TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**BÁO CÁO CUỐI KÌ MÔN KIẾN TRÚC HƯỚNG DỊCH VỤ**

*Người thực hiện:* **TRẦN QUỐC VINH - 52000823**

Lớp: **200503401**

Khóa: **24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Logo

Description automatically generated

**BÁO CÁO CUỐI KÌ MÔN KIẾN TRÚC HƯỚNG DỊCH VỤ**

*Người thực hiện:* **TRẦN QUỐC VINH - 52000823**

Lớp: **200503401**

Khóa: **24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

*TP. Hồ Chí Minh, ngày 31 tháng 3 năm 2024*

*Tác giả*

*Trần Quốc Vinh*

BÁO CÁO ĐƯỢC HOÀN THÀNH

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Khóa luận/Đồ án tốt nghiệp còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Báo cáo cuối kỳ của mình**. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 31 tháng 3 năm 2024*

*Tác giả*

*Trần Quốc Vinh*

**TÓM TẮT**

# **MỤC LỤC**

[**TÓM TẮT** 5](#_Toc166336675)

[**MỤC LỤC** 6](#_Toc166336676)

[**DANH MỤC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ** 7](#_Toc166336677)

[1 CHƯƠNG 1: TỔNG QUAN HỆ THỐNG 9](#_Toc166336678)

[1.1 Giới thiệu đề tài 9](#_Toc166336679)

[1.2 Đặc tả hệ thống 10](#_Toc166336680)

[1.2.1 Giới thiệu về nền tảng Themis 10](#_Toc166336681)

[1.2.2 Kiến trúc hệ thống Themis 11](#_Toc166336682)

[**1.3** Phạm vi đề tài 11](#_Toc166336683)

[**1.3.1** Giới hạn về đối tượng và chức năng 11](#_Toc166336684)

[**1.3.2** Giới hạn về công nghệ 11](#_Toc166336685)

[**1.4** Ý nghĩa thực tiễn 12](#_Toc166336686)

[**2** CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 12](#_Toc166336687)

[**2.1** Yêu cầu chức năng của hệ thống 12](#_Toc166336688)

[2.2 Lược đồ chức năng của hệ thống 14](#_Toc166336689)

[2.2.1 Lược đồ Use Case 14](#_Toc166336690)

[2.2.2 Đặc tả Use Case 14](#_Toc166336691)

[2.3 Lược đồ ER của hệ thống 27](#_Toc166336692)

[3 CHƯƠNG 3: HIỆN THỰC HỆ THỐNG 28](#_Toc166336693)

[3.1 Tổng quan và cách chạy hệ thống 28](#_Toc166336694)

[3.2 Giao diện hệ thống 30](#_Toc166336695)

[3.2.1 End User UI 30](#_Toc166336696)

[3.2.2 Admin UI 32](#_Toc166336697)

[3.3 Danh sách API và document 33](#_Toc166336698)

[**TÀI LIỆU THAM KHẢO** 33](#_Toc166336699)

**DANH MỤC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

[Hình 1 1 Chi phí khi giao dịch bđs truyền thống 10](#_Toc166274755)

[Hình 1 2 Kiến trúc hệ thống Themis 11](#_Toc166274756)

[Hình 2 1 Lược đồ Use case tổng quát 14](#_Toc166274757)

[Hình 2 2 Lược đồ Use case Admin verify property 14](#_Toc166274758)

[Hình 2 3 Lược đồ Use case Admin listing property 16](#_Toc166274759)

[Hình 2 4 Lược đồ Use case User Manage Property 18](#_Toc166274760)

[Hình 2 5 Lược đồ Use case User Request Listing Property 20](#_Toc166274761)

[Hình 2 6 Lược đồ Use case buy or sell token 22](#_Toc166274762)

[Hình 2 7 Lược đồ Use case Withdraw Rental Payment 24](#_Toc166274763)

[Hình 2 8 Lược đồ Use case view or cancel orders 26](#_Toc166274764)

[Hình 2 9: Lược đồ ER hiện thực ở mức vật lý 28](#_Toc166274765)

[Hình 3. 1: 3 schema DB của hệ thống trên Azure MySQL 28](#_Toc166274766)

