

**System Programming**

**Com1564 Coursework**

**Student Name: Duong Bui Dinh**

**Student Roll Number: GT00010**

**Instructor: Hans Anderson**

Contents

[1. Introduction 3](#_Toc291743395)

[2. Analysis and Design 3](#_Toc291743396)

[3. Implementation 5](#_Toc291743397)

[4. Critical evaluation 5](#_Toc291743398)

## Introduction

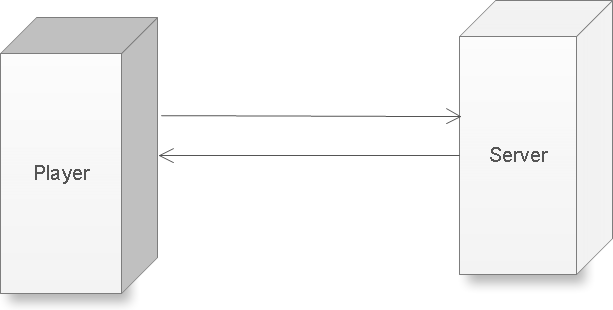
Hangman Game is game software which allows many players to interact over the local area network or internet. The game software has client-server architecture.

The rule to play game:

* Each two players play with each others, they have to find a hidden word to finish the game.
* One player who starts the game first guesses a character, if the hidden word contains the character then this player can continue finding word and the character will be displayed. Opposite, if the word doesn’t contain the character, the second player can play game.
* Who find the hidden word with all characters displayed fist, he/she is the winner.

## Analysis and Design

* 1. System architecture



* 1. Player functions
     1. Find server in Local Area Network (LAN)
     2. Login to server with alias name
     3. View all online players
     4. Join a room to play game
     5. Chat with all online players
     6. Chat with competitor in room when playing.
  2. Server functions
     1. Error Logging
     2. Manage all client
     3. Manage rooms
     4. Control sequence of game
* Player functions
  + **Find server in LAN**

When user runs the client program, the client application can automatically find the server in LAN.

There are steps to do this function:

* Port: (the server should have a fixed port to avoid doing port scanning -time problem).
* The server listens to UDP package data from client.
* Client sends a broadcast UDP package data to all computers in LAN with a fixed port, the package contains client IP Address.
* If server has a dynamic port, then the client has to send a UDP package data in a range of ports.
* After broadcasting the UDP data, the client has to wait for UDP data package from server. The package from server contains IP and Port.
* When client received the IP and Port from the server, the client can connect co Server with TCP/IP Socket.
  + **Login to server with alias Name:**
* User has to enter a name to connect server. The client application will send the name to server. the server manages all names received from all clients
  + **View all online players**

After logging in to server, the server sends an online player list to client. The client application will display the list to user.

* + **Join a room to play game**

After logging in to server, the server sends a room list to client. The client application will display the room list to user. Each room has maximum two players; the user can join any room which has one player inside. If a room has two players, then the room can start playing game.

* + **Chat with all online players**

After logging in to server, the client can chat with all online players.

* + **Chat with competitor player**

After logging in to server, the client can chat with all online players.

* Server functions:
  + **Error Logging**
* The server has to track all errors when running.
  + **Manage all client**
* The server has to manage all client, one client includes following information:
  + - Player information (Name, ID)
    - Socket information
* When client connected: add client
* When client disconnected: remove client
  + **Manage rooms**

The server can calculate the number of rooms for all clients.

When client connect, if lack of room, add new room.

When client disconnect, if remove all unnecessary rooms.

* + **Control sequence of game**

Any room has two players, the room can start playing. The player who joins the room first can start first.

The Plays play with the rule of game until the hidden word is found.

Each time players guess a character of hidden word:

* Client processes the word, if the word contains the character then set the state of character is “1”. By default all characters of hidden word is “0”.
* Send the processed word to server.
* Server can check the received word again, compares it with the word in server. If there is any change, then the client who sends the word can continue playing game and server update the word, the server also send the updated word to client . Or not, the second player can start the game.
* **Protocol Data Unit**

The message pass between client and server has to have following format:

* Header: message type, from PlayerID, to PlayerID
* Body: message contents

## Implementation

* Test

## Critical evaluation