

# University of Information Technology

Faculty of Computer Network and Communications



**PHẠM THANH TÂM – 21522573**

**NGUYỄN ĐÌNH KHA – 21520948**

**PHẠM ĐÔ THÀNH – 21522603**

**NGUYỄN VĂN ANH TÚ – 21520514**

## **FINAL REPORT**

**Subject: Network Programming**

**Class: NT106.N22.ATCL**

## **TOP DOWN PIXEL GAME USING UNITY**

**LECTURER: Đỗ Thị Hương Lan**

**HOCHIMINH CITY, 2023**

## **LỜI CẢM ƠN**

Để đạt được kết quả tốt trong dự án này, ngoài sự nỗ lực hết mình của tất cả các thành viên trong nhóm, chúng em còn nhận được sự hướng dẫn đầy tận tâm từ cô Đỗ Thị Hương Lan. Niềm đam mê của cô trong mỗi buổi học đã cung cấp cho chúng em kiến thức quý báu không chỉ trong phạm vi sách giáo khoa mà còn trong nhiều khía cạnh thực tế và xã hội khác. Với lòng biết ơn chân thành và sâu sắc, chúng em muốn bày tỏ lòng cảm kích đến cô vì đã nhiệt tình và cống hiến cho sinh viên của mình. Điều đó đã là động lực lớn giúp chúng em hoàn thành dự án này một cách thành công.

Mặc dù đã nỗ lực hết sức, nhưng do thời gian và kinh nghiệm hạn chế, đề án này có thể có một số thiếu sót. Chúng em chân thành mong nhận được sự hướng dẫn và giúp đỡ từ cô để bổ sung và nâng cao kiến thức của chúng em để phục vụ cho luận văn tốt nghiệp và các dự án trong tương lai tốt hơn. Chúng em chúc cô sức khỏe dồi dào, thành công và luôn có niềm đam mê bốc lửa với nghề nghiệp để hướng dẫn cho nhiều thế hệ học sinh trong tương lai.

Chúng em chân thành cảm ơn cô!

**Group 14.NT106.N22.ATCL**

# Mục lục

1. Giới thiệu .....	4
1.1 Unity .....	4
1.1.1 Unity là gì .....	4
1.1.2 Ưu điểm của Unity .....	4
1.2 Unity.NETCode .....	5
1.2.1 Unity.NETCode là gì? .....	5
1.2.2 Ưu điểm của Unity.NETCode.....	5
1.3 RelayUnity .....	5
1.3.1 RelayUnity là gì? .....	5
1.3.2 Ưu điểm của RelayUnity .....	6
1.4 PlayFab .....	6
1.4.1 PlayFab là gì? .....	6
1.4.2 Ưu điểm của PlayFab.....	6
2. Tổng quan về trò chơi.....	7
2.1 Ý tưởng .....	7
2.2 Các thành phần trong game và giao diện.....	7
2.3 Cơ chế game .....	9
2.3.1 Gameplay.....	9
2.3.2 GameMechanic .....	10
3. Triển khai Network .....	11
3.1 Cơ sở lý thuyết.....	11
3.1.1 Unity và thư viện Unity.NetCode và RelayUnity.....	11
3.1.2 Game server và client .....	11
4. Cơ sở dữ liệu và Lưu trữ .....	12
4.1 Quản lý tài khoản .....	12
4.2 Bảo mật trong việc lưu trữ .....	13
4.3 Triển khai cơ sở dữ liệu và lưu trữ .....	14
5. Triển khai ứng dụng .....	16
5.1 Triển khai qua LAN .....	16
5.2 Triển khai qua Internet.....	17
6. Bảng phân chia công việc.....	18
7. Đánh giá công việc .....	19

# 1. Giới thiệu

## 1.1 Unity

### 1.1.1 Unity là gì

Unity là một phần mềm phát triển game hay còn được gọi là game engine đa nền tảng được phát triển bởi công ty Unity Technologies, chủ yếu được dùng để phát triển video game cho 27 nền tảng bao gồm máy tính, game consoles (như PlayStation) và điện thoại, . . . Hơn 50% số lượng game trên thị trường được sản xuất bởi Unity. Một vài tựa game vô cùng nổi tiếng được tạo ra bởi Unity có thể được kể đến như Pokémon Go, Hearthstone, Ori And The Blind Forest, Monument Valley, Axie Infinity, . . . Độ “phủ sóng” của Unity rất rộng, có thể được áp dụng phổ biến trong nhiều dòng game khác nhau từ game “hạng nặng” Triple A (AAA) cho đến game giáo dục đơn giản cho con nít.

### 1.1.2 Ưu điểm của Unity

Là một cross-platform engine xét về sức mạnh tổng quan thì Unity có nhiều ưu điểm như:

- Ngôn ngữ hỗ trợ: Unity hỗ trợ người dùng sử dụng C#. Với thư viện .NET của C# có thể cho phép phục vụ mục đích của đồ án này
- Editor: Với Editor, nhà phát triển không cần thiết phải viết Code để sắp đặt các đối tượng trong Game như những Engine khác mà Developer có thể kéo thả, thay đổi vị trí của từng đối tượng trong Game trực tiếp trên Editor.
- Đa nền tảng: là lợi ích thứ 3 rất quan trọng với nhiều công ty cũng như developer. Vì với việc bạn tạo ra Game mà Game đó có thể chạy được trên hầu hết những hệ điều hành quan trọng như Desktop (Mac, Window và Linux) hay Mobile (iOS, Android) hoặc Web (WebGL) thì cũng đã tiết kiệm công sức cũng như chi phí rất nhiều cho doanh nghiệp đó.
- Miễn phí: Và yếu tố cuối cùng chính là chi phí. Với Unity, miễn phí là một điểm thu hút rất nhiều Developer chọn làm việc với game engine này. Tuy

nhien, với các game được tạo ra miễn phí thì bắt buộc phải có Logo Unity trong Game.