[Hình 3. 2 Danh sách microservice 29](#_Toc166274767)

[Hình 3. 3 Frontend microservice 29](#_Toc166274768)

[Hình 3. 4 Cấu trúc folder microservice dùng NodeJS 30](#_Toc166274769)

[Hình 3. 5 Cấu trúc folder microservice dùng Flask 30](#_Toc166274770)

[Hình 3. 6 UI Marketplace 31](#_Toc166274771)

[Hình 3. 7 UI Token Ownership and Rental Income 31](#_Toc166274772)

[Hình 3. 8 UI My Property 32](#_Toc166274773)

[Hình 3. 9 Giao diện Admin UI 32](#_Toc166274774)

[Hình 3. 10 Danh sách API trên Swagger 33](#_Toc166274775)

[Bảng 1 1 Lợi ích của nền tảng Themis 11](#_Toc166274776)

[Bảng 2 1 Yêu cầu chức năng của hệ thống 12](#_Toc166274777)

[Bảng 2 2 Yêu cầu phi chức năng của hệ thống 13](#_Toc166274778)

[Bảng 2 3 Usecase Verify Property 16](#_Toc166274779)

[Bảng 2 4 Listing Property 18](#_Toc166274780)

[Bảng 2 5 User Manage Property 20](#_Toc166274781)

[Bảng 2 6 Use case Listing Property 22](#_Toc166274782)

[Bảng 2 7 Use case buy or sell token 24](#_Toc166274783)

[Bảng 2 8 Use case withraw rental payment 26](#_Toc166274784)

[Bảng 2 9: Use case view or cancel orders 27](#_Toc166274785)

# CHƯƠNG 1: TỔNG QUAN HỆ THỐNG

## Giới thiệu đề tài

Bất động sản Việt Nam đang trải qua giai đoạn phát triển mạnh mẽ, đặc biệt là tại các thành phố lớn như Hà Nội và TP.Hồ Chí Minh. Giá nhà và đất tăng đột biến, tạo ra cơ hội đầu tư và tham gia vào thị trường cho nhà đầu tư trong và ngoài nước.

**Tuy nhiên các giao dịch trong bất động sản truyền thống cũng thường gặp phải nhiều thách thức như:**

* **Số lượng bên thứ ba cao**: BĐS là loại tài sản có giá trị cao và gắn với nhiều vấn đề pháp lý. Do đó các giao dịch liên quan tới BĐS cũng yêu cầu nhiều bên thứ ba tham gia
  + Ngân hàng: hỗ trợ các khoản vay
  + Luật sư/Công chứng viên: xử lý các loại giấy tờ pháp lý
  + Trung tâm môi giới: hỗ trợ người mua và bán trong quá trình giao dịch
  + Các dịch vụ thẩm định/định giá nhà
  + Cơ quan chính phủ

=> Mỗi một bên thứ ba đều yêu cầu lượng phí nhất định, dẫn đến chi phí giao dịch cuối cùng cho cả người mua lẫn bán là rất cao và tốn kém thời gian do phụ thuộc vào bên thứ ba

* **Thiếu thụt thông tin giữa người mua và bán**: Đối với các loại tài sản khác như cổ phiếu/trái phiếu, tìm kiếm thông tin giúp người mua đưa ra quyết định là tương đối dễ dàng. Việc này có thể thực hiện thông qua mạng hoặc các trang báo online như Bloomberg, The Morning Star. Tuy nhiên trong BĐS thì nguồn thông tin chính của người mua là do người bán cung cấp Þ Dẫn tới người bán có thể cung cấp thông tin sai lệch, thiếu thụt và làm ảnh hưởng tới quá trình thượng lượng giá
* **Thủ tục giấy tờ tốn thời gian và chi phí:** Quy trình giấy tờ phức tạp và liên quan tới nhiều bên thứ ba khiến cho giao dịch mất nhiều thời gian, tiền bạc. Việc xử lý nhiều giấy tờ cũng tăng nguy cơ xảy ra sai sót và làm trễ tiến độ giao dịch.
* **Chi phí tham gia vào thị trường BĐS ban đầu cho người mua/bán là rất cao**: Chi phí đầu vào lớn đối với người mua/bán là một thách thức lớn. Người mua phải đối mặt với khoản chi phí như tiền cọc, phí cho trung tâm môi giới. Trong khi đó người bán có thể phải chịu các chi phí liên quan đến việc chuẩn bị và quảng bá bất động sản

