ResearchHub

**Software Requirements Specification**

Version 1.0



**Group Id: S2402AD8B5**

**Supervisor Name:** **Syed Aun Ali Bukhari**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| 20-May-24 | 1.0 | A Software Requirement Specification (SRS) is a document that provides a comprehensive description of how a system will function. It outlines both the functional and non-functional requirements and includes a set of use cases that detail user interactions that the software must support. Additionally, the SRS incorporates usage scenarios and the methodology adopted for the project. | BC190402560 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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**

Scope of Project:

ResearchHub is a web application designed to facilitate collaboration and knowledge sharing among researchers, academics, and students. It provides a platform where users can upload, share, discuss, and discover research papers, articles, datasets, and other scholarly resources. By the end of the project, we will have developed a fully functional research sharing platform that enables users to collaborate, share, and discover research resources effectively. This project will enhance your web development skills while providing a valuable tool for researchers and academics to advance their scholarly work through collaboration and knowledge sharing.

Functional and non-Functional Requirements:

**Functional Requirements:**

1. User Registration and Authentication**:**

Allow users to register accounts and log in securely. Implement authentication mechanisms such as email verification and password hashing to ensure user data security.

1. AdminDashboard**:**

Create an admin dashboard for managing users, content moderation, enforcing community guidelines, and monitoring platform activity.

1. ProfileManagement**:**

Enable users to create and manage their profiles. Include features such as profile pictures, bio descriptions, affiliations, and research interests.

1. ResourceUploadandSharing**:**

Allow users to upload research papers, articles, datasets, presentations, and other scholarly resources. Implement file upload functionality with support for various file formats.

1. ResourceDiscovery**:**

Provide a search functionality for users to discover relevant research resources. Implement search filters based on keywords, authors, categories, publication dates, and citation metrics.

1. DiscussionForums**:**

Create discussion forums where users can engage in scholarly discussions, ask questions, share insights, and collaborate on research topics. Implement features such as threaded comments, upvoting/downvoting, and tagging.

1. CollaborativeProjects**:**

Enable users to create and join collaborative research projects. Each project can have its own discussion forum, document repository, task board, and timeline for milestones.

1. SocialNetworkingFeatures**:**

Implement social networking features such as following/followers, user mentions, and notifications to keep users engaged and connected with their peers.

1. AnalyticsandReporting**:**

Provide analytics dashboards for users to track metrics related to their research contributions, such as download counts, citation counts, and user engagement statistics.

**Non-Functional Requirements**:

1. Performance

* Response Time: The system should respond to user requests within 3 seconds under normal load conditions.
* Scalability: The application must handle up to 10,000 concurrent users without performance degradation.
* Throughput: The system should support the uploading and downloading of up to 1,000 files per minute.

1. Reliability

* Uptime: The application must maintain an uptime of 99.9%, ensuring it is available to users almost all the time.
* Fault Tolerance: The system should handle hardware or software failures gracefully, with minimal disruption to users.

1. Security

* Data Privacy: User data must be protected using encryption protocols during transmission and storage.
* Authentication and Authorization: The system must enforce strong authentication mechanisms and role-based access control to secure sensitive information and resources.

1. Usability

* User Interface: The application should have an intuitive and user-friendly interface, with consistent navigation and accessible design.
* Accessibility: The system should comply with WCAG 2.1 guidelines to ensure accessibility for users with disabilities.

1. Maintainability

* Code Quality: The codebase should follow best practices and coding standards to ensure it is readable, modular, and well-documented.
* Error Handling: The system should provide informative error messages and handle exceptions gracefully to facilitate troubleshooting and maintenance.

