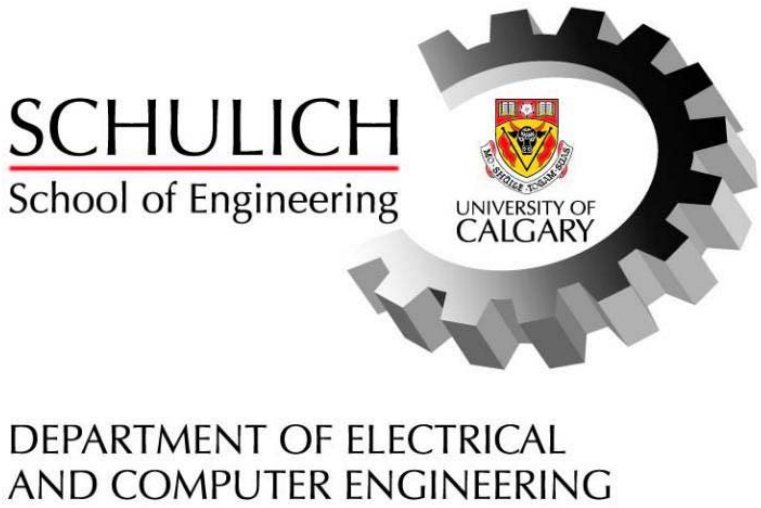
SENG 696 Agent-Based Software Engineering
Course Project



COVID-19 Alert MEDMAS (MEDical Multi-Agent System)

Assignment 1: MAS Design Document

Group I

Jiayu Zhuang / Xinghan Chen / Cheng Li / Xun Sun

1. Case Background

This project is to connect people with the health service and the lab. The idea of this system was motivated by the COVID-19 pandemic. The whole world has been suffering from this virus for 9 months. It is quite difficult to control the spread due to the infectiousness of the disease, and people are not aware of danger when they are close to a high-risk area since they don't have access to the latest information. Therefore, a system is to be set up to establish the interaction between residents, the government health service department, hospitals/clinics and labs so that the alert information could be sent to everyone timely. As a result, people could reduce exposure to risky neighbourhoods and lower the possibility of getting infected.

