

Group 1: 225053661 - Nguyễn Đức Xuân Trường

225051374 - Nguyễn Khánh Vinh

225054038 - Đào Quốc Anh

225052605 - Đặng Trương Phương Nghi

Class: **242.ITE1219E.B02E**

Lecturter: Mr. Hoàng Văn Hiếu

**FINAL PROJECT**

**Course: 2025-2026**

**Course code: ITE1219E.B02E**

**Online Volunteer Coor**

MINISTRY OF EDUCATION & TRAINING

HO CHI MINH CITY UNIVERSITY OF ECONOMICS AND FINANCE

Group 1: 225053661 - Nguyễn Đức Xuân Trường

225051374 - Nguyễn Khánh Vinh

225054038 - Đào Quốc Anh

225052605 - Đặng Trương Phương Nghi

Class: **242.ITE1219E.B02E**

Lecturter: Mr. Hoàng Văn Hiếu

**FINAL PROJECT**

**Course: 2025-2026**

**Course code: ITE1219E.B02E**

**Online Volunteer Coor**

MINISTRY OF EDUCATION & TRAINING

HO CHI MINH CITY UNIVERSITY OF ECONOMICS AND FINANCE

CONTENTS

Contents

[TABLE OF CONTENTS 5](#_Toc202195050)

[TABLE OF PICTURES 6](#_Toc202195051)

[**INTRODUCTION** 8](#_Toc202195052)

[**1. General introduction** 8](#_Toc202195053)

[**2. Reason for choosing the topic** 8](#_Toc202195054)

[**3. Project objectives and contents** 9](#_Toc202195055)

[**4. Thesis structure** 9](#_Toc202195056)

[**CHAPTER 1: OVERVIEW** 10](#_Toc202195057)

[**1. Practical requirements and reasons for choosing the topic** 10](#_Toc202195058)

[**1.1. Actual requirements** 10](#_Toc202195059)

[**1.2. Reason for choosing the topic** 11](#_Toc202195060)

[**2.** **Purpose, objects and scope of research** 11](#_Toc202195061)

[**2.1. Purpose of the topic** 11](#_Toc202195062)

[**2.2. Research object** 12](#_Toc202195063)

[**2.3. Scope of research** 12](#_Toc202195064)

[**3. System analysis** 13](#_Toc202195065)

[**3.1. Functional Requirements** 13](#_Toc202195066)

[**3.2. Non-Functional Requirements** 15](#_Toc202195067)

[**3.3. Other factors to consider** 16](#_Toc202195068)

[**CHAPTER 2: THEORETICAL BASIS** 17](#_Toc202195069)

[**1. Overview of Volunteers and Volunteer Websites** 17](#_Toc202195070)

[**1.1. Overview of Volunteers** 17](#_Toc202195071)

[**1.2. Overview of Volunteer Websites** 18](#_Toc202195072)

[**2. Technology used to build the website** 19](#_Toc202195073)

[**CHAPTER 3: ANALYSIS AND DESIGN** 22](#_Toc202195074)

[**1. Analysis of the current state of volunteer coordination websites** 22](#_Toc202195075)

[**2. System design** 22](#_Toc202195076)

[**2.1. Use-case overview of the system** 22](#_Toc202195077)

[**2.2. Sequence Diagram** 34](#_Toc202195078)

[**2.3. Activity Diagram** 38](#_Toc202195079)

[**2.4. DataBase Diagram** 41](#_Toc202195080)

[**2.5. Website Sketch** 47](#_Toc202195081)

[**CHAPTER 4: BUILD SYSTEM** 53](#_Toc202195082)

[**1. Website home page** 53](#_Toc202195083)

[**2. Login of Admin** 53](#_Toc202195084)

[**3. Admin dashboard** 54](#_Toc202195085)

[**4. Administrator activities management function** 54](#_Toc202195086)

[**5. Administrator users management function** 55](#_Toc202195087)

[**6. Administrator notifications management function** 55](#_Toc202195088)

[**7. Administrator ratings management function** 56](#_Toc202195089)

[**8. Administrator registrations management function** 56](#_Toc202195090)

[**9. Authenticated user** 57](#_Toc202195091)

[**10.The data is fetched from the database** 58](#_Toc202195092)

[**CONCLUDE** 59](#_Toc202195093)

[**1. Functions and Description** 59](#_Toc202195094)

[**2. Limit** 60](#_Toc202195095)

[**2.1. For Users (Volunteers)** 60](#_Toc202195096)

[**2.2. For administrators (Admin/Coordinator)** 60](#_Toc202195097)

[**3. Development direction** 61](#_Toc202195098)

[**REFERENCE** 62](#_Toc202195099)

# TABLE OF CONTENTS

[Table 3.1. Specification Use-case “Register Activities” 24](#_Toc202195100)

[Table 3.2. Specification Use-case “Check-In Activities” 26](#_Toc202195101)

[Table 3.3. Specification Use-case “Search Activities” 27](#_Toc202195102)

[Table 3.4. Specification Use-case “History Activities” 30](#_Toc202195103)

[Table 3.5. Specification Use-case “Management Activities” 31](#_Toc202195104)

[Table 3.6. Specification Use-case “Volunteer Management” 34](#_Toc202195105)

[Table 3.7. Table Users 42](#_Toc202195106)

[Table 3.8. Table Roles 42](#_Toc202195107)

[Table 3.9. Table User\_Roles 42](#_Toc202195108)

[Table 3.10. Table Activities 43](#_Toc202195109)

[Table 3.11. Table Registrations 43](#_Toc202195110)

[Table 3.12. Table Ratings 44](#_Toc202195111)

[Table 3.13. Table Notifications 44](#_Toc202195112)

[Table 3.14. Table Volunteers 45](#_Toc202195113)

[Table 3.15. Table News 45](#_Toc202195114)

[Table 3.16. Table Causes 46](#_Toc202195115)

[Table 3.17. Table Reasons 46](#_Toc202195116)

# TABLE OF PICTURES

[Picture 3.1. Main diagram of Use-case model 22](#_Toc202195117)

[Picture 3.2. Sequence Register Activities 34](#_Toc202195118)

[Picture 3.3. Sequence Check-in 35](#_Toc202195119)

[Picture 3.4. Sequence Management Activities 36](#_Toc202195120)

[Picture 3.5. Sequence Sreach Activities 37](#_Toc202195121)

[Picture 3.6. Activities Register 38](#_Toc202195122)

[Picture 3.7. Activitiy check-in 38](#_Toc202195123)

[Picture 3.8. Activities Sreach 39](#_Toc202195124)

[Picture 3.9. Activities management 40](#_Toc202195125)

[Picture 3.10. Database Diagram 41](#_Toc202195126)

[Picture 3.11. Website home page 47](#_Toc202195127)

[Picture 3.12. Website home page 47](#_Toc202195128)

[Picture 3.13. Website home page 48](#_Toc202195129)

[Picture 3.14. Website home page 48](#_Toc202195130)

[Picture 3.15. Website home page 49](#_Toc202195131)

[Picture 3.14. Log in of Admin 49](#_Toc202195132)

[Picture 3.15. Admin Dashboard 50](#_Toc202195133)

[Picture 3.16. Admin Dashboard 50](#_Toc202195134)

[Picture 3.17. Administrator users management function 51](#_Toc202195135)

[Picture 3.18. Administrator activities management function 51](#_Toc202195136)

[Picture 3.19. Administrator registrations management function 52](#_Toc202195137)

[Picture 4.1. Website home page 53](#_Toc202195138)

[Picture 4.2. Log in of Admin 53](#_Toc202195139)

[Picture 4.3. Admin Dashboard 54](#_Toc202195140)

[Picture 4.4. Administrator activities management function 54](#_Toc202195141)

[Picture 4.5. Administrator users management function 55](#_Toc202195142)

[Picture 4.6. Administrator notifications management function 55](#_Toc202195143)

[Picture 4.7. Administrator ratings management function 56](#_Toc202195144)

[Picture 4.8. Administrator registrations management function 56](#_Toc202195145)

[Picture 4.9. Authenticated user 57](#_Toc202195146)

[Picture 4.10. The data is fetched from the database. 58](#_Toc202195147)

# **INTRODUCTION**

## **1. General introduction**

In the context of an increasingly developing society, volunteer activities have become an indispensable part, contributing to solving many community problems and spreading good values. From environmental protection campaigns, supporting the disadvantaged, to educational and medical events, the role of volunteers is always a key factor in the success of every project. However, traditional volunteer coordination and management often faces many challenges such as manual registration processes, scattered activity information, difficulties in attendance, monitoring and evaluating effectiveness. These limitations not only waste time and resources but also affect the experience of both volunteers and coordinating organizations.

Recognizing the importance of optimizing volunteer work in the digital age, this report presents the "Online Volunteer Coordination System". This is a web application designed to digitize and automate the entire volunteer management process, from registration, coordination, attendance to information management and reporting. The system will act as an effective bridge, helping volunteers easily search and participate in suitable activities, while supporting coordinators to manage their work in a more professional and transparent manner.

Through analyzing the current situation, proposing core and advanced features, along with choosing the right technology (Java Spring MVC, MySQL, HTML/CSS/JavaScript), this topic is expected to bring a practical solution, not only improving the effectiveness of volunteer work but also contributing to building a more dynamic, cohesive and humane community.

## **2. Reason for choosing the topic**

Based on the actual needs and limitations of current methods, the development of an Online Volunteer Coor is necessary and urgent. This topic not only solves current problems but also brings many practical benefits such as improving management and organization efficiency, enhancing connection and interaction, transparency and fairness in recognizing contributions, applying information technology to social activities and finally having opportunities for comprehensive learning and skill development.

Therefore, developing an "Online Volunteer Coor" is not only an effective solution to current problems but also the right and necessary direction in the digital age, contributing to promoting community activities and sustainable development.

## **3. Project objectives and contents**

The main purpose of the "Online Volunteer Coor" project is to develop a complete and efficient web application to automate and optimize the process of managing and coordinating volunteer activities. The system aims at the following goals: connecting volunteers and activities, supporting coordinators, improving professionalism and promoting community activities.

The "Online Volunteer Coor" project focuses on building a complete web application to connect, manage and coordinate volunteer activities in an efficient and professional manner. Developed in Java programming language on Windows platform, this project provides a user-friendly and convenient interface and comprehensive management functions for both volunteers and coordinators, as well as volunteer organizations.

## **4. Thesis structure**

The project report consists of four chapters:

**Chapter 1. Overview:** presents an overview of the actual requirements, context, reasons for choosing the project, and provides information about the research.

**Chapter 2. Theoretical basis:** presents concepts related to the system, development models, and applied technologies

**Chapter 3. Analysis and design:** conducts analysis and design of requirements, system functions, database design, and necessary data processing flows.

**Chapter 4. System development and experimentation:** builds the system, solves problems, and puts them into practice, evaluates performance, and tests the quality of the application.

# **CHAPTER 1: OVERVIEW**

## **1. Practical requirements and reasons for choosing the topic**

In the context of modern society, volunteer activities play an increasingly important role in building communities, supporting disadvantaged people and raising civic awareness. However, the organization and management of traditional volunteer activities are facing many challenges and limitations. The number of volunteer organizations and groups is increasing, leading to the need for a more effective management system.

### **1.1. Actual requirements**

Traditional volunteer coordination methods such as manual spreadsheets, disjointed emails, and unprofessional social media groups are outdated and ineffective in today’s digital age. They cause many problems such as:

* **Difficulty in finding and connecting with volunteers:** Promoting volunteer activities to the right audience is limited, often relying on unfocused communication channels, reducing the ability to attract enough volunteers.
* **Challenges in managing registration and attendance:** Manual registration is time-consuming, error-prone and confusing. Field attendance is also inconvenient, making it difficult to accurately track volunteer participation and commitment.
* **Limited information management of activities:** Information about volunteer events, such as specific times, locations, and job descriptions, is often not centralized and easily accessible. This makes it difficult for both coordinators to organize and volunteers to choose appropriate activities.
* **Lack of effective communication and feedback support tools:** Information exchange between coordinators and volunteers, as well as post-activity feedback collection, is not yet automated and immediate, affecting coordination efficiency.
* **Difficulty in evaluating and recognizing contributions:** Lack of a transparent system to track and evaluate the effectiveness of each activity, as well as to recognize volunteers' efforts and time, can reduce motivation and long-term commitment.

### **1.2. Reason for choosing the topic**

Based on the practical needs and limitations of current methods, the development of an Online Volunteer Coor is necessary and urgent. This topic not only solves the existing problems but also brings many practical benefits:

* **Improve management and organization efficiency:** The application will provide a centralized platform for coordinators to easily create, manage activities, track registrations and attendance. This helps optimize processes, saving time and resources.
* **Enhanced connectivity and interaction:** The system will help volunteers easily search and register for activities that suit their interests and time. Features such as sending email notifications and displaying locations on maps will significantly improve the ability to interact and communicate between parties.
* **Transparency and fairness in recognition of contributions:** The ability to take attendance online and evaluate activities will help accurately and transparently recognize volunteers' efforts, encourage them to actively participate and maintain the spirit of volunteerism.
* **Applying information technology to social activities:** Applying modern technologies such as Spring MVC, MySQL, Google Maps API, JavaMail, and QR code to a meaningful social problem is not only an interesting technical problem but also demonstrates the responsibility of technology to the community.
* **Comprehensive learning and skill development opportunities:** This subject provides a practical environment for students to consolidate their knowledge of Java web programming (MVC, HTTP, database), develop skills in interface design, backend programming, API integration, project management (Maven), and teamwork (Git). These are core skills that are highly demanded in the labor market..

Therefore, developing "Online Volunteer Coor" is not only an effective solution to the current problem but also a correct and necessary direction in the digital age, contributing to promoting community activities and sustainable development.

## **2.** **Purpose, objects and scope of research**

### **2.1. Purpose of the topic**

The main purpose of the "Online Volunteer Coor" project is to develop a complete and effective web application to automate and optimize the process of managing and coordinating volunteer activities. Specifically, the system aims at the following goals:

* **Connecting volunteers and activities:** Creating an easy platform for volunteers to search, register and check in for activities that suit their interests, skills and schedules.
* **Supporting coordinators:** Providing powerful tools for coordinators to create, manage activities, track registration status, check in and evaluate effectiveness in a transparent and effective manner.
* **Improving professionalism:** Converting manual processes to automation, helping volunteer organizations and groups operate professionally, saving time and resources.
* **Promote community activities:** Contribute to encouraging volunteerism, increasing community participation in meaningful activities through a friendly and accessible interface.

### **2.2. Research object**

The research object of the topic is the business process and functional requirements of a volunteer management and coordination system. Specifically, including:

* **Volunteers:** People who need to search, register, participate and check in for volunteer activities.
* **Coordinators:** People responsible for creating, managing, approving activities, verifying attendance and managing volunteer information.
* **Volunteer activities:** Including creating new, updating, displaying detailed information about activities (name, description, time, location, requirements, number of volunteers needed).
* **Related data:** User information, activity information, registration data, attendance data, reviews, feedback, etc.

### **2.3. Scope of research**

**Geographic scope:** The application will be developed to serve organizations, volunteer groups and individual volunteers within the country of Vietnam. However, in the long term, the system can be expanded to support international activities and volunteers.

**Scope of content:**

+ **Research user needs and behaviors:** Including understanding how current organizations manage volunteer activities, the difficulties they encounter, and volunteers' expectations of a registration and activity tracking platform.

+ **Develop application features:** including basic features such as searching, viewing volunteer activity details, registering to participate

**Time scope:** The project will be carried out within a specific period of time, including the initial research stages, system design, application development, testing, and implementation and evaluation. The expected time to complete this entire process will be specified in detail in the project plan.

## **3. System analysis**

### **3.1. Functional Requirements**

Functional requirements describe the tasks the system must perform to meet user needs. For a volunteer coordination system, the main functions include:

**3.1.1. User management:**

* **Register/Login:**
  + Volunteer: Register an account (via email, password, full name and phone number ), update personal information (name, age, gender, address, ...), skills, experience.
  + Organization/Coordinator: Log in to your account, verify your organization information, update your profile (organization name, field of activity, description, contact information).
  + System Administrator: Manage user accounts, authorization.
* **Profile Management:** Allows users to edit personal/organizational information, change password.

**3.1.2. Project/Campaign Management:**

* **Create/Edit Project/Campaign:** Organizations can post project/campaign details (name, description, goals, duration, location, volunteer requirements, number needed, contact).
* **View project/campaign list:** Users (volunteers and organizations) can view active or completed projects/campaigns.
* **Search/Filter projects/campaigns:** Allows volunteers to search for projects by keyword, field, location, time, skill requirements.

**3.1.3. Volunteer and application management:**

* **Apply/Register to Participate:** Volunteers can apply to participate in suitable projects/campaigns.
* **Approve/Reject Applications:** The organization can review applications, accept or reject volunteers.
* **Volunteer List:** Organizations can view a list of approved volunteers for their project.
* **Cancellation/Withdrawal:** Volunteers may cancel their registration if they are unable to participate.

**3.1. 4. Coordination and assignment:**

* **Assigning Tasks/Shifts:** The organization can assign specific tasks or schedule shifts for each volunteer.
* **Internal communication:** The system can integrate messaging and notification tools (email, SMS, push notification) between the organization and volunteers, or between volunteers within the same project.
* **Reminders/Notifications:** Automatically send reminders about work schedules, upcoming tasks, schedule changes.

