**ĐẠI HỌC DUY TÂN**

**TRƯỜNG KHMT**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--------------🙖🙐✰🙖🙐-------------**

****

**Tên đề tài:**

**XÂY DỰNG WEBSITE SMARTORDER – ĐẶT BÀN,**

**GỌI MÓN QUA QR CODE VÀ AI GỢI Ý MÓN ĂN**

**(PRODUCT BACKLOG DOCUMENT)**

**GVHD:** Th. S Hồ Lê Viết Nin

**Thành Viên**

Nguyễn Lương Thanh - 27211245051

Đoàn Ngọc Phương Linh - 27202438733

Hà Thị Thu Trang - 27201201147

Phan Trần Thiện Ân - 27211245105

Phan Thế Vương - 27211226951

**Đà Nẵng, Tháng 03 năm 2025**

|  |  |  |  |
| --- | --- | --- | --- |
| **THÔNG TIN DỰ ÁN** | | | |
| **Dự án viết tắt** | **Smartorder** | | |
| **Tên dự án** | Xây Dựng Website SmartOrder – Đặt Bàn, Gọi Món Qua QR Code Và AI Gợi Ý Món Ăn. | | |
| **Thời gian bắt đầu** | 17/03/2025 | **Thời gian kết thúc** | 17/05/2025 |
| **Lead Institution** | Khoa Công Nghệ Thông Tin, Trường Đại Học Duy Tân | | |
| **Giáo viên hướng dẫn** | ThS. Hồ Lê Viết Nin  Email: [holvietnin@dtu.edu.vn](mailto:holvietnin@dtu.edu.vn)  Phone: 0905455246 | | |
| **Chủ sở hữu dự án & Chi tiết liên hệ** | ThS. Hồ Lê Viết Nin  Email: [holvietnin@dtu.edu.vn](mailto:holvietnin@dtu.edu.vn)  Phone: 0905455246 | | |
| **Đối tác** | Duy Tan University | | |
| **Quản lý dự án & Scrum Master** | Nguyễn Lương  Thanh | [nguyenluongthanh201@gmail.com](mailto:%20nguyenluongthanh201@gmail.com) | 0795142551 |
| **Thành viên nhóm** | Đoàn Ngọc Phương Linh | [linhd4251@gmail.com](mailto:manhnguyen2002.it@gmail.com) | 0932437958 |
| Hà Thị Thu Trang | [thutrangv28@gmail.com](mailto:lephuocloc2002.vn@gmail.com) | 0972146317 |
| Phan Trần Thiện Ân | [phanttranthienan@gmail.com](mailto:%20phanttranthienan@gmail.com) | 0795415443 |
| Phan Thế Vương | [thevuong2k3@gmail.com](mailto:phuong119844@gmail.com) | 0336560061 |

|  |  |  |  |
| --- | --- | --- | --- |
| **TÊN TÀI LIỆU** | | | |
| **Tiêu đề tài liệu** | Product Backlog | | |
| **Tác giả** | Hà Thị Thu Trang | | |
| **Chức năng** | Thành viên | | |
| **Ngày** | 25/03/2025 | **Tên tệp:** | 03\_[KLTN- 25] User Story.docx |
| **URL** |  | | |
| **Access** | Khoa CNTT | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **LỊCH SỬ BẢN SỬA** | | | |
| **Version** | **Person** | **Date** | **Description** |
| 1.0 | Hà Thị Thu Trang | 25/03/2025 | Tạo tài liệu |
| 1.1 | Hà Thị Thu Trang | 26/03/2025 | Chỉnh sửa |

|  |  |  |  |
| --- | --- | --- | --- |
| **PHÊ DUYỆT TÀI LIỆU**  Cần có các chữ ký sau để phê duyệt tài liệu này | | | |
| **Người hướng dẫn** | | Hồ Lê Viết Nin | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |
| **Chủ sở hữu** | | Hồ Lê Viết Nin | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |
| **Scrum Master** | | Nguyễn Lương Thanh | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |
| **Thành viên** | | Đoàn Ngọc Phương Linh | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |
| Hà Thị Thu Trang | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |
| Phan Trần Thiện Ân | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |
| Phan Thế Vương | **Chữ ký:** |  |
| **Ngày:** | 25/03/2025 |

MỤC LỤC

[1. GIỚI THIỆU 1](#_Toc193642362)

[1.1. Mục đích 1](#_Toc193642363)

[1.2. Phạm vi 1](#_Toc193642364)

[1.3. Tham khảo 1](#_Toc193642365)

[1.4. biểu đồ usecase tổng quát 2](#_Toc193642366)

[2. Product Backlog 3](#_Toc193642367)

[2.1. Danh sách các Product Backlog trong Hệ thống 3](#_Toc193642370)

[2.1.1 PB01-Tìm Kiếm 5](#_Toc193642371)

[2.1.2 PB02-Lọc 5](#_Toc193642378)

[2.1.3 PB03-Quét mã QR gọi món 6](#_Toc193642381)

[2.1.4 PB04-Đặt bàn 7](#_Toc193642382)

[2.1.5 PB05-Đăng nhập 8](#_Toc193642385)

