**SOFTWARE ENGINEERING**

**SOFTWARE REQUIREMENT SPECIFICATION**

**SOFTWARE ENGINEERING RESEARCH**

**LABORATORY WEBSITE**

## GROUP MEMBERS:

**KARISHMA MITRA [IIT2021204] RADHIKA GUPTA [IIT2021125] SHRUTI BILOLIKAR [IIT2021172] NAYSHA SINGH [IIT2021140] PANKTI SALVI [IIT2021134]**

# Introduction:

"SERL" is basically a website, which will enable users to access information related to faculty, researchers, publications, projects, courses etcetera of SERL lab.

## Purpose:

To provide detailed information about SERL lab.

### From User’s point of view:

* Only method of communication is university email, SMS.
* Users have to get in touch with the lab members to get the information related to lab.
* Sometimes, urgent intimation of any information is required to the user from the SERL lab or any publication.
* There is a lot of paperwork, users do have to visit the lab numerous times even for minute

requirements, leads to waste of time.

### From SERL lab member’s point of view:

* They have to contact users individually if they need any information.
* It becomes more difficult to deal with the large database.
* No special interface for publications, they have to mass mail the information or any news,

urgent intimations as well.

So, this project is an attempt to overcome these drawbacks.

## Scope:

We describe what features are in the scope of the website and what are not in the scope of the website to be developed.

*In Scope:*

1. Providing the latest information about publications.
2. Providing information about various projects.
3. Providing information about the courses provided.
4. Providing resources to the user.
5. Displaying the contact details of SERL lab authorities.
6. Giving detailed information about the faculty and researchers of SERL lab.

## Definitions, Acronyms, and Abbreviations:

*Acronyms and Abbreviations:*

1. SERL: Software Engineering Research laboratory.
2. SRS: Software Requirement Specification

*Definitions:*

a. SERL lab: A research laboratory specialized in software engineering.

## References:

IEEE SRS Format

## Overview:

The rest of this SRS is organized as follows: Section 2 gives an overall description of the website. It gives some general constraints while making the website and some

assumptions and dependencies that are assumed. Section 3 gives specific requirements which the website is expected to deliver. Functional requirements are given by various use cases. Some performance requirements and design constraints are also given.

# Overall Description:

## Product Perspective:

SERL website is aimed to propagate information of software Engineering reserach laboratory. It provides information about faculties and researchers who are actively working on projects and publications related to software engineering. The website also provides resources related to SERL for those who are interested. One-click feature of the website will save a lot of time of users who have to visit the SERL for small reasons at times.

## Product Functions:

SERL supports the following use cases:

|  |  |
| --- | --- |
| **Use cases** | **Description of use cases** |
| **Admin:** |  |
| Authorized login | Allows Admin to login. |
| Add Faculty | Allows Admin to add new Faculty. |
| Add Researcher | Allows Admin to add new Researcher. |
| Add Publication | Allows Admin to add new Publication. |
| Add Project | Allows Admin to add a new Project. |
| Add Course | Allows Admin to add a new Course. |
| View Feedback | Allows Admin to view feedback |
| Delete Course | Allows Admin to delete an existing Course. |
| Delete Project | Allows Admin to delete an existing Project. |
| Delete Publication | Allows Admin to delete an existing Publication. |
| Delete Faculty | Allows Admin to delete an existing Faculty. |
| Delete Researcher | Allows Admin to delete an existing Researcher. |
| **User:** |  |
| HomePage | Allows user to view home page of website. |
| Faculty | View faculty details. |
| Researcher | View researcher details. |
| Courses | View all the courses. |
| Publications | User can view links to all publications. |
| Projects | User can view description of all projects. |
| Resources | User can view links to various resources. |
| Contact | User can view contact details of SERL lab  authorities. |
| **Faculty:** |  |
| Authorized login | Allows faculty to login |
| Add publication | Allows faculty to add a new publication |
| Add project | Allows faculty to add a new project |
| **Researcher:** |  |
| Authorized login | Allows researcher to login |
| Add publication | Allows researcher to add a new publication |
| Add project | Allows researcher to add a new project |

## User Characteristics:

The user should be familiar with the operation of Web Application.

## Principal Actors:

The two principal actors in SERL website are “Admin”,” Faculty”,” Researcher”, and “User”.

