

**TEAM NAME:** Kindred

**DOMAIN:** Non-profit and NGO Operations

**PROBLEM STATEMENT:** Volunteer and Resource

Coordination

**TEAM MEMBERS :**

**1.Jaideep**

**2.Sajja Venkata Naveen**

**3.Sai Praneeth**

**4.Mythri**

**5.Yasaswini**

**1. Preface :**

- This document defines the requirements for the Volunteer and Resource Coordination System.
- Intended for developers, testers, admins, and end users.

- It explains what the system should do and how it should behave.

## 2. Introduction :

### 2.1 Need for the System:

NGOs face problems like:

Poor volunteer management

Lack of donation transparency

Difficulty in tracking resources

Manual and inefficient processes

### 2.2 Objective:

To develop a centralized platform that connects Volunteers, Donors, Beneficiaries, NGO administrators

## 3. Glossary

- **Volunteer:** Person providing service
- **Donor:** Person donating money/resources
- **Beneficiary:** Person receiving help
- **Admin:** System controller
- **Resource:** Funds or materials donated

## 4. User Requirements

The system should allow users to:

- Access the platform through a web browser
- Register and login securely
- View information based on their role
- Receive notifications and updates
- Use the system easily without technical knowledge

## **5. System Requirements:**

### **Functional requirements:**

Functional requirements describe what a system is supposed to do, including the services it provides and the actions it performs in response to user input.

#### **A. User Management:**

- Volunteers can **register, login, and update profiles**
- NGOs/Admins can **verify volunteers**
- Different roles: **Volunteer, NGO/Admin, Government Authority**

#### **B. Volunteer Management:**

- Register volunteer skills (medical, logistics, rescue, food supply)
- Assign volunteers to tasks based on **location & skills**
- Track volunteer availability (active/inactive)

### **C. Resource Management:**

- Add resources (food, medicine, shelters, vehicles)
- Track resource quantity and location
- Allocate resources to affected areas

### **D. Task & Activity Coordination:**

- Create tasks (rescue, distribution, medical help)
- Assign tasks to volunteers
- Update task status (pending, in-progress, completed)

### **E. Location & Mapping:**

- Identify disaster-affected locations
- Show nearby volunteers and resources
- Route planning for delivery

### **F. Communication & Alerts:**

- Send notifications to volunteers
- Emergency alerts to all users
- Messaging between volunteers and coordinators

### **G. Reporting & Monitoring:**

- Generate reports (volunteers deployed, resources used)

- View real-time dashboards
- Audit logs for actions

## **Non-Functional Requirements:**

Non- Functional requirements specify quality attributes and constraints of a system, such as performance, security and reliability.

### **A. Product Requirements:**

#### **1. Efficiency Requirements:**

- System should handle **large number of users simultaneously**
- Resource allocation should be fast

#### **2. Performance Requirements:**

- Page response time  $\leq$  **2 seconds**
- Notifications delivered in **real time**

#### **3. Space Requirements:**

- Efficient database storage for volunteer & resource data
- Optimize storage for images/documents

#### **4. Usability Requirements:**

- Simple UI for non-technical users

- Mobile-friendly interface
- Supports multiple languages

## **5. Dependability Requirements:**

- System should be available **24/7**
- Data backup and recovery
- Minimal downtime during disasters

## **6. Security Requirements:**

- Secure login and authentication
- Role-based access control
- Encrypt sensitive data (personal info, locations)

## **B. Organizational Requirements:**

### **1. Operational Requirements:**

- System should work on web and mobile
- Must run on cloud infrastructure

### **2. Environmental Requirements:**

- Operates under low bandwidth conditions
- Supports use during power/network instability

### **3. Development Requirements:**

- Developed using standard frameworks
- Easy to maintain and upgrade
- Modular design

### **C. External Requirements:**

#### **1. Regulatory Requirements:**

- Follow government disaster-management guidelines
- Data protection compliance

#### **2. Ethical Requirements:**

- No misuse of volunteer personal data
- Fair task assignment without bias.