A screenshot of a computer

Description automatically generated

Hình 1 1 Chi phí khi giao dịch bđs truyền thống

=> Mỗi một bên thứ ba đều yêu cầu lượng phí nhất định, dẫn đến chi phí giao dịch cuối cùng cho cả người mua lẫn bán là rất cao và tốn kém thời gian do phụ thuộc vào bên thứ ba

## Đặc tả hệ thống

### Giới thiệu về nền tảng Themis

Thấy được các thách thức trên nên tôi xin giới thiệu nền tảng Themis. Themis là một nền tảng giao dịch bất động sản trực tuyến nhắm vào thị trường **Fractional Real Estate Investing**, được xây dựng nhằm giải quyết những thách thức của thị trường bất động sản truyền thống tại Việt Nam. Nền tảng của chúng tôi cung cấp một giải pháp toàn diện cho cả người mua và bán, bao gồm

|  |  |
| --- | --- |
| Giao dịch trực tiếp | * Loại bỏ các bên thứ ba không cần thiết, giúp giảm chi phí giao dịch và tiết kiệm thời gian cho cả người mua và bán. * Nền tảng hỗ trợ thanh toán trực tuyến an toàn và minh bạch. |
| Truy cập thông tin minh bạch | * Cung cấp thông tin đầy đủ và chính xác về bất động sản, bao gồm giá cả, lịch sử giao dịch, tình trạng pháp lý, v.v. * Giúp người mua đưa ra quyết định đầu tư sáng suốt |
| Quy trình giao dịch đơn giản | * Tự động hóa các thủ tục giấy tờ, giúp rút ngắn thời gian giao dịch và giảm thiểu sai sót. * Cung cấp dịch vụ hỗ trợ khách hàng chuyên nghiệp suốt quá trình giao dịch |
| Mở rộng cơ hội đầu tư | * Cho phép người mua đầu tư vào bất động sản với số vốn ban đầu thấp. * Cung cấp đa dạng các sản phẩm bất động sản để đáp ứng nhu cầu của mọi nhà đầu tư |

Bảng 1 1 Lợi ích của nền tảng Themis

Một số điểm chính về Themis:

* Đầu tư dễ dàng: Bắt đầu với số tiền thấp
* Đa dạng hóa danh mục đầu tư: Đầu tư vào nhiều bất động sản để giảm thiểu rủi ro
* Thu nhập thụ động: Nhận tiền thuê nhà từ các bất động sản
* Tăng trưởng: Kiếm tiền từ việc tăng giá trị (appreciation) của bất động sản
* Được quản lý: Themis hợp tác với các bên thứ 3 để xử lý tất cả hoạt động của bất động sản, bạn chỉ cần mua và bán

**Hệ thống Themis được lấy cảm hứng từ 2 web phía dưới**, nếu bạn muốn xem chi tiết cách hoạt động có thể xem qua đây

[Arrived Homes Help Center](https://help.arrived.com/en/)

[What is Lofty? | Lofty Learning Center](https://learn.lofty.ai/en/collections/2200544-what-is-lofty)

### Kiến trúc hệ thống Themis

Dự án bao gồm 3 backend microservices chính:

* **Offer Service**: Quản lý API liên quan tới tạo lệnh mua/bán và khớp lệnh share (cổ phiếu) của bất động sản
* **Rental Service**: Quản lý API liên quan tới tiền thuê nhà và rút tiền từ ví thuê về lại ví chính
* **User Management Service**: Quản lý API liên quan tới quản lý user/admin và thông tin bất động sản của user

A screenshot of a computer

Description automatically generated

Hình 1 2 Kiến trúc hệ thống Themis

Ngoài ra có thêm 2 folder Frontend là

* **Admin UI**: chứa code Frontend cho giao diện của Admin
* **End User UI**: chứa code Frontend cho giao diện của User

