Written by: Rade Cvetkoski

PBT205—Project-based Learning Studio: Technology

Group 2:

Assessment 1 – Prototype development

Report on contact tracing app

1. Demonstrate Knowledge and Understanding of Functional Middleware

To set up the contact tracing app we used RabbitMQ as middleware for message brokering. RabbitMQ is an open-source messaging platform that facilitates connection between systems, ensuring that the users can send and receive messages and not lose them even if the computer crashes.

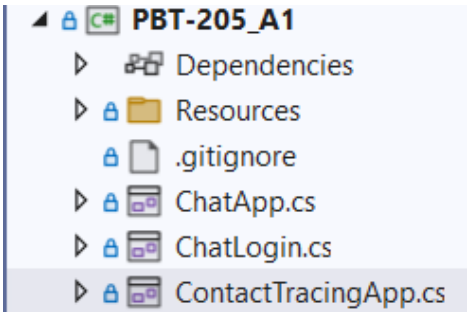
In our prototype we utilize RabbitMQ to set up connection between clients and send queries that later returns the number of times they have contacted each other.

Key roles of RabbitMQ in the app:

- Set up connection: Establish a connection between users to a RabbitMQ server.
- Send queries: Users can send queries among each other.
- Return outcome: Return the outcome of the query (Ex. You have contacted user X times)

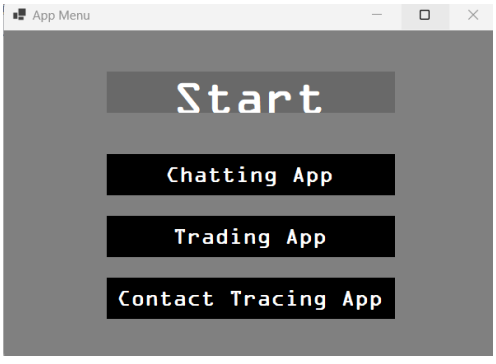
2. Compiled Code

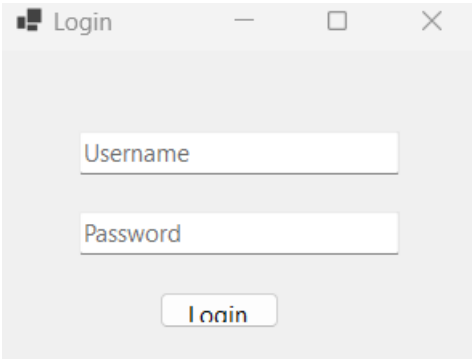
The contact tracing app is compiled in C#, using windows forms for user interface. The main namespace is PBT_205_A1 that inherits in class ContactTracingApp.



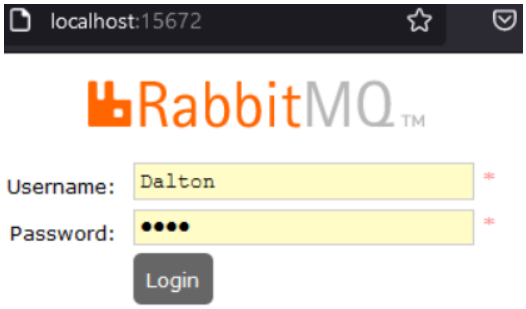
Key features of the compiled code:

- User interface: The app features main menu, where there are text boxes, buttons and clickable buttons with whom the user can interact.

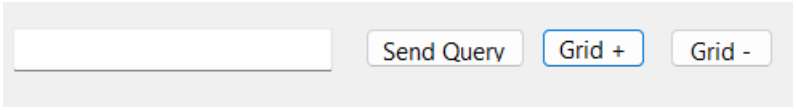




- Utilization of RabbitMQ: RabbitMQ is used to establish the necessary communication for sending and receiving messages.



- Messages: Users can log in using their credentials (Check bullet point above – User interface), send queries to other parties as well as increase the grid up to 30.