## General Constraints:

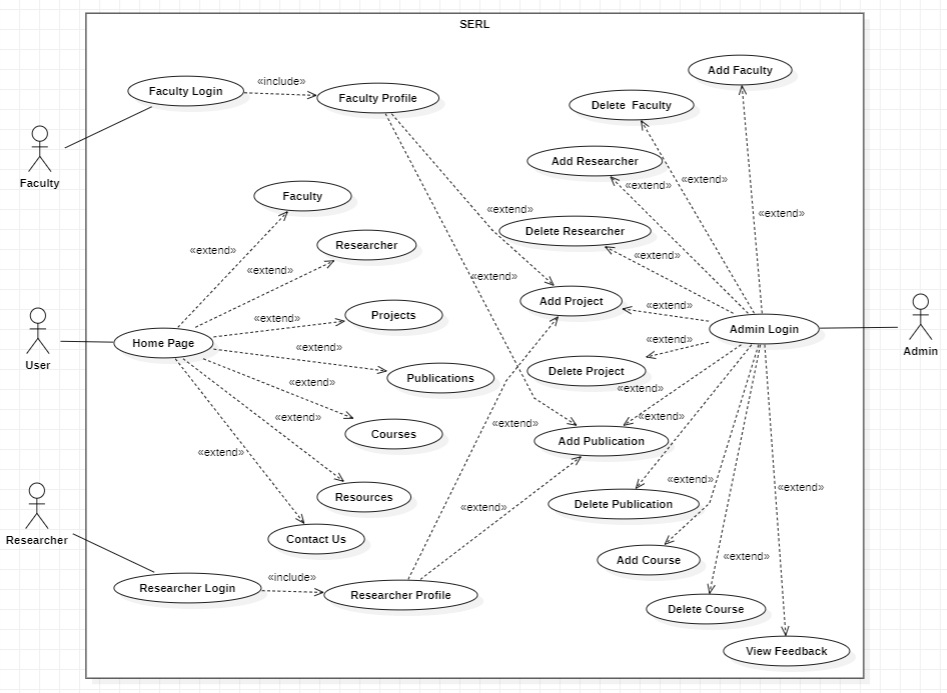
b. Working of SERL Website requires Internet connection.

c. SERL Website is a single-user Application.

## Assumptions and Dependencies:

1. Working of SERL website is dependent on the availability of Internet connection.
2. Admin must be active to insert all the required informations in the website.

# APPENDIX A: USE CASE DIAGRAM

****

1. **Specific Requirements:**

## Functional Requirements:

We describe the functional requirements by giving various use cases.

## Use Case 1:

**Name:** Authorized login **Summary:** Allows Admin to login. **Actors:** Admin,Faculty,Researcher

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on login button.
    - App checks for the authorization of login.

### Extension:

Id or password incorrect and shows error.

### Post-condition:

Admin can now access all features of the app.

## Use Case 2:

**Name:** Add Faculty

**Summary:** Allows Admin to add new Faculty.

**Actors:** Admin

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on Add Faculty button.
    - Admin fills in the required details of the new Faculty to be added.
    - Admin saves the details and the new faculty gets added.

### Extension:

Error in Fetching data from database.

### Post-condition:

Admin can now view details of new added faculty.

## Use Case 3:

**Name:** Add Researcher

**Summary:** Allows Admin to add new Researcher.

**Actors:** Admin

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on Add Researcher button.
    - Admin fills in the required details of the new Researcher to be added.
    - Admin saves the details and the new Researcher gets added.

### Extension:

Error in Fetching data from database.

### Post-condition:

Admin can now view details of new added Researcher.

## Use Case 4:

**Name:** Add Publication

**Summary:** Allows Admin to add new Publication.

**Actors:** Admin, Faculty, Researcher

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on Add Publication button.
    - Admin fills in the required details of the new Publication to be added.
    - Admin saves the details and the new Publication gets added.

### Extension:

Error in Fetching data from the database.

### Post-condition:

Admin can now view details of newly added Publication.

## Use Case 5:

**Name:** Add Project

**Summary:** Allows Admin to add new Project.

**Actors:** Admin, Faculty, Researcher

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on Add Project button.
    - Admin fills in the required details of the new Project to be added.
    - Admin saves the details and the new Project gets added.

### Extension:

Error in Fetching data from database.

### Post-condition:

Admin can now view details of new added Project.

## Use Case 6:

**Name:** Delete Faculty

**Summary:** Allows Admin to delete an existing Faculty.

**Actors:** Admin

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on delete Faculty button.
    - Admin is able to delete Faculty.

### Extension:

Error in deleting data from database.

### Post-condition:

Admin is able to remove the details of an existing faculty.

## Use Case 7:

**Name:** Delete Researcher

**Summary:** Allows Admin to delete an existing Researcher.

**Actors:** Admin

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on delete Researcher button.
    - Admin is able to delete Researcher.

### Extension:

Error in deleting data from database.

### Post-condition:

Admin is able to remove the details of an existing Researcher.

## Use Case 8:

**Name:** Projects

**Summary:** User can view description of all projects.

**Actors:** Any user

### Pre-conditions:

Internet connectivity.

### Main success scenario:

User is able to view all project details.

### Extensions:

NIL.

### Post-condition:

NIL.

## Use Case 9:

**Name:** Resources

**Summary:** User can view links to various resources.

**Actors:** Any user

### Pre-conditions:

Internet connectivity.

### Main success scenario:

User is able to view resources and their corresponding links.

### Extensions:

Access specific resource.

### Post-condition:

User is able to access the desired resource.

## Use Case 10:

**Name:** Contact

**Summary:** User can view contact details of SERL lab authorities.

**Actors:** Any user.

### Pre-conditions:

Internet connectivity.

### Main success scenario:

User gets the contact details of the desired person.

### Extensions:

NIL.

### Post-condition:

User gets the desired contact detail.