**3.1.5. Monitoring and evaluation:**

* **Recording hours:** Volunteers can record their own hours or have the organization confirm the volunteer's hours.
* **Performance Evaluation:** The organization can evaluate the volunteer's performance after completing the task/project.
* **Volunteer Feedback:** Volunteers can provide feedback about the project, organization, or their experience.
* **Statistics/Reports:** The system provides reports on the number of volunteers, hours worked, number of completed projects, etc.

**3.1.6. Additional functions (optional):**

* **Certification:** Automatically generate and issue electronic certificates to volunteers after completing a certain number of hours or projects.
* **Map Integration:** Display project locations on a map, making it easy for volunteers to locate.
* **Multilingual support:** If targeting an international audience.

### **3.2. Non-Functional Requirements**

Non-functional requirements describe the qualities and properties of the system. They are equally important as functional requirements because they directly affect the user experience and stability of the system.

**3.2.1. Performance:**

* **Response speed:** Website/application must load quickly (under 3 seconds), search, registration, browsing operations must have low latency.
* **Load Capacity:** The system must be able to handle a large number of users concurrently (e.g. thousands of volunteers logging in at the same time during large campaigns).
* **Scalability:** The system must be easily scalable to handle future increases in users, projects, and data without compromising performance.

**3.2.2. Usability:**

* **Friendly interface:** Intuitive interface design, easy to understand, easy to use for all users (from young to old).
* **Consistency:** Interface components and operating procedures must be consistent across the system.
* **Accessibility:** Make sure your website/app is accessible across multiple devices (desktops, phones, tablets) and browsers.
* **Multi-platform support:** Web version and mobile app (iOS/Android) available for added convenience.

**3.2.3. Reliability:**

* **Uptime:** The system must be available at all times, with minimal downtime (e.g. 99.9% uptime).
* **Error recovery capability:** The system has a mechanism to back up data and recover quickly when an incident occurs.
* **Data Accuracy:** Ensure information about volunteers, projects, and hours is stored and displayed accurately.

**3.2.4. Integrity:**

* **API:** Provide APIs for easy integration with other systems (e.g. customer management systems, billing systems, analytics tools).
* **Social Network Integration:** Allows login/sharing of information via social networks.

### **3.3. Other factors to consider**

* **Budget and Time:** Available resources will influence the scope and complexity of the system.
* **Technology:** Choose programming languages, frameworks, and databases that suit the requirements and capabilities of the development team.
* **Audience:** Understand your volunteer and organization audience to design the most appropriate features and interface.

By carefully analyzing these functional and non-functional requirements, the development team was able to build a robust, efficient online volunteer coordination system that meets the needs of both volunteers and organizations.

# **CHAPTER 2: THEORETICAL BASIS**

## **1. Overview of Volunteers and Volunteer Websites**

### **1.1. Overview of Volunteers**

**1.1.1. Concept:**

Volunteers are individuals who voluntarily devote their time, effort, skills and sometimes finances to contribute to activities that benefit the community, society or environment without receiving remuneration. Volunteering activities stem from the spirit of mutual love, the desire to share and contribute to create positive values.

**1.1.2. Roles and benefits:**

Volunteers play an extremely important role in many fields:

* For individual volunteers:
  + Personal development: Learn new skills (leadership, teamwork, communication, problem solving), expand knowledge, experience different environments.
  + Improve mental health: Reduce stress, increase happiness, have a purpose in life, build confidence and self-esteem.
  + Expand relationships: Meet like-minded people, connect with the community, build social and professional networks.
  + Advance your career: Volunteer experience can enhance your resume, demonstrate soft skills and a sense of responsibility, and increase your chances of finding a job.
* For the community and society:
  + Solve social problems: Support the disadvantaged and vulnerable (orphans, lonely elderly, people with disabilities), improve education, healthcare, environment, and social security.
  + Build a strong community: Connect members, create a spirit of solidarity, mutual love, and promote a sense of community responsibility.
  + Contribute to sustainable development: Promote socio-economic development through community projects, environmental protection, and climate change response.

**1.1.3. Popular types of volunteer activities:**

Volunteer activities are very diverse, including:

* **Society - Community:** Donate blood, give gifts to the poor, help the homeless, visit the sick/lonely elderly, teach children in remote areas.
* **Environment:** Clean up waste, plant trees, protect wildlife, promote environmental protection.
* **Education - Health:** Support vaccination campaigns, health consultation, study guidance, organize free classes.
* **Culture - Arts:** Support the organization of cultural events, preserve heritage.
* **Emergency support:** Disaster relief, epidemic response.
* **Virtual Volunteering:** Volunteer activities carried out via the internet, such as translation, design, technical support, online consultation.

### **1.2. Overview of Volunteer Websites**

**1.2.1. Role of the volunteer website**

With the development of information technology, volunteer websites have become a powerful tool, helping to:

* Effective connection: A place to centralize information about volunteer activities from many organizations, helping volunteers easily search and register.
* Transparent management: Support organizations to manage information about activities, volunteers, monitor progress and effectiveness.
* Increased access: Expanding access to a large number of people who need and are able to participate in volunteering.
* Saving time and costs: Minimizing manual processes and paperwork, helping both coordinators and volunteers save time and effort.

**1.2.2. Popular volunteer websites:**

* **In Vietnam**
  + **iVolunteer Vietnam:** One of the largest and most reputable communities, regularly posting volunteer opportunities at home and abroad, especially for students and young people.
  + **Community groups/Fanpages on social networks (Facebook):** "Domestic and international experience opportunities", "Volunteer Information",... are popular channels for searching for information.
  + **Websites of organizations and associations:** Websites of Youth Unions at all levels, universities, and non-governmental organizations (NGOs) also often post information about their own volunteer activities.
* **International:**
  + **UN Volunteers (United Nations Volunteers):** The United Nations' global platform that provides volunteer opportunities for development and peace programs around the world, including online volunteering.
  + **VolunteerMatch**: One of the largest platforms in the US, connecting volunteers with nonprofits.
  + **Idealist.org:** Provides information on jobs, volunteer opportunities, and internships in the global nonprofit sector.
  + **WWOOF (World Wide Opportunities on Organic Farms):** Specializes in volunteering on organic farms around the world, in exchange for lodging and food.
  + **Points of Light:** A global network that inspires and equips people to make a difference through volunteering.

**1.2.3. Development trends**

Volunteer websites are increasingly integrating more technology (AI to suggest suitable activities, blockchain to make the donation process transparent and record contributions), focusing on user experience, and promoting online volunteering to expand reach and flexibility. Mobile optimization is also an important trend to increase accessibility.

## **2. Technology used to build the website**

To build a powerful and efficient online volunteer coordination system, this project will use a combination of the following technologies:

**Backend**

* **Java Spring MVC or Java Struts MVC:** These are popular Java frameworks, which provide MVC (Model-View-Controller) structure for developing web applications in an organized and efficient manner. Choosing one of the two depends on the familiarity and specific requirements of the development team.

**Frontend**

* **HTML:** Standard markup language for creating the structure and content of web pages.
* **CSS:** Used to style and design user interfaces, ensuring websites look attractive and professional.
* **JavaScript:** Client-side programming language used to create interactive effects, process dynamic data, and enhance user experience.
* **Bootstrap or JSP (JavaServer Pages):**
  + **Bootstrap:** A powerful CSS framework that helps develop responsive, cross-device compatible interfaces (desktops, tablets, mobile phones) quickly..
  + **JSP:** The technology allows embedding Java code into HTML pages, making it easier to create dynamic web pages, especially useful when working with Spring MVC or Struts MVC.

**Database**

* **MySQL hoặc MS SQL Server:** Two leading relational database management systems (RDBMS). Both are capable of storing and managing large amounts of data efficiently, ensuring information integrity and security.

**Project Management**

* **Maven:** A tool for managing and automating Java project builds. Maven makes it easy to manage dependencies, compile source code, and package applications.

**Web Server**

* **Apache Tomcat:** An open source application server, widely used for deploying Java web applications.

**Version Control**

* **Git (qua GitHub/GitLab):** A distributed version control system that allows multiple developers to work on a project without worrying about code conflicts. GitHub or GitLab will be used as a platform to store source code and manage team workflows.

**Additional Libraries and APIs**

* **JavaMail:** Java library that provides APIs for sending and receiving emails, which will be used to send registration confirmations, attendance or activity updates to volunteers.
* **Google Maps API:** Google Maps API, which allows integrating maps into websites to display the locations of volunteer activities.
* **ZXing or similar:** An open source library for generating and scanning barcodes/QR codes, used to generate QR verification codes for attendance and allow coordinators to scan the code for verification.

Combining these technologies will ensure the system is scalable, secure and provides a good user experience.

# **CHAPTER 3: ANALYSIS AND DESIGN**

## **1. Analysis of the current state of volunteer coordination websites**

Currently, despite numerous websites and technological solutions being applied in volunteer coordination, certain limitations persist in meeting the actual needs of both volunteers and coordinating organizations.

**On the part of volunteer:** The registration and attendance processes are sometimes still manual or not fully automated, causing inconvenience and wasting time. Additionally, there's a lack of advanced features such as activity location maps, automated notifications (via email/app), or the ability to effectively track participation history and evaluate activities.

**On the part of the organizations/coordinators:** Managing and coordinating a large number of volunteers for various activities remains a challenge, especially when tracking registration status, attendance, and task assignments.

## **2. System design**

### **2.1. Use-case overview of the system**

A group of white ovals on a black background

AI-generated content may be incorrect.

Picture 3.1. Main diagram of Use-case model

**2.1.1. Use-case 01 Specification**

|  |  |
| --- | --- |
| **Use Case Code** | UC01 |
| **Use Case Name** | Register Activities |
| **Describe** | Sign up for activities |
| **User (primary actor)** | Volunteer |
| **Điều kiện kích hoạt** | Volunteer clicks “Register Activities” |
| **Preconditions** | - The Volunteer has registered an account and logged into the system. |
| **Postcondition** | - The Volunteer's registration is successfully recorded in the database, with a unique confirmation code generated.  - A QR code containing the confirmation code is generated and made available to the Volunteer.  - The Volunteer receives a confirmation email with activity details (title, date/time, location) and the confirmation code.  - The activity's registered Volunteer count is updated in the database.  - The registration history is logged for reporting and analysis purposes.  - The Coordinator can view the updated list of registered Volunteers for the activity. |
| **Main event stream** | 1. The Volunteer accesses the "Register for Activity" function on the interface.  2. The system displays a list of upcoming activities, including details such as title, date/time, location, quantity  3. The Volunteer selects an activity to register for.  4. The system generates a unique confirmation code for the registration.  5. The system saves the registration details (Volunteer ID, Activity ID, confirmation code, registration date) in the database.  6. The system generates a QR code containing the confirmation code and makes it available for the Volunteer to view or download.  7. The system sends a confirmation email to the Volunteer via EmailService, including the activity details (title, date/time, location), confirmation code, and a link to the QR code.  8. The system displays a message to the Volunteer: "Registration successful. A confirmation email has been sent to your email address."  9. The activity's registered Volunteer count is incremented in the database.  10. The registration process is complete. |
| **Sub-event stream** | 1. If the activity is full → system displays “Activity is full” message  2. If registration fails (e.g., server error) → system shows an error and allows retry |

Table 3.1. Specification Use-case “Register Activities”

**2.1.2. Use-case 02 Specification**

|  |  |
| --- | --- |
| **Use Case Code** | UC02 |
| **Use Case Name** | Check-In Activities |
| **Describe** | - This use case describes the process by which a Volunteer checks in to a volunteer activity they have registered for. |
| **User (primary actor)** | Volunteer |
| **Activation conditions** | - The Volunteer accesses the "Check-in to Activity" function on the web or mobile application interface. |
| **Preconditions** | - The Volunteer has registered an account, logged into the system, and registered for an activity.  - The Volunteer has received a confirmation code or QR code after registration.  - The activity is in the "ongoing" status or within the allowed check-in time frame. |
| **Postcondition** | - The Volunteer's check-in status is successfully updated in the database  - The Volunteer receives a confirmation email for the successful check-in.  - The check-in history is recorded for reporting or analysis purposes. |
| **Main event stream** | 1. The volunteer accesses the "Check-in to Activity" feature on the interface.  2. The system displays a list of activities that the volunteer has registered for and are in a check-in allowed status.  3. The volunteer selects an activity from the list.  4. The volunteer check-in method: confirmation code generated as QR code.  5. The volunteer performs the check-in.  6. The system checks if the confirmation code is valid.  7. If the code is valid, the system updates the status.  8. The system sends a confirmation email to the volunteer including activity information (title, time, location).  9. The system displays the message "Check-in successful" on the interface.  10. The check-in process is complete. |
| **Sub-event stream** |  |

Table 3.2. Specification Use-case “Check-In Activities”

**2.1.3. Use-case 03 Specification**

|  |  |
| --- | --- |
| **Use Case Code** | UC03 |
| **Use Case Name** | Search Activities |
| **Describe** | - This use case enables a Volunteer to search for and view volunteer activities available upcoming or past activities and search. |
| **User (primary actor)** | Volunteer |
| **Activation conditions** | - The Volunteer accesses the "Search Activities" function from the web or mobile application interface.  - The Volunteer enters search criteria or filters to find specific activities. |
| **Preconditions** | - The Volunteer has registered an account and logged into the system.  - There are activities (upcoming or past) available in the system for the Volunteer to search. |
| **Postcondition** | - The system displays a list of activities matching the Volunteer's search criteria or browsing preferences. |
| **Main event stream** | 1. The Volunteer accesses the "Search Activities" function on the interface.  2. The system displays a default list of upcoming activities, showing basic details such as title, date/time, location, and remaining capacity.  3. The Volunteer enters search criteria or selects filters  4. The system updates the displayed list with the filtered results, showing the same basic details as in step 2.  5. The Volunteer selects an activity from the list to view its details.  6.If the Volunteer selects the map option:   * The system integrates with Google Maps API to display the activity location on an interactive map. * The Volunteer can interact with the map (e.g., zoom, view directions).   7. The Volunteer can return to the list to continue browsing or searching.  8. The search process is complete. |
| **Sub-event stream** | If no results are found, the system displays a “No matching activities found” message |

Table 3.3. Specification Use-case “Search Activities”

**2.1.4. Use-case 04 Specification**

|  |  |
| --- | --- |
| **Use Case Code** | UC04 |
| **Use Case Name** | History Activities |
| **Describe** | - This use case enables a Volunteer to view their history of registered or completed volunteer activities within the Volunteer system |
| **User (primary actor)** | Volunteer |
| **Activation conditions** | - The Volunteer accesses the "View Activity History" function from the web or mobile application interface, typically within their personal profile or dashboard.  = The Volunteer navigates to a dedicated history section to review past engagements. |
| **Preconditions** | - The Volunteer has registered an account and logged into the system.  - The Volunteer has previously registered for or participated in at least one activity in the system.  - The system has a stable connection to the database to retrieve the Volunteer's registration and check-in history.  - Activity records, including registration and check-in statuses, are accurately stored in the database. |
| **Postcondition** | - The system displays a list of the Volunteer's registered or completed activities, including details such as title, date/time, location, and check-in status.  - The Volunteer can view detailed information about each activity in their history.  - The system logs the Volunteer's interaction with the history section for analytics purposes  - The Volunteer's activity history remains unchanged in the database unless explicitly modified. |
| **Main event stream** | 1. The Volunteer accesses the "View Activity History" function on the interface, typically from their profile or dashboard.  2. The system retrieves the Volunteer's history from the database, including all activities they have registered for or participated in.  3. The system displays a list of activities, showing basic details such as title, date/time, location, statusand check-in status  4. The Volunteer can sort or filter the list to organize the displayed information.  5. The Volunteer selects an activity from the list to view its details.  6. The system displays the full details of the selected activity, including title, description, start/end date and time, location,, organization name, contact information, and the Volunteer's check-in status for that activity.  7. The Volunteer can return to the list to continue reviewing their history.  8.The history viewing process is complete. |
| **Sub-event stream** | - No Activity History Available   * In step 3, if the Volunteer has not registered for or participated in any activities, the system displays a message: "You have no activity history yet. Start by registering for an activity!" * The Volunteer is redirected to the activity search or registration page to explore available activities. |

Table 3.4. Specification Use-case “History Activities”

**2.1.5. Use-case 05 Specification**

|  |  |
| --- | --- |
| **Use Case Code** | UC05 |
| **Use Case Name** | Management Activities |
| **Describe** | - This use case allows a Coordinator to manage volunteer activities within the VolunteerSync system, including adding new activities, editing existing ones, deleting activities, updating their status and uploading or deleting illustrative images |
| **User (primary actor)** | Admin |
| **Activation conditions** | - The Admin accesses the "Manage Activities" function from the web or mobile application interface.  - The Admin receives a system notification or request to update activity information (e.g., due to a schedule change or new event). |
| **Preconditions** | - The Admin has logged into the system with the role authorized to manage activities.  - The system has a stable connection to the database to store and retrieve activity data.  - A cloud storage service (e.g., AWS S3) is configured and accessible for uploading and managing activity images.  - Existing activities in the system are synchronized and up-to-date before management begins. |
| **Postcondition** | - Activity information (title, description, date/time, status, images) is successfully added, edited, deleted, or updated in the database. |
| **Main event stream** | 1. The Admin accesses the "Manage Activities" function on the interface.  2. The system displays a list of existing activities, including title, start/end date and time, status, maximum capacity, number of registered Volunteers, and image links.  3. The Admin selects an action from the management menu (add, edit, remove…)  4. The system refreshes the activity list to reflect all changes.  5. The management process is complete. |
| **Sub-event stream** | 1. If required fields are missing → system prompts to complete the form  2. If update/delete fails → system shows an error and logs the issue |