[2.1.6 PB06-Đăng xuất 9](#_Toc193642395)

[2.1.7 PB07-Quản lý Menu 10](#_Toc193642396)

[2.1.8 PB08-Quản lý bàn 11](#_Toc193642403)

[2.1.9 PB09-Quản lý nhân viên 12](#_Toc193642406)

[2.1.10 PB10-Quản lý danh mục 13](#_Toc193642408)

[2.1.11 PB11-Thống kê báo cáo 15](#_Toc193642410)

[2.1.12 PB12- Cấu Hình AI Gợi Ý Món Ăn 16](#_Toc193642411)

[2.1.13 PB13-Quản lý đơn hàng 19](#_Toc193642414)

[2.1.14 PB14-Quản lý thanh toán 20](#_Toc193642424)

[2.1.15 PB15-Đánh giá món ăn 22](#_Toc193642426)

[2.2. Phân tích về Sprint Backlog 22](#_Toc193642432)

[3. Hạn chế 23](#_Toc193642433)

[4. Tóm lược 23](#_Toc193642434)

# 

# GIỚI THIỆU

## Mục đích

Mục đích của tài liệu này:

* Xác định yêu cầu, ý tưởng và các vấn đề liên quan đến việc xây dựng dự án.
* Đưa ra các đề xuất dự án, kế hoạch hành động dự án, kiến trúc, giải pháp thực hiện, bao gồm cả về quy hoạch, phát triển, thực hiện và giám sát các dự án.
* Ngoài ra, xác định nhu cầu kinh doanh, các vấn đề hoặc tình huống liên quan đến các dự án khởi tạo, xây dựng và các rủi ro khi triển khai dự án.

## Phạm vi

Bên cạnh việc cung cấp cho bạn các tài liệu, nó sẽ liệt kê tất cả các thành viên trong nhóm phát triển trong dự án này, quy trình và khuôn khổ mà dự án sẽ áp dụng.

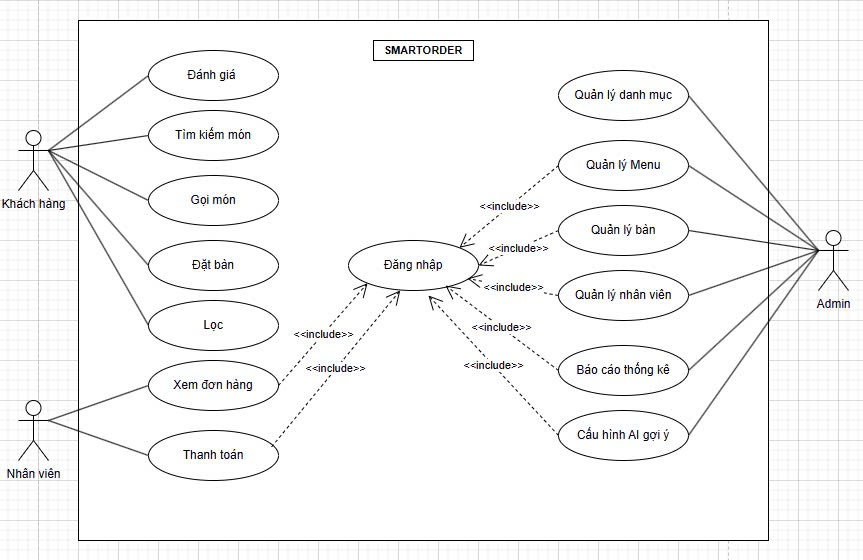
Tài liệu này đưa ra kế hoạch cho từng giai đoạn của quy trình phát triển phần mềm dự trên quy trình Scrum bao gồm: thời gian bắt đầu, thời gian kết thúc và một số ngày làm việc. Đây là kế hoạch chung và sẽ được cập nhật chi tiết về quy trình phát triển phần mềm trong phiên bản tiếp theo của tài liệu. Đề xuất bao gồm giới thiệu các giải pháp, xác định cách tốt nhất để phát triển mà chúng tôi tạo ra, tổng chi phí ước tính, thời gian hoàn vốn và khối lượng hòa vốn cho dự án.

## Tham khảo

*Table 1.1: Danh mục tài liệu tham khảo*

|  |  |  |
| --- | --- | --- |
| No | Document Information | References |
| 1 | Scrum Process | <https://www.mountaingoatsoftware.com/agile/scrum> |
| 2 | HTML, CSS, JS | https://www.w3schools.com/ |
| 3 | React | https://www.react.dev/ |
| 7 | Information | Requirement Document |

