

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO BÀI TẬP LỚN

Học phần: Phát triển Phần mềm hướng dịch vụ

ĐỀ TÀI: GAME ONLINE MULTIPLAYER

GIẢNG VIÊN : NGUYỄN TRỌNG KHÁNH

Họ tên sinh viên: Trần Anh Tú - B18DCCN552

Nhóm môn học: 01

Nhóm bài tập: 12

Hà Nội, tháng 5 năm 2022

MỤC LỤC

I. Phát biểu bài toán	1
II. Giới thiệu thư viện xây dựng api của Ezyfox-server	1
III. Phân tích và Thiết kế	2
IV. Cài đặt	9
V. Kết luận	13
VI. Tài liệu tham khảo	13

I. Phát biểu bài toán

Tài liệu đặc tả xây dựng hệ thống trò chơi online nhiều người chơi.

Hệ thống trò chơi xây dựng phần mềm client dựa trên mô hình MVC, xây dựng một server gồm các RESTful API và một socket server. Trò chơi phía client xây dựng trên Unity framework, lập trình bằng ngôn ngữ C#. Server sử dụng thư viện hỗ trợ xây dựng server của Ezyfox-server lập trình bằng ngôn ngữ Java kết hợp với mongoDB, gồm có:

- HTTP Server gồm các RESTful API đăng nhập, đăng ký và sử dụng dịch vụ đăng nhập bằng gmail của Google; sử dụng dịch vụ thanh toán của Paypal.

- Socket Server kết nối giao thức TCP, cho phép client kết nối và quản lý các chức năng và người chơi theo thời gian thực.

Kịch bản trò chơi: Dự án phát triển trò chơi online nhiều người chơi. Ezy Roulette là phiên bản thu gọn của trò chơi Russian Roulette. Người chơi đăng ký tài khoản sau đó đăng nhập trò chơi. Để có thể chơi, người chơi cần nạp tiền. Trong trò chơi, người chơi sẽ tham gia một vòng quay cùng với tất cả những người chơi đang chơi cùng lúc. Cách thức trò chơi hoạt động:

Vòng quay gồm có một số và màu (một trong hai màu xanh hoặc đỏ). Người chơi sau khi đăng nhập sẽ nhận thời gian đếm ngược trước khi quay từ server. Trong thời gian đó, người chơi có thể đặt cược cho màu mình muốn. Sau khi quay, có kết quả trả về, những người chơi nào đặt cược màu giống với màu kết quả sẽ được nhận thêm tiền so với tiền cược, ngược lại, người chơi sẽ mất tiền đặt cược.

II. Giới thiệu các dịch vụ sử dụng

Thư viện xây dựng Server của Ezyfox-server:

Để phát triển nhanh chóng các trò chơi trực tuyến. Ezyfox phát triển hệ sinh thái nhằm mục đích làm cho mọi thứ trở nên miễn phí và mã nguồn mở cho tất cả những ai quan tâm đến việc xây dựng các trò chơi và ứng dụng nhiều người chơi.

Hệ sinh thái EzyFox hỗ trợ một loạt các thành phần quan trọng nhất để phát triển một sản phẩm doanh nghiệp bao gồm giao thức TCP, UDP, WebSocket API với mã hóa SSL, HTTP RESTful API, Giao thức RPC gọi thủ tục từ xa, Tương tác cơ sở dữ liệu, Bộ nhớ đệm, Hàng đợi tin nhắn.

EzyFox là dịch vụ thư viện mã nguồn mở được một cộng đồng người Việt xây dựng.

HTTP Server xây dựng bằng Spring Boot:

Http Server được xây dựng gồm các API:

- Khởi tạo login token để xác thực các thao tác đăng nhập, đăng ký bằng tài khoản, mật khẩu thông thường.
- Thực hiện chức năng nạp tiền vào tài khoản

Dịch vụ đăng nhập bằng tài khoản Gmail của Google

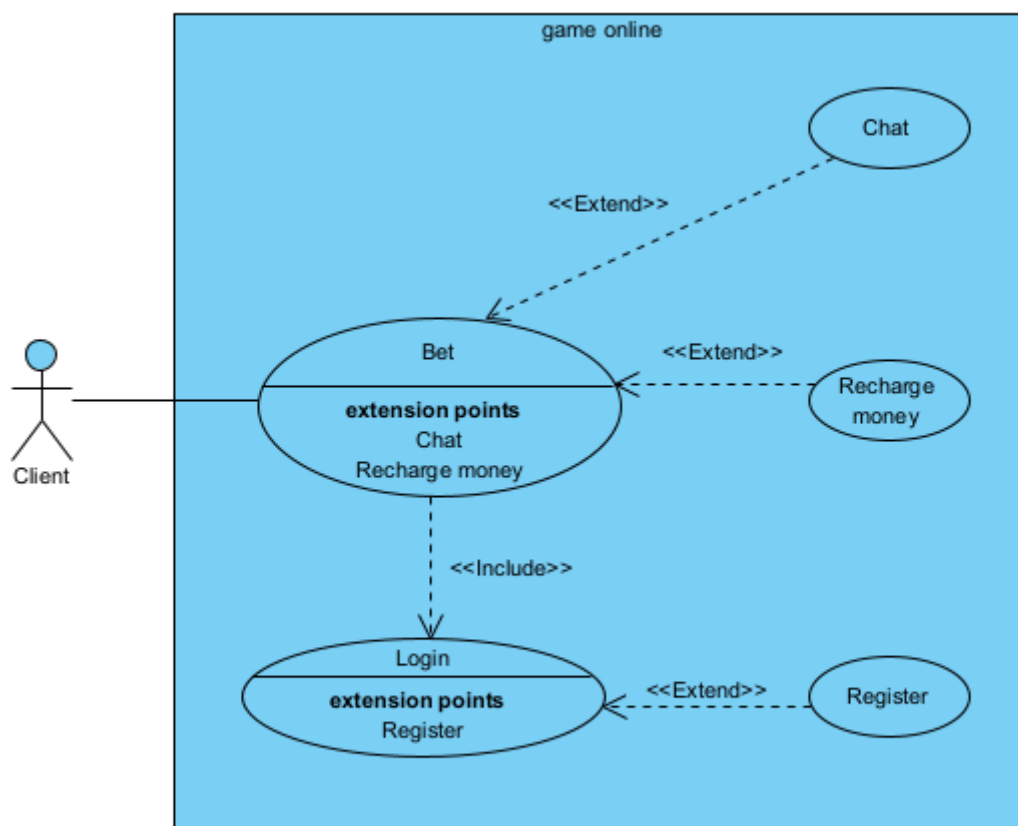
Ngày nay, Google luôn là người đi đầu cho mọi dịch vụ trên mạng. Một trong số đó là dịch vụ Google Workspace, giúp quản lý tài khoản thư điện tử Gmail và hỗ trợ tích hợp đăng nhập qua tài khoản Gmail trên các nền tảng ứng dụng khác nhau. Với Gmail, bạn có thể dễ dàng tham gia các ứng dụng một cách dễ dàng, không tốn thời gian và yên tâm với bảo mật tài khoản ở mức cao của Google.