Table 3.5. Specification Use-case “Management Activities”

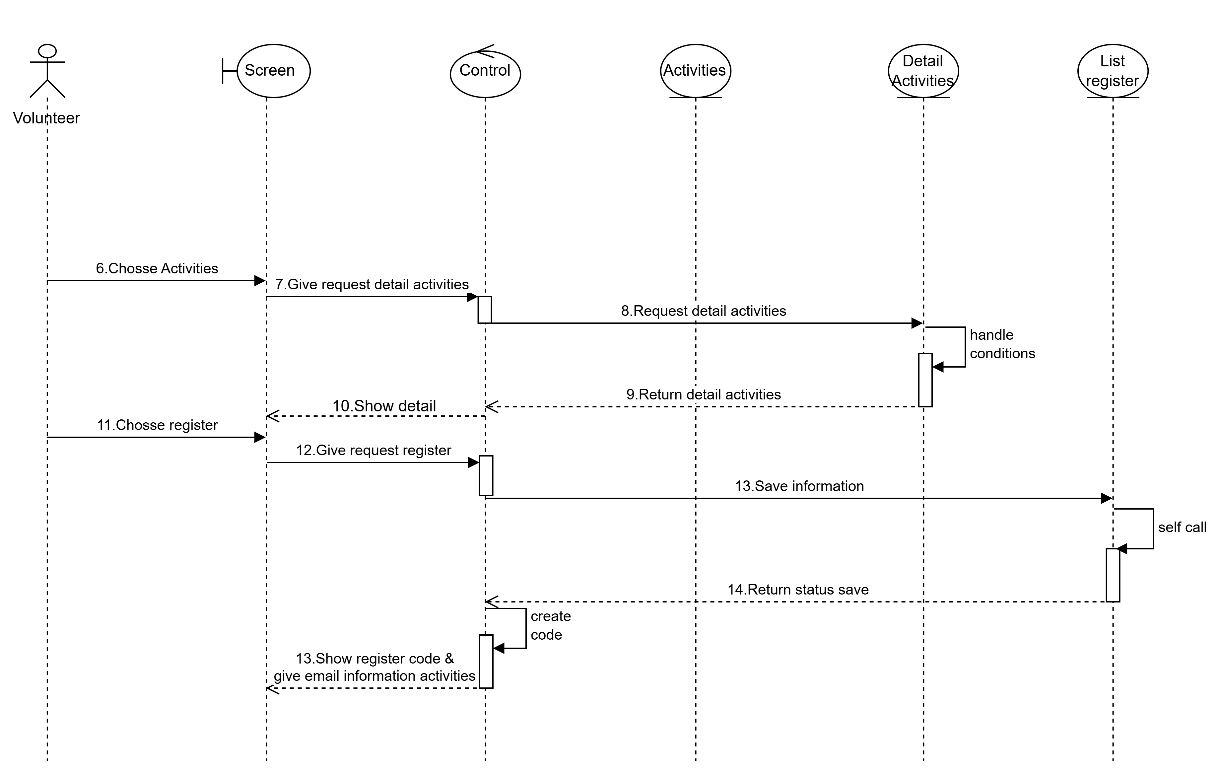
**2.1.6. Use-case 06 Specification**

|  |  |
| --- | --- |
| **Use Case Code** | UC06 |
| **Use Case Name** | Volunteer Management |
| **Describe** | - This use case enables a Coordinator to manage Volunteers within the VolunteerSync system for a specific activity. The Admin can view a list of registered Volunteers, approve or reject their registrations, verify check-ins (QR code), send email notifications for activity updates or check-in confirmations, and view activity ratings submitted by Volunteers |
| **User (primary actor)** | Admin |
| **Activation conditions** | - The Admin accesses the "Volunteer Management" function from the web or mobile application interface, typically after selecting a specific activity.  - The Admin receives a system notification or request to review Volunteer registrations or check-ins. |
| **Preconditions** | - The Admin has registered an account and logged into the system with the role authorized to manage Volunteers.  - The activity for which Volunteers are being managed exists in the system and has associated registrations.  - The system has a stable connection to the database to retrieve and update Volunteer registration and check-in data. |
| **Postcondition** | - The Coordinator successfully views, approves, rejects, or verifies Volunteer registrations and check-ins in the database.  - Email notifications are sent to Volunteers for significant updates or confirmations (if applicable).  - Activity ratings submitted by Volunteers are displayed for the Coordinator’s review.  - The Volunteer list, including check-in statuses, can be exported as a CSV or Excel file.  - All actions (approvals, rejections, check-ins, exports) are logged for auditing and reporting purposes. |
| **Main event stream** | 1.The Coordinator accesses the "Volunteer Management" function on the interface after selecting a specific activity.  2. The system displays a list of Volunteers registered for the activity, including their name, registration date, role (if applicable), and check-in status (e.g., checked-in, not checked-in).  3. The Coordinator selects an action from the management options:   * View the list of registered Volunteers. * Approve or reject a Volunteer’s registration. * Verify a Volunteer’s check-in. * Send email notifications to Volunteers. * View activity ratings submitted by Volunteers. * Export the Volunteer list.   4. The system refreshes the Volunteer list to reflect any updates.  5. The Volunteer management process is complete. |
| **Sub-event stream** | 1. No Volunteers Registered for the Activity   * In step 2, if no Volunteers are registered for the activity, the system displays: "No Volunteers registered for this activity yet." * The Coordinator can return to the activity list or create a new activity.   2. Invalid Confirmation Code or QR Code for Check-in   * In step 3, if the confirmation code is invalid or the QR code scan fails, the system displays: "Invalid code. Please try again or contact support." * The Coordinator can retry or manually mark the check-in if permitted.   3. Email Sending Fails   * In step 3 if the EmailService fails, the system displays: "Action completed, but email sending failed. Please try again later."   The system logs the error and retries sending in the background. |

Table 3.6. Specification Use-case “Volunteer Management”

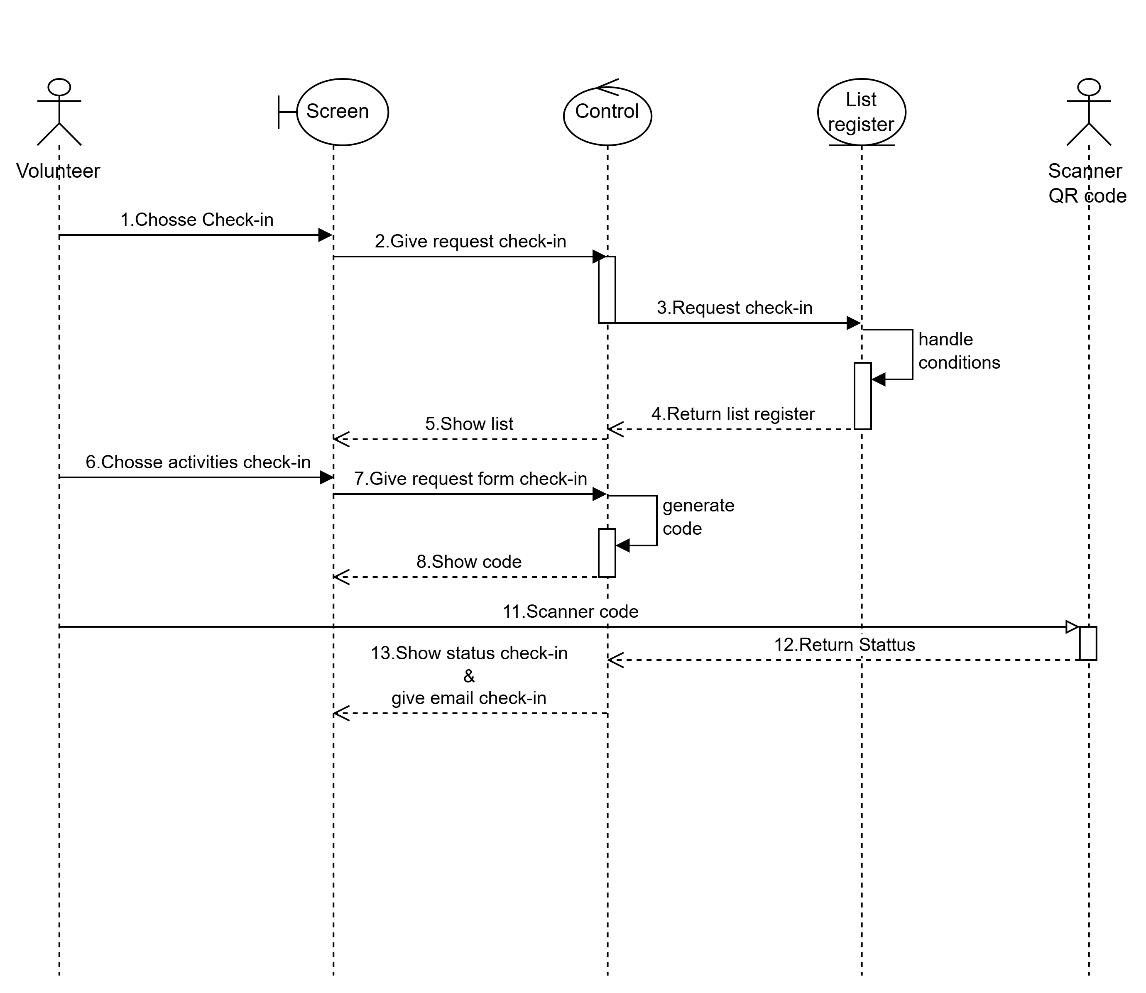
### **2.2. Sequence Diagram**

**2.2.1. Sequence Register Activities**

****

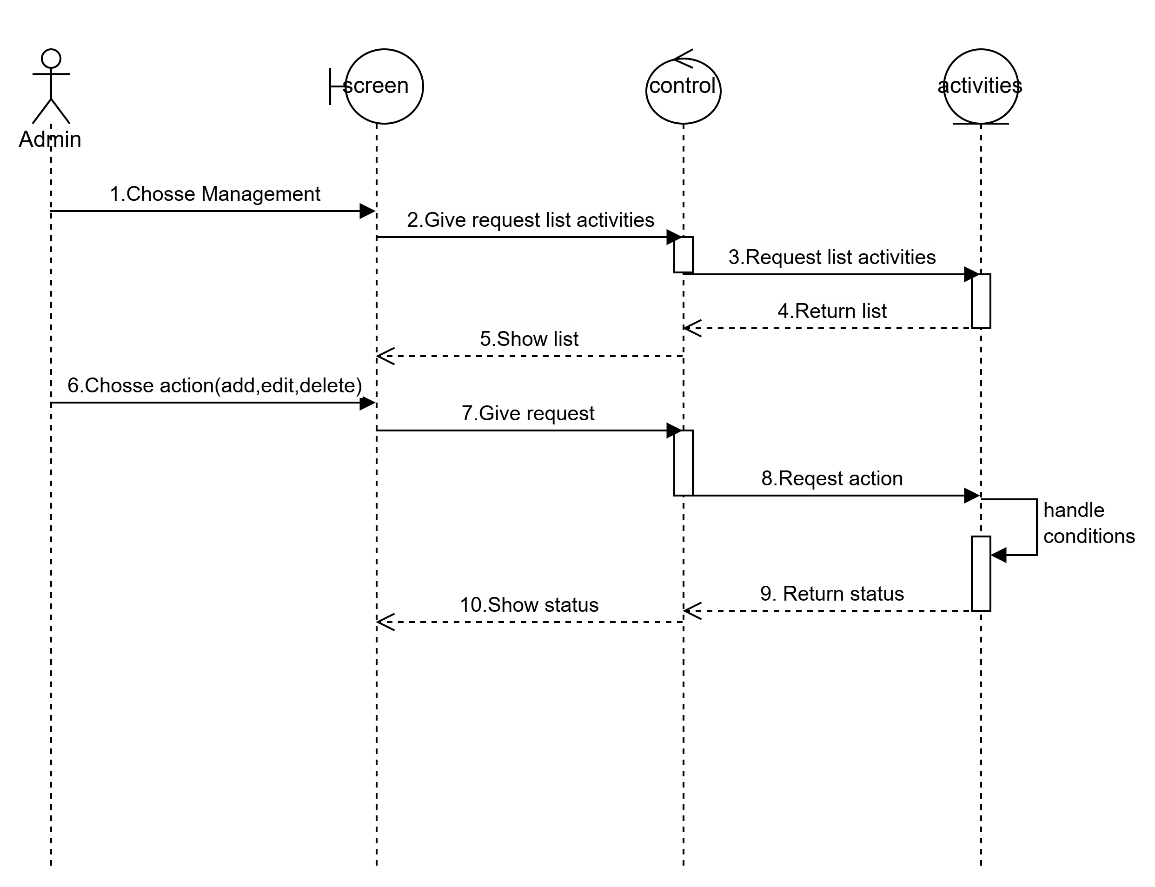
Picture 3.2. Sequence Register Activities

First, the user interacts with the "Screen" to select an activity. The "Screen" sends a request for activity details to the "Control", which then sends a request for activity details to the "Activities". The "Activities" handle conditions related to the activity (e.g., checking information, activity status). This is an internal process within the "Activities". The "Activities" returns the details of the selected activity to the "Control". After receiving the activity details, the user on the "Screen" decides to register and selects the registration function. The "Screen" sends a registration request to the "Control". The "Control" sends a request to save the registration information to the "List register". The "List register" performs an internal "self-call" process, which may be to process data, check validity, or generate a registration code. "List register" returns the status of the information saving process (success or failure) to "Control". "Control" will generate the registration code after receiving the successful saving status. Finally, "Control" sends the registration code and operation information to the user via email, and "Screen" displays this registration code.

**2.2.2. Sequence Check-in**

Picture 3.3. Sequence Check-in

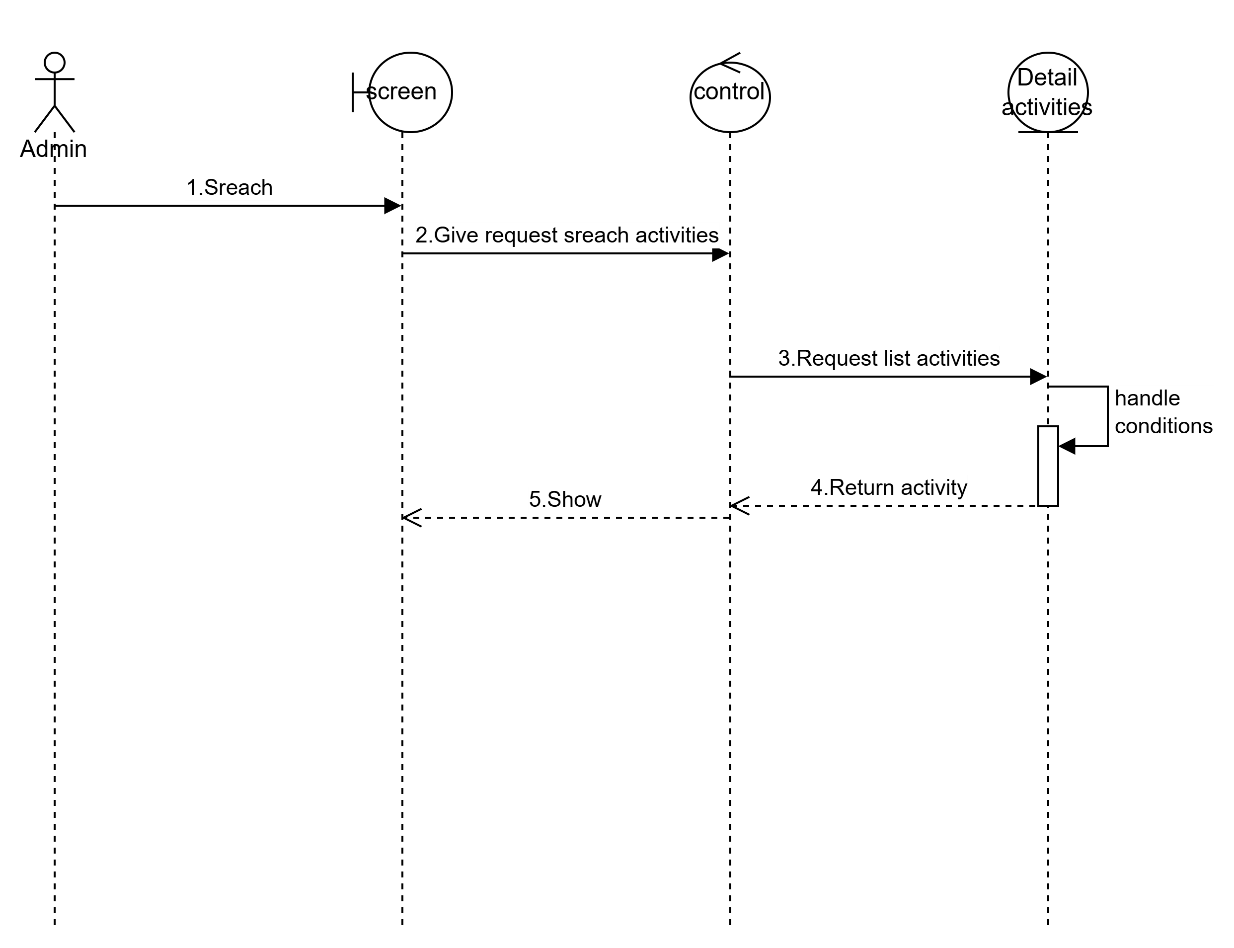
Initially, the user will interact with the "Screen" to select the "Check-in" function. The "Screen" sends a request to perform the check-in function to the "Control". The "Control" then sends the check-in request to the "List register". The "List register" performs internal processing to handle conditions related to the check-in request (e.g. checking the validity of information, registration status). The "List register" returns a list of registrations related to the check-in request to the "Control". The "Control" sends this list to the "Screen" to display to the user. The user on the "Screen" selects a specific activity to check-in from the displayed list. The "Screen" sends a check-in form request to the "Control". The "Control" displays a QR code on the "Screen" to serve the check-in process. A code scanner is used to read the code displayed on the "Screen" or the code provided by the user. After scanning the code, the processing takes place and returns the status of the check-in (e.g. successful, failed, checked-in) to the "Control". Finally, the "Control" displays the check-in status on the "Screen" and sends the check-in confirmation information via email to the user.