## Biểu đồ UseCase Tổng quát



*Hình 1: Biểu đồ usecase tổng quát*

# PRODUCT BACKLOG

2. 1. Danh sách các Product Backlog trong Hệ thống

*Bảng 1: Đặc tả tổng quát các Product Backlog*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Chủ đề** | **Với** | **Mong muốn** | **Nên** | **Độ ưu tiên** |
| PB01 | Tìm Kiếm | Khách hàng | Tìm kiếm món ăn theo tên | Khách hàng có thể Tìm kiếm món ăn theo tên | 1 |
| PB02 | Lọc | Khách Hàng | Lọc ra những món ăn theo yêu cầu | Người dùng có thể Lọc ra giá theo yêu cầu | 1 |
| PB03 | Quét mã QR gọi món | Khách Hàng | Quét QR để gọi món | Khách Hàng có thể Quét QR để gọi món | 3 |
| PB04 | Đặt bàn | Khách Hàng | Đặt bàn trước | Người dùng có thể Đặt bàn trước để tiết kiệm thời gian | 3 |
| PB05 | Đăng nhập | Admin/Nhân Viên | Truy cập vào hệ thống | Người dùng có thể đăng nhập vào hệ thống | 1 |
| PB06 | Đăng Xuất | Nhân viên/ Admin | Đăng xuất khỏi hệ thống | Người dung có thể đăng xuất khỏi hệ thống | 1 |
| PB07 | Quản Lý Menu | Admin | Quản lý các thông tin món ăn | Admin có thể quản lý các thông tin món ăn | 2 |
| PB08 | Quản Lý Bàn | Admin | Quản lý các thống tin của bàn | Admin có thể Quản lý các thông ti của bàn như thời gian trống của bàn, bàn đã được đặt,.. | 2 |
| PB09 | Quản Lý Nhân Viên | Admin | Quản lý thông tin nhân viên | Admin có thể quản lý các thông tin của nhân viên | 1 |
| PB10 | Quản Lý Danh Mục | Admin | Quản lý danh mục thức ăn | Admin có thể Quản lý các thông tin danh mục thức ăn | 1 |
| PB11 | Thống Kê Báo Cáo | Admin | Thống kê báo cáo doanh thu | Admin có thể tạo các thống kê báo cáo doanh thu,.. | 2 |
| PB12 | Cấu hình AI gọi ý món | Admin | Cấu hình AI để gợi ý món ăn cho khác hàng | Admin Cấu hình AI để gợi ý món ăn cho khác hàng | 2 |
| PB13 | Quản Lý Đơn Hàng | Nhân viên | Thực hiện tiếp nhận hoặc huỷ đơn hàng | Nhân viên có thể tiếp nhận hoặc huỷ đơn hàng | 1 |
| PB14 | Quản Lý Thanh Toán | Nhân viên | Thực hiện thanh toán cho các sản phẩm theo nhu cầu của khách hàng | Nhân viên có thể Thực hiện thanh toán cho các sản phẩm theo nhu cầu của khách hàng | 2 |
| PB15 | Đánh giá món ăn | Khách hàng | Khách hàng có thể đánh giá món ăn | Khách hàng có thể đưa ra nhận xét, đánh giá về chất lượng của món ăn dựa trên nhiều tiêu chí khác nhau. | 1 |

***\*Giải thích: Độ ưu tiên càng tăng thì càng quan trọng***

### 2.1.1 PB01-Tìm Kiếm

|  |  |
| --- | --- |
| **ID** | **PB01** |
| **Tiêu đề** | Tìm kiếm sản phẩm |
| **Mô tả** | Người dùng có thể tìm kiếm sản phẩm trên website. |
| **Các bước thực hiện** | 1. Thiết kế giao diện ô tìm kiếm trên trang chủ  2. Xử lý logic kiểm tra từ khoá và gọi API tìm kiếm sản phẩm  3. Lọc kết quả tìm kiếm theo từ khoá  4. Hiển thị kết quả tìm kiếm hoặc thông báo nếu không có sản phẩm phù hợp |
| **Điều kiện trước** | 1.Người dùng vào trang chủ và nhập từ khóa tìm kiếm vào ô “Tìm kiếm”.2.Người dùng nhấn nút "Tìm kiếm" để gửi yêu cầu tìm sản phẩm.3.Hệ thống kiểm tra từ khóa:Nếu từ khóa có trong tên sản phẩm, hệ thống sẽ hiển thị sản phẩm phù hợp.Nếu không có sản phẩm nào khớp, hệ thống hiển thị thông báo: "Không có sản phẩm nào với từ khóa trên". |
| **Điều kiên sau** | Trang chủ đã được tải thành công và người dùng có thể thấy ô tìm kiếm |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện tìm kiếm, gọi API)  -BackEnd:Laravel(xử lý tìm kiếm, truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Thấp |

### 2.1.2 PB02-Lọc

|  |  |
| --- | --- |
| **ID** | **PB02** |
| **Tiêu đề** | Lọc theo giá sản phẩm |
| **Mô tả** | Người dùng có thể lọc theo giá của sản phẩm trên hệ thống. |
| **Các bước thực hiện** | 1. Thiết kế giao diện lọc theo giá  2. Xử lý logic kiểm tra theo giá và gọi API lọc giá  3. Lọc giá theo giá đã chọn  4. Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1. Truy cập vào trang web 2. Nhập số tiền muốn lọc  3. Nhấn nút “Lọc theo giá”  4. Nếu thành công, hiển thị danh sách đã lọc theo tiêu chí.  Nếu thất bại, hiển thị thông báo lỗi với lý do cụ thể. |
| **Điều kiên sau** | Danh sách đã lọc được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện Lọc theo giá, gọi API)  -BackEnd:Laravel(xử lý Lọc , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Thấp |