## Phạm vi đề tài

### Giới hạn về đối tượng và chức năng

Do đây chỉ là phiên bản POC, scope của dự án tương đối nhỏ nên hệ thống UI được thiết kế chỉ dành admin và người dùng cuối. Hoàn toàn không có các giao diện cho các bên thứ 3 có liên quan như cung cấp dịch vụ (dịch vụ định giá, dịch vụ đánh giá nhà,)

### Giới hạn về công nghệ

Dự án này sử dụng các công nghệ sau:

* **Backend**: Flask (Phù hợp với các dự án nhỏ và vừa do không hỗ trợ async) và NodeJs
* **Database**: Azure MySQL
* **Frontend**: HTML,CSS,JS /React
* **Cloud**: Azure Database for MySQL
* **Documentation**: SwaggerAPI
* **Phương thức liên lạc giữa các Microservice**: Request-Response

## Ý nghĩa thực tiễn

* Thông qua dự án lần này, tôi đã giúp cho bản thân mình hiểu được thêm về cách thức hoạt động của nền tảng giao dịch BĐS, Data model trong đó thường bao gồm những thông tin gì và liên kết với nhau như thế nào, quy trình mua/bán nhà như thế nào, kiến thức về Fractional Real Estate Investing
* Tôi còn biết được thêm về quy trình phát triển phần mềm nói chung và các công nghệ liên quan tới phát triển web app

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Yêu cầu chức năng của hệ thống

|  |  |
| --- | --- |
| **Yêu cầu chức năng** | Hệ thống hỗ trợ người dùng đăng ký/đăng nhập  Hệ thống hỗ trợ quản lý property  Hệ thống hỗ trợ đưa property lên marketplace  Hệ thống hỗ trợ mua và bán token  Hệ thống hỗ trợ tìm property theo nhu cầu  Hệ thống hỗ trợ xem token đang sở hữu và rút tiền thuê  Hệ thống hỗ trợ xem các giao dịch  Hệ thống hỗ trợ verify và listing property  Hệ thống hỗ trợ xem lịch sử verify và listing property |

Bảng 2 1 Yêu cầu chức năng của hệ thống

|  |  |
| --- | --- |
| **Yêu cầu phi chức năng** | Hiệu suất hoạt động hệ thống  Sao lưu dữ liệu  Cho phép người dùng tương tác dễ dàng  Bảo mật thông tin khách hàng  Khả năng tương thích với đa nền tảng  Khả năng bảo trì hệ thống |

Bảng 2 2 Yêu cầu phi chức năng của hệ thống

**Tính bảo mật và phân quyền cho user**: Hệ thống yêu cầu user phải đăng nhập bằng multifactor-authentication trước khi sử dụng. Giao diện hệ thống phụ thuộc vào tài khoản mà user đăng nhập.

**Hiệu năng cao (performance):** Với điều kiện kết nốt mạng ổn định, hệ thống phải có khả năng xử lý requests của ít nhất 100.000 users cùng 1 lúc đăng nhập và sử dụng

**Khả năng mở rộng (scalability):** Nếu quy mô công ty và số lượng user tăng lên, hệ thống phải có khả năng mở rộng tới tối đa 10.000.000 users cùng 1 lúc đăng nhập và sử dụng

**Khả năng tương thích với đa nền tảng (compatiablity):** Hệ thống cho phép đăng nhập trên nhiều nền tảng hệ điều hành khác nhau như Window, Android, IOS và nhiều thiết bị như PC, Tablet, Smartphone

**Khả năng bảo trì hệ thống (maintainability):** Hệ thống phải được viết dưới dạng từng module, mỗi module là độc lập và không phụ thuộc vào nhau nhằm hỗ trợ mục đích phát triển và bảo trì dễ dàng

**Tính sẵn dùng (availability):** Hệ thống phải hoạt động 24/7 với tính khả dụng tối thiểu (minium availability) là 95%. Thời gian downtime hằng tuần không quá 5 tiếng