Dịch vụ thanh toán của Paypal

Là một nền tảng hỗ trợ thanh toán lớn trên thế giới, Paypal hỗ trợ quản lý tài khoản, đáp ứng mọi thanh toán qua web, đầu đọc thẻ và hóa đơn điện tử cho mọi dịch vụ kinh doanh ở mọi lúc mọi nơi. Paypal đã xây dựng dịch vụ cho các phần mềm tích hợp thanh toán trực tuyến giúp cho mọi công việc kinh doanh trở lên dễ dàng.

III. Phân tích và Thiết kế

1. Biểu đồ Use case tổng quan



2. Mô hình chức năng

a) Chức năng đăng nhập (Login)

Cách 1:

- 1) Người chơi mở trò chơi.
- 2) Giao diện đăng nhập hiện ra với ô nhập tên đăng nhập, ô nhập mật khẩu nút login, nút login with Gmail và nút register.
- 3) Người chơi nhập tên đăng nhập và mật khẩu chọn vào nút login.
- 4) Sau khi đăng nhập thành công, giao diện trò chơi hiện ra, ngược lại sẽ thông báo lỗi.

Cách 2:

- 3) Người chơi không nhập và chọn login with Gmail
- 4) Giao diện chuyển hướng đến trình duyệt đăng nhập bằng gmail.
- 5) Người chơi đăng nhập tài khoản Gmail.

6) Trình duyệt thông báo đăng nhập thành công, mời quay trở lại trò chơi. Giao diện trò chơi hiện ra.

b) Chức năng đăng ký (Register)

- 1) Người chơi mở trò chơi.
- 2) Giao diện đăng nhập hiện ra với ô nhập tên đăng nhập, ô nhập mật khẩu nút login, nút login with Gmail và nút register.
- 3) Người chơi chọn nút đăng ký (register now).
- 4) Giao diện đăng ký hiện ra gồm 2 trường nhập tài khoản và mật khẩu.
- 5) Người dùng nhập tài khoản mật khẩu và chọn đăng ký (Sign up).
- 6) Giao diện tự động đăng nhập trò chơi nếu đăng ký thành công, ngược lại thông báo lỗi.

c) Chức năng đặt cược (Bet)

- 1) Người chơi đăng nhập thành công vào tài khoản của mình.
- 2) Một giao diện hiện ra với vòng quay, trường thông tin của tài khoản, nút nạp tiền, trường nhập mức đặt cược, ô cược và trường nhập tin nhắn.
- 3) Người chơi nhập mức cược rồi chọn ô cược.
- 4) Ô cược hiện ra tên tài khoản và mức cược.
- 5) Người dùng chờ đếm ngược.
- 6) Hết thời gian đếm ngược, vòng quay bắt đầu quay rồi trả về kết quả cho người dùng.
- 7) Sau khi nhận kết quả, người dùng có thể tiếp tục 1 vòng chơi mới.

d) Chức năng nạp tiền (Add balance)

Cách 1:

- 1) Người chơi đăng nhập thành công vào tài khoản của mình.
- 2) Một giao diện hiện ra với vòng quay, trường thông tin của tài khoản, nút nạp tiền, trường nhập mức đặt cược, ô cược và trường nhập tin nhắn.
- 3) Người chơi chọn nút nạp tiền.
- 4) Giao diện xuất hiện ô chọn nạp qua thẻ và nạp qua Paypal.
- 5) Người chơi chọn nạp qua thẻ.
- 6) Giao diện xuất hiện hai trường nhập mã của thẻ, yêu cầu người chơi mua thẻ trước đó.
- 7) Người chơi nhập hai trường rồi chọn nút nạp.
- 8) Giao diện thông báo trạng thái giao dịch và số dư tài khoản thay đổi khi thành công.

