<https://www.youtube.com/watch?v=YDdja3Xozkg>

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**Đề tài: quản lý thanh toán hóa đơn trực tuyến**

**Nhóm 15 - Sinh viên thực hiện**

1. Phạm Thanh Duy – 16043751
2. Lê Anh Tú – 16021321
3. Nguyễn Xuân Trường Giang – 16040991

# MỤC LỤC

[MỤC LỤC 1](#_Toc14693502)

[DANH MỤC CÁC HÌNH VẼ 1](#_Toc14693503)

[DANH MỤC CÁC BẢNG BIỂU 3](#_Toc14693504)

[CHƯƠNG 1 : GIỚI THIỆU 4](#_Toc14693505)

[1.1 Tổng quan 4](#_Toc14693506)

[1.2 Mục tiêu đề tài 5](#_Toc14693507)

[1.3 Phạm vi đề tài 6](#_Toc14693508)

[1.4 Mô tả yêu cầu chức năng 7](#_Toc14693509)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 8](#_Toc14693510)

[2.1 Cloud Computing với AWS 8](#_Toc14693511)

[2.2 Node.js 13](#_Toc14693512)

[2.3 DynamoDB 15](#_Toc14693513)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 18](#_Toc14693514)

[3.1 Phân tích yêu cầu bằng UML 18](#_Toc14693515)

[3.1.1 Usecase tổng quát 18](#_Toc14693516)

[3.1.2 Danh sách tác nhân và mô tả 19](#_Toc14693517)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 19](#_Toc14693518)

[3.1.4 Tình huống hoạt động 20](#_Toc14693519)

[3.2 Class diagram 37](#_Toc14693520)

[3.3 Deployment diagram 45](#_Toc14693521)

[CHƯƠNG 4 : HIỆN THỰC 46](#_Toc14693522)

[4.1 Cấu hình phần cứng, phần mềm 46](#_Toc14693523)

[4.2 Giao diện của hệ thống 46](#_Toc14693524)

[CHƯƠNG 5 : KẾT LUẬN 46](#_Toc14693525)

[5.1 Kết quả đạt được 46](#_Toc14693526)

[5.2 Hạn chế của đồ án 46](#_Toc14693527)

[5.3 Hướng phát triển 46](#_Toc14693528)

[TÀI LIỆU THAM KHẢO 47](#_Toc14693529)

# DANH MỤC CÁC HÌNH VẼ

[Hình 2‑1 Node.js development 13](#_Toc14693413)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 19](#_Toc14682192)

[Bảng 3‑2 Danh sách các tình huống trong hệ thống 19](#_Toc14682193)

# : GIỚI THIỆU

## Tổng quan

Trong thời đại hiện nay, thời đại của sự phát triển công nghệ thông tin đã tạo ra những thay đổi quan trọng trong đời sống con người cũng như đời sống sản xuất, với sự hỗ trợ của công nghệ thông tin một số ngành nghề thủ công đang được dần thay thế bằng những ứng dụng, những robot thông minh. Lợi ích từ việc tiết kiệm thời gian và chi phí là rất lớn.

Hiện nay, đời sống của mỗi người ngày càng bận rộn và tất bật, công việc ở công ty, việc gia đình, tình cảm… Thời gian rảnh càng bị bó hẹp lại và nhu cầu về những việc có thể xử lý nhanh gọn thì càng tăng. Đa số mọi người đều có kiến thức cơ bản về việc sử dụng thiết bị điện thoại hay máy tính, cho nên hiện nay có rất nhiều dịch vụ trên những thiết bị này có thể đáp ứng những nhu cầu cơ bản như gọi thức ăn nhanh, đặt hàng trực tuyến, gọi phương tiện di chuyển…

Một trong số đó có một dịch vụ có thể giải quyết vấn đề về các hóa đơn trong gia đình như hóa đơn điện, nước, internet… Nhận thức được vấn đề này nên nhóm đã phát triển ứng dụng web Thanh toán hóa đơn trực tuyến để hỗ trợ khách hàng trong việc thanh toán các hóa đơn của mình.

Thanh toán hóa đơn trực tuyến là một dạng thực hiện thanh toán các hóa đơn mà bạn đang sử dụng trong gia đình của bạn thông qua website trực tuyến. Việc sử dụng thanh toán hóa đơn trực tuyến sẽ góp phần thúc đẩy quá trình thanh toán của bạn nhanh hơn, giúp bạn tiết kiệm một khoản thời gian và tiện lợi hơn những cách thanh toán truyền thống. Website Thanh toán hóa đơn trực tuyến của chúng tôi sẽ giúp bạn làm điều đó.

Ứng dụng được phát triển để kết nối doanh nghiệp cung cấp dịch vụ và khách hàng của họ để khách hàng có thể thực hiện thanh toán các loại hóa đơn như hóa đơn điện, hóa đơn nước, hóa đơn Internet ..v.v…. Khách hàng có thể đăng ký thành viên, đăng nhập, nạp tiền vào tài khoản của mình thông qua cổng thanh toán Paypal để thuận tiện cho việc thanh lý các hóa đơn của bạn. Khách hàng chỉ cần cung cấp cho website của chúng tôi mã khách hàng của doanh nghiệp mà bạn đã đăng ký với doanh nghiệp cung cấp dịch vụ. Ngoài chức năng chính thanh toán, khách hàng có thể kiểm soát được lịch sử giao dịch, bao gồm lịch sử thanh toán, lịch sử nạp tiền, chi tiết hóa đơn đã thanh toán. Khách hàng cũng có thể đặt lịch nhắc cho mã khách hàng mà bạn muốn chúng tôi gửi mail nhắc bạn hằng tháng về việc gần tới hạn thanh toán hóa đơn đó. Với giao diện thân thiện, thao tác dễ dàng, website của chúng tôi sẽ đem đến cho bạn một trải nghiệm tuyệt vời và tiện lợi.

Để xây dựng được một website Thanh toán hóa đơn trực tuyến có chất lượng dịch vụ tốt, đề tài đã được phát triển dựa trên nền tảng NodeJS với cơ sở dữ liệu NoSQL nhanh và linh hoạt là DynamoDB cùng với một số dịch vụ của Amazon Web Services.

## Mục tiêu đề tài

* Sự dụng các dịch vụ trên nền tảng điện toán đám mây như là EC2, S3, DynamoDB kết hợp với Platform NodeJS trên công cụ Visual Studio Code để thực hiện và triển khai các chức năng:

\* Doanh nghiệp:

- Xem danh sách các hóa đơn đã thanh toán qua website.

- Cung cấp thông tin hóa đơn cho hội viên thanh toán.

\* Hội viên:

- Đăng ký thành viên và đăng nhập vào hệ thống.

- Nạp tiền vào tài khoản thành viên.

- Thanh toán các hóa đơn mà Hội viên muốn thanh toán (số dư trong tài khoản phải lớn hơn hoặc bằng số tiền thanh toán).

- Xem lại lịch sử các hóa đơn đã thanh toán bao gồm chi tiết hóa đơn.

- Đặt lịch nhắc về việc tới hạn thanh toán hóa đơn hằng tháng.

\* Người quản lý:

- Đăng ký doanh nghiệp vào hệ thống.

- Quản lý được thông tin của hội viên.

- Quản lý được thông tin của doanh nghiệp.

- Báo cáo thống kê các lịch sử giao dịch.

## Phạm vi đề tài

* Các chức năng thực hiện cho đề tài:
* **Đối với Doanh Nghiệp:**
* Cung cấp thông tin hóa đơn.
* Xem danh sách hóa đơn.
* Hủy hợp đồng.
* Đăng ký doanh nghiệp.
* **Đối với Hội viên:**

- Thanh toán hóa đơn

- Xem lịch sử giao dịch.

- Xem hóa đơn thanh toán.

- Đặt lịch nhắc.

- Nạp Tiền.

* **Đối với Người Quản Lý:**

- Lập hóa đơn.

- Gửi hóa đơn.

- Quản lý Hội viên.

- Quản lý doanh nghiệp.

- Báo cáo thống kê.

* Thực hiện với các công nghệ:
* Sử dụng HTML kết hợp với CSS, JavaScript và Bootstrap để tạo giao diện cho website.
* Sử dụng hệ quản trị cơ sở dữ liệu No SQL DynamoDB kết hợp với Platform NodeJS trên công cụ Visual Studio Code để thực hiện và triển khai các chức năng cho website.
* Triển khai áp dụng trên dịch vụ của AWS như EC2, S3 vào website để tăng cường khả năng lưu trữ dữ liệu và bảo mật, kích cỡ linh hoạt trên điện toán đám mây.
* Tải các Dependencies hỗ trợ cho Platform NodeJS để thực hiện ứng dụng như express (dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web), body-parser (lấy dữ liệu trong form với phương thức post), template ejs (tạo giao diện web), fs, paypal-rest-sdk, passport (để đăng nhập và xác thực tài khoản), express-session (lưu lại session tài khoản đã đăng nhập) các dependencies sẽ hổ trợ triển khai các chức năng nhanh chóng và tiện lợi.

Các công việc chưa thực hiện được:

* Thông tin khách hàng từ các doanh nghiệp trong thực tế.
* Không thế nạp tiền bằng thẻ ngân hàng thật.

## Mô tả yêu cầu chức năng

Để thực hiện việc thanh toán các hóa đơn, đầu tiên khách hàng cần đăng ký tài khoản để trở thành hội viên, rồi sau đó đăng nhập vào hệ thống website. Tài khoản của hội viên cần có đủ số dư để thanh toán số tiền trên hóa đơn, để có số dư hội viên có thể sử dụng chức năng nạp tiền vào tài khoản thông qua cổng Paypal. Sau khi có đủ số dư trong tài khoản thì có thể thực hiện chức năng thanh toán hóa đơn với các bước như sau. Trước tiên cần chọn loại hóa đơn muốn thanh toán sau đó chọn doanh nghiệp mà họ đã đăng ký sử dụng dịch vụ. Tiếp theo thì cần nhập vào mã khách hàng của doanh để hiện thị hóa đơn, hệ thống sẽ gửi mã mà hội viên vừa nhập đến doanh nghiệp. Doanh nghiệp sẽ cung cấp lại thông tin hóa đơn theo mã khách hàng đến hệ thống và sau đó hiển thị ra hóa đơn cho hội viên kiểm tra lại thông tin có đúng hay không và tiến hành chọn nút thanh toán. Sau khi thanh toán thành công, số dư trong tài khoản sẽ bị trừ bằng số tiền có trong hóa đơn, số tiền thanh toán đó sẽ chuyển đến tài khoản paypal của người quản lý trang website và số dư của tài khoản doanh nghiệp sẽ được cộng vào tương ứng số tiền hội viên đã trả trừ ra tiền phí giao dịch (phí mà doanh nghiệp đã đăng ký dịch vụ).

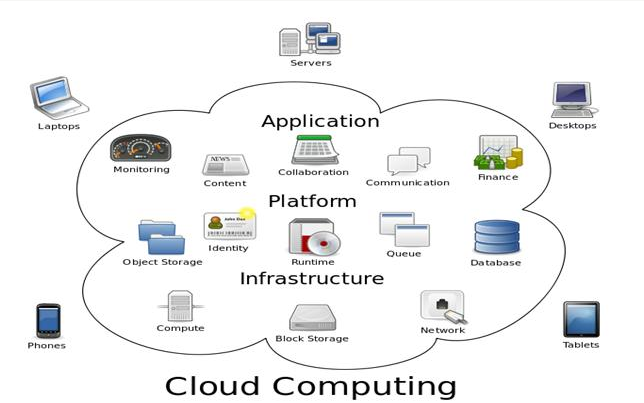
Hội viên có thể xem lại những lịch sử giao dịch của tài khoản bằng cách chọn chức năng lịch sử để xem thông tin các giao dịch mà tài khoản đã thực hiện.

Hội viên cũng có thể đặt lịch nhắc tới hạn thanh toán hóa đơn bằng cách chọn chức năng thanh toán hóa đơn, nhập mã khách hàng và email để nhận lịch nhắc hằng tháng. Sau đó chọn nút đặt lịch nhắc, website sẽ nhắc hội viên thanh toán khi đến thời điểm mà khách hàng đã đặt lịch nhắc trước đó.

# : CƠ SỞ LÝ THUYẾT

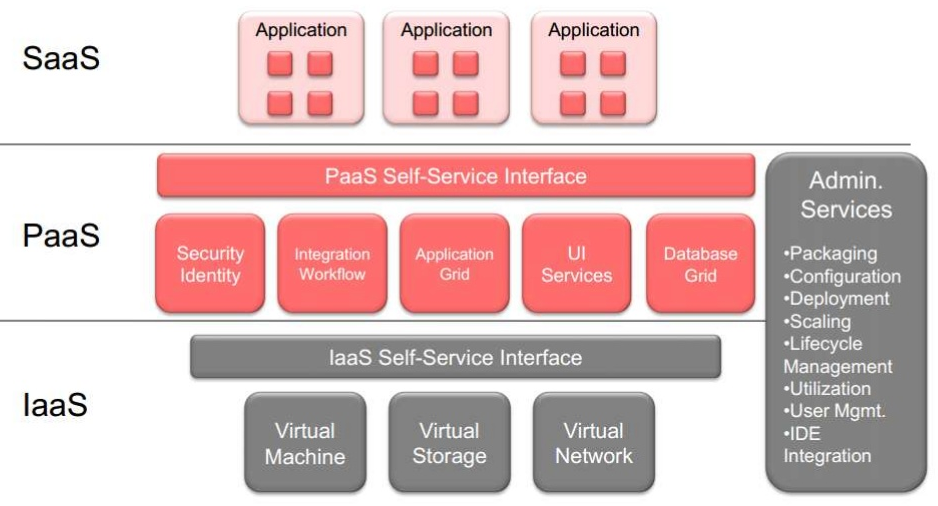
## Cloud Computing với AWS

* **Cloud computing**: theo định nghĩa của IBM là việc cung cấp các tài nguyên máy tính cho người dùng tùy theo mục đích sử dụng thông qua kết nối Internet. Nguồn tài nguyên đó có thể là bất kì thứ gì liên quan đến điện toán và máy tính, ví dụ như phần mềm, phần cứng, hạ tầng mạng cho đến các máy chủ và mạng lưới máy chủ cỡ lớn.



*Hình 2.1: mô tả cloud computing*

* Dịch vụ điện toán đám mây chia thành 3 loại:
* **Infrastructure as a Service (IaaS)**: là hạ tầng được cung cấp như một dịch vụ lưu trữ dữ liệu.
* **Platform as a Service (PaaS)** là nền tảng được cung cấp như một dịch vụ với một mức độ cao hơn một chút so với **IaaS**. **PaaS** không cho phép bạn sở hữu máy chủ, họ chỉ đơn giản là cung cấp cho bạn một nền tảng (Platform) để bạn triển khai ứng dụng của mình, trong trường hợp này bạn đang sử dụng chung 1 máy chủ (Server) với nhiều người khác. Và cơ sở dữ liệu của bạn cũng được đặt trong môi trường lưu trữ với hàng trăm cơ sở dữ liệu của người khác.
* **Software as a Service (SaaS)** là một mức độ cao nhất trong các loại hình dịch vụ đám mây. Nó là một sản phẩm hoàn thiện được vận hành và quản lý bởi một nhà cung cấp. Và nói đơn giản, trong hầu hết trường hợp thì dịch vụ SaaS sẽ cung cấp các phần mềm hoặc ứng dụng chạy trên Internet và có thể được dùng ngay bởi người dùng cuối (end-user).