3. Knowledge and Understanding of Main Topics, Arguments Implemented, and Outputs

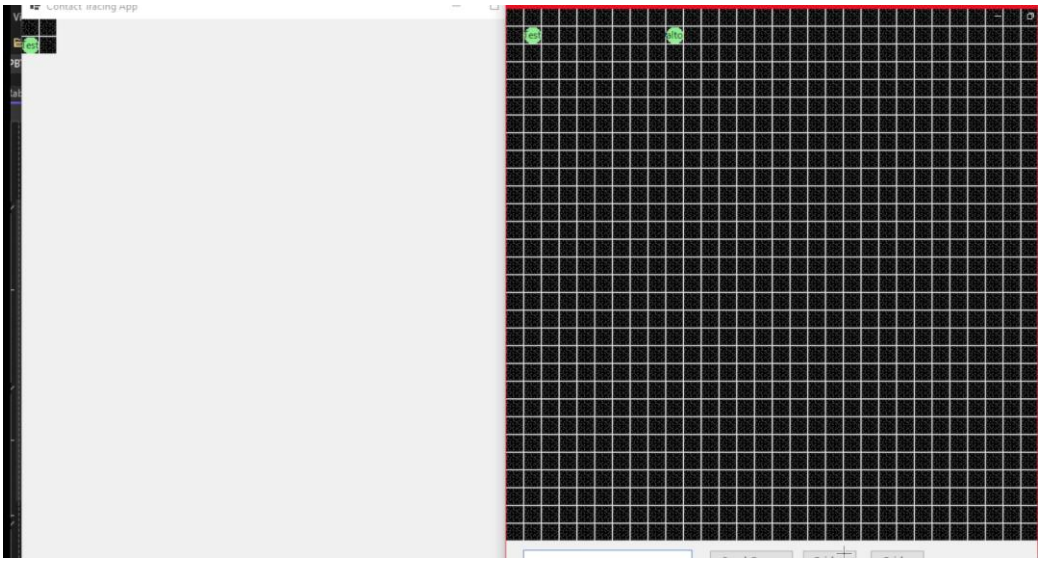
Main topics:

- Utilization of an open source communication software: In our project we went with RabbitMQ.
- User interface: The projects takes user interface in consideration, both in the design and functionalities of the app, starting from the forms, interactive buttons.
- Connection: RabbitMQ is configured with username and password.
- Encryption: Messages are encoded during the process of transmission and decoded when received by the other party.
- Method to move to random square and post the change to RabbitMQ

Outputs:

- User can see how many times they have contacted the other party.

- User can see the movement of squares on the grid at all times.

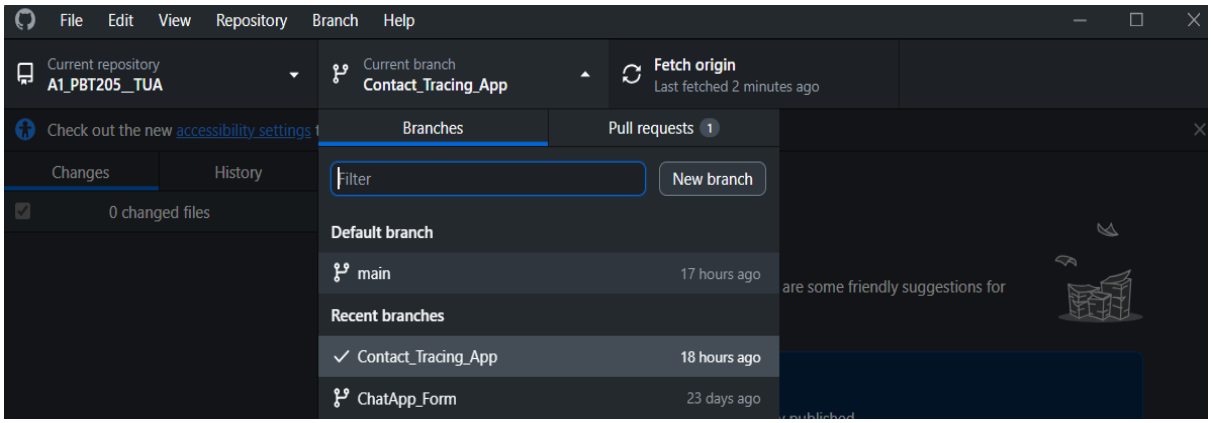


Written by: Rade Cvetkoski

4. Project organization and collaboration

After our first meeting in week 2, we chose **C#** as the main programming language, followed by **GitHub** as the platform where we will share the files and progress, and **Discord** to communicate and organize meetings and **RabbitMQ** as the open source communication platform as previously mentioned.