### 2.1.3 PB03-Quét mã QR gọi món

|  |  |
| --- | --- |
| **ID** | **PB03** |
| **Tiêu đề** | Quét QR để gọi món |
| **Mô tả** | Người dùng có quét mã QR để gọi món . |
| **Các bước thực hiện** | 1. Thiết kế giao diện quét mã QR trên trang ứng dụng.  2.Kết nối với API để lấy thông tin món ăn từ mã QR.  3.Hiển thị thông tin món ăn cho người dùng. |
| **Điều kiện trước** | 1. Khách hàng đến nhà hàng và ngồi vào bàn.  2. Quét mã QR (được đặt sẵn trên bàn) bằng điện thoại.  3. Hệ thống hiển thị thực đơn điện tử, bao gồm:   * Danh sách món ăn, đồ uống kèm hình ảnh, mô tả và giá. * Các tùy chọn như thêm topping, size, ghi chú đặc biệt.   4. Khách hàng chọn món và xác nhận đơn hàng.  5. Hệ thống gửi đơn hàng đến bếp và nhân viên phục vụ.  6. Nhà bếp chế biến món ăn, nhân viên phục vụ mang món đến bàn. |
| **Điều kiện sau** | Người dùng có thể xác nhận đơn hàng sau khi xem thông tin món ăn. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện gọi món, gọi API) -BackEnd:Laravel(xử lý quét mã , truy vấn cơ sở dữ liệu) -Database:MySql(Lưu trữ thông tin) -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

### 2.1.4 PB04-Đặt bàn

|  |  |
| --- | --- |
| **ID** | **PB04** |
| **Tiêu đề** | Đặt bàn |
| **Mô tả** | Người dùng có thể đặt bàn trên website. |
| **Các bước thực hiện** | 1. Thiết kế giao diện đặt bàn trên trang chủ 2. Triển khai API để nhận yêu cầu đặt bàn:  Xử lý xác thực thông tin và lưu vào cơ sở dữ liệu. 3.Gửi phản hồi về trạng thái đặt bàn cho người dùng:  Nếu thành công, hiển thị thông báo xác nhận rằng đặt bàn đã thành công.  Nếu thất bại, hiển thị thông báo lỗi với lý do cụ thể (ví dụ: thời gian không khả dụng). |
| **Điều kiện trước** | 1. Người dung truy cập vào trang web chọn:“Đặt bàn”  2. Chọn ngày, giờ và số lượng khách  3. Chọn bàn  4. Nhập thông tin liên hệ  5. Xác nhận đặt bàn  6. Hệ thống kiểm tra tình trạng bàn trống:   * Nếu còn trống -> Xác nhận đặt bàn * Ngược lại -> Đề xuất chọn giờ khác hoặc bàn khác |
| **Điều kiện sau** | Người dùng nhận được thông báo xác nhận sau khi đặt bàn thành công. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện đặt bàn, gọi API) -BackEnd:Laravel(xử lý đặt bàn , truy vấn cơ sở dữ liệu) -Database:MySql(Lưu trữ thông tin) -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

### 2.1.5 PB05-Đăng nhập

|  |  |
| --- | --- |
| **ID** | **PB05** |
| **Tiêu đề** | Đăng nhập |
| **Mô tả** | Người dùng đăng nhập vào hệ thống |
| **Các bước thực hiện** | 1.Thiết kế giao diện đăng nhập 2. Nhận thông tin đăng nhập3.Tìm kiếm người dung theo tên đăng nhập4. So sánh mật khẩu nhập với mật khẩu mã hoá trong cơ sở dữ liệu. 5. Hiển thị đăng nhập thành công hoặc thông báo lỗi. |
| **Điều kiện trước** | Người dùng truy cập vào trang webNhập các thông tin để đăng nhậpNgười dùng nhấn nút "Đăng nhập".Hệ thống kiểm tra định dạng dữ liệu:Nếu dữ liệu không hợp lệ, hiển thị thông báo: "Dữ liệu nhập không hợp lệ, vui lòng kiểm tra lại."Nếu dữ liệu hợp lệ, hiển thị thông báo: "Đăng nhập thành công!". |
| **Điều kiện sau** | Người dùng nhận được thông báo xác nhận sau khi đăng nhập thành công. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện đăng nhập)  -BackEnd:Laravel(xử lý đăng nhập , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### 2.1.6 PB06-Đăng xuất

|  |  |
| --- | --- |
| **ID** | **PB06** |
| **Tiêu đề** | Đăng xuất |
| **Mô tả** | Người dùng vào website và muốn đăng xuất khỏi hệ thống |
| **Các bước thực hiện** | 1. Thiết kế giao diện đăng xuất trên trang chủ  2. Nhận token từ header của yêu cầu để xác thực người dùng khi nhấn nút đăng xuất.  3. Nếu token hợp lệ, xóa hoặc làm không hợp lệ token để đảm bảo người dùng không còn truy cập vào hệ thống. |
| **Điều kiện trước** | 1.Người dùng đăng nhập vào hê thống trước đó  2.Chọn chức năng Đăng Xuất (Nhấp chọn [Đăng Xuất])  3.Người dùng chuyển về trang chủ |
| **Điều kiện sau** | Người dùng không còn quyền truy cập vào các chức năng yêu cầu xác thực sau khi đăng xuất thành công. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện đăng xuất)  -BackEnd:Laravel(xử lý đăng xuất , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Thấp |