****

*Hình 2.2: dịch vụ diện toán đám mây*

* **Mô hình triển khai** (Deployment Models):
* Private Cloud: các dịch vụ này được quản lý và cung cấp trong tổ chức. Có ít hạn chế hơn về băng thông mạng, ít phơi nhiễm bảo mật hơn và các yêu cầu pháp lý khác so với Đám mây công cộng.
* Public Cloud: các dịch vụ được chuyển đến khách hàng qua Internet từ nhà cung cấp dịch vụ bên thứ ba. (Ví dụ: Amazon)
* Community Cloud: Tài nguyên được chia sẻ bởi một số tổ chức có mối quan tâm chung.
* Hybrid Cloud: Tài nguyên được quản lý một phần bởi các nhà cung cấp dịch vụ và một phần bởi tổ chức.
* **Amazon Web Service (AWS)** là nền tảng dịch vụ đám mây an toàn, mang đến khả năng tính toán, lưu trữ cơ sở dữ liệu, phân phối nội dung và các chức năng khác nhằm giúp các doanh nghiệp mở rộng và phát triển. Khám phá cách hàng triệu khách hàng đang tận dụng sản phẩm và giải pháp đám mây của AWS để xây dựng các ứng dụng phức tạp với tính linh hoạt, khả năng mở rộng và độ tin cậy cao.
* Nền tảng AWS:
* **AWS Storage Services**
  + **Amazon Simple Storage Service (Amazon S3):** Amazon S3 hoặc Amazon Simple Storage Service là một "dịch vụ lưu trữ đơn giản", cung cấp khả năng lưu trữ dữ liệu như: html, video, hình ảnh, …
  + **Amazon Elastic Block Store (EBS):** là một dịch vụ lưu trữ dạng khối dễ sử dụng và hiệu năng cao, được thiết kế để sử dụng với Amazon Elastic Compute Cloud (EC2) cho các khối lượng công việc đòi hỏi tốc độ giao dịch và thông lượng cao ở mọi quy mô.
  + **Amazon Storage Gateway:** là dịch vụ lưu trữ đám mây lai cung cấp cho bạn khả năng truy cập gần như không giới hạn tại chỗ vào bộ lưu trữ đám mây.
* **AWS Compute Services**
  + **Amazon Elastic Compute Cloud (Amazon EC2**): là dịch vụ web cung cấp năng lực điện toán bảo mật và có kích cỡ linh hoạt trên đám mây. Dịch vụ này được thiết kế để giúp các nhà phát triển dễ sử dụng điện toán đám mây ở quy mô web hơn.
  + **Amazon Elastic MapReduce (EMR):** là nền tảng dữ liệu lớn dành cho đám mây dẫn đầu ngành, cho phép các nhóm nhanh chóng xử lý lượng lớn dữ liệu và tiết kiệm chi phí ở quy mô lớn.
* **AWS Database Services**
  + **Amazon DynamoDB:** là một dịch vụ cơ sở dữ liệu độc quyền quản lý hoàn toàn sử dụng NoSQL có hỗ trợ giá trị khóa và cấu trúc dữ liệu tài liệu
  + **Amazon SimpleDB**: Amazon SimpleDB là một cơ sở dữ liệu phân tán. Nó được sử dụng như một dịch vụ web kết hợp với Amazon Elastic Compute Cloud và Amazon S3 và là một phần của Amazon Web Services.
  + **Amazon Relational Database Service (RDS):** à dịch vụ đám mây do Amazon Web Services phát triển với mục tiêu cung cấp giải pháp cài đặt, vận hành và mở rộng dành cho relational database
* **AWS Networking Service**
  + **Amazon Virtual Private Cloud (VPC):** là một dịch vụ điện toán đám mây thương mại cung cấp cho người dùng một đám mây riêng ảo
  + **Amazon Route 53:** là một dịch vụ Hệ thống tên miền có khả năng mở rộng và khả dụng cao. Tên này là một tham chiếu đến cổng TCP hoặc UDP 53, trong đó các yêu cầu máy chủ DNS được giải quyết
* **AWS Content Delivery**
  + **Amazon CloudFront:** Là các mạng phân phối nội dung cung cấp một mạng lưới các máy chủ proxy được phân phối toàn cầu để lưu trữ nội dung, chẳng hạn như video trên web hoặc các phương tiện cồng kềnh khác, cục bộ hơn cho người tiêu dùng
* **AWS Application Services**
  + **Amazon Simple Workflow Service (SWF)**: giúp các nhà phát triển dựng, chạy và thay đổi quy mô các tác vụ nền có các bước thực hiện song song hoặc theo trình tự. Bạn có thể xem Amazon SWF và trình theo dõi trạng thái và trình điều phối tác được quản lý đầy đủ trên Đám mây.
  + **Amazon Simple Notification Service (SNS):** Dịch vụ thông báo đơn giản của Amazon là dịch vụ thông báo được cung cấp như một phần của Dịch vụ web Amazon kể từ năm 2010. Nó cung cấp cơ sở hạ tầng chi phí thấp để gửi tin nhắn hàng loạt, chủ yếu cho người dùng di động.
  + **Amazon Simple Queue Service (SQS**): Dịch vụ xếp hàng đơn giản của Amazon là dịch vụ xếp hàng tin nhắn phân tán được Amazon.com giới thiệu vào cuối năm 2004. Nó hỗ trợ gửi tin nhắn theo chương trình thông qua các ứng dụng dịch vụ web như một cách để giao tiếp qua Internet
* **AWS Deployment and Management Services**
  + **AWS Elastic Beanstalk (beta):** à dịch vụ phối hợp được cung cấp bởi Amazon Web Services để triển khai các ứng dụng phối hợp các dịch vụ AWS khác nhau, bao gồm EC2, S3, Dịch vụ thông báo đơn giản, CloudWatch, tự động cân bằng và Cân bằng tải đàn hồi
  + **AWS CloudFormation:** là một dịch vụ hỗ trợ việc thiết lập các tài nguyên của Amazon Web Service, nhờ đó bạn chỉ cần bỏ ra ít thời gian để quản lý các tài nguyên đó và có thể tập trung vào phát triển ứng dụng chạy trên AWS.
  + **Amazon CloudWatch:** là dịch vụ giám sát và quản lý cung cấp dữ liệu và thông tin chi tiết khả thi cho tài nguyên cơ sở hạ tầng và ứng dụng AWS, ứng dụng lai và ứng dụng tại chỗ. Với CloudWatch, bạn có thể tổng hợp và đánh giá tất cả dữ liệu hiệu suất và hoạt động dưới hình thức nhật ký và số liệu trong cùng một nền tảng.
  + AWS Identity and Access Management (IAM)
* **AWS Analytics Services**

## Node.js



Hình 2‑1 Node.js development

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

* Các đặc tính của NodeJS:
  + **Không đồng bộ**: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
  + Chạy nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
  + Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng.
  + Không đệm: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
  + Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.
* Cách hoạt động: NodeJS sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.
* Express là một web application framework for nodejs, nó cung cấp cho chúng những rất nhiều tính năng mạnh mẽ trên nền tảng web. Express rất dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web. Express hỗ trợ các phương thức HTTP và middleware tạo ra 1 API rất mạnh mẽ và sử dụng dễ dàng hơn. Khi mới tiếp cận với Express mình thực sự bị cuốn hút bởi các API của nó, từ cách sử dụng route, template, đều khá dễ tùy chọn và làm việc. Các tính năng của Express framework phải kể đến như:
  + Cho phép thiết lập các lớp trung gian để trả về các HTTP request.
  + Định nghĩa routing có thể được sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
  + Cho phép trả về các trang HTML dựa vào các tham số truyền vào đến template.

## DynamoDB

* DynamoDB là một dịch vụ NoSQL được quản lý đầy đủ, hoạt động trên cặp khóa giá trị và các tài liệu cấu trúc dữ liệu khác do Amazon cung cấp.
* DynamoDB bao gồm ba đơn vị cơ bản được gọi là table, attribute (thuộc tính) và items. Một bảng chưa một tập hợp các items, một thuộc tính là thành phần đơn giản nhất lưu trữ dữ liệu mà không có bất kỳ phân chia nào nữa và items giữ một tập các thuộc tính.
* Cấu trúc của DynamoDB: DynamoDB bao gồm ba đơn vị cơ bản được gọi là table, attribute (thuộc tính) và items.
* Table là một tập hợp dữ liệu chứa các thông tin. Mỗi môt dòng thông tin gọi là 1 item
* Item là tập hợp các thuộc tính được định danh đại diện cho một đối tượng nào đó.
* Attribute là trường thuộc tính chứa dữ liệu của 1 item. Mỗi thuộc tính được xác định bởi 1 tên.
* Primary Key là một thuộc tính duy nhất cần thiết trong khi tạo bảng, nó không thể rỗng ở bất kỳ điểm nào. Hai mục không thể có khóa chính tương tự.
* Dynamo hỗ trợ hai loại khóa chính:
* Simple Primary Key: là giống như Partition Key (khóa phân vùng), về cơ bản đây là thuộc tính duy nhất. DynamoDB sử dụng giá trị Partition Key để phân biệt các mục trong bảng.
* Composite Primary Key là khóa phân vùng và khóa sắp xếp. Loại khóa này thường được tạo từ hai mục thành phần chính là khóa phân vùng, thành phần phụ là khóa sắp xếp.
* Secondary index:

Để giúp cho việc truy cập dữ liệu, Amazon DynamoDB tạo và quản lý index cho primary key. Việc này giúp cho hệ thống truy cập dữ liệu khá nhanh khi chỉ định primary key. Tuy nhiên, nhiều hệ thống có lẽ sẽ hưởng lợi từ việc có một hoặc hai key thay thế (alternate). Để tăng tốc độ truy cập bạn có thể tạo ra secondary index.

Một secondary index là một cấu trúc dữ liệu chứa tập con của các attributes trong một bảng, cùng với một key thay thế để hỗ trợ cho Query operation. Với một secondary index, các query sẽ không bị giới hạn khi chỉ sử dụng được primary key; bạn có thể truy vấn dữ liệu bằng việc sử dụng key thay thế từ secondary index. Một bảng có thể có nhiều secondary index để hỗ trợ cho nhiều query pattern.

* DynamoDB hỗ trợ hai loại index:
* Global secondary index: một loại index có hash và range key khác với hash và range key từ bảng gốc.
* Local secondary index: một loại index mà có hash key trùng với hash key từ bảng gốc và range key là một attribute khác.
* Làm việc với bảng trong DynamoDB:

Khi tạo bảng chúng ta cần:

* Chỉ định Primary Key: Khi tạo một bảng, chúng ta cần cần chỉ rõ primary key . Primary key để đảm bảo rằng không có hai items nào trong bảng có cùng primary key. DynamoDB hỗ trợ hai loại primary key sau:
* **Hash Primary Key:** Chỉ cần một attribute để tạo ra primary key. DynamoDB xây dựng hash index không được sắp xếp cho primary key attribute.
* **Hash và Range Primary Key:** Cần 2 attributes để tạo ra primary key. Attribute đầu tiền là hash attribute và attribute còn lại là range attribute. DynamoDB xây dựng hash index không được sắp xếp cho hash attribute và range index được sắp xếp cho range attribute. Mỗi item được định danh bởi cặp primary key này. Điều này cho phép 2 items có cùng hash attribute nhưng khác range attribute cùng tồn tại trong một bảng.
* Trong quá trình tạo bảng, bạn chỉ rõ mức têu thụ (throughput) yêu cầu theo đơn vị là capacity unit. Bạn có thể thay đổi (tăng hoặc giảm) chỉ số này qua UpdateTable request:
* **Read capacity unit** - Con số phản ánh số lượng đọc strongly consitent của các item không quá 4KB
* **Write capacity unit** - Số lượng những lần ghi có dung lượng 1KB mỗi giây. Có nghĩa rằng nếu bạn yêu cầu 10 write capacity units là bạn đang yêu cầu mức tiêu thụ 10 writes với dung lượng 1 KB mỗi giây của bảng đó
* Ưu điểm của DynamoDB:
  + **Khả năng mở rộng linh hoạt:** do không bị ràng buộc chặt về các mối quan hệ, cấu trúc lưu trữ nên khả năng mở rộng của DynamoDB rất linh động.
  + **DynamoDB phù hợp với công nghệ đám mây**: Những yêu cầu về lưu trữ lớn của công nghệ đám mây với DynamoDB là một sự lựa chọn đúng đắn.
  + **Tính co giãn**: Khi các dữ liệu và giao dịch gia tăng DynamoDB đáp ứng tốt các nhu cầu co giãn khi cần thiết.
* Nhược điểm của DynamoDB:
  + **Sự mới mẻ:** DynamoDB cũng như SQL là những công nghệ mới do đó không có nhiều người tinh thông cũng như hiểu rõ công nghệ này.
  + **Khó khăn khi cài đặt và sử dụng:** Việc sử dụng được DynamoDB yêu cầu khá nhiều kĩ năng và cách sử dụng của nó cũng có một chút phức tạp

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát

******

*Hình 3-1: Sơ đồ use case tổng quát*

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Doanh nghiệp | Cung cấp thông tin hóa đơn của khách hàng, xem danh sách hóa đơn đã thanh toán. |
| Hội viên | Thực hiện đăng ký tài khoản, thanh toán hóa đơn, xem lịch sử giao dịch, đặt lịch nhắc, nạp tiền. |
| Người quản lý | Thực hiện lập hóa đơn, thông báo email, quản lý hội viên, quản lý doanh nghiệp, đăng ký doanh nghiệp, thống kê phí dịch vụ. |

Bảng 3-1 Danh sách tác nhân và mô tả

### Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Cung cấp thông tin hóa đơn |
| UC02 | Xem danh sách hóa đơn đã thanh toán |
| UC03 | Đăng ký doanh nghiệp |
| UC04 | Đăng ký tài khoản |
| UC05 | Thanh toán hóa đơn |
| UC06 | Xem lịch sử giao dịch |
| UC07 | Đặt lịch nhắc |
| UC08 | Nạp tiền |
| UC09 | Lập hóa đơn |
| UC10 | Thông báo email |
| UC11 | Quản lý Hội viên |
| UC12 | Quản lý doanh nghiệp |
| UC13 | Thống kê phí doanh vụ |

Bảng 3-2 Danh sách các tình huống trong hệ thống

### Tình huống hoạt động

#### 3.1.4.1 UC01\_Cung cấp thông tin hóa đơn

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** *Cung cấp thông tin hóa đơn* | |
| Mục đích: | Cho phép doanh nghiệp cung cấp hóa đơn của khách hàng của họ vào cơ sở dữ liệu |
| Mô tả: | Doanh nghiệp sẽ nhập các thông tin của hóa đơn vào mẫu |
| Tác nhân: | Doanh nghiệp |
| Điều kiện trước: | Khi được nhận mã khách hàng và loại hóa đơn thanh toán từ website. |
| Điều kiện sau: | Hệ thống gửi thông tin hóa đơn của khách hàng đến trang use case UC10\_Lập hóa đơn. |
| Luồng sự kiện chính | 1. Hệ thống tìm thông tin hóa đơn theo mã khách hàng. 2. Hệ thống kiểm tra mã khách hàng. 3. Hệ thống gửi thông tin hóa đơn khách hàng đến use case UC10\_Lập hóa đơn. |
| Luồng sự kiện phụ | 3.1 Hệ thống thông báo mã khách hàng không đúng. |

Bảng 3-3 Đặc tả use case UC01

* Activity diagram



*Hình 3-2 activity UC01*

#### 3.1.4.2 UC02\_Xem danh sách hóa đơn đã thanh toán

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** *Xem danh sách hóa đơn đã thanh toán* | |
| Mục đích: | Cho phép doanh nghiệp xem lại dach sách hóa đơn đã thanh toán của khách hàng của họ |
| Mô tả: | Hệ thống hiển thị thông tin các hóa đơn của khách hàng của doanh nghiệp |
| Tác nhân: | Doanh nghiệp |
| Điều kiện trước: | Đã đăng nhập vào tài khoản của doanh nghiệp. |
| Điều kiện sau: | Hệ hiển thị thông tin các hóa đơn mà doanh nghiệp đã cung cấp thông tin |
| Luồng sự kiện chính | 1. Doanh nghiệp chọn xem danh sách hóa đơn và chọn loại hóa đơn cần xem 2. Hệ thống hiển thị danh sách các hóa đơn tương ứng 3. Doanh nghiệp có thể lọc thông tin thời gian giao dịch 4. Doanh nghiệp nhấn nút lọc 5. Hệ thống hiển thị danh sách tương ứng với thông tin được yêu cầu |
| Luồng sự kiện phụ | Nếu không có hóa đơn nào hệ thống sẽ hiển thị danh sách rỗng |

