A logo of a university

Description automatically generated

Alexandria university

Faculty of computers and data science

RRPM department website

**System design and architecture**

# Table of Contents

[Table of Contents 1](#_Toc2093508076)

[Revision history 2](#_Toc1668189797)

[Document approval 2](#_Toc1734844086)

[1. Introduction 2](#_Toc217357306)

[1.1. Purpose 3](#_Toc1467066280)

[1.2. Scope 3](#_Toc205517532)

[1.2.1. Product Identification 3](#_Toc1392647216)

[1.2.2. Product Capabilities 3](#_Toc1758181890)

[1.2.3. Application, Benefits, and Objectives 3](#_Toc1008652657)

[2. System Overview 3](#_Toc274161786)

[2.1. System Description 4](#_Toc1922185098)

[2.2. Key features 4](#_Toc838848027)

[2.3. Stakeholders 4](#_Toc1460129574)

[3. System architecture 4](#_Toc2136965579)

[3.1. Architecture diagram 5](#_Toc345469352)

[3.1.1. Sequence diagram 6](#_Toc653341792)

[3.2. Components Overview: 11](#_Toc339834913)

[4. Detailed design 13](#_Toc1169796007)

[4.1. Module description 14](#_Toc2141176916)

[4.2. Database Design 15](#_Toc39003367)

[4.2.1. Conceptual Entity-Relationship Diagram (ERD) 15](#_Toc796405355)

[4.2.2. Logical Data model 17](#_Toc1022173224)

[4.3. Ui design 19](#_Toc388272011)

[4.3.1. Wireframes 19](#_Toc896435701)

[4.3.1.2. Main pages (Home- About us – Units – scientific research – contact us) 19](#_Toc24797020)

[4.3.1.2. Secondary pages (single activity page – single scientific research page) 20](#_Toc1494256456)

[4.3.1.3. Admin pages (login page – Unit management page – Activity management page – scientific research management page) 21](#_Toc599217125)

[4.3.2. User Interaction Details for general user 22](#_Toc1313009784)

[4.3.3. User interaction details for Admin user 27](#_Toc956555590)

[5. Deployment 28](#_Toc286352322)

[5.1. Hosting 28](#_Toc804852865)

[5.1.1. Deployment Environment 29](#_Toc964767892)

[5.1.2. Deployment Strategy 29](#_Toc565635587)

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author(s)** | **Comments** |
| 3/12/2024 | Version 1 | * Ghada Essam * Lina Essam | - |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document approval

The following System design and architecture document has been accepted and approved by the following:

|  |  |
| --- | --- |
| **Status** |  |
| **Author(s)** |  |
| **Approver** |  |
| **Document type** |  |
| **Version** |  |
| **Release Date** |  |

# Introduction

## Purpose

The purpose of this document is to provide an overview of the system design and architecture for the RRPM department website. This includes a description of the system architecture, its components, and the database design.

The medical center website is designed to serve general users (patients) by providing information about the units, clinics, and activities. It also includes a collection of medical research articles that users can read.

## Scope

### Product Identification

The Rheumatology, Rehabilitation, and Physical Medicine (RRPM) Department Website aims to provide accurate information about rheumatology and physical health, raise public awareness, and connect patients with specialized medical units.

### Product Capabilities

* General Users:  
  Access information about the department, medical units (clinics, services, activities), organized events, scientific research, and contact details.
* Admin Users:  
  Manage the website content with tools for creating, reading, updating, and deleting information related to medical units, activities, research, and contact details.

### **Application, Benefits, and Objectives**

* Provide detailed medical unit information to help users locate services and clinics.
* Enhance public awareness about rheumatology, rehabilitation, and physical health.
* Empower admins with efficient content management tools.
* Offer reliable contact options for users to connect with the department and its units.

# System Overview

## System Description

The Rheumatology, Rehabilitation, and Physical Medicine (RRPM) Department Website is an online platform designed to provide accessible, reliable, and organized information about the department's units, services, and activities. It aims to serve two primary user groups:

1. General Users (Patients):
   * They can explore detailed information about the department, including individual medical units, clinics, available services, and scheduled activities.
   * The website also features a section for scientific research articles and department-related events, contributing to public health awareness and education.
2. Admin Users:
   * Admins are responsible for maintaining the website's content. They can manage (add, update, delete) all information displayed, including unit details, activities, research articles, and contact information.

The system's architecture ensures a user-friendly interface for visitors and efficient management tools for administrators. It supports secure access, providing general users with read-only access to public content and admin users with role-based control over content management.

By bridging the gap between patients and specialized medical care, the system promotes public awareness, efficient communication, and ease of access to the department’s resources.

## Key features

* Display information about units and clinics
* Display the activities organized by the units or the general department
* Display a collection of scientific research articles
* Display contact information for each unit
* Admin panel for managing content.

## Stakeholders

1. General users

Visitors to the website, including patients and the general public.

1. Administrators

Website admins responsible for managing and maintaining content.

# System architecture

## Architecture diagram

A diagram of a web browser

Description automatically generated

**MVC Diagram**

The Model-View-Controller (MVC) pattern is a software architectural design that separates an application into three interconnected components

* **Model:** Manages the data, logic, and rules of the application.
* **View:** Represents the user interface and displays data from the Model to the user.
* **Controller:** Acts as an intermediary between Model and View, handling user input and updating the Model and View accordingly.

**Browser**

* The browser serves as the entry point for users to interact with the application via HTTP requests.
* It sends requests to the Controller and receives responses, which are rendered as web pages.

**Controller**

* Receives HTTP requests from the Browser.
* Interacts with the Model to retrieve or update data.
* Sends the appropriate response back to the Browser via the View.

**View**

* Displays data and content to users in a structured format (web pages).
* The View dynamically updates based on data received from the Controller

**Model**

The Model handles the data-related logic and communicates with the database to perform CRUD (Create, Read, Update, Delete) operations.

* Responds to API calls from the Controller for data manipulation.
* Updates the database when necessary.
* Sends processed data back to the Controller.