**2.2.3. Sequence Management Activities**

Picture 3.4. Sequence Management Activities

The user interacts with the "Screen" to select a "Management" function (which can be managing activities or some kind of data). The "Screen" sends a request to get a list of activities to the "Control". The "Control" then sends a request to the "Activities" for a list of activities. The "Activities" returns a list of activities to the "Control". The "Control" sends this list to the "Screen" to display to the user. The user on the "Screen" selects a specific management action: add, edit, or delete an activity. The "Screen" sends a request to perform the selected action (add, edit, delete) along with the necessary information to the "Control". The "Control" then sends a request to perform the action to the "Activities". The "Activities" perform internal processing to handle conditions related to the action (e.g., checking data validity when adding/editing, checking permissions when deleting). "Activities" returns the status of the performed action (success or failure) to the "Control". The "Control" displays the status of the action on the "Screen" to notify the user.

**2.2.4.** **Sequence Sreach Activities**

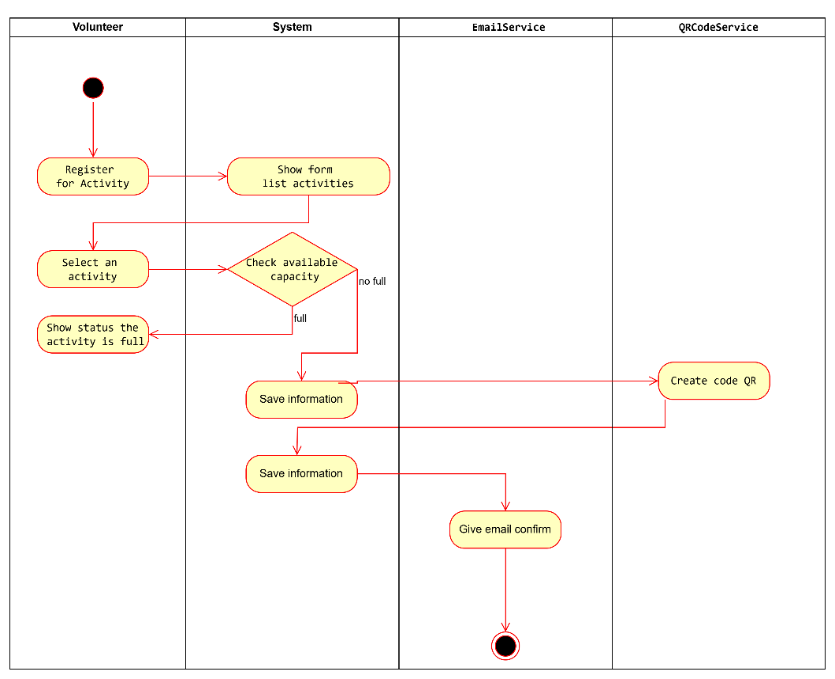
****

Picture 3.5. Sequence Sreach Activities

The user interacts with the "Screen" to perform the search action. The "Screen" sends a request to search for activities to the "Control". The "Control" further sends a request for a list of activities (based on the search criteria) to the "Detail activities". The "Detail activities" perform internal processing to handle the conditions related to the search request (e.g. filtering, sorting, checking data). The "Detail activities" returns the search results (matching activities) to the "Control". The "Control" sends these search results to the "Screen" to display to the user.

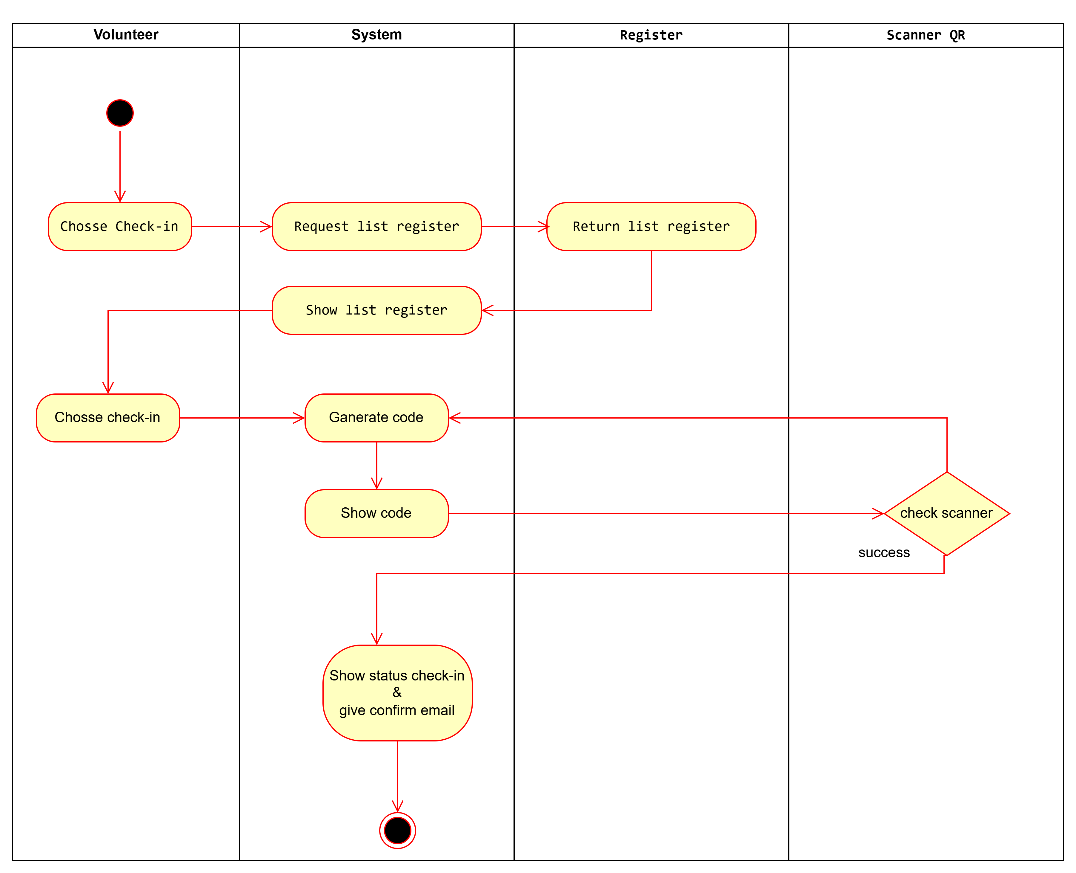
### **2.3. Activity Diagram**

**2.3.1.** **Activities Register**

****

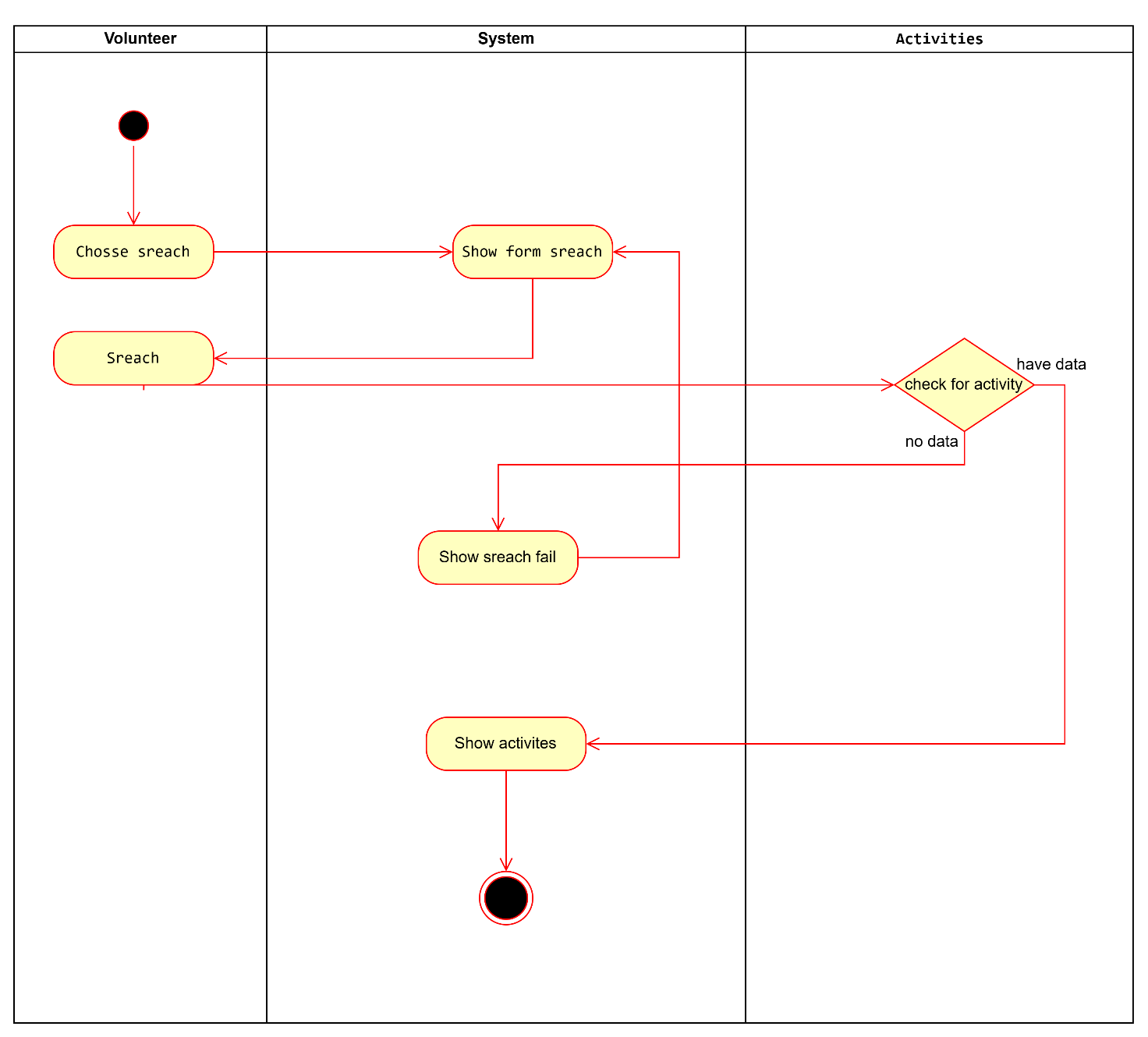
Picture 3.6. Activities Register

**2.3.2.** **Activitiy check-in**

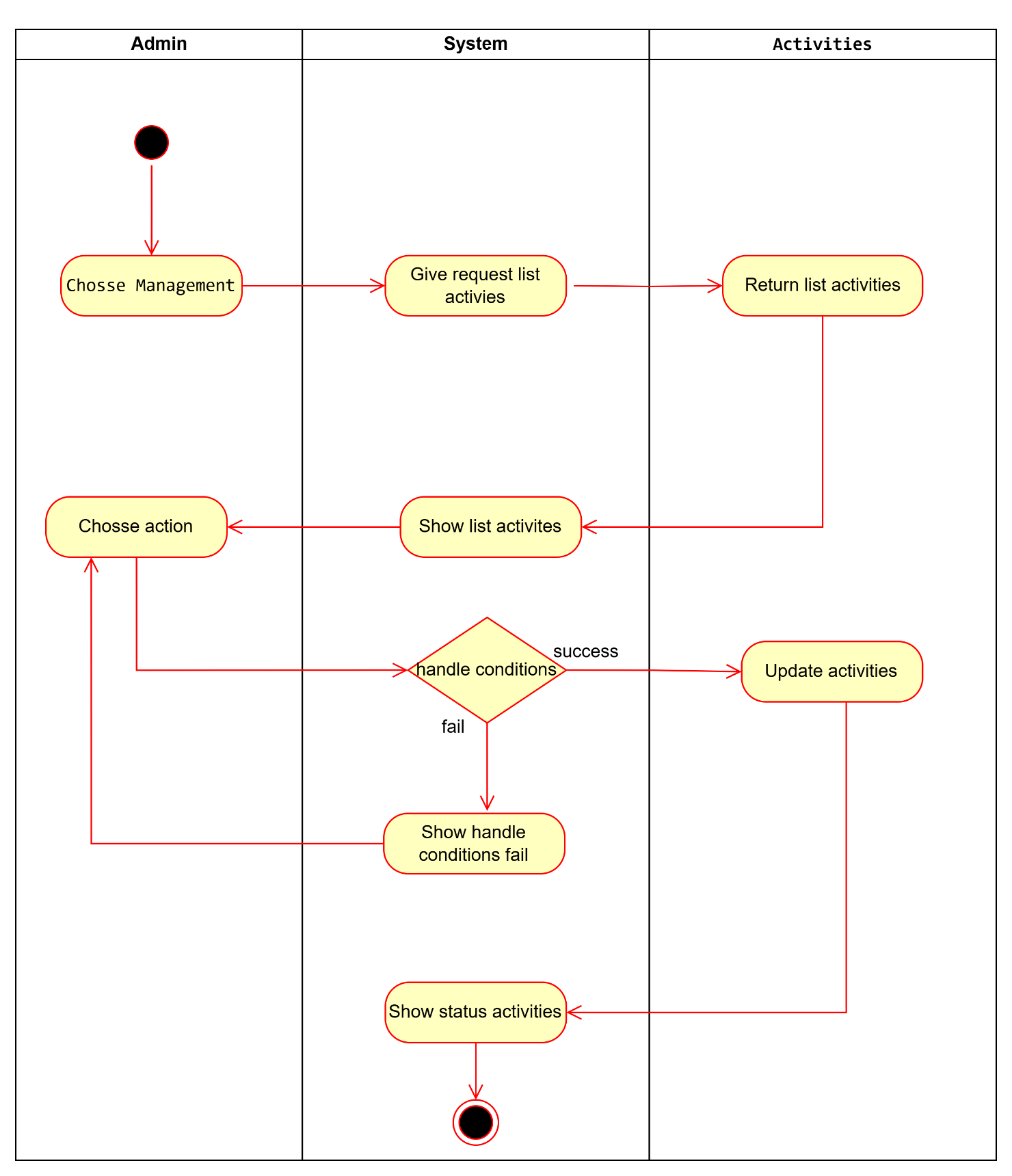
****

Picture 3.7. Activitiy check-in

**2.3.3.** **Activities Sreach**

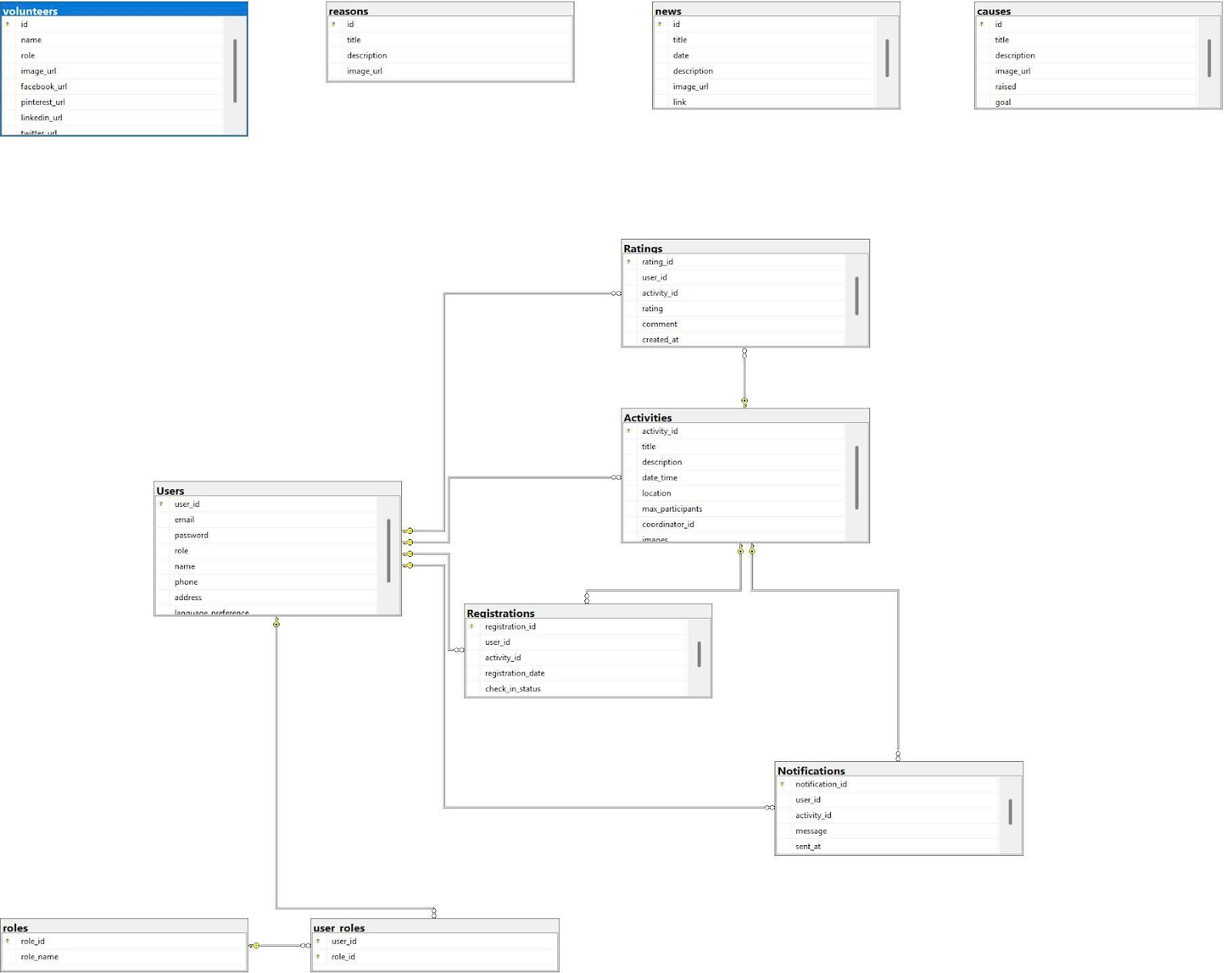
****

Picture 3.8. Activities Sreach

**2.3.4.** **Activities management**

Picture 3.9. Activities management

### **2.4. DataBase Diagram**

****

Picture 3.10. Database Diagram

**2.4.1 Table Users**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| user\_id | User code | BigInt | Primary key |
| email | User email | VarChar(255) | Not Null |
| password | User password | VarChar(255) | Not Null |
| role | Role | VarChar(50) | Not Null |
| name | User name | VarChar(255) | Not Null |
| phone | Phone number | VarChar(20) | Not Null |
| address | Address | Text | Null |
| language\_preference | Preferred Language | VarChar(50) | Null |