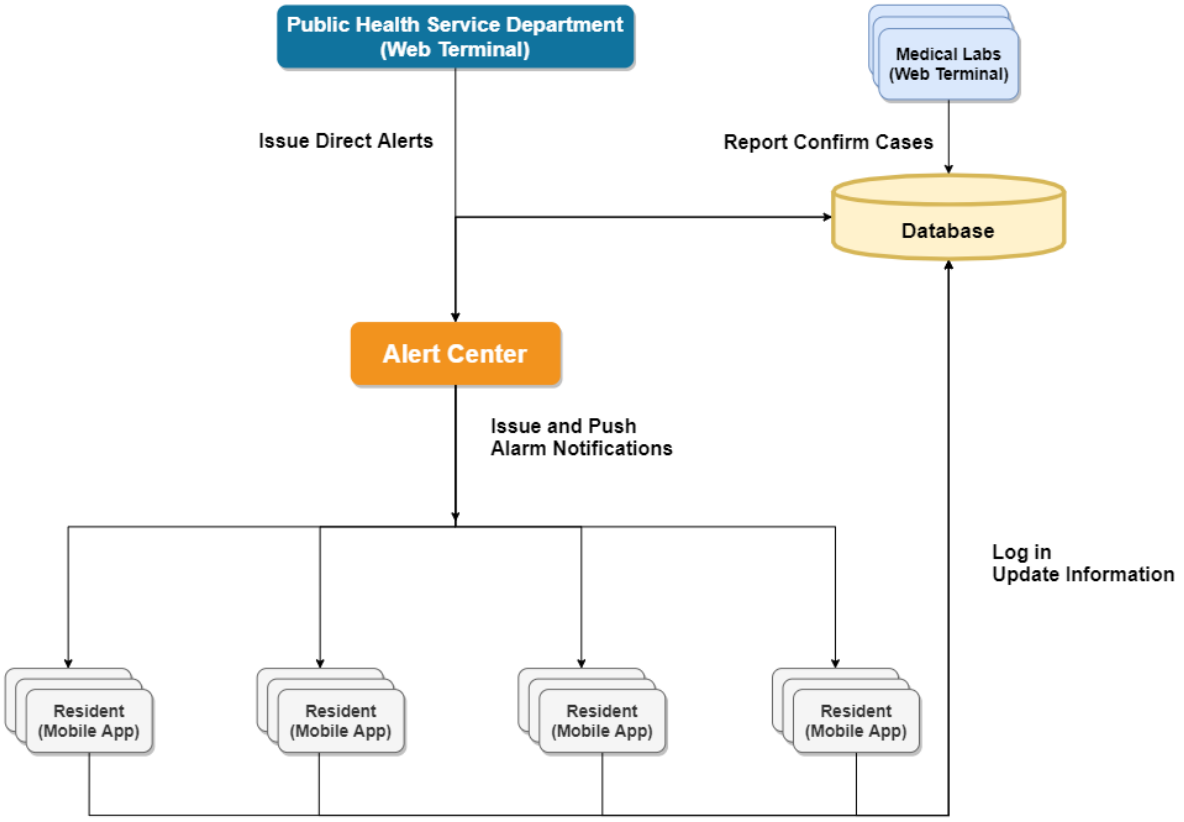
2. Description

The proposed MEDMAS (MEDical Multi-Agent System) is a multi-agent system designed to obtain health information from users and to issue alert to the public.

The alert contains the following information: location, level, time.

The MEDMAS collaborates with the public health department, medical labs and health care plan enroller to identify potential threats and update personal health records.

Ideal system model:



3. Assumptions

The system maintains a user profile where the ID and health information of an individual is kept, a profile contains the following information:

- user name
- user ID
- user address
- user location
- user information last updated time
- user health condition

The user connects to the internet and logins into the system. The user updates personal health information through the user interface. The user location, last updated time is obtained by the device or user input from the user interface, and the user name, ID, address is provided by the user profile automatically.

The user information sent to the system must contain all the information.

The PHS user connects to the system, gets user health information and issues alerts through its interface, the alerts must have all the information and follow a certain format, the alert agent should automatically detect new alerts and publish them.

The lab user connects to the system, retrieves user information from the dataset and updates the health condition and the last updated time.

4. Basic Requirements

- Resident users should log in directly with their names and health card numbers. No registration is required.
- Resident users can update location and health condition with the mobile app. Location information of resident users updates every 5 minutes automatically.
- Physician users from labs can update confirmed cases.
- Officer users from the health service department can issue direct alert to all users.
- Alert center scans the main database and automatically issue alert to:
    - All residents who live in the community where confirmed cases currently live.
    - All residents who stay in the area within 100 meters from confirmed cases.

5. Wish List

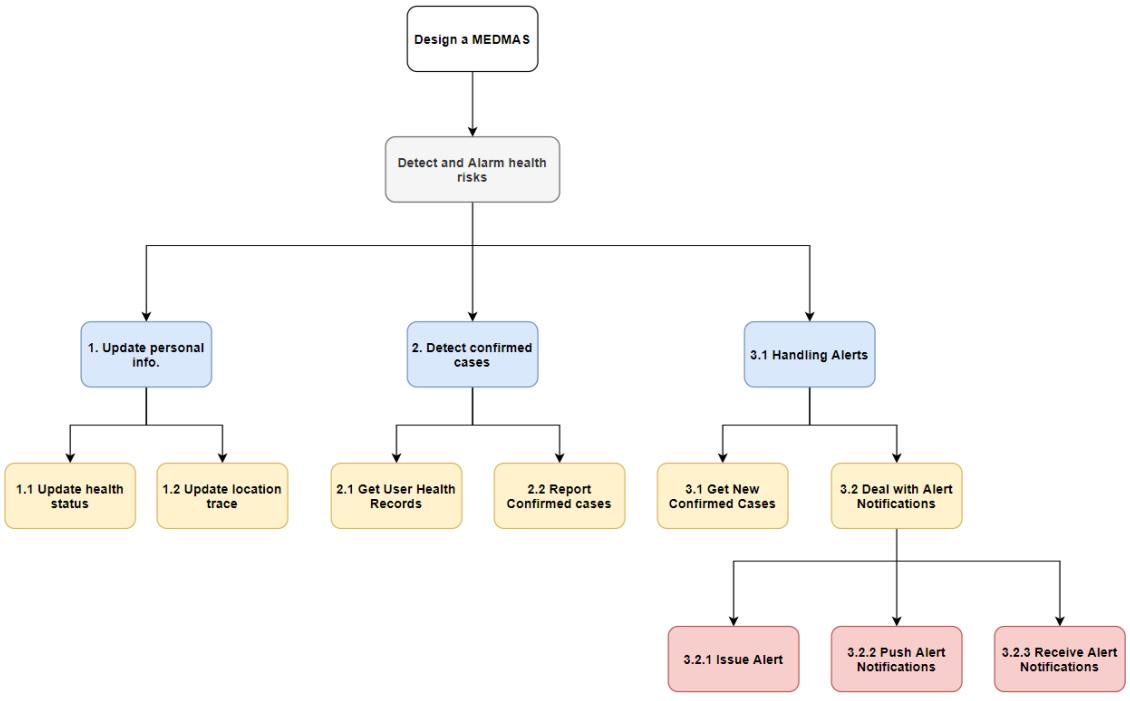
- Expand the database so that the alert could be applicable to other contagious diseases.
- Officer/Doctor/Physician users (for the health service department, hospitals, clinics, labs) should log in with usernames and passwords. Registration is required with validation.
- Resident users can update personal symptom status with the mobile app.
- Doctor/Physician users from hospital /clinics can update residents' personal symptom status.
- Alert is also issued to all residents who live in the community where more than 10 people report symptoms.