*Bảng 3-4 Đặc tả use case UC02*

* Activity diagram



*Hình 3-3 Activity UC02*

#### 3.1.4.3 UC03\_Đăng kí doanh nghiệp

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** *Đăng kí doanh nghiệp* | |
| Mục đích: | Để thêm doanh nghiệp vào hệ thống |
| Mô tả: | Người quản lý nhập thông tin doanh nghiệp để thêm vào hệ thống và sử dụng các dịch vụ của hệ thống. |
| Tác nhân: | Người quản lý |
| Điều kiện trước: | Đăng nhập thành công với tài khoản người quản lý. |
| Điều kiện sau: | Hệ thống lưu thông tin doanh nghiệp vào cơ sở dữ liệu |
| Luồng sự kiện chính | 1. Người dùng chọn đăng kí doanh nghiệp 2. Hệ thống hiển thị giao diện đăng kí doanh nghiệp 3. Người dùng nhập thông tin 4. Hệ thống kiểm tra thông tin nhập vào. 5. Người dùng nhấn đồng ý 6. Hệ thống lưu thông tin doanh nghiệp vào cơ sở dữ liệu. |
| Luồng sự kiện ngoại lệ | 4.1 Nếu thông tin nhập vào không hợp lệ và yêu cầu người dùng nhập lại. |

*Bảng 3-6 Đặc tả use case UC03*

* Activity diagram



*Hình 3-5 activity UC03*

#### 3.1.4.4 UC04\_Đăng kí tài khoản

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** *Đăng ký tài khoản* | |
| Mục đích: | Đăng ký tài khoản website |
| Mô tả: | Use case mô tả việc khách hàng đăng ký tài khoản của website để trở thành hội viên. |
| Tác nhân: | Hội viên |
| Điều kiện trước: | Truy cập vào website thanh toán hóa đơn |
| Điều kiện sau: | Đăng ký hội viên thành công và lưu thông tin hội viên xuống cơ sở dữ liệu. |
| Luồng sự kiện chính | 1. Hội viên chọn chức năng đăng ký tài khoản. 2. Hệ thống hiển thị giao diện đăng ký. 3. Hội viên nhập thông tin hội viên để tiến hành đăng ký. 4. Hội viên chọn nút đăng ký. 5. Hệ thống thông báo đăng ký thành công. |
| Luồng sự kiện ngoại lệ | 4.1 Hệ thống kiểm tra xem có sai thông tin hoặc có bị trùng thông tin hay không. Nếu có thì quay lại bước 3. |

*Bảng 3-7 Đặc tả use case UC04*

* Activity diagram



*Hình 3-6 activity UC04*

* Activity diagram



*Hình 3-6 sequence UC04*

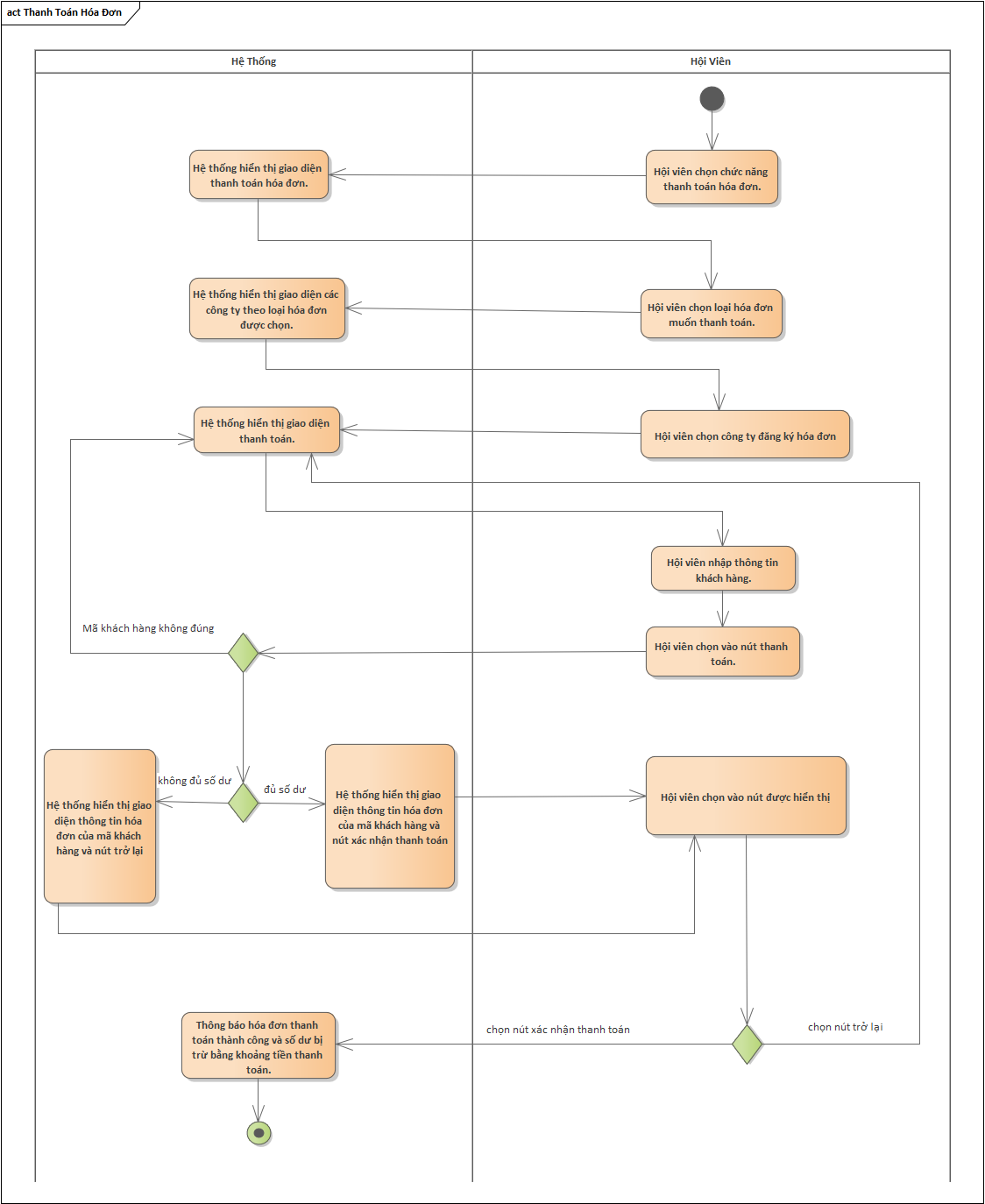
#### 3.1.4.5 UC05\_Thanh toán hóa đơn

Đặc tả use case:

|  |  |
| --- | --- |
| **Use case: Thanh Toán Hóa Đơn** | |
| **Mục đích:** | Thanh toán hóa đơn cho khách hàng. |
| **Mô tả:** | Use case này mô tả việc hội viên thực hiện thanh toán các hóa đơn thông qua chức năng thanh toán của website. |
| **Tác Nhân:** | Hội viên |
| **Điều kiện trước:** | Hội viên đăng nhập vào hệ thống và có số dư tài khoản lớn hơn số tiền trong hóa đơn. |
| **Điều kiện sau** | Thanh toán thành công hóa đơn, lưu lịch sử giao dịch xuống cơ sở dữ liệu. |
| **Luồng sự kiện** | 1. Hội viên chọn chức năng thanh toán hóa đơn.  2. Hệ thống hiển thị giao diện thanh toán hóa đơn.  3. Hội viên chọn loại hóa đơn muốn thanh toán.  4. Hệ thống hiển thị giao diện các công ty theo loại hóa đơn được chọn.  5. Hội viên chọn công ty đăng ký hóa đơn.  6. Hệ thống hiển thị giao diện thanh toán.  7. Hội viên nhập thông tin khách hàng.  8. Hội viên chọn vào nút thanh toán.  9.Hệ thống hiển thị giao diện thông tin hóa đơn của mã khách hàng  10.Khách hàng kiểm tra và chọn vào nút xác nhận thanh toán.  11. Thông báo hóa đơn thanh toán thành công và số dư bị trừ bằng khoảng tiền thanh toán. |
| **Luồng sự kiện ngoại lệ** | 8.1 Hệ thống kiểm tra thông tin khách hàng có đúng hay không. Nếu không quay lại bước 6 và thông báo sai thông tin khách hàng.  8.2 Hệ thống kiểm tra số dư của tài khoản có đủ hay không. Nếu đủ thì hiển thị nút xác nhận thanh toán, nếu không thì hiển thị nút trở lại ở bước 9.  10.1 Khách hàng chọn vào nút trở lại, quay lại bước 6. |

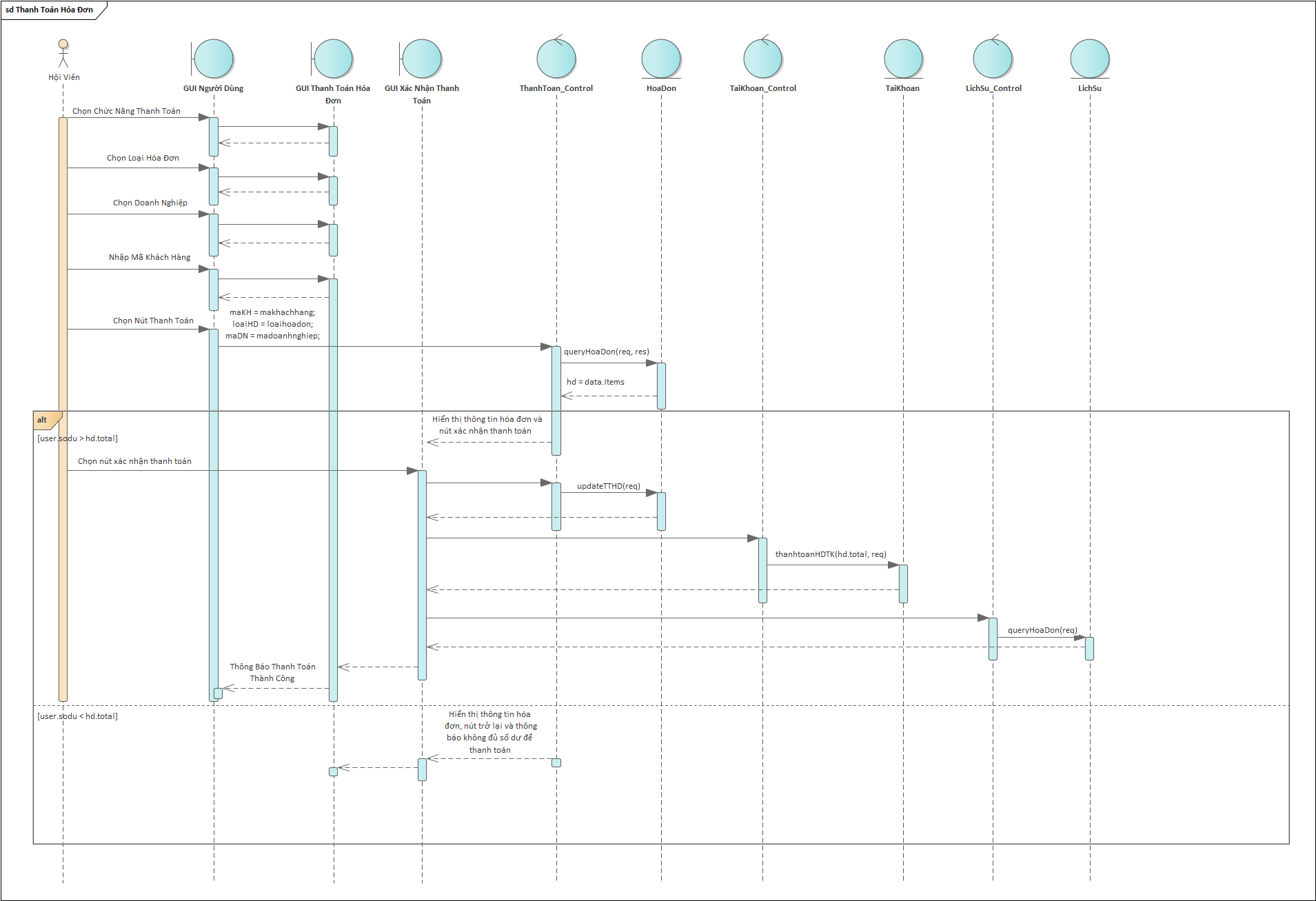
*Bảng 3-8 Đặc tả use case UC05*

* Activity Diagram



*Hình 3-7 activity UC05*

* Sequence Diagram



*Hình 3-7 sequence UC05*

#### 3.1.4.6 UC06\_Xem lịch sử giao dịch

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case: Xem lịch sử giao dịch** | |
| **Mục đích:** | Xem lại các hóa đơn hoặc lịch sử nạp tiền trước đó. |
| **Mô tả:** | Use case mô tả việc hội viên muốn xem lại các hóa đơn hoặc lịch sử nạp tiền. |
| **Tác Nhân:** | Hội viên |
| **Điều kiện trước:** | Hội viên đăng nhập vào hệ thống. |
| **Điều kiện sau** | Hệ thống hiển thị ra các hóa đơn mà hội viên đã thanh toán. |
| **Luồng sự kiện** | 1. Hội viên chọn chức năng lịch sử giao dịch.  2. Hệ thống hiển thị giao diện lịch sử hội viên đã thanh toán.  3. Hội viên có thể lọc hóa đơn hoặc lịch sử nạp tiền theo tháng, năm, loại hóa đơn.  4. Hội viên chọn nút lọc.  5. Hệ thống lọc các trường mà hội viên yêu cầu.  6. Hệ thống hiển thị ra các hóa đơn hoặc lịch sử nạp tiền đã thanh toán trước đó xếp theo ngày tháng thanh toán. |
| **Luồng sự kiện ngoại lệ** |  |

