Project: Thiết kế website thương mại điện tử Octopus

# Mô tả khái quát về nội dung đề tài

Đồ án này phát triển một website thương mại điện tử chuyên mua bán các vật phẩm ảo như key bản quyền phần mềm (Windows, Office, Photoshop), tài khoản số (Netflix, Facebook, Spotify, Canva, v.v.). Vấn đề đặt ra là hiện nay, thị trường giao dịch các loại tài khoản số và key bản quyền còn nhiều bất cập, thiếu nền tảng tập trung để đảm bảo giao dịch an toàn, minh bạch giữa người mua và người bán. Phương pháp thực hiện bao gồm xây dựng một hệ thống web với các chức năng chính như quản lý tài khoản, sản phẩm, đơn hàng và thanh toán trực tuyến.

Công cụ sử dụng bao gồm ReactJS (Vite, Redux) cho frontend, Node.js + Express.js cho backend, MongoDB làm cơ sở dữ liệu, và JWT Authentication để bảo mật.

Kết quả đạt được là một nền tảng hoạt động ổn định, đáp ứng nhu cầu mua bán tài khoản số một cách an toàn và tiện lợi. Hệ thống hỗ trợ người mua đặt hàng nhanh chóng, người bán dễ dàng đăng bán sản phẩm, và quản trị viên kiểm soát nội dung, giao dịch.

Về tính thực tế, website có thể áp dụng ngay vào thị trường, mở rộng với tích hợp cổng thanh toán, hỗ trợ nhiều loại tài khoản số hơn. Qua đồ án, sinh viên đã nâng cao kỹ năng lập trình web, thiết kế hệ thống, xử lý bảo mật và tối ưu hiệu suất website.

Các mục tiêu cụ thể của dự án bao gồm:

1. Xây dựng một nền tảng hoàn chỉnh sử dụng công nghệ MERN Stack (MongoDB, Express.js, React, Node.js) để đảm bảo hiệu suất cao, khả năng mở rộng và trải nghiệm người dùng mượt mà.
2. Phát triển hệ thống quản lý tài khoản, danh mục sản phẩm, và các cơ chế bảo mật để đảm bảo tính toàn vẹn của thông tin và giao dịch.
3. Thiết kế cơ chế giao tài khoản tự động sau khi thanh toán, giảm thiểu sự can thiệp thủ công và tăng tính tiện lợi cho người dùng.
4. Triển khai hệ thống thanh toán đa dạng, tích hợp nhiều phương thức phổ biến tại Việt Nam và quốc tế.
5. Xây dựng hệ thống báo cáo và thống kê chi tiết cho người quản trị, giúp nắm bắt tình hình kinh doanh và đưa ra quyết định phù hợp.

Phạm vi của đồ án bao gồm:

* Phát triển đầy đủ cả frontend và backend của nền tảng.
* Thiết kế và triển khai cơ sở dữ liệu phi quan hệ với MongoDB.
* Xây dựng API RESTful cho tất cả các chức năng của hệ thống.
* Tích hợp các cổng thanh toán PayOS, hỗ trợ thanh toán bảo mật.
* Triển khai các biện pháp bảo mật cho dữ liệu người dùng và thông tin tài khoản.
* Tối ưu hóa hiệu suất và trải nghiệm người dùng trên nhiều thiết bị.

### Giới thiệu về mô hình kinh doanh bán hàng trực tuyến

Mô hình kinh doanh bán tài khoản trực tuyến là một phân khúc đặc biệt trong thị trường thương mại điện tử, tập trung vào việc cung cấp quyền truy cập vào các dịch vụ số thông qua việc chuyển giao thông tin đăng nhập. Lĩnh vực này đã phát triển mạnh mẽ trong những năm gần đây, phản ánh nhu cầu ngày càng tăng của người dùng đối với các dịch vụ giải trí, mạng xã hội và tiện ích trực tuyến.