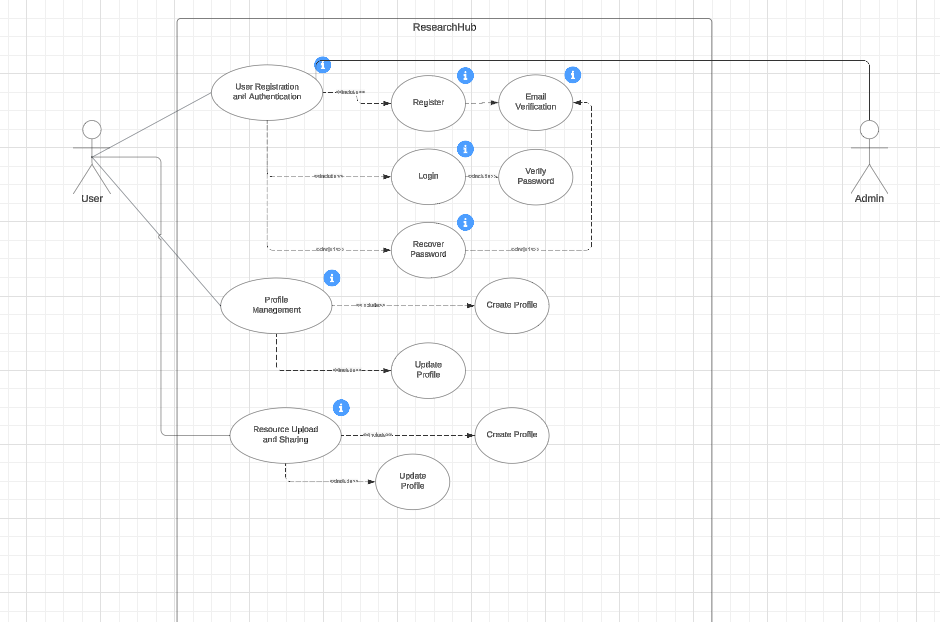
1. Compatibility

* Browser Compatibility: The application should be compatible with the latest versions of major web browsers (Chrome, Firefox, Safari, and Edge).
* Platform Independence: The system should be accessible on various devices, including desktops, laptops, tablets, and smartphones.

1. Backup and Recovery

* Data Backup: Regular backups of all data should be performed daily to prevent data loss.
* Recovery: In case of data loss or system failure, the system should be able to recover data within 1 hour.

Use Case Diagram(s):



Usage Scenarios:

Usage Scenario 1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case* related *to user registration and authentication*** | | | |
| Use case title | | Register user | | |
| Abbreviated title | | Reg\_user | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User | | |
| Description | | This use case enable user to register himself. | | |
| Pre-conditions | | Internet connection should be available. | | |
| Task sequence | | | | Exceptions |
| * New User Registration  1. User navigates to the registration page. 2. User fills out the registration form with required details. 3. User submits the registration form. 4. System sends a verification email to the user's provided email address. 5. User clicks on the verification link in the email to verify their email address. 6. User's account is successfully registered and verified.  * User Log In:  1. Registered user navigates to the login page. 2. User enters their credentials (email and password) into the login form. 3. User clicks the login button. 4. System verifies the credentials. 5. If credentials are correct, the user is logged into their account. 6. If credentials are incorrect, an error message is displayed.  * Email Verification:  1. User receives an email with a verification link after registration. 2. User clicks on the verification link. 3. System verifies the link and marks the user's email as verified. 4. User is redirected to the ResearchHub platform with their email verified.  * Password Recovery:  1. User clicks on the "Forgot Password" link on the login page. 2. User enters their email address in the password recovery form. 3. System sends a password reset link to the user's email. 4. User clicks on the password reset link. 5. User enters a new password and confirms it. 6. System updates the user's password in the database. | | | | 1. User has entered the invalid email phone number. |
| Post  condition | | | User has registered successfully. | |
| Author | | | BC190402560 | |

Usage Scenario 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to profile management*** | | | |
| Use case title | | Profile Management | | |
| Abbreviated title | | Prof\_Manag | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User | | |
| Description | | This use case enable user to update its own profile data. | | |
| Pre-conditions | | Internet connection should be available. | | |
| Task sequence | | | | Exceptions |
| * Create/Edit Profile:  1. A user creates their profile by providing personal information such as profile picture, bio, affiliations, and research interests. 2. The user can later edit their profile to update any information or make changes as needed.  * View Profile:  1. Users can view their own profile as well as profiles of other users on the platform. 2. Profiles display user information, uploaded resources, and activity history. | | | |  |
| Post  condition | | | User Profile updated successfully | |
| Author | | | BC190402560 | |