6. Goals Capturing

List of Derived Goals:

- User update health information
- User update location
- User receive alert
- Officer user issue alert
- Officer user get new confirmed cases
- Lab user get user information
- Lab user update confirmed cases
- Alert center push alert notifications

Goal Hierarchy:



Goal Mapping

- User agent: 1/ 1.1/ 1.2/ 3/ 3.2.3
- Alert agents: 3/ 3.2/ 3.2.2
- Lab agents: 2/ 2.1/ 2.2
- PHS agents: 3/ 3.1/ 3.2/ 3.2.1/

7. Role Identification

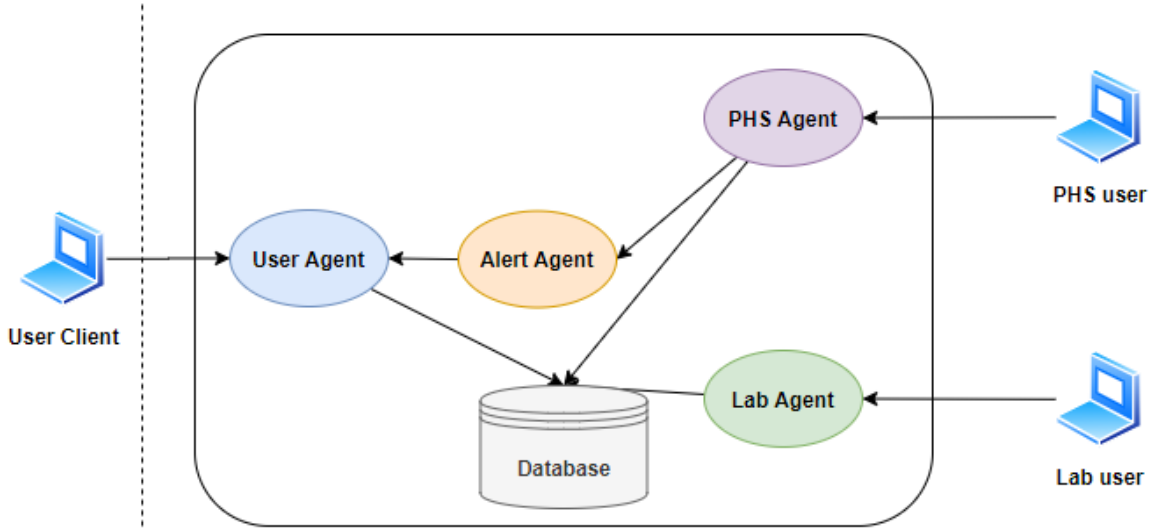
Required roles:

- Personal assistant role
- Alert handling role
- Health Services Handling role: Public Health Department and Medical Lab.

Assign roles to agents:

- Resident user agent
- Alert agent
- PHS agent and Lab agent

System Architecture:



8. Agent Description

User Agent

The User Agent intercepts the user information sent by the user through the User Interface, and then sends it to the database to update the user information.

