

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Ứng dụng NFT vào game Flappy Bird

NGUYỄN TIẾN VƯỢNG

vuong.nt184022@sis.hust.edu.vn

Ngành Công nghệ thông tin và truyền thông

Giảng viên hướng dẫn: TS. TRỊNH ANH PHÚC

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 06/2022

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn sâu sắc đến TS.Trịnh Anh Phúc . Thầy là người đã hướng dẫn, giúp đỡ em rất nhiều trong quá trình nghiên cứu, thực hiện và hoàn thiện ĐATN này.

Em cũng xin cảm ơn tới tất cả thầy cô đã chỉ dạy cho em những kiến thức quý báu trong suốt quá trình học tập, cung cấp một hành trang thiết thực cho tương lai sắp tới.

Cuối cùng, em xin cảm ơn gia đình, bạn bè, đồng nghiệp - những người luôn ở bên cạnh, cổ vũ và động viên tinh thần, giúp em vượt qua những khó khăn trong suốt quá trình học tập tại đại học Bách Khoa Hà Nội.

Do thời gian có hạn, cũng như kiến thức chuyên môn còn non kém, ĐATN của em sẽ không tránh khỏi những thiếu sót và hạn chế. Em rất mong nhận được những đánh giá, nhận xét từ thầy cô và các bạn, để từ đây nhận thấy những ưu, nhược điểm của bản thân, đúc kết và xây dựng kinh nghiệm làm việc một cách vững vàng hơn.

Em xin chân thành cảm ơn!

TÓM TẮT NỘI DUNG ĐỒ ÁN

Hiện nay, công nghệ blockchain trở thành xu hướng của toàn cầu và cơn sốt mang tên game NFT nổi lên trong giới đầu tư tiền ảo. Game NFT là dòng game được xây dựng và phát triển trên nền tảng công nghệ blockchain token hóa tài sản trong game. Mỗi game sử dụng đồng tiền điện tử riêng và chúng có thể quy đổi thành tiền thật thông qua sàn giao dịch điện tử. Thể loại game này cho phép người chơi có thu nhập thực tế dưới dạng non-fungible token (NFT). Token trong NFT game có tác dụng mua bán vật phẩm game, staking vật phẩm/token thưởng, nâng cấp trải nghiệm game, cùng nhiều tiện ích khác. Nguyên nhân quan trọng nhất làm cho game NFT phát triển là do ảnh hưởng của COVID-19, hàng triệu người thất nghiệp, và chuyển sang hình thức kiếm tiền online. Game play-to-earn đáp ứng được nhu cầu này, hình thức vừa chơi vừa kiếm tiền đang được rất nhiều người trên thế giới quan tâm.

Nắm bắt xu hướng toàn cầu, đồ án tốt nghiệp này hướng tới xây dựng một ứng dụng game NFT với kịch bản là Flappy bird quen thuộc. Mục đích của đồ án nhằm nghiên cứu cách module hóa cho game NFT và nghiên cứu nền kinh tế thị trường của game. Tạo cơ sở cho việc về sau phát triển những tựa game lôi cuốn, trải nghiệm thú vị để có thể thu hút được cộng đồng game thủ toàn cầu.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	3
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	5
2.1 Khảo sát hiện trạng	5
2.2 Tổng quan chức năng	5
2.2.1 Biểu đồ use case tổng quát	5
2.2.2 Biểu đồ use case phân rã chức năng Quản lý người dùng	6
2.2.3 Biểu đồ usecase phân rã chức năng Quản lý tiền người dùng	7
2.2.4 Biểu đồ usecase phân rã chức năng Quản lý thanh khoản	7
2.2.5 Biểu đồ usecase phân rã chức năng Quản lý Bird	8
2.2.6 Biểu đồ usecase phân rã chức năng quản lý chợ	8
2.2.7 Quy trình nghiệp vụ	9
2.3 Đặc tả chức năng	10
2.3.1 Đặc tả use case Thêm ví on-chain.....	10
2.3.2 Đặc tả use case Mint Box.....	11
2.3.3 Đặc tả use case mua Bird.....	12
2.3.4 Đặc tả use case Chơi game.....	13
2.3.5 Đặc tả use case Mở Box.....	14
2.4 Yêu cầu phi chức năng	15

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	16
3.1 Blockchain.....	16
3.1.1 Giao dịch [10].....	16
3.1.2 Khối [11]	16
3.1.3 NFT [12].....	17
3.1.4 Token [13].....	17
3.1.5 Smart Contract [14].....	17
3.2 Platform NodeJs và Framework Express	19
3.3 MongoDB [19].....	20
3.4 React Native [21].....	20
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ	22
4.1 Thiết kế kiến trúc.....	22
4.1.1 Lựa chọn kiến trúc phần mềm	22
4.1.2 Thiết kế tổng quan.....	23
4.1.3 Thiết kế chi tiết gói	23
4.2 Thiết kế chi tiết.....	24
4.2.1 Thiết kế giao diện	24
4.2.2 Thiết kế lớp	28
4.2.3 Thiết kế cơ sở dữ liệu	31
4.3 Xây dựng ứng dụng.....	38
4.3.1 Thư viện và công cụ sử dụng	38
4.3.2 Kết quả đạt được	39
4.3.3 Minh họa các chức năng chính	39
4.4 Kiểm thử.....	44
4.4.1 Kiểm thử chức năng Thêm ví on-chain.....	44
4.4.2 Kiểm thử chức năng Mint BirdBox	45

4.4.3 Kiểm thử chức năng Chuyển hợp từ ví on-chain vào ví in-game.....	45
4.4.4 Kiểm thử chức năng Bán Bird trên chợ.....	45
4.4.5 Kiểm thử chức năng Hoán đổi token sang loại tiền điện tử khác	46
4.5 Triển khai	46
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT.....	47
5.1 Smart Contract	47
5.2 Tạo FBT token	47
5.3 NFT hoá Box	47
5.4 Socket.io	48
5.5 Tạo ví in-app	48
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	49
6.1 Kết luận.....	49
6.2 Hướng phát triển.....	49
TÀI LIỆU THAM KHẢO.....	51

DANH MỤC HÌNH VẼ

Hình 1.1	Mô hình dự án	2
Hình 2.1	Biểu đồ usecase tổng quan	6
Hình 2.2	Usecase phân rã chức năng Quản lý người dùng	7
Hình 2.3	Usecase phân rã chức năng Quản lý tiền người dùng	7
Hình 2.4	Usecase phân rã chức năng quản lý thanh khoản	8
Hình 2.5	Usecase phân rã chức năng quản lý Bird	8
Hình 2.6	Usecase phân rã chức năng quản lý chợ	9
Hình 2.7	Quy trình nghiệp vụ	10
Hình 3.1	Cấu trúc của một Blockchain	16
Hình 3.2	Cấu trúc của Smart Contract	18
Hình 4.1	Mô hình áp dụng trong ứng dụng	22
Hình 4.2	Biểu đồ phụ thuộc gói trong ứng dụng	23
Hình 4.3	Thiết kế chi tiết gói	24
Hình 4.4	Thiết kế thành phần giao diện màn Thêm ví on-chain	25
Hình 4.5	Thiết kế thành phần giao diện màn Ví on-chain	26
Hình 4.6	Thiết kế thành phần giao diện màn Mint Box	27
Hình 4.7	Thiết kế thành phần giao diện màn Chợ	28
Hình 4.8	Thiết kế các lớp chính tham gia chức năng Thêm ví on-chain	29
Hình 4.9	Thiết kế các lớp chính tham gia chức năng Mua Bird	29
Hình 4.10	Thiết kế các lớp chính tham gia chức năng Chơi game	30
Hình 4.11	Thiết kế các lớp chính tham gia chức năng Mở BirdBox	31
Hình 4.12	Thiết kế cơ sở dữ liệu	32
Hình 4.13	Giao diện bán Bird	40
Hình 4.14	Giao diện Mở hộp	41
Hình 4.15	Thêm ví on-chain	42
Hình 4.16	Giao diện Mint Box	43
Hình 4.17	Giao diện chơi game	44

DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case Thêm ví on-chain	11
Bảng 2.2	Đặc tả use case Mint Box	12
Bảng 2.3	Đặc tả use case Mua Bird	13
Bảng 2.4	Đặc tả use case Chơi game	14
Bảng 2.5	Đặc tả use case mở Box	15
Bảng 4.1	Thiết kế chi tiết lớp UserController	29
Bảng 4.2	Thiết kế chi tiết phương thức cho use case Mua Bird	29
Bảng 4.3	Thiết kế chi tiết phương thức cho use case Chơi game	30
Bảng 4.4	Thiết kế chi tiết phương thức cho use case Mở BirdBox	31
Bảng 4.5	Bảng User	33
Bảng 4.6	Bảng Wallet	34
Bảng 4.7	Bảng Bird	35
Bảng 4.8	Bảng BirdBox	36
Bảng 4.9	Bảng TranferTransaction	37
Bảng 4.10	Bảng BirdBoxTransaction	38
Bảng 4.11	Danh sách thư viện và công cụ sử dụng	38
Bảng 4.12	Danh sách thư viện sử dụng trong game NFT Flappy Bird	39
Bảng 4.13	Kiểm thử chức năng Thêm ví on-chain	44
Bảng 4.14	Kiểm thử chức năng Mint BirdBox	45
Bảng 4.15	Kiểm thử chức năng Chuyển hộp từ ví on-chain vào ví in-game	45
Bảng 4.16	Kiểm thử chức năng Bán Bird trên chợ	45
Bảng 4.17	Kiểm thử chức năng Hoán đổi token sang loại tiền điện tử khác	46

DANH MỤC TỪ VIẾT TẮT

Viết tắt	Tên tiếng Anh	Tên tiếng Việt
API	Application Programming Inter- face	Giao diện lập trình ứng dụng
FBT	Flappy Bird token	Tên một loại tiền điện tử
EVM	Ethereum Virtual Machine	Máy ảo của Ethereum
JSX	JavaScript XML	Một cú pháp mở rộng
IDO	Initial DEX Offering	Phân phối coin lần đầu trên sàn phi tập trung
NFT	Non-fungible token	Tài sản không thể thay thế
ĐATN		Đồ án tốt nghiệp

DANH MỤC THUẬT NGỮ

Thuật ngữ	Ý nghĩa
Bird	Vật phẩm chim trong game
BirdBox	Vật phẩm hộp trong game
Mint Box	Tạo ra một hộp đựng Bird
Transfer	Chuyển token vào ví trong game
Claim	Chuyển token ra ví ngoài
NFT	Một loại tài sản số hiện diện trên một chuỗi số
Token	Tiền điện tử
Crypto	Một loại tiền điện tử
GameFi	Sự kết hợp giữa Game và tài chính
Ví on-chain	Mọi loại ví lưu trữ tiền điện tử trên Blockchain
Ví in-game	Một loại ví lưu trữ tiền điện tử trong Game
Labor to Earn	Các mô hình kiếm tiền
Learn to Earn Participate to Earn	Học tập và tham dự để kiếm tiền
Create to Earn	Tạo sản phẩm để kiếm tiền
Play to Earn hay GameFi	Chơi game để kiếm tiền
Stacking	Hành động khoá một lượng coin nhất định để được phần thưởng từ chúng.
Real-time	Thời gian thực
Decentralized, fully-decentralized	Phi tập trung
Semi-decentralized	Nửa tập trung, nửa không tập trung
Seed phrase	Một tập hợp các từ khoá dùng để truy cập ví tiền điện tử
Approve	Chấp thuận cho mint hộp
Swap	Trao đổi tiền điện tử
Pool IDO	Thanh khoản trên sàn phi tập trung khi mới mở bán

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

Trong quá trình phát triển và xây dựng phần mềm, bước khởi đầu sẽ là xác định được nhưng vấn đề, xác định được mục tiêu đề tài và định hướng phát triển của dự án. Chương 1 sẽ làm rõ các yêu cầu đặt ra cũng như các tính năng sẽ xây dựng trong ứng dụng.

1.1 Đặt vấn đề

Khi công nghệ tự động hóa và trí tuệ nhân tạo ngày càng phát triển, nhiều người lo lắng về tương lai việc làm. Nếu hàng triệu người lao động mất việc, họ sẽ làm gì để nuôi sống bản thân và gia đình?

Đây không phải là nỗi lo vô căn cứ. Theo một báo cáo từ PricewaterhouseCoopers - một trong bốn công ty kiểm toán hàng đầu thế giới hiện nay, vào giữa 2030, 1/3 công việc của chúng ta có nguy cơ bị máy móc thay thế. Những lao động tay nghề hạn chế sẽ là lực lượng dễ bị đẩy ra bên lề xã hội nhất. Nỗi lo mất việc do máy móc lên ngôi đã hiện diện trong lịch sử loài người từ hàng trăm năm trước. Trong mỗi bước ngoặt của xã hội, nhiều người lại đối mặt với sự thay đổi việc làm không cưỡng lại được.

Trước nỗi lo mất việc của người lao động do quá trình tự động hóa, các nhà kinh tế học nói rằng không cần phải lo lắng. Công nghệ loại bỏ việc làm cũ nhưng cũng sẽ tạo ra việc làm mới, mặc dù chúng ta phải trải qua một quá trình điều chỉnh đau đớn. Điều này đã được chứng minh qua các cuộc Cách mạng công nghiệp trong lịch sử.

Mặc dù đại dịch COVID-19 tước đi công việc của nhiều người, nhưng cũng đã đẩy nhanh làn sóng phát triển công nghệ, thúc đẩy quá trình chuyển đổi xã hội sang thời đại của cuộc Cách mạng công nghiệp 4.0. Trong làn sóng đổi mới này, công nghệ blockchain trong nền kinh tế crypto đã giới thiệu những hình thức công việc hoàn toàn mới.

“Các nền kinh tế crypto đã bắt đầu định hình tương lai việc làm của chúng ta. Chúng xóa nhòa ranh giới giữa việc chơi, học, tương tác và sáng tạo bằng cách cung cấp quyền sở hữu và tạo ra thu nhập,” - Stephen McKeon, Giáo sư tại Đại học Oregon nói. [1]

Edward Castronova - giáo sư truyền thông tại Đại học Indiana Bloomington, người nổi tiếng với công trình nghiên cứu về nền kinh tế của các thế giới tổng hợp, đã đưa ra hai dự đoán trong bài viết: “Thuê người chơi game: Gaming và tương lai của công việc kỹ năng thấp”:

Trong vòng 5 năm tới, một số công ty gaming sẽ trả tiền để người chơi chơi game của họ.

Trong vòng 10 năm tới, trả tiền cho người chơi sẽ trở thành mô hình doanh thu tiêu chuẩn trong ngành công nghiệp gaming. Các khoản thanh toán sẽ nằm dưới dạng crypto hoặc tiền mặt [2].

