jobs.

1. Company name/Firm name
2. Address of the company
3. Expected salary for each role.
4. Job type (Permanent, Contractual, Full time, Part time).
5. Experience required for the offered role. (1 year, 2 years etc. or Fresh).
6. Users can not apply jobs on expired jobs (after due date/last date).

* Complete Authentication with simple JWT.
* Protecting routes from unauthorized users.
* Registered users can upload their CVs.
* Register users can update their profiles including personal information, contact details and CV uploads.
* Different roles should be assigned to users with varying levels of access right to ensure proper management and security like Super Admin, Content Manager/ Frontend user.
* Content Editors/Managers have the authority to approve and publish job listings submitted by the user.
* User can search and filter job listings by using check boxes based on keywords, location, job type, education, experience and salary range.
* Users should be able to see the list of applied jobs.
* Registered companies can see their posted jobs and update the posted jobs and see the list of the candidates who applied to their posted jobs.
* Uploading files to Cloudinary (CVs).
  1. **Administrative Roles:**

1. **Super Admin:** Responsible for overall site management, user management, content moderation, and configuration settings.

**Access:** Super Admin have the full access to all admin functionalities.

1. **Content Editor / Manager:** Content creation, editing and publishing. Managing jobs, images, videos and other media. Ensuring content is accurate and up to date.

**Access:** Access to content management features.

1. **Frontend users:** Registered companies can enter their jobs on job portal and registered users can apply for jobs and can upload their CVs.

**2.2 Reporting and Analytics:**

* **Topic stats Generation:** The system generates statistics related to job searches for example if a user searches a job of a **Full Stack Developer** the system shows the user about total jobs available, average salary, maximum salary and minimum salary etc.
  1. **Deployment and Integration:**
* **Deployment Platforms:** The application will be deployed on Railway and Vercel platforms.
* **File Uploads (CVs):** CV upload will be facilitated through integration with Cloudinary.

**Non-Functional Requirements:**

* **Accessibility:** System must be accessible from all kind of web browsers. It is accessible from both desktop and mobile browser.
* **Performance:** Optimized rendering and caching mechanism should be implemented to minimize page load times.
* **Reliability:** System should be highly available, with minimal downtime for maintenance or updates.
* **Security:** User authentication and authorization mechanism should be robust to prevent unauthorized access to sensitive data.
* **Scalability:** The architecture should support horizontal scaling to accommodate an increasing number of users and jobs listing. Load balancing mechanism should be implemented to distribute traffic evenly across servers.