Cách 2:

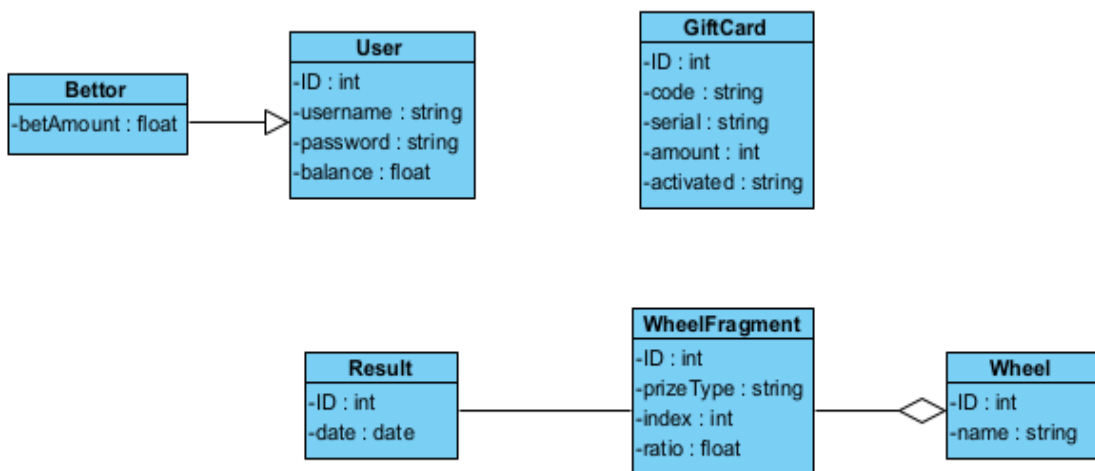
- 5) Người chơi chọn nạp qua Paypal.
- 6) Giao diện chuyển hướng đến trình duyệt web nạp.
- 7) Người chơi nhập thông tin nạp và chọn thanh toán.
- 8) Giao diện web yêu cầu đăng nhập Paypal.
- 9) Người chơi đăng nhập và chọn tiếp tục thanh toán.
- 10) Giao diện web thông báo thành công, mời quay về trò chơi.
- 11) Người chơi quay về trò chơi và thấy số dư tài khoản thay đổi.

e) Chức năng nhắn tin (Chat)

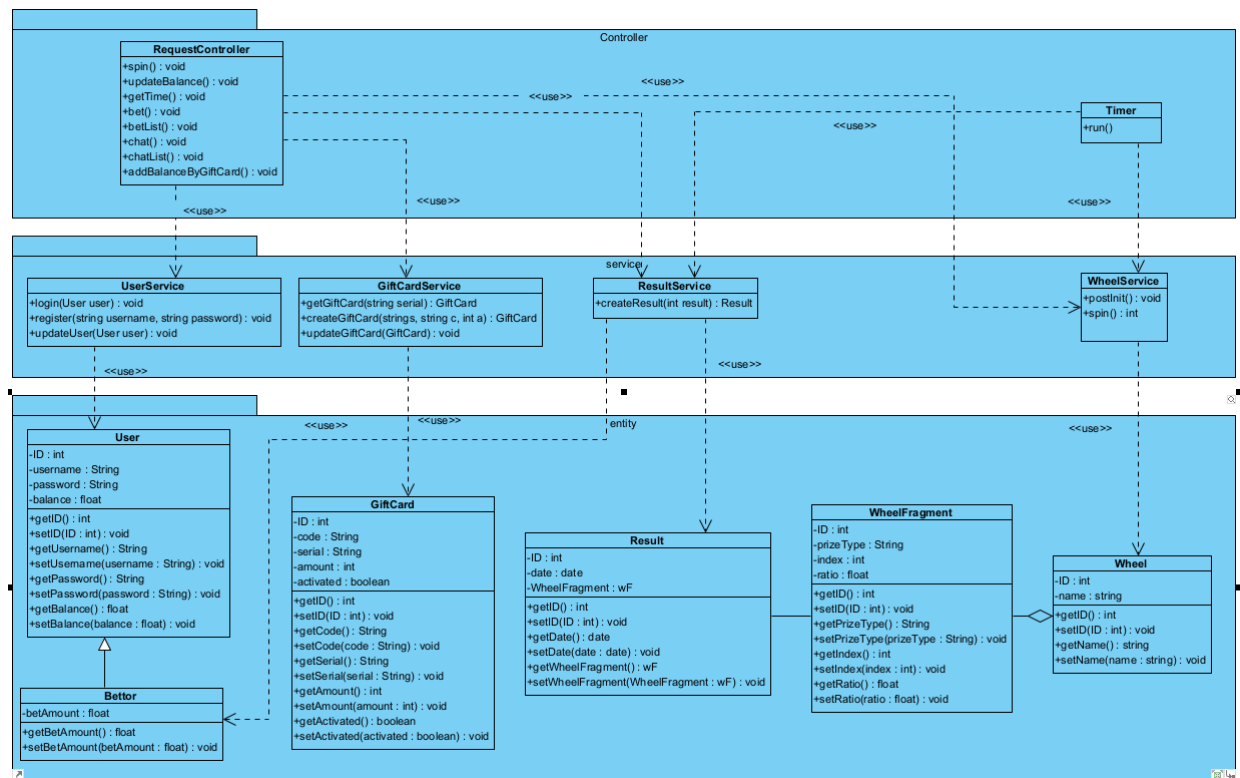
- 1) Người chơi đăng nhập thành công vào tài khoản của mình.
- 2) Một giao diện hiện ra với vòng quay, trường thông tin của tài khoản, nút nạp tiền, trường nhập mức đặt cược, ô cược và trường nhập tin nhắn.
- 3) Người chơi có thể nhắn tin với tất cả mọi người đang chơi bằng cách nhập tin nhắn rồi chọn gửi.
- 4) Giao diện hiện tin nhắn lên cho các người dùng khác đang chơi.