## 1.2 Unity.NETCode

### 1.2.1 Unity.NETCode là gì?

Unity.NETCode là một framework được cung cấp bởi Unity Technologies để hỗ trợ việc phát triển game đa người chơi trực tuyến (multiplayer) trên nền tảng Unity.

### 1.2.2 Ưu điểm của Unity.NETCode

- Cung cấp một loạt các công cụ và tính năng để giúp các nhà phát triển tạo ra các trò chơi đa người chơi trực tuyến một cách dễ dàng hơn và nhanh chóng hơn. Nó giúp đơn giản hóa việc xây dựng các tính năng đa người chơi như đồng bộ hóa dữ liệu, phát hiện va chạm, lập lịch và quản lý kết nối giữa các client.
- Hỗ trợ nhiều mô hình kết nối khác nhau như client-server, peer-to-peer và hybrid, cho phép các nhà phát triển tùy chỉnh hệ thống mạng cho phù hợp với nhu cầu của trò chơi của mình.

## 1.3 RelayUnity

### 1.3.1 RelayUnity là gì?

RelayUnity là một dịch vụ đám mây (cloud service) được cung cấp bởi Unity Technologies để giúp cho việc phát triển game đa người chơi trực tuyến trên nền tảng Unity trở nên dễ dàng hơn. Relay Unity cung cấp các tính năng để quản lý kết nối giữa các client trong game đa người chơi, đồng bộ hóa dữ liệu và giảm thiểu độ trễ trong trò chơi.

### 1.3.2 Ưu điểm của RelayUnity

- RelayUnity sử dụng một mạng lưới toàn cầu (global network) để giảm thiểu độ trễ (latency) trong việc kết nối giữa các client trong trò chơi đa người chơi. Nó cũng hỗ trợ việc phát hiện và xử lý các sự cố kết nối (connection issues) giữa các client, giúp giảm thiểu thời gian chết (downtime) trong trò chơi.
- Relay Unity cũng cung cấp các tính năng bảo mật để đảm bảo an toàn cho các kết nối trong trò chơi đa người chơi, bao gồm việc mã hóa dữ liệu và xác thực định danh (authentication).

## 1.4 PlayFab

### 1.4.1 PlayFab là gì?

PlayFab là một platform dịch vụ đám mây (cloud service) được cung cấp bởi Microsoft để hỗ trợ việc phát triển game trực tuyến (online games) trên nền tảng đa nền tảng (multi-platform), bao gồm các nền tảng như Windows, Xbox, PlayStation, iOS và Android.

### 1.4.2 Ưu điểm của PlayFab

- Quản lý người chơi: PlayFab cung cấp các tính năng để quản lý người chơi, bao gồm đăng ký, đăng nhập, xác thực và quản lý tài khoản. Điều này giúp cho các nhà phát triển game dễ dàng quản lý người chơi của mình và tạo ra trải nghiệm tốt nhất cho họ.
- Đa nền tảng: PlayFab hỗ trợ việc phát triển trò chơi trên nhiều nền tảng khác nhau, bao gồm Windows, Xbox, PlayStation, iOS và Android. Điều này giúp cho các nhà phát triển có thể đưa trò chơi của mình đến nhiều người chơi hơn và mở rộng thị trường của mình.
- Các tính năng đa dạng: PlayFab cung cấp các tính năng đa dạng để hỗ trợ việc phát triển trò chơi trực tuyến, bao gồm quản lý người chơi, đồng bộ hóa dữ liệu, phân tích và thống kê, thanh toán và nhiều tính năng khác.
- Dịch vụ đám mây: PlayFab là một dịch vụ đám mây, cho phép các nhà phát triển không cần phải tạo ra và vận hành các hệ thống backend riêng cho trò

chơi của mình. Điều này giúp tiết kiệm chi phí và tập trung vào việc phát triển trò chơi.

## 2. Tổng quan về trò chơi

### 2.1 Ý tưởng

Game Bombman là một trò chơi được lấy ý tưởng từ những game đặt bom tuổi thơ trên các trang web online mà bất kì bạn trẻ nào cũng từng chơi qua. Nơi mà người chơi sẽ là 1 player được đưa vào một sân đấu trường gồm các player khác và họ sẽ đấu với nhau để trở thành người thắng cuộc.