Các hình thức kinh doanh chính trong mô hình này bao gồm:

1. **Bán tài khoản Premium/VIP**: Cung cấp tài khoản có đặc quyền cao cấp cho các dịch vụ streaming như Netflix, Spotify, hay Disney+, thường với giá thấp hơn so với đăng ký trực tiếp.
2. **Chia sẻ tài khoản**: Cho phép nhiều người dùng chia sẻ chi phí của một tài khoản cao cấp, tận dụng khả năng đa người dùng của nhiều nền tảng.
3. **Tài khoản theo thời hạn**: Cung cấp quyền truy cập trong một khoảng thời gian nhất định, từ một tháng đến một năm, với nhiều lựa chọn giá cả.
4. **Tài khoản mạng xã hội**: Bán các tài khoản Facebook, Instagram, Twitter với lượng người theo dõi hoặc tương tác cao cho mục đích marketing.

Dự án này nhằm mục đích hợp pháp hóa và chuẩn hóa thị trường này thông qua:

* Xác minh nguồn gốc hợp pháp của các tài khoản
* Bảo vệ thông tin cá nhân của người bán và người mua
* Đảm bảo tính minh bạch trong giao dịch
* Cung cấp các chính sách bảo hành và hỗ trợ sau bán hàng
* Tuân thủ các quy định pháp luật liên quan đến thương mại điện tử và dịch vụ số

Hệ thống sẽ thu lợi nhuận từ các nguồn:

* Phí hoa hồng từ mỗi giao dịch thành công
* Gói đăng ký cao cấp cho người bán (để tiếp cận nhiều khách hàng hơn)
* Dịch vụ quảng cáo và đẩy sản phẩm
* Phí xác minh tài khoản cao cấp cho người dùng

### Đối tượng người dùng mục tiêu

Nền tảng này nhắm đến nhiều nhóm đối tượng người dùng khác nhau, mỗi nhóm với những nhu cầu và mong đợi riêng biệt:

1. **Người tiêu dùng cá nhân**:

* Người dùng tìm kiếm giải pháp tiết kiệm cho các dịch vụ giải trí trực tuyến
* Người mới làm quen với các nền tảng số và muốn thử nghiệm trước khi cam kết dài hạn
* Người dùng không có khả năng thanh toán quốc tế hoặc gặp khó khăn trong việc đăng ký trực tiếp
* Người tiêu dùng quan tâm đến việc tiết kiệm chi phí thông qua việc chia sẻ tài khoản

1. **Người bán tài khoản**:

* Cá nhân muốn kiếm thêm thu nhập từ việc bán lại các tài khoản không sử dụng
* Đại lý chính thức của các dịch vụ số tìm kiếm kênh phân phối mới
* Nhà cung cấp chuyên nghiệp với số lượng lớn tài khoản các loại

1. **Người quản trị và vận hành nền tảng**:

* Quản trị viên hệ thống giám sát hoạt động
* Nhân viên chăm sóc khách hàng và hỗ trợ kỹ thuật
* Đội ngũ marketing và phát triển kinh doanh

Hiểu rõ về các đối tượng người dùng này sẽ giúp định hướng thiết kế giao diện, xây dựng tính năng và phát triển các chính sách phù hợp với từng nhóm. Nền tảng sẽ chú trọng vào việc tạo ra trải nghiệm người dùng trực quan, an toàn và thân thiện, đồng thời đáp ứng các nhu cầu đặc thù của mỗi nhóm đối tượng.

Trong quá trình phát triển, dự án sẽ thực hiện các khảo sát và phân tích nhu cầu người dùng để không ngừng cải thiện và tối ưu hóa nền tảng, đảm bảo đáp ứng tốt nhất mong đợi của các đối tượng mục tiêu.

