**VIETNAM** **NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF SCIENCE**

**FACULTY OF INFORMATION TECHNOLOGY**



**LAB02 REPORT**

**SUBJECT: ARTIFICIAL INTELLIGENCE**

**Decision Tree with scikit-learn**

**Theory lecturers & instructors:**

Mr. Le Giang Thanh.

Mrs. Nguyen Thi Thanh Huyen.

**Students:**

Le Tan Dat 20127011

Dang Tien Dat 20127458

Pham Thi Anh Phat 20127680

CONTENTS

[I.INTRODUCTION 3](#_Toc80122182)

[II.CONTENTS OF REPORT: 4](#_Toc80122183)

*[A.SOCKET PROJECT](#_Toc80122184)* [4](#_Toc80122184)

[1. Programming language: C Sharp (C#), using Winform C# to design interface. 4](#_Toc80122185)

[2. Program script: 4](#_Toc80122186)

[3. Program structure 7](#_Toc80122187)

[ Implementation protocol: TCP. 7](#_Toc80122188)

[ Main functions: 7](#_Toc80122189)

[](#_Toc80122190) *[Client - Server communication:](#_Toc80122190)* [7](#_Toc80122190)

[](#_Toc80122191) *[Battle ship:](#_Toc80122191)* [10](#_Toc80122191)

[ Structure type: Divide into 2 namespaces: 11](#_Toc80122192)

[o Namespace server: 11](#_Toc80122193)

[o Namespace client: 13](#_Toc80122194)

*[B.WIRESHARK PROJECT](#_Toc80122195)* [15](#_Toc80122195)

[III.REFERENCES: 15](#_Toc80122196)

# INTRODUCTION

First of all, we would like to send our sincere thanks to the University of Science have brought Network Computer to the education program. Specially, we would like to thank deeply our theory lecturer & instructor – Mr. Le Giang Thanh and Mrs.Nguyen Thi Thanh Huyen, who have taught and imparted valuable knowledge to us during school time. During studying times, we have improved ourselves with many useful skills, serious and effective – learning spirit. This will definitely be valuable knowledge, plant the seed for us to go out into the wider world.

After this last – term project, we have improved a lot about communication skills, time management skills and problem – solving. So precious that we are learnt how to work like a team, listen to our partner, ask for helps and share difficulties with each other. Besides, acquiring knowledge by search on the web is one of the necessary skills needed in the Faculty of Information Technology.

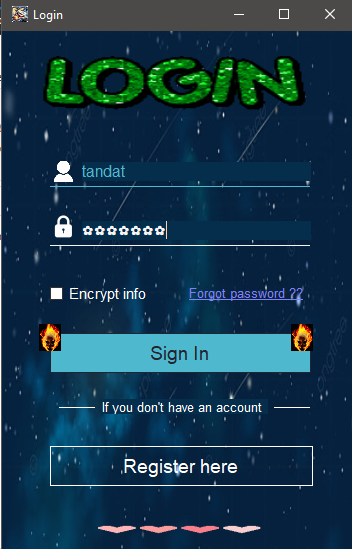
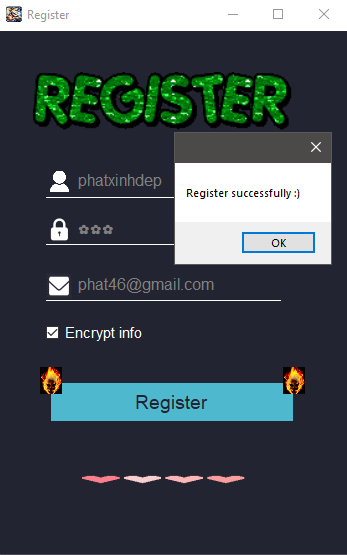
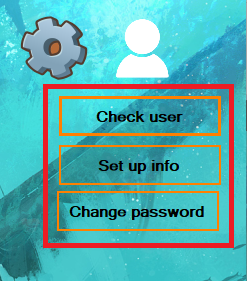
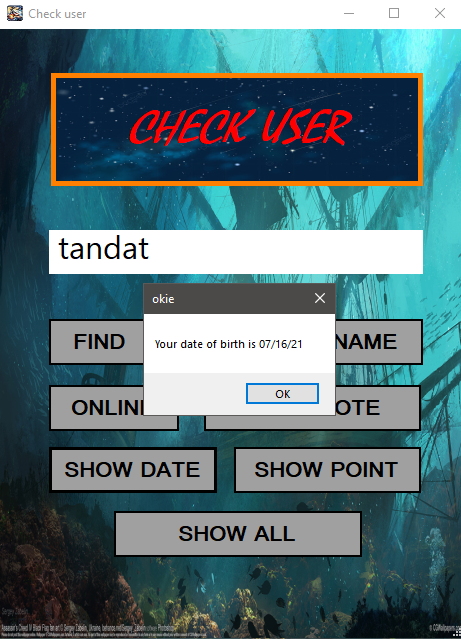
Network Computer is an interesting subject, very useful with high practicality. Guaranteed to provide enough knowledge, associated with the actual demand of students. However, because of our limited knowledge and receptive ability, even we have tried our best but certainly, our project is hard to avoid deficiencies and mistakes, we are hope that our report is considered and feedback to make our project more complete.