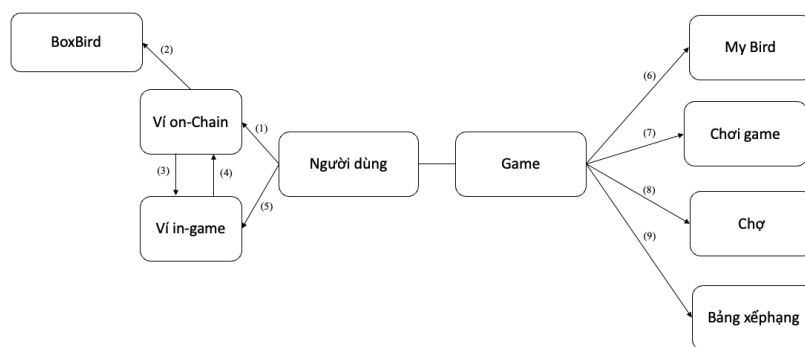
Không thể phủ nhận rằng GameFi đang là một đề xuất hấp dẫn dành cho những người thất nghiệp hoặc lao động tay nghề thấp. Đặc biệt, đối với giới trẻ, đây là một trong số ít lĩnh vực mà họ có lợi thế so với những người lớn tuổi nhiều kinh nghiệm nhờ vào kỹ năng sử dụng công nghệ và quỹ thời gian tương đối dồi dào của họ.

Hơn nữa, GameFi cho phép mọi người chuyển đổi thời gian lãng phí thành thu nhập một cách tương đối dễ dàng. Giờ đây, mọi người có thể dùng thời gian chờ xe buýt hoặc xếp hàng tại ngân hàng để kiếm thêm một ít tiền. Thay vì lướt TikTok hoặc xem YouTube, mọi người có thể chơi game để kiếm tiền.

Từ những nhu cầu thực tế kể trên, em đã thực hiện đề tài "Ứng dụng NFT vào game Flappy bird". Mục đích của đề tài là để nghiên cứu cách xây dựng một GameFi và nghiên cứu nền kinh tế trong game NFT.

1.2 Mục tiêu và phạm vi đề tài

Với những vấn đề đã trình bày trong phần 1.1, em đặt ra mục tiêu là xây dựng một ứng dụng game chạy trên cả hai hệ điều hành là Android và iOS. Game sẽ giúp người dùng giải trí đồng thời kiếm tiền thông qua việc chơi game hay là đầu tư token hoặc là mua bán Bird trong game.



Hình 1.1: Mô hình dự án

Hình trên thể hiện mô hình của đồ án. Để chơi game, người dùng cần sở hữu đồng token FBT để mua BirdBox. Ứng dụng cung cấp chức năng quản lý ví của người dùng. Người dùng có thể xem trực tiếp ví ngay trong ứng dụng. Các chức năng chính của ứng dụng là:

- (1) : Thêm ví on-Chain: Người dùng có thể thêm được ví MetaMask vào game và sử dụng tiền để mua BirdBox
- (2) : Mint Box, xem BirdBox: Người dùng sẽ dùng tiền trong ví để tạo ra BirdBox mình muốn
- (3) : Chuyển BirdBox và token vào ví in-game
- (4) : Claim token: Người dùng sẽ chuyển tiền từ trong game ra ví on-chain
- (5) : Tạo ví in-game
- (6) : Xem Bird, mở Box, huỷ bán
- (7) : Chơi game, trả thưởng
- (8) : Mua, bán Bird
- (9) : Xem bảng xếp hạng

1.3 Định hướng giải pháp

Sản phẩm sẽ có các chức năng như: (i) Tạo ra token FBT (Flappy Bird Token - đồng tiền lưu hành trong game), (ii) xử lý hoạt động trả thưởng khi chơi game, (iii) chuyển tiền khi mua bán trên chợ, (iv) triển khai dự án trên Binance Smart Chain Testnet... Với mục tiêu đáp ứng nhu cầu của người chơi, ứng dụng sẽ cố gắng tối đa các tính năng cần có cũng như tương thích với các thiết bị. Tuy nhiên, do thời gian của đồ án có hạn nên ứng dụng sẽ được triển khai trên nền tảng mobile trước, sau này có thời gian sẽ phát triển và mở rộng trên nền tảng website.

Mã nguồn của ứng dụng sẽ chia làm ba phần: smart contract - xử lý các nghiệp vụ có kết nối đến blockchain, backend – xử lý các nghiệp vụ logic, kết nối với dữ liệu để thực hiện các tính năng; và frontend – hiển thị dữ liệu, tương tác với người dùng.

Phần smart contract sẽ được xây dựng trên nền tảng Ethereum, sử dụng ngôn ngữ lập trình Solidity [3]. Phần backend sẽ được xây dựng trên NodeJs, sử dụng ngôn ngữ lập trình JavaScript [4]. Với ưu điểm nhanh hơn và nhẹ hơn của JavaScript so với các ngôn ngữ lập trình khác, ứng dụng xây dựng sẽ đảm bảo một hiệu năng tốt nhất cho người dùng. Cơ sở dữ liệu sẽ được lưu trữ bằng cơ sở dữ liệu MongoDB. Phần frontend sử dụng thư viện React Native. Ứng dụng có thể chạy trên cả iOS và Android.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về khảo sát hiện trạng đề tài, tổng quan các chức năng trong

hệ thống đồng thời đặc tả một số use case chính.

Trong Chương 3, em sẽ trình bày về các công nghệ và công cụ được sử dụng trong đề án.

Chương 4 sẽ trình bày chi tiết về phân tích thiết kế hệ thống và trình bày sâu hơn về luồng hoạt động của một số nghiệp vụ chính trong hệ thống, thiết kế giao diện, cơ sở dữ liệu và cuối cùng là triển khai hệ thống.

Trong Chương 5 em sẽ trình bày nội dung về các giải pháp và đóng góp nổi bật nhất cũng như cách giải quyết các khó khăn trong quá trình thực hiện đề án.

Trong Chương 6 em sẽ kết luận đề án, sau đó trình bày các ưu nhược điểm của hệ thống, những gì em đã và chưa làm được trong đề án tốt nghiệp lần này. Đồng thời, em sẽ đề xuất các định hướng mà em muốn phát triển hệ thống trong tương lai.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

Với những yêu cầu đã đặt ra ở chương 1, chương 2 sẽ khảo sát hiện trạng thực tế, làm rõ các bước phân tích, yêu cầu phi chức năng của ứng dụng.

2.1 Khảo sát hiện trạng

Hiện nay những mô hình Labor to Earn mới bao gồm: (i) Learn to Earn & Participate to Earn [5], (ii) Create to Earn, (iii) Play to Earn [6] hay GameFi [7]. Tiêu biểu nhất cho hiện tượng GameFi này chính là Axie Infinity [8].

Axie Infinity là một trò chơi dựa trên blockchain, trong đó người chơi mua những con quái vật dễ thương gọi là Axies và đem chúng đi chiến đấu. Người chơi giành chiến thắng trong các trận đấu này hoặc hoàn thành nhiệm vụ hàng ngày sẽ kiếm được một loại tiền tệ trong game, gọi là Smooth Love Portion (SLP). SLP có thể được đổi thành fiat (tiền pháp định).

Theo Coindesk, tính đến tháng 7/2021, trung bình một người chơi Axie Infinity có thể kiếm được khoảng 4,500 SLP/tháng (= \$800). Trò chơi đã có sự tăng trưởng lớn từ 30,000 người chơi game hàng ngày trong tháng 4/2021 lên hơn 1 triệu vào tháng 8, với hầu hết người chơi đến từ các thị trường mới nổi như Philippines, Venezuela và Brazil.

Trên thực tế, có rất nhiều người Philippines đã bỏ việc để chơi Axie Infinity hàng giờ đồng hồ mỗi ngày vì nó trả “lương” tốt hơn. Thậm chí, một số người kiếm được gấp 3 lần so với thu nhập bình thường của họ.

Axie Infinity đại diện cho một mô hình kinh doanh mới dựa trên blockchain, nơi những người sáng tạo trò chơi chia sẻ doanh thu của họ với người chơi. Mỗi đô la mà người chơi kiếm được vốn có thể đi thẳng vào túi những nhà phát triển. Thế nhưng, thay vì tính phí truy cập, bán quảng cáo hay buộc người dùng mua in-app và giữ tất cả lợi nhuận cho riêng mình, các nhà phát triển Axie quyết định chia sẻ lợi nhuận với người chơi, hy vọng rằng trò chơi sẽ được ứng dụng rộng rãi.

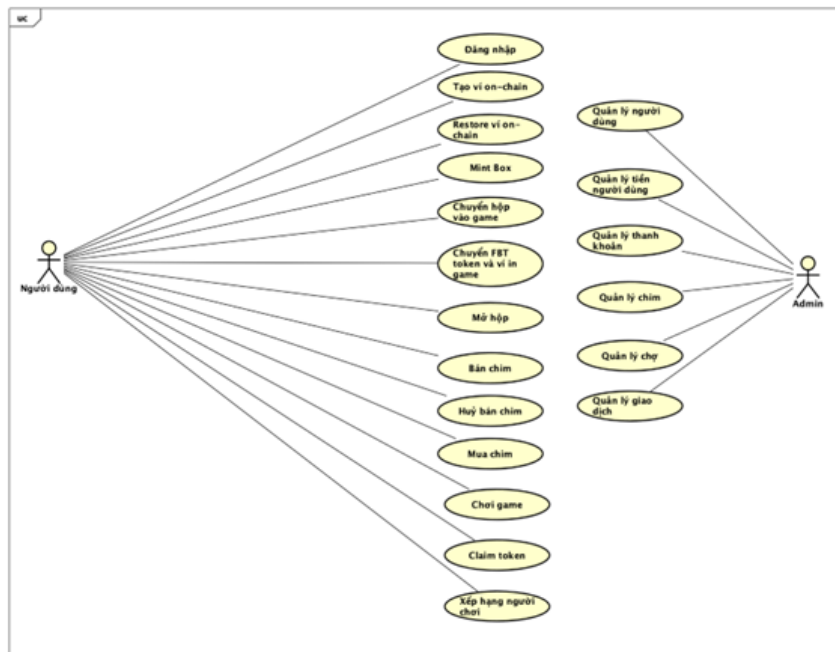
Mặc dù một số người cho rằng nền kinh tế của trò chơi không bền vững, nhưng cho đến nay chiến lược này vẫn đang hiệu quả: 60% người chơi biết về Axie do được bạn bè hay thành viên trong gia đình giới thiệu, và trò chơi đang trên đà tạo ra doanh thu hơn 1 tỷ đô trong năm 2021.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát

Hệ thống gồm có 2 tác nhân là quản trị viên, người dùng.

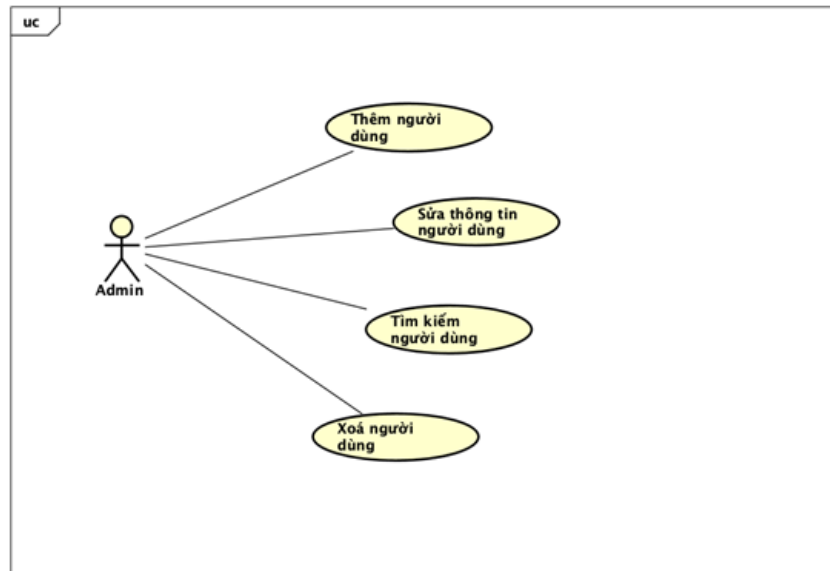
Để vào ứng dụng, người dùng cần sử dụng tài khoản email để đăng nhập, hệ thống sẽ gửi mã code xác nhận về email mà người dùng nhập. Sau khi xác thực mã code thành công, người dùng sẽ phải thêm ví hoặc tạo ví mới. Sau khi thêm ví/ tạo ví thành công, người dùng sẽ dùng đồng FBT token trong ví để mua BirdBox NFT. Sau khi mua BirdBox thành công, người dùng có thể mở BirdBox để nhận được Bird có các thuộc tính khác nhau. Khi có Bird, người dùng sẽ tham gia chơi game để kiếm token. Ngoài ra, người dùng còn có thể mua hoặc đặt bán Bird ở chợ, xem bảng xếp hạng xếp theo số token người chơi kiếm được. Quản trị viên có chức năng quản lý các tài khoản người dùng, danh sách Bird của người dùng, danh sách ví của người dùng, số token người chơi kiếm được.



