SOFTWARE REQUIREMENTS SPECIFICATION

for

Network Hardware Resource Management System

Prepared by :

Team : Code_Till_Cope

- **Shreyansh Verma (2019113012)**
- **Aryan Dubey (2019113014)**
- **Aaditya Sharma (2019113009)**

Contents

1 Description

1.1 Project Description

The Network Hardware Resource Management System is a web-based application designed to efficiently manage hardware resources within a network environment. It allows users to add new hardware resources, remove existing ones, monitor resource usage, and manage Docker containers.
Servers/workstations have existing resources like CPU, GPU, Disk, Operating Systems and to utilize the full potential of them these need to be managed effectively.
The application will help us in keeping full control of all the hardware resources.

The project is divided into three phases.

1.1.1 Phase-1

In phase-1 we will create a web-interface that will list down all the hardware resources begin used in our network environment.
This will allow us to see what all resources are begin used in the network.

1.1.2 Phase-2

In phase-2 we will add the functionality that will enable the admin to monitor the usage of different resources. This will enable us to figure out which of the resources are begin over-used or under-used and we can check several aspects like.

1. Server up-time
2. Memory Usage
3. Disk Usage
4. User Info
5. View User/Group-Based Resources Usage Statistics

1.1.3 Phase-3

In phase-3 we will create an API that is responsible for allocation/de-allocation of hardware resources.
This API will enable the admin to effectively manage the hardware resources by allocating/de-allocating from a particular server/workstation to other.

2 Requirements

2.1 Functional Requirements

2.1.1 Use Case: List Hardware Resources

Description: This use case allows authorized users to view a list of existing hardware resources, including their specifications and usage status.

Steps:

1. User logs into the system.
2. User navigates to the "List Hardware Resources" section.
3. The system retrieves the list of hardware resources from the database.
4. The system displays a table or list of hardware resources, including details such as type, location, and usage status.
5. User can apply filters or search criteria to refine the list.
6. User can click on a resource to view detailed information.

2.1.2 Use Case: View Live Resource Usage

Description: Authorized Users can monitor real-time CPU, GPU, memory, and storage usage of individual server/workstation nodes and active Docker instances.

Steps:

1. User logs into the system.
2. User selects the resource to monitor (e.g., CPU usage).
3. The system displays live resource usage data.

2.1.3 Use Case: Add New Hardware Resources

Description: This use case enables authorized users to add new hardware resources to the network.

Steps:

1. User logs into the system.
2. User navigates to the "Add Hardware Resources" section.
3. User enters hardware details such as type, location, and specifications.
4. User submits the form.
5. The system adds the new hardware resource to the inventory.

2.1.4 Use Case: Remove Existing Hardware Resources

Description: This use case allows authorized users to remove existing hardware resources from the network.

Steps:

1. User logs into the system.
2. User navigates to the "Remove Hardware Resources" section.
3. User selects the hardware resource to be removed.
4. User confirms the removal.
5. The system updates the resource inventory and removes the hardware.

2.1.5 Use Case: View User/Group-Based Resources Usage Statistics

Description: This use case allows users to view resource usage statistics categorized by users and groups, helping identify resource-intensive activities and allocate resources more efficiently.

Steps:

1. User logs into the system.
2. User navigates to the "View Usage Statistics" section.
3. User selects a user or group for which they want to view resource usage statistics.
4. The system retrieves and compiles resource usage data for the selected user or group.
5. The system presents usage statistics in graphical or tabular form, including CPU, GPU, memory, and storage usage over a specified time period.
6. User can filter and customize the view of statistics (e.g., by date range, specific resource type).

2.1.6 Use Case: Manage Docker Containers

Description: This use case allows users to manage Docker containers, including viewing active instances, terminating overused containers, and handling hung containers.

Steps:

- **View Active Docker Containers**
  1. User logs into the system.
  2. User navigates to the "Manage Docker Containers" section.
  3. The system retrieves the list of active Docker container instances.
  4. The system displays a table or list of active Docker containers, including details such as container ID, image, status, and resource usage.
- **Terminate Overused Docker Containers**
  1. User selects a Docker container from the list.
  2. User identifies overused containers based on specified resource thresholds or time limits.
  3. User initiates the termination of the selected overused container.
  4. The system terminates the selected Docker container instance.
- **Handle Hung Docker Containers**
  1. User identifies hung Docker containers based on their unresponsiveness.
  2. User selects a hung Docker container from the list.
  3. User initiates the termination of the selected hung container.
  4. The system terminates the selected hung Docker container instance.

2.2 Non-Functional Requirements

2.2.1 Usability Requirements

- **Responsive Design:** The system must be responsive, supporting desktop and mobile devices.
- **Minimal Steps:** Users should be able to perform actions with a minimal number of steps.
- **Cross-Browser Compatibility:** The system should work seamlessly on major web browsers.