### 2.2 Các thành phần trong game và giao diện



Figure 1. Giao diện chính của game

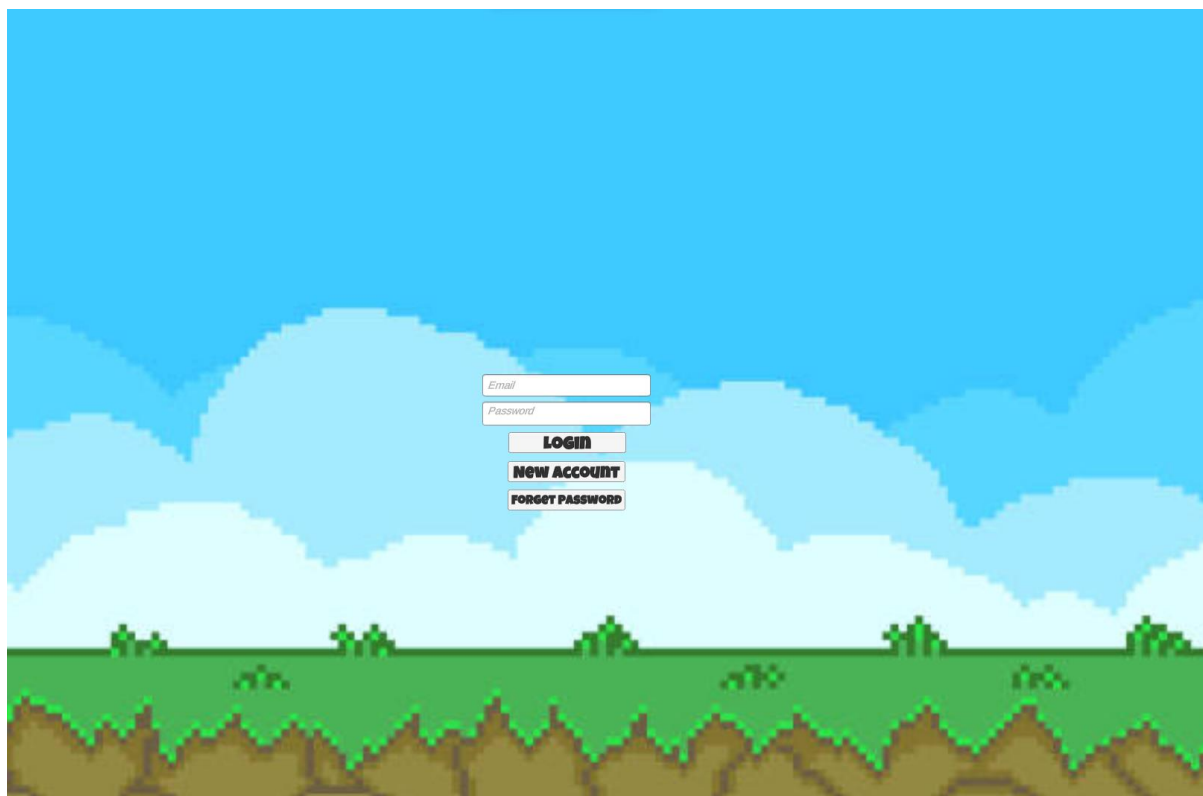


Figure 2. Giao diện đăng nhập và đăng ký

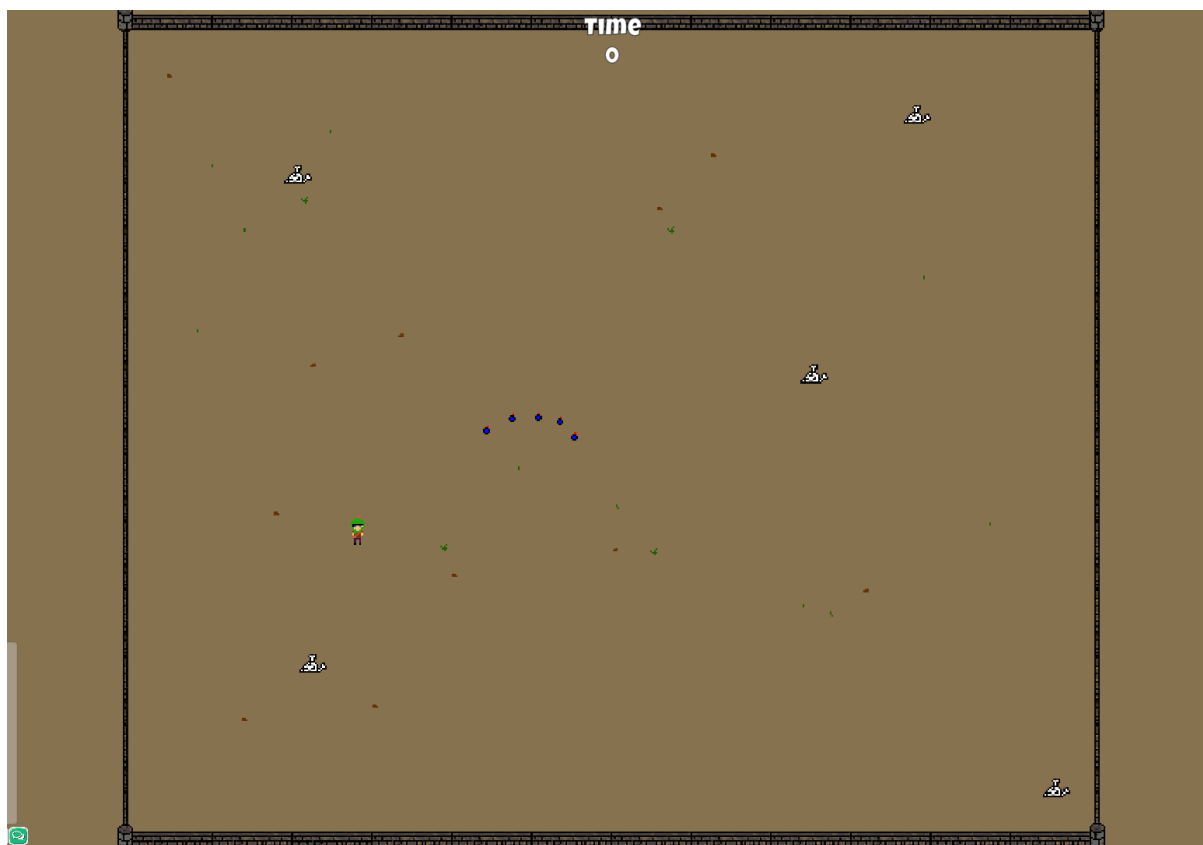




Figure 3. Giao diện trong game

Các vật thể trong game:

- Player: Là 1 character được điều khiển bởi player
- Map : Đấu trường, khu vực để người chơi di chuyển và tranh đấu
- Bomb : quả bom được tạo ra bởi người chơi
- Bullet : Sau khi quả bom nổ sẽ tạo thành các bullet nếu trúng người chơi sẽ bị trừ 1 điểm

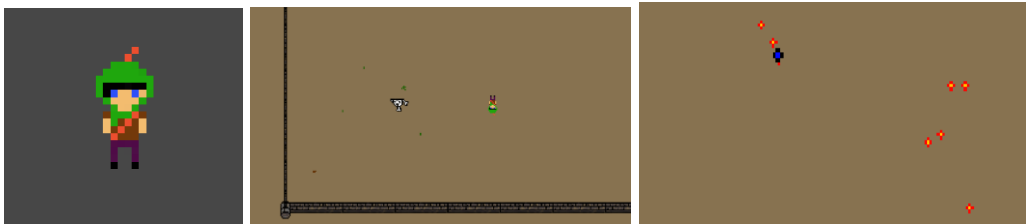


Figure 4. Các vật thể trong game

## 2.3 Cơ chế game

### 2.3.1 Gameplay

Mỗi người chơi sẽ điều khiển 1 nhân vật di chuyển trong khu vực đấu trường để đặt bom và tiêu diệt lẫn nhau, đồng thời người chơi cũng phải khéo léo né tránh các Bullet được tạo ra sau khi bom nổ để hạn chế số điểm trừ một cách tối đa.

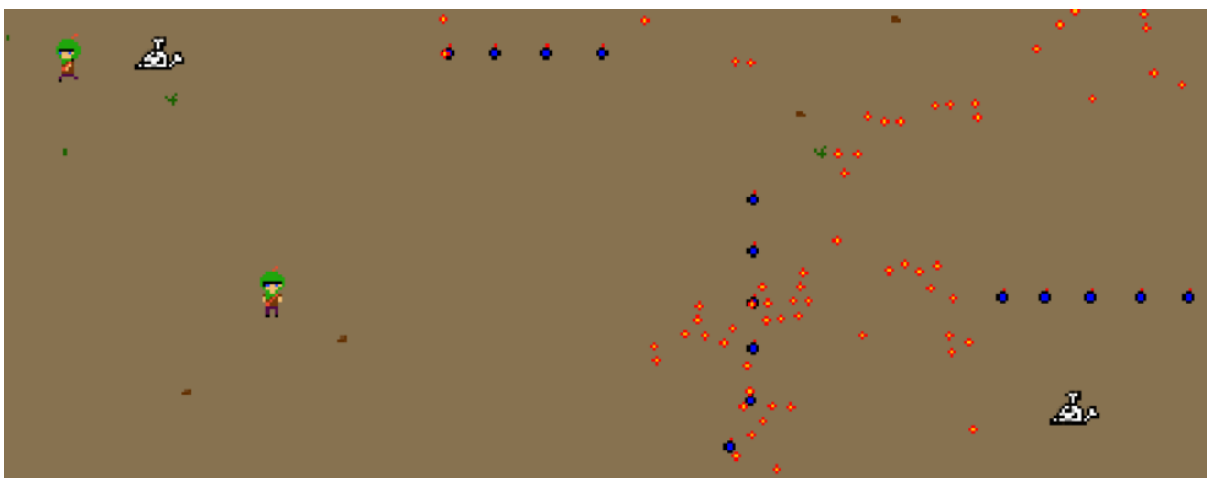


Figure 5. Gameplay

### 2.3.2 GameMechanic

Một số cơ chế chính có mặt trong game để hỗ trợ Gameplay như:

- Điều khiển: Người chơi sử dụng các phím WASD hoặc các mũi tên lên xuống trái phải trong keyboard
- Cơ chế hồi sinh: Sau khi người chơi trúng Bullet sẽ được hồi sinh ngẫu nhiên tại vị trí nào đó trong đấu trường ngay lập tức
- Tính điểm: Khi player trúng Bullet sẽ bị coi như là chết và bị trừ 1 điểm, điểm của player nào bị trừ ít nhất sẽ thắng cuộc
- Nhắn gửi tin: Những người chơi trong cùng server có thể nhắn và gửi tin đến mọi người.



Figure 6. Khung chat

## 3. Triển khai Network

### 3.1 Cở sở lý thuyết

#### 3.1.1 Unity và thư viện Unity.NetCode và RelayUnity

Như đã đề cập ở trên Unity.Netcode và RelayUnity giúp ta dễ dàng thực hiện việc kết nối và gửi dữ liệu qua mạng. Nó cho phép ta thực hiện tạo các kết nối, gửi dữ liệu UDP và TCP đến server và ngược lại. Tùy thuộc vào nhu cầu mà ta sẽ xác định phương thức gửi riêng cho từng tác vụ

#### 3.1.2 Game server và client

Server và Client hay còn gọi là máy chủ và máy khách. Mặc dù cùng một tựa game, nhưng cả hai đều mang những nhiệm vụ khác biệt trong vấn đề xử lý thông tin:

- **Game server:** Là một hệ thống máy chủ được thiết kế để quản lý các thông tin và dữ liệu của game. Đây là nơi lưu trữ tất cả những gì bạn thấy trong game, bao gồm cả thông tin về nhân vật, vật phẩm, bản đồ, cốt truyện, đồng thời cũng là nơi thực hiện việc tính toán các tương tác vật lý trong trò chơi. Tất cả những gì bạn làm trong game đều được lưu lại trên máy chủ.
- **Client:** Nơi người chơi tương tác vào trò chơi như di chuyển, tấn công kẻ địch. Client không trực tiếp tính toán và xử lý mà sẽ gửi thông tin những hành động này tới server để xử lý, server sau khi xử lý và tính toán sẽ trả lại kết quả và hiển thị cho người chơi.