3. Mô hình hóa thực thể

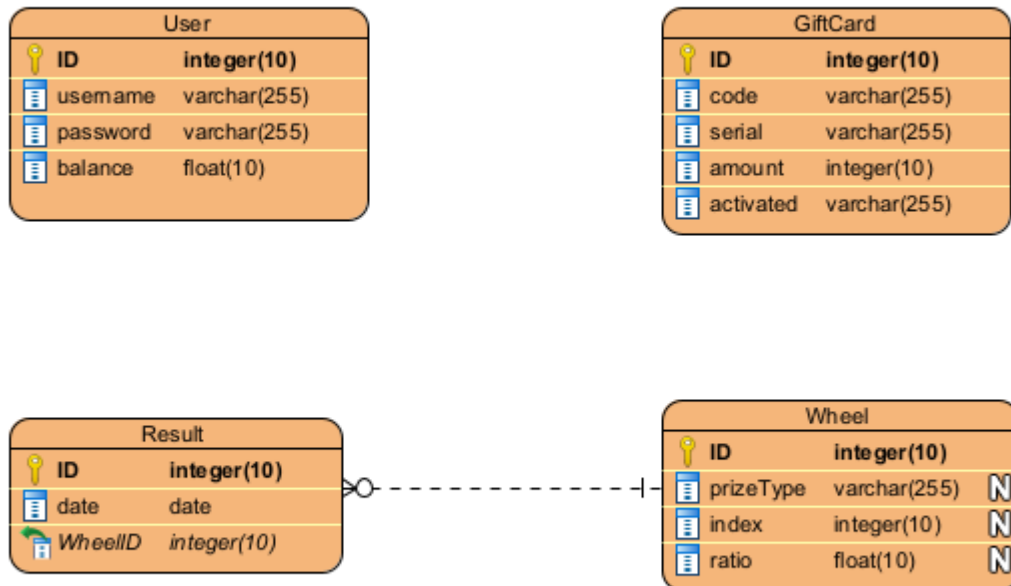
a) Sơ đồ lớp phân tích thực thể



b) Sơ đồ lớp thiết kế



4. Thiết kế CSDL



5. Kịch bản client - server

a) Chức năng đăng ký

Client	Server
Người dùng mở trò chơi	Http Server chờ yêu cầu

	Socket Server chờ kết nối
Giao diện đăng nhập/đăng ký hiện ra	Http Server chờ yêu cầu Socket Server chờ kết nối
Người dùng chọn đăng ký (Register now)	Http Server chờ yêu cầu Socket Server chờ kết nối
Giao diện đăng ký hiện ra	Http Server chờ yêu cầu Socket Server chờ kết nối
Người dùng nhập tài khoản mật khẩu và chọn đăng ký (Sign up)	Http Server chờ yêu cầu Socket Server chờ kết nối
Giao diện gửi thông tin đến Request Controller	Http Server chờ yêu cầu Socket Server chờ kết nối
Request Controller gửi api chứa yêu cầu đến Http Server	Http Server nhận yêu cầu và gửi thông tin đến UserService Socket Server chờ kết nối
Client chờ response	UserService kiểm tra và thực hiện tạo tài khoản trên database rồi gửi response là token đăng nhập Socket Server chờ kết nối
Client nhận response và thực hiện gửi token yêu cầu đăng nhập tới Socket Server	Socket Server kiểm tra token, gửi response thành công và thiết lập kết nối client nếu thành công đăng ký. Http Server chờ yêu cầu
Client nhận response, kết nối với Socket Server và giao diện hiển thị trò chơi nếu response là thành công, ngược lại thông báo lỗi.	Socket Server chờ yêu cầu Http Server chờ yêu cầu

b) Chức năng đăng nhập

Đăng nhập tài khoản thường:

Client	Server
Người dùng mở trò chơi	Http Server chờ yêu cầu Socket Server chờ kết nối
Giao diện đăng nhập/đăng ký hiện ra	Http Server chờ yêu cầu Socket Server chờ kết nối
Người dùng nhập tài khoản mật khẩu và chọn đăng nhập (Login)	Http Server chờ yêu cầu Socket Server chờ kết nối
Giao diện gửi thông tin đến Request Controller	Http Server chờ yêu cầu Socket Server chờ kết nối
Request Controller gửi api chứa yêu cầu đăng nhập đến Http Server	Http Server nhận yêu cầu và gửi thông tin đến UserService. Socket Server chờ kết nối
Client chờ response	UserService kiểm tra và gửi response là 1 token cho phép đăng nhập Socket Server chờ kết nối
Client nhận response và thực hiện gửi token với request đăng nhập tới Socket Server.	Socket Server kiểm tra token, gửi response thành công và thiết lập kết nối client nếu token chính xác. Http Server chờ yêu cầu

Client nhận response, kết nối với Socket Server và giao diện hiển thị trò chơi nếu response là thành công, ngược lại thông báo lỗi.	Socket Server chờ yêu cầu Http Server chờ yêu cầu
---	--

Đăng nhập qua Gmail Google:

Client	Server
Người dùng mở trò chơi	Http Server chờ yêu cầu Socket Server chờ kết nối
Giao diện đăng nhập/đăng ký hiện ra	Http Server chờ yêu cầu Socket Server chờ kết nối
Người dùng chọn đăng nhập qua Google	Http Server chờ yêu cầu Socket Server chờ kết nối
Giao diện mở trình duyệt đăng nhập qua Google	Http Server chờ yêu cầu Socket Server chờ kết nối
Người dùng đăng nhập vào tài khoản google	
Web View gửi request Http server và hiển thị thông báo đăng nhập thành công mời người dùng quay lại trò chơi	Http Server nhận yêu cầu và lưu login token của người dùng vào hàng đợi đăng nhập Socket Server chờ kết nối
Client gửi request lấy login token	Http Server kiểm tra và gửi response là 1 token cho phép đăng nhập Socket Server chờ kết nối
Client nhận response và thực hiện gửi token với request đăng nhập tới Socket Server.	Socket Server kiểm tra token, gửi response thành công và thiết lập kết nối client nếu token chính xác. Http Server chờ yêu cầu
Client nhận response, kết nối với Socket Server và giao diện hiển thị trò chơi nếu response là thành công, ngược lại thông báo lỗi.	Socket Server chờ yêu cầu Http Server chờ yêu cầu

c) Chức năng đặt cược

Client	Socket Server
Người dùng đăng nhập thành công	Server thiết lập kết nối thành công với client
Giao diện trò chơi hiện ra	Server chờ yêu cầu
Request Controller gửi api chứa yêu cầu nhận thời gian đến server	Server nhận yêu cầu và gửi thông tin đến WheelService
Client chờ response	WheelService kiểm tra và thực hiện rồi gửi response
Client nhận response thời gian và hiển thị lên giao diện	Server chờ yêu cầu
Người chơi nhập mức cược rồi chọn ô cược	Server chờ yêu cầu