### 2.1.7 PB07-Quản lý Menu

|  |  |
| --- | --- |
| **ID** | **PB07** |
| **Tiêu đề** | Quản lý Menu |
| **Mô tả** | Người dùng có thể thêm, chỉnh sửa hoặc xóa món ăn, cập nhật giá và trạng thái món |
| **Các bước thực hiện** | 1.Thiết kế trang quản lý menu2. Thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu.3. Nhận dữ liệu từ form khi thêm hoặc sửa món ăn.Kiểm tra tính hợp lệ của dữ liệu 4.Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1. Admin đăng nhập vào hệ thống 2. Truy cập trang "Quản lý Menu".  3. Hệ thống hiển thị danh sách menu hiện có, bao gồm:   * Tên món, mô tả, giá, hình ảnh và trạng thái (kích hoạt/tạm dừng).   4. Admin có thể thêm các món mới   * Nhấn nút “Thêm” * Điền các thông tin của món như Tên món, giá,.. * Nếu như món ăn chưa có trong menu khi nhấn nút “Thêm” sẽ thêm thành công và lưu vào cơ sở dữ liệu * Ngược lại, nếu đã tồn tại hệ thống sẽ báo “Thêm không thành công” và cập nhật lại cơ sở dữ liệu   5. Admin có thể xoá các món   * Chọn món cần xoá và nhấn nút “Xoá”.   6. Admin có thể cập nhật thông tin các món đã có   * Chọn món cần cập nhật * Điền các thông tin cần cập nhật |
| **Điều kiện sau** | Thông tin menu được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện quản lý menu, gọi API)  -BackEnd:Laravel(xử lý quản lý menu , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### 2.1.8 PB08-Quản lý bàn

|  |  |
| --- | --- |
| **ID** | **PB08** |
| **Tiêu đề** | Quản lý bàn |
| **Mô tả** | Người dùng có thể thêm, cập nhật bàn |
| **Các bước thực hiện** | 1. Thiết kế giao diện quản lý bàn  2. Thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu.  3. Nhận dữ liệu từ form khi thêm hoặc sửa bàn.  Kiểm tra tính hợp lệ của dữ liệu  4.Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1. Đăng nhập vào hệ thống 2. Truy cập vào chức năng quản lý bàn  3. Hệ thống hiển thị các bàn hiện có  4. Admin có thể:  **Thêm bàn mới**   * Chọn nút “Thêm” * Điền các thông tin của bàn * Nhấn nút xác nhận * Kiểm tra dữ liệu hợp lệ -> Thông báo “Thêm thành công” * Ngược lại -> Thông báo lỗi   **Cập nhật bàn**   * Chọn bàn cần cập nhật * Cập nhật trạng thái của bàn(Trống/ Đã đặt). |
| **Điều kiện sau** | Thông tin bàn được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện quản lý bàn, gọi API)  -BackEnd:Laravel(xử lý quản lý bàn , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### 2.1.9 PB09-Quản lý nhân viên

|  |  |
| --- | --- |
| **ID** | **PB09** |
| **Tiêu đề** | Quản lý nhân viên |
| **Mô tả** | Người dùng có thể thêm, sửa, xoá thông tin nhân viên |
| **Các bước thực hiện** | 1. Thiết kế giao diện quản lý nhân viên  2. Thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu.  3. Nhận dữ liệu từ form khi thêm hoặc sửa thông tin nhân viên.  Kiểm tra tính hợp lệ của dữ liệu  4.Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1. Đăng nhập vào trang web  2. Chọn chức năng quản lý nhân viên  3. Hệ thống sẽ hiển thị danh sách các nhân viên hiện có  4. Admin có thể:  Thêm nhân viên mới   * Hệ thống hiển thị biểu mẫu nhập thông tin nhân viên mới bao gồm: * Tên, Số điện thoại, Ngày làm việc,Trạng thái,... * Nhấn nút “Lưu” * Hệ thống xác nhận và lưu vào cơ sở dữ liệu.   Cập nhật   * Có thể cập nhật thông tin nhân viên và nhất nút “Cập nhật”. * Thông tin sẽ được lưu vào cơ sở dữ liệu   Xoá nhân viên   * Chọn nhân viên muốn xoá * Nhấn nút Xoá * Xác nhận xoá * Nhân viên có đơn hàng đang xử lý chỉ có thể bị vô hiệu hóa thay vì xóa. |
| **Điều kiện sau** | Thông tin nhân viên được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện quản lý nhân viên, gọi API)  -BackEnd:Laravel(xử lý quản lý nhân viên , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Đooj phức tạp** | Trung bình |