- Server: Tính toán và truyền dữ liệu.



- Player:  
Người chơi



- Database:  
Lưu trữ dữ  
liệu tài khoản

Figure 7. Mô hình các Node trong đồ án

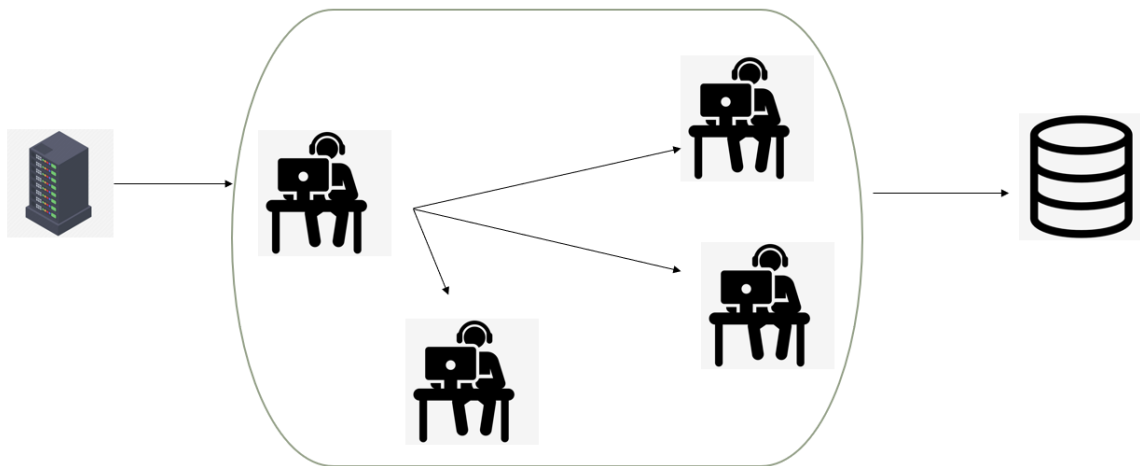


Figure 8. Mô hình hệ thống

Khi này 1 player sẽ kết nối trực tiếp tới server bằng cách bấm vào Create phòng để host, player này sẽ đóng vai trò như 1 server con và các player khác sẽ đóng vai trò là Client. Khi người host tạo phòng sẽ tạo ra 1 mã code cho phòng đó, và các players khác sẽ nhập code để vào phòng.



Figure 9. Tạo phòng và vào phòng

## 4. Cơ sở dữ liệu và Lưu trữ

### 4.1 Quản lý tài khoản

Dữ liệu tài khoản của người chơi sẽ được lưu trữ trên một database đó là PlayFab, database này sẽ đóng vai trò lưu trữ cũng như việc truy xuất thông tin đăng nhập và đăng ký tài khoản.





Most recent logins		Search players				Search	Clear
Type	Name	Last login ↓	Created	Country/region	VTD		
	49DC1336721CB744 khada123	Jun 20, 2023 5:59 PM	13 days ago	Viet Nam	\$0.00		
	5115C83004F7AEE1 cop123	Jun 20, 2023 4:18 PM	13 days ago	Viet Nam	\$0.00		
	3A3CADC87F6C947E tudu123	Jun 20, 2023 3:55 PM	13 days ago	United States	\$0.00		

Figure 10. Dữ liệu tài khoản được lưu trên database

7 logins in the last 30 days

Date	Platform	Source	IP address	City	Country/region
6/20/2023 5:59 PM	PlayFab	PlayFab	171.235.147.40	Ho Chi Minh City	VN
6/20/2023 5:56 PM	PlayFab	PlayFab	171.235.147.40	Ho Chi Minh City	VN
6/20/2023 5:38 PM	PlayFab	PlayFab	171.235.147.40	Ho Chi Minh City	VN
6/20/2023 5:32 PM	PlayFab	PlayFab	113.22.245.185	Ho Chi Minh City	VN
6/20/2023 5:08 PM	PlayFab	PlayFab	42.115.115.121	Hanoi	VN
6/20/2023 5:02 PM	PlayFab	PlayFab	171.235.147.40	Ho Chi Minh City	VN
6/20/2023 5:01 PM	PlayFab	PlayFab	171.235.147.40	Ho Chi Minh City	VN

Figure 11. Dữ liệu đăng nhập được lưu trên database


**PLAYFAB**

### Reset your 'My Game' account password

**Password**

**Repeat Password**

**Reset Password**

Figure 12. Cơ chế phục hồi mật khẩu

## 4.2 Bảo mật trong việc lưu trữ

Sau khi đăng kí, dữ liệu mật khẩu sẽ không hiện rõ ràng trên Database nhưng sẽ được hash ra thành một mã bởi thuật toán SHA-256 và thành một mã ID. Mỗi giá trị sẽ tương ứng với 1 tài khoản cố định.