# PHÂN TÍCH YÊU CẦU

## Yêu cầu chức năng

### Quản lý người dùng (đăng ký, đăng nhập, quản lý tài khoản)

Hệ thống cần hỗ trợ các chức năng quản lý người dùng toàn diện, đảm bảo trải nghiệm mượt mà và an toàn cho người dùng từ khi bắt đầu tạo tài khoản đến khi sử dụng các tính năng nâng cao:

1. **Đăng ký tài khoản**:

* Đăng ký bằng email và mật khẩu
* Đăng ký thông qua tài khoản mạng xã hội (OAuth với Google, Facebook)
* Xác minh email thông qua liên kết kích hoạt
* Tùy chọn đăng ký làm người mua hoặc người bán (với quy trình xác minh khác nhau)
* Thu thập thông tin cá nhân cần thiết (tên, số điện thoại, địa chỉ)

1. **Đăng nhập và xác thực**:

* Đăng nhập với email/tên đăng nhập và mật khẩu
* Hỗ trợ đăng nhập thông qua OAuth (Google, Facebook)
* Xác thực hai yếu tố (2FA) cho tài khoản cao cấp
* Khôi phục mật khẩu qua email
* Duy trì phiên đăng nhập an toàn với JWT

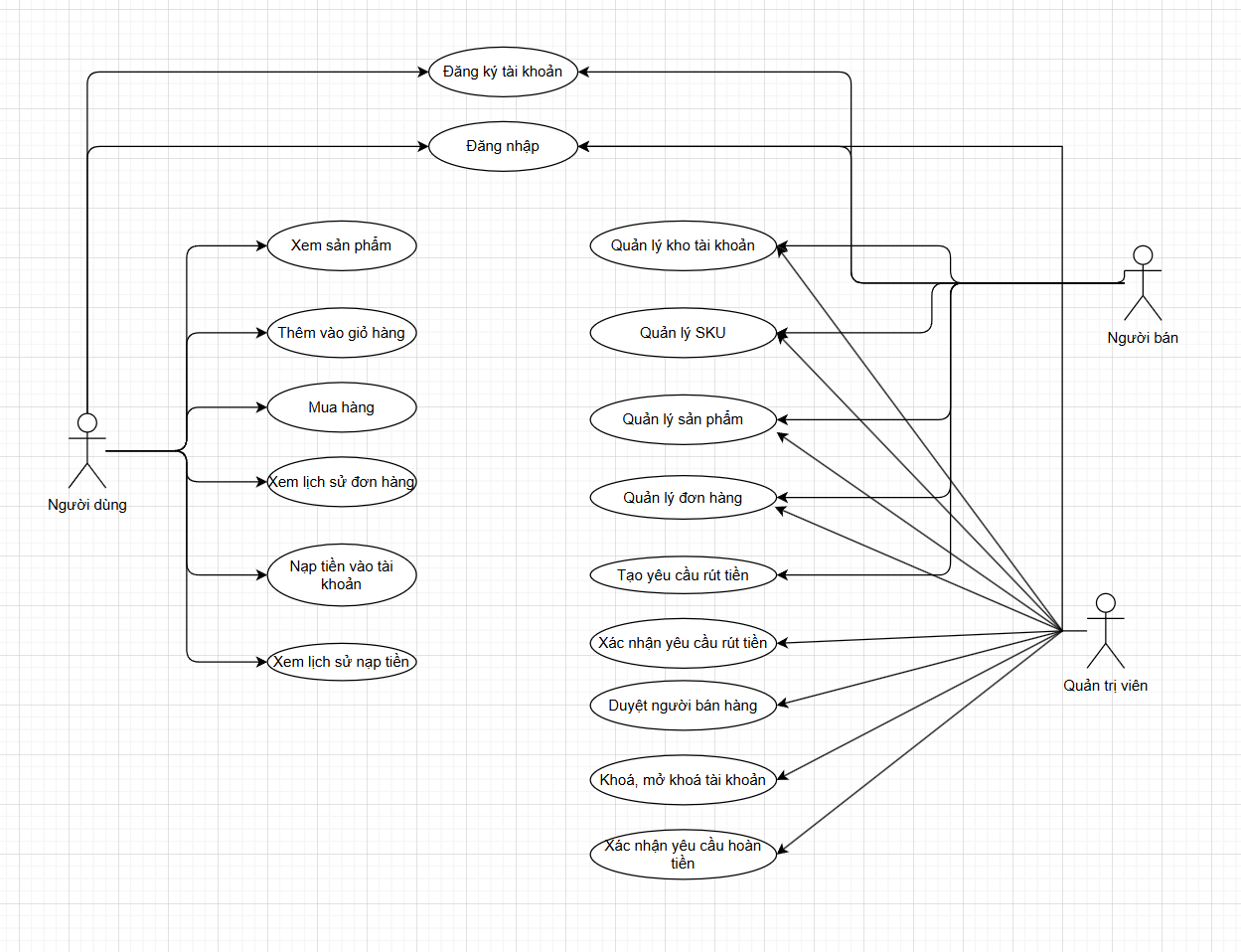
1. **Quản lý hồ sơ người dùng**:

* Xem và chỉnh sửa thông tin cá nhân
* Thay đổi mật khẩu
* Quản lý địa chỉ giao hàng/thanh toán
* Liên kết/hủy liên kết với tài khoản mạng xã hội
* Quản lý phương thức thanh toán
* Xem lịch sử giao dịch và đơn hàng

1. **Phân quyền và vai trò**:

* Người dùng thông thường (mua hàng)
* Người bán (bán tài khoản, quản lý kho)
* Quản trị viên (quản lý toàn bộ hệ thống)

**Sơ đồ usecase của hệ thống:**



Hình ảnh 1: Minh hoạ usecase của hệ thống

Các tác nhân trong sơ đồ:

* Người dùng
* Người bán
* Quản trị viênA diagram of a computer

  AI-generated content may be incorrect.

### Quản lý sản phẩm

Hệ thống cần có khả năng quản lý hiệu quả nhiều loại tài khoản số khác nhau, với các đặc điểm và yêu cầu quản lý riêng biệt:

1. **Thêm và quản lý sản phẩm**:
   * Giao diện thêm sản phẩm (tài khoản) mới
   * Nhập thông tin đăng nhập an toàn
   * Tải lên hình ảnh minh họa
   * Chọn danh mục và thuộc tính
   * Thiết lập giá và chính sách
2. **Phân loại sản phẩm**:
   * Tạo và quản lý danh mục (mạng xã hội, streaming, công cụ, v.v.)
   * Thiết lập thuộc tính cho từng loại tài khoản (thời hạn, số người dùng, v.v.)
   * Nhóm sản phẩm theo nhà cung cấp, loại dịch vụ
   * Gắn thẻ và từ khóa cho sản phẩm
3. **Quản lý kho và tình trạng**:
   * Theo dõi số lượng tài khoản có sẵn
   * Tự động cập nhật trạng thái sau khi bán
   * Cảnh báo khi số lượng tồn kho thấp
   * Kiểm tra tính hợp lệ của tài khoản định kỳ
4. **Quản lý giá và khuyến mãi**:
   * Thiết lập giá bán
   * Tạo mã giảm giá và chương trình khuyến mãi
   * Giá theo gói (1 tháng, 3 tháng, 1 năm)

A diagram of a diagram

AI-generated content may be incorrect.

Hình ảnh 2: Sơ đồ thực thể của sản phẩm

### Giỏ hàng và thanh toán

Hệ thống cần có quy trình giỏ hàng và thanh toán đơn giản nhưng đầy đủ tính năng, đảm bảo trải nghiệm mua sắm suôn sẻ:

1. **Quản lý giỏ hàng**:
   * Thêm/xóa sản phẩm khỏi giỏ hàng
   * Cập nhật số lượng sản phẩm
   * Tính toán tổng tiền tự động
   * Lưu giỏ hàng cho phiên đăng nhập sau
   * Áp dụng mã giảm giá
2. **Quy trình thanh toán**:
   * Form thanh toán với thông tin cần thiết
   * Lựa chọn phương thức thanh toán
   * Xác nhận đơn hàng trước khi thanh toán
   * Hiển thị tóm tắt đơn hàng
   * Xử lý giao dịch an toàn
3. **Phương thức thanh toán**:
   * Chuyển khoản ngân hàng (PayOS)
   * Bảo mật thông tin thanh toán
4. **Xử lý sau thanh toán**:
   * Cập nhật trạng thái đơn hàng
   * Tạo hóa đơn điện tử
   * Cập nhật kho sản phẩm
5. **Xử lý lỗi và hoàn tiền**:
   * Xử lý thanh toán không thành công
   * Quy trình hoàn tiền
   * Xử lý tranh chấp và khiếu nại
   * Chính sách đảm bảo và bảo hành

**Sequence Diagram cho quy trình nạp tiền bằng PayOS:**

A screenshot of a computer

AI-generated content may be incorrect.

### Tìm kiếm và lọc sản phẩm

Hệ thống cần có khả năng tìm kiếm và lọc hiệu quả để người dùng dễ dàng tìm được sản phẩm phù hợp:

1. **Chức năng tìm kiếm**:

* Tìm kiếm theo từ khóa
* Tìm kiếm theo danh mục
* Tìm kiếm nâng cao với nhiều tiêu chí

1. **Bộ lọc sản phẩm**:

* Lọc theo danh mục
* Lọc theo giá

1. **Sắp xếp kết quả**:

* Sắp xếp theo giá (tăng/giảm)
* Sắp xếp theo độ phổ biến
* Sắp xếp theo thời gian đăng
* Sản phẩm mới nhất

1. **Hiển thị kết quả**:

* Hiển thị dạng lưới hoặc danh sách
* Phân trang kết quả
* Hiển thị thông tin tóm tắt
* Xem nhanh thông tin chi tiết
* Thêm vào giỏ hàng

### Quản lý đơn hàng

Hệ thống cần có khả năng quản lý đơn hàng hiệu quả từ khi tạo đến khi hoàn thành:

1. **Tạo và xử lý đơn hàng**:
   * Tạo đơn hàng mới từ giỏ hàng
   * Tạo mã đơn hàng duy nhất
   * Ghi nhận thông tin người mua và sản phẩm
   * Tự động tính tổng tiền và thuế/phí
2. **Theo dõi trạng thái đơn hàng**:
   * Các trạng thái: Chờ thanh toán, Đã thanh toán, Đang xử lý, Đã giao, Hoàn thành, Hủy
   * Hiển thị timeline trạng thái đơn hàng
   * Thông báo khi trạng thái thay đổi
   * Cập nhật trạng thái theo thời gian thực
3. **Giao tài khoản**:
   * Gửi thông tin đăng nhập qua email an toàn
   * Hiển thị thông tin tài khoản trong trang đơn hàng
   * Hướng dẫn sử dụng tài khoản
   * Hỗ trợ kỹ thuật sau bán hàng
4. **Quản lý hóa đơn**:
   * Tạo hóa đơn điện tử
   * Lưu trữ lịch sử đơn hàng
5. **Xử lý vấn đề và hoàn tiền**:
   * Báo cáo vấn đề với đơn hàng
   * Yêu cầu hoàn tiền/đổi tài khoản
   * Xử lý tranh chấp
   * Chính sách bảo hành và đảm bảo
   * Đánh giá mức độ hài lòng

State Diagram để thể hiện các trạng thái khác nhau của đơn hàng và cách chuyển đổi giữa các trạng thái

A diagram of a data flow

AI-generated content may be incorrect.

