**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**Viện Công nghệ thông tin và Truyền thông**



**Thiết kế giao thức tầng ứng dụng**

**Học phần: Lập trình mạng**

**Nhóm sinh viên:**

Nguyễn Bá Ngọc – 20173287

Nguyễn Đức Quyết – 20177021

Lê Đình Khánh – 20173194

**Giảng viên hướng dẫn:** TS. Phạm Huy Hoàng

*Hà Nội, ngày 12 tháng 06 năm 2020*

**Mục lục**

[I. Tổng quan hệ thống 3](#_Toc43820000)

[1. Mô tả 3](#_Toc43820001)

[2. Biểu đồ usecase 4](#_Toc43820002)

[2.1. Biểu đồ usecase tổng quan hệ thống 4](#_Toc43820003)

[2.2. Các biểu đồ usecase phân rã 5](#_Toc43820004)

[II. Thiết kế giao thức tầng ứng dụng 6](#_Toc43820005)

[1. Mô tả một số mã 6](#_Toc43820006)

[1.1. Status code 6](#_Toc43820007)

[1.2. Error code 6](#_Toc43820008)

[1.3. Request code 7](#_Toc43820009)

[1.4. Response Code từ Server 7](#_Toc43820010)

[2. Chi tiết thiết kế 8](#_Toc43820011)

[2.1. Đăng nhập 8](#_Toc43820012)

[2.2. Đăng ký 9](#_Toc43820013)

[2.3. Xem tất cả phòng 10](#_Toc43820014)

[2.4. Tạo phòng 10](#_Toc43820015)

[2.5. Vào phòng 11](#_Toc43820016)

[2.6. Bắt đầu game 12](#_Toc43820017)

[2.7. Rời phòng 13](#_Toc43820018)

[2.8. Xóa phòng 14](#_Toc43820019)

[2.9. Chat 14](#_Toc43820020)

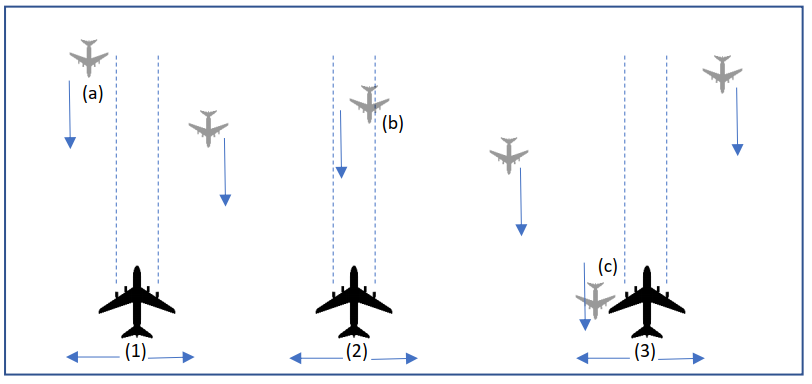
[2.10. Người chơi bị hạ 15](#_Toc43820021)

[2.11. Địch bị hạ 15](#_Toc43820022)

[2.12. Thao tác người chơi (di chuyển: trái, phải và bắn) 15](#_Toc43820023)

# Tổng quan hệ thống

## Mô tả

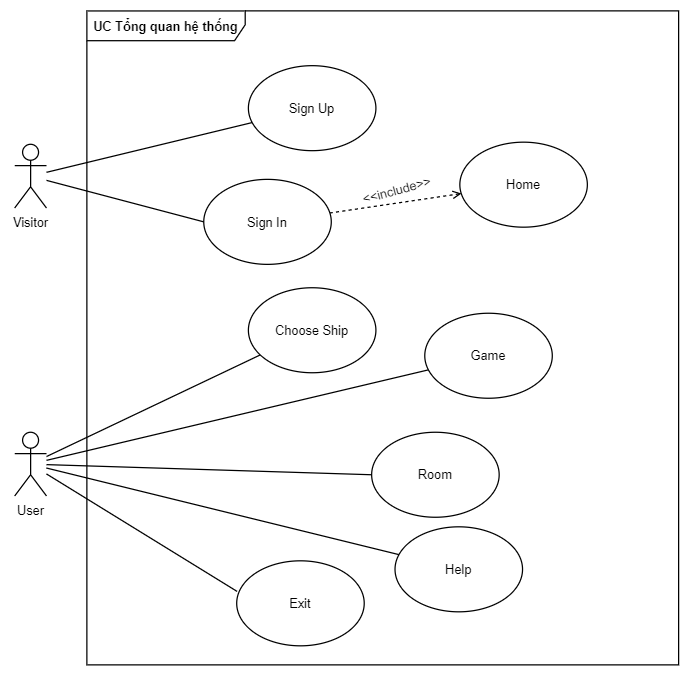
****

Hệ thống game máy bay chạy trên mạng theo mô hình client-server:

1. Server đợi nghe kết nối từ client để join vào game.
2. Nhiều client kết nối đến server, login, yêu cầu join game & start (sau khi đã có đủ người join).
3. Trong một bài (level), các client điều khiển máy bay của mình (số 1,2,3 trong hình vẽ) để di chuyển sang trái, phải và bắn đạn. Các máy bay địch di chuyển từ xa về gần (từ phía trên màn hình xuống dưới). Bài game level càng cao thì tốc độ máy bay địch càng lớn.
4. Màn hình của mỗi client hiển thị đầy đủ các máy bay quân ta (là những người join vào game) và các máy bay quân địch (được tạo ra ngẫu nhiên). Yêu cầu các màn hình client luôn được cập nhật thông tin từ server để hiển thị giống nhau và thể hiển giống nhau vị trí & di chuyển của các máy bay ta & địch.
5. Máy bay quân ta có thể bắn đạn vào máy bay địch. Trường hợp trúng đạn (điểm (b) trên hình), máy bay địch bị xóa khỏi màn hình của tất cả client.
6. Máy bay quân ta có thể bị máy bay địch đâm vào (điểm (c) trên hình). Khi đó người điều khiển máy bay quân ta nếu hết khiên thì sẽ bị thua và out khỏi game. Các người chơi còn lại vẫn tiếp tục.

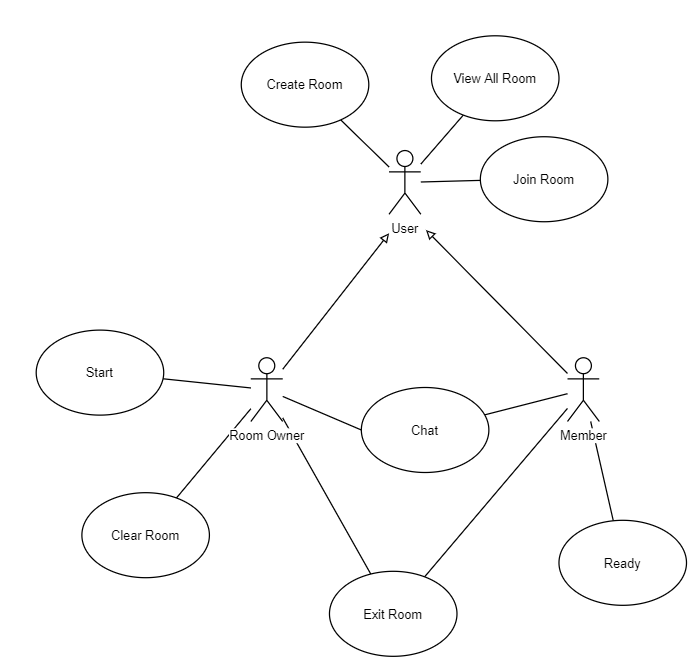
## Biểu đồ usecase

### Biểu đồ usecase tổng quan hệ thống

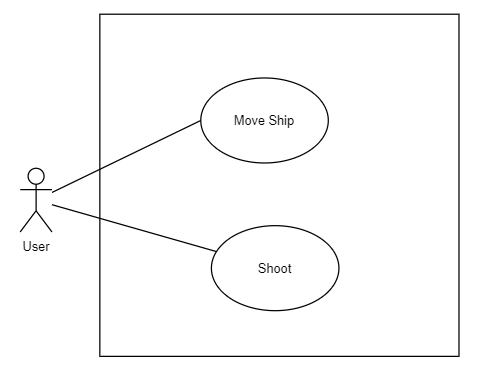
****

### Các biểu đồ usecase phân rã

#### Đặc tả usecase “Room”



#### Đặc tả usecase “Game”

****

# Thiết kế giao thức tầng ứng dụng

## Mô tả một số mã

### Status code

|  |  |
| --- | --- |
| **Code** | **Mô tả** |
| 0 | Success |
| 1 | Fail |

### Error code

|  |  |
| --- | --- |
| **Code** | **Mô tả** |
| 400 | Server lỗi |
| 401 | Lỗi đăng nhập |
| 402 | Lỗi đăng ký |
| 403 | Lỗi tạo phòng |
| 404 | Lỗi tham gia phòng |
| 405 | Lỗi rời phòng |
| 406 | Lỗi xóa phòng |