## Lược đồ chức năng của hệ thống

### Lược đồ Use Case

A diagram of a property

Description automatically generated

Hình 2 1 Lược đồ Use case tổng quát

### Đặc tả Use Case

A diagram of a company

Description automatically generated

Hình 2 2 Lược đồ Use case Admin verify property

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-1 | |
| **Use Case** | Verify Property | |
| **Scenario** | Chọn chức năng quản lý property | |
| **Trigger Event** | Admin chọn chức năng quản lý property | |
| **Description** | Admin đăng nhập vào hệ thống để chọn chức năng verify property | |
| **Actor(s)** | Admin | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện đăng nhập | |
| **Post-condition(s)** | Admin đăng nhập thành công vào hệ thống | |
| **Flow of event** | Actor | System |
| 1. Chọn mục Property tại sidebar | 1.1 Chuyển sang giao diện quản lý Property  1.2 Fetch danh sách các property được yêu cầu verify từ user. |
| 2. Chọn nút Verify  2.1 Chọn pass và nhấn accept  2.2 Chọn fail và nhấn accpet | 2. Hệ thống hiển thị modal gồm option: pass hoặc fail  2.1 Hệ thống cập nhật property đã được verified  2.2 Hệ thống cập nhật property chưa được verified và gửi mail về cho user |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 3 Usecase Verify Property

A diagram of a company

Description automatically generated

Hình 2 3 Lược đồ Use case Admin listing property

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-2 | |
| **Use Case** | Listing Property | |
| **Scenario** | Chọn chức năng quản lý property | |
| **Trigger Event** | Admin chọn chức năng quản lý property | |
| **Description** | Admin đăng nhập vào hệ thống để chọn chức năng listing property | |
| **Actor(s)** | Admin | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện đăng nhập | |
| **Post-condition(s)** | Admin đăng nhập thành công vào hệ thống | |
| **Flow of event** | Actor | System |
| 1. Chọn mục Property tại sidebar | 1.1 Chuyển sang giao diện quản lý Property  1.2 Fetch danh sách các property được yêu cầu listing từ user |
| 2. Chọn nút Listing  2.1 Chọn pass và nhấn accept  2.2 Chọn fail và nhấn accpet | 2. Hệ thống hiển thị modal gồm option: pass hoặc fail  2.1 Hệ thống cập nhật property đã được listed  2.2 Hệ thống cập nhật property chưa được listed và gửi mail về cho user |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 4 Listing Property

A diagram of a diagram

Description automatically generated

Hình 2 4 Lược đồ Use case User Manage Property

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-3 | |
| **Use Case** | Manage Property | |
| **Scenario** | Chọn tạo mới một property | |
| **Trigger Event** | User nhấn nút chọn tạo mới một property | |
| **Description** | User đăng nhập vào hệ thống để chọn chức năng tạo mới property | |
| **Actor(s)** | User | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện chức năng | |
| **Post-condition(s)** | User đăng nhập thành công vào hệ thống và đang ở trang My Property | |
| **Flow of event** | Actor | System |
| 1. Nhấn nút tạo mới property  1.2 Nhấn submit | 1.1 Hiển thị modal để nhập thông tin property  1.3 Hệ thống kiểm tra nếu thiếu thông tin sẽ báo lỗi. Ngược lại, sẽ tạo mới property và hiển thị lên giao diện |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 5 User Manage Property

A diagram of a process

Description automatically generated

Hình 2 5 Lược đồ Use case User Request Listing Property

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-4 | |
| **Use Case** | Listing Property | |
| **Scenario** | Yêu cầu đưa property lên marketplace | |
| **Trigger Event** | User nhấn vào nút listing property nếu nó khả dụng  User nhấn vào nút listing property | |
| **Description** | User nhấn vào nút verify property trên property card. Sau khi property được verify thì nút listing property sẽ khả dụng. Khi người dùng nhấn vào thì sẽ chọn tiếp các dịch vụ của bên thứ 3. Cuối cùng nhấn accept và đợi phản hồi từ admin. | |
| **Actor(s)** | User | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện chức năng | |
| **Post-condition(s)** | User đăng nhập thành công vào hệ thống và đang ở trang My Property. Đã có property trong danh sách | |
| **Flow of event** | Actor | System |
| 1. User nhấn nút verify tại property card  1.2 User nhấn accept | 1.1 Hiển thị modal để confirm yêu cầu từ user  1.3 Hệ thống chuyển trạng thái của property sang pending. |
| 2.1 User nhấn listing property  2.3 Nhấn chọn các dịch vụ và chọn accept | 2. Khi được verified bởi Admin thì hệ thống chuyển trạng thái property sang verified và nút listing property sẽ khả dụng.  2.2 Hệ thống hiện modal yêu cầu chọn các dịch vụ của bên thứ 3  2.4 Hệ thống kiểm tra và trừ cash balance của người dùng, chuyển trạng thái property sang pending. Khi các dịch vụ được mark passed bởi admin thì property sẽ lên marketplace. Nếu không thể listing thì sẽ thông báo về mail cho người dùng kèm lí do. |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 6 Use case Listing Property