## Yêu Cầu Phi Chức Năng

### Bảo mật thông tin tài khoản và dữ liệu người dùng

1. **Bảo mật dữ liệu**:

* Mã hóa thông tin đăng nhập và mật khẩu (cả lưu trữ và truyền tải)
* Kiểm soát truy cập dữ liệu theo vai trò

1. **Xác thực và ủy quyền**:

* JWT (JSON Web Tokens) cho xác thực API
* Chính sách mật khẩu mạnh
* Quản lý phiên đăng nhập an toàn

1. **Phòng chống tấn công**:

* Bảo vệ chống SQL Injection
* Bảo vệ chống Cross-Site Scripting (XSS)

1. **Kiểm soát và giám sát**:

* Ghi nhật ký (logging) toàn diện các hoạt động

1. **Quản lý sự cố**:

* Thông báo vi phạm cho người dùng bị ảnh hưởng
* Quy trình đánh giá và cải thiện bảo mật

### Trải nghiệm người dùng và giao diện

Giao diện người dùng cần được thiết kế trực quan, thân thiện và hiệu quả để tạo trải nghiệm tích cực cho người dùng:

1. **Thiết kế giao diện**:

* Giao diện hiện đại, sạch sẽ, dễ sử dụng
* Thiết kế responsive cho mọi thiết bị
* Bảng màu và typography nhất quán
* Tương thích đa trình duyệt

1. **Trải nghiệm người dùng**:

* Điều hướng trực quan và dễ dàng
* Số lượng click tối thiểu để hoàn thành tác vụ
* Thông báo phản hồi rõ ràng cho người dùng
* Hỗ trợ các mẫu UI/UX phổ biến
* Trải nghiệm mượt mà, không bị gián đoạn

1. **Khả năng tiếp cận**:

* Độ tương phản màu sắc phù hợp
* Hỗ trợ điều hướng bằng bàn phím
* Văn bản thay thế cho hình ảnh

1. **Tương tác và phản hồi**:

* Animation và chuyển động tinh tế
* Hiệu ứng loading thích hợp
* Thông báo lỗi và thành công rõ ràng
* Form validation trực quan
* A diagram of a computer

  AI-generated content may be incorrect.

# THIẾT KẾ HỆ THỐNG

## Kiến trúc hệ thống

### Sơ đồ kiến trúc tổng thể MERN Stack

Kiến trúc MERN Stack trong hệ thống thương mại điện tử bán tài khoản số và key bản quyền được tổ chức theo mô hình phân lớp như sau:

A screenshot of a computer

AI-generated content may be incorrect.

Hình ảnh 3: Minh hoạ kiến trúc của một dự án sử dụng MERN stack

Kiến trúc MERN Stack của hệ thống được thiết kế để phân tách rõ ràng giữa các thành phần, giúp dễ dàng phát triển, bảo trì và mở rộng. Mỗi thành phần có vai trò riêng biệt:

1. **Frontend (React + Redux)**:
   * UI Components: Thành phần giao diện người dùng, hiển thị dữ liệu và tương tác với người dùng
   * Redux Store: Quản lý trạng thái toàn cục của ứng dụng
   * API Services: Các service gọi API đến backend
2. **Backend (Node.js + Express)**:
   * API Routes: Định nghĩa các endpoint của API
   * Middlewares: Xử lý xác thực, phân quyền, và validation trước khi request đến controller
   * Controllers: Xử lý logic điều khiển, nhận request và trả về response
   * Services: Chứa logic nghiệp vụ phức tạp
3. **Database (MongoDB)**:
   * Lưu trữ dữ liệu dưới dạng các collection như Users, Products, Orders...
4. **External Services**:
   * Payment Gateways: Cổng thanh toán như Stripe, PayPal
   * Email Service: Dịch vụ gửi email xác nhận, thông báo
   * Cloud Storage: Lưu trữ hình ảnh và tài liệu

### Mô hình client-server

A screenshot of a computer

AI-generated content may be incorrect.