## Use Case 11:

**Name:** HomePage

**Summary:** Allows user to view home page of website

**Actors:** User

### Pre-conditions:

* + - Internet connectivity.

### Main success scenario:

* + - User is able to access homepage

### Post-condition:

User can now access all features of the website.

## Use Case 12:

**Name:** Faculty

**Summary:** View faculty details

**Actors:** Any user

### Pre-conditions:

* + - Internet connectivity.

Main success scenario:

* + - User is able to access faculty information

### Extension:

User is able to view faculty’s profile pages.

### Post-condition:

NIL

## Use Case 13:

**Name:** Researcher

**Summary:** View researcher details

**Actors:** Any user

### Pre-conditions:

* + - Internet connectivity.

### Main success scenario:

* + - User is able to access researcher information

### Extension:

User is able to view researcher’s profile pages.

### Post-condition:

NIL

## Use Case 14:

**Name:** Publications

**Summary:** User can view links to all publications.

**Actors:** Any user

### Pre-conditions:

Internet connectivity.

### Main success scenario:

* + - User is able to view all publications of SERL lab.

### Extensions:

Access specific publication.

### Post-condition:

User is able to open the desired publication link.

## Use Case 15:

**Name:** View Feedback

**Summary:** Admin can view all feedbacks

**Actors:** Admin

### Pre-conditions:

Internet connectivity.

### Main success scenario:

Admin is able to view all feedbacks.

### Extensions:

Access all feedbacks.

### Post-condition:

User is able to view desired feedback.

## Use Case 16:

**Name:** Add course

**Summary:** Allows Admin to add a new course

**Actors:** Admin

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks add course button.
    - Admin is able to add new course.

### Extension:

Error in adding data in the database.

### Post-condition:

Admin is able to add a new course.

## Use Case 17:

**Name:** Delete Course

**Summary:** Allows Admin to delete an existing Course.

**Actors:** Admin

### Pre-conditions:

* + - Internet connectivity.
    - Only for allowed mail ids.

### Main success scenario:

* + - Admin clicks on delete Course button.
    - Admin is able to delete Course.

### Extension:

Error in deleting data from the database.

### Post-condition:

Admin is able to remove the details of an existing Course.

## Use Case 18:

**Name:** Course

**Summary:** View all courses

**Actors:** Any user

### Pre-conditions:

* + - Internet connectivity.

Main success scenario:

* + - User is able to access all course-related information

### Extension:

User is able to view courses provided along with their details.

### Post-condition:

NIL

## Non-Functional Requirements:

* + Performance: Target response times can be defined, such as achieving a page load time of less than 2 seconds or ensuring that search queries are processed within 500 milliseconds.
  + Usability: Conduct user testing and gather feedback to evaluate the website's usability. This can include metrics such as task completion rates, user satisfaction scores, and average time to find desired information.
  + Maintainability: Encourage adherence to coding standards, such as clean code principles, proper code commenting, and consistent naming conventions. Specify the use of version control systems like Git to track code changes and facilitate collaboration among developers.
  + Compatibility: Determine the supported browsers and versions, such as the latest versions of Chrome, Firefox, Safari, and Edge. Identify the required operating systems, such as Windows, macOS, and Linux, along with the supported mobile platforms (iOS and Android).

## Hardware Requirements:

* Device: A computer or mobile device (e.g., laptop, desktop, smartphone, tablet) with a compatible operating system and web browser.
* Processor: A modern processor (e.g., Intel Core i3 or equivalent) capable of running the operating system and web browser smoothly.
* Memory (RAM): A minimum of 4 GB RAM is recommended for optimal performance, but higher RAM capacity (8 GB or more) may be beneficial for handling resource-intensive tasks or multiple applications simultaneously.
* Storage: Sufficient storage space to store the operating system, web browser, and other software applications used in conjunction with the SERL Lab website. A minimum of 128 GB of storage is typically recommended.
* Network Connectivity: A stable internet connection with sufficient bandwidth to access and interact with the SERL Lab website seamlessly.

## Software Requirements:

* Web Browser: Users should have a modern web browser that supports HTML5, CSS3, and JavaScript. Recommended browsers include Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, or Opera. It's essential to keep the browser updated to ensure compatibility with the latest web standards and features.
* Operating System: The SERL Lab website is compatible with various operating systems, including Windows, macOS, and Linux distributions. Users should have a compatible operating system installed on their devices.
* Internet Connectivity: Users need a stable internet connection to access the SERL Lab website and its resources. A reliable and sufficient bandwidth is recommended for seamless browsing and interaction.

## Design Constraints:

* Responsive Design: The website should be designed to provide a seamless and optimized user experience across different devices and screen sizes, including desktops, laptops, tablets, and smartphones. It should adapt its layout and content presentation dynamically based on the user's device.
* Accessibility: The website design should conform to accessibility standards, ensuring that individuals with disabilities can access and navigate the website effectively. This includes providing alternative text for images, supporting keyboard navigation, and ensuring proper color contrast for readability.
* Cross-Browser Compatibility: The website should be compatible with a range of modern web browsers, including popular options such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera. It should be tested and optimized to work consistently across these browsers.