**ĐẠI HỌC DUY TÂN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--------------🙖🙐✰🙖🙐-------------**

****

**Tên đề tài:**

**XÂY DỰNG WEBSITE ĐẶT ĐỒ ĂN TÍCH HỢP THANH TOÁN VNPAY VÀ AI CHATBOX TƯ VẤN**

**∙•🙞🟏🙜•∙**

**TÀI LIỆU PRODUCT BACKLOG**

GVHD: ThS. Tôn Thất Bình

Thành viên:

1. Phạm Xuân Nam – 0937
2. Trần Hữu Thủy – 0852
3. Phan Nữ Ngọc Hà – 8140
4. Trương Văn Trí – 1912
5. Mai Quang Vũ – 1372

*Đà Nẵng, 2025*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **THÔNG TIN DỰ ÁN** | | | | |
| **Dự án viết tắt** |  | | | |
| **Tên dự án** | Xây dựng website đặt đồ ăn tích hợp thanh toán VNPAY và AI ChatBox tư vấn | | | |
| **Thời gian bắt đầu** | 17/03/2025 | **Thời gian kết thúc** | 23/05/2025 | |
| **Lead Institution** | Khoa Công Nghệ Thông Tin, Đại Học Duy Tân | | | |
| **Giáo viên hướng dẫn** | ThS. Tôn Thất Bình  Email: tonthatbinh@dtu.edu.vn  Phone: 0979681065 | | | |
| **Chủ sở hữu dự án & Chi tiết liên hệ** | Trần Hữu Thủy  Email: huuthuytrann3004[@gmail.com](mailto:Huynhdngochoang@gmail.com)  Phone: 0943752093 | | | |
| **Đối tác** | Duy Tan University | | | |
| **Quản lý dự án &Scrum Master** | Phạm Xuân Nam | xuannampham2001[@gmail.com](mailto:trannhuthanh221202@gmail.com) | | 0359606379 |
| **Thành Viên Nhóm** | Trương Văn Trí | tritruongvan189[@gmail.com](mailto:manhnguyen2002.it@gmail.com) | | 0336880153 |
| Phan Nữ Ngọc Hà | pnnh1306[@gmail.com](mailto:lephuocloc2002.vn@gmail.com) | | 0948305246 |
| Trần Hữu Thủy | huuthuytrann3004[@gmail.com](mailto:Huynhdngochoang@gmail.com) | | 0943752093 |
| Mai QuangVũ | maiquangvu2306[@gmail.com](mailto:phuong119844@gmail.com) | | 0962230603 |

**THÔNG TIN TÀI LIỆU**

|  |  |
| --- | --- |
| **Tên dự án** | Xây dựng website đặt đồ ăn tích hợp thanh toán VNPAY và AI ChatBox tư vấn |
| **Tiêu đề tài liệu** | Product-Backlog Document |
| **Người thực hiện** | Phan Nữ Ngọc Hà |

**LỊCH SỬ CHỈNH CẬP NHẬT TÀI LIỆU**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phiên bản** | **Người chỉnh sửa** | **Ngày** | **Ghi chú** |
| 1.0 | Phan Nữ Ngọc Hà | 18/03/2025 | Tạo tài liệu |

**PHÊ DUYỆT TÀI LIỆU**

|  |  |  |  |
| --- | --- | --- | --- |
| **Người hướng dẫn** | Tôn Thất Bình | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |
| **Chủ sở hữu** | Trần Hữu Thủy | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |
| **Scrum Master** | Phạm Xuân Nam | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |
| **Thành viên** | Trương Văn Trí | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |
| Phan Nữ Ngọc Hà | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |
| Trần Hữu Thủy | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |
| Mai Quang Vũ | **Chữ ký** |  |
| **Ngày** | ………/ /2025 |

**MỤC LỤC**

[1. GIỚI THIỆU 1](#_Toc198495425)