Hình 2.1: Biểu đồ usecase tổng quan

2.2.2 Biểu đồ use case phân rã chức năng Quản lý người dùng

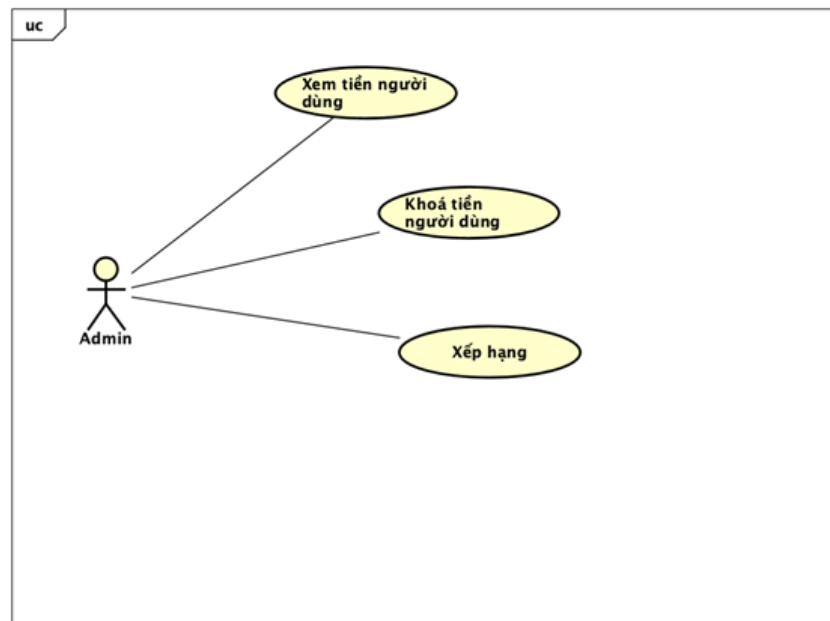
Usecase quản lý người dùng thực hiện bởi quản trị viên. Quản lý người dùng bao gồm thêm người dùng, sửa thông tin người dùng, tìm kiếm người dùng, xóa người dùng. Hình 2.2 mô tả usecase quản lý người dùng:



Hình 2.2: Usecase phân rã chức năng Quản lý người dùng

2.2.3 Biểu đồ usecase phân rã chức năng Quản lý tiền người dùng

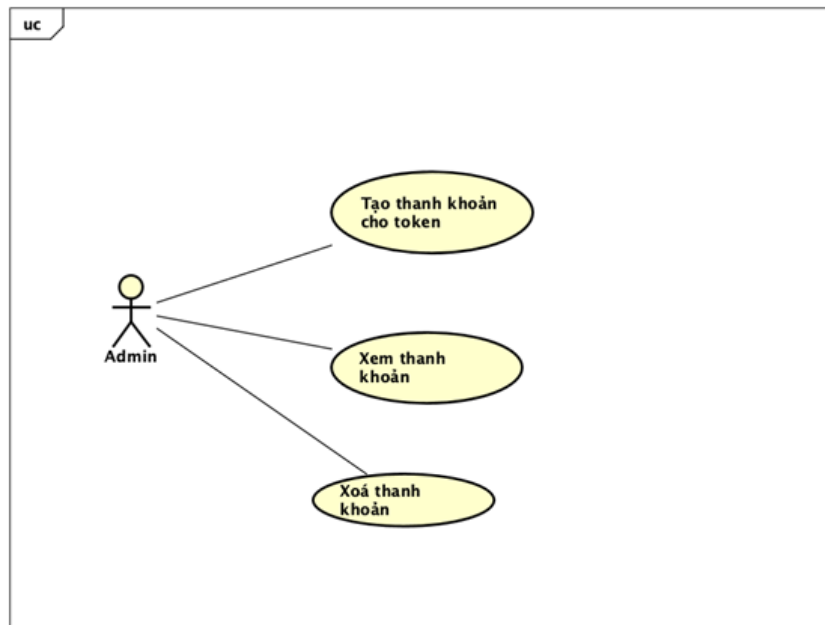
Usecase quản lý tiền người dùng thực hiện bởi quản trị viên. Quản lý tiền người dùng bao gồm xem tiền người dùng, khóa tiền người dùng, xếp hạng. Hình 2.3 mô tả usecase quản lý tiền người dùng:



Hình 2.3: Usecase phân rã chức năng Quản lý tiền người dùng

2.2.4 Biểu đồ usecase phân rã chức năng Quản lý thanh khoản

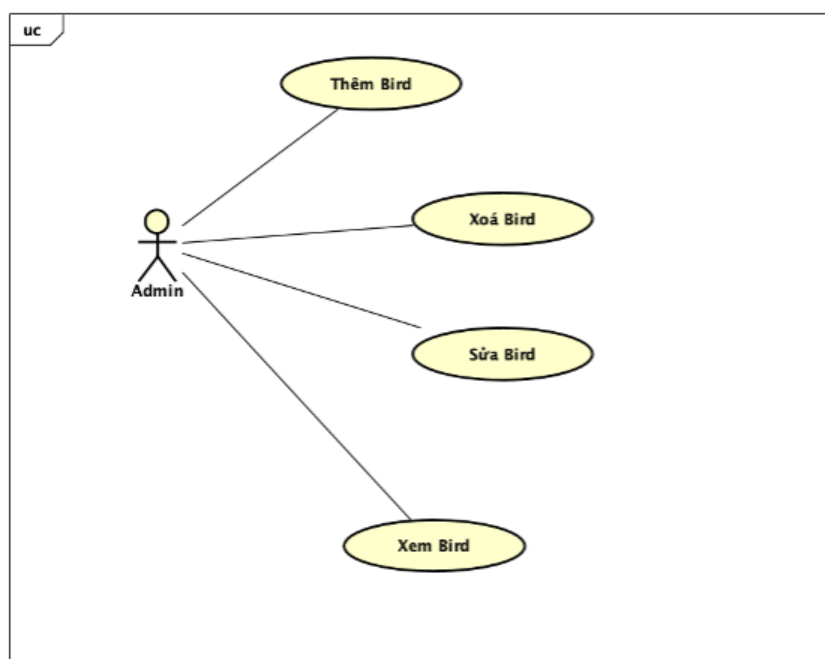
Usecase quản lý thanh khoản thực hiện bởi quản trị viên. Quản lý người dùng bao gồm thêm thanh khoản, xem thanh khoản, xóa thanh khoản. Hình 2.4 mô tả usecase quản lý thanh khoản:



Hình 2.4: Usecase phân rã chức năng quản lý thanh khoản

2.2.5 Biểu đồ usecase phân rã chức năng Quản lý Bird

Usecase quản lý Bird thực hiện bởi quản trị viên. Quản lý Bird bao gồm thêm Bird, sửa Bird, xem Bird, xóa Bird. Hình 2.5 mô tả usecase quản lý Bird:

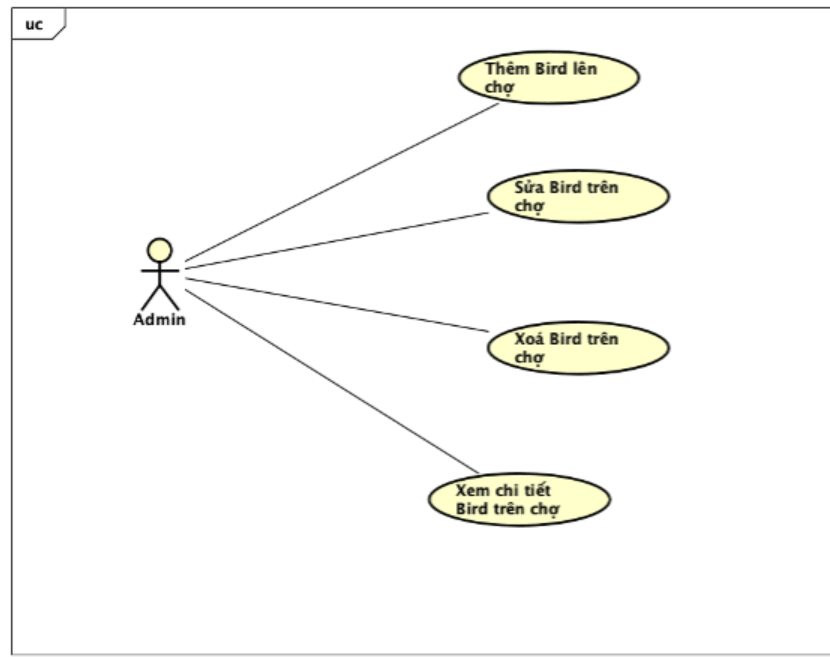


Hình 2.5: Usecase phân rã chức năng quản lý Bird

2.2.6 Biểu đồ usecase phân rã chức năng quản lý chợ

Usecase quản lý chợ thực hiện bởi quản trị viên. Quản lý chợ bao gồm thêm Bird lên chợ, sửa Bird bán trên chợ, xóa Bird bán trên chợ, xem Bird bán trên chợ. Hình

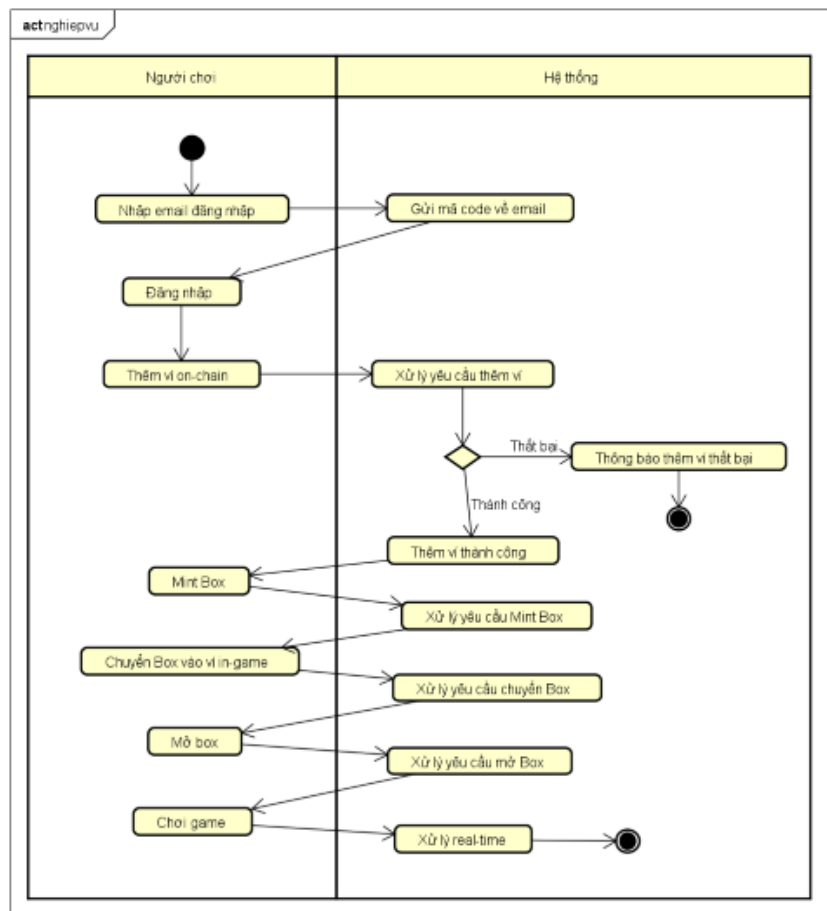
2.6 mô tả usecase quản lý chợ:



Hình 2.6: Usecase phân rã chức năng quản lý chợ

2.2.7 Quy trình nghiệp vụ

Hình 2.7 mô tả quy trình nghiệp vụ của game NFT Flappy Bird. Người dùng sử dụng tài khoản email để đăng nhập, hệ thống sẽ gửi mã code xác nhận về email mà người dùng nhập. Sau khi xác thực mã code thành công, người dùng sẽ phải thêm ví hoặc tạo ví mới. Sau khi thêm ví/ tạo ví thành công, người dùng sẽ dùng đồng FBT token trong ví để mua BirdBox NFT. Sau khi mua BirdBox thành công, người dùng có thể mở BirdBox để nhận được Bird có các thuộc tính khác nhau. Khi có Bird, người dùng sẽ tham gia chơi game để kiếm token. Ngoài ra, người dùng còn có thể mua hoặc đặt bán Bird ở chợ, xem bảng xếp hạng xếp theo số token người chơi kiếm được.



Hình 2.7: Quy trình nghiệp vụ

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Thêm ví on-chain

Đặc tả use case thêm ví on-chain được mô tả trong Bảng 2.1.

Bảng 2.1: Đặc tả use case Thêm ví on-chain

Mã Usecase	UC001	Tên Usecase	Thêm ví on-chain
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đã đăng nhập		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn gửi mã xác thực
	2	Hệ thống	Lắng nghe sự kiện và gửi mã xác thực về tài khoản email đã đăng nhập
	3	Người dùng	Nhập mã xác thực và seed phrase của ví muốn thêm và gửi lên hệ thống
	4	Hệ thống	Kiểm tra mã xác thực có hợp lệ không
	5	Hệ thống	Kiểm tra seed phrase có hợp lệ không
	6	Hệ thống	Trả về thông báo thêm ví thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Thông báo mã xác thực không hợp lệ
	5a	Hệ thống	Thông báo seed phrase không hợp lệ

2.3.2 Đặc tả use case Mint Box

Đặc tả usecase Mint Box được mô tả chi tiết trong Bảng 2.2.

Bảng 2.2: Đặc tả use case Mint Box

Mã Usecase	UC002	Tên Usecase	Mint Box
Tác nhân	Người dùng		
Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn Mint Box ở phần giao diện
	2	Hệ thống	Chuyển màn hình sang màn hình Mint Box
	3	Người dùng	Chọn các BirdBox muốn mua và chọn mua
	4	Hệ thống	Hệ thống xử lý yêu cầu Mint Box
	5	Hệ thống	Thông báo Mint Box thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	4a	Hệ thống	Thông báo lỗi không đủ số dư

2.3.3 Đặc tả use case mua Bird

Đặc tả use case Mua Bird được mô tả trong Bảng 2.3.

Bảng 2.3: Đặc tả use case Mua Bird

Mã Usecase	UC003	Tên Usecase	Mua Bird
Tác nhân	Người dùng		
Tiền điều kiện	Không		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn Bird cần mua và ấn nút "Buy"
	2	Hệ thống	Lắng nghe sự kiện và xử lý yêu cầu mua Bird
	3	Hệ thống	Thông báo mua Bird thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	3a	Hệ thống	Thông báo lỗi không đủ số dư

2.3.4 Đặc tả use case Chơi game

Đặc tả use case Chơi game được mô tả trong Bảng 2.4.

Bảng 2.4: Đặc tả use case Chơi game

Mã Usecase	UC004	Tên Usecase	Chơi game
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đã đăng nhập		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn vào chơi game và ấn nút "Start"
	2	Hệ thống	Lắng nghe sự kiện và kiểm tra Bird còn lượt chơi rồi cho chơi
	3	Người dùng	Bắt đầu chơi game
	4	Hệ thống	Lắng nghe và xử lý điểm của người chơi kiểm và trả thưởng tương ứng với điểm cho người dùng
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Thông báo lỗi không đủ năng lượng

2.3.5 Đặc tả use case Mở Box

Đặc tả use case mở Box trong bảng Bảng 2.5.