1. **Use Case Diagram(s):**



1. **Usage Scenarios:**

|  |  |  |
| --- | --- | --- |
| **Use Case Related to User Registration** | | |
| **Use Case Title** | **Register** | |
| **Use Case Id** | UC-01 | |
| **Actors** | User | |
| **Description:**   * This use case will be used for creating a new user account. * User will provide all the required information for registration. | | |
| **Pre-Condition:**   * The Website is running and active with live database. | | |
| **Task-Sequence:**   * User will access the website. * Navigate to the registration page. * Fill out registration form. * Submit the Sign-up form. | | **Exceptions** |
| * Invalid Input field. * User already exists. |
| **Post-Condition:**   * User Account created Successfully. | | |
| **Created History:** 10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to User Login** | | |
| **Use Case Title** | **Login** | |
| **Use Case Id** | UC-02 | |
| **Actors** | User | |
| **Description:**   * This use case will be used for Logging In to the accounts. * Existing users logs into their account on the Job Portal system. | | |
| **Pre-Condition:**   * Website is running and active with live database. * A user must have an account to login. | | |
| **Task-Sequence:**   * Navigate to the Login Page. * Enter username and password. * Click on the Login button. | | **Exceptions** |
| * Incorrect username or password. |
| **Post-Condition:**   * User is Logged in Successfully. | | |
| **Created History:** 10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to User Logout** | | |
| **Use Case Title** | **Logout** | |
| **Use Case Id** | UC-03 | |
| **Actors** | User | |
| **Description:**   * This use case will be used for Log out. * Logged in users log out their account on the Job portal system. | | |
| **Pre-Condition:**   * User must be Logged in. | | |
| **Task-Sequence:**   * Users will navigate to the log out button. * Users click on log out button. * System logs out the user. | | **Exceptions** |
| **-** |
| **Post-Condition:**   * User is Logged out successfully. | | |
| **Created History:**10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Job searching** | | |
| **Use Case Title** | **Search Job** | |
| **Use Case Id** | UC-04 | |
| **Actors** | User | |
| **Description:**   * This use case will be used for job searching on online job portal. * User will search for jobs based on keywords and location. | | |
| **Pre-Condition:**   * Website must be live and connected to database. | | |
| **Task-Sequence:**   * User will navigate to job search page. * Enter keywords and locations. * Click on search button. | | **Exceptions** |
| * No matching jobs found. |
| **Post-Condition:**   * List of matching job listing is displayed. | | |
| **Created History:**10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Filter Job** | | |
| **Use Case Title** | **Filter Job** | |
| **Use Case Id** | UC-05 | |
| **Actors** | User | |
| **Description:**   * This use case is use for filtering jobs using checkboxes based on specified criteria. * Users can filter (using checkboxes) jobs on the basis of job type, education, experience, Salary Range. | | |
| **Pre-Condition:**   * Website is live and connected to database. | | |
| **Task-Sequence:**   * Navigate to job filtering section. * Select filter criteria. * Apply filters. | | **Exceptions** |
| * No jobs found matching filter criteria. |
| **Post-Condition:**   * List of filtered job listings is displayed. | | |
| **Created History:**10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Apply Job** | | |
| **Use Case Title** | **Apply Job** | |
| **Use Case Id** | UC-06 | |
| **Actors** | Registered Users | |
| **Description:**   * This use case will be used for applying for job. * User can apply for job by submitting their CVs. | | |
| **Pre-Condition:**   * User is logged in. * User has uploaded CV in his/her profile. | | |
| **Task-Sequence:**   * User creates an account. * User Login to their account. * User uploads their CVs. * User navigates to the job details page. * User search and filter jobs. * User clicks on “Apply now” button. * User submits the application. | | **Exceptions** |
| * CV upload fails. * User don’t have an account or logged in. * Application submission fails. |
| **Post-Condition:**   * Application is submitted successfully. | | |
| **Created History:**10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to View Applied Jobs** | | |
| **Use Case Title** | **View Applied Jobs** | |
| **Use Case Id** | UC-07 | |
| **Actors** | Registered User | |
| **Description:**   * This use case will be used to view the list of jobs users have applied. | | |
| **Pre-Condition:**   * User is Logged in. | | |
| **Task-Sequence:**   * Logged in user navigate to the profile. * Select “Applied Jobs” section. | | **Exceptions** |
| * No applied jobs found. |
| **Post-Condition:**   * List of applied jobs is displayed. | | |
| **Created History:** 10-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Update Profile** | | |
| **Use Case Title** | **Update profile** | |
| **Use Case Id** | UC-08 | |
| **Actors** | Registered user | |
| **Description:**   * This use case will be used for updating profile information such as contact details, CVs etc. | | |
| **Pre-Condition:**   * User is logged in. | | |
| **Task-Sequence:**   * Navigate to the profile. * Update profile information. * Save changes. | | **Exceptions** |
| * Invalid input field. |
| **Post-Condition:**   * Profile information is updated successfully. | | |
| **Created History:**13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to View Posted Jobs** | | |
| **Use Case Title** | **View Posted Jobs** | |
| **Use Case Id** | UC-09 | |
| **Actors** | Registered Company | |
| **Description:**   * Registered companies view the list of jobs the have posted on Job Portal. * This will display the list of all the jobs company has posted on portal. | | |
| **Pre-Condition:**   * User is logged in as a registered company. | | |
| **Task-Sequence:**   * Navigate to the profile. * Select “Posted Jobs” section. | | **Exceptions** |
| * No posted jobs found. |
| **Post-Condition:**   * List of posted jobs is displayed. | | |
| **Created History:** 13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Uploading CVs** | | |
| **Use Case Title** | **Upload CV** | |
| **Use Case Id** | UC-10 | |
| **Actors** | Registered users | |