**Database**

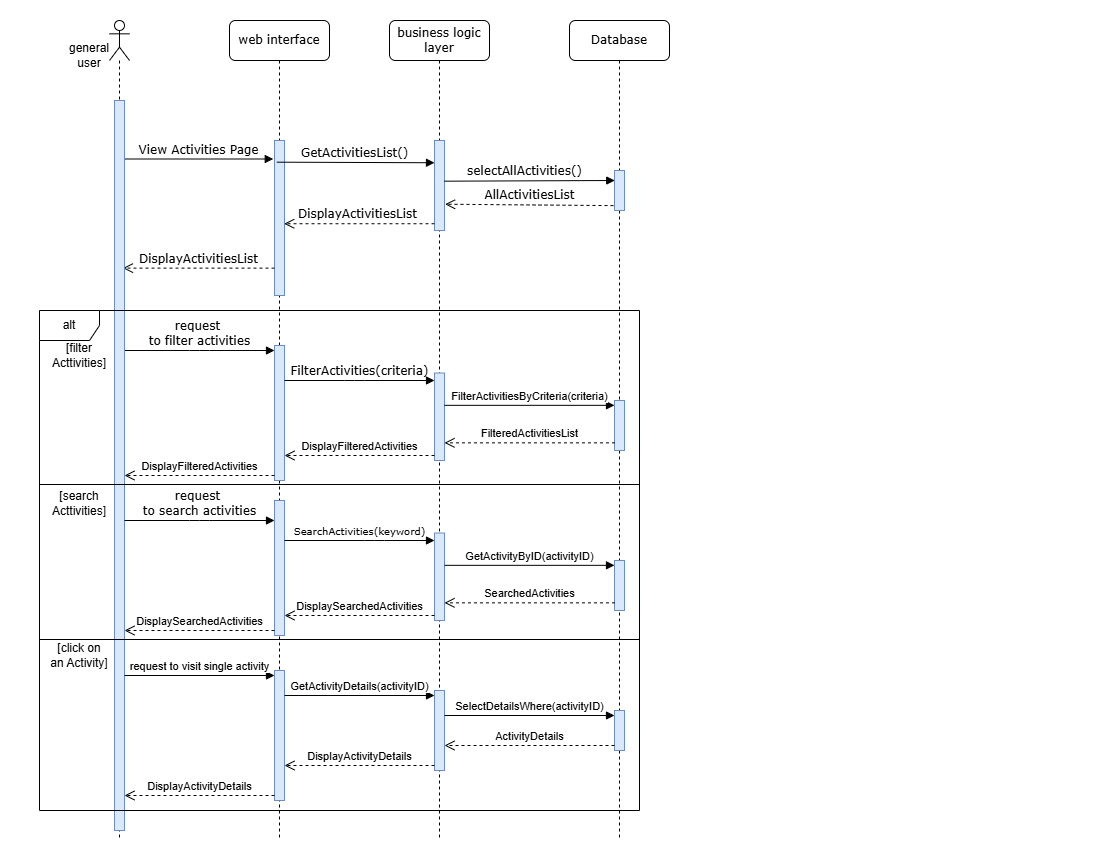
* Stores all persistent data for the application.
* Handles CRUD operations requested by the Model via the Data Access Layer.

**Flow of Operations**

* The user interacts with the browser (e.g., entering data in a form or clicking a button).
* The browser sends an HTTP request to the Controller.
* The Controller processes the request, verifies authentication, and communicates with the Model.
* The Model retrieves or updates data in the Database as necessary.
* The Controller sends the processed data to the View, which generates a response for the browser.
* The browser displays the content or page to the user.

### Sequence diagram

1. **For general user**



This sequence diagram demonstrates how a general user interacts with a web-based system to carry out various actions related to activities, including viewing, filtering, searching for activities, and accessing the details of a single activity.

The user's interaction with the activities page follows a repetitive pattern seen across other pages, where they primarily view, sort, and filter data.

**Actors:**

1. **General User**:
   * Represents the end-user interacting with the web interface to perform operations like viewing, filtering, searching, or selecting activities.

**Objects:**

1. **Web Interface**:
   * The front-end interface where the user interacts with the application through pages .
   * Responsible for displaying results and sending user requests to the business logic layer.
2. **Business Logic Layer**:
   * Represents the application's core logic responsible for processing user requests and retrieving/manipulating data. It acts as a bridge between the web interface and the database.
3. **Database**:
   * The storage system where all activity-related data is stored (activity details, search criteria, etc.).
   * Provides the data required by the business logic layer upon request.

**Functions:**

1. **ViewActivitiesPage()**:
   * Function called by the general user to access the activities page.
   * Initiates the retrieval of all available activities from the database.
2. **GetActivitiesList()**:
   * A function in the business logic layer that retrieves all activities from the database.
3. **selectAllActivities()**:
   * A database query that selects all records in the activities table and returns the results to the business logic layer.
4. **Filter Activities(criteria)**:
   * Functionality allowing the user to apply filters (by category, date) to narrow down the list of activities.
5. **FilterActivitiesByCriteria(criteria)**:
   * Logic in the business layer to apply the filter criteria and retrieve matching activities from the database.
6. **SearchActivities(keywords)**:
   * Functionality allowing the user to search activities by entering keywords.
7. **GetActivityByID(activityID)**:
   * Retrieves details of a single activity when the user selects one.
8. **SelectDetailsWhere(ActivityID)**:
   * A database query to fetch specific activity details based on the activity ID.

**Summary of Interactions**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Action** | **Web Interface Action** | **Business Logic Layer Action** | **Database Action** |
| View activities | Request activity list | Fetch all activities | Select all activities |
| Filter activities | Send filter criteria | Apply filter criteria | Query activities by criteria |
| Search activities | Send search keyword | Search matching activities | Query activities by keyword |
| View activity details | Request details by activity ID | Retrieve activity details by ID | Select activity details by ID |

A diagram with blue squares

Description automatically generated

1. **For Admin**

This sequence diagram illustrates how an admin user interacts with various system components to perform management tasks, including managing units, activities, and research. The interactions follow a structured workflow starting with authentication and proceeding to different management functionalities. (Added that the Admin user could do the same functionality for General user as usual )

**Actors**

1. **Admin**:
   * Represents the administrator who has access to privileged management features of the system. The admin performs operations such as logging in, managing units, activities, and research.

**Objects**

1. **Login Page**:
   * The web interface component where the admin enters login credentials (username and password) to access the system.
2. **Authentication Service**:
   * A service responsible for validating admin credentials and providing secure access to the management functionalities.
3. **Manage Units**:
   * A system module that allows the admin to perform Create, Read, Update, Delete (CRUD) operations on unit-related data.
4. **Activities Management Page**:
   * A system module dedicated to managing activities. It provides CRUD functionality for activity records.
5. **Research Management Page**:
   * A module that facilitates the management of research-related data, with CRUD operations available.
6. **Database**:
   * The underlying data storage system where all unit, activity, and research-related data is stored and retrieved.