Giao diện gửi thông tin đến Request Controller	Server chờ yêu cầu
Request Controller gửi api chứa yêu cầu đến server	Server nhận yêu cầu và gửi thông tin đến UserService
Client chờ response	UserService kiểm tra và thực hiện rồi gửi thông tin đến ResultService
Client chờ response	ResultService kiểm tra và thực hiện rồi gửi response
Client nhận response và hiển thị lên giao diện	Server chờ yêu cầu

d) Chức năng nạp tiền

Nạp tiền bằng thẻ:

Client	Http Server
Người dùng đang trong giao diện trò chơi chọn nạp tiền	Server chờ yêu cầu
Giao diện bảng chọn loại nạp	Server chờ yêu cầu
Người chơi chọn nạp tiền bằng thẻ	Server chờ yêu cầu
Giao diện nhập mã nạp hiện ra	Server chờ yêu cầu
Người dùng nhập mã thẻ rồi chọn nạp	Server chờ yêu cầu
Giao diện gửi thông tin đến Request Controller	Server chờ yêu cầu
Request Controller gửi api chứa yêu cầu đến server	Server nhận yêu cầu và gửi thông tin đến GiftCardService
Client chờ response	GiftCardService kiểm tra và thực hiện rồi gửi thông tin đến UserService
Client chờ response	UserService kiểm tra và thực hiện rồi gửi response
Client nhận response và số dư tài khoản thay đổi hiển thị lên giao diện	Server chờ yêu cầu

Nạp tiền qua Paypal:

Client	Http Server
Người dùng đang trong giao diện trò chơi chọn nạp tiền	Server chờ yêu cầu
Giao diện bảng chọn loại nạp	Server chờ yêu cầu
Người chơi chọn nạp tiền qua Paypal	Server chờ yêu cầu
Giao diện chuyển sang trình duyệt với giao diện nạp	Server chờ yêu cầu
Người dùng điền các trường và ấn nút checkout. Web view gửi yêu cầu nạp tiền đến Http server	Http server tiếp nhận request và chuyển hướng trang web đến trang đăng nhập Paypal
Người dùng thực hiện đăng nhập vào tài khoản Paypal và tiến hành thanh toán.	Server chờ yêu cầu

Web View gửi request đến Http server và chuyển hướng người dùng đến trang thanh toán thành công.	Http server tiếp nhận request và gọi UserService tiến hành nạp tiền cho người dùng.
--	---

e) Chức năng nhắn tin

Client	Socket Server
Người dùng đăng nhập thành công	Server thiết lập kết nối thành công với client
Giao diện trò chơi hiện ra	Server chờ yêu cầu
Người chơi chọn trường nhập tin nhắn, rồi chọn gửi (Send)	Server chờ yêu cầu
Giao diện gửi thông tin đến Request Controller	Server chờ yêu cầu
Request Controller gửi api chứa yêu cầu đăng nhập đến server	Server nhận yêu cầu và gửi thông tin đến WheelService
Client chờ response	WheelService thực hiện kết nối tin nhắn gửi response cho các client đang chơi
Client nhận response và hiển thị tin nhắn lên màn hình	Server chờ yêu cầu

IV. Cài đặt

Bài toán xây dựng trò chơi online multiplayer đặt ra một số vấn đề như là độ tin cậy, thời gian thực, xây dựng api,.... EzyFox hỗ trợ thư viện lập trình giúp xử lý các vấn đề trên một cách dễ dàng.

Các RESTful API được xây dựng trên Http Server:

Tên API	Http request	Mô tả
login()		Cho phép người dùng đăng nhập bằng cách gửi một token cho phép đăng nhập
loginWithGoogle()		Cho phép người dùng đăng nhập nhanh qua tài khoản của Gmail của Google
register()		Cho phép người dùng đăng ký tài khoản và nhận lại 1 token cho phép đăng nhập
addBalanceByGiftCard (User, Request)		Cho phép người dùng nạp thêm tiền qua thẻ quà tặng có sẵn
addBalanceByPaypal()		Cho phép người dùng nạp thêm tiền qua dịch vụ thanh toán của Paypal