| **Description:**   * User uploads their CV file to their profile for job application. | | |
| **Pre-Condition:**   * User is logged in. | | |
| **Task-Sequence:**   * Navigate to profile. * Select option to upload CV. * Choose file to upload. * Confirm upload. | | **Exceptions** |
| * Invalid file format. |
| **Post-Condition:**   * CV uploaded Successfully. | | |
| **Created History:** 13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Adding Jobs** | | |
| **Use Case Title** | **Add Jobs** | |
| **Use Case Id** | UC-11 | |
| **Actors** | Registered Company | |
| **Description:**   * Registered Companies adds a new job listing to the job portal. | | |
| **Pre-Condition:**   * User is logged in as a registered company. | | |
| **Task-Sequence:**   * Navigate to “Add Job” section. * Fill out job details form. * Click Add Job Button. | | **Exceptions** |
| * Invalid input fields. |
| **Post-Condition:**   * Job added successfully. | | |
| **Created History:**13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Update Job** | | |
| **Use Case Title** | **Update Jobs** | |
| **Use Case Id** | UC-12 | |
| **Actors** | Registered Company | |
| **Description:**   * Registered companies edit details of a job listings they have posted. | | |
| **Pre-Condition:**   * User is logged in as a Registered company. | | |
| **Task-Sequence:**   * Navigate to “Update Jobs” section. * Select job to update. * Click update job button. | | **Exceptions** |
| * Job not found. |
| **Post-Condition:**   * Job details are updated successfully. | | |
| **Created History:**13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to View Candidates** | | |
| **Use Case Title** | **View Candidates** | |
| **Use Case Id** | UC-13 | |
| **Actors** | Registered Company | |
| **Description:**   * This use case will be used for viewing the list of candidates who have applied for their job listing. | | |
| **Pre-Condition:**   * User is logged in as Registered company and is the owner of the job. | | |
| **Task-Sequence:**   * Navigate to the “View Candidates section” * Select job to view candidates. | | **Exceptions** |
| * No candidates found. |
| **Post-Condition:**   * List of candidates who applied is displayed. | | |
| **Created History:**13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Approve Jobs** | | |
| **Use Case Title** | **Approve Jobs** | |
| **Use Case Id** | UC-14 | |
| **Actors** | Content Manager/Editor | |
| **Description:**   * Content Manager approves job listings posted by registered companies. | | |
| **Pre-Condition:**   * Content Manager is logged in. | | |
| **Task-Sequence:**   * Navigate to “Approve Jobs” section. * Select Job to Approve. | | **Exceptions** |
| * Job not found. |
| **Post-Condition:**   * Job listing is approved successfully | | |
| **Created History:**13-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Publish Jobs** | | |
| **Use Case Title** | **Publish Jobs** | |
| **Use Case Id** | UC-15 | |
| **Actors** | Content Manager/Editor | |
| **Description:**   * Content Manager publishes approved jobs listings to make them visible to users. | | |
| **Pre-Condition:**   * Content Manager is Logged in. | | |
| **Task-Sequence:**   * Navigate to “Publish Jobs section”. * Select job to publish. | | **Exceptions** |
| * Job not found. |
| **Post-Condition:**   * Job published successfully. | | |
| **Created History:**14-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to See overall stats** | | |
| **Use Case Title** | **See over all stats** | |
| **Use Case Id** | UC-16 | |
| **Actors** | Super Admin | |
| **Description:**   * The super admin wants to view comprehensive statistics about the job portal system. | | |
| **Pre-Condition:**   * Super Admin is Logged in. | | |
| **Task-Sequence:**   * Super Admin logs into the Admin Dashboard. * Super Admin navigates to the overall stats section. * Super Admin views statistics such as total number of users, number of job listings etc. | | **Exceptions** |
| * None |
| **Post-Condition:**   * Super Admin gains insights into the overall performance and usage of the job portal system. | | |
| **Created History:**14-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to User Management** | | |
| **Use Case Title** | **User Management** | |
| **Use Case Id** | UC-17 | |
| **Actors** | Super Admin | |
| **Description:**   * The Super Admin manage user accounts and permissions. | | |
| **Pre-Condition:**   * Super Admin is logged in. | | |
| **Task-Sequence:**   * Super Admin logs into the admin dashboard. * Super Admin Navigates to the user management section. * Super Admin performs actions like adding new users, modifying user details or deactivating user accounts. | | **Exceptions** |
| * None |
| **Post-Condition:**   * User accounts are managed successfully by the super admin. | | |
| **Created History:**14-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Content Moderation** | | |
| **Use Case Title** | **Content Moderation** | |
| **Use Case Id** | UC-18 | |
| **Actors** | Super Admin | |
| **Description:**   * The Super Admin moderate user generated content. | | |
| **Pre-Condition:**   * Super Admin is logged in. | | |
| **Task-Sequence:**   * Super Admin logs into the admin dashboard. * Super Admin navigates to the "Content Moderation" section. * Super Admin reviews and moderate user-generated content. | | **Exceptions** |
| * None |
| **Post-Condition:**   * User-generated content is moderated and aligned with community guidelines by the Super Admin. | | |
| **Created History:**14-05-2024 | | |
| **Author:** BC210200939 | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Related to Configuration Setting** | | |
| **Use Case Title** | **Configuration Setting** | |
| **Use Case Id** | UC-19 | |
| **Actors** | Super Admin | |
| **Description:**   * The Super Admin configure system settings and parameters. | | |
| **Pre-Condition:**   * Super Admin is logged in. | | |
| **Task-Sequence:**   * Super Admin logs into the admin dashboard. * Super Admin navigates to the "Configuration Setting" section. * Super Admin modifies settings related to email notifications, security policies, etc. | | **Exceptions** |
| * None |
| **Post-Condition:**   * System settings and parameters are configured as per the requirements by the Super Admin. | | |
| **Created History:**14-05-2024 | | |
| **Author:** BC210200939 | | |