Bảng 2.5: Đặc tả use case mở Box

Mã Usecase	UC005	Tên Usecase	Mở Box
Tác nhân	Người dùng		
Tiền điều kiện	Người dùng đã đăng nhập		
Luồng sự kiện chính (Thành công)	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn hộp cần mở và ấn nút "Open"
	2	Hệ thống	Lắng nghe sự kiện và trả về kết quả mở hộp cho người dùng
Luồng sự kiện thay thế	Không		

2.4 Yêu cầu phi chức năng

- Giao diện đẹp, thân thiện với người dùng.
- Các thao tác phía người dùng trong trường hợp có lỗi phải xác định được lỗi gây ra do người dùng hay do hệ thống.
- Hệ thống đảm bảo khả năng mở rộng, dễ bảo trì.
- Đảm bảo hệ thống có thể phục vụ số lượng người sử dụng lớn.
- Dữ liệu được cập nhật liên tục.
- Đảm bảo hiệu năng với số lượng người dùng cùng 1 thời điểm lớn.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

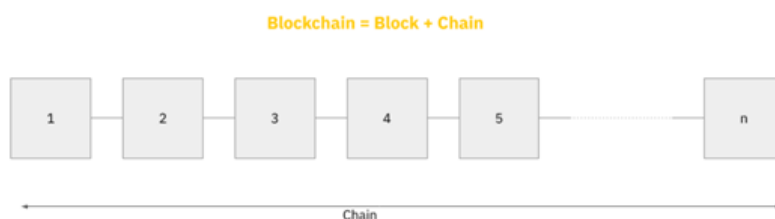
Để có thể thực hiện được chính xác, hiệu quả các tính năng người sử dụng yêu cầu, giai đoạn phân tích và thiết kế kiến trúc là rất cần thiết. Phân tích và thiết kế sẽ mang đến bức tranh toàn cảnh của ứng dụng xây dựng, hiểu rõ ưu, nhược điểm của các giải pháp để tìm ra hướng đi tối ưu. Chương 3 sẽ làm rõ về các công nghệ sử dụng phát triển ứng dụng.

3.1 Blockchain

Blockchain [9] là một cơ sở dữ liệu phân cấp lưu trữ thông tin trong các khối(block) được liên kết với nhau bằng mã hóa và mở rộng theo thời gian để tạo thành một chuỗi(chain). Mỗi khối trong Blockchain sẽ được liên kết với khối trước đó, chứa thông tin về thời gian khởi tạo kèm một mã thời gian và dữ liệu giao dịch.

Blockchain có thể được xem là một cuốn sổ cái điện tử được phân phối trên nhiều máy tính khác nhau, lưu trữ mọi thông tin giao dịch và đảm bảo các thông tin đó không thể bị thay đổi dưới bất kỳ hình thức nào.

Cấu trúc của một Blockchain gồm khối(block) và chuỗi các giao dịch.



Hình 3.1: Cấu trúc của một Blockchain

3.1.1 Giao dịch [10]

Đây là nền tảng trọng tâm. Các giao dịch thể hiện các hoạt động chuyển nhượng hoặc thay đổi giá trị tài sản hữu hình hoặc vô hình giữa các bên.

3.1.2 Khối [11]

Mỗi khối gồm 3 thành phần: Dữ liệu(data), mã hàm băm(hash) và mã hàm băm của khối trước nó.

- Dữ liệu: Các bản ghi dữ liệu đã được xác thực của người dùng được bảo vệ bằng thuật toán mã hóa tùy thuộc vào từng blockchain.
- Mã hàm băm(hash): Là chuỗi các ký tự và số được tạo một cách ngẫu nhiên và không giống nhau. Nó đại diện riêng cho block đó và được mã hóa bằng thuật toán mã hóa. Mã hash dùng để phát hiện sự thay đổi trong các khối.

- Mã băm của khối trước (previous hash): Nó dùng để các khối liên kế nhận biết được khối nào trước khối nào sau và nối với nhau.

3.1.3 NFT [12]

NFT là viết tắt của Non-fungible Token, là loại token có tính độc nhất và không thể bị thay thế bởi những token khác. Đây có thể là tài sản ảo hoặc phiên bản mã hóa của tài sản trong thế giới thực. Vì mỗi NFT là duy nhất và chúng không thể hoán đổi cho nhau, nên chúng có thể hoạt động như bằng chứng xác thực và quyền sở hữu lĩnh vực kỹ thuật số.

Nhưng các Non-fungible Token thì đại diện cho những vật phẩm hay tài sản có giá trị sưu tầm khác nhau, chính vì thế NFT được ứng dụng bởi các nghệ sĩ, họa sĩ hay nhà phát triển game để token hóa các sản phẩm của mình và biết chúng trở thành duy nhất.

3.1.4 Token [13]

Token là đồng tiền điện tử. Nó được tạo ra trên Blockchain đã có sẵn nhờ tạo ra các hợp đồng thông minh, phổ biến nhất là Ethereum.

Ưu điểm:

- Tích hợp Smart Contracts để tạo ra Token đơn giản
- Đa dạng hoá chức năng như: bỏ phiếu, nhận lãi, Stacking
- Đơn giản hoá quy trình sử dụng, không cần phải mất nhiều bước để bảo mật thông tin mà vẫn giao dịch và thanh toán nhanh chóng

3.1.5 Smart Contract [14]

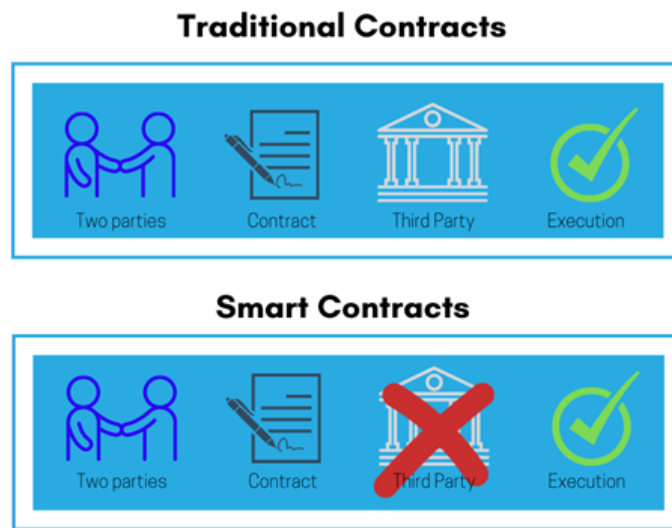
Smart contract (hay hợp đồng thông minh) là các chương trình chạy trên blockchain. Hợp đồng thông minh cũng giống như một hợp đồng kỹ thuật số bị bắt buộc thực hiện bởi một bộ quy tắc cụ thể. Các quy tắc này do bộ mã máy tính xác định trước mà tất cả các nút (node) trong mạng đều phải sao chép và thực thi các quy tắc đó.

Về bản chất, Smart contract chỉ là một đoạn mã chạy trên hệ thống phân tán (blockchain), cho phép tạo ra các giao thức Permissionless (tức là không cần trao quyền). Điều đó có nghĩa là:

- Hai bên trong hợp đồng có thể đưa ra các cam kết thông qua blockchain mà không cần phải biết danh tính hay tin tưởng lẫn nhau.
- Họ có thể đảm bảo rằng nếu các điều kiện của hợp đồng không được thỏa mãn, hợp đồng sẽ không được thực thi.

Ngoài ra, việc sử dụng hợp đồng thông minh loại bỏ nhu cầu đối với bên trung

gian, giúp giảm đáng kể chi phí hoạt động.



Hình 3.2: Cấu trúc của Smart Contract

Mỗi blockchain có một phương pháp triển khai hợp đồng thông minh khác nhau, ví dụ trên Cosmos có WASM, Polkadot có ink!,... Trong đó nổi bật nhất vẫn là Smart contract chạy trên máy ảo của Ethereum (Ethereum Virtual Machine - EVM [15]).

Trên Ethereum [16], các Smart Contract chịu trách nhiệm thực thi và quản lý các hoạt động diễn ra trên blockchain khi những người dùng tương tác với nhau. Bất kỳ địa chỉ nào không phải là smart contract đều được là Tài khoản độc lập (Externally Owned Account - EOA). Do đó, smart contract sẽ do máy tính kiểm soát và EOA do người dùng kiểm soát.

Smart contract Ethereum bao gồm một mã hợp đồng và hai khóa công khai [17]:

- Khóa công khai thứ nhất là khóa do người tạo hợp đồng cung cấp.
- Khóa còn lại đại diện cho chính hợp đồng, khóa này có vai trò như một mã định danh kỹ thuật số duy nhất cho mỗi smart contract.

Smart contract được triển khai thông qua giao dịch blockchain và chúng chỉ được kích hoạt khi một Tài khoản độc lập (EOA) hoặc các smart contract khác call chúng. Tuy nhiên, kích hoạt đầu tiên luôn từ phía EOA (người dùng).

Ưu và nhược điểm của Smart Contract

Ưu điểm:

- Hợp đồng thông minh là một bộ mã có thể lập trình, có khả năng tùy chỉnh cao và có thể được thiết kế theo nhiều cách khác nhau để có thể cung cấp nhiều

loại dịch vụ và giải pháp.

- Ngoài ra, hợp đồng thông minh là các chương trình phi tập trung và tự thực hiện(self-executing), chúng giúp tăng tính minh bạch và giảm chi phí hoạt động. Nếu được triển khai đúng cách, chúng cũng có thể tăng hiệu quả vận hành và giảm chi phí hành chính.

Nhược điểm:

- Smart Contract dựa trên hệ thống blockchain nên không thể sửa đổi và can thiệp. Một khi Smart Contract được viết ra, nếu muốn thay đổi chỉ có cách viết lại một hợp đồng mới. Tính chất không thể thay đổi là một ưu điểm lớn, tuy nhiên trong một số trường hợp có thể là nhược điểm.

3.2 Platform NodeJs và Framework Express

NodeJs là một platform tạo môi trường để thực thi ngôn ngữ Javascript phía máy chủ. Với việc hỗ trợ xử lý bất đồng bộ, cùng với tích hợp Google V8 engine , việc biên dịch mã nguồn của máy chủ sẽ trở nên nhanh hơn, mềm dẻo hơn, cũng như đảm bảo tính mở rộng sau này.

Để tạo dựng một bộ khung cấu trúc của ứng dụng thì NodeJs cần tích hợp thêm các framework bên trong nó. Với ưu thế mềm dẻo, tối ưu hóa và linh hoạt, Express framework [18] là lựa chọn phổ biến hàng đầu để tích hợp với NodeJs.

Các ưu điểm nổi bật khi sử dụng Express kết hợp với NodeJS có thể kể đến như là:

- Đảm bảo tính đơn giản, dễ điều chỉnh theo yêu cầu sử dụng.
- Hoạt động dựa trên URL và phương thức HTTP, hỗ trợ định nghĩa các route trong ứng dụng.
- Hỗ trợ biên dịch các trang HTML động qua các view engine (EJS, Pug và Mustache) .
- Hỗ trợ việc viết REST API phía máy chủ.
- Hệ thống quản lý thư viện đa dạng, phong phú, dễ dàng cài đặt và tích hợp các thư viện cần thiết. (thông qua Node package manager) .

Việc kết hợp giữa NodeJs và Express Framework tạo nên bộ khung vững chắc phía máy chủ. Không chỉ tiết kiệm thời gian khi xây dựng website, sự kết hợp còn giúp giảm thiểu các yêu cầu phải xử lý lên phía máy chủ, mang đến trải nghiệm tối ưu cho người dùng.

3.3 MongoDB [19]

Mọi dữ liệu khách hàng khi sử dụng ứng dụng cần được lưu trữ một cách tối ưu và hiệu quả. Vì vậy, việc lựa chọn một hệ quản trị dữ liệu tối ưu thực sự rất cần thiết.

Hiện nay, mô hình lưu trữ dữ liệu được chia thành 2 loại chính: mô hình dữ liệu quan hệ RDBMS (Relational Database Management System) và mô hình dữ liệu phi quan hệ NoSQL [20] (Not only SQL).

Mô hình dữ liệu quan hệ (tiêu biểu là hệ quản trị dữ liệu MySQL) có tiêu chí ràng buộc dữ liệu rất mạnh mẽ. Mọi dữ liệu khi truy xuất cơ sở dữ liệu quan hệ đều được kiểm tra ràng buộc chặt chẽ. Điều này sẽ đảm bảo tính toàn vẹn cho dữ liệu.

Trái ngược với mô hình dữ liệu quan hệ là mô hình dữ liệu phi quan hệ (tiêu biểu là hệ quản trị dữ liệu MongoDB). Khi sử dụng cơ sở dữ liệu phi quan hệ, mọi dữ liệu khi truy xuất sẽ không bị ràng buộc về quan hệ, vì vậy thời gian truy xuất các dữ liệu độc lập sẽ nhanh hơn so với cơ sở dữ liệu quan hệ. Tuy nhiên, trong trường hợp dữ liệu cần truy xuất liên quan đến các dữ liệu khác, cơ sở dữ liệu quan hệ lại phát huy tính hiệu quả của nó.

Khi xây dựng ứng dụng game, lựa chọn cơ sở dữ liệu phi quan hệ MongoDB, đảm bảo thời gian truy cập nhanh nhất, kết hợp thư viện SocketIO mang đến tính chất “real-time” cho việc chơi game.

3.4 React Native [21]

Được phát triển bởi Facebook, React Native là một framework hướng đến phát triển ứng dụng di động đa nền tảng. Với sự trợ giúp của React Native, lập trình viên (developer) có thể sử dụng JavaScript để tạo ra mobile apps (ứng dụng di động) hỗ trợ cho cả nền tảng Android và iOS.

React Native sở hữu rất nhiều ưu điểm tuyệt vời. Cụ thể là:

- Có thể tái sử dụng code.
- Cộng đồng người dùng lớn
- Tính ổn định và tối ưu

Tuy nhiên nó vẫn tồn tại những nhược điểm sau:

- Yêu cầu Native code.
- Hiệu năng kém hơn so với Native App.
- Bảo mật chưa thật sự tốt do dùng JavaScript. Do sử dụng JavaScript, người dùng cũng sẽ bị ảnh hưởng bởi những đặc điểm của JavaScript: dễ làm dễ sai,

dẫn đến khó duy trì về sau.

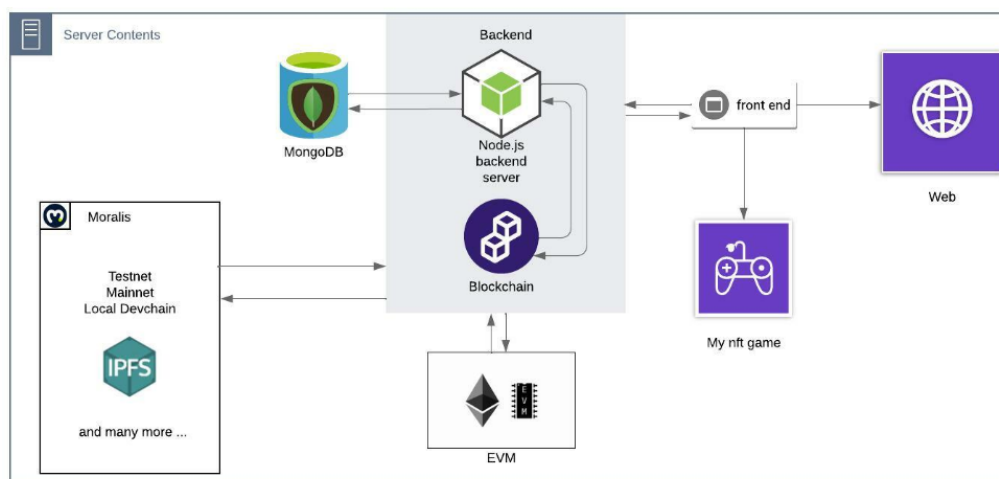
- Quản lý bộ nhớ.
- Tùy biến chưa thật sự tốt ở một số module.
- Không thích hợp cho các app cần năng lực tính toán cao (hash, crypto, etc).