Sincerely yours.

# II. CONTENTS OF REPORT:

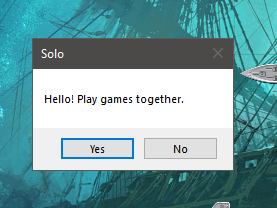
## *A.SOCKET PROJECT*

1. Programming language: C Sharp (C#), using Winform C# to design interface.
2. Program script:

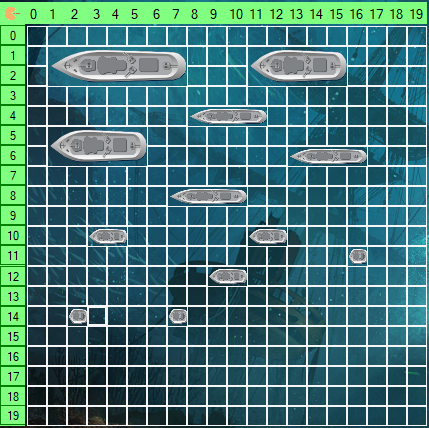
* Firstly, server passively waits for and responds to clients.
* Client initiates the communication.
* Using multi thread for each client which connects to server.
* Form main is form login, so when client connects to server, form login is displayed, if client did not have account before, form register is displayed while form login hide.
* Different between encypt and no encypt info in database.
* After that, form game is displayed, client can change password, set up information and check user.
* Client can enter rival name, if rival name is online and accept client’s request, form battle ship is displayed.

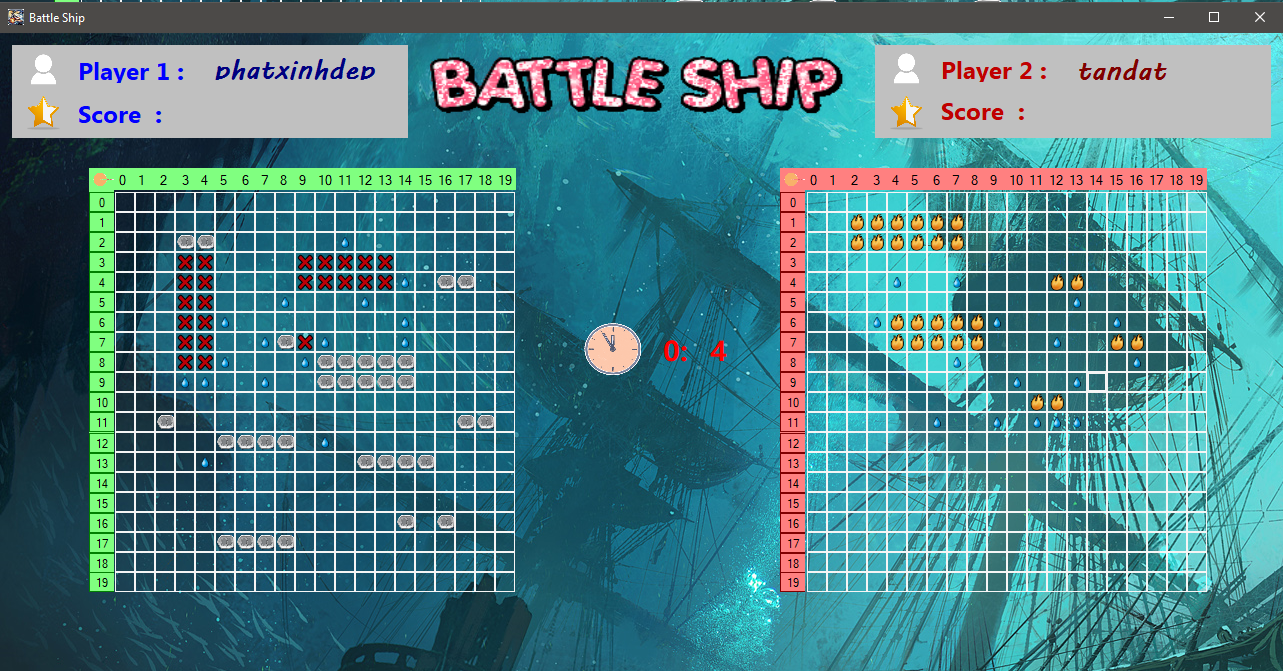
Client 1:

Client 2:



* And then, client has 60 seconds to set up ships into map



* After 60 seconds to set up ships, form battle ship is displayed and clients can play together. Client will have 20s for each turn.
* If client win game, they have 100 points else have no point.
* Hoi co choi tiep k?...
* In end game, form login is displayed, close form login to close socket client connection. Socket server is still wait for connecting.

1. Program structure

#### Implementation protocol: TCP.

#### Main functions:

##### ***Client - Server communication:***

TCP Socket:

Server Client

SOCKET()

CONNECT()

SEND()

RECEIVE()

CLOSE()

SOCKET()

BIND()

ACCEPT()

RECEIVE()

SEND()

LISTEN()

CLOSE()

MULTI THREAD

###### **Server:**

* static void Connect();
* Include commands: bind, Thread Listen and thread DoWork – function to accept client socket and send-receive loops.
* public static void serverReceive(Socket socket, ref string data);
* Server socket receive data.
* public static void serverSend(Socket socket, string result);
* Server socket send data.
* public staticList<string[]>ReadFile(string path);
* Read the database. Parameter string is path file.
* public static void WriteFile(string path,List< string[]> res);
* Write the database. Parameter string is path file. List<string[]> is structure for storing information in the database.
* public static void login(Socket socket);
* Receive information and check it in database. Send true if it is valid information else send false.
* public static void register(Socket socket);
* Receive information and check it in database. Send false if an account has already existed else send true and add information into database.
* public static void Check\_user(ref string data, ref string check);
* Receive username and check in database. Send false if it does not exist in database else send respond for client’s request. Parameter string check is check information is encrypted or not.
* public static void change\_newpass(ref string data, ref bool check);
* Receive old pass and check in database. Send false if it is incorrect pass else send true and update new pass in database. Parameter string check is to check information which is encrypted or not.
* public static void set\_up\_user(ref string data, ref string check);
* Receive information and check in database. And update new information in database and send true (to message is update successfully ). Parameter string check is to check information which is encrypted or not.
* public static void onlineChange(Socket socket);
* When client exit game, status is changes.

###### **Client:**

* void Connect();
* Client socket connect to IP Address of server
* public static void clientReceive(Socket socket, ref string data);
* Client socket receive data.
* public static void clientSend(Socket socket, string result);
* Client socket send data.
* Close();
* Client socket close when form login close .
* Event Login\_Click:
* Send option login to server and receive respond, if true form game is displayed else show message box is error information.
* Event Register\_Click:
* Send option register to server and receive respond, if true form game is displayed else show message box is client has existed already.
* Event Check\_user\_Click:
* Send option check user to server and receive respond, and show respond in message box.
* Event Change\_pass\_Click:
* Send option change pass to server and receive respond, if true show message is update successfully else update failed.
* Event setup\_info\_Click:
* Send option set up info to server and receive respond, if true show message is update successfully else update failed.

##### ***Battle ship:***

###### **Server:**

* public static void battle\_ship\_goto(Socket socket);
* Receive user name rival and check rival is online or not. And send request to rival and wait for respond. Then send respond to client 1 true if rival accept request otherwise send false.
* public static void map\_client(Socket socket);
* Write the map file.
* public static void attack(Socket socket);
* Receive attack position and check it in the map file, if it has ship in this position, send true otherwise send false.
* public static void handling\_attack(Socket socket, string data);
* Handle with status game:

“battle” : when 2 clients start attacking together.

“true\_accepted”: if request is accepted.

“false\_accepted”: if request is not accepted.

“send\_map”: send client’s map.

“attack”: call attack function and check it in map file.

“win”: if player win the game.

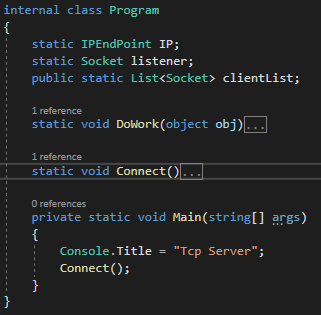
* public static List<string[]> ReadFile\_map(string path);
* Read the map file.
* public static void win\_lose(Socket socket);
* If client 1 win the game, update client’s point in database and send client 2 lost the game.
* public static void cleanFileMap();
* Use for delete file holding user’s map.