1. **Adopted Methodology**

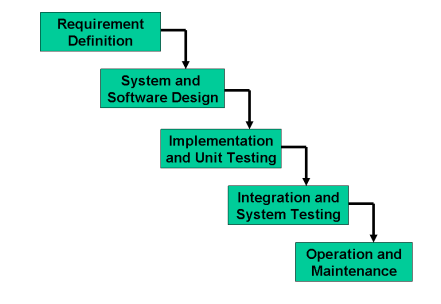
Software development methodology is defined as a framework for developing information systems, focusing on planning and organization. It benefits both teams and customers by improving efficiency and adaptability to changes.

Software engineering methodologies are important for creating better software projects. It helps developers plan, produce, and test software. These methods simplify tasks, improve collaboration, and achieve timely and budget-friendly outcomes.

**VU Process Model:**

VU Process Model is combination of Waterfall and Spiral model.

**Waterfall Model:**



**Requirement Analysis and Definition:**

What - The systems services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification. **2. System and Software Design:**

How – The system design process partitions the requirements to either hardware of software systems. It establishes and overall system architecture. Software design involves fundamental system abstractions and their relationships.

**Implementation and Unit Testing:**

How – During this stage the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specifications. **4. Integration and system testing:**

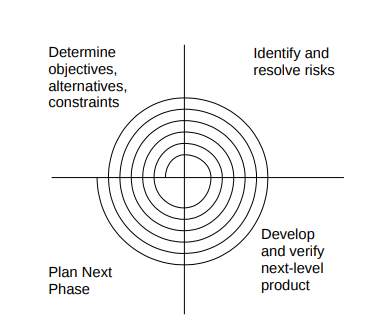
The individual program unit or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

1. **Operation and Maintenance:**

Normally this is the longest phase of the software life cycle. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in earlier stages of the life-cycle, improving the implementation of system units and enhancing the system’s services as new requirements are discovered.

**Spiral Model:**

This model was developed by Barry Boehm. The Spiral Model is Waterfall model plus risk analysis. In this case each stage is preceded by identification of alternatives and risk analysis and is then followed by evaluation and planning for the next phase. If risks cannot be resolved, project is immediately terminated.



**Determine objectives, alternatives, constraints:**

Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

**Identify and resolve Risks:**

During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.