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

Với những gì đã phân tích trong chương 2 và công nghệ đã lựa chọn được ở chương 3. Trong chương 4 sẽ trình bày nội dung về thiết kế, ứng dụng sẽ được triển khai xây dựng và kiểm thử.

4.1 Thiết kế kiến trúc

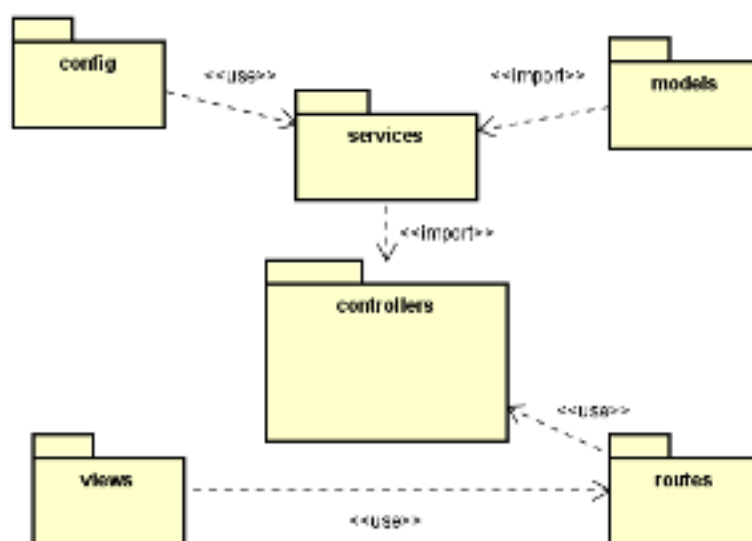
4.1.1 Lựa chọn kiến trúc phần mềm



Hình 4.1: Mô hình áp dụng trong ứng dụng

Hình 4.1 mô tả kiến trúc xây dựng game NFT. Theo mô hình này, ứng dụng được chia thành 2 khối, đảm nhận các nhiệm vụ khác nhau. Front-end có vai trò: nhận các yêu cầu của người dùng, gửi các yêu cầu đó đến Back-end và nhận các kết quả xử lý từ Back-end, trả về cho người dùng. Khối Back-end sẽ bao gồm server và smart contract. Trong đó server là nơi sẽ nhận các yêu cầu từ Front-end và gọi đến Blockchain thông qua Smart Contract.

4.1.2 Thiết kế tổng quan



Hình 4.2: Biểu đồ phụ thuộc gói trong ứng dụng

Sự phụ thuộc gói của ứng dụng được thể hiện qua Hình 10, trong đó:

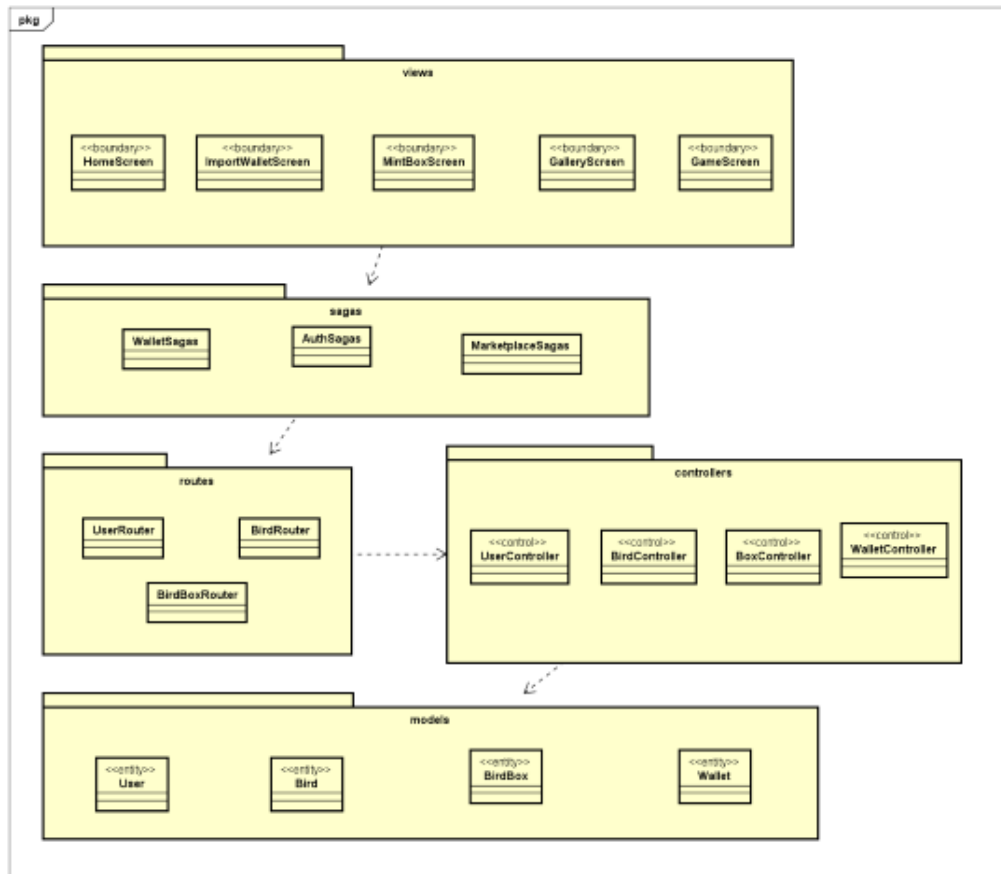
- Gói views: là nơi lưu trữ các template để hiển thị dữ liệu. Dữ liệu hiển thị được nhận từ API hoặc nhận trực tiếp từ các controller.
- Gói routes: chia thành web và api. Các route web là những route chịu trách nhiệm cho hiển thị dữ liệu từ controller. Các route api sẽ nhận dữ liệu từ controller, sau đấy hiển thị lên view.
- Gói controllers: xử lý các request từ routes. Các dữ liệu nhận từ request sẽ được kiểm tra, xử lý trước khi chuyển cho Service.
- Gói services: Nhận dữ liệu từ controller, sử dụng dữ liệu từ config và model (nếu cần thiết), chịu trách nhiệm truy vấn và xử lý đối với cơ sở dữ liệu.

Ngoài các gói chính trên, khi xây dựng ứng dụng đặt lịch khám bệnh, còn có thêm các gói: lang (hiển thị ngôn ngữ), helper (các hàm dùng chung), migration, seeder (tạo bảng và dữ liệu mẫu cho database)

4.1.3 Thiết kế chi tiết gói

Hình 4.3 thể hiện thiết kế chi tiết gói cho nghiệp vụ đặt hàng. Có 5 gói là views, sagas, routes, controllers, models, trong đó views là package chứa các lớp bao có chức năng tương tác với người dùng, sagas là package chứa các lớp có chức năng nhận yêu cầu của lớp bao và tương tác với server, routes là package chứa các lớp định nghĩa các đầu api, controllers là package chứa các lớp xử lý nghiệp vụ tương

ứng với các chức năng, models là package chứa các lớp thực thể tương ứng với các bảng trong cơ sở dữ liệu.



Hình 4.3: Thiết kế chi tiết gói

4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

a, Thiết kế thành phần giao diện màn hình thêm ví on-chain

Màn hình thêm ví on-chain được chia thành 4 thành phần con như Hình 4.4. Bao gồm:

- Nút back: Thành phần nút có chức năng quay lại màn trước đó.
- Mã code: Thành phần text input để người dùng nhập vào mã code.
- Seed Phrase: Thành phần text input để người dùng nhập vào cụm từ khôi phục của ví
- Nút Import: Thành phần nút có chức năng gửi yêu cầu import ví.

03:37

IMPORT WALLET

Email Verification code

Send code

Seed Phrase

Enter the Seed Phrase word and separate with space

IMPORT WALLET

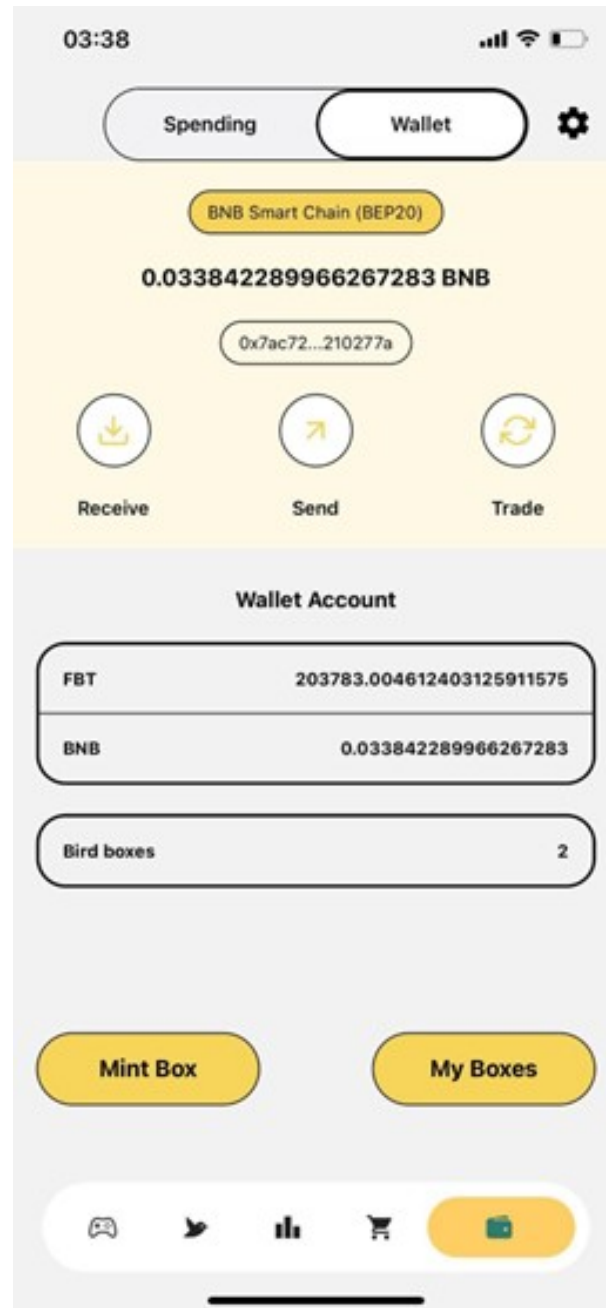
Hình 4.4: Thiết kế thành phần giao diện màn Thêm ví on-chain

b, Thiết kế thành phần giao diện màn hình Ví on-chain

Màn hình Ví on-chain được chia thành 4 thành phần con như Hình 4.5. Bao gồm:

- Các chức năng ví: Thành phần các nút: nút nhận token, chuyển token, swap token.
- Số token: Thành phần hiển thị số token đang có trong ví.
- Nút Mint Box: Thành phần nút có chức năng điều hướng đến màn mint box.
- Nút My boxes: Thành phần nút có chức năng điều hướng đến màn danh sách

box của tài khoản ví.

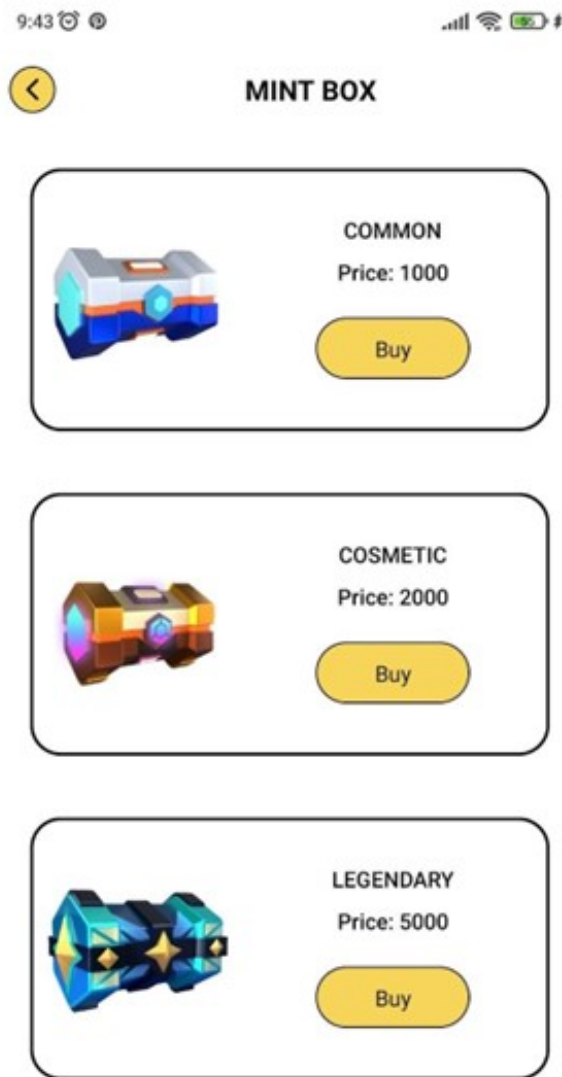


Hình 4.5: Thiết kế thành phần giao diện màn Ví on-chain

c, Thiết kế thành phần giao diện màn hình Mint Box

Màn hình Mint Box được chia thành 2 thành phần con như Hình 4.6. Bao gồm:

- Nút back: Thành phần nút có chức năng quay lại màn trước đó.
- Danh sách các loại Box: Thành phần hiển thị 3 loại Box với 3 giá tiền khác nhau để người dùng mua và mint.