#### **3. Legislative Requirements:**

- Comply with IT laws and data privacy acts
- Maintain legal audit logs

#### **4. Accounting Requirements:**

- Track donations and resource funding
- Generate financial usage reports

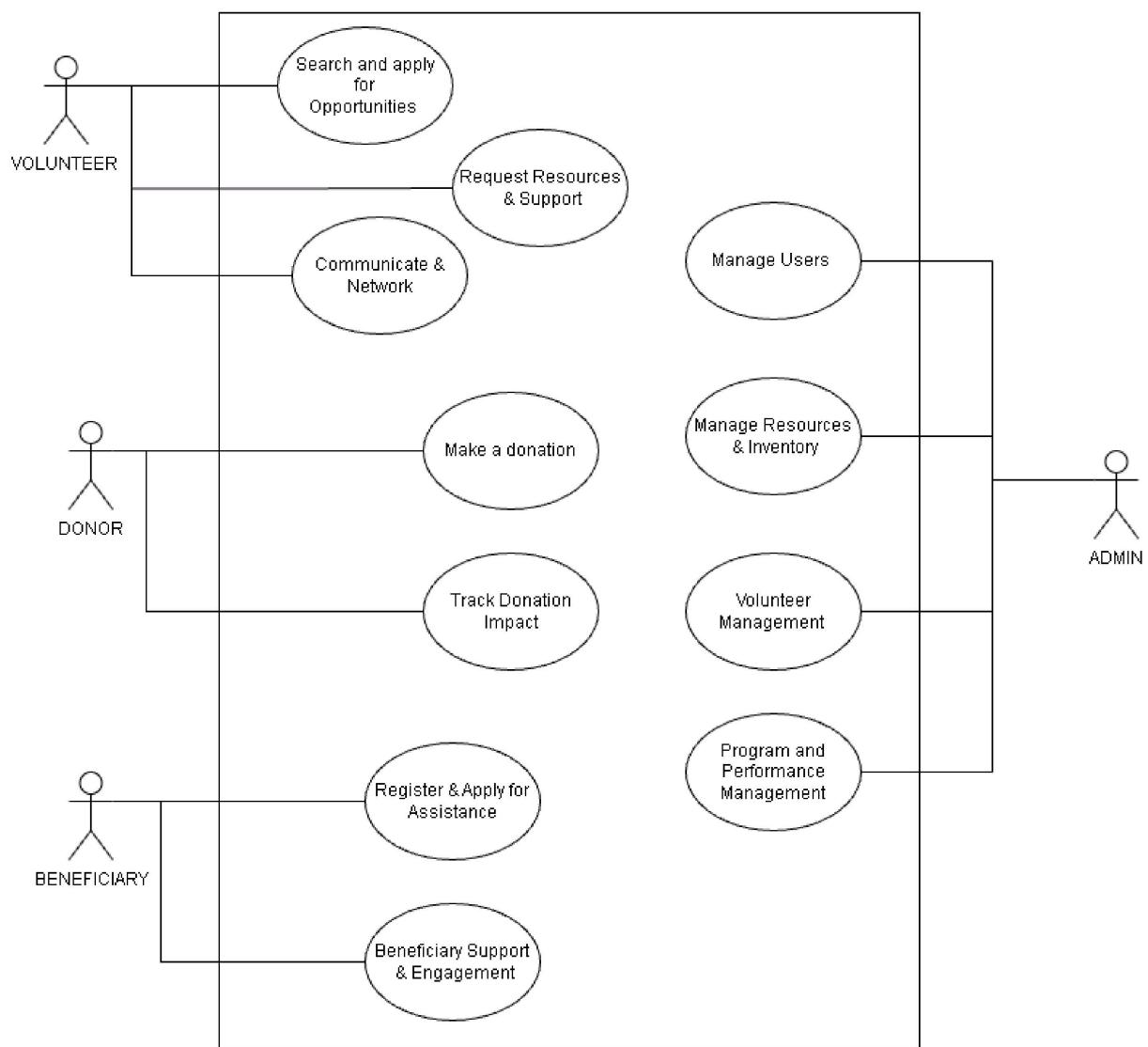
#### **5. Safety / Security Requirements:**

- Protect system from cyber attacks

- Prevent false emergency requests

## **6.System Models**

### **6.1 USE CASE:**



## 6.2 ACTIVITY DIAGRAMS:



Admin accesses Volunteer Management system

View Volunteer list and available programs

Verify Volunteer Credentials and conduct background checks

Credentials Verified?

NO

Reject Volunteer

YES

Approve Volunteer for Program

set travel radius and transportation preference

Match Volunteers to Opportunities based on skills

Suitable match found?