Usage Scenario 3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to Resource Upload and Sharing*** | | | |
| Use case title | | Resource Upload and Sharing | | |
| Abbreviated title | | Resou\_upload\_ad\_shr | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User, Admin | | |
| Description | | This use case enables user and admin to manage project/repository. | | |
| Pre conditions | | Internet connection should be available. | | |
| Task sequence | | | | Exceptions |
| * Uploading Research Resources:  1. User navigates to the upload page. 2. User selects the type of resource to upload (research paper, article, dataset, presentation). 3. User fills out the upload form with resource details (title, authors, abstract, etc.). 4. User uploads the resource file. 5. System validates and saves the uploaded resource. 6. Resource is successfully uploaded and added to the user's profile.  * Sharing Research Resources:  1. User selects the resource they want to share from their profile. 2. User clicks on the share button. 3. System generates a shareable link for the resource. 4. User can share the link with others via email, social media, or other communication channels. 5. Recipients can access and view the shared resource on ResearchHub. | | | |  |
| Post  condition | | | User has uploaded resource successfully.  Admin has managed uploaded project/repository successfully. | |
| Author | | | BC190402560 | |

Usage Scenario 4:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to resource discovery*** | | | |
| Use case title | | Resource Discovery | | |
| Abbreviated title | | Resour\_disco | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User | | |
| Description | | This use case enables the user to find his required project/repository. | | |
| Pre conditions | | Internet connection should be available. User logged in. | | |
| Task sequence | | | | Exceptions |
| 1. User enters keywords or selects filters on the search page. 2. User submits the search query. 3. System retrieves relevant resources based on the search query. 4. User can browse through the search results and view details of individual resources. | | | |  |
| Post  condition | | | User has found his project/repository successfully. | |
| Author | | | BC190402560 | |

Usage Scenario 5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to discussion forums*** | | | |
| Use case title | | Discussion Forums | | |
| Abbreviated title | | Discus\_frms | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | Donor | | |
| Description | | This use case enables the users to post their views on the project/repository. | | |
| Pre conditions | | Internet connection should be available. | | |
| Task sequence | | | | Exceptions |
| 1. Users engage in scholarly discussions, ask questions, share insights, and collaborate on research topics within discussion forums. 2. They can create new threads, post comments, upvote/downvote content, and tag discussions. | | | |  |
| Post  condition | | | User has successfully posted his comments on the project/repository. | |
| Author | | | BC190402560 | |

Usage Scenario 6:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to collaborative projects*** | | | |
| Use case title | | Collaborative Projects | | |
| Abbreviated title | | Collaborat\_projts | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User | | |
| Description | | This use case enables the user to create or join any collaborative project | | |
| Pre conditions | | Internet connection should be available. User logged in. | | |
| Task sequence | | | | Exceptions |
| 1. Users create collaborative research projects or join existing projects to collaborate with peers. 2. Each project may have its own discussion forum, document repository, task board, and timeline for milestones. | | | |  |
| Post  condition | | | User has joined or created collaborative project. | |
| Author | | | BC190402560 | |

Usage Scenario 7:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to social networking features*** | | | |
| Use case title | | social networking features | | |
| Abbreviated title | | Social\_networking | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User | | |
| Description | | This use case enables the user to engage/give reaction with/to the project/repository. | | |
| Pre conditions | | Internet connection should be available. User logged in. | | |
| Task sequence | | | | Exceptions |
| * Follow/Unfollow Users:  1. Users can follow other users to stay updated on their activities and contributions. 2. They can also unfollow users if they no longer wish to receive updates from them.  * Mention Users:  1. Users can mention other users in comments or discussions to draw their attention to specific content or topics.  * Receive Notifications:   + - 1. Users receive notifications for various activities such as mentions, new comments, new followers, and updates on collaborative projects. | | | |  |
| Post  condition | | | User has successfully engaged with projects/ repository. | |
| Author | | | BC190402560 | |