A diagram of a person with text

Description automatically generated with medium confidence

Hình 2 6 Lược đồ Use case buy or sell token

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-5 | |
| **Use Case** | Sell/buy token | |
| **Scenario** | Bán token sau khi property được đưa lên marketplace | |
| **Trigger Event** | Tại giao diện chi tiết property, người dùng nhấn nút sell | |
| **Description** | Người dùng nhấn nút sell property, chọn số lượng token và giá | |
| **Actor(s)** | User | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện chức năng | |
| **Post-condition(s)** | User đăng nhập thành công vào hệ thống và đang ở trang chi tiết property. Current token phải khả dụng. | |
| **Flow of event** | Actor | System |
| 1. User nhấn nút sell/buy  1.2 User nhập đầy đủ thông tin và nhấn submit | 1.1 Hiển thị modal yêu cầu user nhập số lượng token và giá  1.3 Hệ thống cập nhật token balance của user và mục order book của property |
|  | 2. Nếu tồn tại một cặp order khớp lệnh thì sẽ thực hiện transaction và cập nhật trạng thái completed trong mục my order |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 7 Use case buy or sell token

A diagram of a payment method

Description automatically generated

Hình 2 7 Lược đồ Use case Withdraw Rental Payment

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-6 | |
| **Use Case** | Withdraw Rental Payment | |
| **Scenario** | Rút tiền thuê hàng từ các token đã bán | |
| **Trigger Event** | Tại giao diện token ownership and rental payment | |
| **Description** | Người dùng nhấn nút withdraw để chuyển lợi nhuận về cash balance | |
| **Actor(s)** | User | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện chức năng | |
| **Post-condition(s)** | User đăng nhập thành công vào hệ thống và đang ở trang token ownership and rental payment | |
| **Flow of event** | Actor | System |
| 1. User nhấn nút withdraw | 1.1 Hệ thống chuyển số dư về tài khoản user, cập nhật cash balance và số tiền thuê của các token.  1.3 Hệ thống hiển thị thông báo thành công |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 8 Use case withraw rental payment

A diagram of a process

Description automatically generated

Hình 2 8 Lược đồ Use case view or cancel orders

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | UC-6 | |
| **Use Case** | View/cancel order | |
| **Scenario** | User có thể xem và hủy các order | |
| **Trigger Event** | Nhấn vào mục my order để xem danh sách các order. Nhấn vào nút cancel để hủy order | |
| **Description** | User nhấn vào mục my order để xem chi tiết các order. Sau đó, nếu muốn hủy order thì nhấn nút cancel | |
| **Actor(s)** | User | |
| **Pre-condition(s)** | Tài khoản khả dụng và được kết nối internet khi thực hiện đăng nhập | |
| **Post-condition(s)** | User đã đăng nhập thành công vào hệ thống | |
| **Flow of event** | Actor | System |
| 1. User nhấn vào mục my order | 1.1 Hiển thị danh sách các order |
| 2. User nhấn cancel order | 2.1 Hệ thống cập nhật order status |
| **Exception** | Hệ thống xảy ra lỗi trong quá trình xử lý dữ liệu và hiển thị thông báo lỗi | |

Bảng 2 9: Use case view or cancel orders

## Lược đồ ER của hệ thống

Để xem ảnh rõ hơn có thể vào folder erd trong Git repo

A screenshot of a computer

Description automatically generated

Hình 2 9: Lược đồ ER hiện thực ở mức vật lý

# CHƯƠNG 3: HIỆN THỰC HỆ THỐNG

Link Github của dự án, phiên bản microservice: [VinhQuocTran/Microservice-FinalTerm (github.com)](https://github.com/VinhQuocTran/Microservice-FinalTerm)