Table 3.7. Table Users

**2.4.2 Table Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| role\_id | Role code | Int | Primary key |
| role\_name | Role Name | VarChar(50) | Not Null |

Table 3.8. Table Roles

**2.4.3 Table User\_Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| user\_id | User code | BigInt | Foreign key |
| role\_id | Role code | Int | Foreign key |

Table 3.9. Table User\_Roles

**2.4.4 Table Activities**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| activity\_id | Activity code | Int | Primary key |
| title | Title | VarChar(200) | Not Null |
| description | Describe | Text | Null |
| date\_time | Time of event | DateTime | Not Null |
| location | Location | VarChar(255) | Not Null |
| max\_participants | Maximum number of participants | Int | Not Null |
| coordinator\_id | Coordinator code | BigInt | Foreign key |
| images | Image | VarChar(500) | Null |

Table 3.10. Table Activities

**2.4.5 Table Registrations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| registration\_id | Registration code | Int | Primary key |
| user\_id | User code | BigInt | Foreign key |
| activity\_id | Activity code | Int | Foreign key |
| registration\_date | Registration date | Date | Not Null |
| check\_in\_status | Attendance status | Boolean | Not Null |

Table 3.11. Table Registrations

**2.4.6 Table Ratings**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| rating\_id | Rating Code | Int | Primary key |
| user\_id | User code | BigInt | Foreign key |
| activity\_id | Activity code | Int | Foreign key |
| rating | Rating Points | Int | Not Null |
| comment | Comment | Text | Null |
| created\_at | Creation time | DateTime | Not Null |

Table 3.12. Table Ratings

**2.4.7 Table Notifications**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| notification\_id | Notification code | Int | Primary key |
| user\_id | User code | BigInt | Foreign key |
| activity\_id | Activity code | Int | Foreign key |
| message | Notification content | Text | Not Null |
| sent\_at | Date sent | DateTime | Not Null |

Table 3.13. Table Notifications

**2.4.8 Table Volunteers**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| id | Volunteer code | Int | Primary key |
| name | Volunteer Name | VarChar(255) | Not Null |
| role | Role | VarChar(100) | Not Null |
| image\_url | Avatar | VarChar(255) | Null |
| facebook\_url | Facebook Link | VarChar(255) | Null |
| pinterest\_url | Pinterest Link | VarChar(255) | Null |
| linkedin\_url | LinkedIn Links | VarChar(255) | Null |
| twitter\_url | Twitter Link | VarChar(255) | Null |

Table 3.14. Table Volunteers

**2.4.9 Table News**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| id | News code | Int | Primary key |
| title | Title | VarChar(255) | Not Null |
| date | Date posted | Date | Not Null |
| description | Describe | Text | Null |
| image\_url | Image | VarChar(255) | Null |
| link | Link | VarChar(255) | Null |

Table 3.15. Table News

**2.4.10 Table Causes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| id | Reason code | Int | Primary key |
| title | Title | VarChar(255) | Not Null |
| description | Describe | Text | Null |
| image\_url | Image | VarChar(255) | Null |
| raised | Amount raised | Float | Not Null |
| goal | Target | Float | Not Null |

Table 3.16. Table Causes

**2.4.11 Table Reasons**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute name** | **Interpretation** | **Data type** | **Binding** |
| id | Reason code | Int | Primary key |
| title | Title | VarChar(255) | Not Null |
| description | Describe | Text | Null |
| image\_url | Image | VarChar(255) | Null |

Table 3.17. Table Reasons

### **2.5. Website Sketch**

**2.5.1. Interface**

**2.5.1.1. Website home page**

A green and white rectangular object with black text

AI-generated content may be incorrect.

Picture 3.11. Website home page

A green and white card

AI-generated content may be incorrect.

Picture 3.12. Website home page

A green and white website

AI-generated content may be incorrect.

Picture 3.13. Website home page

A screenshot of a website

AI-generated content may be incorrect.

Picture 3.14. Website home page

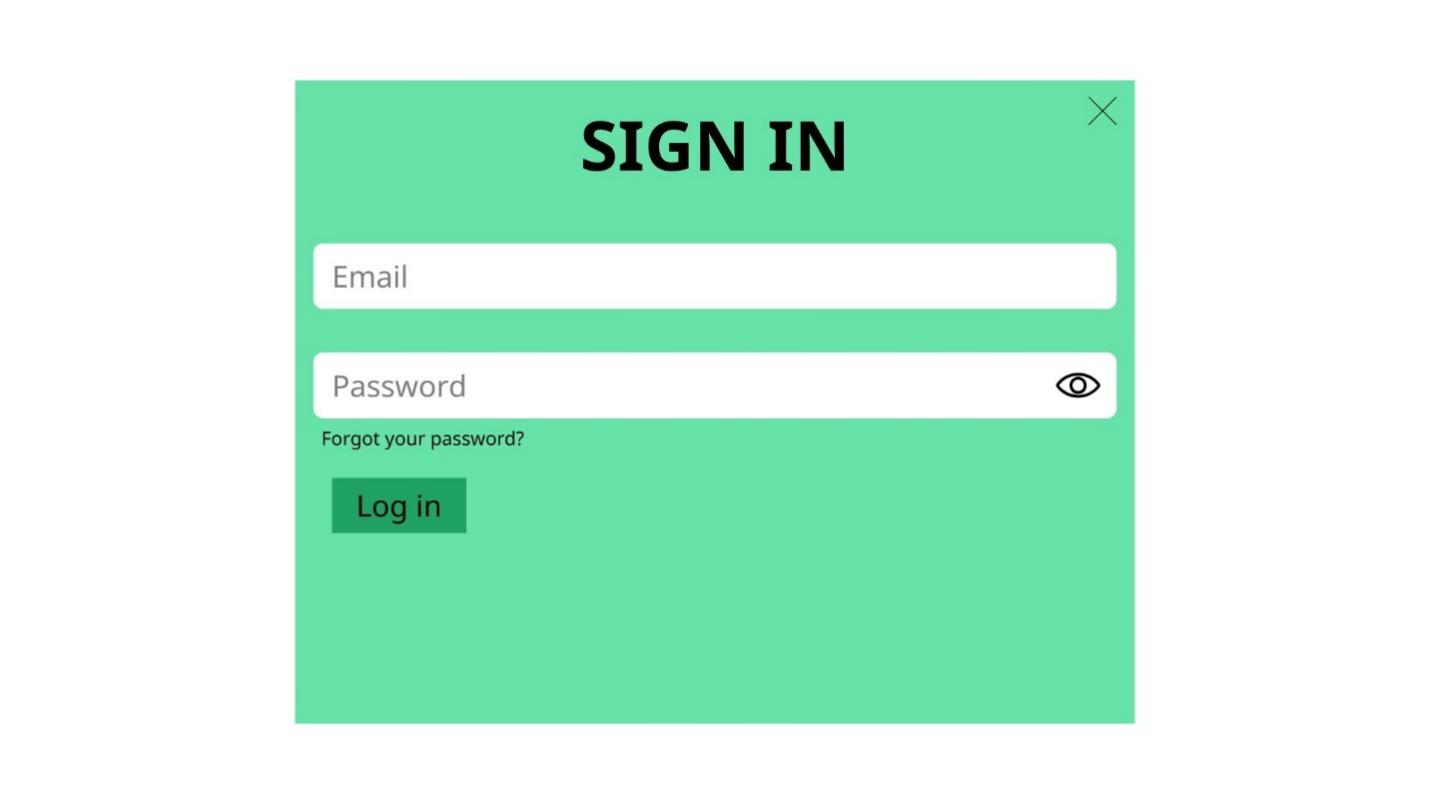
A screenshot of a donation form

AI-generated content may be incorrect.