### 2.1.10 PB10-Quản lý danh mục

|  |  |
| --- | --- |
| **ID** | **PB10** |
| **Tiêu đề** | Quản lý danh mục |
| **Mô tả** | Người dùng có thể tổ chức và phân loại các món ăn trong nhà hàng |
| **Các bước thực hiện** | 1. Thiết kế giao diện quản lý danh mục.  2. Thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu.  3. Nhận dữ liệu từ form khi thêm hoặc sửa thông tin danh mục.  Kiểm tra tính hợp lệ của dữ liệu  4.Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1. Đăng nhập vào website  2. Truy cập vào chức năng quản lý danh mục  3. Hệ thống hiển thị các danh mục hiện có.  4. Admin có thể:  Thêm mới danh mục   * Chọn nút “Thêm” * Hiển thị biểu mẫu nhập thông tin danh mục mới * Nhập Mã danh mục, Tên, Mô tả, Trạng thái,.. * Nhấn nút Lưu * Kiểm tra dữ liệu hợp lệ -> Thông báo “Thêm thành công” * Ngược lại, hiển thị thông báo “Thêm thất bại” * Hệ thống xác nhận và lưu vào cơ sở dữ liệu   Cập nhật danh mục   * Chọn danh mục cần cập nhật * Nhập thông tin muốn cập nhật * Có thể cập nhật trạng thái của danh mục(Kích hoạt/Tạm dừng) * Nhấn nút Lưu * Hệ thống xác nhận và lưu vào cơ sở dữ liệu   Xoá danh mục   * Chọn danh mục cần xoá * Nhấn nút Xoá * Nếu danh mục có món ăn đang hoạt động, chỉ có thể ẩn thay vì xóa. * Hệ thống xác nhận và cập nhật lại cơ sở dữ liệu |
| **Điều kiện sau** | Thông tin danh mục được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện quản lý danh mục, gọi API)  -BackEnd:Laravel(xử lý quản lý danh mục , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### 2.1.11 PB11-Thống kê báo cáo

|  |  |
| --- | --- |
| **ID** | **PB11** |
| **Tiêu đề** | Thống kê báo cáo |
| **Mô tả** | Người dùng có thể nắm bắt thông tin quan trọng về hoạt động của nhà hàng, từ đó đưa ra quyết định hợp lý nhằm cải thiện hiệu suất kinh doanh. |
| **Các bước thực hiện** | 1. Thiết kế giao diện thống kê báo cáo  2. Xử lý logic kiểm tra yêu cầu và gọi API lấy báo cáo  3. Lọc kết quả theo thời gian, theo loại báo cáo,..  4.Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1. Đăng nhập vào website  2. Truy cập vào chức năng thống kê báo cáo.  3. Hệ thống hiển thị các tùy chọn báo cáo, bao gồm:   * Doanh thu theo ngày, tuần, tháng. * Món ăn bán chạy nhất. * Tình trạng bàn (trống/đã đặt). * Thông tin về nhân viên (hiệu suất làm việc).   4. Người quản lý chọn loại báo cáo cần xem.  Hệ thống hiển thị biểu mẫu để nhập thông tin lọc (nếu cần), bao gồm:   * Thời gian (ngày bắt đầu, ngày kết thúc). * Danh mục món ăn (nếu cần). * Nhân viên (nếu cần).   5. Người quản lý điền thông tin lọc và nhấn "Tạo báo cáo".  6. Hệ thống truy xuất dữ liệu từ cơ sở dữ liệu và tạo báo cáo.  7. Hệ thống hiển thị báo cáo dưới dạng bảng hoặc biểu đồ, bao gồm:  Tổng doanh thu, số lượng món ăn bán ra, và các thông tin liên quan. |
| **Điều kiện sau** | Thông tin nhân viên được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện thống kê báo cáo, gọi API)  -BackEnd:Laravel(xử lý thống kê , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

### 2.1.12 PB12- Cấu Hình AI Gợi Ý Món Ăn

|  |  |
| --- | --- |
| **ID** | **PB12** |
| **Tiêu đề** | Cấu Hình AI Gợi Ý Món Ăn |
| **Mô tả** | Chức năng cấu hình AI gợi ý món ăn giúp nhà hàng tối ưu hóa trải nghiệm của khách hàng bằng cách cung cấp các gợi ý phù hợp với sở thích và xu hướng tiêu dùng, từ đó tăng cường doanh thu và sự hài lòng của khách |
| **Các bước thực hiện** | 1.Thiết kế giao diện cấu hình AI gợi ý món  2. Xác định tiêu chí gợi ý món ăn  3. Xây dựng mô hình AI gợi ý món ăn  4. Triển khai API gợi ý món ăn  5. Tích hợp vào giao diện Web |
| **Điều kiện trước** | 1. Đăng nhập vào hệ thống  2. Truy cập vào chức năng cấu hình AI:   * Admin chọn tùy chọn "Cấu hình AI gợi ý món ăn" từ menu.   3. Kết nối với AI bên ngoài:   * Hệ thống kiểm tra kết nối với AI tích hợp từ bên ngoài. * Nếu kết nối không thành công, hiển thị thông báo lỗi và yêu cầu kiểm tra cài đặt. * Nếu kết nối thành công, chuyển sang bước tiếp theo.   4. Thiết lập tiêu chí gợi ý:  Hệ thống hiển thị giao diện cấu hình các tiêu chí gợi ý, bao gồm:   * Lịch sử đặt món. * Món bán chạy. * Thời gian trong ngày. * Danh mục ưa thích.   5. Admin chọn các tiêu chí mong muốn.  Điều chỉnh trọng số:   * Hệ thống hiển thị biểu mẫu để admin điều chỉnh trọng số cho từng tiêu chí. * Admin nhập mức độ ưu tiên cho các tiêu chí đã chọn. * Nhấn nút "Lưu trọng số".   6. Gửi thông tin cấu hình đến AI:   * Hệ thống gửi các tiêu chí và trọng số đã cấu hình đến AI bên ngoài thông qua API. * Nhận phản hồi từ AI, xác nhận rằng cấu hình đã được cập nhật.   7. Xem trước kết quả gợi ý:   * Hệ thống yêu cầu AI tạo danh sách món ăn gợi ý dựa trên cấu hình đã thiết lập. * AI xử lý và gửi lại danh sách món ăn. * Hệ thống hiển thị danh sách món ăn mà AI sẽ gợi ý cho khách hàng.   8.Tích hợp dữ liệu thực tế:   * Hệ thống tự động cập nhật thông tin từ đơn hàng vào AI để tối ưu hóa gợi ý. * AI nhận dữ liệu và điều chỉnh gợi ý dựa trên thông tin mới nhất.   9.Bật/tắt AI gợi ý:   * Admin chọn tùy chọn bật hoặc tắt tính năng gợi ý món ăn. * Hệ thống gửi yêu cầu đến AI để cập nhật trạng thái gợi ý. |
| **Điều kiện sau** | Các gợi ý món ăn được cập nhật ngay lập tức và có thể theo dõi hiệu quả qua các báo cáo. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện Cấu hình AI gợi ý món, gọi API)  -BackEnd:Laravel(xử lý Cấu hình AI gợi ý món , truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |



### 2.1.13 PB13-Quản lý đơn hàng

|  |  |
| --- | --- |
| **ID** | **PB13** |
| **Tiêu đề** | Quản lý đơn hàng |
| **Mô tả** | Nhân viên có thể xem, tìm kiếm và duyệt đơn hàng trên hệ thống. |
| **Các bước thực hiện** | 1. Thiết kế giao diện quản lý đơn hàng  2. Thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu.  3. Nhận dữ liệu từ form khi thêm hoặc sửa thông tin đơn hàng.  Kiểm tra tính hợp lệ của dữ liệu  4.Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1.Đăng nhập vào hệ thống với tư cách là nhân viên.  Thực hiện các chức năng quản lý đơn hàng:   * Xem danh sách đơn hàng từ mới đến cũ. * Duyệt đơn hàng: Chuyển trạng thái đơn hàng từ "Chờ duyệt" sang "Đã duyệt".   Nhân viên có thể chọn hủy một đơn hàng nếu cần thiết.   * Nhấn nút "Hủy đơn hàng". * Hệ thống yêu cầu xác nhận hành động hủy. * Nếu xác nhận, hệ thống cập nhật trạng thái đơn hàng thành "hủy" và thông báo thành công. |
| **Điều kiên sau** | Thông tin đơn hàng được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện quản lý đơn hàng, gọi API)  -BackEnd:Laravel(xử lý quản lý đơn hàng, truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

### 2.1.14 PB14-Quản lý thanh toán

|  |  |
| --- | --- |
| **ID** | **PB14** |
| **Tiêu đề** | Quản lý thanh toán |
| **Mô tả** | Người dùng có thể quản lý thanh toán một cách hiệu quả, từ việc xem, cập nhật trạng thái đến thêm mới và hoàn trả giao dịch, giúp tối ưu hóa quy trình thanh toán trong nhà hàng. |
| **Các bước thực hiện** | 1. Thiết kế giao diện quản lý thanh toán  2. Xây dựng API tạo đơn hàng.  Tích hợp VNPay/Momo.(nếu thanh toán bằng Banking)  3. Xử lý callback sau thanh toán  4. Hiển thị trạng thái thanh toán trên giao diện. |
| **Điều kiện trước** | 1.Khách hàng yêu cầu thanh toán:  Khách hàng gọi nhân viên phục vụ hoặc tới quầy thanh toán để yêu cầu thanh toán.  2. Nhân viên kiểm tra đơn hàng:   * Nhân viên xác nhận thông tin đơn hàng của khách hàng bằng cách tra cứu trong hệ thống hoặc kiểm tra hóa đơn. * Hiển thị chi tiết các món ăn đã đặt, số lượng và tổng giá trị đơn hàng.   3. Chọn phương thức thanh toán:  Khách hàng xác nhận chi tiết đơn hàng và tiến hành thanh toán  Nhân viên hỏi khách hàng về phương thức thanh toán mà họ muốn sử dụng:   * Tiền mặt: Khách hàng đưa tiền mặt cho nhân viên. * Ngân hàng: Khách hàng có thể chuyển khoản qua ngân hàng.   4. Quản lý thanh toán bằng tiền mặt:  Nếu khách hàng thanh toán bằng tiền mặt:   * Nhân viên nhập số tiền khách đưa vào hệ thống. * Hệ thống kiểm tra và xác nhận số tiền. * Nhân viên trả lại tiền thừa (nếu có) cho khách hàng. * Cập nhật trạng thái giao dịch thành "đã thanh toán". * In hóa đơn thanh toán cho khách.   Quản lý thanh toán qua ngân hàng:   * Nếu khách hàng chọn thanh toán qua ngân hàng: * Nhân viên cung cấp thông tin tài khoản ngân hàng hoặc mã QR để khách hàng chuyển khoản. * Khách hàng thực hiện chuyển khoản từ tài khoản của họ. * Nhân viên xác nhận giao dịch đã hoàn tất (sau khi nhận được thông báo từ ngân hàng). * Cập nhật trạng thái giao dịch thành "đã thanh toán" và in hóa đơn cho khách.   5.Ghi nhận và lưu trữ:   * Hệ thống tự động lưu trữ thông tin giao dịch thanh toán để phục vụ cho việc báo cáo và thống kê sau này. |
| **Điều kiện sau** | Thông tin giao dịch thanh toán được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện quản lý thanh toán, gọi API)  -BackEnd:Laravel(xử lý quản lý thanh toán, truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Cao |