The User Agent, in our architecture, receives an alert from the Alert agent and displays it to the user interface.

Alert Agent

The alert agent, as its name suggests, handles the alert processing part of the system. It takes the alert from the PHS agent and then distributes the alert to user agents. The alert distribution is based on the location information given in the alert. The alert agent will send the alert to the user agent that has the same location as the location information in the alert. The user agent then presents the alert to the user.

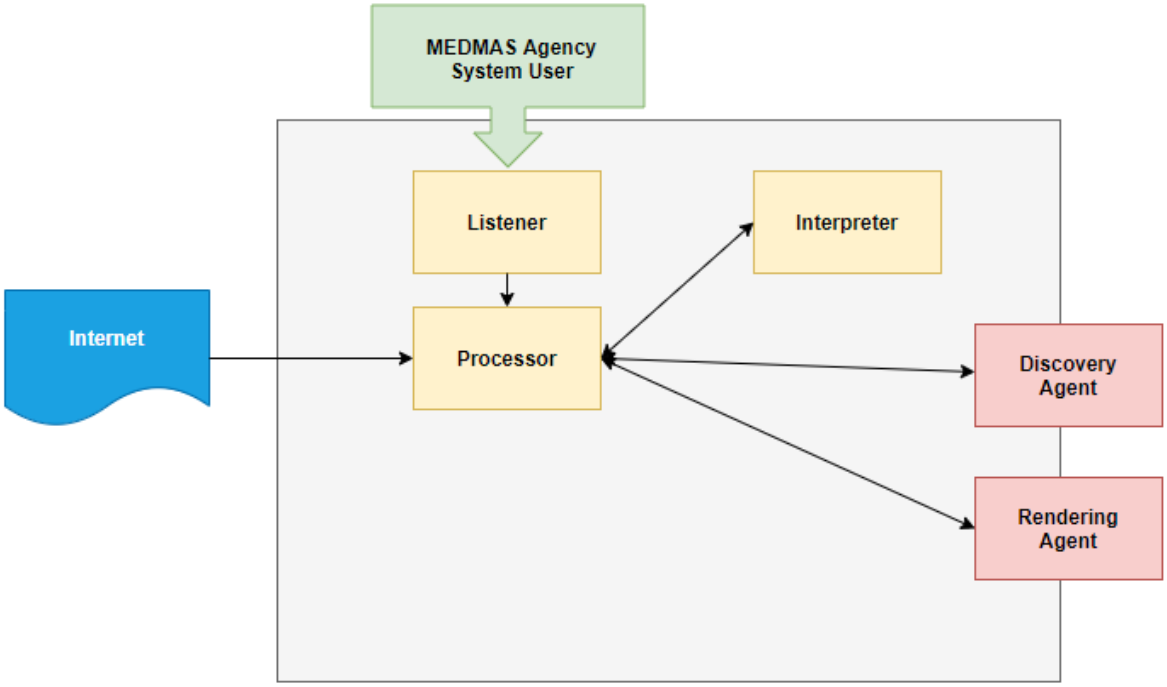
PHS Agent

The PHS agent in our system takes the responsibility of handling the communication between the PHS user and the system. It requests the information of recently updated confirmed cases from the database, and displays it to the PHS user. It also intercepts the information of alerts given by the PHS user, wraps them, and sends them as a package of alerts to the Alert agent. The alert agent then handles the publication of the alerts.

Lab Agent

The Lab agent is responsible for the communication between the Medical Labs and the system. It requests the information of recently updated user profiles from the database, and displays it to the Medical Lab. It also acquires the change made to the requested user profiles, and updates the user records in the database.

9. Agent Internal Architecture



Listener

The Listener component listens to a port for any incoming requests from the MEDMAS application.

Interpreter

The Interpreter parses and interprets the messages. We assume that all agents have agreed on a Document Type Definition (DTD).

Processor

The Processor receives the input. It uses the Interpreter to parse the document, and calls the appropriate function to run a process.

DiscoveryAgent

The Discovery Agent provides the service discovery base-service (a superset of UDDI). May be implemented as an external service.

RenderingAgent

The Rendering Agent is optional; it can be used by the Processor to render data before sending it back to the calling function. May be implemented as an external service