Picture 3.15. Website home page

**2.5.2. Function**

**2.5.2.1. Log in of Admin & Volunteer**

****

Picture 3.14. Log in of Admin

**2.5.2.2. Register of Volunteer**

****

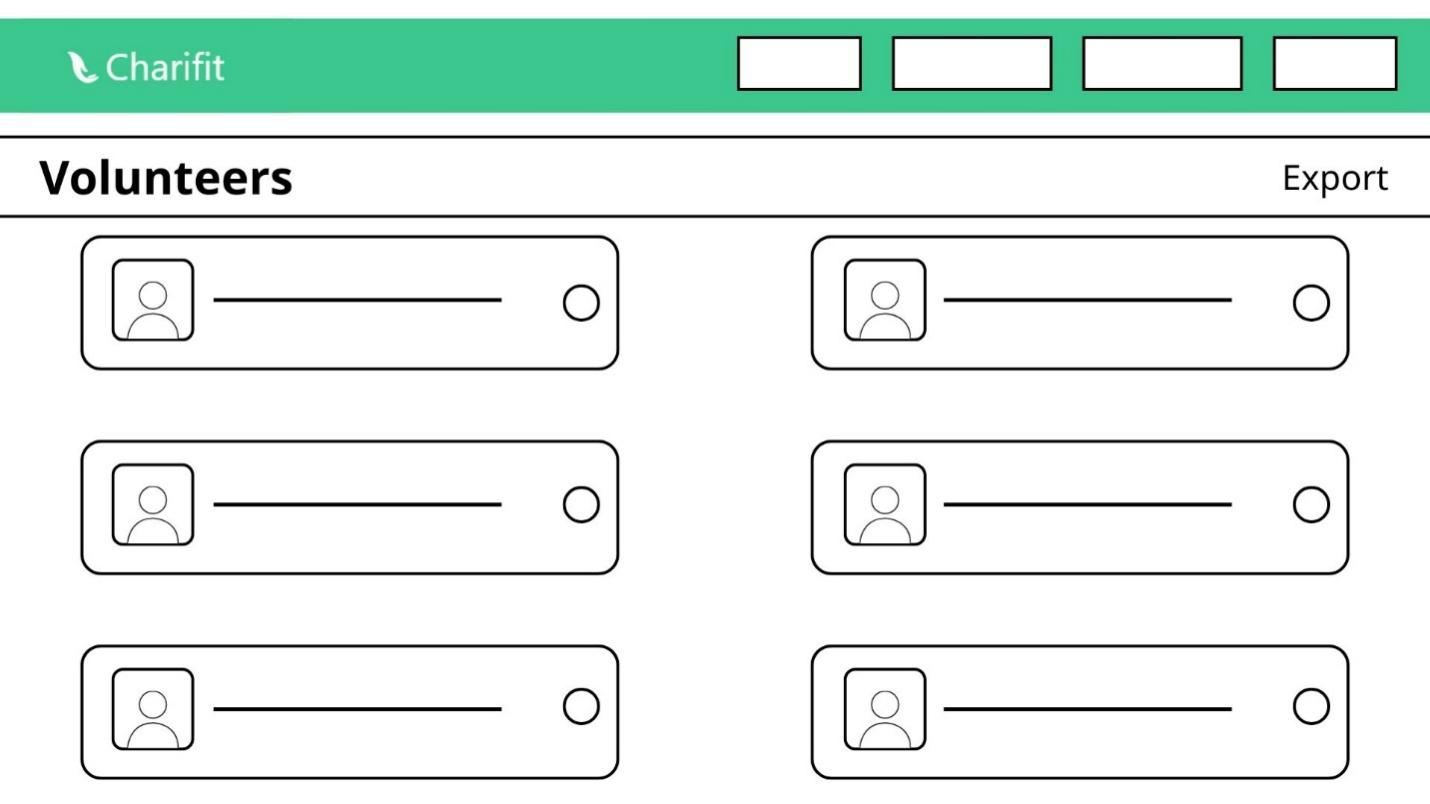
Picture 3.15. Admin Dashboard

**2.5.2.3. Admin Dashboard**

****

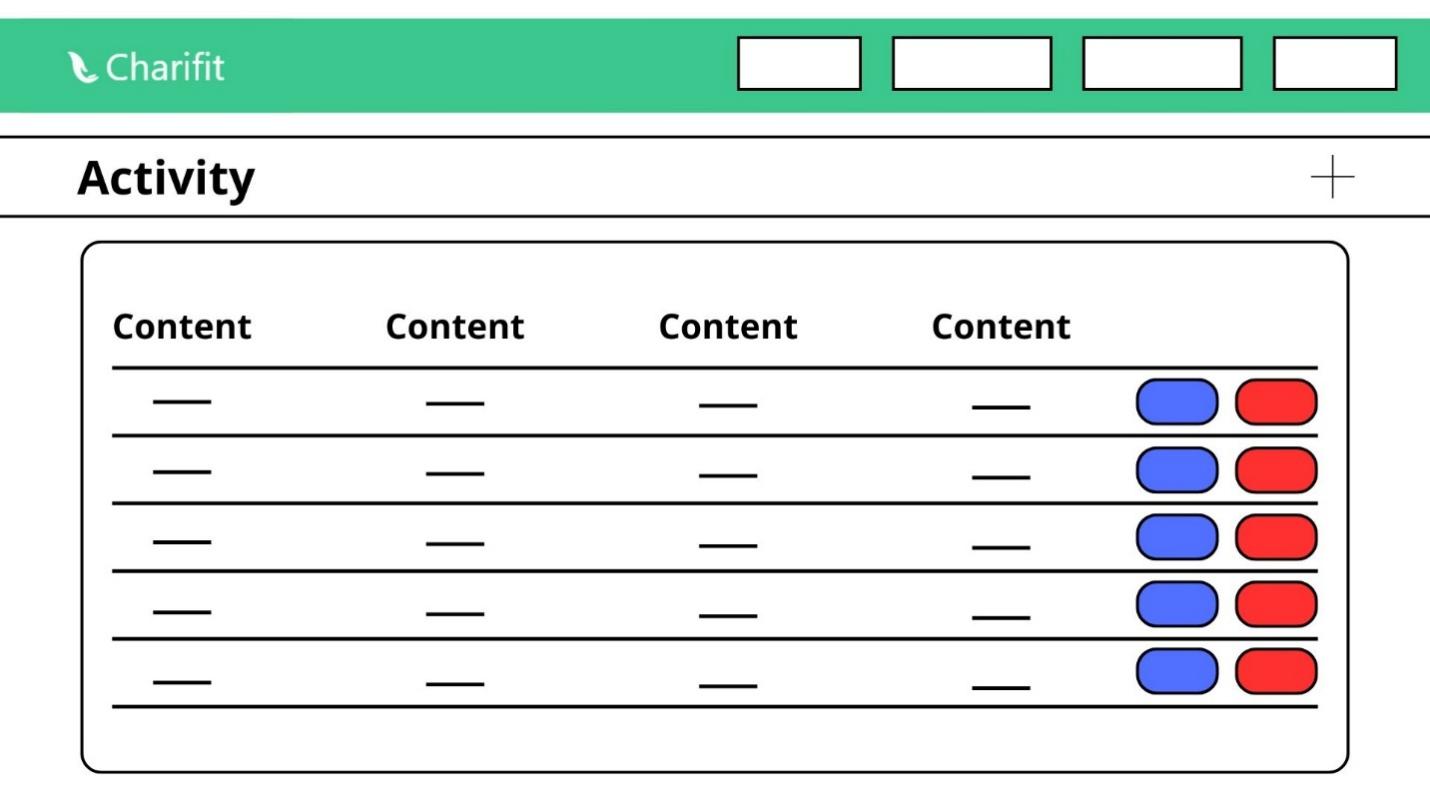
Picture 3.16. Admin Dashboard

**2.5.2.3. Administrator users management function**

****

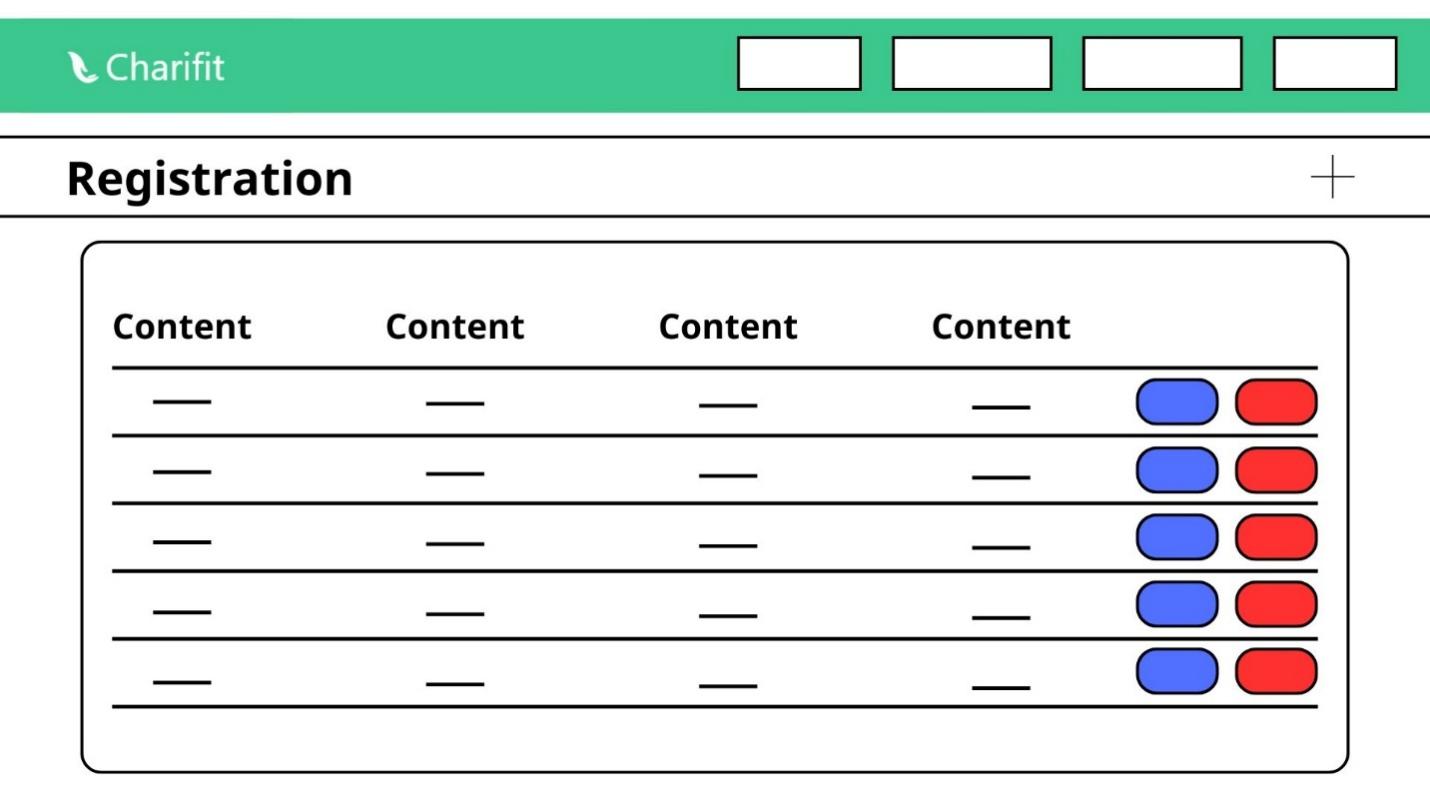
Picture 3.17. Administrator users management function

**2.5.2.4. Administrator activities management function**

****

Picture 3.18. Administrator activities management function

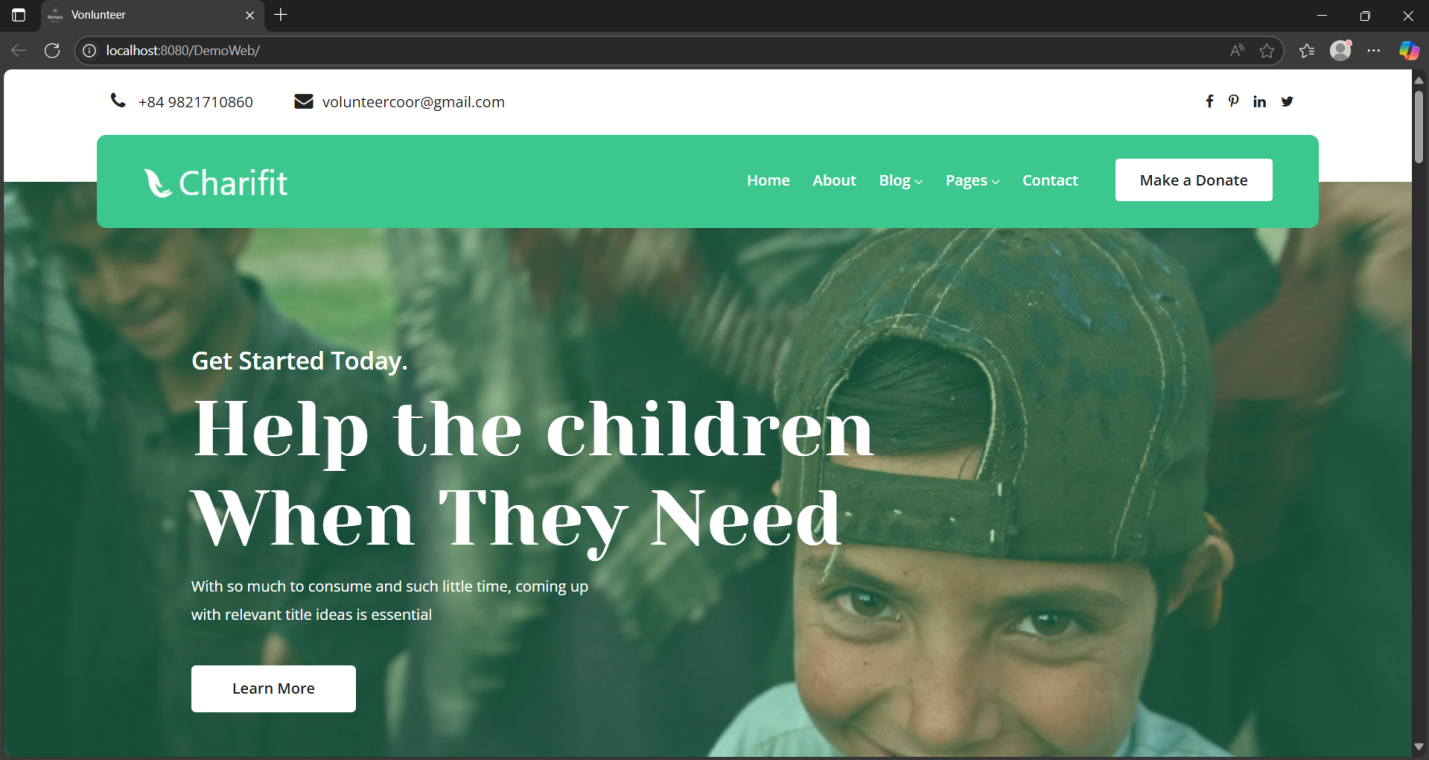
**2.5.2.5. Administrator registrations management function**