*Bảng 3-9 Đặc tả use case UC06*

* Activity diagram



*Hình 3-8 activity UC06*

* Sequence



*Hình 3-8 sequence UC06*

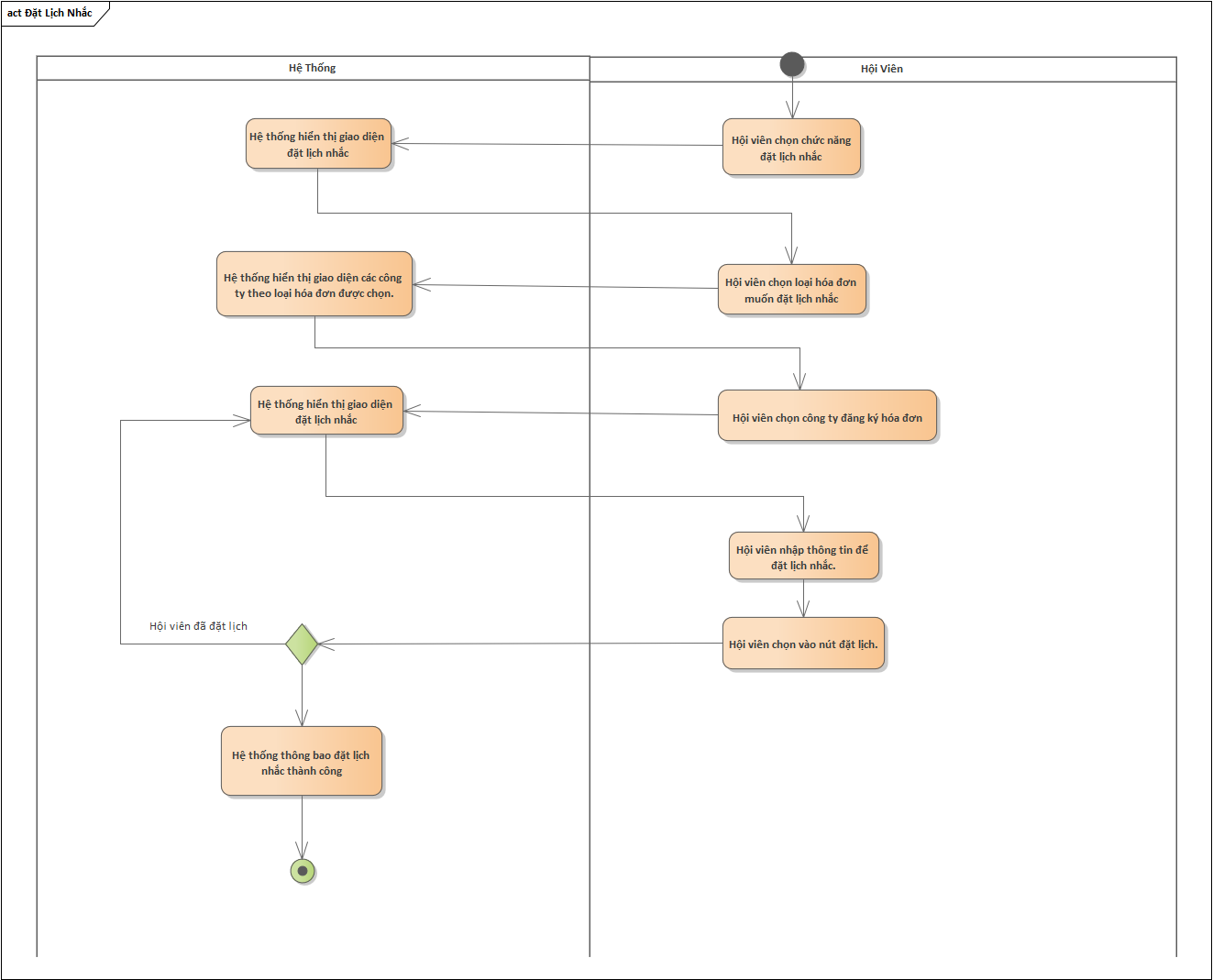
#### 3.1.4.7 UC07\_Đặt lịch nhắc

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case: *Đặt lịch nhắc*** | |
| **Mục đích:** | Đặt lịch nhắc cho hội viên biết gần tới hạn thanh toán hóa đơn hằng tháng. |
| **Mô tả:** | Use case mô tả việc hội viên muốn đặt lịch nhắc để thông báo thanh toán. |
| **Tác Nhân:** | Hội viên. |
| **Điều kiện trước:** | Hội viên đăng nhập vào hệ thống. |
| **Điều kiện sau** | Hệ thống thông báo đặt lịch nhắc thành công và lưu xuống cơ sở dữ liệu. |
| **Luồng sự kiện** | 1. Hội viên chọn chức năng đặt lịch nhắc.  2. Hệ thống hiển thị giao diện đặt lịch nhắc.  3. Hội viên chọn loại hóa đơn muốn đặt lịch nhắc.  4. Hệ thống hiển thị giao diện các công ty theo loại hóa đơn được chọn.  5. Hội viên chọn công ty đăng ký hóa đơn.  6. Hệ thống hiển thị giao diện đặt lịch nhắc.  7. Hội viên nhập thông tin để đặt lịch nhắc.  8. Hội viên chọn nút đặt lịch nhắc.  9. Hệ thống kiểm tra hội viên có đặt lịch nhắc hay chưa  10. Hệ thống thông báo đặt lịch nhắc thành công. |
| **Luồng sự kiện ngoại lệ** | 10.1 Thông báo lịch nhắc này đã đặt rồi và quay lại bước 6. |

*Bảng 3-10 Đặc tả use case UC07*

* Activity diagram



*Hình 3-9 activity use case UC07*

#### 3.1.4.8 UC08\_Nạp tiền

* Đặc tả use case

|  |  |  |
| --- | --- | --- |
| **Use case:** ***Nạp tiền*** | | |
| **Mục đích:** | Nạp tiền vào tài khoản của hội viên đã đăng ký để dùng thanh toán các hóa đơn. |
| **Mô tả:** | Use case mô tả việc hội viên nạp tiền vào tài khoản để dùng thanh toán các hóa đơn. |
| **Tác Nhân:** | Hội viên. |
| **Điều kiện trước:** | Hội viên đăng nhập vào hệ thống. |
| **Điều kiện sau** | Hệ thống thông báo nạp tiền thành công và lưu số tiền nạp xuống cơ sở dữ liệu. |
| **Luồng sự kiện** | 1. Hội viên chọn chức năng nạp tiền.  2. Hệ thống hiển thị giao diện nạp tiền.  3. Hội viên chọn cổng thanh toán muốn nạp vào.  4. Hệ thống hiển thị các thông tin cần để nạp tiền.  5. Hội viên nhập thông tin và số tiền muốn nạp.  6. Hội viên chọn nút nạp tiền.  7. Hệ thống thông báo nạp tiền thành công và số tiền được cộng vào số dư của tài khoản. |
| **Luồng sự kiện ngoại lệ** | 6.1 Hệ thống kiểm tra thông tin có đúng hay không. Nếu không quay lại bước 4 và thông báo sai thông tin. |

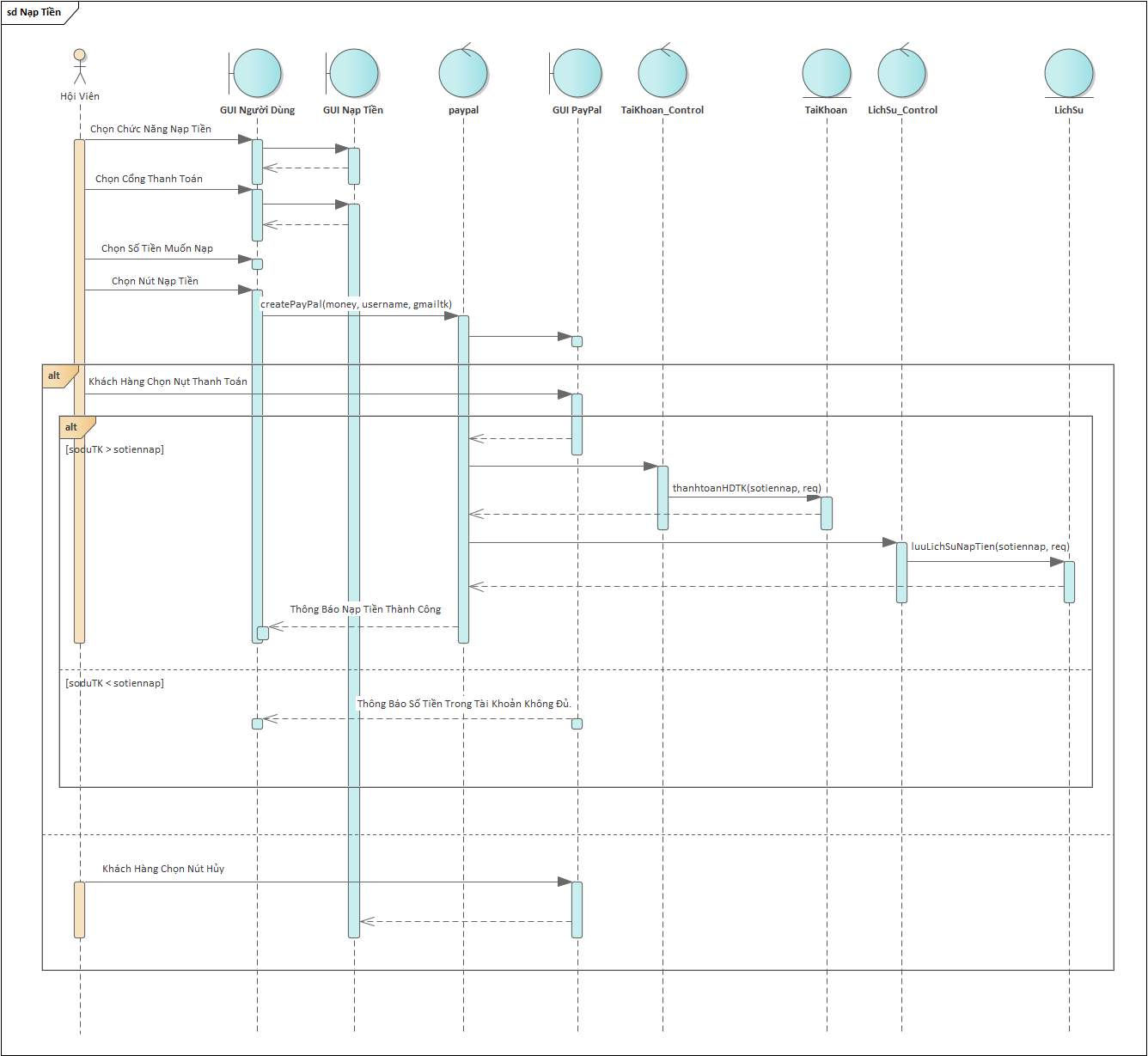
*Bảng 3-11 Đặc tả use case UC08*

* Activity diagram



*Hình 3-10 activity UC08*

* Sequence diagram



*Hình 3-10 sequence UC08*

#### 3.1.4.9 UC09\_Lập hóa đơn

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case: *Lập hóa đơn*** | |
| Mục đích: | Tạo ra hóa đơn điện tử của hội viên. |
| Mô tả: | Use case thực hiện việc tạo ra hóa đơn mà hội viên yêu cầu. |
| Tác nhân: | Hội viên |
| Điều kiện trước: | Hội viên thực hiện use case thanh toán hóa đơn. |
| Điều kiện sau: | Tạo hóa đơn thành công. |
| Luồng sự kiện chính | 1. Hệ thống yêu cầu thực hiện use case UC01\_Cung cấp thông tin hóa đơn. 2. Hệ thống tạo hóa đơn theo thông tin cung cấp của doanh nghiệp. 3. Tạo hóa đơn thành công. |
| Luồng sự kiện phụ | 3.1. Thông báo chưa tới kỳ hạn thanh toán. |

*Bảng 3-12 Đặc tả use case UC09*

* Activity diagram



*Hình 3-11 activity UC09*

#### 3.1.4.10 UC10\_Thông báo email

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case: *Thông báo email*** | |
| Mục đích: | Gửi email tới hội viên. |
| Mô tả: | Use case thực hiện việc gửi email tới hội viên đã thực hiện đặt lịch nhắc trước đó. |
| Tác nhân: | Hội viên |
| Điều kiện trước: | Đã thực hiện use case UC09\_Lập hóa đơn và use case UC07\_Đặt lịch nhắc |
| Điều kiện sau: | Gửi email thành công đến hội viên. |
| Luồng sự kiện chính | 1. Hệ thống kiểm tra hội viên có đặt lịch nhắc hay không? 2. Hệ thống lấy thông tin hội viên đã đặt lịch nhắc trước đó. 3. Hệ thống lấy thông tin hóa đơn từ use case UC09\_Lập hóa đơn. 4. Hệ thống gửi hóa đơn đến email của hội viên. |
| Luồng sự kiện phụ | Không có. |

*Bảng 3-13 Đặc tả use case UC10*

* Activity diagram



*Hình 3-12 activity UC10*

#### 3.1.4.11 UC11\_Quản lý hội viên

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** *Quản lý hội viên* | |
| Mục đích: | Hiển thị các chức năng quản lý hội viên |
| Mô tả: | Hiển thị chức năng tìm kiếm, cập nhật, xóa, sửa doanh nghiệp |
| Tác nhân: | Người quản lý website |
| Điều kiện trước: | Đăng nhập vào tài khoản hội viên |
| Điều kiện sau: | Hệ thống hiển thị các chức năng quản lý hội viên |
| Luồng sự kiện chính | 1. Người quản lý website chọn Quản lý hội viên. 2. Hệ thống hiển thị giao diện quản lý hội viên. |
| Luồng sự kiện phụ |  |

*Bảng 3-14 Đặc tả use case UC11*

* Activity diagram

*Hình 3-13 activity UC11*

#### 3.1.4.12 UC12\_Quản lý doanh nghiệp

* Đặc tả use case

|  |  |
| --- | --- |
| **Use case:** *Quản lý doanh nghiệp* | |
| Mục đích: | Hiển thị các chức năng quản lý doanh nghiệp |
| Mô tả: | Hiển thị chức năng tìm kiếm, cập nhật, xóa, sửa doanh nghiệp |
| Tác nhân: | Hội viên |
| Điều kiện trước: | Đăng nhập vào tài khoản Hội viên |
| Điều kiện sau: | Hệ thống hiển thị các chức năng Quản lí doanh nghiệp |
| Luồng sự kiện chính | 1. Người dùng chọn Quản lí doanh nghiệp 2. Hệ thống hiển thị giao diện quản lý doanh nghiệp |
| Luồng sự kiện phụ |  |

*Bảng 3-15 Đặc tả use case UC12*

* Activity diagram

*Hình 3-14 activity UC12*

#### 3.1.4.13 UC13\_Thống kê phí dịch vụ

* Đặc tả

|  |  |
| --- | --- |
| **Use case:** *Thống kê phí dịch vụ* | |
| Mục đích: | Thống kê phí dịch vụ doanh nghiệp đã đăng ký sử dụng và hội viên thanh toán hóa đơn. |
| Mô tả: | Use case thực hiện việc thông kê các khoản phí dịch vụ mà doanh nghiệp đã đăng ký và thu phí khi Hội viên thực hiện thanh toán hóa đơn. |
| Tác nhân: | Hội viên |
| Điều kiện trước: | Đăng nhập vào tài khoản Hội viên |
| Điều kiện sau: | Hệ thống hiển thị danh sách khoảng phí đã thu được. |
| Luồng sự kiện chính | 1. Người quản lý chọn Thống kê phí dịch vụ. 2. Hệ thống hiển thị danh sách các khoảng phí đã thu. |
| Luồng sự kiện phụ |  |

*Bảng 3-16 Đặc tả use case UC13*

* Activity



*Hình 3-15 activity UC13*

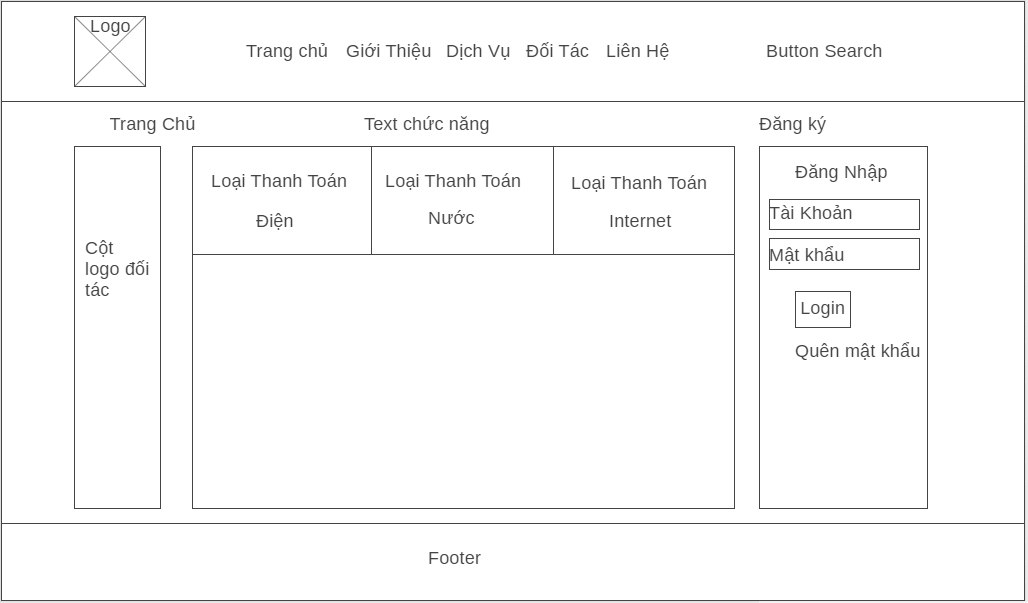
## Class diagram

Mô hình lớp (Mô hình lớp đầy đủ + Mô hình lớp bao gồm các lớp Entity)

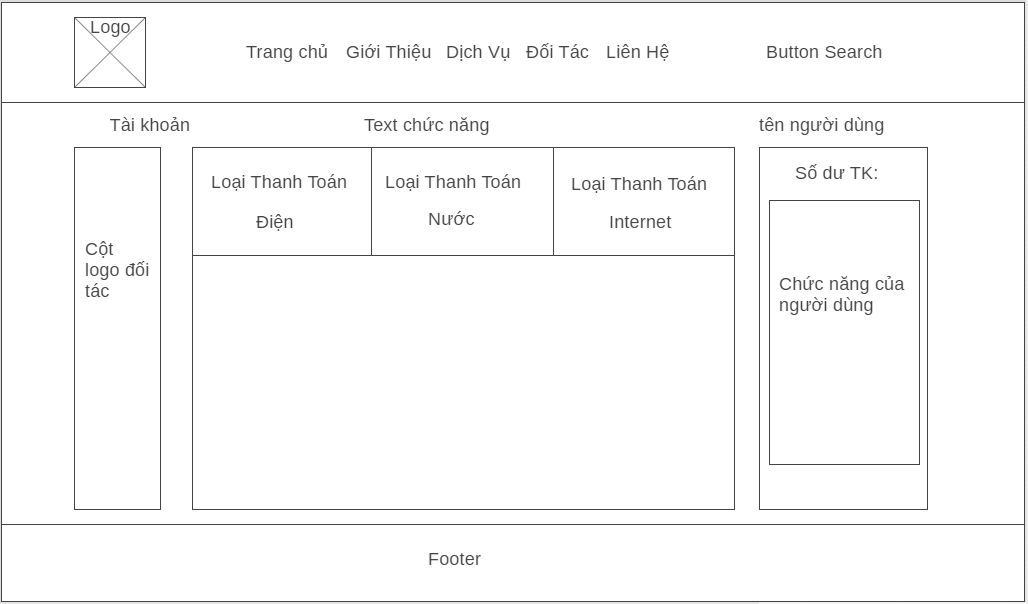


*Hình 3-16 sơ đồ classdiagram*

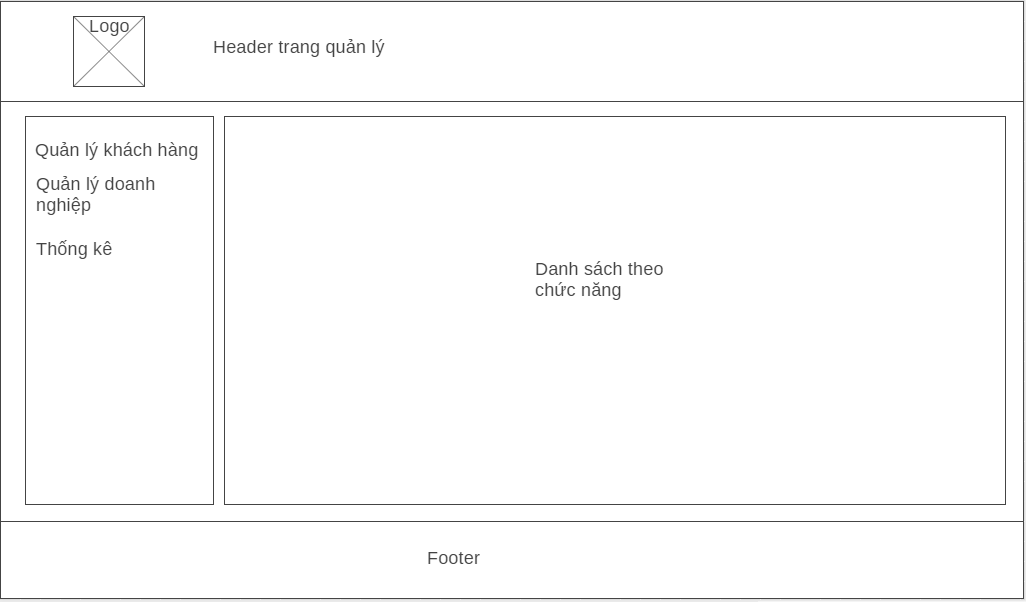
## Mô hình Mockup

Trang chủ:

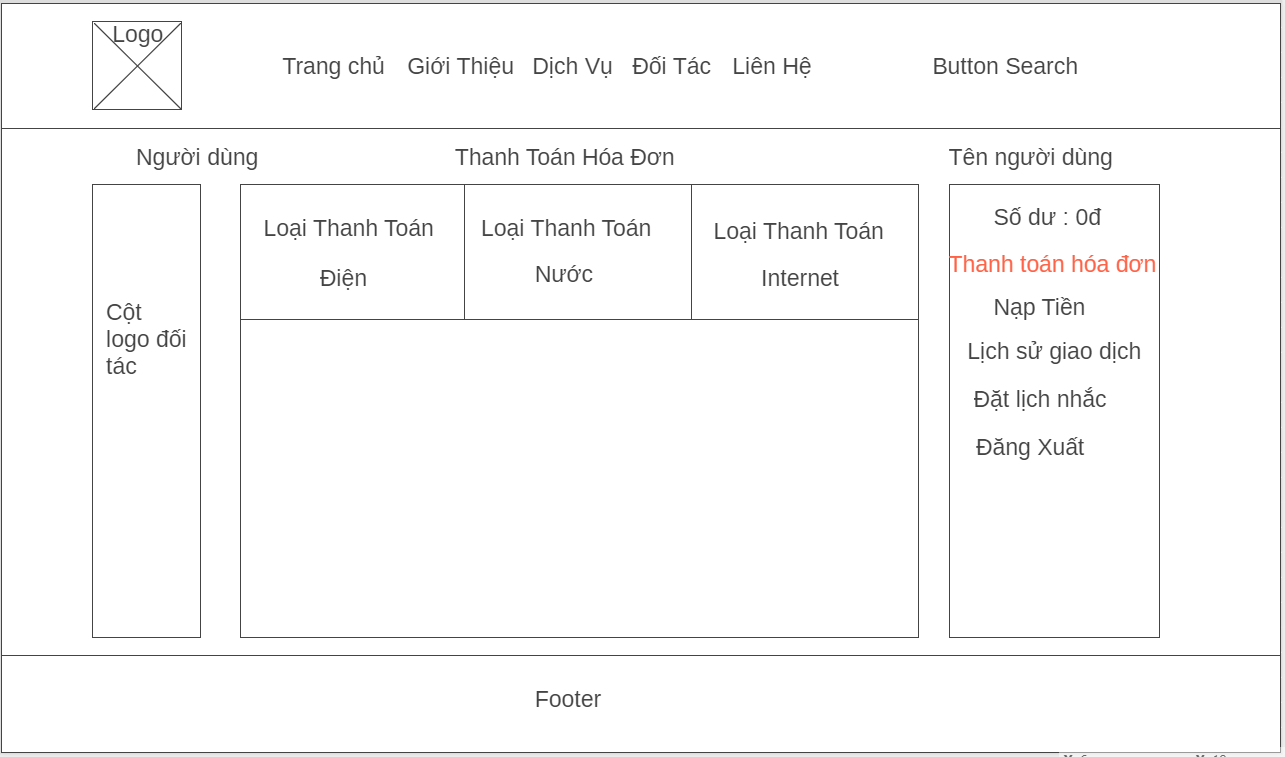
*Hình 3.3 – 1 Trang chủ*

Trang người dùng:

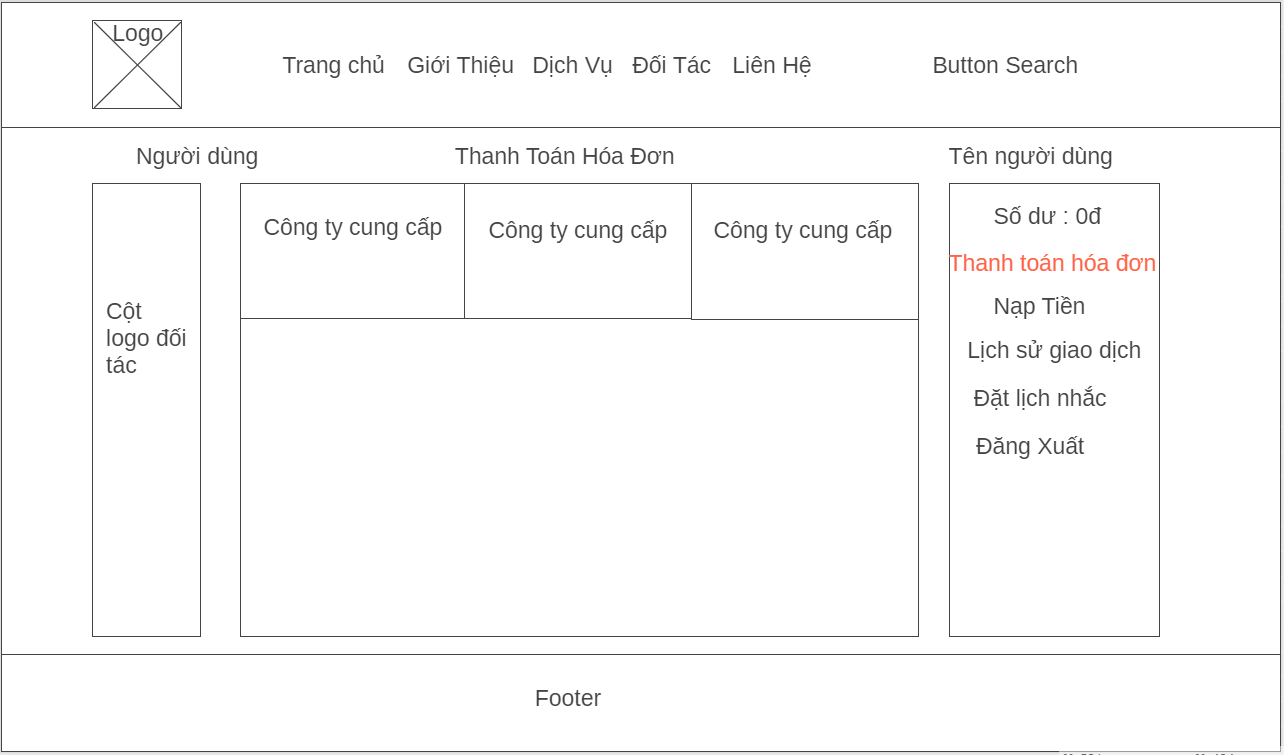
*Hình 3.3 – 2 Trang người dùng*

Trang Người Quản Lý:

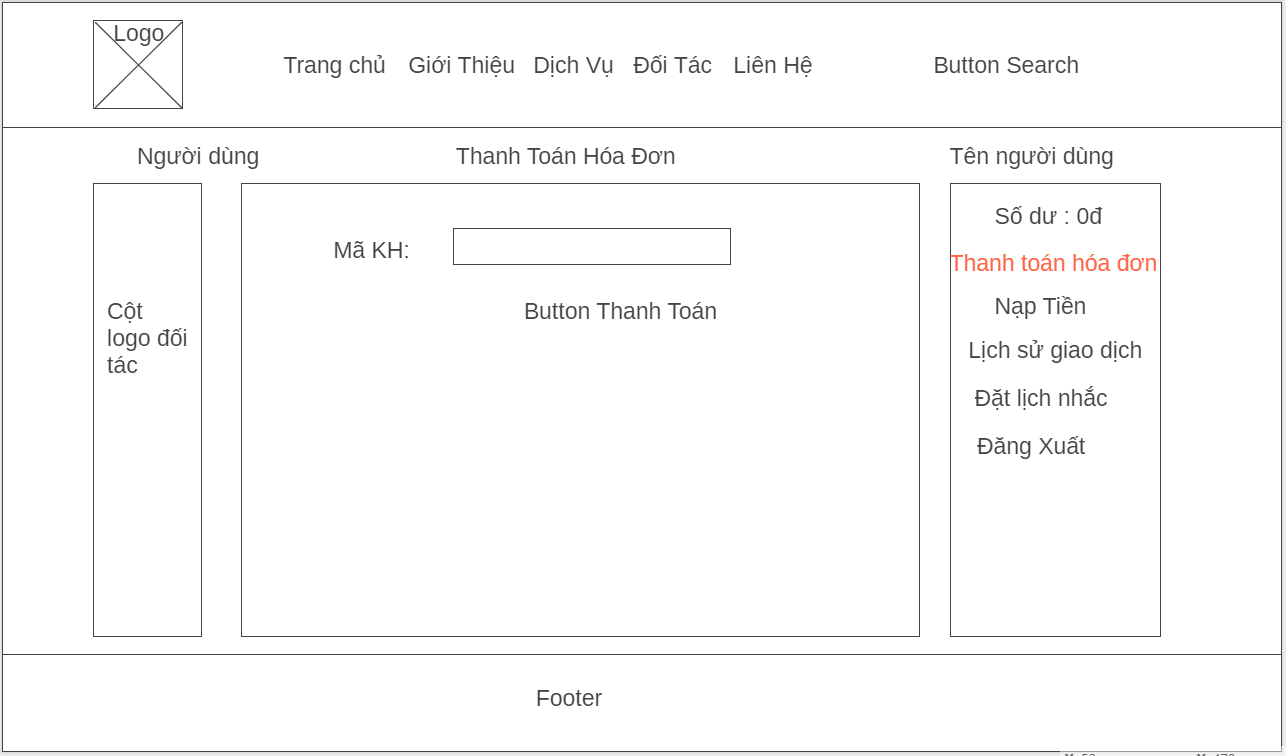
*Hình 3.3 – 3 Trang người quản lý*

Trang Chức Năng Thanh Toán Hóa Đơn:

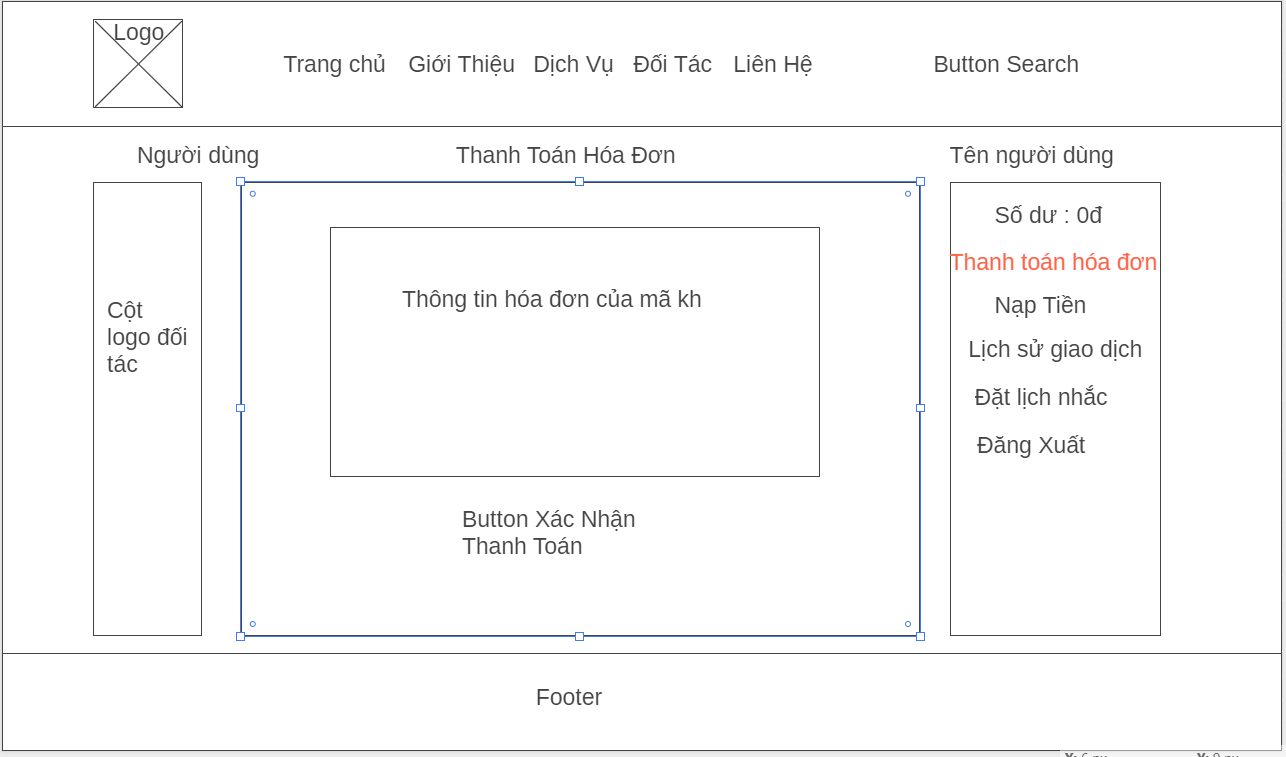
*Hình 3.3 ‑ 4 Trang Chức Năng Thanh Toán Hóa Đơn*



*Hình 3.3 ‑ 5 Trang Chức Năng Thanh Toán Hóa Đơn*

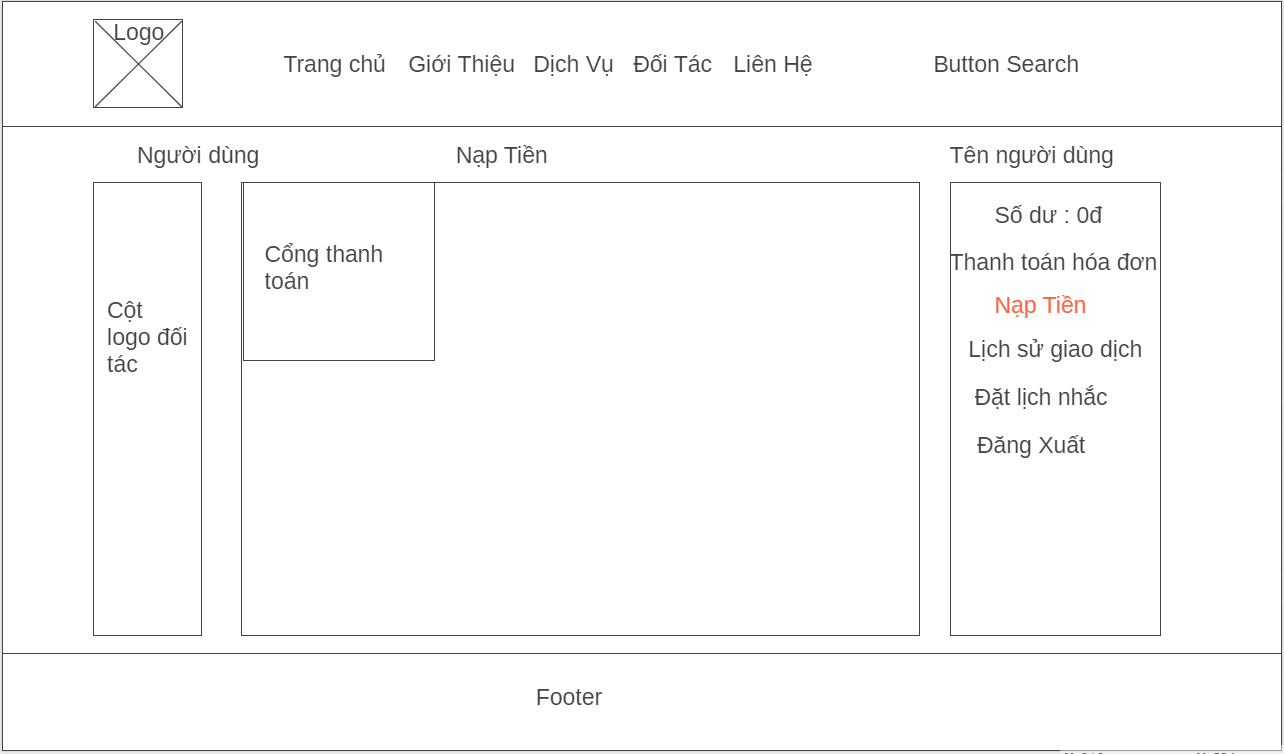


*Hình 3.3 ‑ 6 Trang Chức Năng Thanh Toán Hóa Đơn*

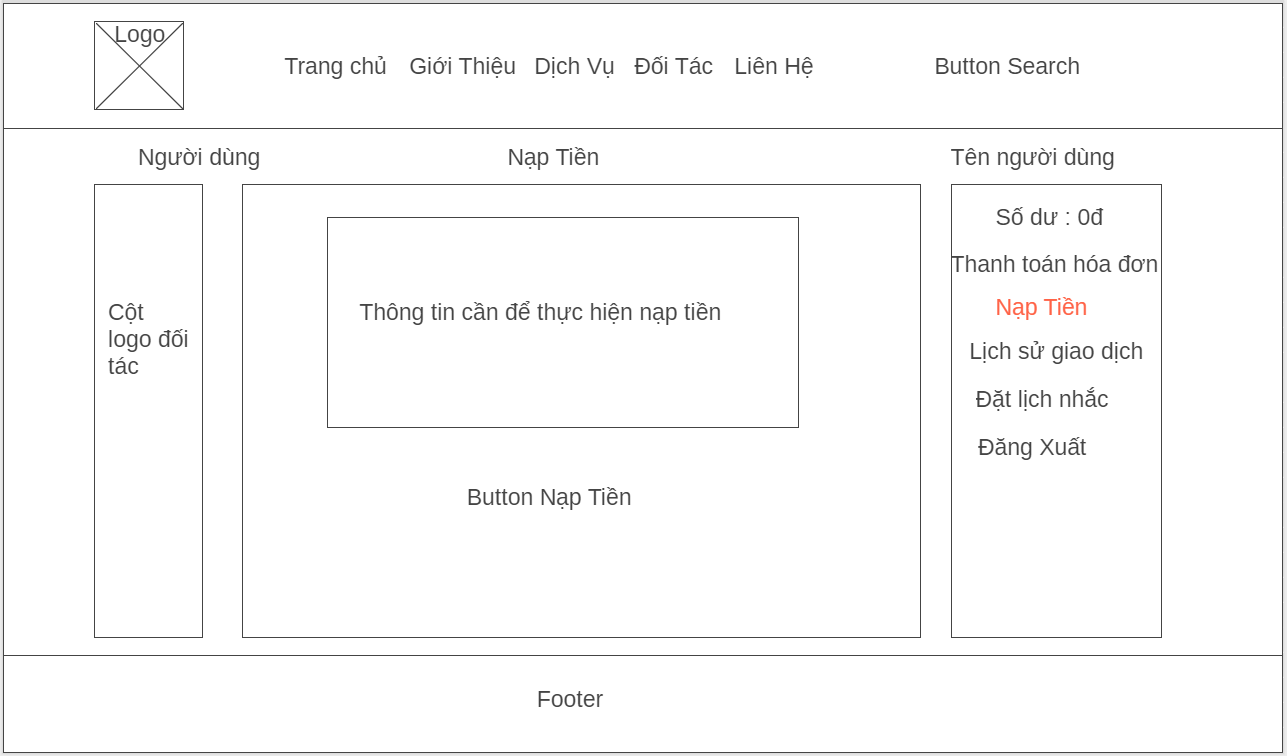


*Hình 3.3 ‑ 7 Trang Chức Năng Thanh Toán Hóa Đơn*

Trang chức năng nạp tiền:

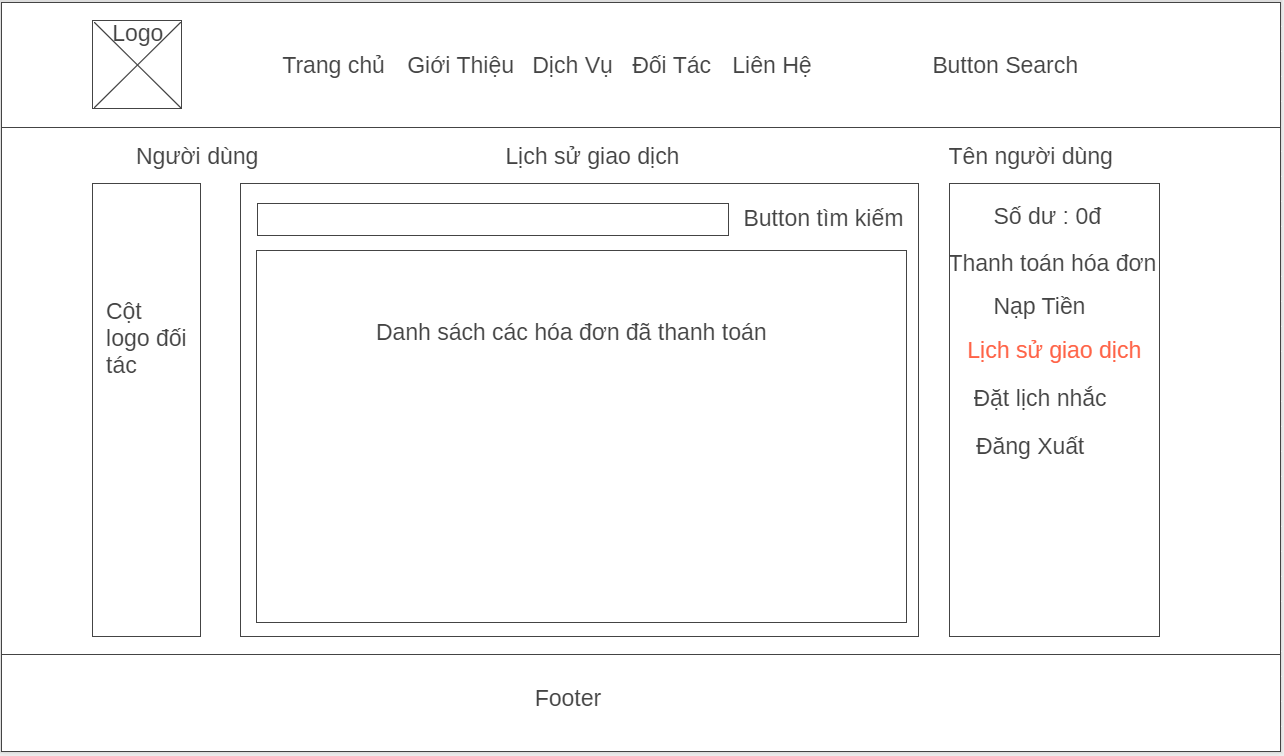


*Hình 3.3 ‑ 8 Trang chức năng nạp tiền*



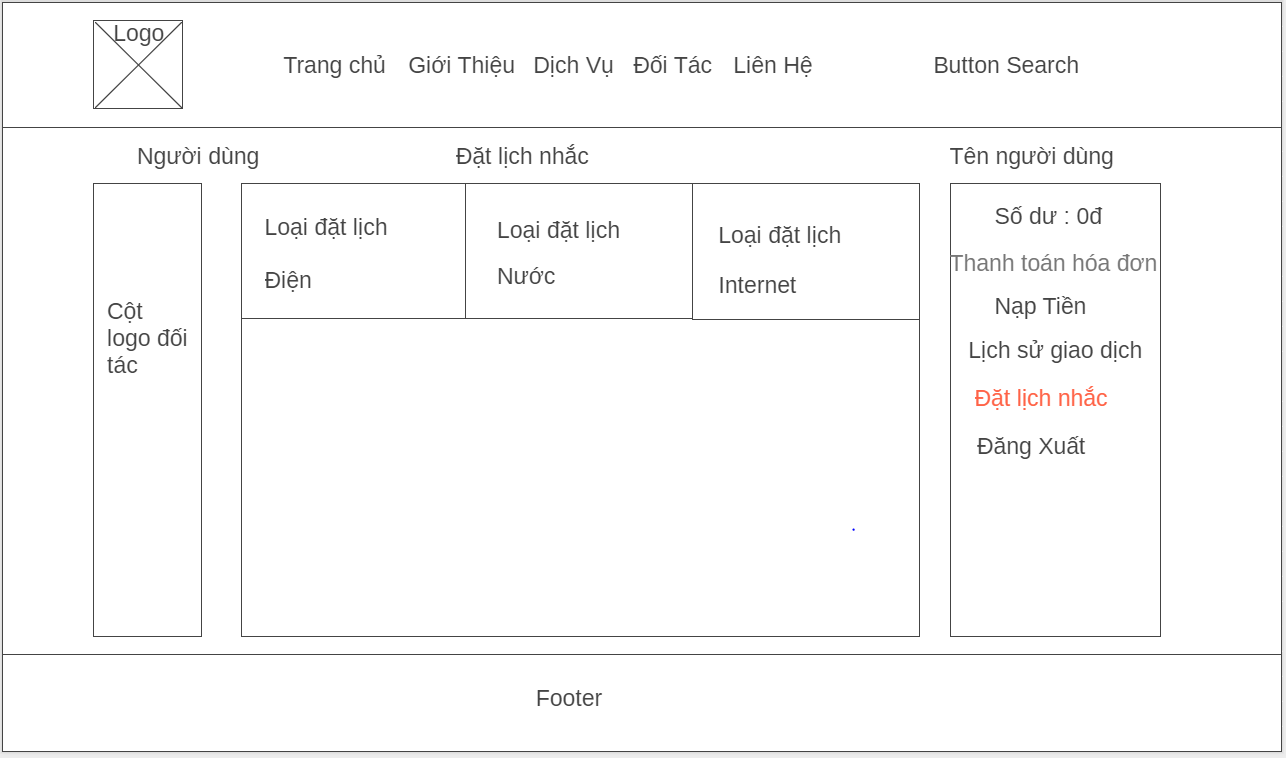
*Hình 3.3 ‑ 9 Trang chức năng nạp tiền*

Trang chức năng lịch sử giao dịch

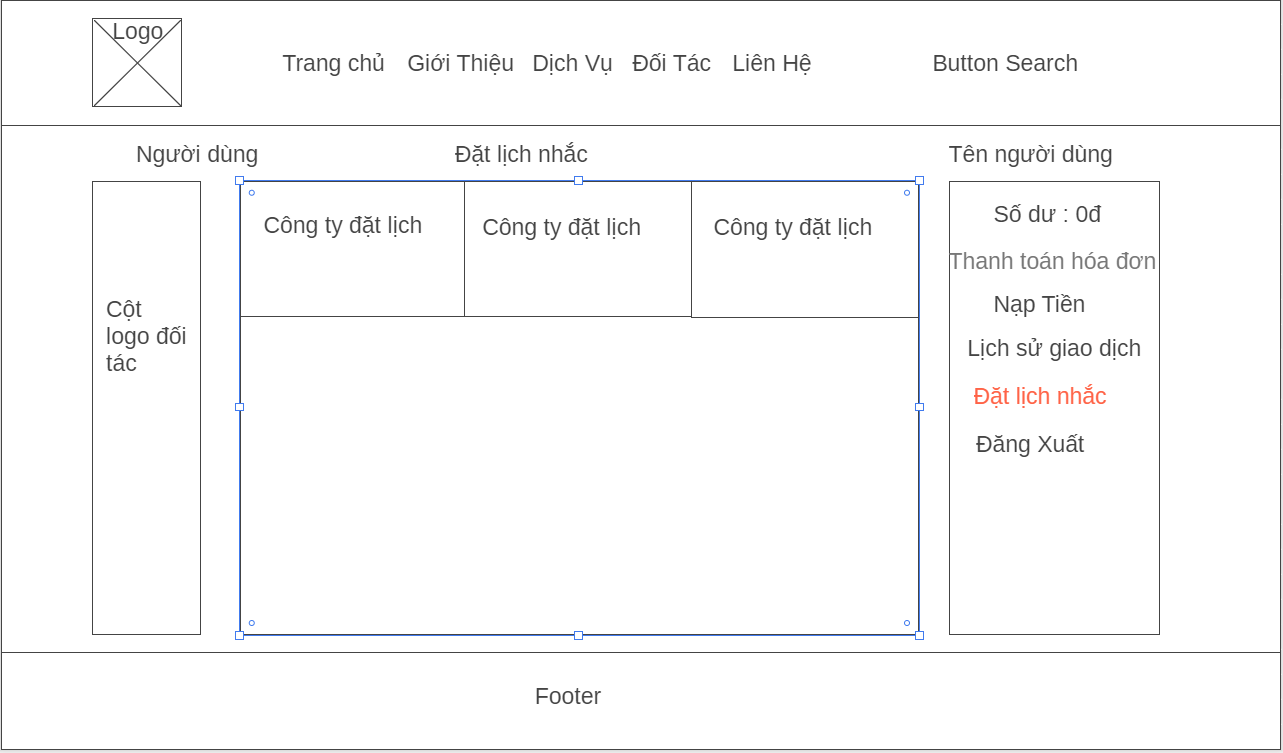


*Hình 3.3 ‑ 10 Trang chức năng lịch sử giao dịch*

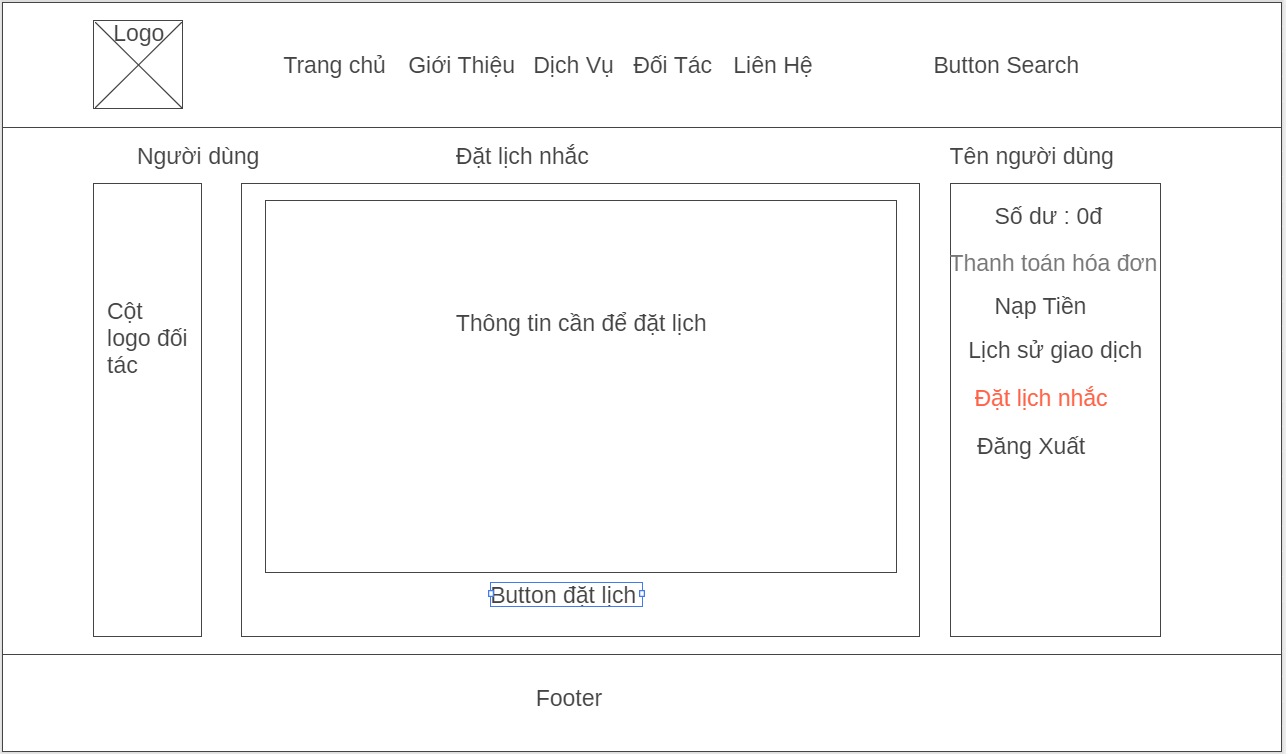
Trang chức năng đặt lịch nhắc



*Hình 3.3 – 11 Trang chức năng đặt lịch nhắc*



*Hình 3.3 ‑ 12 Trang chức năng đặt lịch nhắc*



*Hình 3.3 – 13 Trang chức năng đặt lịch nhắc*

## Deployment diagram



*Hình 3.4 – 1 Deployment diagram*

# : HIỆN THỰC

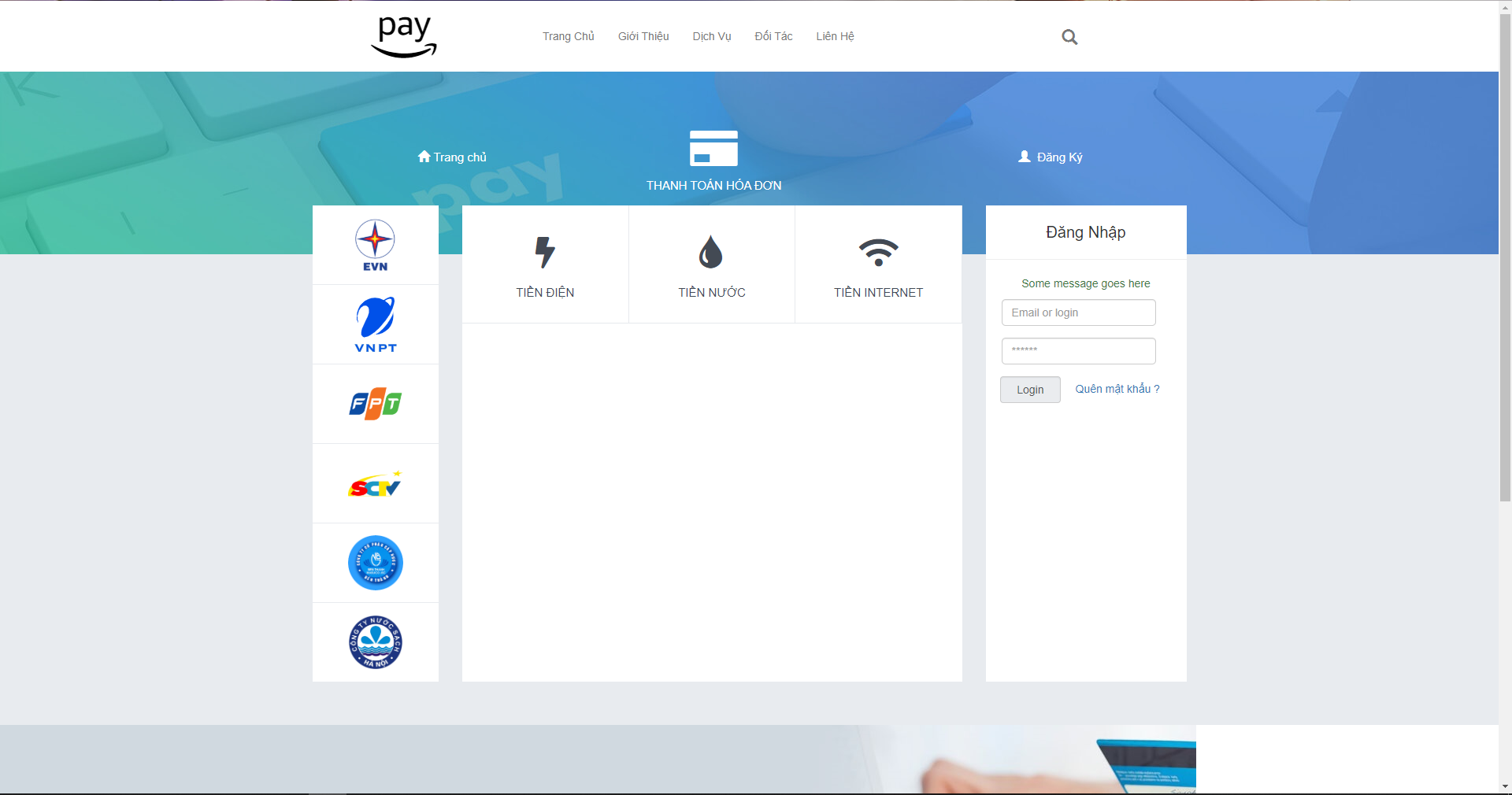
## Cấu hình phần cứng, phần mềm

Đồ án sử dụng các dịch vụ của AWS:

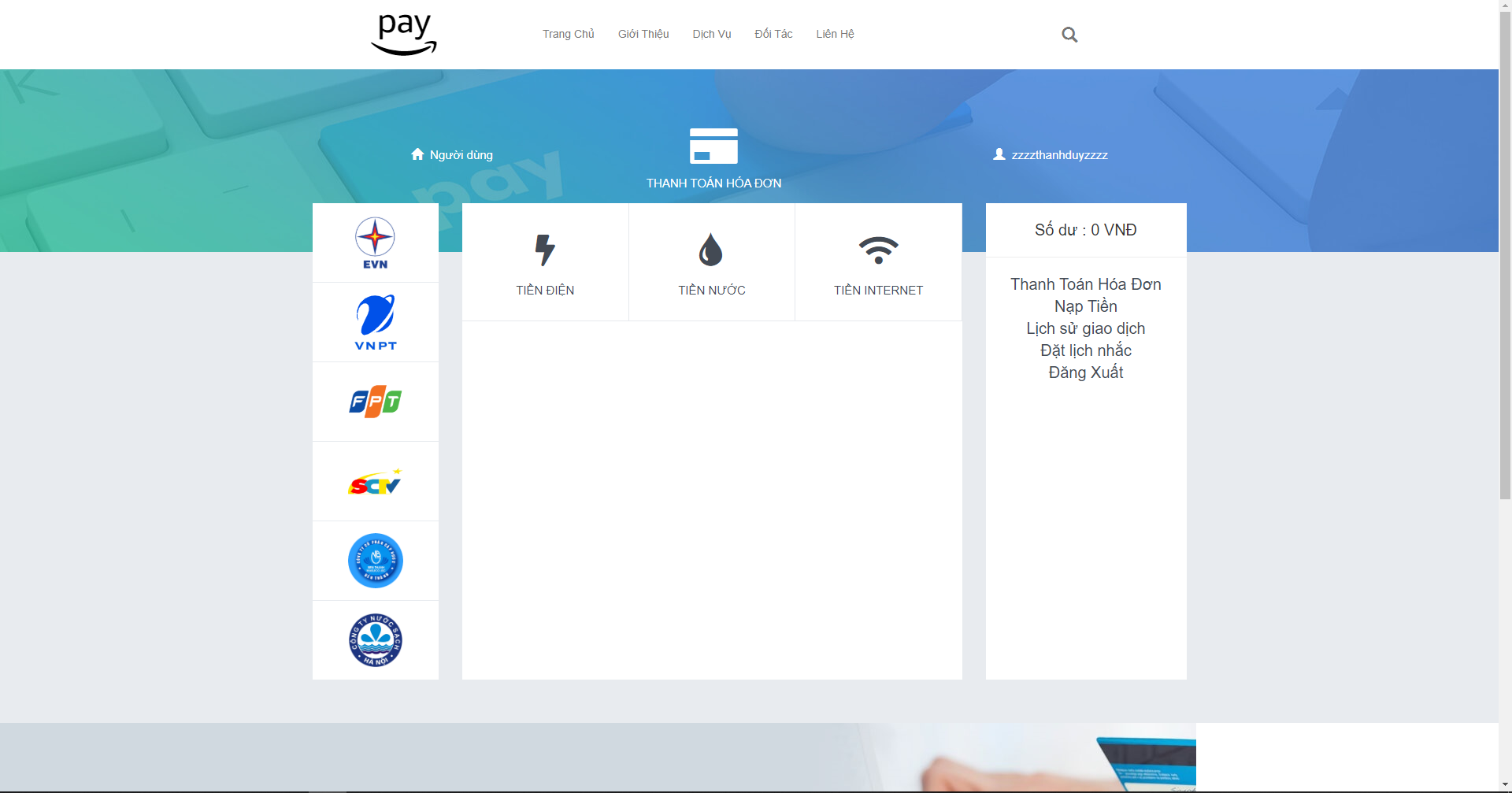
* CSDL DynamoDB
* EC2 Service
* S3 Service
* Kết hợp với nền tảng node.js
* Phần cứng
  + CPU intel Core i5, Ram tối thiểu 4GB,HDD tối thiểu 10GB trống.
* Phần mềm
  + Phía server cần:
    - Visual Studio Code
    - Bộ cài đặt AWSCLI
    - Bộ cài đặt Node.js
  + Phía Client cần
    - Trình duyệt web

## Giao diện của hệ thống

1. Trang chủ và đăng nhập



1. Giao diện người dùng



## Test case

| **Test ID** | **Chức năng** | **Mô tả** | **Điều kiện trước** | **Dữ liệu Test** | **Kết quả mong muốn** |
| --- | --- | --- | --- | --- | --- |
| *1* | Cung cấp thông tin hóa đơn |  | Đã đăng nhập thành công tài khoản doanh nghiệp | Nhập file | Dữ liệu từ file được lưu vào cơ sở dữ liệu |
| *2* | Xem danh sách hóa đơn | Cho phép doanh nghiệp xem các hóa đơn mà khách hàng của họ đã thanh toán | Đã đăng nhập *thành công* vào tài khoản của doanh nghiệp. | Doanh nghiệp nhấp vào chức năng xem danh sách hoá đơn | Hiển thị danh sách các hóa đơn mà khách hàng đã thanh toán |
| 3 | Đăng ký doanh nghiệp | Cho phép doanh nghiệp đăng kí tài khoản doanh nghiệp | Đăng nhập thành công với tài khoản người quản lý. | Test1:   * Tên doanh nghiệp: ”Điện lực TPHCM” * Email: [dlhcm@gmail.com](mailto:dlhcm@gmail.com) * Mật khẩu: “dltphcm123” * Số điện thoại: “1234567890” * Địa chỉ:”Gò vấp, TPHCM” * Loại dịch vụ: Điện * URL Logo: <http://media.homefinder.vn/img/2016/4/23/4e9abae0_1433926676_logo_new_hcmc_3.png?p=rup> | Đăng kí thành công |
| 4 | Đăng ký tài khoản | Cho phép hội viên đăng kí tài khoản hội viên | Truy cập vào website thanh toán hóa đơn | Nhập:   * Tên tài khoản:’abc’ * Email: [abc@gmail.com](mailto:abc@gmail.com) * Mật khẩu: ‘123456’ * Nhập lại MK: ‘123456’ * Số điện thoại:’0123456789’ * Địa chỉ:’Gò vấp’ | Đăng kí thành công |
| Nhập rỗng. | Thông báo yêu cầu nhập |
| Nhập email: ‘aa’ | Yêu cầu nhập đúng định dạng |
| Nhập số điện thoại: 123 | Yêu cầu nhập đúng định dạng |
| 5 | Thanh toán hóa đơn | Usecase cho phép hội viên thực hiện thanh toán các hóa đơn thông qua chức năng thanh toán của website. | Hội viên đã đăng nhập thành công vào hệ thống và có số dư tài khoản lớn hơn số tiền trong hóa đơn. | Nhập đúng mã khách hàng và chọn đúng nhà cung cấp:   * Chọn nhà cung cấp điên EVNSPC * Mã khách hàng: 400001 | Hiển thị form thanh toán và thông báo thành công khi khách hàng nhấn thanh toán |
| Nhập không đúng mã khách hàng:   * Chọn nhà cung cấp điên EVNSPC * Mã khách hàng: 400001 | Thông báo không có khách hàng này |
| 6 | Xem lịch sử giao dịch | Cho phép hội viên xem lại lịch sử giao dịch của mình | Hội viên đã đăng nhập thành công vào hệ thống. |  |  |
| 7 | Đặt lịch nhắc | Cho phép hội viên đặt lịch nhắc thanh toán khi đến thời hạn | Hội viên đăng nhập vào hệ thống vào hệ thống. |  |  |
| 8 | Nạp tiền | Cho phép hội viên nạp tiền vào tài khoản ảo của website | Hội viên đã đăng nhập thành công vào hệ thống. |  |  |
| 9 | Lập hóa đơn | Hệ thống tạo ra hóa đơn lấy thông tin từ hóa đơn của doanh nghiệp | Hội viên thực hiện use case thanh toán hóa đơn. |  |  |
| 10 | Thông báo email | Use case thực hiện việc gửi email tới hội viên đã thực hiện đặt lịch nhắc trước đó. | Đã thực hiện use case Lập hóa đơn và use case Đặt lịch nhắc |  |  |
| 11 | Quản lý Hội viên | Hiển thị *thống kê thông tin về hội viên* | Hội viên đã đăng nhập thành công vào hệ thống. |  |  |
| 12 | Quản lý doanh nghiệp | *Hiển thị các thông tin thống kê về doanh nghiệp* | Doanh nghiệp đã đăng nhập thành công vào hệ thống. |  |  |
| 13 | Thống kê phí doanh vụ | Use case thực hiện việc thông kê các khoản phí dịch vụ mà doanh nghiệp đã đăng ký và thu phí khi Hội viên thực hiện thanh toán hóa đơn | Admin đã đăng nhập thành công vào hệ thống. |  |  |

## Báo cáo kết quả test (Test report)

| **Test ID** | **Ngày testing** | **Người tham gia Test** | **Pass/Fail** | **Độ nghiêm trọng** | **Tóm tắt lỗi** |
| --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |

# : KẾT LUẬN

## Kết quả đạt được

* Tìm hiểu về công nghệ điện toán đám mây kết hợp để xây dựn ứng dụng
* Sử dụng HTML kết hợp với CSS, JavaScript và Bootstrap để tạo giao diện cho website.
* Sử dụng hệ quản trị cơ sở dữ liệu No SQL DynamoDB kết hợp với Platform NodeJS trên công cụ Visual Studio Code để thực hiện và triển khai các chức năng cho website.
* Triển khai áp dụng trên dịch vụ của AWS như EC2, S3 vào website để tăng cường khả năng lưu trữ dữ liệu và bảo mật, kích cỡ linh hoạt trên điện toán đám mây.
* Tải các Dependencies hỗ trợ cho Platform NodeJS để thực hiện ứng dụng như express (dễ dàng để phát triển các ứng dụng nhanh dựa trên Node.js cho các ứng dụng Web), body-parser (lấy dữ liệu trong form với phương thức post), template ejs (tạo giao diện web), fs, paypal-rest-sdk, passport (để đăng nhập và xác thực tài khoản), express-session (lưu lại session tài khoản đã đăng nhập) các dependencies sẽ hổ trợ triển khai các chức năng nhanh chóng và tiện lợi.
* Ứng dụng đã được xây dựng với các chức năng cơ bản của một website thanh toán điện tử online:
  + **Đối với Doanh Nghiệp:**
* Cung cấp thông tin hóa đơn.
* Xem danh sách hóa đơn.
* Hủy hợp đồng.
* Đăng ký doanh nghiệp.
* **Đối với Hội viên:**

- Thanh toán hóa đơn

- Xem lịch sử giao dịch.

* Xem hóa đơn
* Đặt lịch nhắc
* Nạp tiền
* **Đối với Người Quản Lý:**
* Lập hóa đơn.
* Gửi hóa đơn.
* Quản lý Hội viên.
* Quản lý doanh nghiệp.

## Hạn chế của đồ án

* Trong quá trình phát triển nhóm đã có những hạn chế sau:
  + Do dự án chỉ ở mức demo nên không có những cơ sở dữ liệu thực tế từ phía doanh nghiệp để liên kết mà chỉ ở mức tự tạo và áp cứng.

## Hướng phát triển

# TEST PLAN

## Giới thiệu

* 1. **Mục tiêu kiểm thử**
* Lập kế hoạch cụ thể kiểm thử cho ứng dụng quản lý thanh toán hóa đơn trực tuyến.
* Đảm bảo việc kế hoạch kiểm thử được thực hiện đúng kế hoạch và đầy đủ, phân chia công việc trong nhóm phù hợp và hợp lý.
* Nâng cao mức độ tiện lợi của ứng dụng.
* Đảm bảo ứng dụng không còn lỗi khi đưa ra cho người sử dụng.
  1. **Phân tích ứng dụng**
* Đối tượng sử dụng website là các doanh nghiệp và khách hàng đã đăng ký với website.
* Mục tiêu của Website dùng để thực hiện các chức năng thanh toán hóa đơn, nạp tiền, đặt lịch nhắc cho khách hàng.
* Website hoạt động khi server đang chạy và người dụng mở trang web.
* Phần mềm:

|  |  |  |
| --- | --- | --- |
| **Tên phần mềm** | **Phiên bản** | **Loại** |
| NodeJS | 1.37.1.0 |  |
| DynamoDB |  | Cơ sở dữ liệu |
| Microsoft Windows 10 | 10 | Hệ điều hành |

*Bảng 4-1 yêu cầu phần mềm*

* Phần cứng:

|  |  |  |  |
| --- | --- | --- | --- |
| **CPU** | **RAM** | **HDD** | **Architecture** |
| Intel Core i7, 2.3 GHz | 8 GB | 1 TB | 64 bit |

*Bảng 4-2 yêu cầu phần cứng*

* 1. **Các ràng buộc về quy trình kiểm thử**

|  |  |  |
| --- | --- | --- |
| TT | Tài nguyên | Mô tả |
| 1 | Máy chủ | Server, Database server và API cài ứng dụng sẽ được kiểm thử. |
| 2 | Công cụ | Nodejs kiểm tra server đang chạy.  Dynamodb kiểm tra cơ sở dữ liệu.  Postman kiểm tra API cho ứng dụng. |
| 3 | Mạng | Sử dụng Internet để chạy thử và kiểm tra ứng dụng. |
| 4 | Máy tính Client | Sử dụng hệ điều hành window 10.  Sử dụng trình duyệt (Firefox, Chrome) chạy trang website. |
| 5 | Các tài nguyên khác | Tài khoản AWS (AWS Starter) |

*Bảng 4-3 môi trường kiểm thử*

* 1. **Phạm vi kiểm thử**
* Theo các chức năng đã Đặc tả, đề tài quản lý thanh toán hóa đơn trực tuyến chỉ tập trung vào kiểm thử tất cả các chức năng và giao diện bên ngoài của website (trên các trình duyệt khác nhau như Firefox, Chrome).
* Không thực hiện kiểm thử mức độ hiệu suất ứng dụng, logic cơ sở dữ liệu.
  1. **Rủi ro**

|  |  |
| --- | --- |
| Rủi ro | Giải pháp |
| Thành viên trong nhóm thiếu các kỹ năng cần thiết để kiểm thử website ứng dụng. | Tự học hỏi, tìm hiểu thêm, tham gia các khóa học liên quan đến kiểm thử để nâng cao kỹ năng của các thành viên nhóm. |
| Lịch trình thực hiện đề tài trong thời gian ngắn, công việc nhiều, công nghệ tiếp cận là mới. | Đặt mức độ ưu tiên kiểm thử cho từng hoạt động kiểm thử. |
| Ước lượng thời gian thực thi ứng dụng chưa hợp lý và vượt quá | Xác định phạm vi rõ ràng khi bắt đầu ứng dụng, lưu ý khi lập kế hoạch dự án và phải theo dõi tiến độ trong quá trình làm. |

*Bảng 4-4 rủi ro và giải pháp*

## Yêu cầu kiểm thử

* 1. **Danh sách kiểm thử**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả vắn tắt chức năng (outline)** | **Ước lượng số lượng tình huống kiểm kiểm thử (test case).** | **Ghi chú** |
| **1** | Cung cấp thông tin hóa đơn | Doanh nghiệp nhập các thông tin về hóa đơn | 1 |  |
| **2** | Xem danh sách hóa đơn | Hiển thị danh sách hóa đơn | 1 |  |
| **3** | Đăng ký doanh nghiệp | Doanh nghiệp đăng kí tài khoản hệ thống | 5 |  |
| **4** | Đăng ký tài khoản | Khách hàng đăng kí tài khoản | 5 |  |
| **5** | Thanh toán hóa đơn | Khách hàng thanh toán hóa đơn | 5 |  |
| **6** | Xem lịch sử giao dịch | Xem lịch sử giao dịch của tài khoản | 3 |  |
| **7** | Đặt lịch nhắc | Đặt lịch nhắc thanh toán | 1 |  |
| **8** | Nạp tiền | Khách hàng nạp tiền vào tài khoản hệ thống | 5 |  |
| **9** | Lập hóa đơn | Hệ thống lập hóa đơn cho chức năng thanh toán | 2 |  |
| **10** | Thông báo email | Hệ thống thông báo lịch thanh toán. | 2 |  |
| **11** | Quản lý hội viên | Thống kê các thông tin về hội viên | 2 |  |
| **12** | Quản lý doanh nghiệp | Quản lý thông tin về doanh nghiệp | 1 |  |
| **13** | Thống kê doanh thu | Thống kê doanh thu | 1 |  |

*Bảng 4-5 danh sách các chức năng kiểm thử*

* 1. **Điều kiện chấp nhận**
* Phạm vi bao phủ của Test (Test coverage): trên 90% chức năng hệ thống.
* Số lượng các trường hợp kiểm thử (Đơn vị / Tích hợp / Các trường hợp thử nghiệm hệ thống): hoàn thành tất cả các trường hợp kiểm thử.
* Số lượng lỗi / Trọng số lỗi: số lượng lỗi ít và chấp nhận được, hạn chế tối thiểu trọng lỗi đến mức có thể.

## Kỹ thuật kiểm thử

* 1. **Kiểm thử đơn vị**

*Sau khi code xong một phần thì thực hiện kiểm thử đoạn code, nếu có lỗi thì chỉnh sửa lại.*

* 1. **Kiểm thử Module/chức năng**

|  |  |
| --- | --- |
| **Mục tiêu** | Đảm bảo chức năng kiểm thử với mục tiên thích hợp, bao gồm dữ liệu đầu vào, navigation, quá trình xử lý và kết quả nhận được. |
| **Kỹ thuật** | Thực thi mỗi use case, luồng hoạt động cho use-case , hoặc chức năng, dùng dữ liệu đúng, dữ liệu không đúng để xác định:  -   Kết quả mong muốn khi dữ liệu đưa vào là đúng  -   Các thông báo lỗi, cảnh báo hiển thị khi dữ liệu không chính xác đưa vào.  -   Quy tắc nghiệp vụ áp dụng cho trường hợp test.  - Sử dụng các công cụ kiểm thử (Test tool) … |
| **Tiêu chí hoàn thành chức năng** | -   Tất cả các kế hoạch kiểm thử cho chức năng được thực hiện  -   Chỉnh sửa các lỗi đã phát hiện |
| **Điều kiện đặc biệt** | Xác định hoặc mô tả các mục hoặc vấn đề (bên trong hoặc bên ngoài) mà có ảnh hưởng đến việc triển khai và thực hiện kiểm thử chức năng. |

*Bảng 4-6 kiểm thử chức năng*

* 1. **kiểm thử giao diện**

|  |  |
| --- | --- |
| **Mục tiêu** | Kiểm thử các vấn đề:  + Di chuyển từ cửa sổ giao diện này sang giao diện khác, từ field này sang field khác dùng phím tab, di chuyển chuột hoặc các phím tổ hợp)  + Đối tượng cửa sổ, menu, kích cỡ, vị trí, trạng thái. |
| **Kỹ thuật** | Tạo hoặc sửa đổi các kiểm thử cho mỗi cửa sổ để xác minh điều hướng và trạng thái đối tượng thích hợp cho từng cửa sổ ứng dụng |
| **Tiêu chí hoàn thành chức năng** | Mỗi cửa sổ được test hoàn thiện và các lỗi xảy ra nằm trong phạm vi chấp nhận được |
| **Điều kiện đặc biệt** | Không phải tất cả các thuộc tính tùy chỉnh của bên thứ ba có thẻ truy cập được |

*Bảng 4-7 kiểm thử giao diện ứng dụng*

* 1. **kiểm thử hệ thống**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thành viên** | **Vai trò** | **Nhiệm vụ** | **Ước lượng thời gian thực hiện** |
| 1 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Test Manager | Quản lý toàn bộ việc kiểm thử ứng dụng .  Xác định tài nguyên phù hợp cho việc kiểm thử. |  |
| 2 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Tester | Thực hiện các kiểm thử, log kết quả, cáo cáo các lỗi. |  |
| 3 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Developer thực hiện Test | Thực hiện các test cases, test program, test suite… |  |
| 4 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Test Administrator | Xây dựng và đảm bảo môi trường kiểm thử, tài nguyên được quản lý và duy trì.  Hỗ trợ Tester sử dụng môi trường kiểm thử để thực hiện kiểm thử. |  |
| 5 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Thành viên SQA | Phụ trách đảm bảo chất lượng cho ứng dụng phần mềm/website.  Kiểm thử để xác nhận xem quy trình kiểm thử có đáp ứng các yêu cầu đã được xác định hay không. |  |

*Bảng 4-8 kế hoạch kiểm thử*

## IV. Kế hoạch nguồn nhân lực thực hiện kiểm thử cho ứng dụng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Thành viên** | **Vai trò** | **Nhiệm vụ** | **Ước lượng thời gian thực hiện** |
| 1 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Test Manager | Quản lý toàn bộ việc kiểm thử ứng dụng .  Xác định tài nguyên phù hợp cho việc kiểm thử. |  |
| 2 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Tester | Thực hiện các kiểm thử, log kết quả, cáo cáo các lỗi. |  |
| 3 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Developer thực hiện Test | Thực hiện các test cases, test program, test suite… |  |
| 4 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Test Administrator | Xây dựng và đảm bảo môi trường kiểm thử, tài nguyên được quản lý và duy trì.  Hỗ trợ Tester sử dụng môi trường kiểm thử để thực hiện kiểm thử. |  |
| 5 | Nguyễn Xuân Trường Giang, Lê Anh Tú, Phạm Thanh Duy | Thành viên SQA | Phụ trách đảm bảo chất lượng cho ứng dụng phần mềm/website.  Kiểm thử để xác nhận xem quy trình kiểm thử có đáp ứng các yêu cầu đã được xác định hay không. |  |

*Bảng 4-8 kế hoạch kiểm thử*

# TÀI LIỆU THAM KHẢO

**Các tài liệu từ silde**

Các tài liệu từ Internet

1. <https://nodejs.org/en/docs/>
2. <https://viblo.asia/p/gioi-thieu-ve-dynamodb-phan-1-jamoG87nMz8P>
3. <https://viblo.asia/p/mot-cai-nhin-tong-quan-nhat-ve-nodejs-Ljy5VeJ3lra>
4. <https://expressjs.com/en/guide/routing.html>
5. <https://viblo.asia/p/nodejs-tutorial-phan-4-express-framework-924lJXpNKPM>
6. <https://aws.amazon.com/vi/swf/>
7. <https://aws.amazon.com/vi/dynamodb/>
8. <https://aws.amazon.com/vi/ebs/>
9. <https://aws.amazon.com/vi/sns/>
10. <https://zxthanhhuy.wordpress.com/2015/08/04/uu-va-nhuoc-diem-cua-nosql/>