Usage Scenario 8:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to analytics and reporting*** | | | |
| Use case title | | Analytics and Reporting | | |
| Abbreviated title | | Anly\_&\_repo | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User, Admin | | |
| Description | | This use case enables the user and admin to track engagement to their project/repository. | | |
| Pre conditions | | Internet connection should be available. User logged in. | | |
| Task sequence | | | | Exceptions |
| * View Download Counts:  1. Users can view the number of times their uploaded resources (papers, articles, datasets, presentations) have been downloaded. 2. This helps users gauge the popularity and reach of their contributions.  * View Citation Counts:  1. Users can see how many times their uploaded resources have been cited by other users. 2. Citation counts provide insight into the impact and influence of their research within the community.  * View User Engagement Statistics:  1. Users have access to dashboards that display various engagement metrics such as views, likes, comments, and shares on their resources and profile. 2. This information helps users understand how their content is being interacted with and the level of engagement from the community.  * Admin Analytics Dashboard:  1. Admins can access an analytics dashboard to monitor platform-wide metrics such as total number of users, active users, new registrations, and overall resource uploads. 2. This helps admins track the growth and activity of the platform.  * User Activity Reports:  1. Admins can generate reports on individual user activity, including login frequency, resource uploads, and participation in discussions. 2. These reports help identify highly active users as well as those who might need assistance or encouragement to engage more.  * Content Performance Reports:  1. Admins can generate reports on the performance of uploaded content, including download counts, citation counts, and engagement metrics. 2. This helps identify popular and impactful resources and guides content moderation and promotion efforts.  * Community Health Metrics:  1. Admins can view metrics related to community health, such as the number of discussions, active projects, and reported issues. 2. This information is crucial for maintaining a positive and productive community environment.  * Resource Usage Trends:  1. Both users and admins can analyze trends in resource usage over time, such as peaks in downloads or citations during certain periods. 2. This helps users understand the temporal impact of their work and admins to plan for high-traffic periods.  * Custom Reports:  1. Admins can create custom reports based on specific criteria, such as user demographics, types of resources uploaded, or engagement levels in different research areas. 2. Custom reports provide tailored insights to support strategic decisions and platform improvements.  * Notification of Key Metrics:  1. Users receive notifications about key metrics such as milestones in download counts, citation counts, or significant changes in engagement metrics. 2. Admins receive alerts about critical platform metrics, such as spikes in activity or potential issues detected through anomaly detection in usage patterns. | | | |  |
| Post  condition | | | User has successfully watched with projects/ repository. | |
| Author | | | BC190402560 | |

Usage Scenario 9:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Use case related to admin dashboard*** | | | |
| Use case title | | Analytics and Reporting | | |
| Abbreviated title | | Anly\_&\_repo | | |
| Use case ID | | ABBN\_FR\_1 | | |
| Actors | | User, Admin | | |
| Description | | This use case enables the user and admin to track engagement to their project/repository. | | |
| Pre conditions | | Internet connection should be available. User logged in. | | |
| Task sequence | | | | Exceptions |
| * Manage Users:  1. Admins can manage user accounts, including approving new registrations, blocking problematic users, and deleting inactive accounts.  * Moderate Content:  1. Admins review and moderate content uploaded to the platform to ensure compliance with community guidelines and quality standards.  * Enforce Community Guidelines:  1. Admins enforce community guidelines by taking actions against users who violate platform rules or engage in inappropriate behavior.  * Monitor Platform Activity:  1. Admins monitor platform activity, including user interactions, resource uploads, and discussion forum participation, to identify trends and issues.  * Generate Reports:  1. Admins generate reports on platform usage, user engagement, and content statistics to analyze platform performance and make informed decisions. | | | |  |
| Post  condition | | | User has successfully watched with projects/ repository. | |
| Author | | | BC190402560 | |

Adopted Methodology

Software Development Methodologies

Software development methodologies are also known as system development methodologies. Actually, this is system development life cycle, or it can be called System Development Process. Software development methodologies are frames that are used tostructure, control and plan the process of developing system.