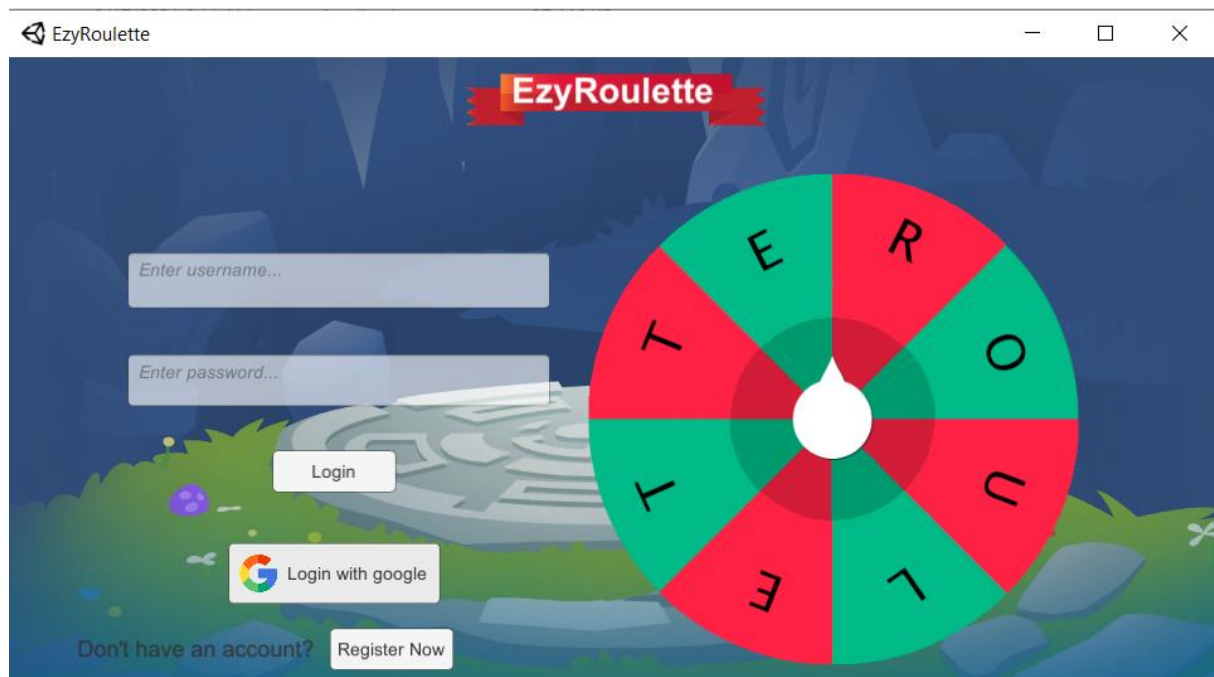
Sau khi kết nối đến Socket Server, các request được gọi thông qua các câu lệnh đơn giản (Command):

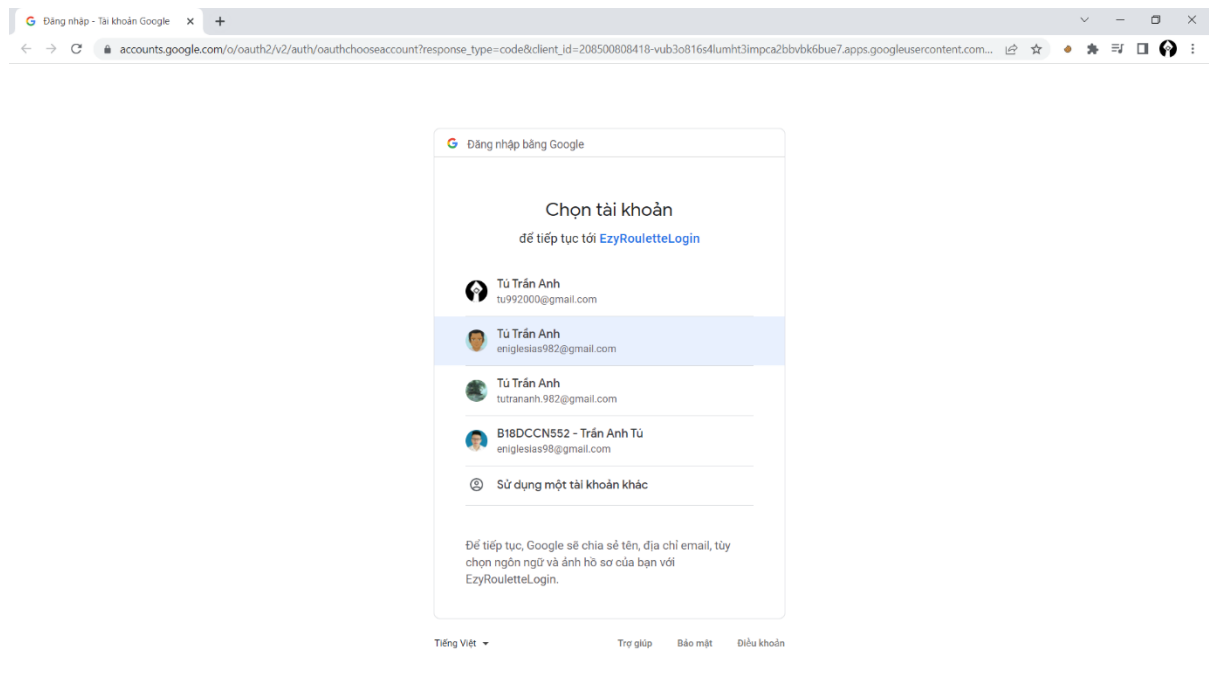
Tên API	Command	Mô tả
---------	---------	-------

getTime(User)	“getTime”	Lấy thời gian đếm ngược của mỗi vòng quay
spin(User)	“spin”	Lấy kết quả vòng quay vừa kết thúc
updateBalance(User)	“updateBalance”	Cập nhật số dư tài khoản của người dùng
bet(User, Request)	“bet”	Thực thi thao tác đặt cược của người dùng
betList(User)	“betList”	Cho phép lấy danh sách người dùng đã đặt cược
chat(User, Request)	“chat”	Thực thi thao tác nhắn tin của người dùng
chatList(User)	“chatList”	Cho phép lấy thông tin chat của những người dùng chat để hiện lên giao diện