****

Picture 3.19. Administrator registrations management function

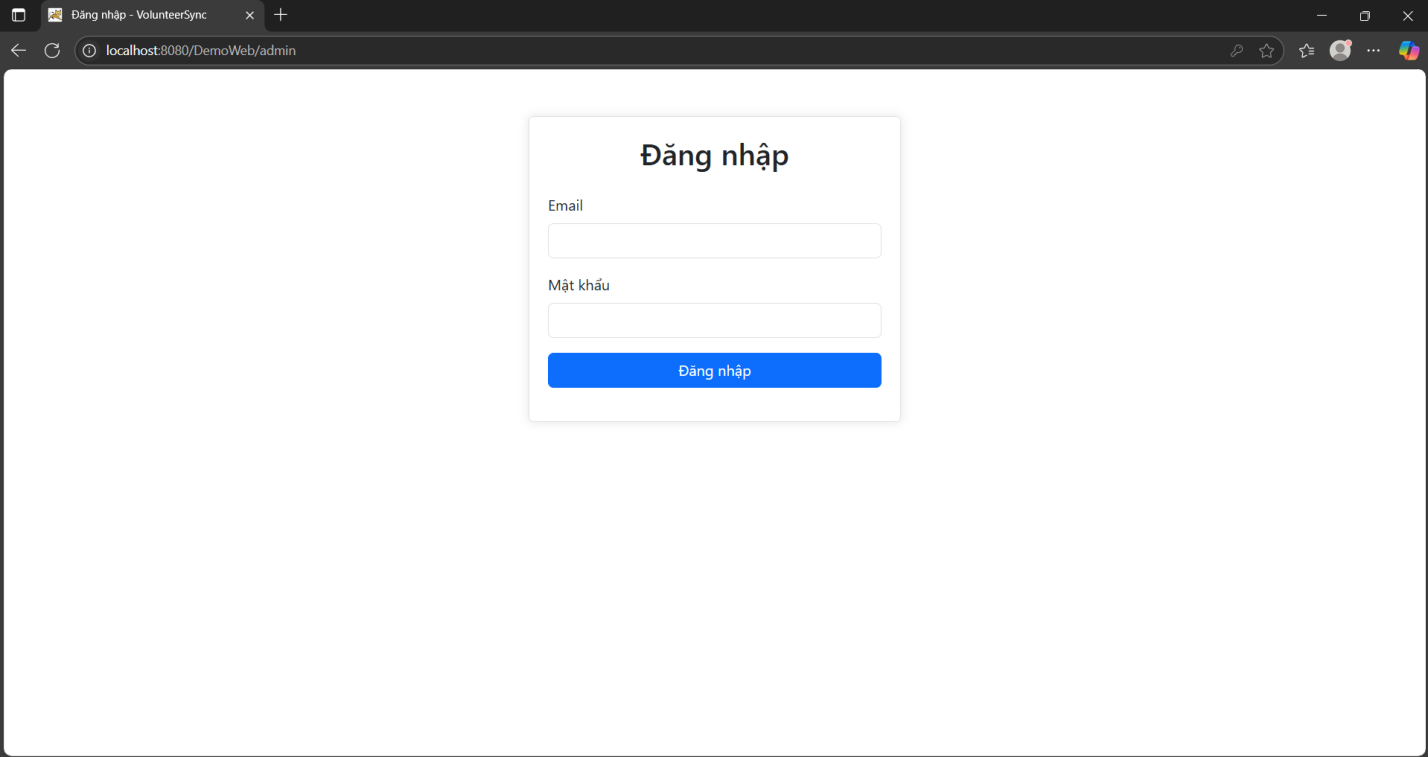
# **CHAPTER 4: BUILD SYSTEM**

## **1. Website home page**



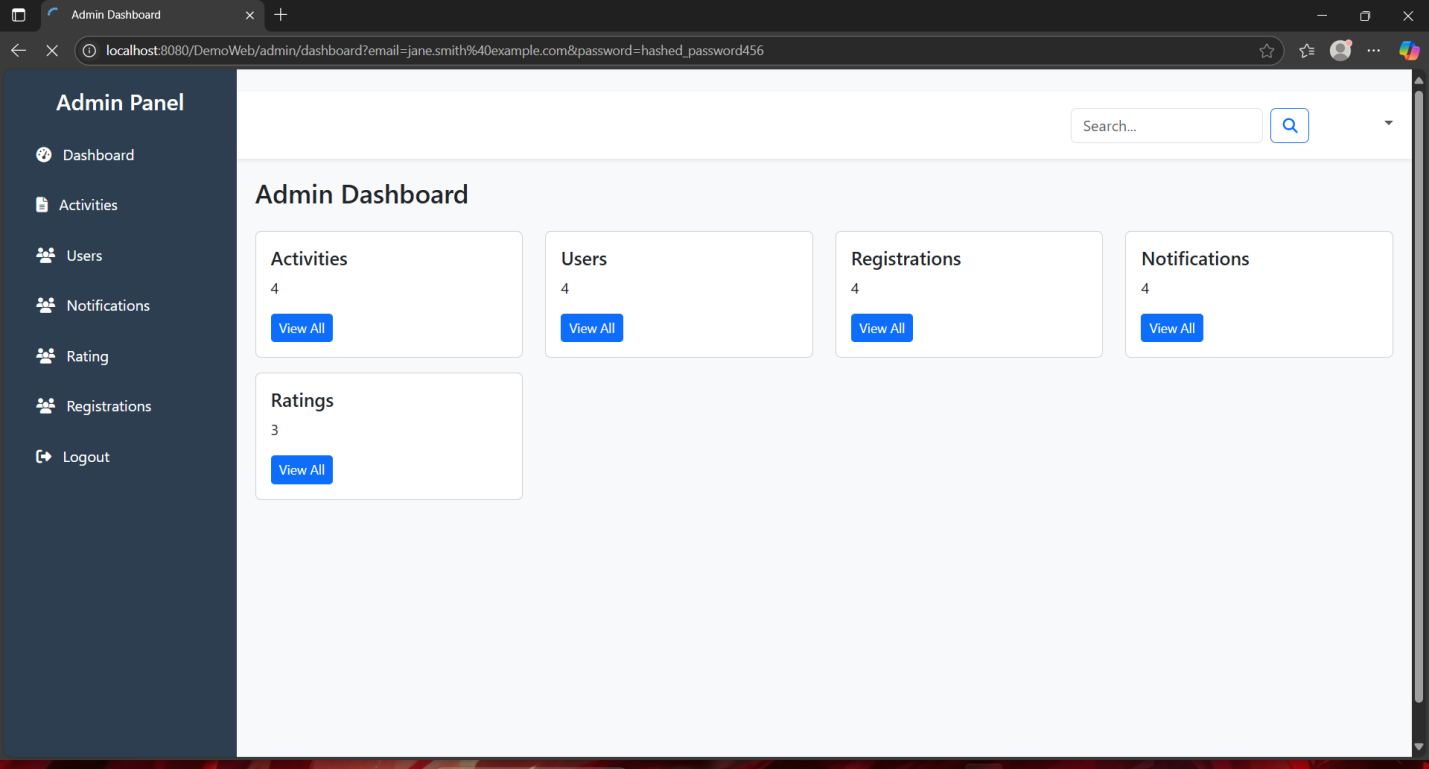
Picture 4.1. Website home page

## **2. Login of Admin**



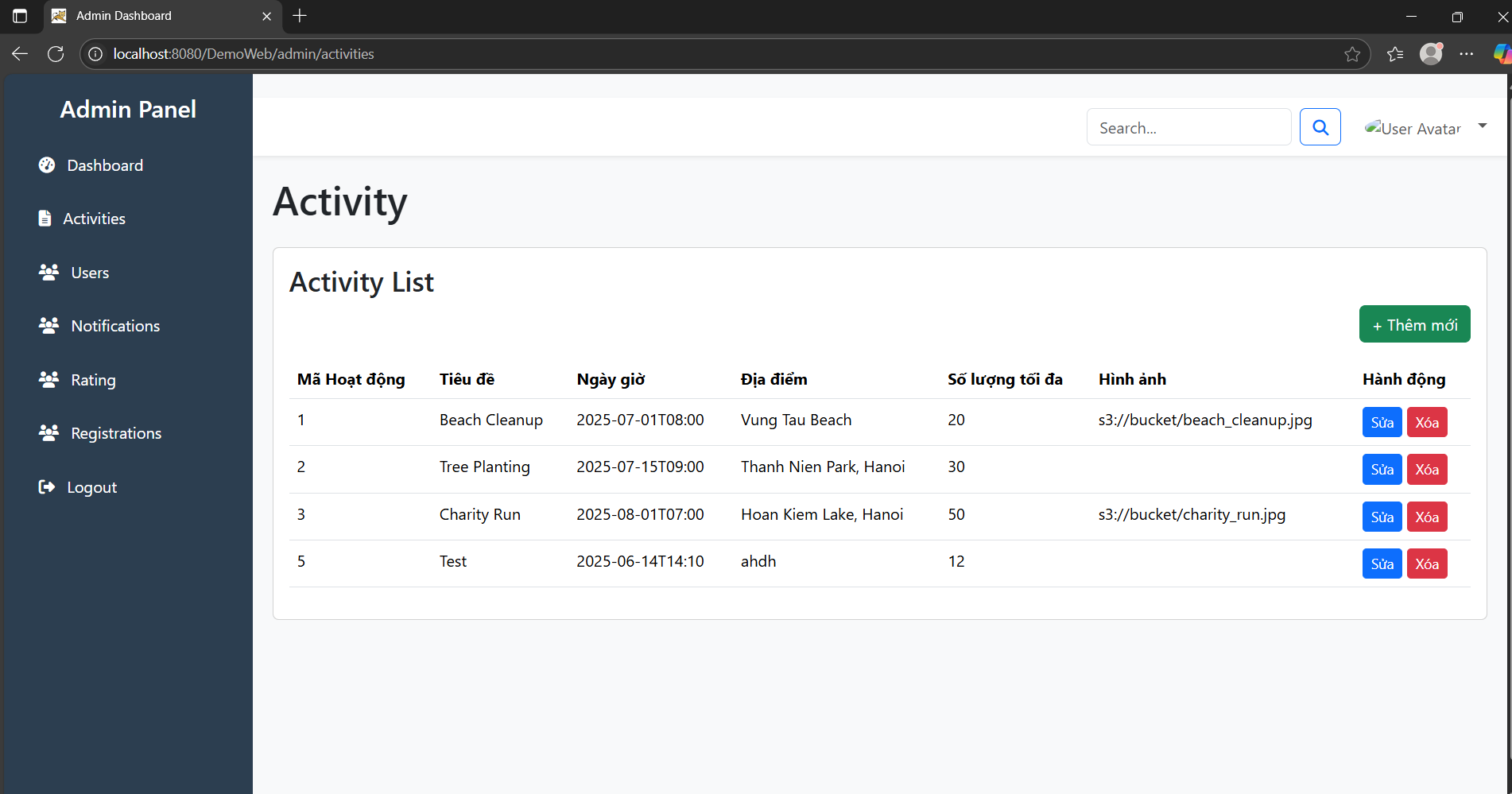
Picture 4.2. Log in of Admin

## **3. Admin dashboard**



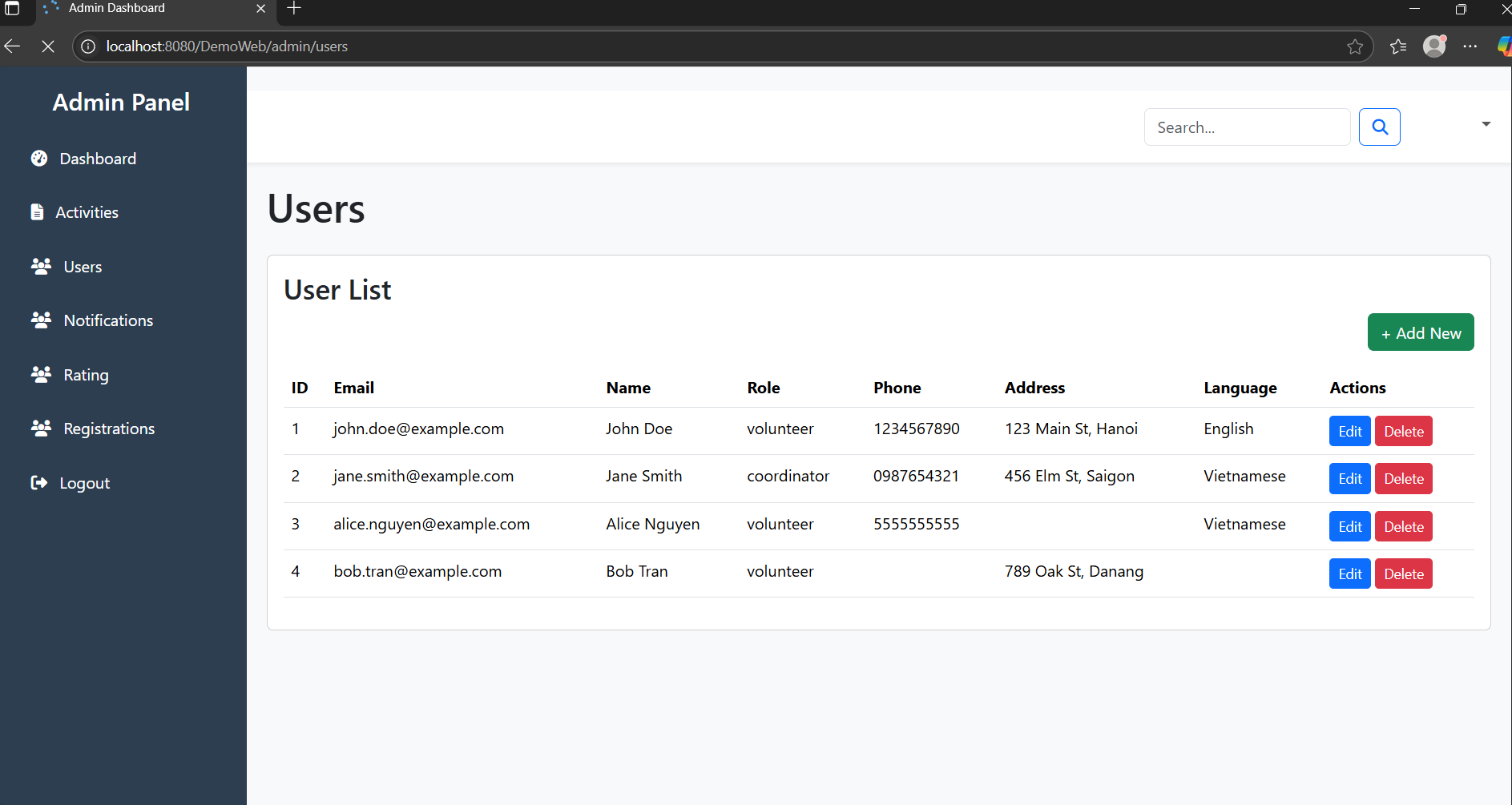
Picture 4.3. Admin Dashboard

## **4. Administrator activities management function**



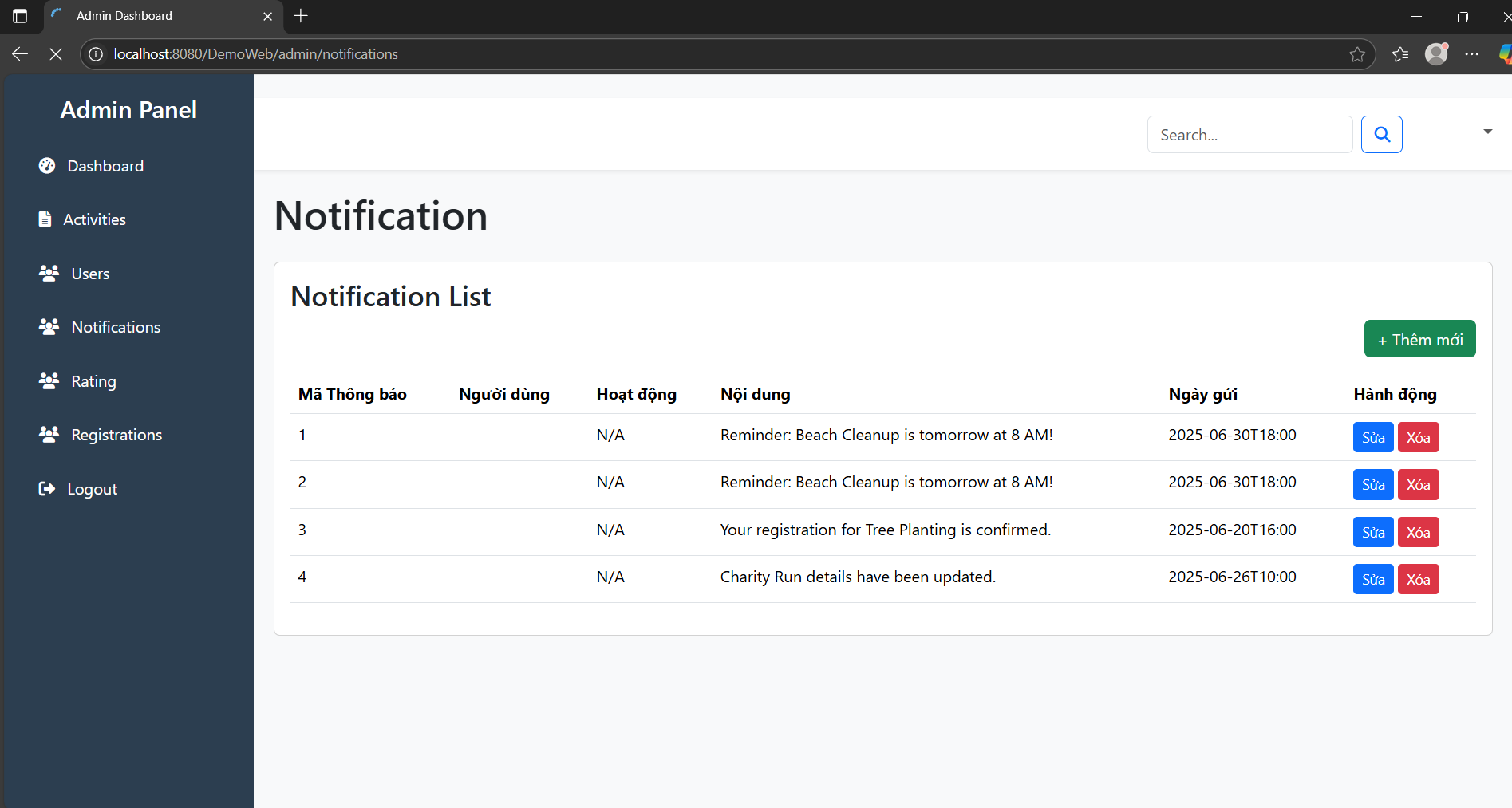
Picture 4.4. Administrator activities management function

## **5. Administrator users management function**



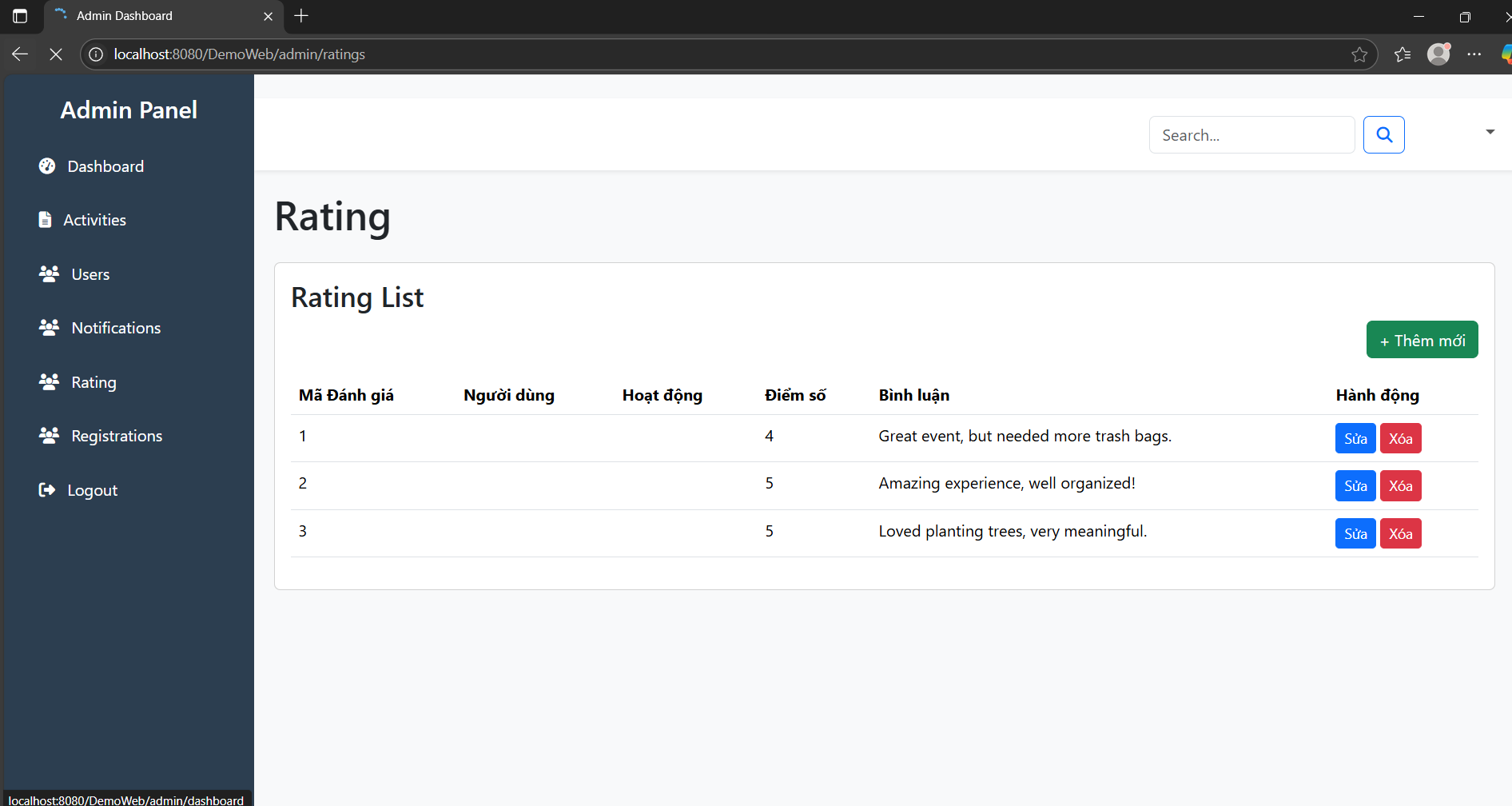
Picture 4.5. Administrator users management function

## **6. Administrator notifications management function**



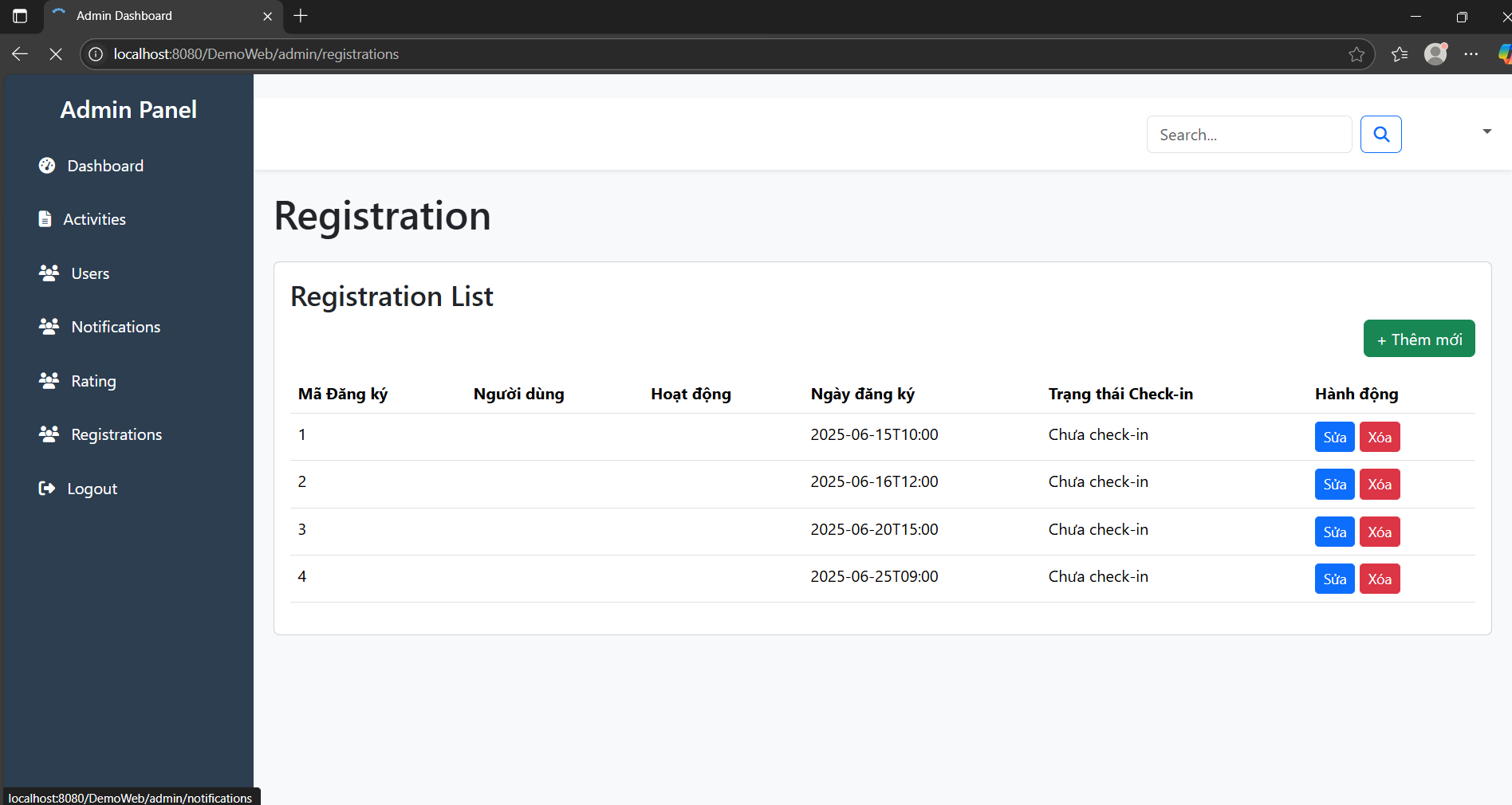
Picture 4.6. Administrator notifications management function