Figure 13. Dữ liệu ID được lưu trên database

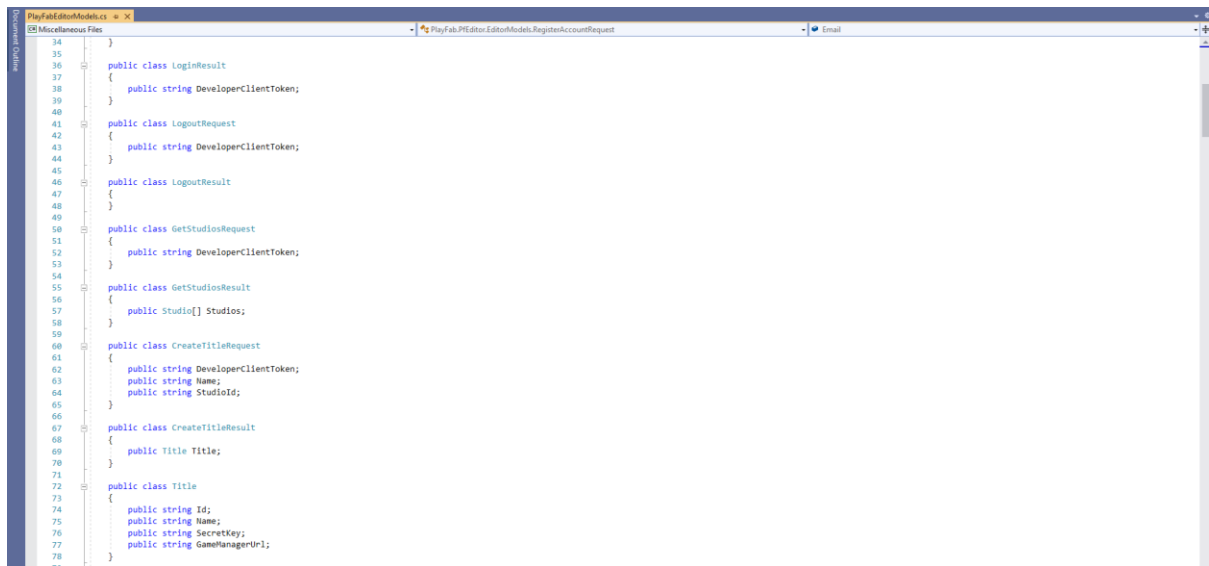
### 4.3 Triển khai cơ sở dữ liệu và lưu trữ

Để lưu các giá trị tài khoản ta sẽ khởi tạo các biến và kết nối lên PlayFab ta sẽ liên kết API tới PlayFab :

```
PlayFabEditorModels.cs
Miscellaneous Files
PlayFab.PfEditor.EditorModels.RegisterAccountRequest

1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4
5 namespace PlayFab.PfEditor.EditorModels
6 {
7     public class RegisterAccountRequest
8     {
9         public string Email;
10        public string Password;
11        public string StudioName;
12        public string DeveloperToolProductName;
13        public string DeveloperToolProductVersion;
14    }
15
16    public class RegisterAccountResult
17    {
18        public string DeveloperClientToken;
19    }
20
21    public class LoginRequest
22    {
23        public string Email;
24        public string Password;
25        public string TwoFactorAuth;
26        public string DeveloperToolProductName;
27        public string DeveloperToolProductVersion;
28    }
29
30    public class LoginWithAADRequest
31    {
32        public string DeveloperToolProductName;
33        public string DeveloperToolProductVersion;
34    }
35
36    public class LoginResult
37    {
38        public string DeveloperClientToken;
39    }
40
41    public class LogoutRequest
42    {
43        public string DeveloperClientToken;
44    }
45
46    public class LogoutResult
47    {
```

Figure 14. Khởi tạo các giá trị tài khoản



```
34 }
35
36 public class LoginResult
37 {
38     public string DeveloperClientToken;
39 }
40
41 public class LogoutRequest
42 {
43     public string DeveloperClientToken;
44 }
45
46 public class LogoutResult
47 {
48 }
49
50 public class GetStudiosRequest
51 {
52     public string DeveloperClientToken;
53 }
54
55 public class GetStudiosResult
56 {
57     public Studio[] Studios;
58 }
59
60 public class CreateTitleRequest
61 {
62     public string DeveloperClientToken;
63     public string Name;
64     public string StudioId;
65 }
66
67 public class CreateTitleResult
68 {
69     public Title Title;
70 }
71
72 public class Title
73 {
74     public string Id;
75     public string Name;
76     public string SecretKey;
77     public string GameManagerUri;
78 }
```

Figure 15. Khởi tạo các giá trị tài khoản



```
79
80 public class Studio
81 {
82     public static Studio OVERRIDE = new Studio { Id = "", Name = PlayFabEditorHelper.STUDIO_OVERRIDE, Titles = null };
83
84     public string Id;
85     public string Name;
86
87     public Title[] Titles;
88
89     public Title GetTitle(string titleId)
90     {
91         if (Titles == null)
92             return null;
93         for (var i = 0; i < Titles.Length; i++)
94             if (Titles[i].Id == titleId)
95                 return Titles[i];
96         return null;
97     }
98
99     public string GetTitleSecretKey(string titleId)
100     {
101         var title = GetTitle(titleId);
102         return title == null ? "" : title.SecretKey;
103     }
104 }
105
106 public class GetTitleDataRequest
107 {
108     public List<string> Keys;
109 }
110
111 public class GetTitleDataResult
112 {
113     public Dictionary<string, string> Data;
114 }
115
116 public class SetTitleDataRequest
117 {
118     public string Key;
119     public string Value;
120 }
121
```