Phiên bản Blockchain: [VinhQuocTran/Finalterm-Real-Estate-Blockchain (github.com)](https://github.com/VinhQuocTran/Finalterm-Real-Estate-Blockchain)

## Tổng quan và cách chạy hệ thống

Dự án được hiện thực bằng các công nghệ sau:

* **Backend**: Flask và NodeJs
* **Database**: Azure MySQL
* **Frontend**: HTML,CSS,JS /React
* **Cloud**: Azure Database for MySQL
* **Documentation**: SwaggerAPI
* **Phương thức liên lạc giữa các Microservice**: Request-Response

A screenshot of a computer

Description automatically generated

Hình 3. 1: 3 schema DB của hệ thống trên Azure MySQL

Bạn cần khởi động 5 folder (3 backend và 2 frontend) để chạy hệ thống Themis

* Folder admin\_v2\_ui: **npm run dev**
* Folder end\_user\_ui: **npm run dev**
* Folder offer\_service: **npm run start**
* Folder rental\_service:
  + Kích hoạt venv: **.venv\Scripts\activate**
  + Chạy Flask server: **python main.py**
* Folder user\_management\_service: **npm run start**

Tổng cộng 3 microservice (không tính UI) trong hệ thống: Rental Service, Offer Service, User Management Service

A screenshot of a computer screen

Description automatically generated

Hình 3. 2 Danh sách microservice

Đối với Frontend: Chứa các file template html, css... liên quan tới UI

A screenshot of a computer screen

Description automatically generated

Hình 3. 3 Frontend microservice

Cấu trúc folder của 2 microservice (User Management và Offer Service) sử dụng NodeJS

A screenshot of a computer screen

Description automatically generated

Hình 3. 4 Cấu trúc folder microservice dùng NodeJS

Cấu trúc folder của 1 Rental microservice sử dụng Flask

A screenshot of a computer screen

Description automatically generated

Hình 3. 5 Cấu trúc folder microservice dùng Flask

Videe demo hệ thống phiên bản blockchain: [Lofty Application Demo (youtube.com)](https://www.youtube.com/watch?v=hpG0Eyt1if8)

## Giao diện hệ thống

### End User UI

**UI Marketplace**: User có thể xem danh sách các bất động sản được list lên sàn để bán và giá token của bất động sản đó

A screenshot of a website

Description automatically generated

Hình 3. 6 UI Marketplace

**UI Token Ownership and Rental Income**: Xem số token sỡ hữu và tổng tiền thuê nhận được từ token của bất động sản, ngoài ra còn chức năng rút tiền từ ví thuê về ví chính

A screenshot of a computer

Description automatically generated

Hình 3. 7 UI Token Ownership and Rental Income

**UI My Property**: Quản lý Property của User. Mỗi Property sẽ có 2 trạng thái: Verified và Listed. 1 bất động sản nếu muốn được listing thì phải được verify trước. Quá trình verify và listing nằm bên giao diện Admin

A screenshot of a computer

Description automatically generated

Hình 3. 8 UI My Property

### Admin UI

A screenshot of a computer

Description automatically generated

Hình 3. 9 Giao diện Admin UI

## Danh sách API và document

Cách truy cập vào API docs được auto generate bởi Swagger của User Management Microservice

* Vào folder “user\_management\_service”: **npm run start**
* Vào browser theo đường link: [Swagger UI](http://localhost:3000/api-docs/#/)

A screenshot of a computer

Description automatically generated

Hình 3. 10 Danh sách API trên Swagger

Danh sách xem chi tiết tại link swagger trên

**TÀI LIỆU THAM KHẢO**

1. [CS504070 — Service-Oriented Architecture [Phuc H. Duong] (duonghuuphuc.com)](https://cs504070.duonghuuphuc.com/)
2. [Arrived Homes Help Center](https://help.arrived.com/en/)
3. [What is Lofty? | Lofty Learning Center](https://learn.lofty.ai/en/collections/2200544-what-is-lofty)