[1.1. Mục đích. 1](#_Toc198495426)

[1.2. Phạm vi. 1](#_Toc198495427)

[1.3. Tài liệu tham khảo. 1](#_Toc198495428)

[2. Product Backlog 2](#_Toc198495429)

[2.1. Sơ đồ Use case tổng quát 2](#_Toc198495430)

[2.2. Đặc tả Product Backlog 3](#_Toc198495431)

[2.2.1. PB01 - Đăng ký 3](#_Toc198495432)

[2.2.2. PB02 - Đăng nhập 4](#_Toc198495433)

[2.2.3. PB03 – Đăng xuất 5](#_Toc198495434)

[2.2.4. PB04 – Đổi mật khẩu 5](#_Toc198495435)

[2.2.5. PB05 – Quên mật khẩu 6](#_Toc198495436)

[2.2.6. PB06 – Phân quyền 7](#_Toc198495437)

[2.2.7. PB07 – Tìm kiếm món ăn 8](#_Toc198495438)

[2.2.8. PB08 – Giỏ hàng 9](#_Toc198495439)

[2.2.9. PB09 – Đặt hàng 10](#_Toc198495440)

[2.2.10. PB10 – Thanh toán trực tuyến 10](#_Toc198495441)

[2.2.11. PB11 – Quản lý người dùng 11](#_Toc198495442)

[2.2.12. PB12 – Quản lý món ăn 12](#_Toc198495443)

[2.2.13. PB13 – Thống kê doanh thu 13](#_Toc198495444)

[2.2.14. PB14 – Quản lý danh mục 13](#_Toc198495445)

[2.2.15. PB15 – Chatbox tư vấn 14](#_Toc198495446)

[2.2.16. PB16 – Quản lý voucher 15](#_Toc198495447)

[2.2.17. PB17 – Đánh giá và bình luận 16](#_Toc198495448)

[2.2.18. PB18 – Quản lý đơn hàng 17](#_Toc198495449)

[3. Ưu tiên và ước tính 18](#_Toc198495450)

[3.1. 2.3. Chia theo từng Sprirnt 19](#_Toc198495451)

# GIỚI THIỆU

Product Backlog là tài liệu quan trọng trong quy trình Scrum, đóng vai trò là danh sách ưu tiên tất cả các tính năng, yêu cầu và công việc cần thực hiện để phát triển sản phẩm. Nội dung của Product Backlog thường bao gồm các User Story, bên cạnh đó còn có thể bao gồm yêu cầu chức năng, yêu cầu phi chức năng, lỗi (bugs) và các vấn đề kỹ thuật khác liên quan đến sản phẩm.

Product Backlog luôn mang tính chất linh hoạt và liên tục được cập nhật trong suốt vòng đời dự án. Các mục trong backlog có thể được bổ sung, chỉnh sửa hoặc thay đổi thứ tự ưu tiên theo yêu cầu mới từ Product Owner nhằm đảm bảo sản phẩm cuối cùng đáp ứng tốt nhất nhu cầu của người dùng và mục tiêu kinh doanh.

## Mục đích.

Tài liệu này thể hiện các yêu cầu cấp cao trong quan điểm của người dùng cuối. Những yêu cầu của người dùng sẽ được chia thành nhiều nhiệm vụ để phân công cho nhóm phát triển. Ngoài ra, tài liệu này cũng có các tiêu chuẩn chấp nhận, hữu ích cho người kiểm tra để tạo kế hoạch kiểm tra và kiểm thử.

## Phạm vi.

* Các user story trong dự án
* Các yêu cầu chức năng và phi chức năng
* Sơ đồ Use case

## Tài liệu tham khảo.

*Bảng 1: Tài liệu tham khảo*

|  |  |  |
| --- | --- | --- |
| **No** | **Reference** | **Note** |
| 1 | <http://agilebench.com/blog/the-product-backlog-for-agile-teams> | Cách tạo Product Backlog |
| 2 | <http://www.mountaingoatsoftware.com/agile/scrum/product-backlog/example/> | Ví dụ về Product Backlog |
| 3 | KLTN-SE.01.Product Proposal.docx | Proposal Document |

# Product Backlog

## Sơ đồ Use case tổng quát

A diagram of a diagram

AI-generated content may be incorrect.