2.2.2 Performance Requirements

- **Response Time:** The system must maintain a minimal response time for live resource usage data.
- **Scalability:** The system should handle multiple users simultaneously without performance degradation.

2.2.3 Security Requirements

- **Authentication:** Users must authenticate to access sensitive data and perform actions.
- **Data Encryption:** Sensitive data, including user credentials, must be stored and transmitted securely using encryption.

2.3 Use Case Diagram

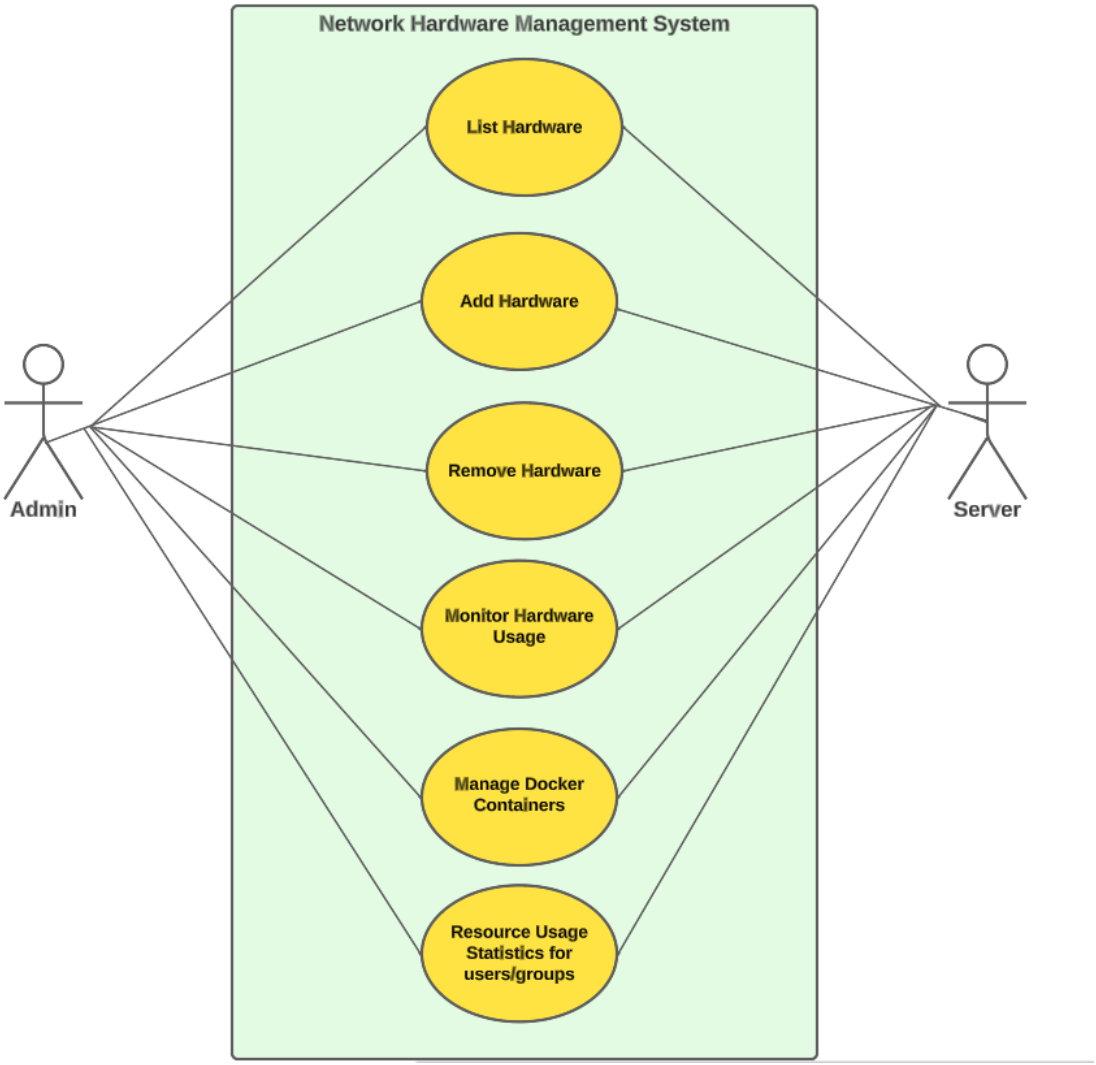

Figure 2.1: Use Case Diagram

3 Project Deliverables

3.1 List of Deliverables

List of items that will be delivered as part of the project:

- Fully functional Network Hardware Resource Management System web application.
- Use case and activity diagrams for all functional requirements.
- Documentation explaining system architecture, deployment, and maintenance.
- Test cases and their reports.