## **7. Administrator ratings management function**



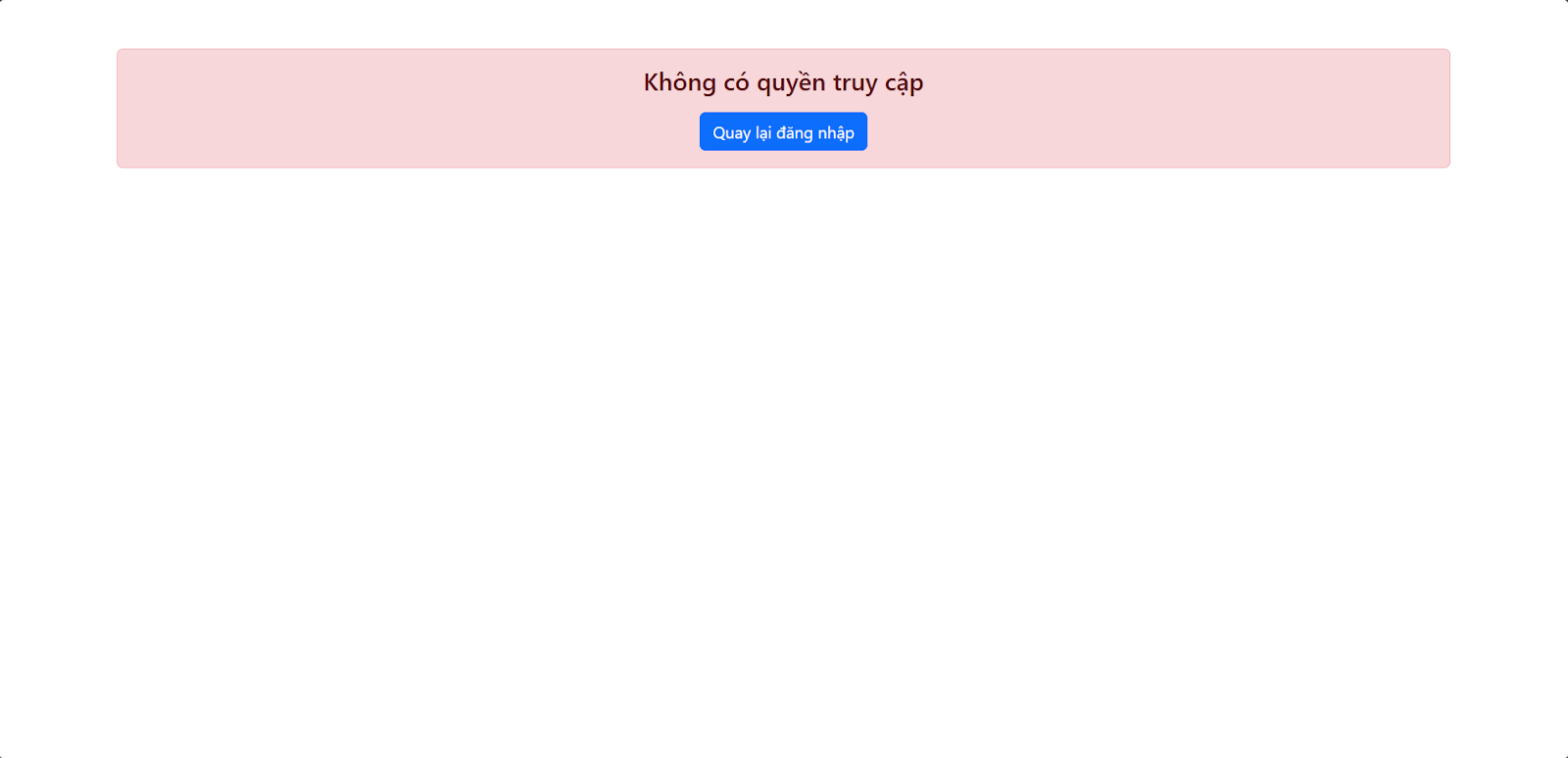
Picture 4.7. Administrator ratings management function

## **8. Administrator registrations management function**



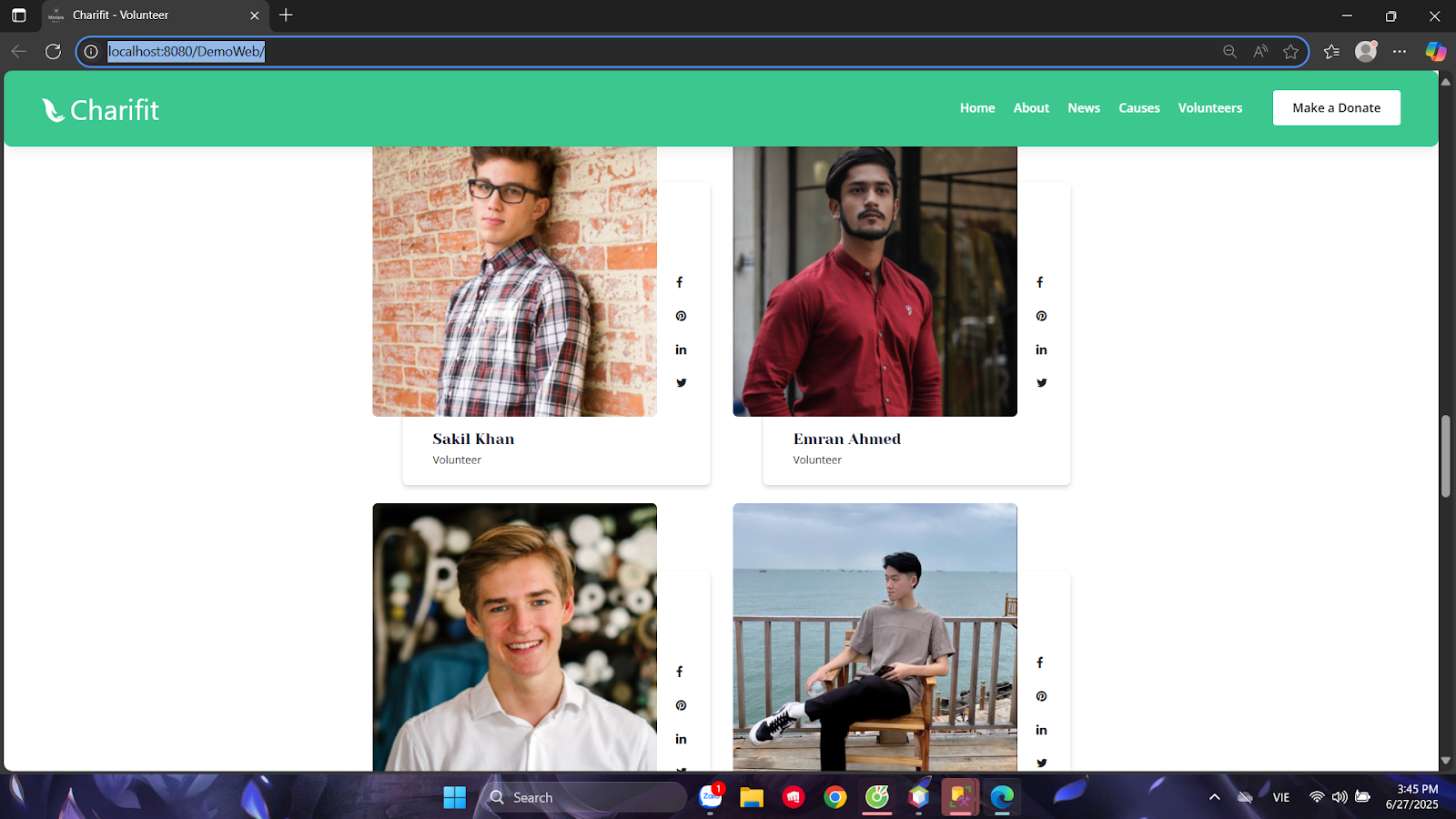
Picture 4.8. Administrator registrations management function

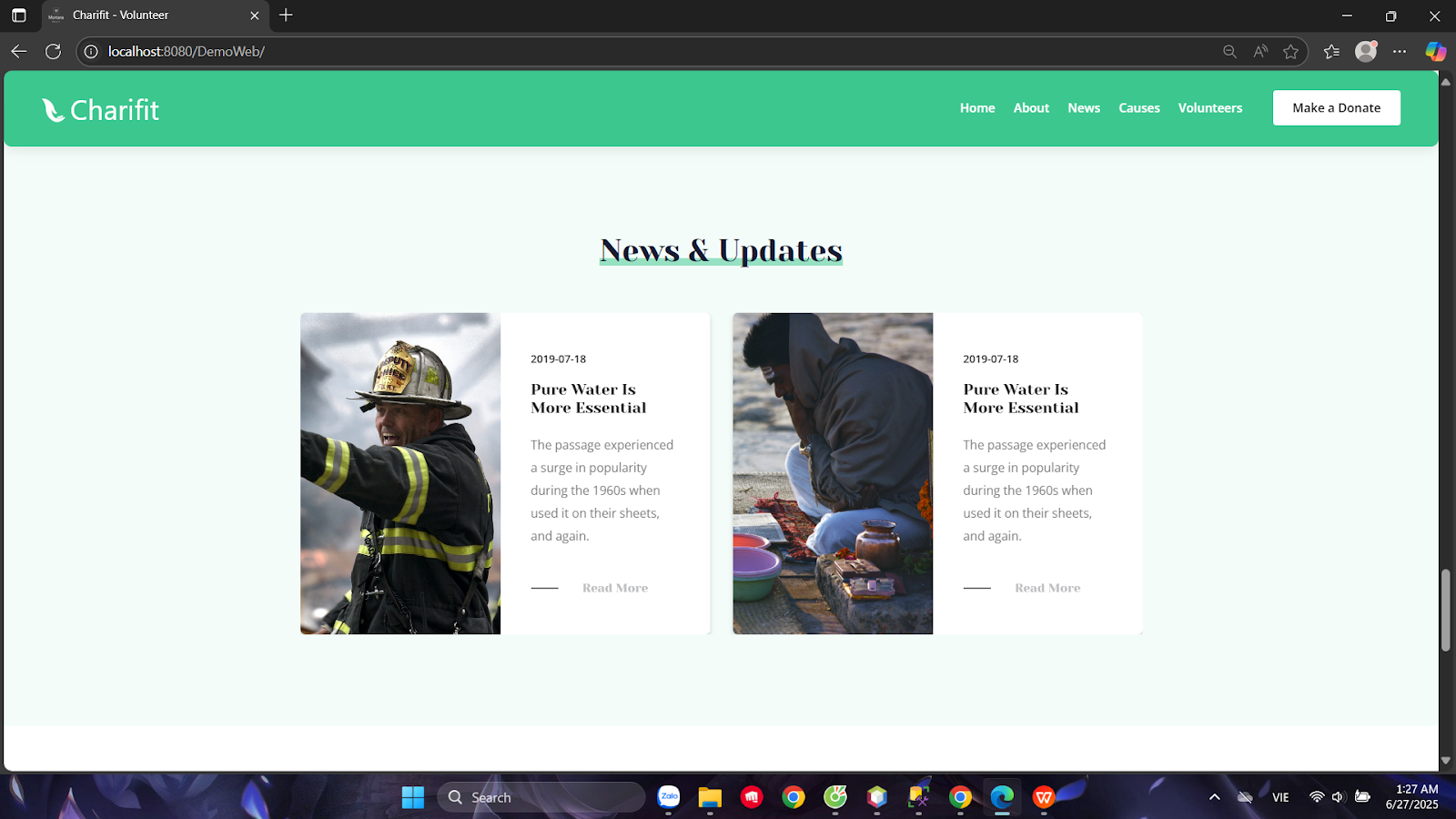
## **9. Authenticated user**

**

Picture 4.9. Authenticated user

## **10.The data is fetched from the database**





Picture 4.10. The data is fetched from the database.

# **CONCLUDE**

## **1. Functions and Description**

The thesis presents the process of researching and building a volunteer website for those who want to search for or participate in volunteer activities, or can become volunteers on the website itself. Through an in-depth analysis of functional requirements and non-functional criteria, we can see that an effective system is not simply a collection of features but also a delicate balance between performance, ease of use, security and scalability. A thoughtfully designed system will go beyond the role of a regular management tool. It becomes a powerful bridge, connecting volunteers with countless opportunities for meaningful contributions, regardless of geographical distance or time zone.

The website is built on the Java Spring MVC model, combined with web technologies such as HTML, CSS, JavaScript and Bootstrap, along with the MS SQL Server database management system. The following are details of the functions available in the system:

* **Login of Admin:** Provides a secure portal that allows administrators to access the system, ensuring only authorized users can perform management operations.
* **Administrator activities management function:** Allows administrators to monitor and adjust activities taking place in the system, thereby maintaining the stability and continuity of services.
* **Administrator users management function:** Support administrators to add, edit, delete or update user information, and manage their permissions in the system.
* **Administrator notifications management function:** Helps administrators create, edit, and send notifications to users, ensuring important information is communicated promptly and effectively.
* **Administrator ratings management function:** Provide tools for administrators to monitor and adjust ranking criteria, ensuring fairness and transparency in the system.
* **Administrator registrations management function:** Allows administrators to review and manage new registrations, approving or rejecting requests, ensuring a tightly controlled user flow.

These functions together create a comprehensive management platform, making it easy for administrators to operate, monitor and develop the system optimally.

## **2. Limit**

While they make for a fairly comprehensive governance system, there are still some potential limitations that need to be considered to improve the security, efficiency, and scalability of the system.

### **2.1. For Users (Volunteers)**

- The interface is still rudimentary and simple:

* The design is not user-friendly, the layout lacks aesthetics, and is not optimized for mobile devices.
* There are no interactive effects or modern user experiences.

- No profile page:

* Volunteers cannot view or manage personal information.
* Activity history or hours of participation are not displayed.

- Lack of notification and update system:

* When there is a new activity, schedule change or cancellation, users are not notified.
* There is no mechanism to send emails or notifications on the interface.

- No real evaluation function:

* There is no way to provide feedback to the organizers or administrators.

### **2.2. For administrators (Admin/Coordinator)**

- No notification function, management support:

* There is no tool to send notifications to users (eg: send emails, popups, internal news).
* No system to support quick response management or handling of situations.

- Unable to export data:

* Unable to export volunteer or activity lists to Excel or PDF files for synthesis and reporting.
* Causing difficulties in statistics and monitoring of activity effectiveness.

## **3. Development direction**

To enhance user experience and expand system capabilities, we plan to develop the following additional features in the future:

* **Email notification integration:** The system will be upgraded to automatically send confirmation emails to users after they successfully register or check-in. This will be done through JavaMail integration, helping users receive instant notifications, enhancing the reliability and professionalism of the service.
* **Generate and display QR codes for check-in:** To optimize the check-in process, we will develop a feature to generate and display confirmation codes in the form of QR codes. This QR code can be used for quick and convenient check-in at events or activities, minimizing waiting time and increasing management efficiency.
* **Display activity location on map:** For live activities, the system will integrate Google Maps API to display the exact location of the activity on the map. This feature will help users easily locate and find their way to the location, enhancing the experience of participating in the activity.

These improvements not only make the system more user-friendly and convenient, but also optimize the management process, contributing to the sustainable development of the platform in the future.

To ensure the long-term success and sustainability of the system, it is essential to maintain a continuous development roadmap based on real-world feedback from users. Regularly updating technology, upgrading features and improving user experience will help the system stay relevant and competitive. Furthermore, integrating new technologies such as artificial intelligence (AI) to suggest suitable volunteer opportunities, or blockchain to enhance transparency in recognition of contributions, can open up outstanding development potential.

In short, an Online Volunteer Coordination System is not just a simple technological solution; it is a powerful catalyst for community development. By modernizing and optimizing the volunteer coordination process, we not only improve the efficiency of volunteer activities but also contribute to awakening, nurturing and maximizing the power of kindness, creating positive and sustainable impacts for society.

# **REFERENCE**

<https://www.donorsearch.net/resources/ai-for-nonprofits/#:~:text=C%C3%A1%20nh%C3%A2n%20h%C3%B3a%20c%C3%A1c%20y%C3%AAu,ph%C3%B9%20h%E1%BB%A3p%20v%E1%BB%9Bi%20th%C6%B0%C6%A1ng%20hi%E1%BB%87u>.

<https://www.onlineschools.org/student-volunteer-guide/>

<https://veo.com.vn/bat-mi-loi-ich-cua-viec-tinh-nguyen-ma-ban-chua-biet>

<https://tuoitrethudo.vn/cong-nghe-va-mang-xa-hoi-thuc-day-phong-trao-tinh-nguyen-272074.html#:~:text=Ch%E1%BB%89%20c%E1%BA%A7n%20m%E1%BB%99t%20video%20ng%E1%BA%AFn,%C4%91%C6%B0%E1%BB%A3c%20%C4%91%E1%BA%B7t%20l%C3%AAn%20h%C3%A0ng%20%C4%91%E1%BA%A7u>.

<http://tuoitrekontum.org.vn/3-chuong-trinh/xay-dung-doan/giai-phap-ung-dung-cong-nghe-so-mang-xa-hoi-trong-thu-hut-quan-ly-dieu-phoi-nguon-luc-cho-cac-hoat-dong-tinh-nguyen-nam-.html>