YES

Assign volunteers and allocate resources to programs

NO

Search for alternative volunteers

Send assignment notifications to volunteers

Volunteer accepts assignment

Track volunteer hours

Hours Logged Automatically

YES

System records check-in and check-out times

NO

Volunteer submits hours manually

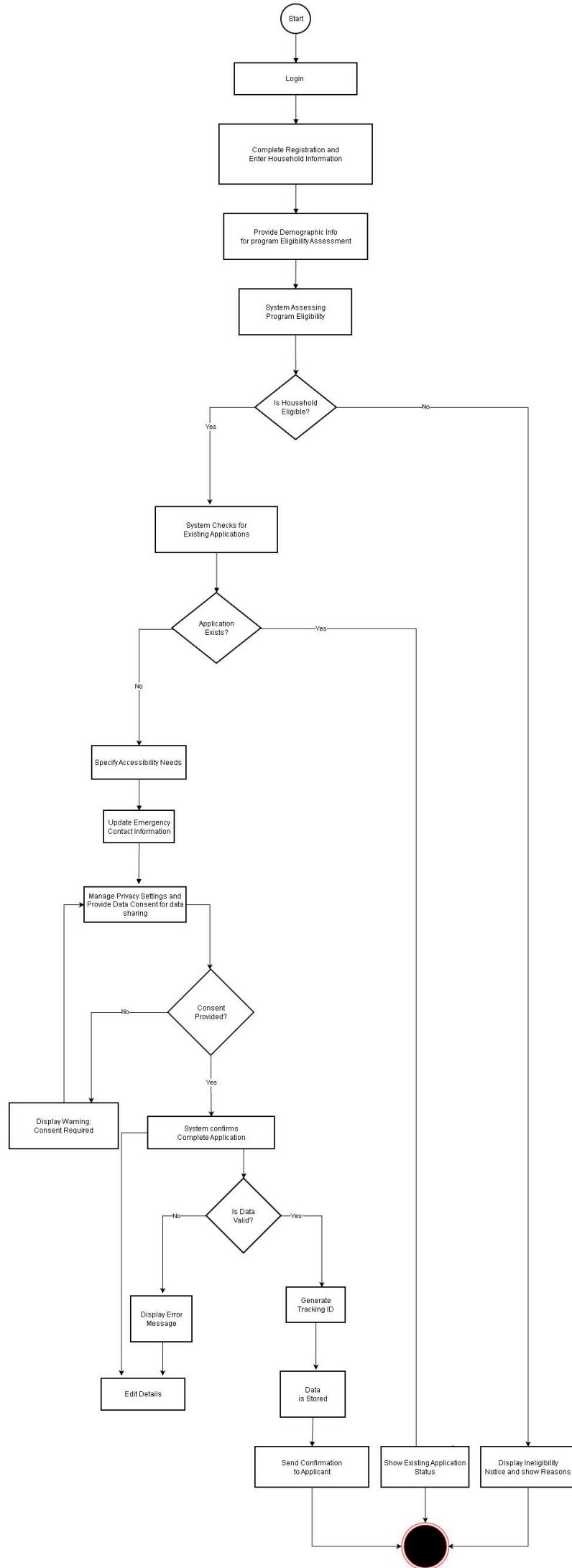
Admin reviews and approves hours

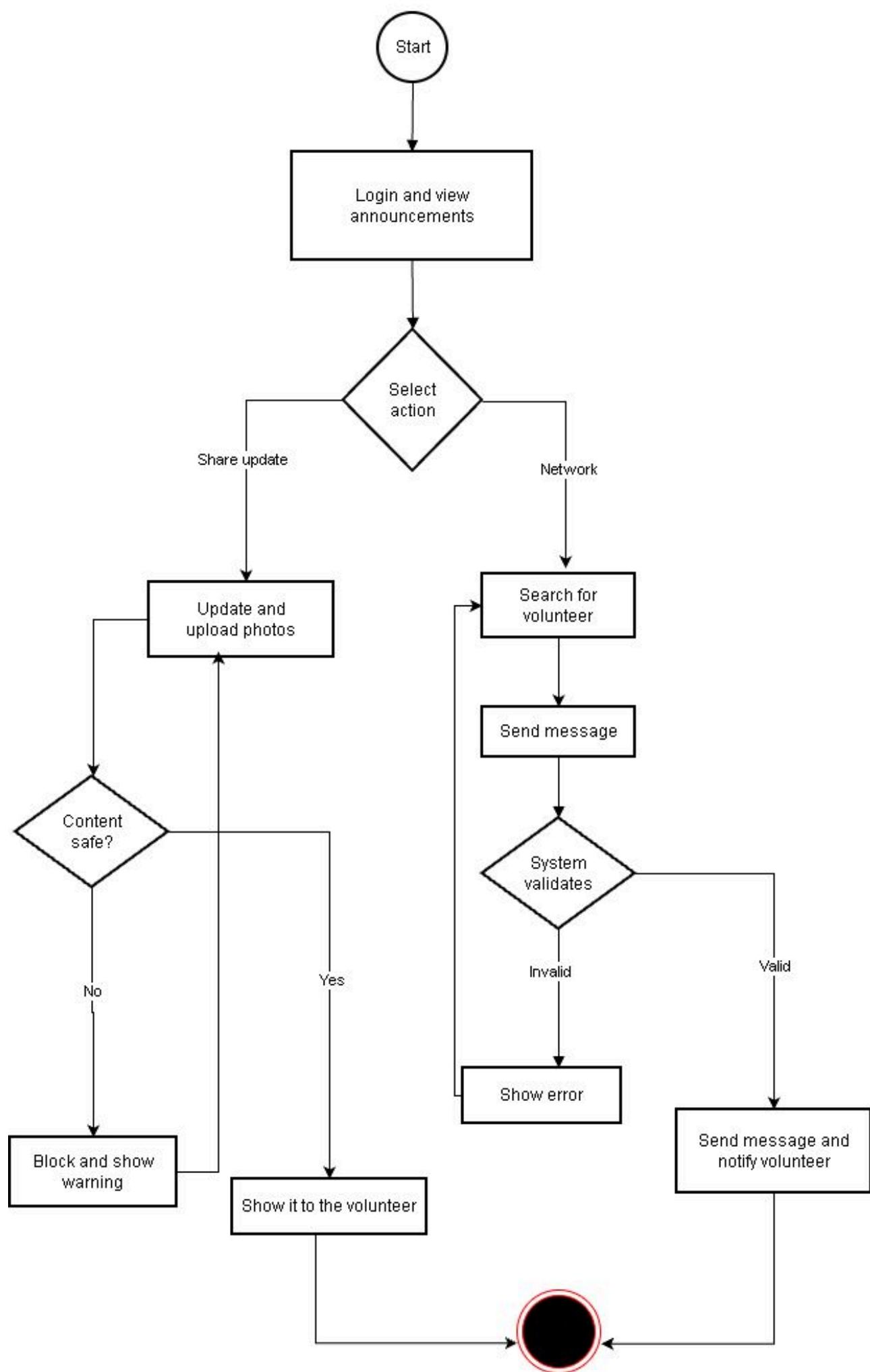
Update volunteer records