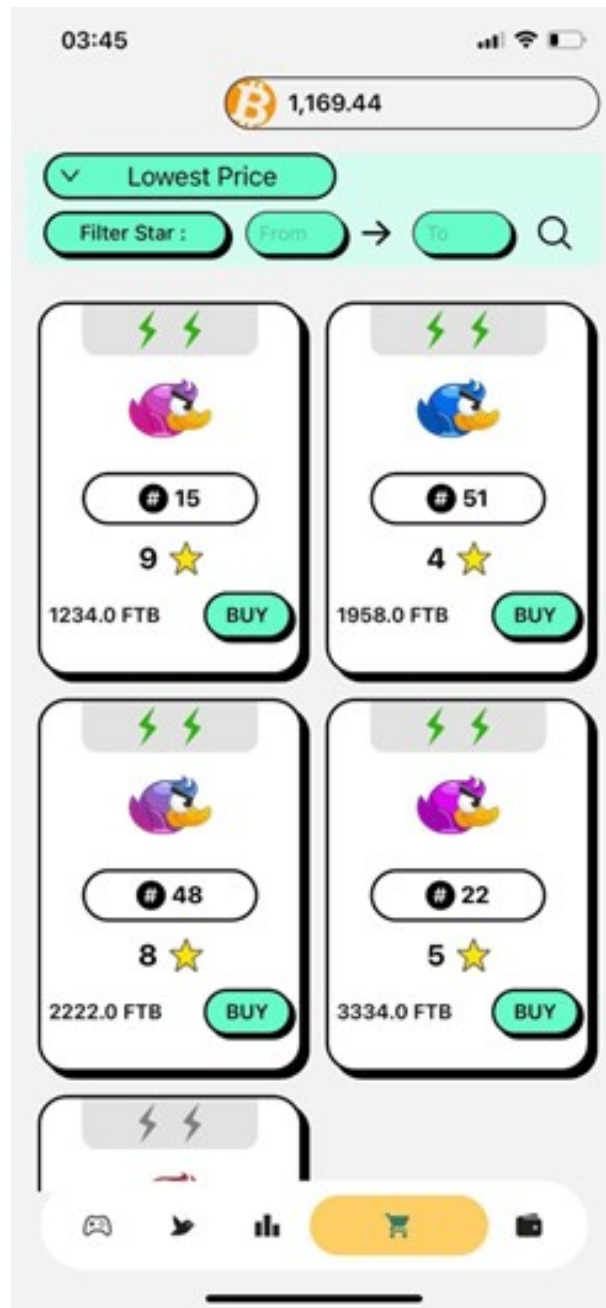


Hình 4.6: Thiết kế thành phần giao diện màn Mint Box

d, Thiết kế thành phần giao diện màn hình Chợ

Màn hình Chợ được chia thành 2 thành phần con như Hình 4.7. Bao gồm:

- Filter: Thành phần có chức năng chọn các loại filter theo giá bán, theo sao của Bird.
- Danh sách Bird trên chợ: Thành phần hiển thị danh sách Bird được đăng bán trên chợ.

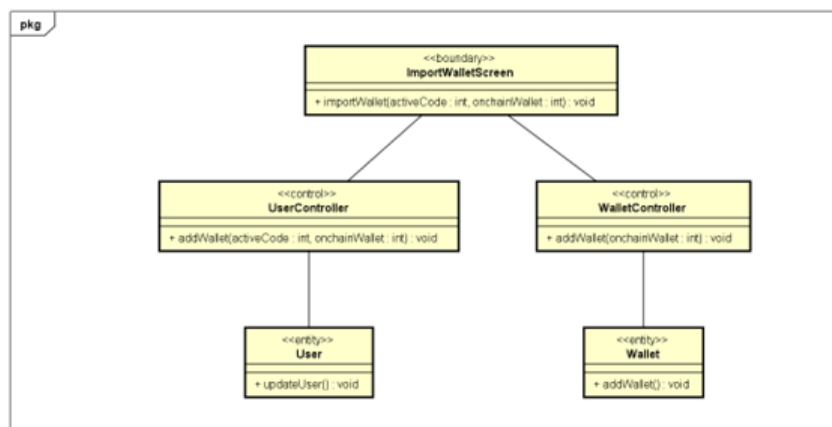


Hình 4.7: Thiết kế thành phần giao diện màn Chợ

4.2.2 Thiết kế lớp

a, Usecase thêm ví on-chain

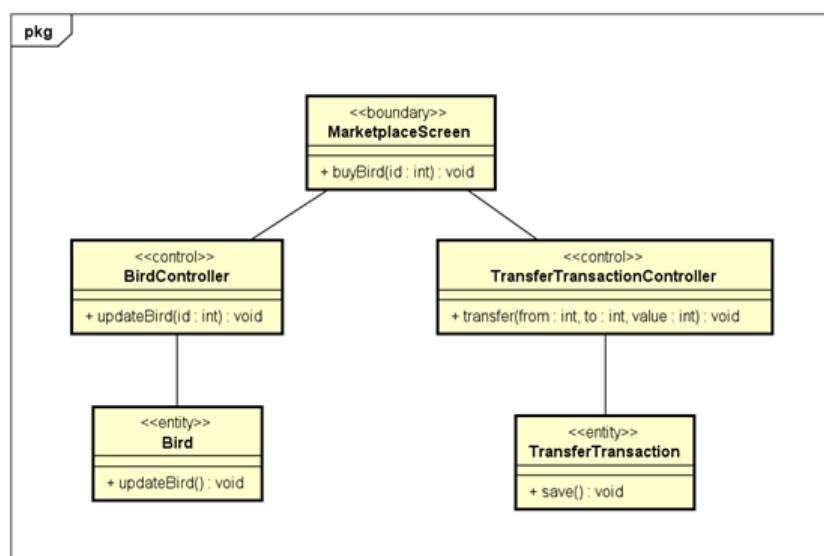
Hình 11 mô tả các lớp chính tham gia chức năng Thêm ví on-chain

**Hình 4.8:** Thiết kế các lớp chính tham gia chức năng Thêm ví on-chain**Bảng 4.1:** Thiết kế chi tiết lớp UserController

Tên phương thức	Danh sách tham số	Ý nghĩa
addWallet	activeCode, onChainWallet	Liên kết tài khoản với ví on-chain

b, Usecase Mua Bird

Hình 12 mô tả các lớp chính tham gia chức năng Mua Bird

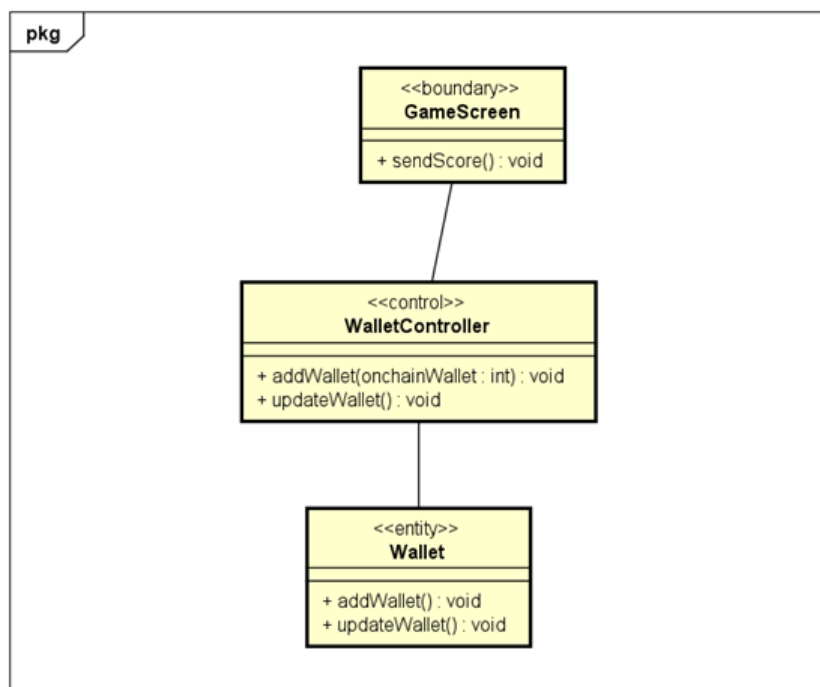
**Hình 4.9:** Thiết kế các lớp chính tham gia chức năng Mua Bird**Bảng 4.2:** Thiết kế chi tiết phương thức cho use case Mua Bird

Tên phương thức	Danh sách tham số	Ý nghĩa
updateBird	id	Cập nhật thông tin người sở hữu Bird

Tên phương thức	Danh sách tham số	Ý nghĩa
transfer	from, to, value	Ghi lại số token cần chuyển từ bên mua cho bên bán

c, Usecase Chơi game

Hình 13 mô tả các lớp chính tham gia chức năng Chơi game



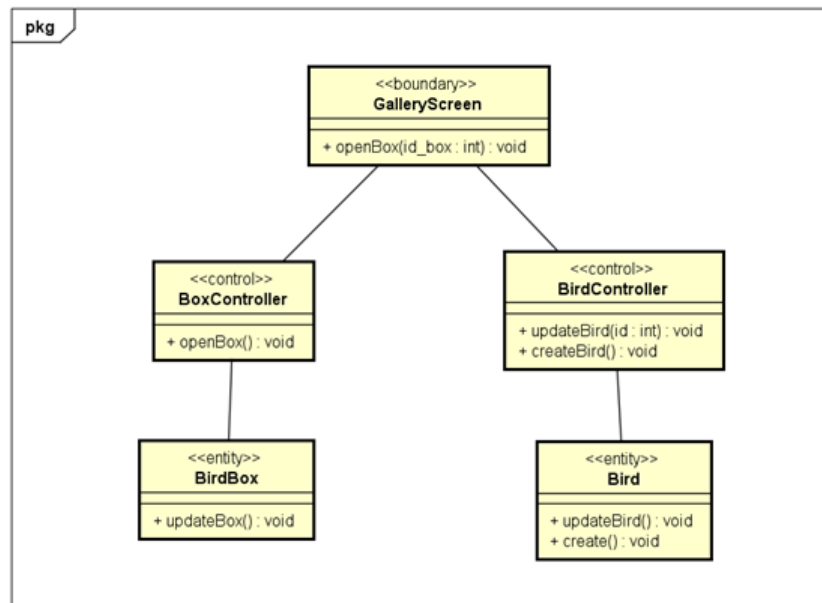
Hình 4.10: Thiết kế các lớp chính tham gia chức năng Chơi game

Bảng 4.3: Thiết kế chi tiết phương thức cho use case Chơi game

Tên phương thức	Danh sách tham số	Ý nghĩa
updateWallet	không	Cập nhật số token của ví người chơi

d, Usecase Mở BirdBox

Hình 14 mô tả các lớp chính tham gia chức năng Mở BirdBox

**Hình 4.11:** Thiết kế các lớp chính tham gia chức năng Mở BirdBox**Bảng 4.4:** Thiết kế chi tiết phương thức cho use case Mở BirdBox

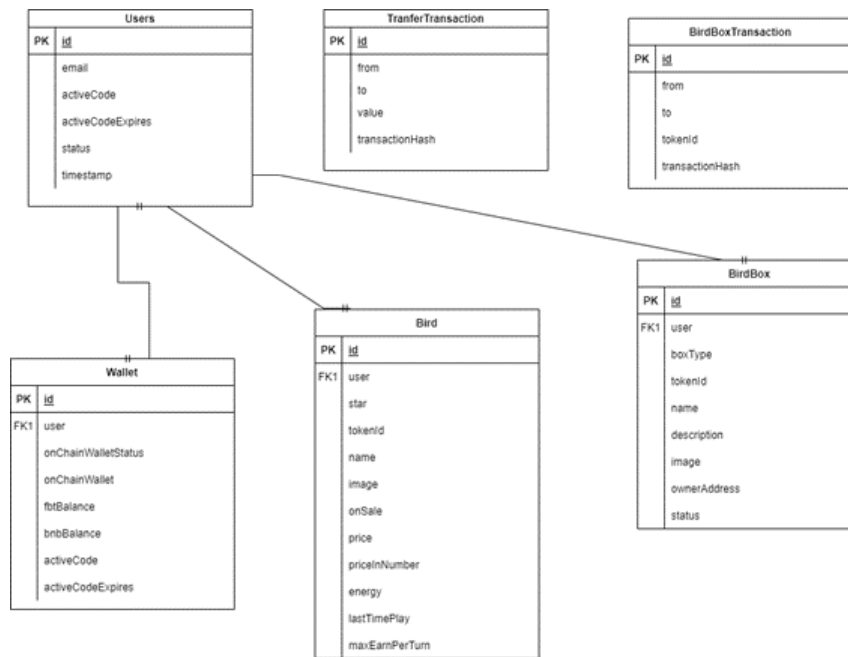
Tên phương thức	Danh sách tham số	Ý nghĩa
openBox	không	Xóa BirdBox
createBird	không	Tạo ra 1 Bird bất kỳ

4.2.3 Thiết kế cơ sở dữ liệu

Thiết kế cơ sở dữ liệu cho ứng dụng NFT hóa vào game Flappy Bird được thể hiện ở Hình 25.

Chi tiết thiết kế các bảng như sau: (tổng số 6 bảng)

- Bảng Users: lưu trữ các thông tin của người sử dụng, bao gồm: id, email, activeCode, activeCodeExpires, status, timestamp.
- Bảng Wallet: lưu trữ thông tin của ví trong game, bao gồm: id, user, onChainWalletStatus, onChainWallet, fbtBalance, bnbBalance, activeCode, activeCodeExpires
- Bảng Bird: lưu trữ thông tin của Bird trong game, bao gồm: id, user, star, tokenId, name, image, onSale, price, priceInNumber, energy, lastTimePlay, max-EarnPerTurn.
- Bảng BirdBox: lưu trữ thông tin của hộp Bird, bao gồm: id, boxType, tokenId, name, description, image, ownerAdress, status.
- Bảng TranferTransaction: lưu trữ thông tin giao dịch chuyển tiền trong game, bao gồm: id, from, to, transactionHash



Hình 4.12: Thiết kế cơ sở dữ liệu

a, Bảng User

Bảng User để lưu thông tin người dùng bao gồm:

Bảng 4.5: Bảng User

STT	Trường	Kiểu dữ liệu	Kích thước	Ý nghĩa	Ghi chú
1	id	integer		Mã người dùng	Khóa chính Not NULL
2	email	varchar	255	Email của người dùng	
3	activeCode	integer		Mã kích hoạt	
4	activecodeExpires	time		Thời gian hết hạn mã kích hoạt	
5	status	varchar	255	Trạng thái của người dùng	
6	timestamp	time		Thời gian đăng nhập	

b, Bảng Wallet

Bảng Wallet để lưu thông tin ví trong game bao gồm:

Bảng 4.6: Bảng Wallet

STT	Trường	Kiểu dữ liệu	Kích thước	Ý nghĩa	Ghi chú
1	id	integer		Mã ví	Khóa chính Not NULL
2	user	varchar	255	Mã của người dùng	
3	onChainWallet	integer		Vi on-chain	
4	onChainWalletStatus	varchar	255	Trạng thái ví on-chain	
5	fbtBalance	varchar	255	Số lượng fbt	
6	bnbBalance	varchar	255	Số lượng bnb	
7	activeCode	integer		Mã kích hoạt ví	
8	activeCodeExpires	time		Thời gian hết hạn mã kích hoạt ví	

c, Bảng Bird

Bảng Bird để lưu thông tin Bird của người dùng bao gồm:

Bảng 4.7: Bảng Bird

STT	Trường	Kiểu dữ liệu	Kích thước	Ý nghĩa	Ghi chú
1	id	integer		Mã Bird	Khóa chính Not NULL
2	user	varchar	255	Mã của người dùng	Khóa ngoài
3	star	integer		Số sao của Bird	
4	tokenId	integer		Mã BirdBox	
5	name	varchar	255	Tên Bird	
6	onSale	boolean		Trạng thái bán/ không bán	
7	energy	integer		Số năng lượng	
8	lastTimePlay	time		Thời gian cuối chơi game	
9	price	varchar	255	Giá Bird theo chữ	
10	priceInNumber	integer		Giá Bird theo số	
11	maxEarnPerTurn	integer		Số tiền có thể kiếm được trong một lần chơi	

d, Bảng BirdBox

Bảng BirdBox dùng để lưu thông tin hộp của con Bird bao gồm:

Bảng 4.8: Bảng BirdBox

STT	Trường	Kiểu dữ liệu	Kích thước	Ý nghĩa	Ghi chú
1	id	integer		Mã BirdBox	Khóa chính Not NULL
2	ownerAddress	varchar	255	Địa chỉ chủ sở hữu	
3	boxType	integer		Số loại Box	
4	tokenId	integer		Mã BirdBox	
5	name	varchar	255	Tên của BirdBox	
6	discription	varchar	255	Mô tả BirdBox	
7	image	varchar	255	Ảnh của Bird	
8	user	varchar		Mã người dùng	Khóa ngoài

e, Bảng TranferTransaction

Bảng TranferTransaction dùng để lưu trữ thông tin giao dịch của người dùng bao gồm:

Bảng 4.9: Bảng TranferTransaction

STT	Trường	Kiểu dữ liệu	Kích thước	Ý nghĩa	Ghi chú
1	id	integer		Mã giao dịch	Khóa chính Not NULL
2	from	varchar	255	Địa chỉ người gửi	
3	to	varchar	255	Địa chỉ người nhận	
4	value	varchar	255	Giá trị	
5	transactionHash	varchar	255	Mã băm của giao dịch	

f, Bảng BirdBoxTransaction

Bảng BirdBoxTransaction dùng để lưu trữ dữ liệu của người dùng khi chuyển hợp Bird từ ví vào game bao gồm:

Bảng 4.10: Bảng BirdBoxTransaction

STT	Trường	Kiểu dữ liệu	Kích thước	Ý nghĩa	Ghi chú
1	id	integer		Mã giao dịch BirdBox	Khóa chính Not NULL
2	from	varchar	255	Địa chỉ người gửi	
3	to	varchar	255	Địa chỉ người nhận	
4	value	varchar	255	Giá trị	
5	transactionHash	varchar	255	Mã băm của giao dịch	
6	tokenId	varchar	255	Mã số của BirdBox	

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Trong quá trình phát triển hệ thống, em đã sử dụng một số công cụ hỗ trợ được em liệt kê trong Bảng 4.11.

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual Studio Code	https://code.visualstudio.com/
Ngôn ngữ lập trình	Javascript	https://www.w3schools.com/js/default.asp
Framework lập trình giao diện phía frontend	React Native	https://reactnative.dev/
Framework lập trình phía backend	Express	https://expressjs.com
Hệ quản trị cơ sở dữ liệu	MongoDB	https://mongodb.com
Ngôn ngữ lập trình Smart Contract	Solidity	https://docs.soliditylang.org/en/v0.8.15/

Bảng 4.11: Danh sách thư viện và công cụ sử dụng

Ngoài ra em còn sử dụng một số thư viện bên thứ 3 việc phát triển hệ thống được dễ dàng, tập trung tối đa vào phần nghiệp vụ. Các thư viện được em liệt kê ở Bảng 4.12.

Thư viện	Phiên bản	Mục đích
Ethers	5.6.9	Thư viện tương tác với Blockchain
Axios	0.19.0	Thư viện gửi HTTP request
Redux	4.1.2	Thư viện quản lý state
React-redux	7.2.8	Kết nối React Component đến Redux store

Bảng 4.12: Danh sách thư viện sử dụng trong game NFT Flappy Bird

4.3.2 Kết quả đạt được

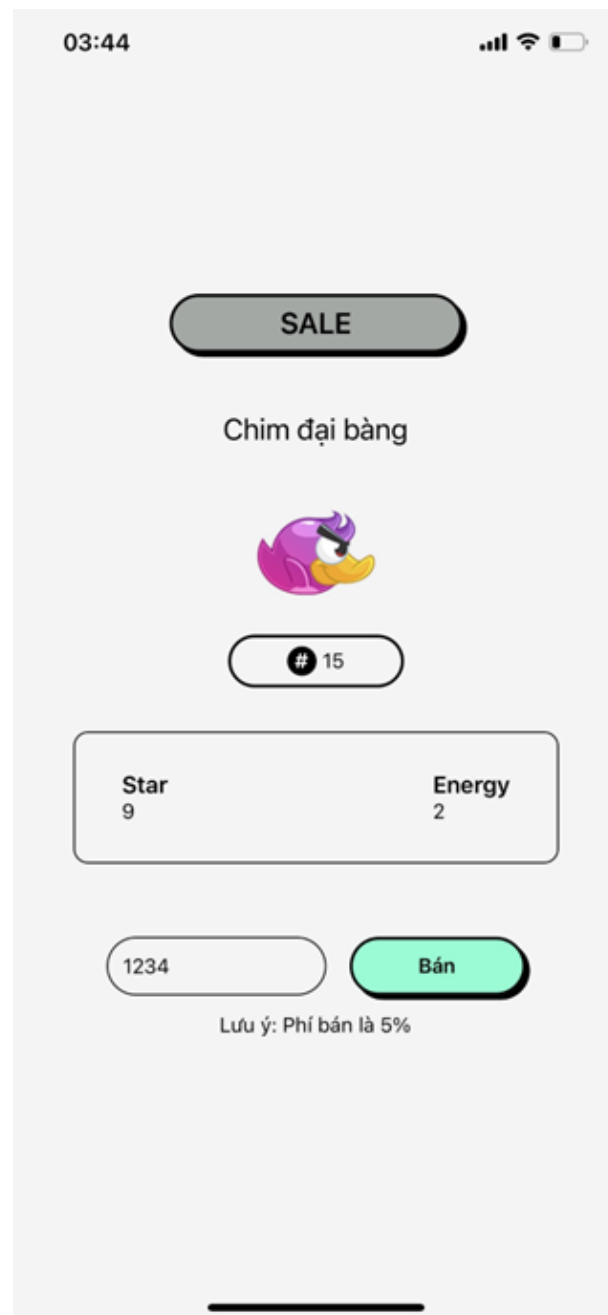
Hệ thống xây dựng đáp ứng tính năng sử dụng cho 2 người là người dùng và quản trị viên.

Người dùng thì có khả năng đăng nhập, đăng xuất tài khoản của mình khỏi hệ thống. Người dùng có thể chơi game khi đã đăng nhập. Khi chơi game người dùng sẽ được trả thưởng. Người dùng có thể tích hợp nhiều ví trên ứng dụng để có thể mua hộp và mở hộp để có được Bird để chơi game.

Quản trị viên có chức năng quản lý các tài khoản người dùng, danh sách Bird của người dùng, danh sách ví của người dùng, số token người chơi kiếm được.

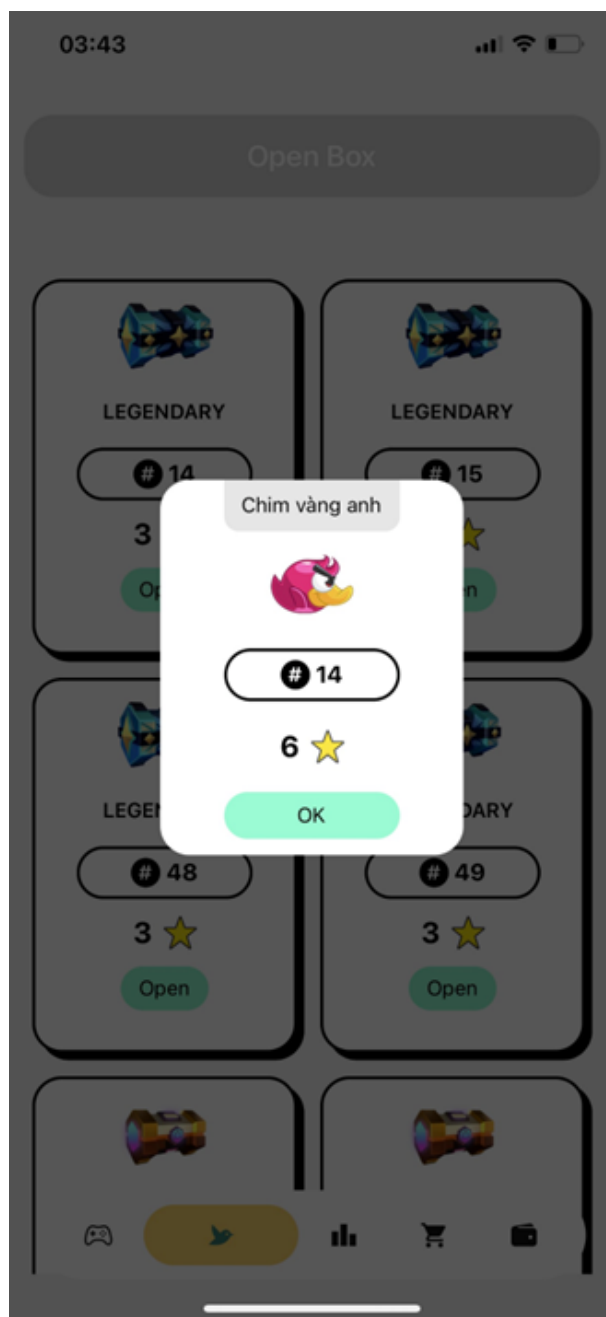
4.3.3 Minh họa các chức năng chính

Hình 4.13 thể hiện giao diện của màn hình Bán Bird dành cho người dùng. Người dùng có thể mang Bird của mình muốn bán để đưa mang bán lên chợ của ứng dụng với số tiền mình mong muốn được nhận về.



Hình 4.13: Giao diện bán Bird

Hình 4.14 thể hiện giao diện của màn hình Mở hộp dành cho người dùng. Người dùng có thể mở các hộp Bird đã mình từ ví rồi chuyển vào trong ứng dụng để mở hộp và nhận được Bird để có thể chơi kiếm tiền.



Hình 4.14: Giao diện Mở hộp

Hình 4.15 thể hiện giao diện của màn hình Thêm ví on-chain. Người dùng có thể thêm ví hoặc tạo ví on-chain mới.

9:39 9:39

CREATE WALLET

Email Verification code

Send code

Seed Phrase

tourist below noodle diamond perfect innocent civil
wasp teach gap loan taxi

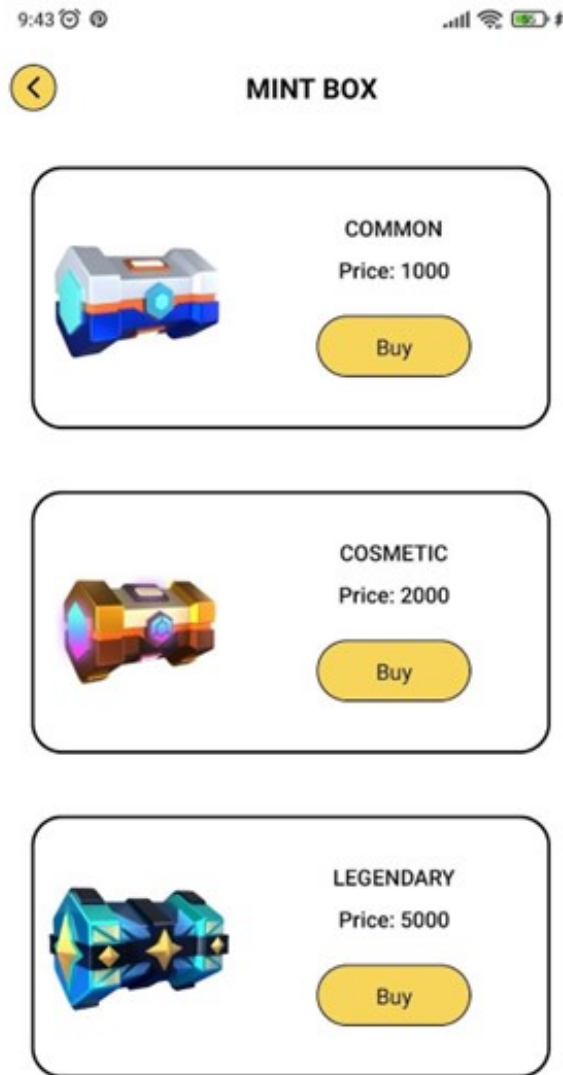
Enter Seed Phrase Again

Enter the Seed Phrase word and separate with space

CREATE WALLET

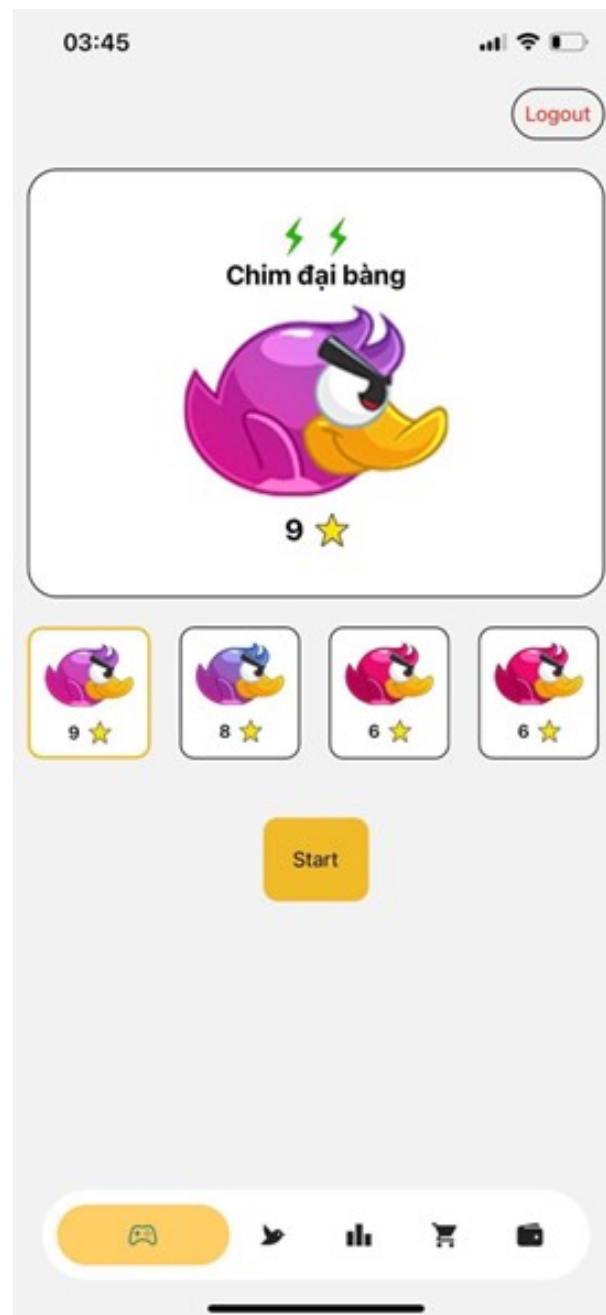
Hình 4.15: Thêm ví on-chain

Hình 4.16 thể hiện giao diện của màn hình Mint Box. Người dùng chọn loại hộp muốn mint.