*Hình 2.1: Use case tổng quát*

## Đặc tả Product Backlog

### PB01 - Đăng ký

|  |  |
| --- | --- |
| **ID** | PB01 |
| **Tiêu đề** | Đăng ký |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Người dùng đăng ký tài khoản để sử dụng hệ thống |
| **Các bước thực hiện** | 1. Frontend hiển thị form đăng ký (Họ tên, Email, SĐT, Mật khẩu, Xác nhận mật khẩu)  2. Người dùng nhập thông tin và nhấn nút Đăng ký  2.1 Gửi yêu cầu đăng ký (API POST) tới Backend  2.2 Backend kiểm tra dữ liệu, mã hóa mật khẩu, lưu vào database  2.3 Gửi email xác thực tài khoản (nếu cần)  3. Hiển thị thông báo đăng ký thành công hoặc lỗi nếu có |
| **Điều kiện trước** | 1. Người dùng chưa có tài khoản  2. Hệ thống backend đã kết nối CSDL và email server |
| **Điều kiện sau** | 1. Tài khoản mới được lưu trong database với trạng thái mặc định (chờ xác thực hoặc đã xác thực)  2. Người dùng có thể đăng nhập bằng thông tin đã đăng ký |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | 1. Email không được trùng lặp  2. Mật khẩu phải tuân thủ chính sách bảo mật (độ dài, ký tự đặc biệt, số, chữ hoa) |

### PB02 - Đăng nhập

|  |  |
| --- | --- |
| **ID** | PB02 |
| **Tiêu đề** | Đăng nhập |
| **Tác nhân** | Người dùng (Admin, User) |
| **Mô tả** | Người dùng đăng nhập vào hệ thống. |
| **Các bước thực hiện** | 1. Frontend hiển thị giao diện form đăng nhập  2. Người dùng nhập email và mật khẩu rồi nhấn nút Đăng nhập  2.1 Gửi yêu cầu POST đến API /login  2.2 Backend kiểm tra thông tin, trạng thái tài khoản  2.3 Nếu thành công: trả về JWT token → lưu vào localStorage và điều hướng đến giao diện chính  2.4 Nếu thất bại: hiển thị lỗi cụ thể (sai thông tin, tài khoản bị khóa, ...) |
| **Điều kiện trước** | 1. Người dùng đã có tài khoản hợp lệ  2. Backend và CSDL hoạt động ổn định |
| **Điều kiện sau** | 1. Token được lưu tại frontend, dùng xác thực request tiếp theo  2. Giao diện hiển thị các chức năng dành cho người đăng nhập |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | 1. Sau số lần sai mật khẩu nhất định → khóa tài khoản tạm thời  2. Token hết hạn → cần đăng nhập lại hoặc dùng refresh token |

### PB03 – Đăng xuất

|  |  |
| --- | --- |
| **ID** | PB03 |
| **Tiêu đề** | Đăng xuất |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Người dùng đăng xuất khỏi hệ thống |
| **Các bước thực hiện** | 1. Người dùng nhấn nút Đăng xuất  2. Frontend xóa JWT token khỏi localStorage  3. Chuyển hướng về màn hình đăng nhập/trang chủ |
| **Điều kiện trước** | Người dùng đang đăng nhập (token còn hiệu lực) |
| **Điều kiện sau** | Token bị xóa, người dùng không thể truy cập tài nguyên bảo vệ |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS |
| **Độ phức tạp** | Thấp |
| **Điều kiện ràng buộc** | 1. Nếu có refresh token → cũng phải hủy bỏ khi đăng xuất  2. Nếu dùng session → xóa session phía server |