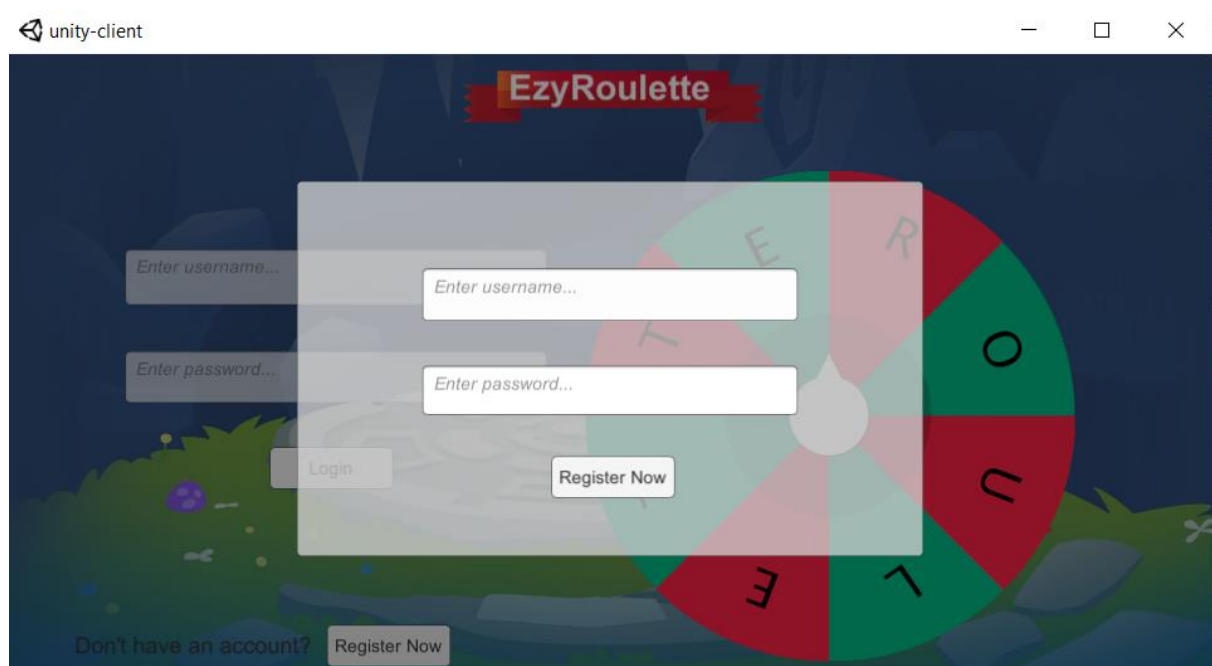
Giao diện:

- Giao diện đăng nhập:

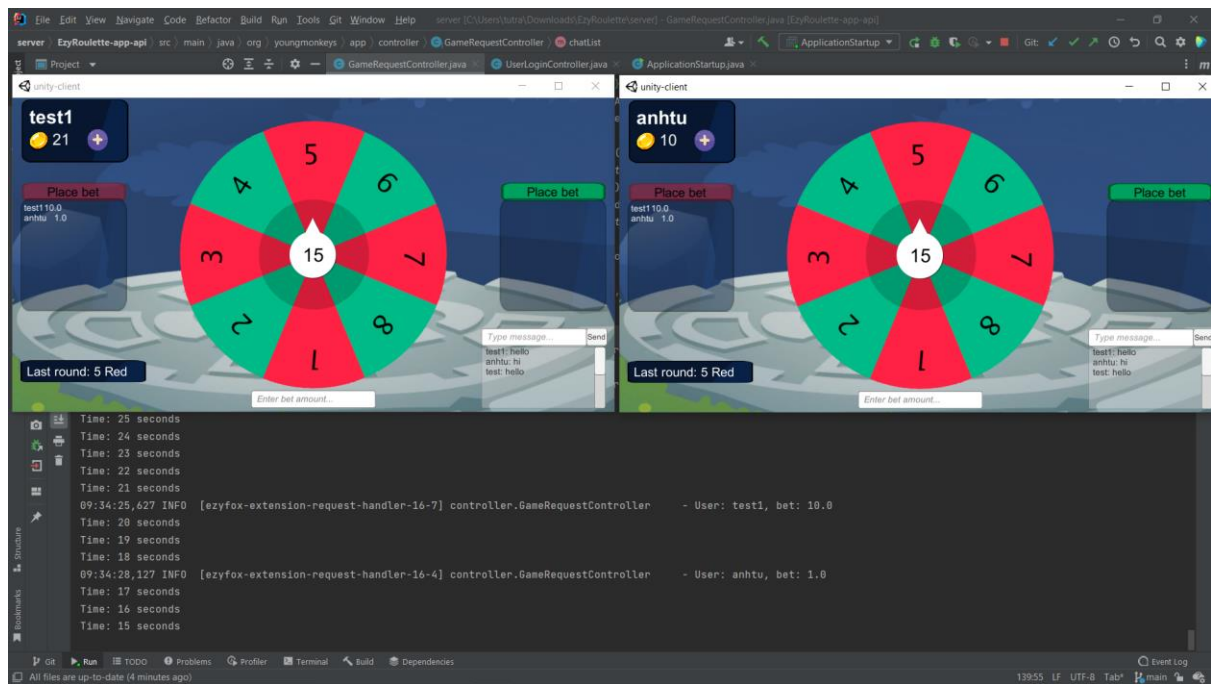




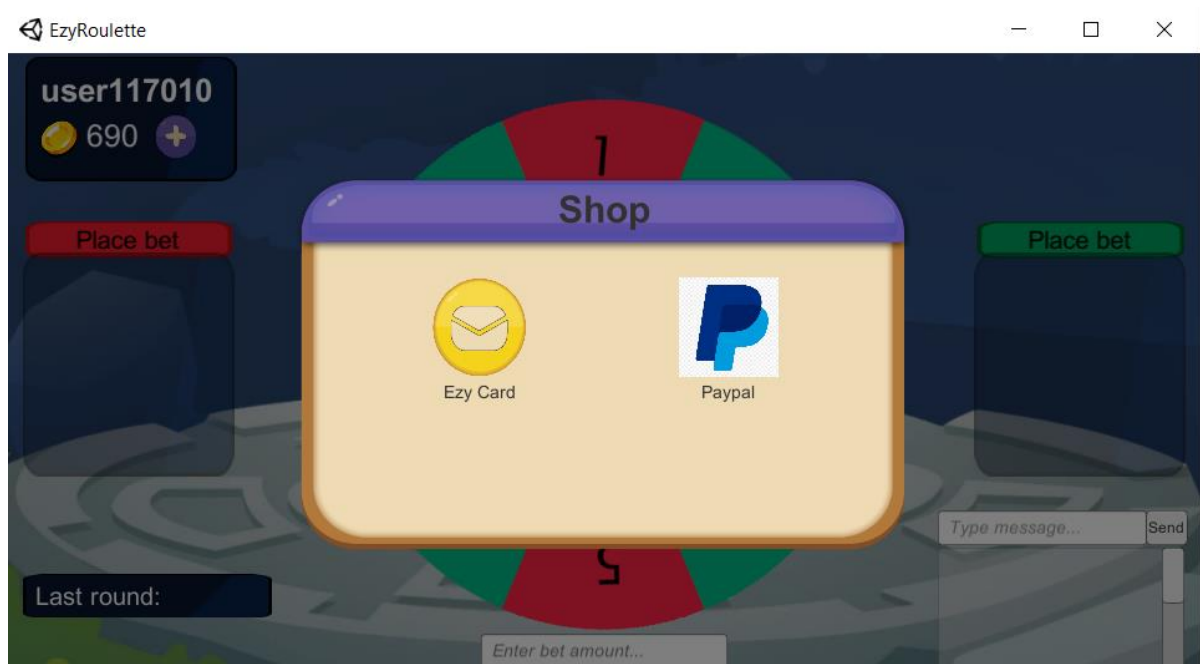
- Giao diện đăng ký:

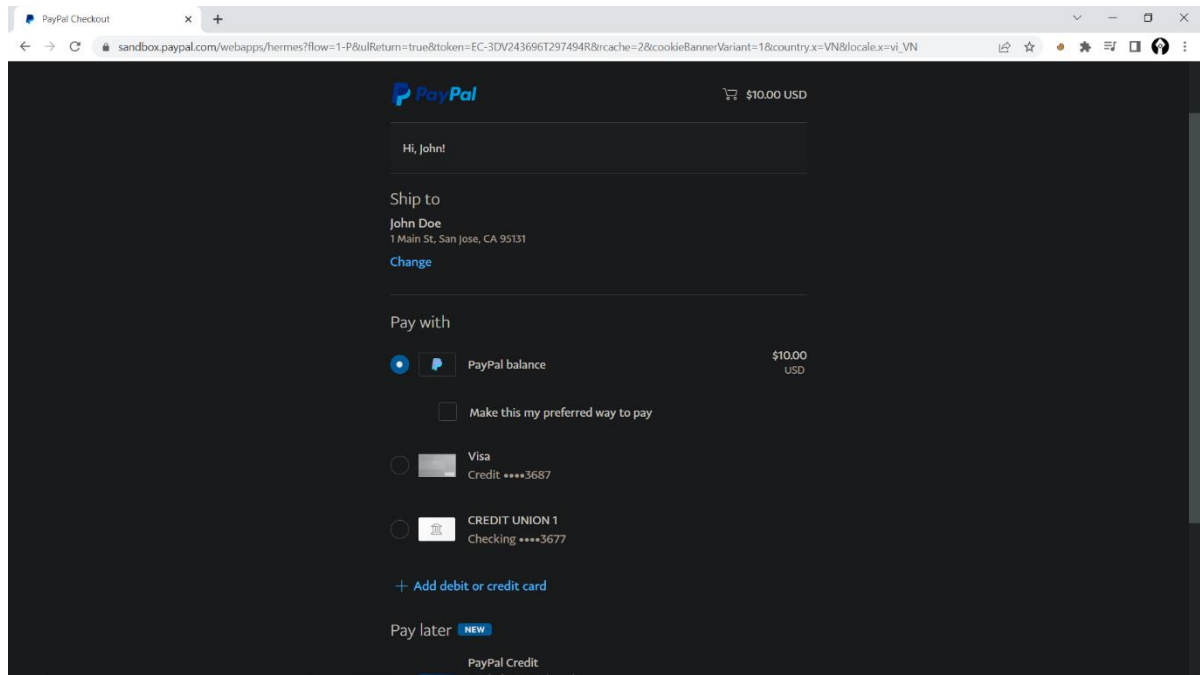


- Giao diện trò chơi (chat, đặt cược):



- Giao diện nạp xu:





V. Kết luận

Sau quá trình học tập và tìm hiểu về lập trình hướng dịch vụ. Hiện tại nhóm đã hoàn thành ứng dụng game online multiplayer với các chức năng cơ bản. Sản phẩm là một hệ thống client-server giao tiếp với nhau qua các API và socket, có sử dụng dịch vụ tài khoản của Google và dịch vụ thanh toán của Paypal.

Hạn chế: Trong quá trình vừa học vừa làm project nên nhóm không thể tránh được những sai sót như code dài dòng, thừa dữ liệu. Một vài case có thể còn lỗi không thông báo cho người dùng khi thao tác thành công hoặc thất bại, chưa thể hiện được như thiết kế.

Trò chơi xây dựng để học tập, không có mục đích khác. Hy vọng nhận được sự đóng góp để nhóm có thể xây dựng bài toán tốt hơn.

VI. Tài liệu tham khảo

- [1] Youngmonkey Team, "EzyFox," Youngmonkey, 2020. [Online]. Available: <https://youngmonkeys.org/project/ezyfox-sever/>.
- [2] Nhóm 12, "Github," 5 2022. [Online]. Available: <https://github.com/tutrananh/EzyRoulette/>.
- [3] Paypal, "REST APIs," [Online]. Available: <https://developer.paypal.com/api/rest/>.
- [4] Tutorialspoint, "Google OAuth2 Sign-In," [Online]. Available: https://www.tutorialspoint.com/spring_boot/spring_boot_google_oauth2_sign_in.htm.