Generates activity reports

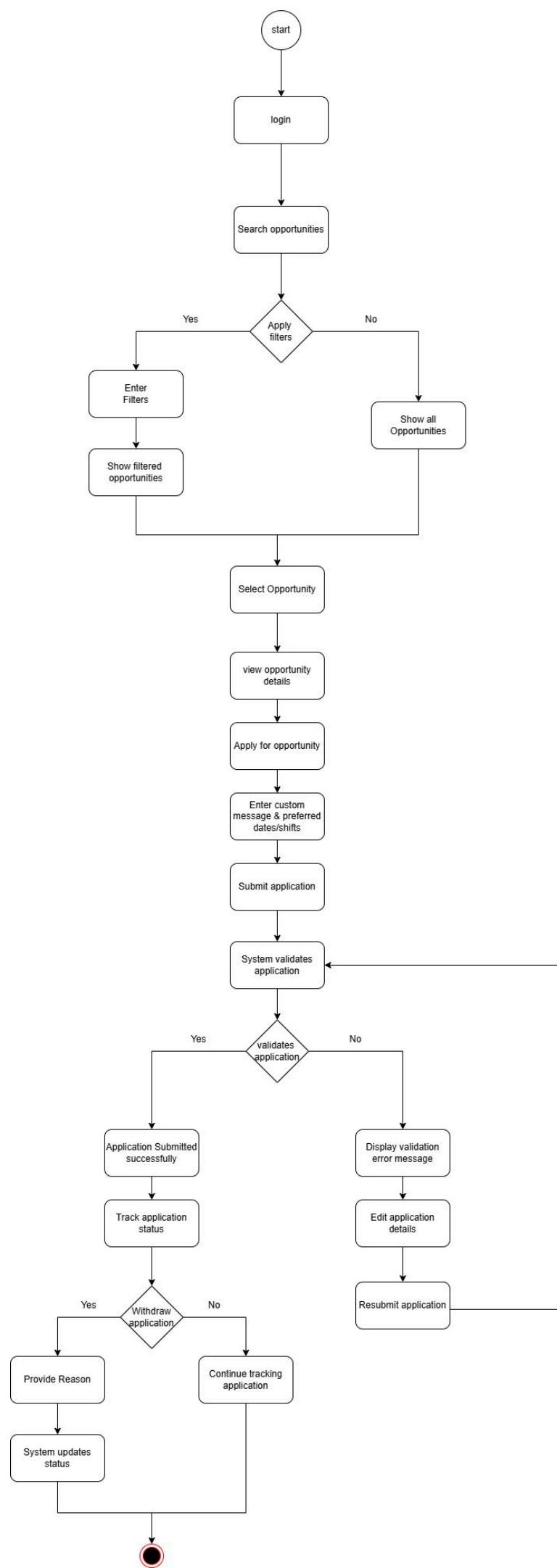


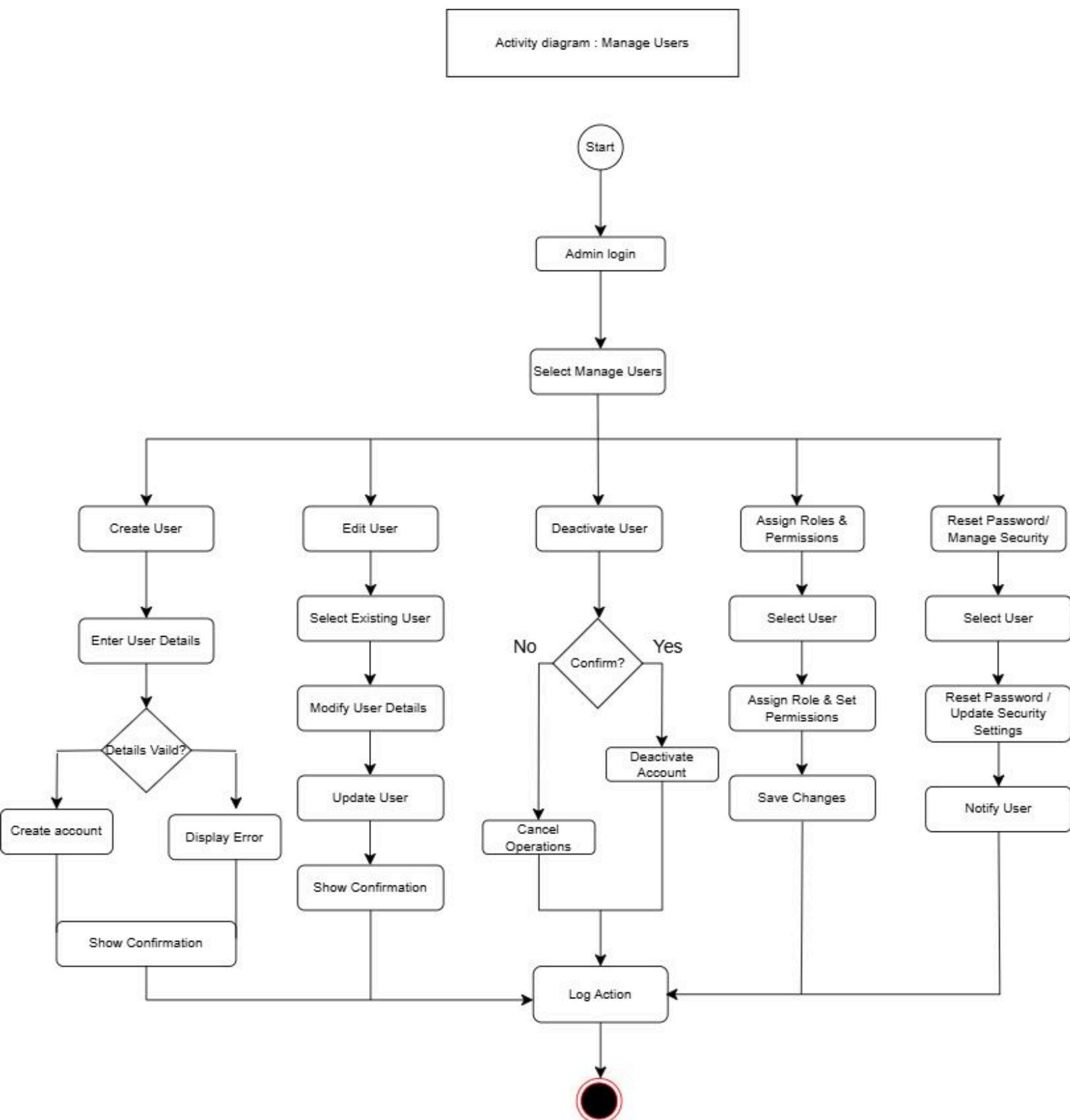
Register and apply for assistance





Activity Diagram: Search and apply for opportunities





# SEQUENCE DIAGRAMS:

