HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

**FACULTY OF COMPUTER SCIENCE AND ENGINEERING**

🙠🕮🙢



PROJECT REPORT

**RESTAURANT POS 2.0**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEAM**  **MEMBER** | 20  NGUYỄN VĂN MINH HẢI  NGUYỄN ANH KHOA  TRẦN MINH KHÔI  VÕ NGUYỄN MINH TÚ  HỒ VĨNH TRUNG HẬU | 1913260  1611616  1913845  1915828  1810912 | |
| **LECTURER** | Assoc. Prof. Quan Thanh Tho | |

🙠🕮🙢

**Table of contents**

[NỘI DUNG 1](#_Toc86006738)

[GIỚI THIỆU 1](#_Toc86006739)

[Bối cảnh của dự án 1](#_Toc86006740)

[Stakeholders của dự án 1](#_Toc86006741)

[Internal Stakeholders 1](#_Toc86006742)

[External Stakeholders 2](#_Toc86006743)

[Các yêu cầu dự kiến của ứng dụng 2](#_Toc86006744)

[ĐẶC TẢ CHI TIẾT 3](#_Toc86006745)

[Non-Functional Requirements 3](#_Toc86006746)

[1) Non-functional requirements cho toàn hệ thống 3](#_Toc86006747)

[a) Yêu cầu của sản phẩm 3](#_Toc86006748)

[b) Yêu cầu về tổ chức hệ thống 3](#_Toc86006749)

[2) Non-functional requirements cho các feature cụ thể 4](#_Toc86006750)

[Functional Requirement 4](#_Toc86006751)

[1) Functions 4](#_Toc86006752)

[2) Use-case diagram tổng quát cho toàn hệ thống 5](#_Toc86006753)

[3) Use-case diagram cho feature Food Ordering và đặc tả chi tiết 6](#_Toc86006754)

[View Menu 6](#_Toc86006755)

[View Order History 7](#_Toc86006756)

[View Order Status 8](#_Toc86006757)

[Modify Cart 8](#_Toc86006758)

[Give Note 9](#_Toc86006759)

[Cancel Order 10](#_Toc86006760)

[Conduct Purchase 10](#_Toc86006761)

[4) Activity Diagram 12](#_Toc86006762)

[5) Sequence Diagram 13](#_Toc86006763)

[View Menu 13](#_Toc86006764)

[Modify Cart 13](#_Toc86006765)

[Conduct Purchase 14](#_Toc86006766)

[Give Note 15](#_Toc86006767)

[View Order Status 16](#_Toc86006768)

[Cancel Order 16](#_Toc86006769)

[View Order History 17](#_Toc86006770)

[6) Class Diagram 17](#_Toc86006771)

[7) Mô tả Architectural Approach của nhóm 18](#_Toc86006772)

[a) Khái niệm 18](#_Toc86006773)

[b) Các thành phần trong mô hình MVC 18](#_Toc86006774)

[c) Luồng đi dữ liệu 19](#_Toc86006775)

[d) Ưu điểm và nhược điểm của mô hình MVC 21](#_Toc86006776)

[Ưu điểm mô hình MVC 21](#_Toc86006777)

[Nhược điểm mô hình MVC 22](#_Toc86006778)

[8) Deployment Diagram 22](#_Toc86006779)

[TỔNG HỢP ĐÓNG GÓP CỦA CÁC THÀNH VIÊN 23](#_Toc86006780)

NỘI DUNG

GIỚI THIỆU

Bối cảnh của dự án

* Trong tình hình dịch covid 19 diễn ra ngày càng phức tạp và nguy hiểm, hệ thống bán hàng của một nhà hàng cần thay đổi. Trong giai đoạn này, hệ thống bán hàng tự động được coi là giải pháp hữu ích. Sự cần thiết của việc hỗ trợ dịch vụ cho khách hàng mang về và việc hỗ trợ trong việc thanh toán nhanh chóng cho quản lý nhà hàng đặc biệt là các nhà hàng có hệ thống lớn.
* Thực đơn được in bằng giấy ngày càng đang lỗi thời, và thiếu tiện lợi trong việc thay đổi món ăn, tốn khoản chi phí cho tiền hóa đơn. Việc chọn món ăn theo cách truyền thống gây mất thời gian và sẻ gặp những sự cố sai sót. Đặc biệt việc tiếp xúc trực tiếp không đảm bảo vệ sinh, nguy cơ lây lan dịch bệnh.

Tất cả điều đó dẫn đến việc phải có hệ thống POS trên nền tảng web để phục vụ quá trình xử lý chọn món ăn, đặt hàng và thanh toán một cách tiện lợi nhất cho khách hàng và nhà hàng.

Stakeholders của dự án

Internal Stakeholders

Người sử dụng

* Khách hàng của nhà hàng.
* Nhân viên tại các điểm bán hàng.
* Đầu bếp.
* Quản lý nhà hàng.
* Bộ phận IT của nhà hàng.

Người phát triển ứng dụng

* + - Designer. Người phát triển giao diện cho ứng dụng.
    - Quản lý dự án phát triển ứng dụng.
    - Các developer tham gia phát triển ứng dụng.
    - Tester. Người kiểm thử ứng dụng.

External Stakeholders

* Các cơ quan để đảm bảo quyền lợi của người dùng, bảo mật thông tin người dùng.
* Bộ y tế.
* Các bên liên quan như ngân hàng, thẻ tín dụng.

Các yêu cầu dự kiến của ứng dụng

Xây dựng website bán hàng có các tính năng sau:

* Truy cập bằng QR code.
* Đăng ký, đăng nhập.
* Đặt chỗ.
* Hiển thị danh sách món ăn của nhà hàng.
* Hiển thị món ăn theo loại.
* Chọn món ăn thêm vào giỏ hàng.
* Chỉnh sửa giỏ hàng.
* Đặt đơn hàng.
* Thanh toán đơn hàng.
* Quản lý khách hàng.
* Quản lý thực đơn.
* Xuất thông tin đơn hàng đã đặt cho nhân viên.
* Lưu trữ số đơn hàng, tổng tiền và doanh thu cho nhà hàng.

Yêu cầu hệ thống:

* Dễ sử dụng, thời gian đáp ứng nhanh.
* Dễ nâng cấp và bảo trì.
* Bảo mật thông tin khách hàng

Các ràng buộc của hệ thống

* Xây dựng hệ thống trên nền tảng web, quét mã QR.
* Chỉ thanh toán bằng tài khoản Momo, ngân hàng.
* Mỗi tài khoản chỉ đặt một đơn hàng.
* Món ăn được hiển thị theo thực đơn.
* Đăng ký, đăng nhập bằng username và password.
* Chỉ quản lý nhà hàng mới được chỉnh sửa menu của nhà hàng, và được nhận báo cáo về số đơn hàng, tổng tiền và doanh thu của nhà hàng

ĐẶC TẢ CHI TIẾT

Non-Functional Requirements

1. Non-functional requirements cho toàn hệ thống
2. Yêu cầu của sản phẩm

Yêu cầu về cách thức sử dụng

* Giao diện đơn giản, người dùng chỉ cần tối đa 5 lần click chuột là đạt được mục đích

Yêu cầu về tính hiệu quả

* Hệ thống luôn trong tình trạng hoạt động (trừ bảo trì) (24/24)
* Có thể thực hiện tối thiểu 300 giao dịch mỗi ngày

Yêu cầu về bảo mật

* Chỉ Admin account mới được cấp quyền chỉnh sửa hệ thống
* Thông tin của khách hàng phải được giữ bí mật

1. Yêu cầu về tổ chức hệ thống

Yêu cầu về môi trường của ứng dụng

* Hệ thống được xây dựng bằng công nghệ web và có thể dùng QR code, nên người dùng có thể dùng ứng dụng mà không cần cài đặt.
* Không có sự tiếp xúc trực tiếp giữa nhân viên nhà hàng (phục vụ, đầu bếp...) và khách hàng

Yêu cầu về hệ thống

* Hệ thống có thể được sử dụng từ các thiết bị thông dụng (smartphone, laptop, tablet)

1. Non-functional requirements cho các feature cụ thể

Ordering Food

* Thời gian từ lúc đặt đơn đến khi có món ăn không được quá 30ph
* Tình trạng Menu, đơn hàng phải được cập nhật liên tục (chu kỳ <5s)

Table Reservation

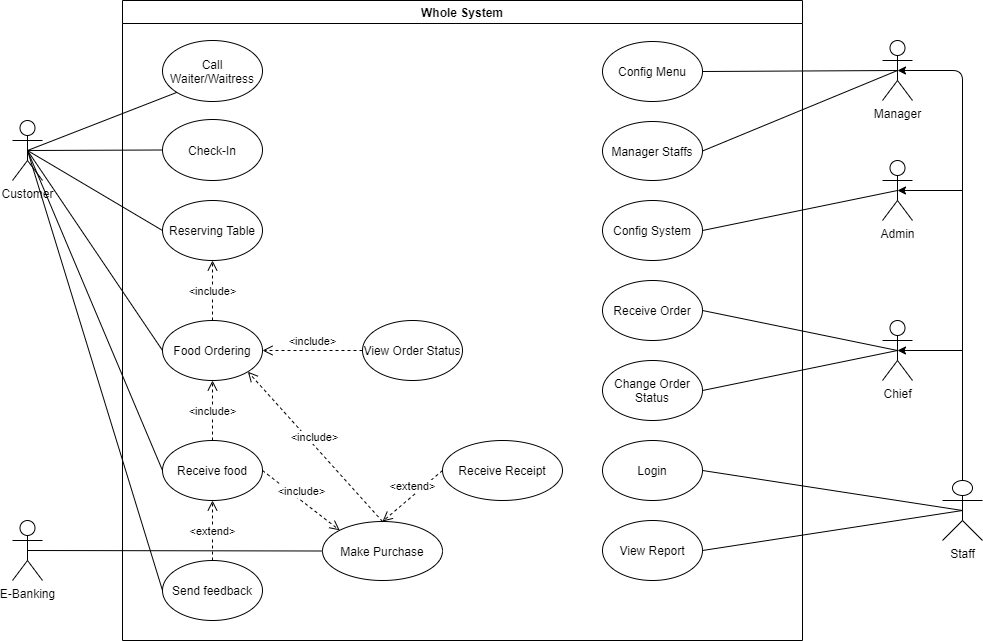
* Tình trạng bàn trống phải được cập nhật liên tục (chu kỳ <5s)
* Không có tình trạng 2 khách cùng đặt 1 bàn cùng 1 thời điểm

Functional Requirement

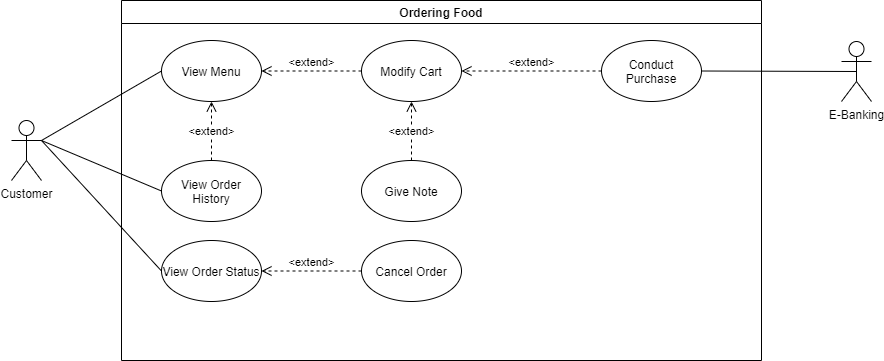
1. Functions

* Food Ordering
* Reserving Table
* Config Menu
* Manage Customers
* Config System
* View Order Status
* Change Order Status
* View Report
* Send Feedback
* Call waiter/waitress
* Receive Receipt
* Make Purchase
* Check-In
* Login

1. Use-case diagram tổng quát cho toàn hệ thống



1. Use-case diagram cho feature Food Ordering và đặc tả chi tiết



**Scenario**

View Menu

|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | View Menu |
| **Description** | Use-Case cho phép người dùng có thể xem Menu |
| **Actor** | Khách hàng |
| **Trigger Condition** | Sau khi khách hàng thực hiện check-in thành công hoặc nhấn vào button Menu trên UI |
| **Pre-Conditions** | -User đã thực hiện check-in thông qua việc quét mã QR trên bàn |
| **Post-Conditions** | -Giao diện Menu hiện ra |
| **Main-Event Stream** | 1. Thỏa mãn Trigger Condition 2. Hệ thống hiển thị giao diện Menu 3. Hệ thống thực hiện làm mới Menu mỗi 5s |
| **Sub-Event Stream** | A-Người dùng mất kết nối mạng  -Gửi thông báo mất kết nối, yêu cầu tái kết nối  B-Hệ thống bảo trì  -Gửi thông báo hệ thống bảo trì  C-Cơ sở dữ liệu mất kết nối   1. Gửi thông báo mất kết nối 2. Khi tái kết nối, quay lại bước 2 ở Main Event |
| **Non-Functional Requirements** | Không |

View Order History

|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | View Order History |
| **Description** | Người dùng có thể xem lại lịch sử đặt món của mình (được đồng bộ với tài khoản google dùng để check-in) |
| **Actor** | Khách hàng |
| **Trigger Condition** | Người dùng click button Xem Lịch Sử Đơn Hàng |
| **Pre-Conditions** | User đã thực hiện check-in thông qua việc quét mã QR trên bàn |
| **Post-Conditions** | Hệ thống hiển thị giao diện Order History |
| **Main-Event Stream** | 1. Người dùng click button Xem Lịch Sử Đơn Hàng 2. Hệ thống hiển thị giao diện Lịch Sử Đơn Hàng |
| **Sub-Event Stream** | A. Người dùng nhấn Quay Lại button   * Hệ Thống hiển thị giao diện trước khi vào Lịch Sử Đơn Hàng |
| **Non-Functional Requirements** | Thông tin khách hàng phải được giữ bí mật |

View Order Status

|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | View Order Status |
| **Description** | Người dùng có thể xem trạng thái của các đơn hàng hiện tại |
| **Actor** | Khách hàng |
| **Trigger Condition** | Người dùng click button view order status |
| **Pre-Conditions** | User đã thực hiện check-in thông qua việc quét mã QR trên bàn |
| **Post-Conditions** | Hệ thống hiển thị trạng thái đơn hàng |
| **Main-Event Stream** | 1. Người dùng click vào button view order status 2. Hệ thống hiển thị trạng thái đơn hàng |
| **Sub-Event Stream** | A. Người dùng nhấn quay lại   * Hệ thống quay lại giao diện trước đó   B. Người dùng có thể hủy món chưa làm   * Xóa món ăn khỏi danh sách |
| **Non-Functional Requirements** | Không |

Modify Cart

|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | Modify Cart |
| **Description** | Người dùng có thể chỉnh sửa số lượng của món ăn trong giỏ hàng, thêm hoặc xóa món ăn |
| **Actor** | Khách hàng |
| **Trigger Condition** | Sau khi người dùng chọn thực đơn và click vào danh mục giỏ hàng |
| **Pre-Conditions** | User đã thực hiện check-in thông qua việc quét mã QR trên bàn và truy cập vào trang thực đơn |
| **Post-Conditions** | Hệ thống ghi nhận giỏ hàng sau khi thay đổi |
| **Main-Event Stream** | 1. Người dùng click vào button giỏ hàng 2. Thêm hoặc xóa món ăn. 3. Tăng giảm số lượng bằng các button + hoặc - 4. Gửi cho hệ thống giỏ hàng sau khi thay đổi |
| **Sub-Event Stream** | A. Người dùng nhấn quay lại   * Hệ thống quay lại giao diện trước đó |
| **Non-Functional Requirements** | Giao diện cần phải dễ sử dụng, thân thiện với người dùng khi chọn món |

Give Note

|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | Give note |
| **Description** | Ghi chú về đơn hàng |
| **Actor** | Khách hàng |
| **Trigger Condition** | Khách hàng viết ghi chú và xác nhận hoàn thành |
| **Pre-Conditions** | User đã thực hiện check-in thông qua việc quét mã QR trên bàn và trang chỉnh sửa giỏ hàng |
| **Post-Conditions** | Hệ thống ghi nhận ghi chú của khách hàng |
| **Main-Event Stream** | 1. Người dùng click vào button give note 2. Thêm đánh giá 3. Xác nhận đánh giá cho hệ thống |
| **Sub-Event Stream** | A. Người dùng nhấn quay lại   * Hệ thống quay lại giao diện trước đó |
| **Non-Functional Requirements** | Không |

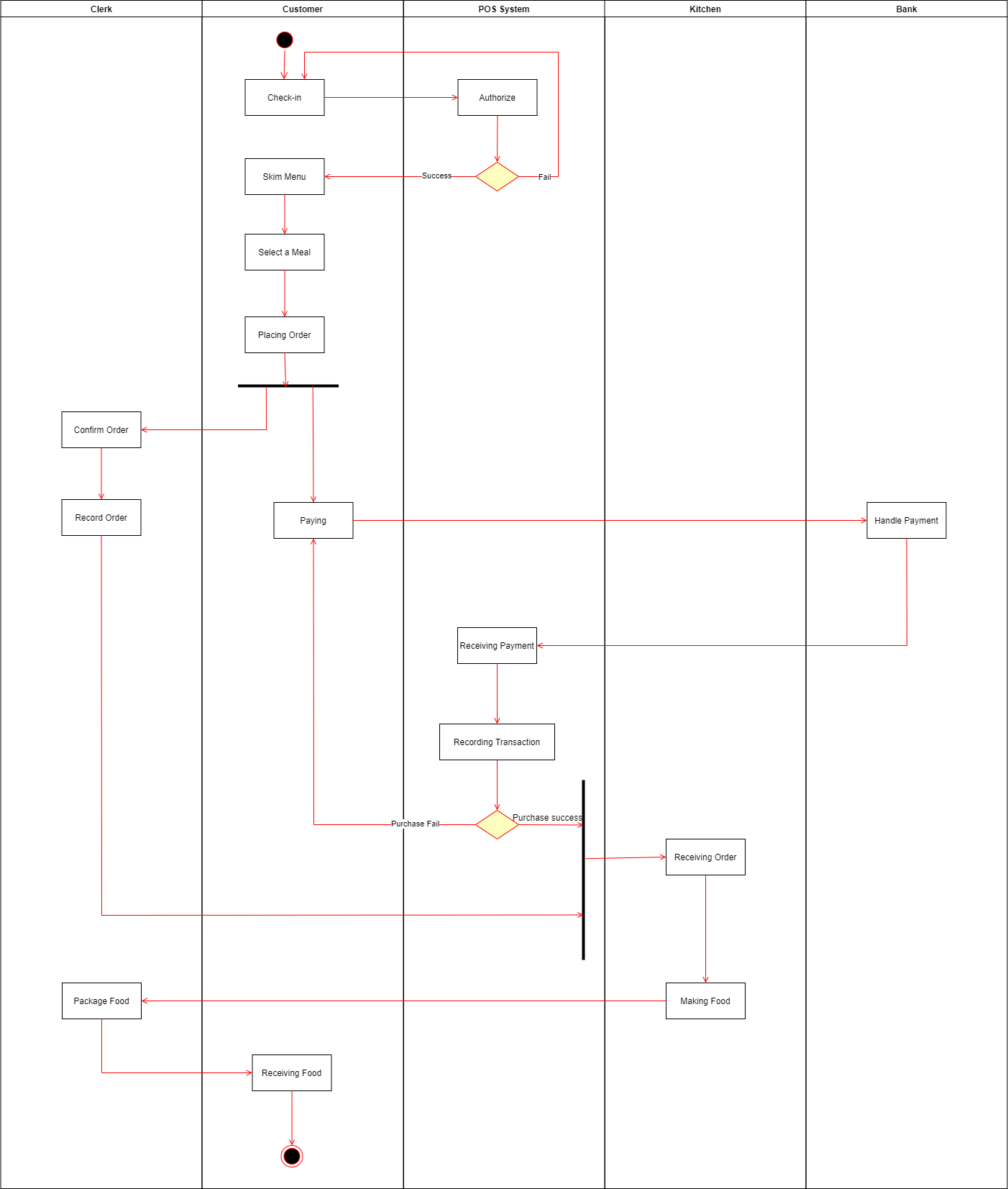
Cancel Order

|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | Cancel Order |
| **Description** | Người dùng có thể hủy đơn hàng nếu tính trạng đơn hàng phù hợp |
| **Actor** | Khách hàng |
| **Trigger Condition** | Người dùng bấm hủy đơn trong giao diện Tình Trạng Đơn Hàng |
| **Pre-Conditions** | Người dùng đã đặt món và đã thanh toán |
| **Post-Conditions** | Thực hiện quá trình hủy đơn hông báo cho người dùng kết quả |
| **Main-Event Stream** | * + - 1. Người dùng bấm hủy đơn       2. Hệ thống gửi thông báo đến nhân viên quản lý |
| **Sub-Event Stream** | A-Nhân viên quản lý duyệt hủy đơn   1. Hệ thống hủy đơn hàng và hoàn tiền 2. Thông báo cho người dùng   B-Nhân viên quản lý không duyệt hủy đơn   1. Hệ thống thông báo cho người dùng |
| **Non-Functional Requirements** | Không |

Conduct Purchase

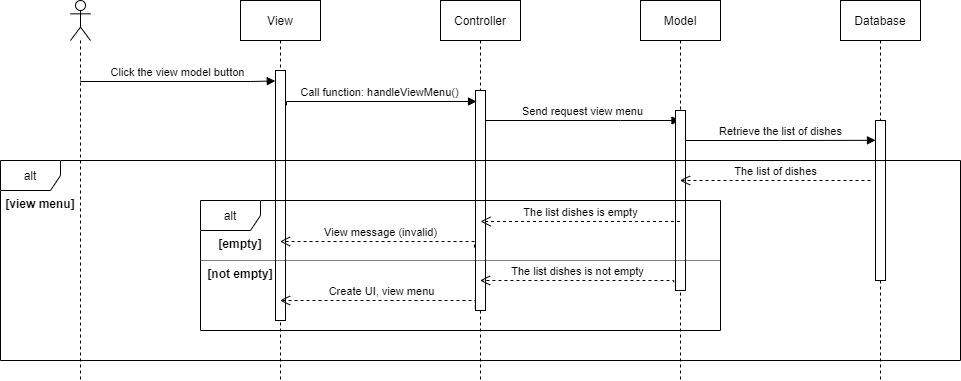
|  |  |
| --- | --- |
| **Use-Case** | **Content** |
| **Use-Case Name** | Conduct Purchase |
| **Description** | Tiến hành mua hàng và thanh toán |
| **Actor** | Khách hàng |
| **Trigger Condition** | Khách hàng xác nhận thanh toán |
| **Pre-Conditions** | User đã thực hiện check-in thông qua việc quét mã QR trên bàn và truy cập vào trang thực đơn và thực hiện chỉnh sửa giỏ hàng |
| **Post-Conditions** | Hệ thống xác nhận thanh toán thành công. Gửi đơn hàng cho nhà bếp. |
| **Main-Event Stream** | 1. Khách hàng chọn thanh toán 2. Trang web chuyển đến trang thanh toán 3. Tiến hành thanh toán bằng ngân hàng 4. Hệ thống xác nhận thanh toán thành công 5. Hệ thống gửi đơn hàng cho nhà bếp |
| **Sub-Event Stream** | 1. Thanh toán không thành công do tài khoản không đủ, hoặc hệ thống ngân hàng đang bảo trì 2. Nhà hàng không còn đủ nguyên liệu cho món 3. Người dùng có thể quay lại giao diện trước đó |
| **Non-Functional Requirements** | Giao diện thanh toán dễ sử dụng, và phải thực hiện nhanh chóng. |

1. Activity Diagram

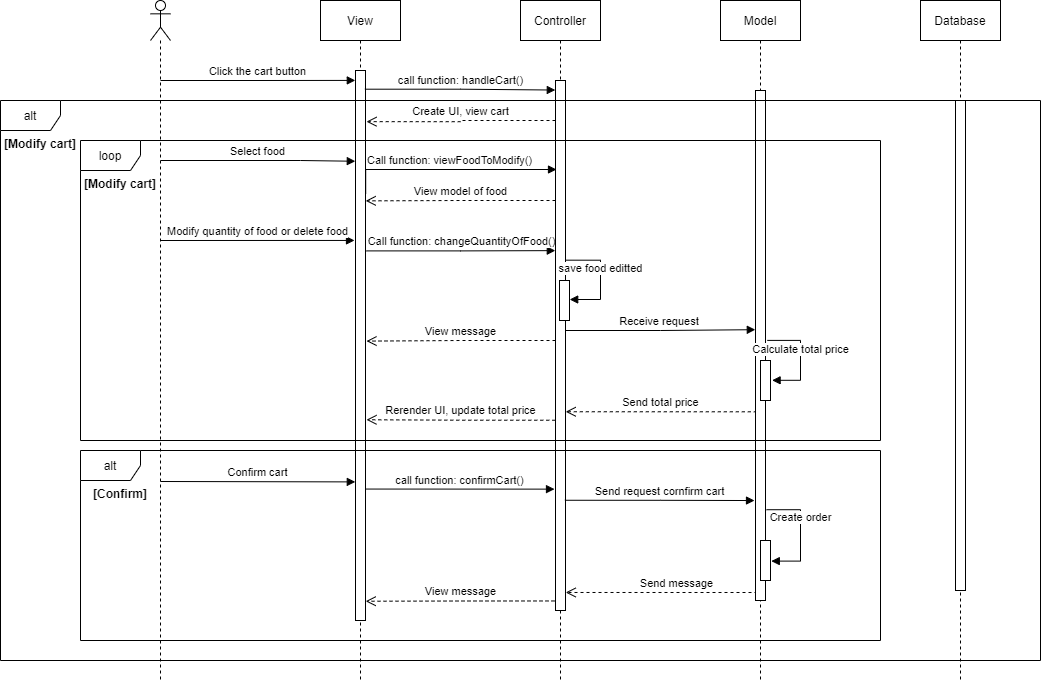


1. Sequence Diagram

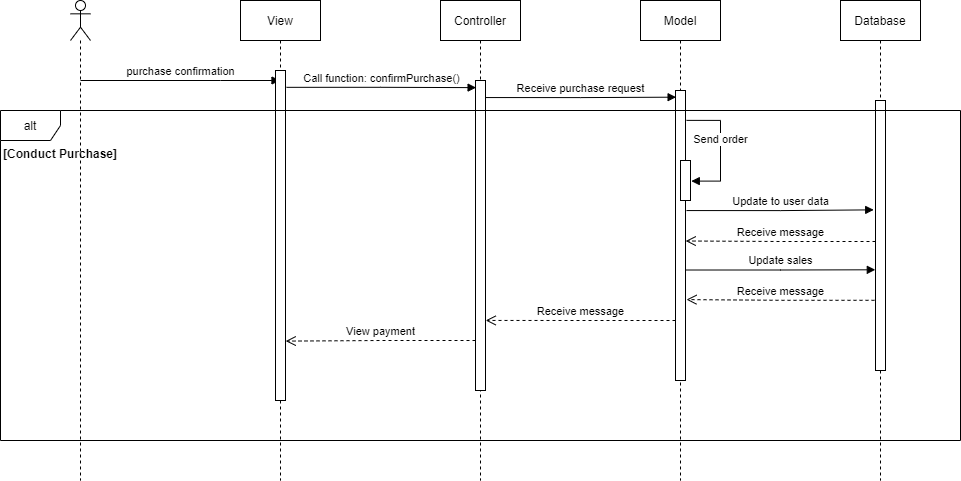
View Menu



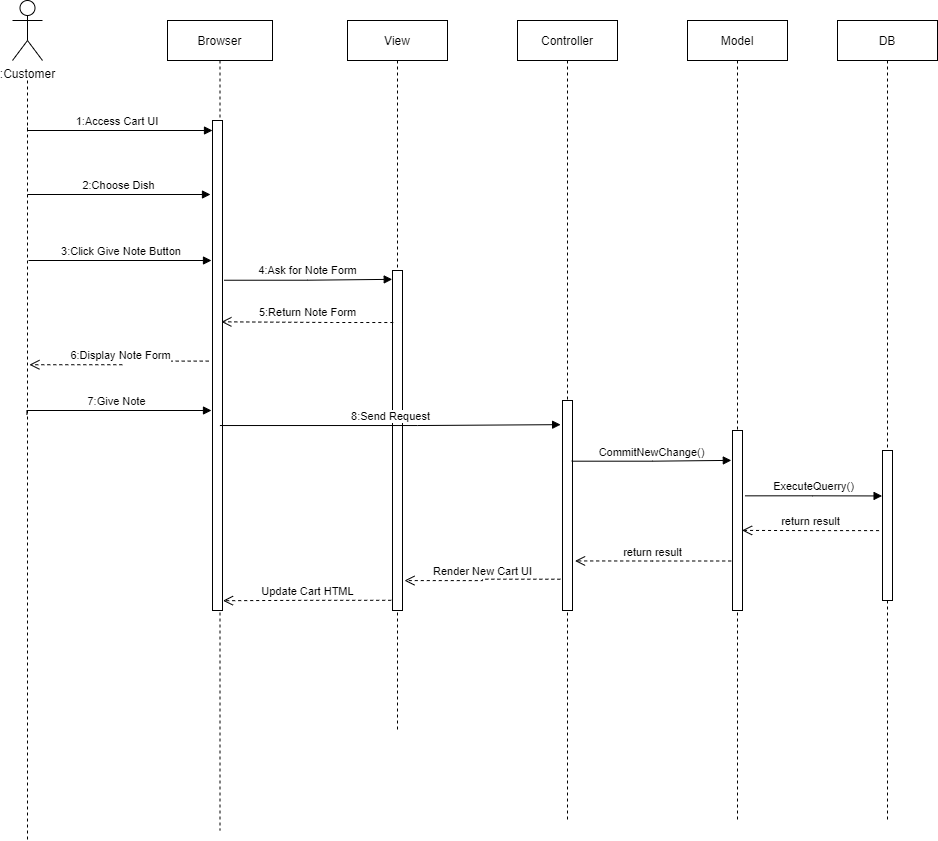
Modify Cart



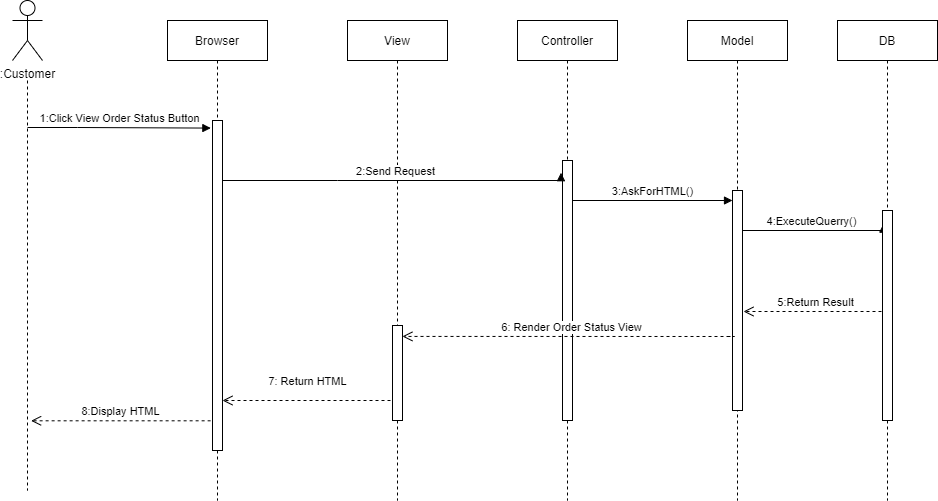
Conduct Purchase



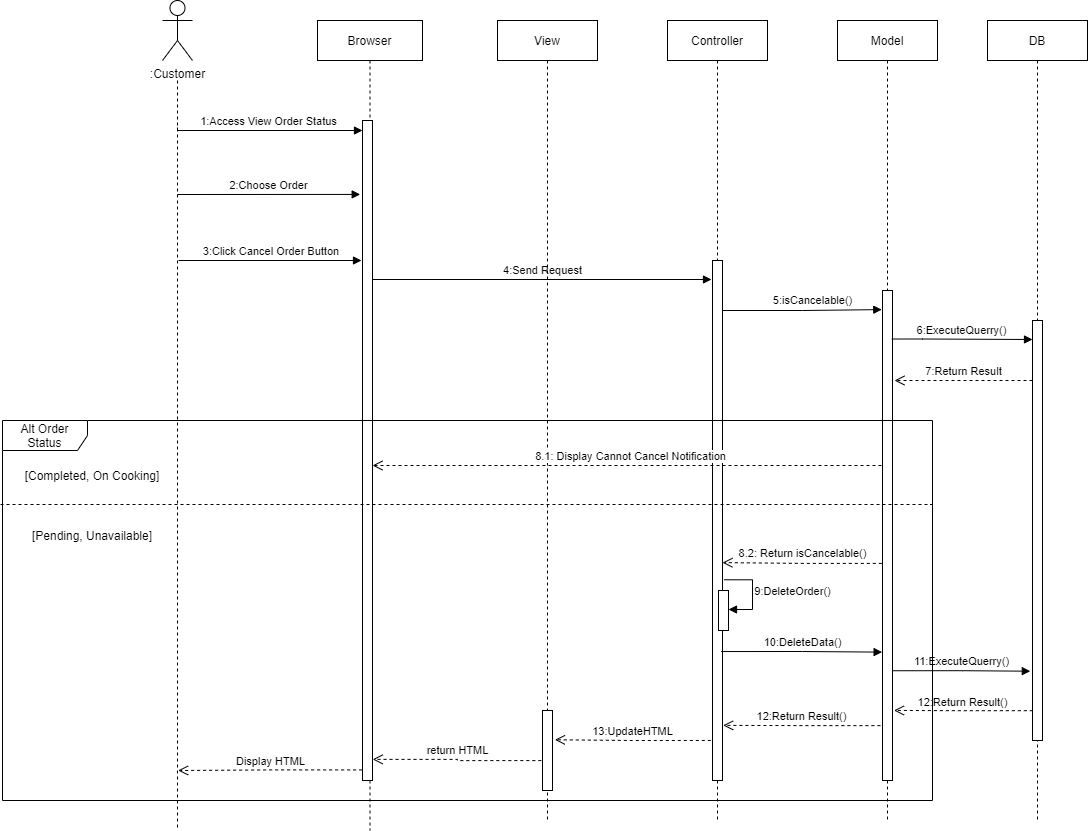
Give Note



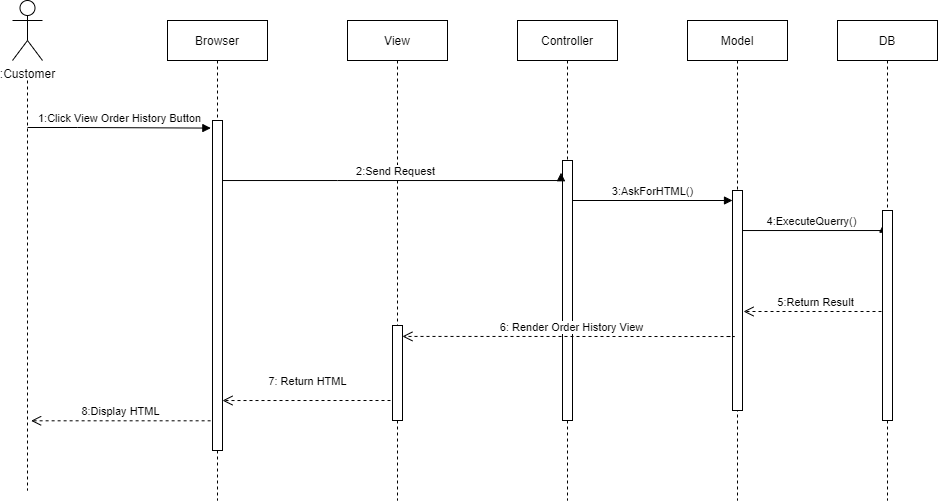
View Order Status



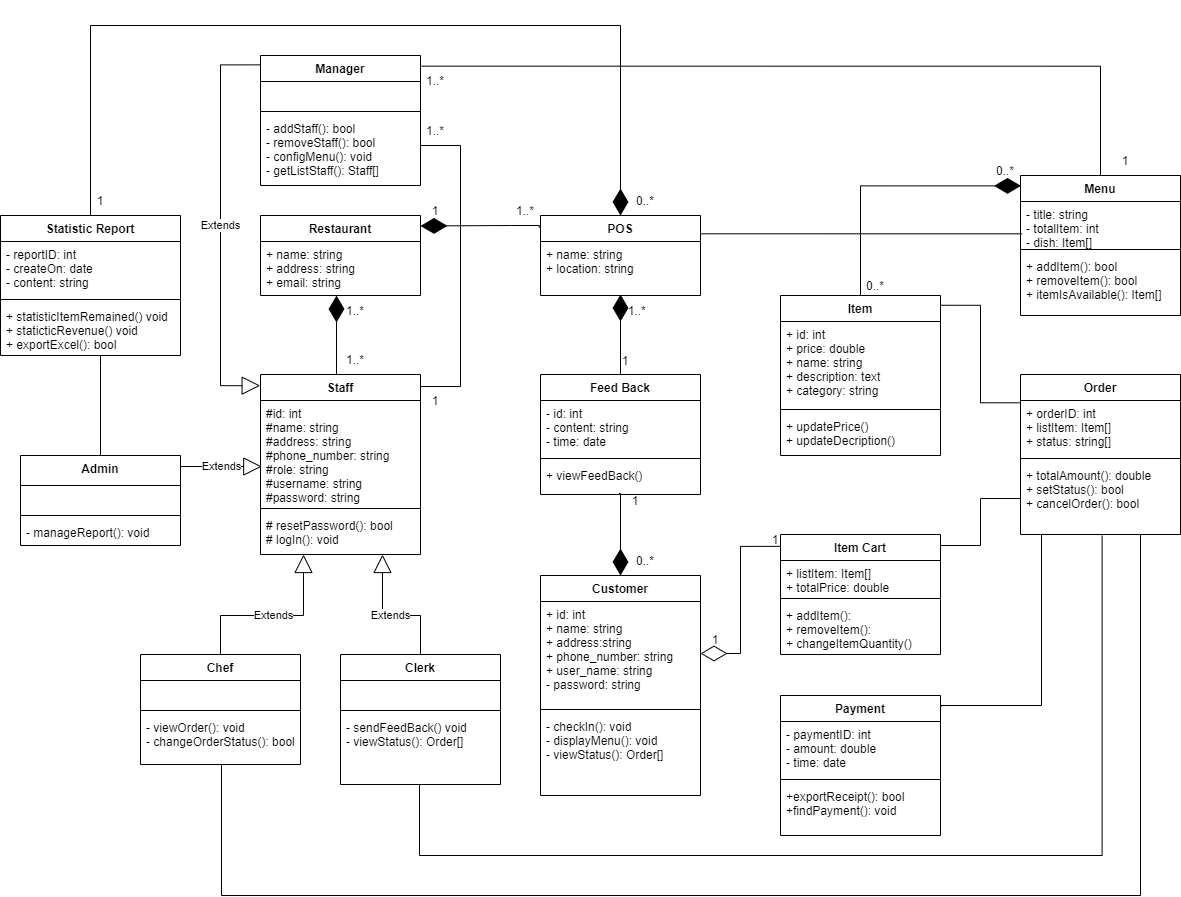
Cancel Order



View Order History



1. Class Diagram

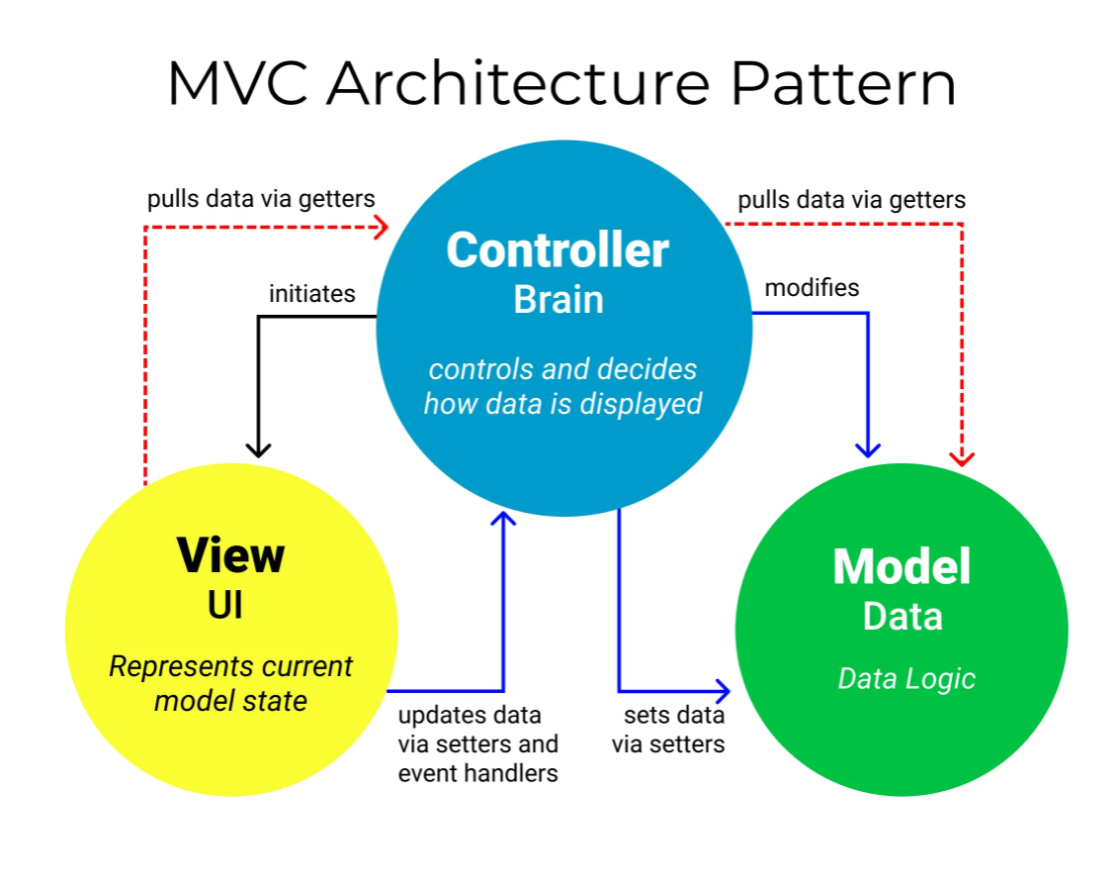


1. Mô tả Architectural Approach của nhóm

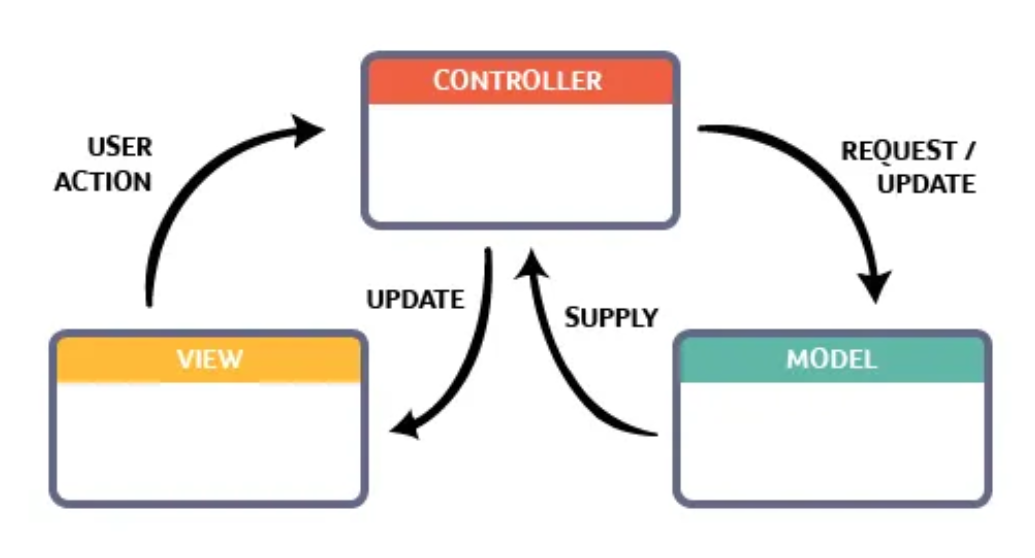
* Nhóm chọn Kiến trúc phát triển hệ thống theo Kiến trúc MVC.
* Dưới đây là các thông tin sơ lược về mô hình MVC nhóm đang theo đuổi.

1. Khái niệm

* MVC là viết tắt của cụm từ “Model-View-Controller“. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau. Chúng bao gồm Model (dữ liệu), View (giao diện) và Controller (bộ điều khiển).
* Đơn giản hơn, là mô hình này được chia thành 3 phần trong source code. Và mỗi phần đảm nhận vai trò và nhiệm vụ riêng biệt nhau và độc lập.
* Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.



1. Các thành phần trong mô hình MVC



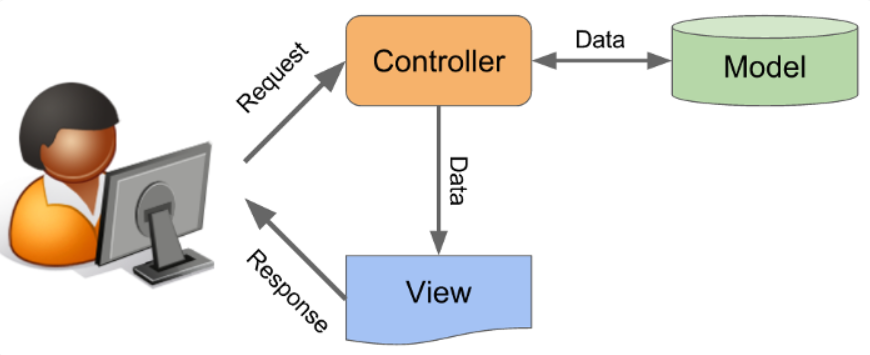
Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

* Model: là nơi chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu (mysql, mssql… ), nó sẽ bao gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu…
* View: là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
* Controller: là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/ function xử lý nhiều nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiển thị dữ liệu đó ra cho người dùng nhờ lớp View.

c) Luồng đi dữ liệu

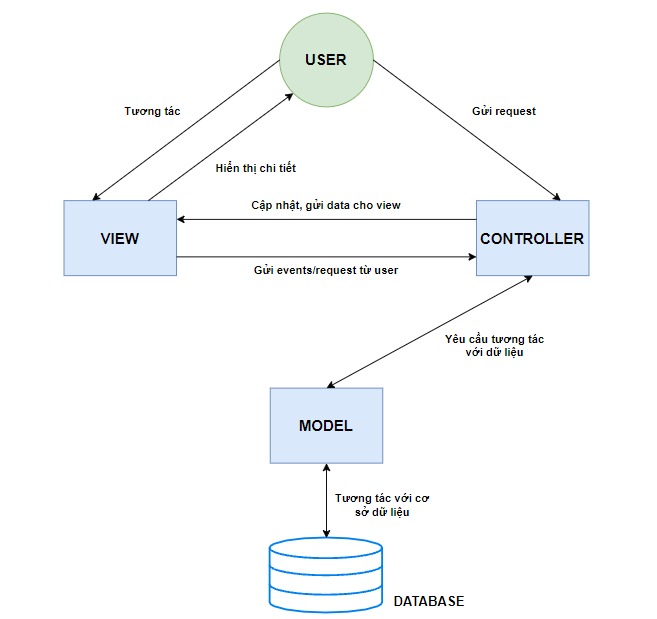
Về luồng xử lý trong của mô hình MVC có thể được hình dung cụ thể và chi tiết qua từng bước dưới đây:

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.



Ở đây, View không giao tiếp trực tiếp với Model. Sự tương tác giữa View và Model sẽ chỉ được xử lý bởi Controller.

Kiến trúc dự kiến sử dụng của nhóm:



d) Ưu điểm và nhược điểm của mô hình MVC

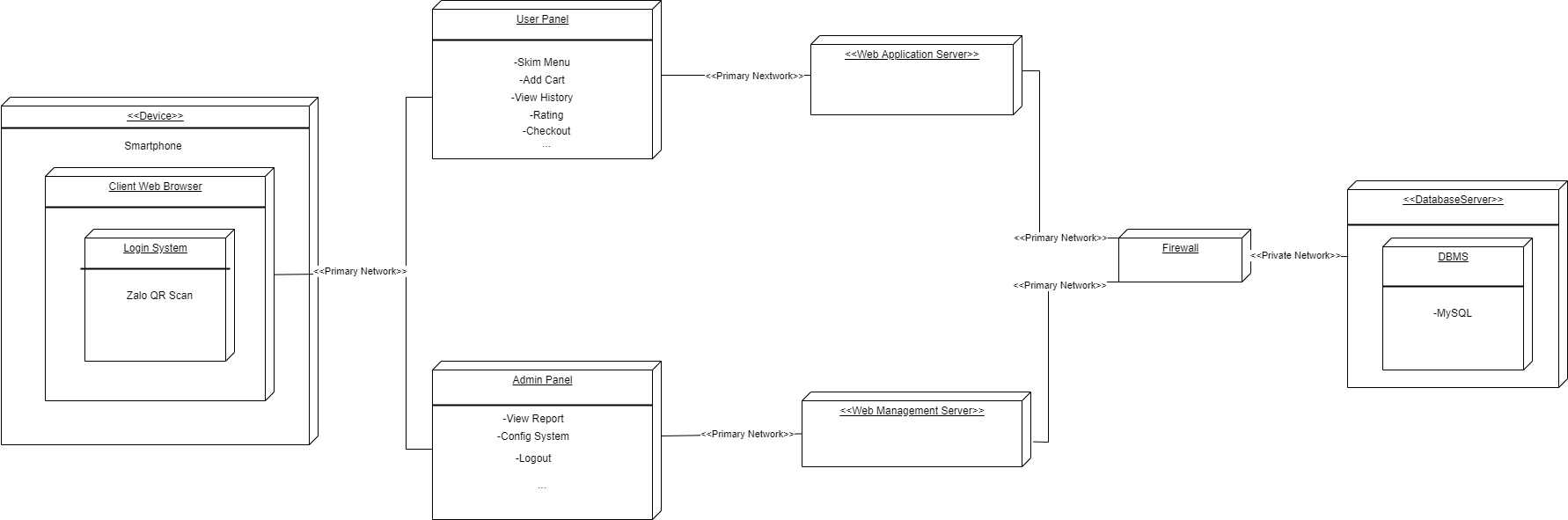
Ưu điểm mô hình MVC

* Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông (Bandwidth) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.
* Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
* Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau.
* Sử dụng mô hình MVC chức năng controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
* Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
* Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
* Hỗ trợ TDD (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các woon test case.
* Phiên bản mới nhất của MVC hỗ trợ trợ responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.

Nhược điểm mô hình MVC

* MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.
* Không thể Preview các trang như ASP.NET
* Khó triển khai.

1. Deployment Diagram



**Mô tả deployment diagram**

* Để truy cập vào hệ thống, người dùng cần có phần mềm quét QR code (Zalo,...) và sử dụng smartphone (mọi hệ điều hành). Sau khi quét QR code, người dùng bật browser và đưa đến trang đăng nhập/check-in  của hệ thống. Tại đây, người dùng cần có tài khoản Google Account hoặc tài khoản admin hệ thống để tiếp tục.
* Thông qua kết nối mạng, browser yêu cầu trang giao diện từ Web Application Server và Web Management Server. Thành phần View của hệ thống được chia thành 2 loại: User Panel (dành cho nhân viên hệ thống, khách hàng,...) và Admin Panel (dành cho riêng admin và nhân viên quản lý)
* Firewall là hệ thống an ninh nhằm đảm bảo quyền truy xuất vào Database là hợp lệ. Database Server quản lý Database qua DBMS là MySQL.

TỔNG HỢP ĐÓNG GÓP CỦA CÁC THÀNH VIÊN

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task | | Hải | Khôi | Tú | Khoa | Hậu |
| 1 | 1.1 | 1 | 0.5 | 1 | 0.5 | 1 |
| 1.2 | 1 | 0.5 | 1 | 0.5 | 1 |
| 1.3 | 1 | 0.5 | 1 | 0.5 | 1 |
| Tổng task 1 | | 3 | 1.5 | 3 | 1.5 | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task | | Hải | Khôi | Tú | Khoa | Hậu |
| 2 | 2.1 | 0.5 | 0.25 | 0.25 | 0.25 | 1 |
| 2.2 | 1 | 0.25 | 0.25 | 0.25 | 1 |
| 2.3 | 1 | 0.25 | 0.25 | 0.25 | 0.5 |
| Tổng task 2 | | 2.5 | 0.75 | 0.75 | 0.75 | 2.5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task | | Hải | Khôi | Tú | Khoa | Hậu |
| 3 | 3.1 | 1 | 0 | 0 | 0 | 0 |
| 3.2 | 0 | 0 | 0 | 0 | 1 |
| Tổng task 3 | | 1 | 0 | 0 | 0 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Hải | Khôi | Tú | Khoa | Hậu |
| 1 | 3 | 1.5 | 3 | 1.5 | 3 |
| 2 | 2.5 | 0.75 | 0.75 | 0.75 | 2.5 |
| 3 | 1 | 0 | 0 | 0 | 1 |
| Tổng | 6.5 | 2.25 | 3.75 | 2.25 | 6.5 |
| Phần trăm | 100% | 34.62% | 57.69% | 34.62% | 100% |

-End-