Figure 16. Khởi tạo các giá trị tài khoản

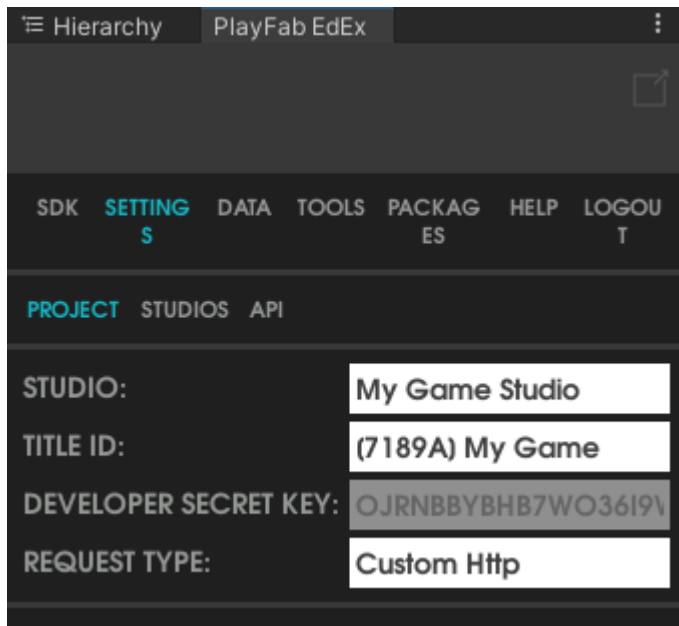


Figure 17. Liên kết API tới Database

## 5. Triển khai ứng dụng

### 5.1 Triển khai qua LAN

Để có thể triển khai ứng dụng qua LAN ta cần một số điều kiện từ 2 bên Server và Client là:

- Kết nối chung một mạng cục bộ: Cả 2 sẽ phải kết nối chung một mạng cục bộ như wifi.
- Địa chỉ IP và Port: Client cần phải biết được địa chỉ IP và Port của server được tạo để có thể kết nối.

Ngoài ra ta có thể sử dụng loop back ip là 127.0.0.1 để có thể tiến hành kết nối nếu server và client cùng chạy trên một máy.

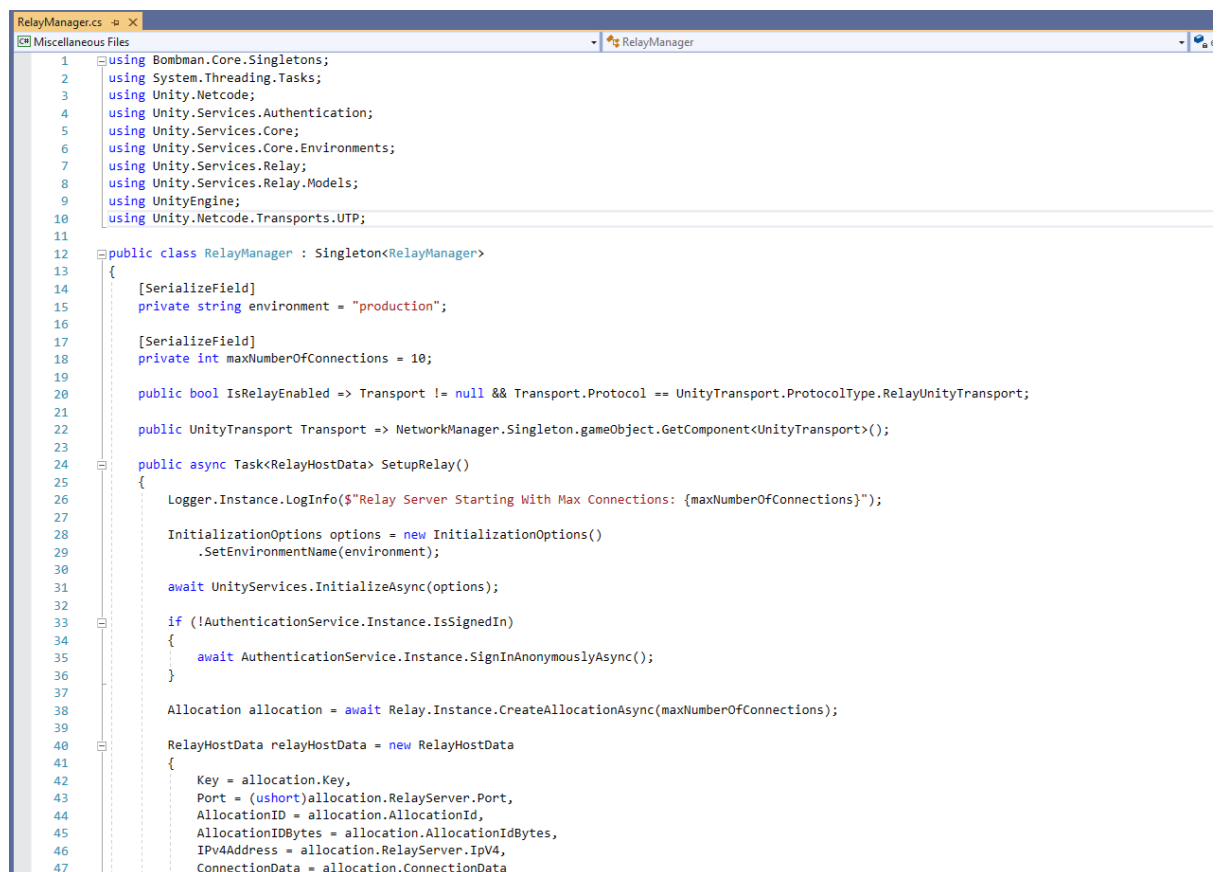


## 5.2 Triển khai qua Internet

Để triển khai server qua internet thì việc sử dụng Local IP để kết nối là điều không thể. Vậy nên ta cần sử dụng một số dịch vụ cung cấp từ bên thứ 3. Và ở đây sẽ sử dụng RelayUnity.