### PB04 – Đổi mật khẩu

|  |  |
| --- | --- |
| **ID** | PB04 |
| **Tiêu đề** | Đổi mật khẩu |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Người dùng đổi mật khẩu trong tài khoản của mình |
| **Các bước thực hiện** | 1. Người dùng truy cập trang Đổi mật khẩu  2. Nhập mật khẩu cũ, mật khẩu mới, xác nhận mật khẩu mới  3. Gửi yêu cầu POST đến API đổi mật khẩu  4. Backend xác minh mật khẩu cũ, mã hóa mật khẩu mới và cập nhật database  5. Trả về thông báo thành công hoặc lỗi |
| **Điều kiện trước** | Người dùng đã đăng nhập và tài khoản còn hoạt động |
| **Điều kiện sau** | 1. Mật khẩu được thay đổi thành công  2. Người dùng có thể đăng nhập bằng mật khẩu mới |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | 1. Mật khẩu cũ phải đúng  2. Mật khẩu mới phải tuân theo chuẩn bảo mật  3. Có thể yêu cầu đăng nhập lại sau khi đổi mật khẩu |

### PB05 – Quên mật khẩu

|  |  |
| --- | --- |
| **ID** | PB06 |
| **Tiêu đề** | Quên mật khẩu |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Người dùng khôi phục mật khẩu khi không đăng nhập được |
| **Các bước thực hiện** | 1. Người dùng chọn “Quên mật khẩu” tại màn hình đăng nhập  2. Nhập email đăng ký  3. Gửi yêu cầu đến API để lấy lại mật khẩu  4. Backend kiểm tra email, tạo mã OTP hoặc token reset và gửi đến email  5. Người dùng truy cập link trong email hoặc nhập OTP để tạo mật khẩu mới  6. Backend cập nhật mật khẩu mới và thông báo thành công |
| **Điều kiện trước** | 1. Email tồn tại trong hệ thống  2. Dịch vụ email hoạt động bình thường |
| **Điều kiện sau** | 1. Người dùng có thể đăng nhập bằng mật khẩu mới  2. Token/OTP chỉ có hiệu lực trong thời gian giới hạn |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | 1. Mã OTP hoặc link reset chỉ dùng 1 lần và có thời gian hết hạn  2. Mật khẩu mới không được trùng mật khẩu cũ gần nhất |

### PB06 – Phân quyền

|  |  |
| --- | --- |
| **ID** | PB05 |
| **Tiêu đề** | Phân quyền |
| **Tác nhân** | Admin |
| **Mô tả** | Chức năng cho phép admin cấp quyền và quản lý vai trò của người dùng trong hệ thống như người dùng thông thường, khách hàng, admin... |
| **Các bước thực hiện** | **1. Xem danh sách người dùng và vai trò** 1.1. Admin truy cập mục phân quyền.  1.2. Gửi API GET để lấy danh sách người dùng và quyền hiện tại.  1.3. Backend truy vấn và trả dữ liệu (tên, email, vai trò).  1.4. Giao diện hiển thị danh sách người dùng cùng quyền hiện tại.  **2. Thay đổi vai trò người dùng** 2.1. Admin chọn người dùng cần thay đổi quyền.  2.2. Hiển thị popup chọn vai trò.  2.3. Gửi API PUT với ID người dùng và vai trò mới.  2.4. Backend xử lý cập nhật, ghi log thao tác.  2.5. Giao diện hiển thị thông báo và cập nhật danh sách. |
| **Điều kiện trước** | 1. Admin đã đăng nhập.  2. Hệ thống có bảng vai trò và người dùng liên kết. |
| **Điều kiện sau** | Vai trò mới được lưu, ghi log thao tác phân quyền. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Cao |
| **Điều kiện ràng buộc** | Chỉ Admin mới được phân quyền |

### PB07 – Tìm kiếm món ăn

|  |  |
| --- | --- |
| **ID** | PB07 |
| **Tiêu đề** | Tìm kiếm món ăn |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Chức năng cho phép người dùng tìm kiếm món ăn theo tên, loại,... |
| **Các bước thực hiện** | 1. Nhập từ khóa cần tìm kiếm 2. Gửi API GET với tham số truy vấn.3. Backend truy vấn dữ liệu phù hợp. 4. Trả kết quả danh sách món ăn.  5. Hiển thị kết quả trên giao diện. |
| **Điều kiện trước** | Hệ thống có sẵn món ăn trong CSDL. |
| **Điều kiện sau** | Danh sách kết quả hiển thị theo điều kiện tìm kiếm. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Thấp |
| **Điều kiện ràng buộc** | Không có |

### PB08 – Giỏ hàng

|  |  |
| --- | --- |
| **ID** | PB08 |
| **Tiêu đề** | Giỏ hàng |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Cho phép người dùng thêm, chỉnh sửa, xóa món ăn trong giỏ hàng của mình. |
| **Các bước thực hiện** | 1. Thêm món vào giỏ  2. Cập nhật số lượng món ăn  3. Xóa món khỏi giỏ  4. Gửi các API POST, PUT, DELETE tương ứng  5. Backend lưu trạng thái giỏ theo User  6. Giao diện hiển thị giỏ hàng cập nhật |
| **Điều kiện trước** | Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | Giỏ hàng được cập nhật và lưu |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | Mỗi người dùng chỉ quản lý được giỏ hàng của chính mình, không vượt quá số lượng tồn kho |

### PB09 – Đặt hàng

|  |  |
| --- | --- |
| **ID** | PB09 |
| **Tiêu đề** | Đặt món ăn |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Người dùng có thể đặt món trong giỏ và theo dõi tình trạng đơn hàng |
| **Các bước thực hiện** | 1. Xác nhận đơn hàng từ giỏ  2. Nhập thông tin giao hàng  3. Gửi API POST đặt đơn hàng  4. Backend xử lý tạo đơn và trả trạng thái  5. Hiển thị xác nhận thành công/thất bại |
| **Điều kiện trước** | Có món ăn trong giỏ |
| **Điều kiện sau** | Tạo đơn hàng thành công, lưu trạng thái "đang xử lý" |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | Chỉ người dùng đã đăng nhập mới được đặt hàng, thông tin giao hàng phải hợp lệ |

### PB10 – Thanh toán trực tuyến

|  |  |
| --- | --- |
| **ID** | PB10 |
| **Tiêu đề** | Thanh toán trực tuyến qua VNPAY |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Hỗ trợ người dùng thanh toán bằng cổng VNPAY an toàn và tiện lợi. |
| **Các bước thực hiện** | 1. Chọn phương thức thanh toán VNPAY2. Gửi yêu cầu thanh toán đến API thanh toán3. Chuyển hướng sang cổng VNPAY4. Người dùng thực hiện thanh toán5. VNPAY trả kết quả callback6. Backend xác nhận trạng thái đơn hàng7. Hiển thị thông báo cho người dùng |
| **Điều kiện trước** | Có đơn hàng chưa thanh toán |
| **Điều kiện sau** | Đơn hàng cập nhật trạng thái "đã thanh toán" |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Cao |
| **Điều kiện ràng buộc** | Chỉ thanh toán các đơn hàng còn hiệu lực, không áp dụng cho đơn hàng đã hủy hoặc hết hạn |

### PB11 – Quản lý người dùng

|  |  |
| --- | --- |
| **ID** | PB11 |
| **Tiêu đề** | Quản lý tài khoản người dùng |
| **Tác nhân** | Admin |
| **Mô tả** | Cho phép admin xem danh sách người dùng, tìm kiếm, cập nhật, xóa hoặc khóa tài khoản. |
| **Các bước thực hiện** | 1. Truy cập trang quản lý người dùng  2. Gửi API GET danh sách người dùng  3. Tìm kiếm hoặc lọc người dùng theo tên, email, vai trò  4. Gửi API PUT/DELETE để cập nhật hoặc xóa  5. Backend xử lý và trả kết quả |
| **Điều kiện trước** | Admin đã đăng nhập vào hệ thống |
| **Điều kiện sau** | Dữ liệu người dùng được cập nhật trong CSDL |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | Chỉ Admin mới được phép chỉnh sửa hoặc xóa tài khoản |

### PB12 – Quản lý món ăn

|  |  |
| --- | --- |
| **ID** | PB12 |
| **Tiêu đề** | Quản lý món ăn |
| **Tác nhân** | Admin |
| **Mô tả** | Admin có thể thêm, sửa, xóa và hiển thị danh sách các món ăn. |
| **Các bước thực hiện** | 1. Admin truy cập trang quản lý món ăn.  2. Chọn thao tác (thêm/sửa/xóa).  3. Gửi dữ liệu đến backend qua API.  4. Backend xử lý và lưu vào DB.  5. Giao diện cập nhật thay đổi. |
| **Điều kiện trước** | Danh mục món ăn đã có trong hệ thống. |
| **Điều kiện sau** | Dữ liệu món ăn được cập nhật thành công và hiển thị trên trang. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | Kiểm tra trùng tên, validate giá và hình ảnh. |

### PB13 – Thống kê doanh thu

|  |  |
| --- | --- |
| **ID** | PB13 |
| **Tiêu đề** | Thống kê doanh thu |
| **Tác nhân** | Admin |
| **Mô tả** | Admin có thể xem thống kê doanh thu theo ngày, tháng, năm hoặc tùy chỉnh thời gian. |
| **Các bước thực hiện** | 1. Admin chọn khoảng thời gian thống kê.  2. Hệ thống gửi yêu cầu truy vấn dữ liệu.  3. Backend tổng hợp doanh thu từ đơn hàng.  4. Giao diện hiển thị biểu đồ/tổng tiền. |
| **Điều kiện trước** | Hệ thống có dữ liệu đơn hàng. |
| **Điều kiện sau** | Thống kê hiển thị chính xác, nhanh chóng. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | Truy vấn tối ưu, thời gian phản hồi < 3s. |

### PB14 – Quản lý danh mục

|  |  |
| --- | --- |
| **ID** | PB14 |
| **Tiêu đề** | Qaunr lý danh mục |
| **Tác nhân** | Admin |
| **Mô tả** | Admin có thể thêm, sửa, xóa các danh mục món ăn (ví dụ: Đồ uống, Món chính, Tráng miệng). |
| **Các bước thực hiện** | 1. Truy cập giao diện quản lý danh mục.  2. Thêm/sửa/xóa danh mục.  3. Gửi dữ liệu đến backend.  4. DB cập nhật thông tin danh mục. |
| **Điều kiện trước** | Admin đã đăng nhập vào hệ thống |
| **Điều kiện sau** | Danh mục được cập nhật đúng và phản ánh trên các trang liên quan. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Thấp |
| **Điều kiện ràng buộc** | Không trùng tên danh mục, không xóa danh mục đã có món ăn. |

### PB15 – Chatbox tư vấn

|  |  |
| --- | --- |
| **ID** | PB15 |
| **Tiêu đề** | Chatbox tư vấn người dùng |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | Người dùng có thể đặt câu hỏi về món ăn, đơn hàng, chương trình khuyến mãi qua chatbot tư vấn. |
| **Các bước thực hiện** | 1. Người dùng mở cửa sổ chat.  2. Nhập câu hỏi.  3. Gửi tin nhắn qua API.  4. Backend gọi AI để trả lời.  5. Giao diện hiển thị câu trả lời. |
| **Điều kiện trước** | Người dùng đăng nhập. Chatbox đã tích hợp. |
| **Điều kiện sau** | Trả lời trong 2-5s, phản hồi đúng nội dung. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Cao |
| **Điều kiện ràng buộc** | Không trả lời nội dung nhạy cảm, có fallback nếu không hiểu. |