Hình ảnh 4: Minh hoạ mô hình Client-Server

Mô hình client-server trong hệ thống được thiết kế để tối ưu hóa việc phân chia trách nhiệm và truyền tải dữ liệu hiệu quả:

1. **Tách biệt Frontend và Backend**:
   * Frontend (React) hoạt động độc lập trên trình duyệt, tập trung vào trải nghiệm người dùng
   * Backend (Express) chịu trách nhiệm xử lý logic nghiệp vụ, tương tác với cơ sở dữ liệu và các dịch vụ bên ngoài
2. **Giao tiếp qua REST API**:
   * Frontend và Backend giao tiếp thông qua RESTful API
   * Dữ liệu được trao đổi dưới dạng JSON, dễ dàng xử lý ở cả hai phía
3. **Stateless Backend**:
   * Backend không lưu trữ trạng thái của phiên làm việc, giúp dễ dàng mở rộng quy mô
   * Xác thực người dùng thông qua JWT (JSON Web Tokens)

### Luồng dữ liệu và tương tác

Luồng dữ liệu điển hình trong hệ thống bán tài khoản số diễn ra như sau:

1. **Đăng ký/Đăng nhập**:
   * Người dùng nhập thông tin → Frontend validate → Gửi request đến Backend
   * Backend xác thực → Tạo JWT → Trả về token
   * Frontend lưu token và cập nhật trạng thái đăng nhập

A diagram of a computer

AI-generated content may be incorrect.

Hình ảnh 5: Minh hoạ quy trình tạo tài khoản

1. **Duyệt và tìm kiếm sản phẩm**:
   * Người dùng tìm kiếm/lọc → Frontend gửi request với tham số tìm kiếm
   * Backend query database → Trả về kết quả → Frontend hiển thị

A diagram of a computer

AI-generated content may be incorrect.

Hình ảnh 6: Minh hoạ quy trình duyệt và tìm kiếm sản phẩm

1. **Mua tài khoản/key**:
   * Người dùng thêm vào giỏ hàng → Frontend cập nhật giỏ hàng (lưu local/redux)
   * Người dùng thanh toán → Frontend gửi thông tin giỏ hàng đến Backend
   * Backend xác thực → Gọi cổng thanh toán → Xử lý thanh toán thành công
   * Backend cập nhật trạng thái đơn hàng → Gửi thông tin tài khoản/key đến người dùng
   * Frontend hiển thị thông tin tài khoản/key đã mua

A diagram of a data flow

AI-generated content may be incorrect.

Hình ảnh 7: Minh hoạ quy trình mua hàng

## Thiết kế cơ sở dữ liệu

### Mô hình dữ liệu MongoDB (các collection)

Mô hình dữ liệu MongoDB được thiết kế để phù hợp với đặc thù của hệ thống bán tài khoản số và key bản quyền:

1. **User Collection**:
   * Lưu trữ thông tin người dùng bao gồm email, mật khẩu (đã được hash), thông tin cá nhân (username)
   * Phân quyền thông qua trường role (admin, user)
2. **Product Collection**:
   * Chứa thông tin cơ bản (tên, mô tả, hình ảnh)
   * Kết nối với danh mục và danh mục phụ
   * Liên kết với người bán (người dùng có vai trò là người bán)
   * Có nhiều SKU (đơn vị lưu kho) cho các biến thể khác nhau
3. **SKU Collection**:

Xử lý các biến thể cụ thể của sản phẩm:

* + Liên kết với sản phẩm gốc
  + Theo dõi giá, hàng tồn kho và số liệu bán hàng
  + Duy trì ràng buộc duy nhất cho mỗi sản phẩm (không trùng tên SKU)

1. **Inventory Collection**

Lưu trữ thông tin xác thực kỹ thuật số thực tế:

* + Tham chiếu SKU
  + Theo dõi trạng thái thông tin xác thực (có sẵn, đã bán, không hợp lệ)
  + Liên kết đến đơn hàng khi đã bán
  + Liên kết với người bán

1. **Category Collection**:
   * Phân loại sản phẩm (Netflix, Spotify, Windows, Office...)
2. **Order Collection**:

Ghi lại các giao dịch mua hàng:

* + Lưu trữ đơn hàng và trạng thái
  + items là mảng chứa các sản phẩm trong đơn hàng
  + Theo dõi phương thức thanh toán và trạng thái

1. **Transaction Collection**:

Lưu trữ hồ sơ tài chính:

* + Các loại: mua hàng, hoàn tiền, rút ​​tiền, gửi tiền
  + Theo dõi trạng thái (đang chờ xử lý, đã hoàn tất, không thành công)
  + Tham chiếu đơn hàng và người dùng
  + Hỗ trợ xử lý thanh toán qua PayOS

A diagram of a computer

AI-generated content may be incorrect.

Hình ảnh 8: Mô hình các collection trong MongoDB

### Mối quan hệ giữa các collection

1. Collection User

Mối quan hệ:

* **User → Product**: Quan hệ 1-n (một User có thể tạo nhiều Product nếu là seller)
* **User → Order**: Quan hệ 1-n (một User mua nhiều Order)
* **User → Transaction**: Quan hệ 1-n (một User có nhiều Transaction)
* **User → Inventory**: Quan hệ 1-n (một User quản lý nhiều Inventory nếu là seller)

1. Collection Product

* **Product → User**: Quan hệ n-1 (nhiều Product thuộc về một User/seller)
* **Product → Category**: Quan hệ n-1 (nhiều Product thuộc một Category)
* **Product → SKU**: Quan hệ 1-n (một Product có nhiều SKU)

1. Collection SKU (Stock Keeping Unit)

**Mối quan hệ**:

* **SKU → Product**: Quan hệ n-1 (nhiều SKU thuộc về một Product)
* **SKU → Inventory**: Quan hệ 1-n (một SKU có thể có nhiều item Inventory)
* **SKU → Order**: Quan hệ n-n (nhiều SKU có thể xuất hiện trong nhiều Order)

1. Collection Inventory

**Mối quan hệ**:

* **Inventory → SKU**: Quan hệ n-1 (nhiều Inventory item thuộc về một SKU)
* **Inventory → User**: Quan hệ n-1 (nhiều Inventory thuộc một seller/User)
* **Inventory → Order**: Quan hệ n-1 (nhiều Inventory có thể được bán trong một Order)

1. Collection Order

**Mối quan hệ**:

* **Order → User**: Quan hệ n-1 (nhiều Order thuộc về một User/buyer)
* **Order → SKU**: Quan hệ n-n (nhiều Order chứa nhiều SKU khác nhau)
* **Order → Inventory**: Quan hệ 1-n (một Order có thể bán nhiều Inventory items)
* **Order → Transaction**: Quan hệ 1-n (một Order có thể tạo ra nhiều Transaction)

1. Collection Transaction

**Mối quan hệ**:

* **Transaction → User**: Quan hệ n-1 (nhiều Transaction thuộc về một User)
* **Transaction → Order**: Quan hệ n-1 (nhiều Transaction có thể liên quan đến một Order)

1. Collection Category

**Mối quan hệ**:

* **Category → Product**: Quan hệ 1-n (một Category có nhiều Product)

**Phân tích Mối quan hệ Chi tiết**

Mối quan hệ Phức tạp:

1. Quy trình Bán hàng:

User (seller) → tạo Product → tạo nhiều SKU → upload nhiều Inventory items

User (buyer) → tạo Order → chọn nhiều SKU → mua Inventory items → tạo Transaction

1. Quy trình Cập nhật Hàng tồn kho:

Khi Order được tạo → giảm stock của các SKU liên quan

Cập nhật trạng thái Inventory từ 'available' sang 'sold'

Liên kết Inventory với Order thông qua trường order

1. Quy trình Quản lý Tài chính:

User thực hiện giao dịch → tạo Transaction

Transaction liên kết với User và có thể liên kết với Order

Cập nhật balance của User dựa trên Transaction