**Functions**

1. **Login(username, password):**
   * Function that collects the admin's credentials and forwards them to the authentication service for validation.
2. **Validate():**
   * A process in the authentication service that checks the provided credentials against stored data in the database.
   * Returns a success or error message.
3. **CRUD:**
   * Represents Create, Read, Update, and Delete operations performed by the admin in the respective management modules:
     + Manage Units: CRUD operations for unit data.
     + Activities Management Page: CRUD operations for activities.
     + Research Management Page: CRUD operations for research data.
4. **Display Message:**
   * Displays a success or error message after performing an action to inform the admin of the result.

Summary of Interactions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Admin Action** | **Login Page Action** | **Authentication Service Action** | **Database Action** | **Module Response** |
| Login | Send credentials | Validate credentials | Fetch matching credentials | Display login success/error |
| Open Unit Management Page | Navigate to the page | Send CRUD request | Process CRUD operations | Display success/error |
| Open Activities Page | Navigate to the page | Send CRUD request | Process CRUD operations | Display success/error |
| Open Research Page | Navigate to the page | Send CRUD request | Process CRUD operations | Display success/error |

## Components Overview:

A diagram of a process flow

Description automatically generated

**Architecture Components**

**Front End (React)**

* The front-end is built using React and serves as the user interface of the application.
* It enables users to interact with the system through various pages.
* Sends requests to the back end using API calls to retrieve and manipulate data.

**Back End (Node.js)**

Acts as the server-side logic that processes requests from the front-end and interacts with the database.

**Components**:

* **API Gateway**: Handles incoming API calls, routes them to the correct functions, and responds to the front-end.
* **Authentication Module**: Manages user authentication and authorization, ensuring secure access.
* **Business Logic Layer**: Contains application-specific rules and logic for handling data and requests.

**External APIs**

* Enhance the application's functionality by integrating external services.
* The Google Maps API is used to fetch location data for features such as mapping or location-based search.

**Database**

* Stores and manages all application data
* CRUD (Create, Read, Update, Delete) operations are performed on these tables by the back end to fulfill user requests.

**Flow of Operations**

* The user interacts with the front-end interface (React) by navigating through pages and performing actions.
* The front end sends HTTP requests to the back-end API Gateway.
* A request to retrieve data from a page would be routed to the relevant handler in the business logic layer.
* The back-end processes the request:

Validates the input.

Authenticates the user through the Authentication Module.

Executes the required business logic.

Performs CRUD operations on the database, if necessary.

* If external data is needed (location data), the back-end fetches it from external APIs like Google Maps.
* The processed data or result is sent back to the front-end, where it is displayed to the user

# Detailed designA diagram of a computer program Description automatically generated with medium confidence

## Module description

**Three-Tier Architecture**

this architecture is particularly suited for systems that require scalability, modularity, and separation of concerns.

The **Three-Tier Architecture** consists of three distinct layers:

Separating these layers ensures that changes in one layer (UI redesign) do not affect other layers, improving maintainability

* **Presentation Layer (Front-End):** Responsible for the user interface and experience. In your case, this layer is represented by the React-based front-end with pages such as the Home Page, Activities Page, Research Page, and Contact Us Page.
* **Business Logic Layer (Application/Back-End):** Encapsulates the core functionality and processes. This is represented by the Node.js backend in your system, handling authentication, API calls, and the business logic layer.
* **Data Layer (Database):** Manages persistent data storage, including CRUD operations for entities like the User Table, Activities Table, Research Table, and Units Table.

The **Three-Tier Architecture** aligns with the system's non-functional requirements:

* **Scalability**: Handles concurrent requests and increasing loads.
* **Performance**: Front-end and back-end layers optimize user interaction and data processing.
* **Security**: Isolates layers to enhance data security.
* **Portability**: Ensures cross-platform compatibility with React and API-based communication.

**The non-functional sign in the diagram:**

**Performance**: Use a stopwatch icon next to client-side logic components and annotate with "Optimized for fast loading times".

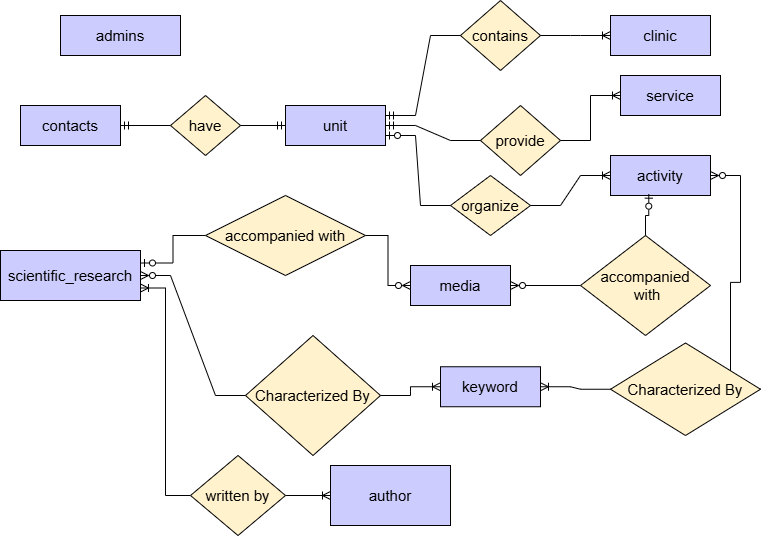
**Maintainability**: Use a document icon next to backend components and annotate with "Clear documentation and error handling".

**Portability**: Use a globe icon next to client-side logic components and annotate with "Compatible with multiple browsers and devices"(will be more clearly in the deployment).