**Develop and verify next-level product:**

During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

**Plan Next Phase:**

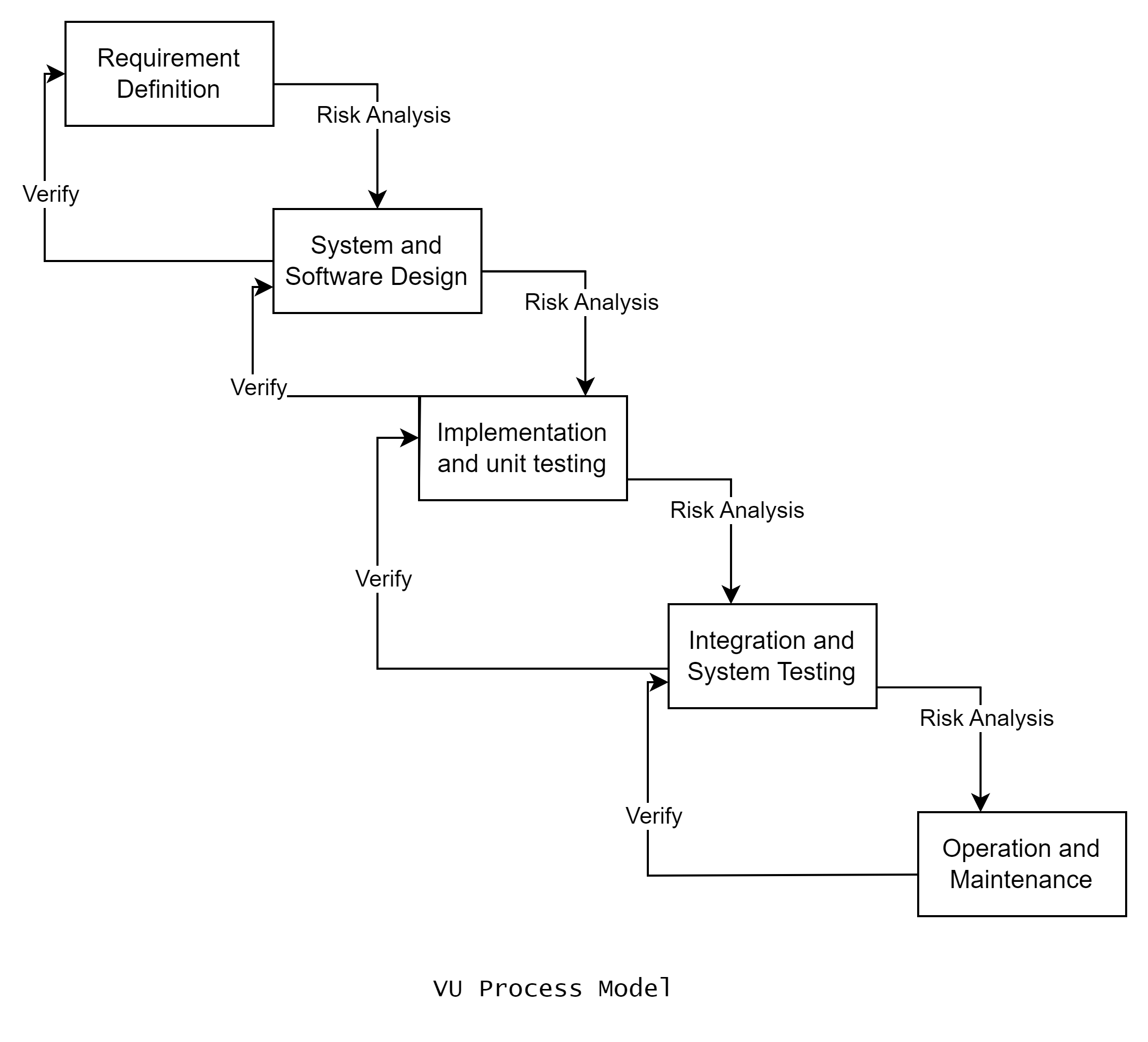
In the fourth quadrant, the Customers evaluate the so-far developed version of the software. In the end, planning for the next phase is started.

**Chosen Methodology:**

I will choose **VU Process Model** for this project.

**VU Process Model:**

* It is the combination of waterfall and spiral model. It is also called hybrid approach of system development.
* It has five phases and which are Gathering and analyzing requirement, planning, analysis and design, development and final report.
* In the requirement phase the system goals, services and constrains are established after discussion with user.
* The planning phase is synchronized effort to find all the potential roadblocks and draft a robust plan to tackle each.
* This phase usually involves a lot of brainstorming within software development team as well as with respective client.
* It maximizes the quality of system and minimizes any risks and errors.



**Reasons for choosing this Methodology:**

Following are the basic reasons for choosing VU Process Model for our project.

* Our project is divided into different phases which we have to complete in sequence and submit to our supervisor.
* If some mistakes are found than he suggests us to correct that mistake and improve our project. This process will be adopted due to its spiral nature.
* When the phase is well worked-out, and also accepted by our supervisor then we will proceed to next phase. This will be done due to the waterfall nature of the VU Process model.
* Hence both models are suppressed in single form and that will be our VU process model. It will provide error free result as each step is done in a sequence.

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

