**Final Project Report: Communication Ecosystem for Multi-Party Applications**

# Executive Summary

This project presents the design and implementation of a digital platform that enables a collaborative ecosystem connecting NGOs, hospitals, educational institutions, logistics, and government agencies to provide comprehensive services to the needy.

# Introduction

Objective: The system is designed to facilitate collaboration among NGOs, hospitals, educational institutions, logistics companies, and government agencies to provide essential services such as education, shelter, clothing, food, and healthcare to the needy.

Problem Definition

The ecosystem addresses the complex challenge of delivering a range of essential services to underprivileged populations by leveraging the collective capabilities and resources of multiple entities.

# ****Object Model****

A diagram of a company

Description automatically generated

# Stakeholder Analysis

Identify stakeholders and define the value each entity brings to the ecosystem, establishing the need for an interconnected network of services.

# Use Cases

Develop use cases to capture the functional requirements of the ecosystem, detailing the processes that will deliver the intended services to end-users.

# System Design

**Core Components**: Describe the design of the system's core components, including the EcoSystem, Network, Enterprise Directory, and Organizations.

**Object Model**: Illustrate the object model that defines the entities and their relationships within the ecosystem.

# Core Components:

**Ecosystem**: This is the highest level of the system that encapsulates the entire business model. It likely contains logic to manage the overall operations and interactions among different enterprises and their networks.

**Network:** Represents a network of connections among various entities. This could be geographical or organizational networks that help in segregating and managing the ecosystem effectively.

**Enterprise Directory:** A collection of enterprises within the ecosystem. It manages different types of enterprises like NGO, Hospital, Logistics, and Government.

**Enterprise:** The enterprise entity is subdivided into several types, each representing a sector such as NGO, Hospital, Logistics, and Government. Each enterprise type has its specific roles and responsibilities within the system.

**Organization Directory:** Manages a collection of organizations under an enterprise. It helps in categorizing and structuring the internal divisions or departments.

**Organization:** Individual organizations like Delivery, Analytics, Doctor, Pharmacy, Education, Health, Inventory, Volunteer, and Admin organizations are managed within this entity. Each organization focuses on a specific aspect of service delivery.

**Employee Directory and User Account Directory:** These manage the details of employees and user accounts, respectively, which are crucial for the operation of the ecosystem, enabling user authentication and authorization.

**Work Queue and Work Request:** These components manage the workflow within the system. A work queue holds pending tasks, and work requests are specific tasks created by or assigned to users.

**Role**: Defines different roles within the system, such as System Admin, Education, Health, Inventory, Analyst, Doctor, Volunteer, Pharmacist, and Admin roles. Each role has specific permissions and responsibilities.

# Code Snippets

**Analysis:**

for(WorkRequest request:requestList){

HealthDoctorWorkRequest r = (HealthDoctorWorkRequest) request;

if(map.containsKey(r.getDoctorType())){

int oldCount = map.get(r.getDoctorType());

int newCount = oldCount+r.getTotalPatients();

map.put(r.getDoctorType(), newCount);

}

else{

map.put(r.getDoctorType(), r.getTotalPatients());

}

}

**Enterprise Type:**

public enum EnterpriseType {

Government("Government"),

Hospital("Hospital"),

Logistic("Logistic"),

NGO("NGO");

private String value;

private EnterpriseType(String value) {

this.value = value;

}

public String getValue() {

return value;

}

@Override

public String toString() {

return value;

}

}

# Code Interface Screenshots:

A group of people with text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a medical application

Description automatically generated

A screenshot of a patient analysis

Description automatically generated

# Implementation Techniques

Detail the programming techniques used in the Java development of the system, emphasizing the integration of the user interface with the backend logic.

# Application Functionality

**Swing Application:** Demonstrate a running Swing application that addresses the challenges at an ecosystem scale.

**Role-Based Authentication:** Provide details of the authentication module ensuring secure access to the system.

**Reporting Module:** Describe the implementation of the reporting module that provides a summarized view of system data.

**Configuration Module:** Explain the configuration module with test cases that validate the solution's correctness.

# Functionality and Workflow:

**NGO Enterprise:** Connects with other enterprise types to facilitate donations and services. It can create work requests for Books, Clothes, and other necessities to be handled by the Logistics Enterprise.

**Hospital Enterprise:** Provides health services to beneficiaries identified by the NGOs. It can handle health-related work requests and coordinate with the Health Organization for delivery of services.

**Education Organization:** Works under the NGO or independently to provide educational services. It can create and manage work requests related to educational needs.

**Logistics Enterprise:** Critical for the distribution of goods like books and clothes. It receives work requests from the NGO and coordinates delivery.

**Government Enterprise:** It may play a regulatory role or provide support and authorization for various activities. It can also be a direct provider of services like shelter or subsidies.

# Technical Implementation:

Java-based application likely using a graphical user interface (GUI) built with Swing or a similar framework.

The business logic is encapsulated within Java classes that correspond to the entities in the object model.

Data persistence might be handled by a database, indicated by the DB4O folder in the Business directory, which could be an object-oriented database.

# Testing and Validation

Explain the testing approach and present test cases as evidence of the solution's effectiveness and reliability.

# Conclusion

Reflect on the collaborative effort and the greater value achieved through the partnership of different enterprises within the ecosystem.

# Collaborators

**Rama Chandra Kashyap Mamidipalli(002812095)**

**Vidhi Patel(002641341)**

**Vaibhavee Gamit(002821886)**