## Database Design

### Conceptual Entity-Relationship Diagram (ERD)

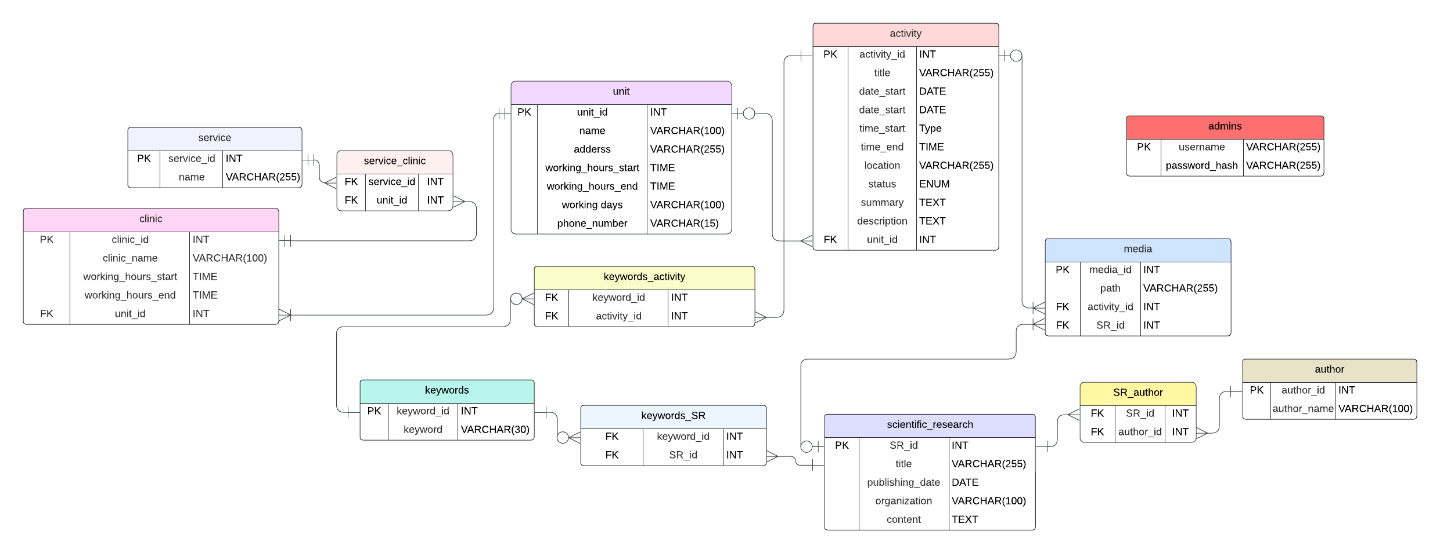
* **Purpose:**
* Provides a high-level, conceptual representation of the system's entities and relationships.
* Focuses on the logical structure of the database, showing how entities relate to one another.
* **Key Components:**
* Entities: Represented as rectangles (e.g., *Clinic*, *Service*, *Unit*).
* Relationships: Represented as diamonds (e.g., *contains*, *provides*, *organizes*).
* Cardinality: The lines connecting entities may have notation to indicate the cardinality of relationships (e.g., one-to-many, many-to-many).
* Representation:



* **Description:**
* **Admins:** represents system administrators, storing information for authentication, including username and password
* **Unit:** an entity that contains information about each unit
  + Each unit provides one or more service
  + Each unit contains one or more clinic
  + A unit may organize zero or more activity
  + Each unit have its contact information
* **Clinic:** an entity that contains information about clinics of unit
  + A clinic is a part of one unit
* **Service :** an entity that describes a service provided by a unit
  + A service is provided by one or more unit
* **Activity:** an entity that contains information about the activities of a unit or an activity by the general department
  + An activity may be organized by one unit or directly by the general department.
  + An activity may be accompanied by zero or more media elements
  + An activity must be characterized by one or more keywords
* **scientific\_research:** an entity that contains the data of each scientific research article
  + Each scientific research must be written by **one or more authors**.
  + Each scientific research may be accompanied by with zero or more media element
  + Each scientific research must be characterized by a certain keyword
* **media:** an entity that contains all the media elements displayed in activities or in the scientific research articles
  + A media element must be assigned to exactly one scientific research or one activity.
* **keyword**: an entity that contains all keywords that characterizes an activity or a scientific research to be used in search
  + A keyword can be associated with one or more activities and one or more scientific research articles
* **author:** an entity which contains information about authors who have written a scientific research
  + an author can write one or more scientific research

### Logical Data model

* **Purpose:**
* Represents the physical database structure with normalized tables.
* Focuses on the implementation details required for development.
* **Key Components:**
* Tables: Each entity from the ERD is converted into a table (e.g., *Unit*, *Clinic*, *Activity*).
* Columns: Define the attributes of each entity with data types (e.g., INT, VARCHAR(255), DATE).
* Primary Keys (PK): Unique identifiers for each row in a table.
* Foreign Keys (FK): Relationships between tables, where one table references the primary key of another table.
* **Representation:**



* **Description:**

1. **Admins**:
   * Contains credentials for admin users (e.g., username and password\_hash), used for authentication and authorization.
2. **Unit**:
   * Represents organizational divisions within the department. Each unit (unit\_id) includes attributes like name, address, working\_hours\_start, working\_hours\_end, and phone\_number.
   * A unit may organize one or more **Activities**, contain one or more **Clinics**, and offer one or more **Services** (via a many-to-many relationship through service\_unit).
3. **Clinic**:
   * Represents the clinics under each unit. A clinic (clinic\_id) has a name (clinic\_name) and is associated with one unit (unit\_id).
4. **Service**:
   * Represents services offered by the medical center. Services (service\_id) can be linked to one or more units through the service\_unit relationship table.
5. **Activity**:
   * Represents events or programs organized by a unit or the general department. Attributes include title, date, time, location, status, summary, and description.
   * Activities may involve **Media** (e.g., images or videos) and be associated with one or more **Keywords** through the keywords\_activity table.
6. **Scientific Research**:
   * Represents scientific research articles authored by individuals. Each research (SR\_id) includes attributes such as title, publishing\_date, and content.
   * Research can have multiple **Authors** and be linked to one or more **Keywords** via keywords\_SR.
7. **Author**:
   * Represents individuals who have authored scientific research. Each author (author\_id) is associated with one or more research articles (SR\_id).
8. **Media**:
   * Stores media elements such as images or videos linked to either an **Activity** or **Scientific Research** through foreign keys (activity\_id and SR\_id).
9. **Keywords**:
   * Represents tags used to categorize or search for **Activities** and **Scientific Research**. Keywords (keyword\_id) are linked to both entities through their respective relationship tables: keywords\_activity and keywords\_SR.

**Key Relationships:**

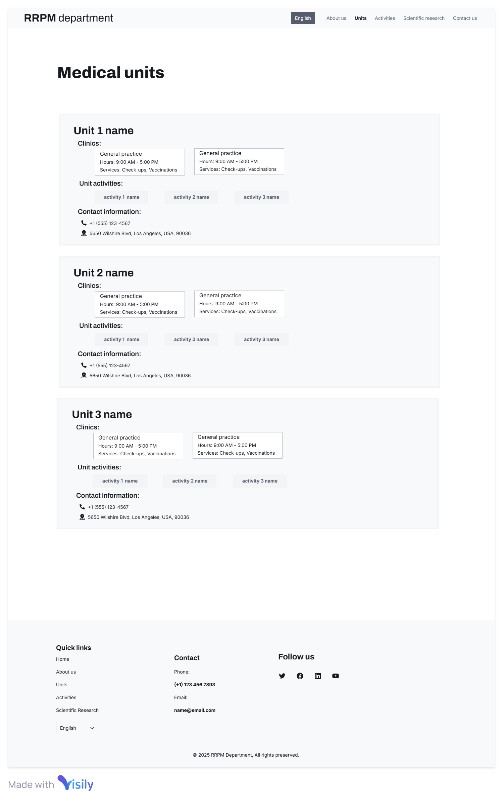
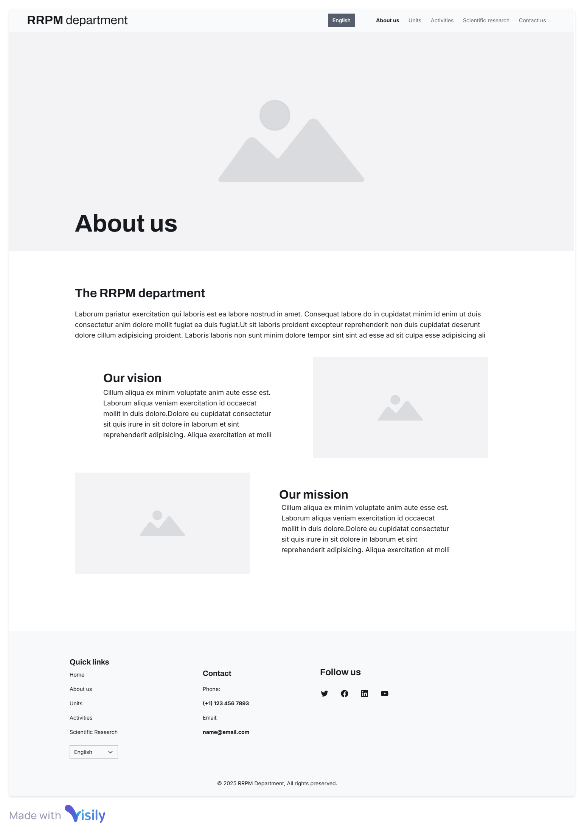
* **Unit and Activity**: A unit (unit\_id) organizes zero or more activities (activity\_id).
* **Unit and Clinic**: Each unit (unit\_id) contains one or more clinics (clinic\_id).
* **Unit and Service**: A unit (unit\_id) offers one or more services through a many-to-many relationship with the service\_unit table.
* **Activity and Media**: Each activity (activity\_id) can involve one or more media elements (media\_id).
* **Scientific Research and Author**: Each scientific research article (SR\_id) can have one or more authors (author\_id).
* **Keywords**: Keywords (keyword\_id) are linked to both activities (activity\_id) and scientific research articles (SR\_id) via separate many-to-many relationship tables.

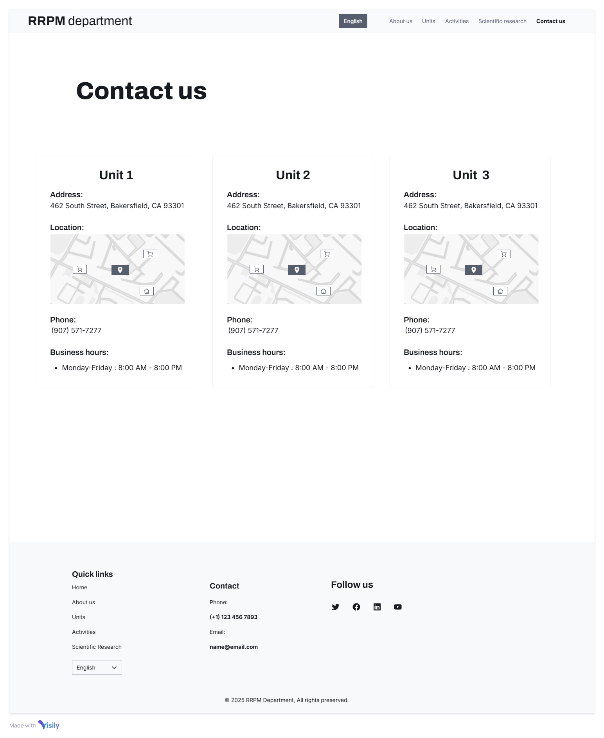
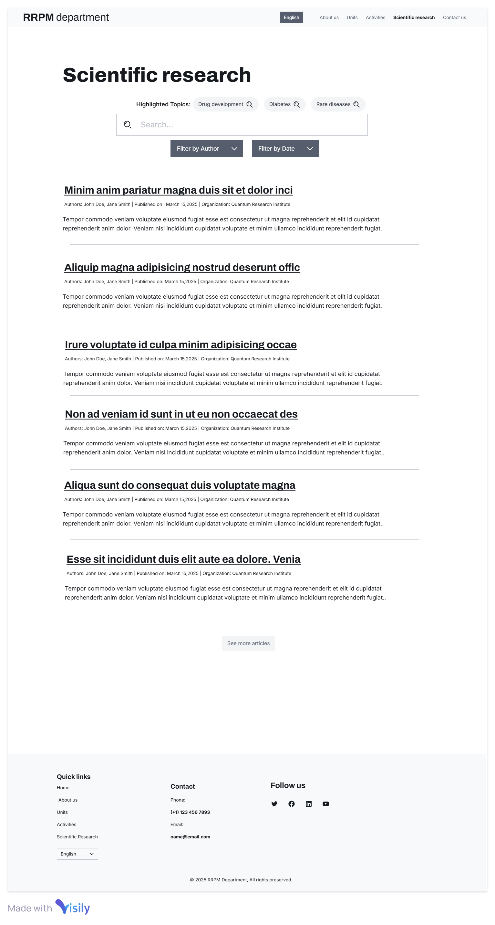
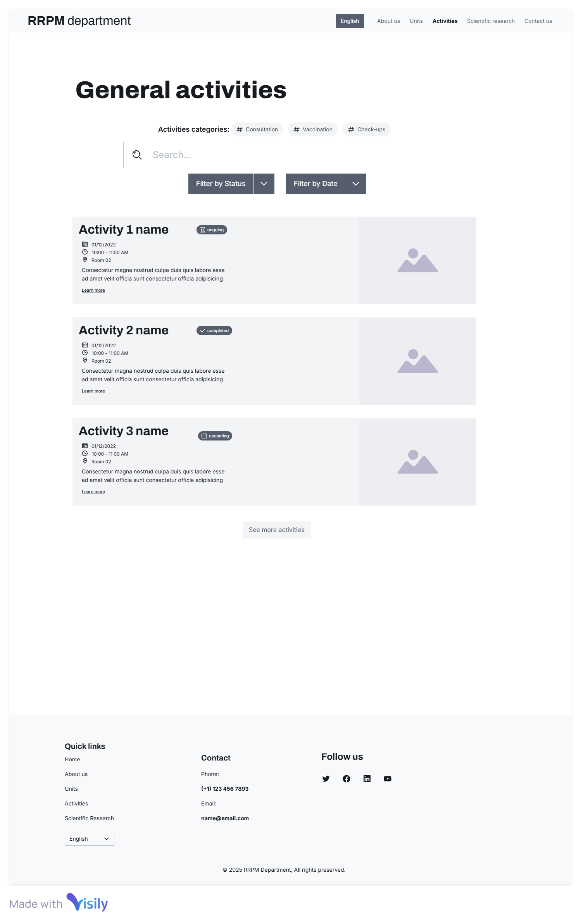
## Ui design

The UI Design section describes the visual and interactive aspects of the system, showcasing key interface elements to ensure a user-friendly experience.

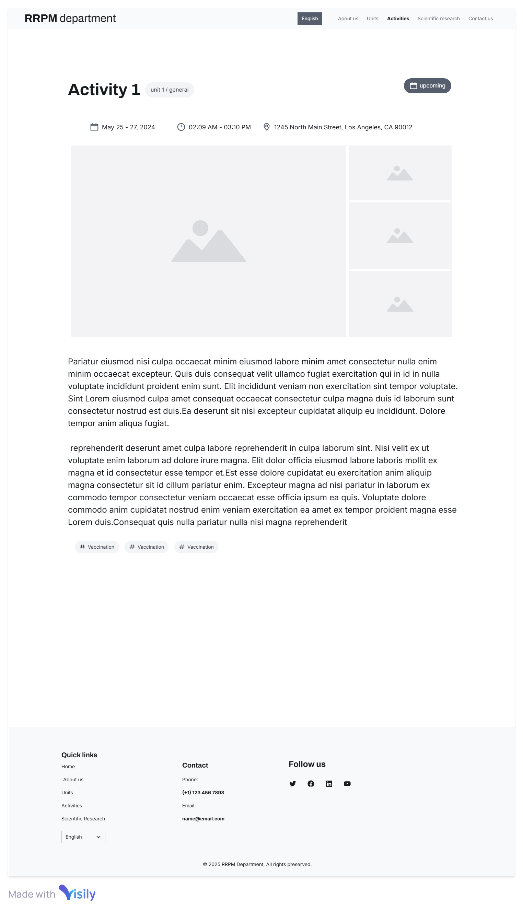
### Wireframes

#### 4.3.1.2. Main pages (Home- About us – Units – scientific research – contact us)

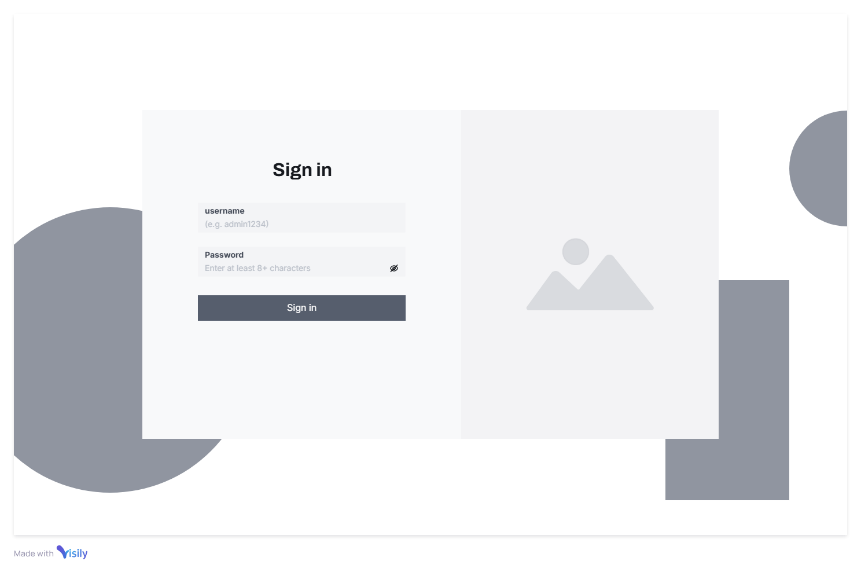
Home page


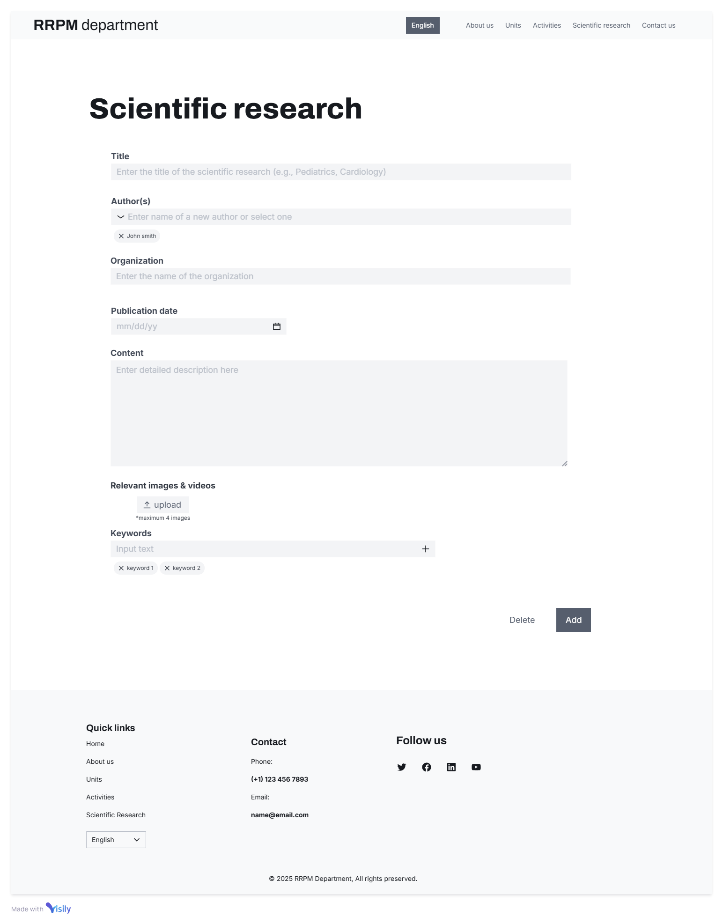
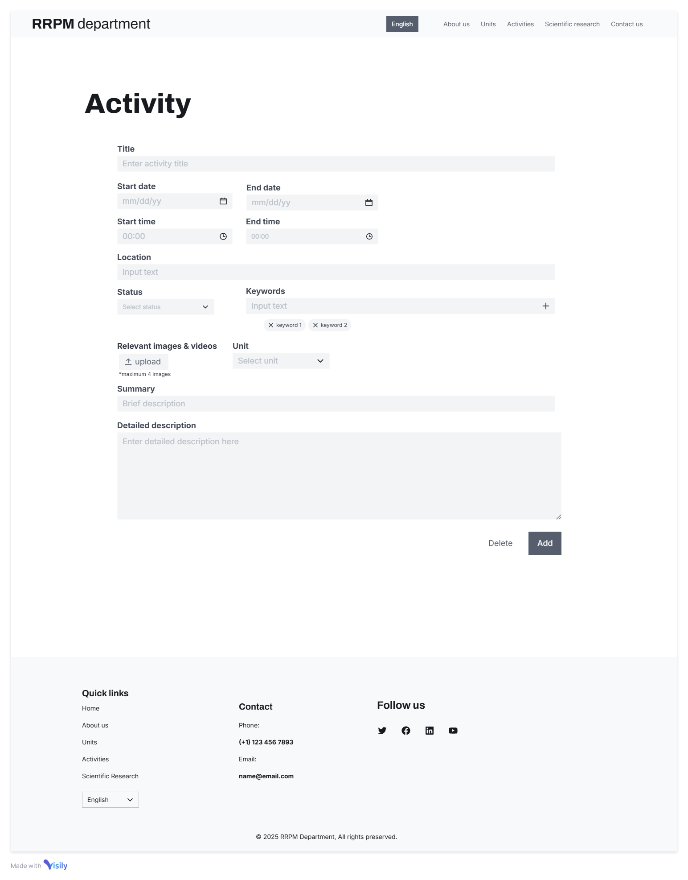
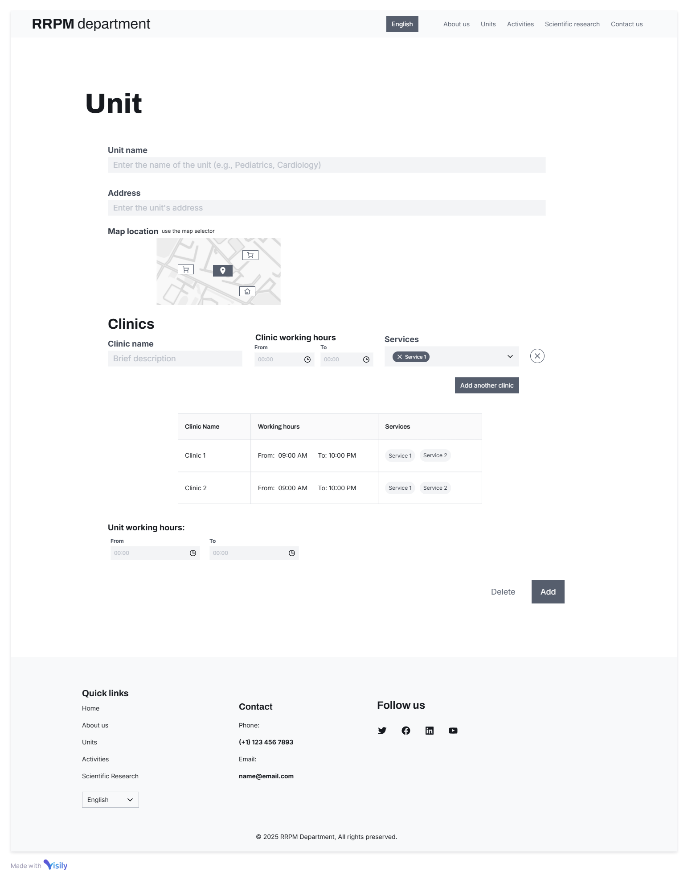


#### 4.3.1.2. Secondary pages (single activity page – single scientific research page)



4.3.1.3. Admin pages (login page – Unit management page – Activity management page – scientific research management page)





### User Interaction Details for general user

**Home Page**

The home page serves as the central entry point of the website and includes the following interactive elements:

**Navigation Bar**

* Fixed in all pages.
* **Links**: Provides access to key sections of the website: About Us, Units, Activities, Scientific Research, and Contact Us.
* **Language Selector**: Enables users to switch the website language (e.g., English, Arabic).
* **Interaction**: Clicking on any link redirects the user to the respective section or page.

**Featured Medical Units**

* Displays highlighted medical units.

**Latest Activities**

* **Activity Cards**: Showcases ongoing, upcoming, or completed activities along with the date and summary on each card.
* **Explore Button**: Each activity card includes an "Explore" button.
* **Interaction**: Clicking on the "Explore" button redirects users to a detailed single activity page.

**Read Latest Scientific Insights**

* **Scientific Research Articles**: Displays titles and publication dates of selected scientific articles.
* **Interaction**: Clicking on a title redirects users to the corresponding detailed scientific research page.
* **Explore More Button**: Navigates users to the scientific research section, where additional articles are available.
* **Pagination Arrows**: Allows users to browse through more articles.

**Footer**

* Fixed in all pages
* **Quick Links**: Includes links to major sections such as Home, About Us, Units, and more.
* **Contact Information**: Displays a phone number and email address.
* **Follow Us**: Provides social media icons that link to the department’s social media pages.
* **Language Selector**: Another language switcher available at the bottom of the page.

**About Us Page**

* Provides an overview of the department, its mission, and its vision.

**Units Page**

* **Units List**: Displays a list of medical units with summaries for each.
* **Details for Each Unit**:
  + List of associated clinics with their working hours and services.
  + List of unit activities .
  + Contact information for the unit (phone number and address).
* **Interaction**: Clicking on a unit’s activity redirects users to a single activity page with detailed information.

**Activities Page**

**Activities List**

* Displays a list of general activities with summaries.
* Each activity block includes:
  + **Information Displayed**:
    - Name of the activity.
    - Status badge (e.g., Ongoing, Completed, Upcoming).
    - Date and time.
    - Location (e.g., Room 02).
    - Summary.
    - "Learn More" link.
  + **Interaction**: Clicking "Learn More" redirects users to a detailed single activity page.