### PB16 – Quản lý voucher

|  |  |
| --- | --- |
| **ID** | PB16 |
| **Tiêu đề** | Quản lý mã giảm giá (voucher) |
| **Tác nhân** | Người dùng (Admin, User) |
| **Mô tả** | Admin có thể tạo, chỉnh sửa, và xóa mã giảm giá phục vụ chương trình khuyến mãi. User có thể áp dụng mã voucher khi thanh toán nếu đủ điều kiện. |
| **Các bước thực hiện** | ***Đối với Admin:***  1. Truy cập trang quản trị → mục “Quản lý Voucher”.  2. Thêm mới voucher với các trường: mã, hình thức giảm, giá trị giảm, số lượng áp dụng, ngày bắt đầu/kết thúc, điều kiện.  3. Chỉnh sửa hoặc xóa voucher nếu còn hiệu lực.  4. Hệ thống tự động vô hiệu hóa voucher nếu hết hạn hoặc hết lượt.  ***Đối với User:***  1. Truy cập trang thanh toán.  2. Nhập mã voucher và nhấn “Áp dụng”.  3. Hệ thống kiểm tra tính hợp lệ.  3.1. Nếu hợp lệ → giảm giá vào đơn hàng.  3.2. Nếu không hợp lệ → hiển thị thông báo lỗi. |
| **Điều kiện trước** | 1. Admin hoặc User phải đăng nhập thành công vào hệ thống.  2. Phải có dữ liệu voucher trong hệ thống (do admin tạo). |
| **Điều kiện sau** | 1. Admin có thể quản lý đầy đủ thông tin voucher.  2. User được áp dụng giảm giá nếu voucher hợp lệ. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Cao |
| **Điều kiện ràng buộc** | 1. Một voucher chỉ áp dụng 1 lần/khách hàng/đơn hàng.  2. Admin có thể huỷ hoặc ngừng hoạt động voucher bất kỳ lúc nào trước khi hết hạn. |

### PB17 – Đánh giá và bình luận

|  |  |
| --- | --- |
| **ID** | PB17 |
| **Tiêu đề** | Đánh giá và bình luận món ăn |
| **Tác nhân** | Người dùng (User) |
| **Mô tả** | User có thể đánh giá (1–5 sao) và bình luận món ăn sau khi đơn hàng hoàn tất. Có thể sửa hoặc xóa bình luận cá nhân, và xem đánh giá từ người khác. |
| **Các bước thực hiện** | 1. User đăng nhập vào hệ thống.  2. Vào mục lịch sử đơn hàng đã hoàn tất.  3. Chọn món ăn muốn đánh giá.  4. Nhập số sao (1–5) và viết bình luận.  5. Gửi đánh giá → lưu vào hệ thống.  6. Có thể sửa hoặc xóa bình luận cá nhân.  7. Các người dùng khác có thể xem đánh giá đã đăng. |
| **Điều kiện trước** | 1. Người dùng đã đăng nhập.  2. Đã hoàn tất ít nhất 1 đơn hàng chứa món ăn muốn đánh giá |
| **Điều kiện sau** | 1. Đánh giá và bình luận được lưu vào hệ thống.  2. Hệ thống cập nhật điểm trung bình cho từng món ăn. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL * API: Swagger |
| **Độ phức tạp** | Trung bình |
| **Điều kiện ràng buộc** | 1. Chỉ người dùng đã đặt và nhận món mới được đánh giá.  2. Mỗi món ăn trong đơn chỉ được đánh giá một lần. |