There are several models for such processes each describing approach to different variety of tasks, or activities that take place during the process. The names of the some of the methodologies are as follows:

* Agile software Development
* Incremental model
* Crystal Methods
* Extreme programming
* Waterfall model
* Spiral model

My adopted methodology is combination of these two given methodologies as follows:

Waterfallmethodology:

The first published model of the software development process was derived from other engineering processes. Because of the cascade from one phase to another, this model is known as the waterfall model. This model is also known as linear sequential model.

The waterfall model is a documentation-driven model. It therefore generates complete and comprehensive documentation and hence makes the maintenance task much easier. I t however suffers from facts that the client feedback is received when the product is finally delivered and hence any errors in the requirement specifications are not discovered until the product is sent to the client after completion. This therefore has major time and cost related consequences.

SpiralMethodology:

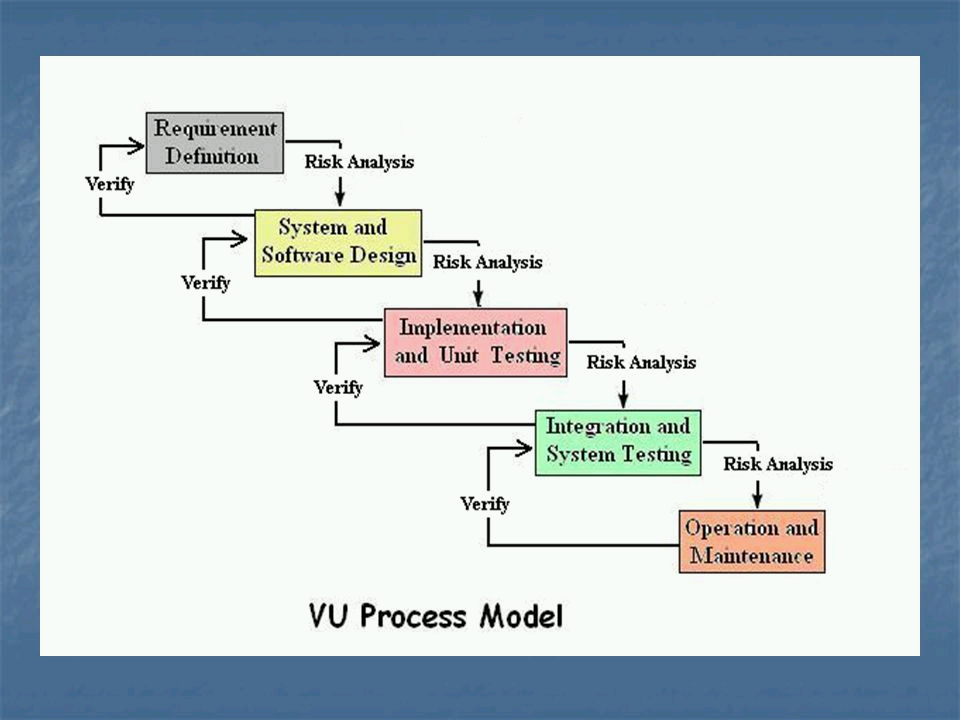
This model was developed by Barry Boehm. The main idea of this model is to avert risk as there is always an element of risk in development of software. For example, key personnel may resign at a critical juncture, the manufacturer of the software development may go bankrupt, etc.

In its simplified form, 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 immediately terminated.

The main strength of the Spiral Model comes from the fact that it is very sensitive to the risk. Because of the spiral nature of development it is easy to judge how much to test and there is no distinction between development and maintenance.

Each phase in Spiral model start with a design goal and end with a client reviewing the process.

AdoptedMethodology:



The above refers to the **VU process model**, which integrates elements from both the **waterfall** and **spiral models**. It consists of five phases: learning and analyzing requirements, creating the **Software Requirements Specification (SRS)**, Test Phase 1, Test Phase 2, and finally, delivering the final product. During the requirement phase, system goals, services, and constraints are established through discussions with users.

Benefitsofusingthismethodology**.**

* This is sequenced model
* This model is easy to understand and use
* Each phase has specific deliverable
* In this model phases are completed and processed one at a time.
* Avoidance of risk is enhanced.
* Additional functionality can be added later.

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