**Filters**

* **Activities Category Filter (Tags)**:
  + Clicking on a tag filters activities by the selected category.
* **Search Bar**:
  + Enables users to search for activities by keywords or titles.
  + Provides dynamic updates to the activity list while typing (live search).
* **Filter Buttons**:
  + **Filter by Status**: Dropdown menu with options like "Ongoing," "Completed," and "Upcoming."
  + **Filter by Date**: Dropdown menu to filter by year.
  + **Interaction**: Clicking applies the selected filter and updates the activity list.

**Pagination**

* **See More Activities**:
  + Clicking this button loads additional activity blocks through either infinite scrolling or a new page.
  + **Interaction**: Provides a smooth transition or loader animation while fetching new content.

**Scientific Research Page**

Displays a list of research articles with summaries and includes the following interactive elements:

**Highlighted Topics (Tag Filters)**

* Clicking on a topic dynamically filters the research articles below by the selected tag.
* Supports selecting multiple tags for combined filtering.

**Search Bar**

* **Action**: Allows users to search for specific research articles using keywords and titles.
* **Interaction**: Typing a query dynamically updates the list of articles below (live search).

**Filter Buttons (Author and Date)**

* **Filter by Author**:
  + Opens a dropdown to select an author.
* **Filter by Date**:
  + Opens a dropdown to select a certain year.
* **Interaction**: Clicking applies the selected filter, updating the articles list below.

**Scientific research List**

* Each article block includes:
  + **Information Displayed**:
    - Title of the research.
    - Authors.
    - Publication date.
    - Organization.
    - Part of the content.
  + **Interaction**: Clicking on the title redirects users to the Single Research Page for detailed content.

**Pagination**

* **See More Articles**:
  + Clicking this button loads additional research articles through either infinite scrolling or a new page.

**Contact Us Page**

* **Featured Medical Units**:
  + Summarizes contact information for each unit, including:
    - Address.
    - Embedded Google Map with a pin at the unit’s exact location.
    - Phone number.
    - Business hours.
  + **Interaction**: Users can zoom in/out or click the map for full Google Maps redirection.

**Single Activity Page**

Displayed when the user clicks on the title of a specific activity.

**Key Elements:**

* **Title of the Activity**.
* **Badge**: Indicates the unit name (if the activity belongs to a specific unit) or "General" (if the activity is not associated with any unit).
* **Date and Time**.
* **Location**.
* **Representative Images or Videos**.
* **Detailed Description**.
* **Tags of the Activity**.

**Single Scientific Research Page**

Displayed when the user clicks on the title of a specific scientific research article.

**Key Elements:**

* **Title**: Includes the title of the scientific research, author, and publishing date.
* **Representative Media**: Images and videos (maximum of 4 elements).
* **Full Content**: Displays the complete content of the research.
* **Tags**: Keywords associated with the research.
* **Related Scientific Research**:
  + Displays research articles with matching keywords.
  + If no matching articles are available, random articles are displayed.

### User interaction details for Admin user

Admin users access the admin panel via a specific link and login with a username and password. After logging in, they can view the same pages as general users, with additional functionalities to manage units, activities, and scientific research.

**Unit management page**

* **Unit name**: Text input for entering the unit’s name.
* **Address**: Text input for entering the unit’s address.
* **Map location**: Interactive map for setting the unit's location with a draggable pin.
* **Clinics section**:
  + **Clinic name** : Text input to enter the name of a clinic associated with the unit.
  + **Clinic working hours:** Time inputs for "From" and "To" to pecify the working hours of the clinic.
  + **Clinic services:** Tagging system to add services provided by the clinic using keywords or select from already existing services.
  + **Add Another Clinic Button:** a button to add a new clinic entry row to the clinics table.
* **Unit Working Hours**: Time inputs for "From" and "To" to set the general working hours for the unit.
* **Delete Button**: A button to delete the current unit.

This page is used for adding or editing units, with pre-filled fields displayed when editing.

**Activity management page**

* + **Title**: A text input box for entering the activity’s name.
  + **Start Date/End Date**: Date pickers with calendar icons to select the activity's start and end dates.
  + **Start Time/End Time**: Time input fields with clock icons to specify the start and end times.
  + **Location**: A text box for entering the activity’s location.
  + **Status**: Dropdown menu for selecting the activity’s status (Completed, Upcoming, Ongoing).
  + **Keywords**: Tagging system where users can type and add keywords for the activity or select already existing keywords.
  + **Relevant Media**: A file upload section allowing up to 4 media files.
  + **Unit**: A dropdown menu to select the related unit or “general” if the activity is not associated with any unit.
  + **Summary**: A single-line text box for a brief description of the activity.
  + **Detailed Description**: A multi-line text area for providing a more detailed explanation.
  + **Delete Button**: A button to delete the current activity.
  + **Add Button**: A button to save the activity.

This page is used for adding or editing activities, with pre-filled fields displayed when editing.

**Scientific research management page**

* **Title**: A text input box for entering the title of the scientific research.
* **Author(s)**: A dropdown menu with the ability to select an existing author or add a new author’s name.
* **Organization**: A text input box for entering the name of the organization affiliated with the research.
* **Publication Date**: A date picker with a calendar icon to select the publication date of the research.
* **Content**: A multi-line text area to provide a detailed description of the research content.
* **Relevant Media**: A file upload section allowing the addition of up to 4 images or videos related to the research.
* **Keywords**: A tagging system where users can type and add relevant keywords to categorize the research or select from the already existing keywords.
* **Delete Button**: A button to delete current scientific research.
* **Add Button**: A button to save or create the scientific research.

This page is used for adding or editing activities, with pre-filled fields displayed when editing.

# Deployment

## Hosting

The hosting environment for the system will be determined during the final stages of development. The following considerations will guide the hosting decision:

* **Scalability**: The hosting solution should handle future increases in user traffic and data volume.
* **Reliability**: High uptime and availability will be critical to ensure continuous access to the system.
* **Security**: Hosting must include measures to protect sensitive data and prevent unauthorized access.
* **Cost-Effectiveness**: The chosen solution will balance performance with budget constraints.

Potential hosting options being evaluated include:

* Cloud-Based Hosting: Platforms like AWS, Google Cloud, or Azure for flexibility and scalability.

#### **5.1.1. Deployment Environment**

* **Development Environment**: Used for coding and testing during the development phase.
* **Staging Environment**: A pre-production environment for final testing before release.
* **Production Environment**: The live environment where the system will be accessible to users.

#### **5.1.2. Deployment Strategy**

* **Incremental Deployment**: The system may be deployed module by module to minimize risks.
* **Rollback Mechanisms**: Deployment will include fallback mechanisms to revert to the previous version in case of failure.
* **Monitoring Tools**: Performance and error monitoring tools will be integrated to ensure smooth deployment.