###### **Client:**

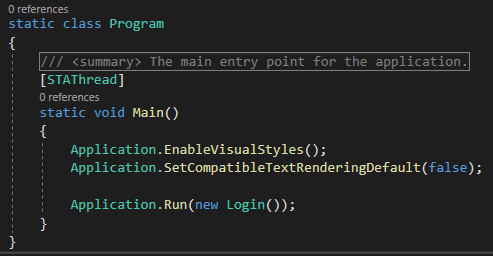
* Event frm\_battle\_ship\_Load:
* Draw two maps for client and start thread receive positions.
* void handling\_pos(Button btn);
* Get coordinates of attacked ship.
* void rec\_send\_btn(Button btn);
* Send position to server and count the number if player hit ships correctly. If count = 55 send option win to server. If hit ships incorrectly, swap client’s turn.
* void YesNo();
* Use for handling options of user about continue or not.
* Event btn\_Click:
* Clients choose position to attack and send coordinates to server.

#### Structure type: Divide into 2 namespaces:

##### **Namespace server:**

* Internal class Program:
* Main class in server. When built solution, class Program run firstly. Include main thread, connect function.
* Class Functions: login, register, set up information,…
* Class login\_register:
* Include funtions related to login and register.
* Class checkUser:
* All functions to check user and set up information .
* Class change\_pass:
* Include funtions to change password.
* Class related to battle ship
* Class player:
* Include lists to manage client in game.
* Class battle\_ship:
* Functions to battle ship between 2 clients.
* Class winLose:
* Functions to handle wins and losses.
* Class receive\_map:
* Include funtions to receive map and check the player's turn is successful or missed.
* Class connect\_and\_attack:
* Include function to handle client’s turn.
* Public class connectToClient:
* Include functions related to receive and send of server.
* Subclass
* Public class fFile:
* Include functions related to write and read file.
* Public class maHoa:
* Include encrypt and decypt function.
* Class Cons\_server
* Include variables to use in entire code.

##### **Namespace client:**

* static class program: Main class
* Include main function - load form login.
* Class functions
* public partial class Login : Form
* Functions, events related to login.
* public partial class Register : Form
* Functions, events related to register.
* public partial class form\_change\_password : Form
* Functions and events to change password.
* public partial class new\_pass : Form
* Functions, events related to change password.
* public partial class check\_user : Form
* Functions, events related to check user.
* public partial class set\_up\_user : Form
* Functions, events related to set up information.
* Class related to play game.
* public partial class frm\_battle\_ship : Form
* Function to draw two maps for clients, and thread receive position to attack.

Besides, we also divide into many regions : thread, timer and handling to manage code easily.

* class Socket\_using
* Client socket to use in entire code.
* class Player
* Variables username and point of clients to use in entire code.
* Subclass
* public class connectToServer
* Include functions related to receive and send of client.
* public class maHoa
* Functions encrypt and decrypt.
* Entended :
* static class AudioContext
* To play music when enter the room, when player win the game or when 2 players attack together.
* For each turn, players have 20s, so we use timer to do that:
  + void khoi\_tao\_timer();
  + Event time\_cool\_down\_Tick.

*B.WIRESHARK PROJECT*

* Packet capture tool: Wireshark Win64 Version 3.4.7.
* Virtual machine: Vmware Workstation Window 10 x64.
* Network interface: Wmware Network Adapter Vmnet8.
* IP Address: 192.168.172.129.
* Port: 9999.
* Protocol: TCP.
* Username: tandat.
* Password: 2.
* When user click , both username and password will be encrypted and sent to Server. Otherwise, just send username and password without encrypt.

|  |  |
| --- | --- |
| Packet comparision table | |
| Default data | Encrypted data |
| **Function:** Login. | |
|  |  |
| **Function:** Register. | |
|  |  |
| **Function:** Check user’s information. | |
|  |  |
| **Function:** Setup user’s information. |  |
|  |  |
| **Function:** Change password. |  |
|  |  |

**Summary:** With login, register and change password functions will have differences in ***encrypted*** and ***unencrypted*** data (username, password), in message has a flag which is true when sending ***encrypted*** message and false when sending ***unencrypted*** message.

III.REFERENCES:

1. <https://www.howkteam.vn/course/lap-trinh-game-caro-voi-c-winform-14>
2. <https://www.tutorialspoint.com/csharp/csharp_multithreading.htm>
3. <https://yinyangit.wordpress.com/2011/06/22/socket-communication-with-tcp-client-server/>
4. Video tutorials of Mr.Thanh and Mrs.Huyen.