### 2.1.15 PB15-Đánh giá món ăn

|  |  |
| --- | --- |
| **ID** | **PB15** |
| **Tiêu đề** | Đánh giá món ăn |
| **Mô tả** | Khách hàng có thể đưa ra nhận xét, đánh giá về chất lượng của món ăn dựa trên nhiều tiêu chí khác nhau |
| **Các bước thực hiện** | 1. Thiết kế giao diện đánh giá món ăn  2. Gọi API đánh giá món ăn  3. Lọc đánh giá theo số sao, bình luận về món ăn  4. Hiển thị thông báo thành công hoặc thất bại |
| **Điều kiện trước** | 1.Sau khi khách hàng thực hiện thanh toán   * Nhân viên sẽ gửi hoá đơn và trong đó có mã QR để khách hàng có thể đánh giá món ăn mà mình đã dùng   2. Khách hàng quét mã sẽ hiện thông tin những món mình đã dùng  3. Khách hàng nhấn vào món cần đánh giá chọn số sao và có thể nêu lên cảm nhận hoặc góp ý với món ăn đó |
| **Điều kiên sau** | Đánh giá món ăn được cập nhật ngay lập tức trong hệ thống và trên giao diện người dùng. |
| **Công nghệ sử dụng** | -FrontEnd:ReactJs (Giao diện đánh giá món, gọi API)  -BackEnd:Laravel(xử lý đánh giá món, truy vấn cơ sở dữ liệu)  -Database:MySql(Lưu trữ thông tin)  -API:Restful API (Gửi và nhận dữ liệu từ backend) |
| **Độ phức tạp** | Trung bình |

* 1. Phân tích về Sprint Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Function | Started Date | Due Date |
| Sprint 1 | Develop | 27/03/2025 | 17/04/2205 |
| Sprint 2 | Develop | 18/04/2025 | 12/05/2025 |

1. Hạn chế

|  |  |
| --- | --- |
| Hạn chế | Tình trạng |
| Thời gian | Thời gian để hoàn thành dự án là 358 giờ. |
| Nhân lực | Toàn bộ nhóm, tổng cộng 5 người làm việc cùng nhau để hoàn thành dự án. |
| Các yêu cầu | Dựa trên các yêu cầu của Product owner. |

1. Tóm lược

|  |  |  |
| --- | --- | --- |
| Name | Mô tả | Role |
| Product Owner | Người đưa ra yêu cầu. | Cung cấp thông tin để phát triển hệ thống. Ra quyết định nghiệm thu và triển khai dự án, làm unit test. |
| Scrum Master | Đây là bên liên quan lãnh đạo, quản lý Nhóm phát triển hệ thống. | Kiểm soát, quản lý, giám sát, đảm bảo dự án hoàn thành đúng thời hạn, ngân sách, đúng kế hoạch và yêu cầu. |
| Requirement Analyzer | Đây là một bên liên quan làm việc với các Nhà phân tích để truyền đạt chính xác các yêu cầu hoặc nhu cầu thành các yêu cầu được sử dụng cho thiết kế. | Chỉ định chi tiết của một hoặc nhiều phần chức năng của hệ thống bằng cách mô tả một hoặc các khía cạnh của yêu cầu.  Điều này sẽ bao gồm chức năng và phi chức năng. |
| Software Architect | Đây là một bên liên quan nắm vai trò chính, dẫn dắt sự phát triển hệ thống. | Chịu trách nhiệm về kiến trúc phần mềm, bao gồm các quyết định kỹ thuật quan trọng hạn chế thiết kế tổng thể cho dự án. |
| Coder | Đây là bộ phận lập trình phần mềm. | Thực hiện dự án. |