

Lebanese American University
School of Arts and Sciences
Department of Computer Science and Mathematics
CSC 430 – Computer Networks

Network Programming Project
Developing a *COVID*@UNI Application

Project Guidelines

- Students are required to work in groups of **4 students**.
- Each group is required to work on the project on their own. Any copying will be counted as cheating and treated accordingly.
- Group members are expected to put **balanced effort** on the development of the different requirements of the project.
- Project deliverables are to be submitted using BlackBoard in a single .zip file.
- Good luck!

General Description

In this project, you are required to design and implement using JAVA a COVID prevention application to keep university campus safe. The project is composed of two parts. In the first part, you are expected to develop the basic features of a COVID prevention application using a client/server paradigm. For the purpose of your application, you decide whether TCP or UDP is to be used as the underlying transport layer technology. Users create accounts and log in a centralized server that has to keep track of the health status of all students and instructors and, moreover, provide instructions to stop the spread of COVID. You have to define the protocol to be used between the client and the server and implement it. In the second part, the application has to be extended to support a hybrid architecture that allows peer-to-peer communication where clients can update each other about their health status. As an optional feature and for additional credit, you are asked to **design your system to support privacy**. This extension will be assessed based on its novelty and completeness.

Project Implementation

I. Client-Server \overline{COVID} @UNI Application (90 pts)

Client - Design and implement a COVID prevention client that has the following features and functionalities:

- The client has a graphical user interface (GUI).
- The user opens the client application and connects to the server application using the server domain name and port number.
- The user is given the option to register or to login.
- If the user is not registered, they can sign up and provide the server with name, photo, email address, username, password, and vaccination status. If the user is vaccinated, they are asked to upload the vaccination certificate to complete their portfolio.
- The user sends regular updates to the server including their location.
- In the case of a registered user, the user logs in with their existing account where the server performs authentication by verifying the username and password.
- The user can query for the total number of active cases and other interesting statistics that you may choose to include.
- The user has the option to identify a group of trusted people to whom they wish to disclose their health status.
- The user can check the status of any other user who indicated them as *trusted*.
- The user shares the positive PCR result, if any, with the server.
- The user has three status conditions, *contagious* if they tested positive, *at risk* if they were exposed to the virus, and *safe* otherwise.
- The user gets update from the server if they got close to a user with *contagious* status or with *at risk* status.

Server - Design and implement a COVID prevention server that has the following features and functionalities:

- The server application takes as a command line argument the port number on which it would be listening.
- The server allows the users to register and login.
- Every time a new user signs up for a new account, the server adds a new user.
- Every time a user logs in, the server authenticates them by verifying their username and password.
- The server application manages a database that stores user accounts and details.
- The server has to keep track of the IP address, port number, and location of each user (if the user is online).
- The server changes the user status to *contagious* if they tested positive, *at risk* if they were exposed to the virus, and *safe* otherwise.
- If infected, the server keeps track of the number of days the user spent in quarantine, and once completed, it changes the user status to *safe*.
- Depending on the user status, the server provides instructions related to quarantining, doing a PCR test, taking precautionary measures, and others.
- The server allows the user to retrieve some virus statistics on campus.

- The server sends notifications and instructions to all users who got close to someone tested positive or at risk.
- A user is set *at risk* when they get close to an infected person for a fixed duration. This duration can be set to 3 minutes for testing purposes.
- The server serves multiple clients simultaneously.

II. Extending *COVID@UNI* Application with Peer-to-Peer Functionality (10 pts)

In this part, you are asked to allow the system function in a hybrid client/server and peer-to-peer architecture. For peer-to-peer, users, once their status has been changed to either *contagious* or *at risk*, send notifications to all other users who had been within their vicinity to alert them. In this part, privacy is even more critical than in the previous part where all communications were taking place through the server.

Deliverables

- Source code of the project with documentation.
- A 6-page report and an appendix that includes snapshots of the application. The report should include the following information:
 - List of the group members: name, ID, email address.
 - The report is divided into two main parts, each detailing one of the project phases: Client-Server and Peer-to-Peer.
 - Description of the protocol implemented between the communicating entities.
 - Description of the functionalities of your application. Please note the implemented additional features explicitly.
 - snapshots of the application to demonstrate the main functionalities.
 - A table showing a breakdown of the project into multiple tasks. Please indicate next to each task the name of the group member who was mainly responsible for its implementation.
- Submit the code and report using BlackBoard in a single .zip file. The name of the zip file should be the family names of the group members with underscore separation, i.e., “FamilyNameI_FamilyNameII_FamilyNameIII_FamilyNameIV.zip”.
- You are expected to perform a demo to present the features supported by your application. You should be ready to answer any question related to the implementation of the project.