Hình 4.16: Giao diện Mint Box

Hình 4.17 thể hiện giao diện của màn hình Chơi game. Người dùng Bird để bắt đầu chơi game.



Hình 4.17: Giao diện chơi game

4.4 Kiểm thử

Đồ án thực hiện bước kiểm thử sản phẩm sau cùng bằng kỹ thuật Black-Box. Chi tiết kiểm thử một số chức năng được mô tả phía dưới.

4.4.1 Kiểm thử chức năng Thêm ví on-chain

Bảng 4.13: Kiểm thử chức năng Thêm ví on-chain

Đầu vào	Đầu ra	Kết quả
Nhập email người dùng	Hệ thống gửi mã code về email đã nhập	Đạt

Đầu vào	Đầu ra	Kết quả
Bỏ trống trường seed phrase	Nút “import wallet” bị disable	Đạt
Người dùng nhập đúng 12 từ khóa seed phrase không hợp lệ	Hiển thị thông báo lỗi seed phrase không hợp lệ	Đạt

4.4.2 Kiểm thử chức năng Mint BirdBox

Bảng 4.14: Kiểm thử chức năng Mint BirdBox

Đầu vào	Đầu ra	Kết quả
Đối với địa chỉ ví lần đầu đúc hộp, ứng dụng sẽ hiển thị nút “Approve”, người dùng ấn nút “Approve”	Khi ấn nút “Approve”, hệ thống kiểm tra balance của tài khoản, nếu thỏa mãn thì approve thành công, nút “Approve” chuyển thành nút “Buy”	Đạt
Khi trạng thái của nút là “Buy”, người dùng ấn nút “Buy”, hệ thống sẽ gửi yêu cầu mint box và ví sẽ bị trừ đi số token tương ứng với giá box	Khi ấn nút “Buy”, hệ thống xử lý xong sẽ thông báo mint box thành công, token ở ví bị trừ đi và box sẽ được hiển thị ở màn “My boxes”	Đạt

4.4.3 Kiểm thử chức năng Chuyển hộp từ ví on-chain vào ví in-game

Bảng 4.15: Kiểm thử chức năng Chuyển hộp từ ví on-chain vào ví in-game

Đầu vào	Đầu ra	Kết quả
Trong danh sách “My boxes”, người dùng ấn nút transfer của box mà mình muốn chuyển vào trong ví in-game	Khi ấn nút “Transfer”, NFT box sẽ được chuyển từ ví on-chain vào ví total, và trong ví in-game của người dùng sẽ hiển thị NFT mà đã chuyển.	Đạt

4.4.4 Kiểm thử chức năng Bán Bird trên chợ

Bảng 4.16: Kiểm thử chức năng Bán Bird trên chợ

Đầu vào	Đầu ra	Kết quả
Người dùng không nhập giá bán	Hệ thống thông báo lỗi giá đặt bán không hợp lệ	Đạt
Người dùng nhập giá là một chuỗi không phải số	Hệ thống thông báo lỗi giá đặt bán không hợp lệ	Đạt

Đầu vào	Đầu ra	Kết quả
Sau khi nhập giá bán hợp lệ, người dùng ấn nút “Sell” để đặt bán Bird ở trên chợ	Hệ thống thông báo thành công và Bird được đặt bán sẽ hiển thị ở danh sách Bird ở màn “Market-place”	Đạt

4.4.5 Kiểm thử chức năng Hoán đổi token sang loại tiền điện tử khác

Bảng 4.17: Kiểm thử chức năng Hoán đổi token sang loại tiền điện tử khác

Đầu vào	Đầu ra	Kết quả
Người dùng nhập vào số lượng token lớn hơn số lượng token mà họ có trong ví in-game	Hệ thống thông báo số lượng token phải nhỏ hơn hoặc bằng với số token có trong ví in-game	Đạt
Sau khi nhập số token và chọn loại token muốn swap, người dùng ấn nút “Swap”	Hệ thống nhận yêu cầu và xử lý thành công sẽ trả về thông báo swap thành công, số token bị trừ đi tương ứng và số đồng tiền điện tử được swap thành sẽ được tăng lên	Đạt

4.5 Triển khai

Sau quá trình xây dựng, ứng dụng game NFT Flappy Bird sẽ được triển khai trên appstore và có file apk. Chi tiết quá trình triển khai như sau:

1. Các contract đã được deploy trên testnet của Bince Smart Chain.
2. Phần service API backend đã được triển khai trên Heroku.
3. Hệ quản trị cơ sở dữ liệu MongoDB sử dụng dịch vụ MongoDB Atlas của mLab.
4. Sử dụng thư viện SocketIO của Node.js

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Trong chương 5 sẽ làm nổi bật những giải pháp đã sử dụng và những đóng góp của giải pháp đó cho dự án.

5.1 Smart Contract

Trong game NFT thì phần không thể thiếu đó là Smart Contract.

Smart Contract (hay Hợp đồng thông minh) là các chương trình chạy trên blockchain. Hợp đồng thông minh cũng giống như một hợp đồng kỹ thuật số bị bắt buộc thực hiện bởi một bộ quy tắc cụ thể. Các quy tắc này do bộ mã máy tính xác định trước mà tất cả các nút (node) trong mạng đều phải sao chép và thực thi các quy tắc đó.

Về bản chất, Smart Contract chỉ là một đoạn mã chạy trên một hệ thống phân tán (blockchain), cho phép tạo ra các giao thức Permissionless (tức là không cần trao quyền). Điều đó có nghĩa là:

- Hai bên trong hợp đồng có thể đưa ra các cam kết thông qua blockchain mà không cần phải biết về danh tính hay tin tưởng lẫn nhau.
- Họ có thể đảm bảo rằng nếu các điều kiện của hợp đồng không được thỏa mãn, hợp đồng sẽ không được thực thi.

Ngoài ra, việc sử dụng hợp đồng thông minh loại bỏ nhu cầu đối với các bên trung gian, giúp giảm đáng kể chi phí hoạt động.

5.2 Tạo FBT token

Đối với game truyền thống thì việc thanh toán sẽ phải qua bên thứ ba. Vì vậy em tạo ra một FBT token làm đồng tiền cho game Flappy Bird . Nó là một đồng tiền điện tử vì nó rất tiện lợi. Khi đó có thể nạp, rút tiền, mua bán Bird, mint Box... với đồng tiền này mà không cần bất kỳ bên nào. Và khi cần FBT token không chỉ là mua bán trong game. Người dùng có thể chuyển, bán , swap sang bất kỳ một đồng tiền điện tử nào một cách dễ dàng ngoài ứng dụng.

5.3 NFT hoá Box

Để phát hành nhân vật thì những người làm game đều sẽ có những cách phát hành nhân vật riêng của họ. Như game truyền thống thì thường sẽ là người dùng nạp tiền vào vào mua nhân vật mình muốn rồi sau đó chơi. Để tăng tính thú vị cho game thì em quyết định NFT nhân vật của mình vào một cái hộp. Khi đó người dùng sẽ phải dùng tiền FBT token và một ít phí BNB để mint box. Khi đó sẽ tạo ra một Box NFT chứa nhân vật bên trong. Và khi mint box thành công người dùng có thể chuyển, bán cho người khác hoặc là sẽ chuyển vào game rồi mở hộp đó ra lấy

nhân vật.

5.4 Socket.io

Trong phần game khi người dùng chơi game, tiền thưởng sẽ được trả ngay lập tức khi người dùng vượt qua chúng ngại vật. Thường các game sẽ được trả thưởng khi kết thúc trò chơi nhưng như vậy, khi người dùng đang chơi mất mạng hoặc thoát đột ngột sẽ không nhận được thưởng khi đang vượt qua chúng ngại vật. Vì vậy chúng ta cần xử lý "real- time" là trả thưởng ngay khi người dùng qua chướng ngại vật.

5.5 Tạo ví in-app

Trong dự án này, sử dụng cơ chế 2 ví là ví in-app và ví on-chain. Để bắt đầu chơi game, người dùng cần import ví on-chain vào hệ thống. Sau đó người dùng chuyển token từ ví on-chain vào ví in-app. Để mua Bird trên chợ, người dùng sẽ mua bằng token có trong ví in-app. Người dùng cũng có thể chuyển token từ ví in-app về ví on-chain. Khi sử dụng ví in-game người dùng sẽ không phải trả phí trong mạng lưới Blockchain. Như vậy sẽ giúp người dùng giảm phí giao dịch và tăng tốc độ giao dịch.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Cùng với sự hướng dẫn của TS. Trịnh Anh Phúc, em đã xây dựng được ứng dụng game NFT Flappy Bird. Ứng dụng game đã đáp ứng được những chức năng cơ bản của một game NFT hiện nay. Trong quá trình xây dựng ứng dụng game NFT Flappy Bird, em đã học được quy trình xây dựng một game NFT và hiểu được về ứng dụng của blockchain trong lĩnh vực GameFi.

Các ưu điểm nổi bật của ứng dụng xây dựng có thể kể đến như là :

- Hệ thống tích hợp ví vào trong game giúp người dùng có thể gửi tiền điện tử vào ví để nạp tiền vào tài khoản chi tiêu của họ.
- Play to Earn: Người dùng cần sở hữu tối thiểu 1 Bird có thể chơi game và nhận được token trong game.
- Thị trường giao dịch NFT trong ứng dụng: Người dùng có thể giao dịch NFT của họ trên thị trường giao dịch.
- Hoán đổi trong ứng dụng: Người dùng có thể hoán đổi token thưởng nhận được từ game sang các loại tiền điện tử khác.

Tuy nhiên, trong quá trình xây dựng ứng dụng, do thời gian và kinh phí không cho phép, ứng dụng vẫn còn những nhược điểm như là :

- Khi vào game, còn mất thời gian để khởi tạo dữ liệu dẫn đến trải nghiệm người dùng không tốt.
- Chưa xây dựng ứng dụng web.

6.2 Hướng phát triển

Trong tương lai, khi ứng dụng được đầu tư thời gian cũng như kinh phí xây dựng, em dự định sẽ mở rộng theo các hướng sau:

- Triển khai các contract lên mainnet của Binae Smart Chain
- Tích hợp được nhiều các chain khác của EVM
- Xây dựng cộng đồng: tạo pool IDO, phát hành token cho cộng đồng,...
- Tối ưu code để chạy nhanh hơn khi mới vào app
- Xây dựng game trên cả nền tảng web để tăng thêm lựa chọn cho người dùng.

TÀI LIỆU THAM KHẢO

- [1] *Trích dẫn giáo sư stephen*. [Online]. Available: <https://medium.com/collab-currency/the-future-of-work-5d7b47cff308> (visited on 08/05/2022).
- [2] *Trích dẫn bài khảo sát*. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3068303 (visited on 08/05/2022).
- [3] *Learn solidity*. [Online]. Available: <https://docs.soliditylang.org/en/v0.8.15/> (visited on 07/30/2022).
- [4] *Learn javascript*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (visited on 07/30/2022).
- [5] P. Lynch and J. Rothchild, *Learn to Earn: A Beginner's Guide to the Basics of Investing and*. Simon and Schuster, 2012.
- [6] S. B. De Jesus, D. Austria, D. R. Marcelo, C. Ocampo, A. J. Tibudan, and J. Tus, "Play-to-earn: A qualitative analysis of the experiences and challenges faced by axie infinity online gamers amidst the covid-19 pandemic,"
- [7] L. V. Kiong, *DeFi, NFT and GameFi Made Easy: A Beginner's Guide to Understanding and Investing in DeFi, NFT and GameFi Projects*. Liew Voon Kiong, 2021.
- [8] D. A. Aguila, J. M. Bartolata, and J. G. Estrañero, "Axeing the axie infinity (ai): The ai of modern gaming, business model strategem, and global economy towards cryptocurrency era,"
- [9] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, vol. 14, no. 4, pp. 352–375, 2018.
- [10] S. Gupta and M. Sadoghi, "Blockchain transaction processing," *arXiv preprint arXiv:2107.11592*, 2021.
- [11] S. Abdullah, S. Rothenberg, E. Siegel, and W. Kim, "School of block—review of blockchain for the radiologists," *Academic radiology*, vol. 27, no. 1, pp. 47–57, 2020.
- [12] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *arXiv preprint arXiv:2105.07447*, 2021.
- [13] J. Y. Lee, "A decentralized token economy: How blockchain and cryptocurrency can revolutionize business," *Business Horizons*, vol. 62, no. 6, pp. 773–784, 2019.

- [14] Z. Zheng, S. Xie, H.-N. Dai, *et al.*, “An overview on smart contracts: Challenges, advances and platforms,” *Future Generation Computer Systems*, vol. 105, pp. 475–491, 2020.
- [15] Y. Fu, M. Ren, F. Ma, Y. Jiang, H. Shi, and J. Sun, “Evmfuzz: Differential fuzz testing of ethereum virtual machine,” *arXiv preprint arXiv:1903.08483*, 2019.
- [16] G. Wood *et al.*, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
- [17] M. E. Hellman, “An overview of public key cryptography,” *IEEE Communications Magazine*, vol. 40, no. 5, pp. 42–49, 2002.
- [18] *Learn expressjs*. [Online]. Available: <https://expressjs.com/> (visited on 07/30/2022).
- [19] *Learn mongoose*. [Online]. Available: <https://mongoosejs.com/> (visited on 07/30/2022).
- [20] S. Tiwari, *Professional nosql*. John Wiley & Sons, 2011.
- [21] *Learn react-native*. [Online]. Available: <https://reactnative.dev/> (visited on 07/30/2022).