### Request code

|  |  |
| --- | --- |
| **Code** | **Mô tả** |
| 11 | Đăng nhập |
| 12 | Đăng ký |
| 20 | Xem tất cả phòng hiện có |
| 21 | Tạo phòng |
| 22 | Vào phòng |
| 23 | Bắt đầu game |
| 24 | Chat room |
| 25 | Sẵn sàng |
| 26 | Rời phòng |
| 27 | Xóa phòng |
| 41 | Thao tác người chơi (di chuyển, bắn) |
| 42 | Người chơi bị hạ |
| 43 | Địch bị hạ |
| 60 | Thiết lập kết nối udp |

### Response Code từ Server

|  |  |
| --- | --- |
| **Code** | **Mô tả** |
| 69 | Server lỗi |
| 81 | Người mới vào phòng |
| 82 | Thao tác người chơi (di chuyển, bắn) |
| 83 | Thiết lập kết nối udp |
| 84 | Gửi các thành phần game tới người chơi (địch,…) |
| 85 | Cập nhật điểm của người chơi |
| 86 | Level mới |
| 87 | Chiến thắng |
| 88 | Thất bại |
| 89 | Người chơi bị hạ |
| 110 | Đăng nhập |
| 120 | Đăng ký |
| 200 | Xem tất cả phòng |
| 210 | Tạo phòng |
| 220 | Vào phòng |
| 230 | Bắt đầu |
| 240 | Chat |
| 250 | Sẵn sàng |
| 260 | Rời phòng |
| 270 | Xóa phòng |
| 280 | Đặt chủ phòng mới |
| 290 | Thông báo người chơi rời phòng |

## Chi tiết thiết kế

Dữ liệu truyền được xây dựng là một đối tượng JSON, trước khi truyền sẽ được chuyển thành String. Bên nhận thực hiển chuyển ngược lại từ String thành JSON để xử lý.

### Đăng nhập

Request:

{

“req\_code”: 11,

“username”: <username>,

“password”: <password>

}

Response:

* Success:

{

“tcp\_code”: 110,

“status”: 1,

“user\_id”: <userId>

}

* Failed

{

“tcp\_code”: 110,

“status”: 0,

“error\_code”: 401,

“message”: “Tài khoản hoặc mật khẩu không chính xác”

}

{

“tcp\_code”: 110,

“status”: 0,

“error\_code”: 401,

“message”: “Tài khoản đang được sử dụng”

}

### Đăng ký

Request:

{

“req\_code”: 12,

“username”: <username>,

“password”: <password>

}

Response:

* Success

{

“tcp\_code”: 120,

“status”: 1

}

* Failed

{

“tcp\_code”: 120,

“status”: 0,

“error\_code”: 402,

“message”: “Tài khoản đã tồn tại”

}

### Xem tất cả phòng

Request:

{

“req\_code”: 20

}

Response:

{

“tcp\_code”: 200,

“status”: 1,

“list\_room”: [<list room information>]

}

***Note: Mỗi phần tử của list\_room là một JSONArray có dạng sau:***

[<roomID>, <roomName>, <capacity>, <isRunning>, <hasPassword>]

### Tạo phòng

Request:

{

“req\_code”: 21,

“room\_name””: <roomName>,

“room\_owner”: <roomOwner>,

“size”: <roomSize>,

“ship”: <shipName>,

“room\_pass”: <roomPass> **(có hoặc không)**

}

Response:

* Success

{

“tcp\_code”: 210,

“status”: 1,

“room”: {

“room\_id”: <id>,

“room\_name”: <roomName>,

“has\_pass”: <0 or 1>,

“room\_size”: <roomSize>

}

}

* Failed

{

“tcp\_code”: 210,

“status”: 0,

“error\_code”: 403,

“message”: “Phòng đã tồn tại. Hãy chọn tên khác”

}

### Vào phòng

Request:

{

“req\_code”; 22,

“room\_name”: <roomName>,

“player\_name”: <playerName>,

“ship”: <shipName>,

“room\_pass”: <roomPass> **(có hoặc không)**

}

Response:

* Success

{

“tcp\_code”: 220,

“status”: 1,

“room”: {

“room\_id”: <id>,

“room\_name”: <roomName>,

“has\_pass”: <0 or 1>,

“room\_size”: <roomSize>,

“owner\_id”: <ownerId>,

“team”: [<list informations of players>]

}

}

***Note: player’s information***

{

“name”: <playerName>,

“player\_id”: <playerId>,

“ship”: <shipName>

}

* Failed

{

“tcp\_code”: 220,

“status”: 0,

“error\_code”: 404,

“message”: <one of the messages below>

}

***Note: messages***

* *“Phòng không tồn tại”*
* *"Mật khẩu phòng không chính xác"*
* *"Phòng yêu cầu mật khẩu"*
* *"Phòng đã đầy"*
* *"Đang trong trận đấu"*

### Bắt đầu game

Mô tả: Chủ phòng gửi gói tin yêu cầu start game, server khi nhận được gói tin sẽ thực hiện gửi gói tin TCP yêu cầu các Client (các thành viên trong phòng) thiết lập kết nốt UDP.

Client request:

{

“req\_code”: 23

}

Server response:

* Success

{

“tcp\_code”: 230,

“status”: 1

}

* Failed

{

“tcp\_code”,

“status”: 0

}

Yêu cầu thiết lập kết nối:

{

“tcp\_code”: 83

}

### Rời phòng

Request:

{

“req\_code”: 26,

“player\_id”: <playerId>

}

Response:

* Success:

{

“tcp\_code”: 260,

“status”: 1

}

*Notify to all members*

{

“tcp\_code”: 290,

“player\_id”: <playerId>

}

*New owner*

{

“tcp\_code”: 280,

“owner\_id”: <ownerId>

}

* Failed:

{

“tcp\_code”: 260,

“status”: 0,

“error\_code”: 405,

“message”: “Đang trong trận đấu”

}

### Xóa phòng

**Mô tả:** Chủ phòng gửi request đến server thông báo xóa phòng, nếu đang trong game thì sẽ nhận được response failed, ngược lại server gửi response xóa phòng thành công đến tất cả các người chơi trong phòng.

Request:

{

“req\_code”: 27,

“player\_id”: <playerId>

}

Response:

* Status

{

“tcp\_code”: 270,

“status”: 1

}

* Failed

{

“tcp\_code”: 270,

“status”: 0,

“error\_code”: 406,

“message”: “Đang trong trận đấu”

}

### Chat

**Mô tả:** người chơi gửi bản tin request có chứa tin nhắn, server thực hiện gửi tin nhắn đó đến các người chơi còn lại trong phòng

Request:

{

“req\_code”: 24,

“sender”: <senderName>,

“data”: [<senderName>, <message>]

}

The server sends a response to all members

{

“tcp\_code”: 240,

“data”: [<senderName>, <message>]

}

### Người chơi bị hạ

**Mô tả:** khi người chơi bị hạ, sẽ gửi gói tin thông báo tới server, server gửi gói tin thông báo tới các người chơi khác (gửi id của người chơi vừa bị hạ đến những người chơi còn lại)

Request:

{

“req\_code”: 42,

“player\_id”: <playerId>

}

The server notifies to members

* Success

{

“tcp\_code”: 89,

“player\_id”: <playerId>

}

### Địch bị hạ

**Mô tả:** Khi người chơi bắn hạ quân địch sẽ gửi gói tin tới server (gồm id của người chơi và của địch vừa bị hạ), server thực hiện kiểm tra và nếu đúng là người chơi này bắn hạ địch thì gửi về điểm của người chơi.

Request:

{

“req\_code”: 43,

“enemy\_id”: <enemyId>,

“killer\_id”: <killerId>

}

Response:

{

“tcp\_code”: 85,

“score”: <yourScore>

}

### Thao tác người chơi (di chuyển: trái, phải và bắn)

**Mô tả:** các thao tác (di chuyển, bắn) của người chơi sẽ được gửi đến server thông qua giao thức UDP. Server thực hiện chuyển tiếp thông tin về thao tác này đến các người chơi còn lại

Gói tin gửi đi gồm các trường dữ liệu sau:

* thông tin về người chơi: shipId, vị trí, thao tác (trái, phải, bắn)
* id người chơi
* id của phòng

Request:

{

“req\_code”: 41,

“player\_id”: playerId,

“room\_id”: roomId,

“data”: [<shipId>, <layoutX>, <action>]

}

Response:

{

“udp\_code”: 82,

“payload”: data

}