### PB18 – Quản lý đơn hàng

|  |  |
| --- | --- |
| **ID** | PB18 |
| **Tiêu đề** | Quản lý đơn hàng |
| **Tác nhân** | Admin |
| **Mô tả** | Admin có thể xem, lọc, tìm kiếm, cập nhật trạng thái và xóa đơn hàng trong hệ thống để đảm bảo quy trình vận hành và hậu cần. |
| **Các bước thực hiện** | 1. Admin đăng nhập vào hệ thống quản trị.  2. Truy cập mục “Quản lý đơn hàng”.  3. Xem danh sách đơn hàng: mã đơn, khách hàng, trạng thái, tổng tiền,...  4. Lọc theo trạng thái hoặc tìm kiếm theo tên khách/mã đơn.  5. Xem chi tiết đơn: món ăn, số lượng, giá, địa chỉ, thanh toán,…  6. Cập nhật trạng thái đơn theo tiến trình xử lý.  7. Xóa đơn hàng bị hủy hoặc không hợp lệ (popup xác nhận). |
| **Điều kiện trước** | 1. Admin đăng nhập hệ thống với quyền phù hợp.  2. Có dữ liệu đơn hàng trong hệ thống. |
| **Điều kiện sau** | 1. Admin theo dõi, xử lý trạng thái đơn hàng chính xác.  2. Tránh đơn hàng sai lệch hoặc không được xử lý. |
| **Công nghệ sử dụng** | * Frontend: VueJS, TailwindCSS * Backend: NodeJS, ExpressJS * Database: MySQL   API: Swagger |
| **Độ phức tạp** | Cao |
| **Điều kiện ràng buộc** | 1. Chỉ Admin có quyền mới được cập nhật hoặc xóa đơn.  2. Trạng thái đơn hàng phải được cập nhật đúng quy trình. |

# Ưu tiên và ước tính

*Bảng 3.1: Bảng mức độ ưu tiên*

|  |  |  |
| --- | --- | --- |
| **Mức độ ưu tiên** | **Mức độ ảnh hưởng** | **Ký hiệu** |
| Rất cao | Ảnh hưởng nghiêm trọng | 1 |
| Cao | Ảnh hưởng nghiêm trọng | 2 |
| Trung bình | Có ảnh hưởng | 3 |
| Thấp | Không ảnh hưởng | 4 |

|  |  |  |
| --- | --- | --- |
| **ID** | **Chủ đề** | **Sự ưu tiên** |
| PB01 | Đăng ký | 2 |
| PB02 | Đăng nhập | 2 |
| PB03 | Đăng xuất | 2 |
| PB04 | Đổi mật khẩu | 3 |
| PB05 | Quên mật khẩu | 3 |
| PB06 | Phân quyền | 1 |
| PB07 | Tìm kiếm món ăn | 3 |
| PB08 | Giỏ hàng | 1 |
| PB09 | Đặt hàng | 1 |
| PB10 | Thanh toán trực tuyến | 1 |
| PB11 | Quản lý người dùng | 2 |
| PB12 | Quản lý món ăn | 2 |
| PB13 | Thống kê doanh thu | 2 |
| PB14 | Quản lý danh mục | 2 |
| PB15 | Chatbox tư vấn | 2 |
| PB16 | Quản lý voucher | 3 |
| PB17 | Đánh giá và bình luận | 4 |
| PB18 | Quản lý đơn hàng | 2 |

## 2.3. Chia theo từng Sprirnt

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint** | **Function** | **Started Date** | **Due Date** |
| Sprint 2 | PB01, PB02, PB03, PB04, PB05, PB06, PB11. | 29/03/2025 | 11/04/2025 |
| Sprirnt 3 | PB08, PB09, PB10, PB12, PB14, PB16, PB17, PB18. | 12/04/2025 | 01/05/2025 |
| Sprirnt 4 | PB07, PB13, PB15. | 02/05/2025 | 15/05/2025 |