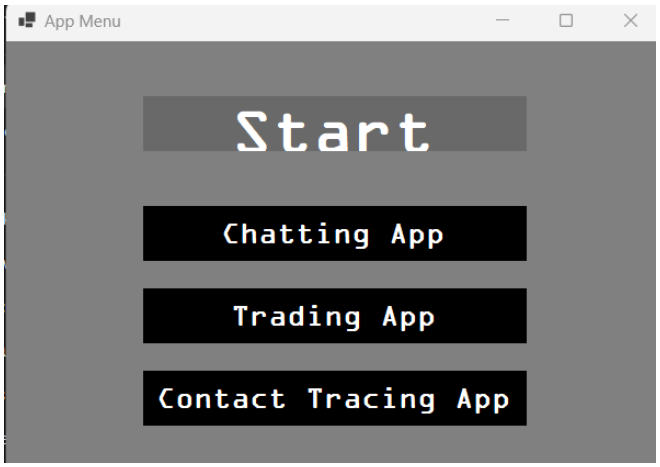
The contact tracing app was developed by Dalton Christopher, and all the files were stored on a GitHub repository where all members of the group had easy access to every prototype.



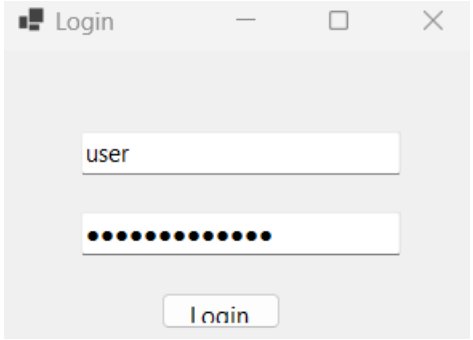
The original app was run on Windows PowerShell, and had few bugs, after some discussion the files were moved to Windows forms application for enhanced user experience.

5. Windows forms | Product display

Main menu:

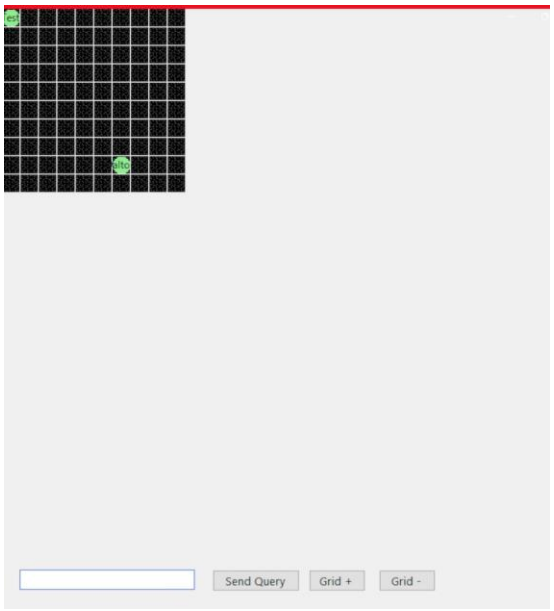


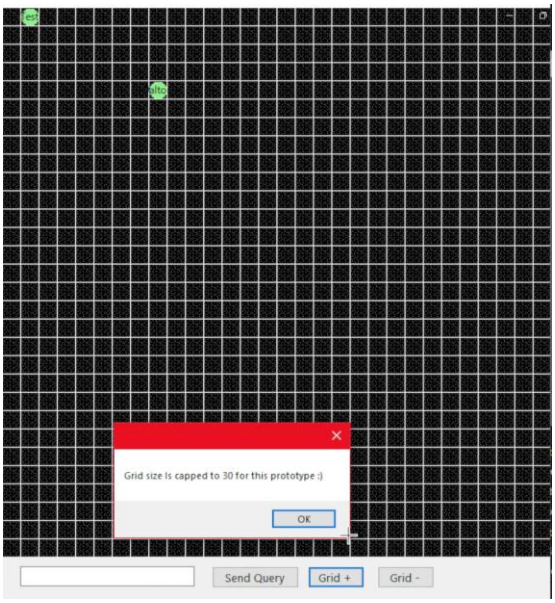
Log in section:



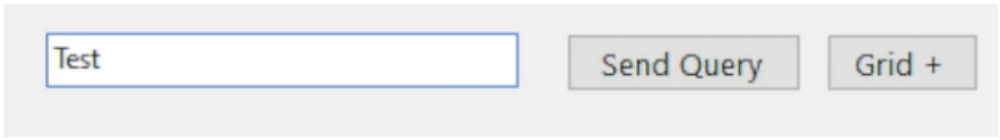
Written by: Rade Cvetkoski

Changing the size of the grid:





Before and after sending contact tracing query:



Comments:

Grid resizing in real time can position the position-marker out of bound for the other client, if the grid is scaled up and back down this issue can be resolved. For a final product this bug would need to be addressed.

With the current final build scaling one client will send a update that invokes an event for the other client, keeping the grid equal over all clients.
This client-to-client grid sizing update is handled through the position messages in the tracker class invoking an event in the main Contact tracing class.

-Dalton Christopher