# Family Locator and Safety Network

Mihaela Hudisteanu

Alexandru Ioan Cuza University of Iasi

## 1 Introduction

The objective of this project is to create an application that integrates real-time location tracking and safety alert features, bettering how families stay connected. In order to ensure ease of use, the application will have an intuitive and user-friendly graphical interface. In this UI, the user will be able to follow the location of a family member, while being able to enter specific commands. Family Locator and Safety Network, while aspiring to be a creative and personal approach, it draws inspiration from more popular applications such as "Find My Phone".

## 2 Applied Technologies

To achieve real-time location tracking and seamless communication between users, the app uses network communication technologies like TCP and sockets, combined with multithreading. For the user interface the SFML will be used, and for data managing SQLite.

### 2.1 TCP

The app uses a concurrent TCP server. TCP establishes a connection before data exchange, ensuring a reliable and ordered flow of information[1]. In the context of this project, such a connection is necessary because it guarantees that each user receives the exact location of others and can send alarms.

### 2.2 Sockets

For the communication between server and clients, the app uses stream sockets. Through the sockets different types of messages are sent and received, according to a defined communication protocol.

### 2.3 Threads

One user needs at the same time to send its location, send alarms and receive such information from the other users, so it is necessary to use multithreading in the client program. Because different types of data are send by different threads through the socket, to ensure data integrity the program uses mutex[1].

The server creates one thread for each new client. Some resources, such as the list of active clients, are shared between these threads. To prevent multiple threads from simultaneously modifying shared data, mutexes are used[1].

# 3 Application Structure

In the following section a detailed diagram of the application is presented. It showcases the communication protocol between the server and clients, and the order in which the messages are sent and received. Some of this messages are: location updates from one client to the server, location updates of one client from the server to the other clients, alarms, and other notifications.

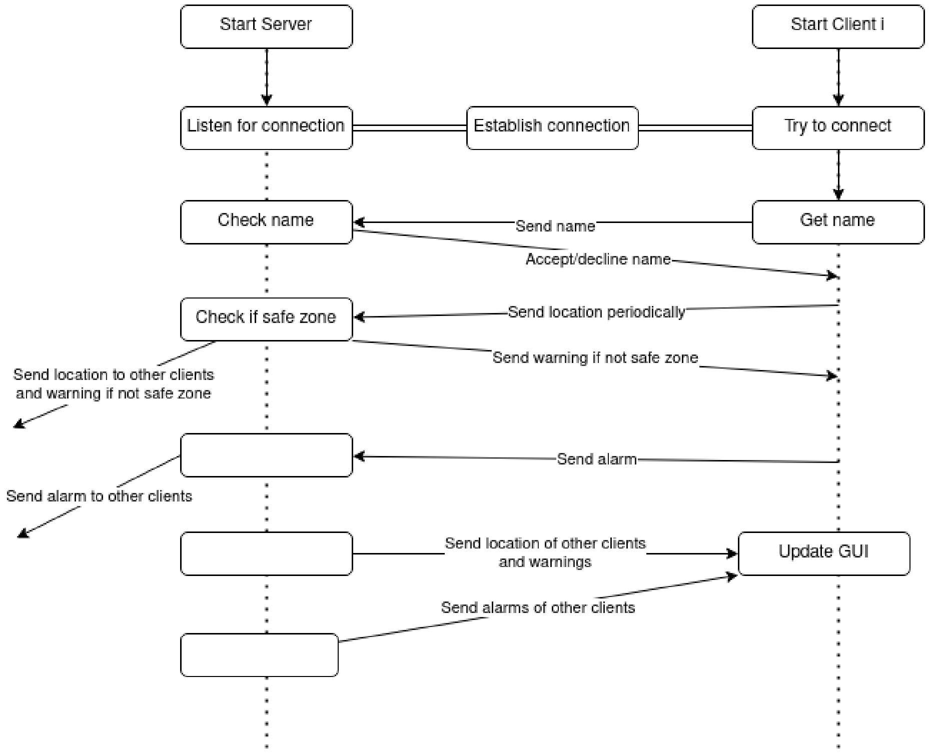
Start Server
Start Client i
Listen for connection
Establish connection
Try to connect
Check name
Send name
Get name
Accept/decline name
Check if safe zone
Send location periodically
Send warning if not safe zone
Send location to other clients and warning if not safe zone
Send alarm
Send alarm to other clients
Send location of other clients and warnings
Update GUI
Send alarms of other clients

**Fig. 1.** Diagram of the application

The server will also interact with a database, where the safe predefined locations will be stored. To do this, the program will use SQLite[2].

# 4 Implementation Aspects

## 4.1 Application-level protocol

The graphical user interface will look like a 2D plane, modeled as an imaginary grid, where each user is represented by a point. Safe zones will be marked by green circles on the plane. It will also have an alarm button. When a user interacts with the application, at first they need to enter a name and if it is accepted, they start seeing on the map the other connected users. The user can press the alarm button and they also receive notifications when the other users press their alarm button or are in unsafe zones.

### 4.2 Real life usage

It is clear that the usage of this application is for families to monitor each other and in case of emergency to be notified quickly in order to be able to help their loved ones.

### 4.3 Code details

Regarding the implementation, one innovative aspect is the random algorithm used to simulate user data. The next location of a user will be chosen at random in their proximity.

The safe zones will be represented as circles, so naturally, to store this data, we will use a class, which contains the coordinates of the centre, as well as the radius of the circle. To check if the user is in a safe zone, the circle equation will be used.

To represent the clients, we will use a class, which will store data about them and the corresponding socket. For cleaner code we will use STL vectors.

## 5 Conclusions

In conclusion, the application provides basic functionalities for monitoring the members of a family and their safety. Future improvements that can be made: defining different types of zones, other than safe ones, such as restaurant or school, and enabling a last seen system.

## References

1. Computer Networks Homepage, http://www.info.uaic.ro/ computernetworks
2. The SQLite Library, https://www.sqlite.org/index.html
3. Beej's Guide to Network Programming,https://beej.us/guide/bgnet/html//index.html