Để triển khai vào tựa game ta sẽ cấu hình và khởi tạo RelayUnity server

- Triển khai và khởi tạo RelayUnity server:



```
1 using Bombman.Core.Singletons;
2 using System.Threading.Tasks;
3 using Unity.Netcode;
4 using Unity.Services.Authentication;
5 using Unity.Services.Core;
6 using Unity.Services.Core.Environments;
7 using Unity.Services.Relay;
8 using Unity.Services.Relay.Models;
9 using UnityEngine;
10 using Unity.Netcode.Transports.UTP;
11
12 public class RelayManager : Singleton<RelayManager>
13 {
14     [SerializeField]
15     private string environment = "production";
16
17     [SerializeField]
18     private int maxNumberOfConnections = 10;
19
20     public bool IsRelayEnabled => Transport != null && Transport.Protocol == UnityTransport.ProtocolType.RelayUnityTransport;
21
22     public UnityTransport Transport => NetworkManager.Singleton.gameObject.GetComponent<UnityTransport>();
23
24     public async Task<RelayHostData> SetupRelay()
25     {
26         Logger.Instance.LogInfo($"Relay Server Starting With Max Connections: {maxNumberOfConnections}");
27
28         InitializationOptions options = new InitializationOptions()
29             .SetEnvironmentName(environment);
30
31         await UnityServices.InitializeAsync(options);
32
33         if (!AuthenticationService.Instance.IsSignedIn)
34         {
35             await AuthenticationService.Instance.SignInAnonymouslyAsync();
36         }
37
38         Allocation allocation = await Relay.Instance.CreateAllocationAsync(maxNumberOfConnections);
39
40         RelayHostData relayHostData = new RelayHostData
41         {
42             Key = allocation.Key,
43             Port = (ushort)allocation.RelayServer.Port,
44             AllocationID = allocation.AllocationId,
45             AllocationIDBytes = allocation.AllocationIdBytes,
46             IPv4Address = allocation.RelayServer.IpV4,
47             ConnectionData = allocation.ConnectionData
```

Figure 18. Code cấu hình và khởi tạo RelayUnity Server

```

47         ConnectionData = allocation.ConnectionData
48     };
49
50     relayHostData.JoinCode = await Relay.Instance.GetJoinCodeAsync(relayHostData.AllocationID);
51
52     Transport.SetRelayServerData(relayHostData.Ipv4Address, relayHostData.Port, relayHostData.AllocationIDBytes,
53         relayHostData.Key, relayHostData.ConnectionData);
54
55     Logger.Instance.LogInfo($"Relay Server Generated Join Code: {relayHostData.JoinCode}");
56
57     return relayHostData;
58 }
59
60 public async Task<RelayJoinData> JoinRelay(string joinCode)
61 {
62     Logger.Instance.LogInfo($"Client Joining Game With Join Code: {joinCode}");
63
64     InitializationOptions options = new InitializationOptions()
65         .SetEnvironmentName(environment);
66
67     await UnityServices.InitializeAsync(options);
68
69     if (!AuthenticationService.Instance.IsSignedIn)
70     {
71         await AuthenticationService.Instance.SignInAnonymouslyAsync();
72     }
73
74     JoinAllocation allocation = await Relay.Instance.JoinAllocationAsync(joinCode);
75
76     RelayJoinData relayJoinData = new RelayJoinData
77     {
78         Key = allocation.Key,
79         Port = (ushort)allocation.RelayServer.Port,
80         AllocationID = allocation.AllocationID,
81         AllocationIDBytes = allocation.AllocationIDBytes,
82         ConnectionData = allocation.ConnectionData,
83         HostConnectionData = allocation.HostConnectionData,
84         Ipv4Address = allocation.RelayServer.Ipv4,
85         JoinCode = joinCode
86     };
87
88     Transport.SetRelayServerData(relayJoinData.Ipv4Address, relayJoinData.Port, relayJoinData.AllocationIDBytes,
89         relayJoinData.Key, relayJoinData.ConnectionData, relayJoinData.HostConnectionData);
90
91     Logger.Instance.LogInfo($"Client Joined Game With Join Code: {joinCode}");
92
93     return relayJoinData;

```

Figure 19. Code cấu hình và khởi tạo RelayUnity Server

## 6. Bảng phân chia công việc

MSSV	Họ và tên	Công việc
21520948	<u>Nguyễn Đình Kha</u>	<u>Gamelogic</u> , <u>network environment</u> <u>synchrolize</u>
21520514	<u>Nguyễn Văn Anh Tú</u>	<u>GameDesign</u> , <u>relay connection</u>
21522603	<u>Phạm Đô Thành</u>	<u>Login-register</u> , <u>bug fix</u> , <u>linking</u>
21522573	<u>Phạm Thanh Tâm</u>	<u>Playfab setting</u> , <u>connect database</u>

Figure 20. Bảng phân chia công việc

## 7. Đánh giá công việc

Tiêu chí	Mô tả	Hoàn thành hay chưa
I/O	Làm việc với các luồng nhập xuất	Đã hoàn thành
Database	Kết nối làm việc với Database	Đã hoàn thành
Thread	Áp dụng đa luồng	Đã hoàn thành
Sign up / Sign in	Đăng ký, đăng nhập, lưu trữ trạng thái	Đã hoàn thành
MultiClient	Có nhiều Client hoạt động trong mô hình	Đã hoàn thành
MultiServer	Có nhiều Server trong mô hình hoạt động	Đã hoàn thành
Cryptography	Mã hóa dữ liệu để bảo mật thông tin	Chưa hoàn thành
Demo via LAN	Demo qua LAN	Đã hoàn thành
Demo via INTERNET	Demo qua Internet	Đã hoàn thành
Load Balancing	Phân chia công việc hợp lý cho các server	Đã hoàn thành

LINK DEMO: [https://drive.google.com/file/d/16ro\\_APruB4uNBK1RDkzVJ02-pmByolB1/view?usp=sharing](https://drive.google.com/file/d/16ro_APruB4uNBK1RDkzVJ02-pmByolB1/view?usp=sharing)