**LỜI MỞ ĐẦU**

Nhu cầu về ăn uống luôn là nhu cầu thiết yếu của xã hội. Với gần 100 triệu dân, Việt Nam được nhận định là thị trường tiềm năng cho các doanh nghiệp trong ngành ăn uống. Ngày nay, mức sống của người dân ngày một tăng cao và hiện đại hơn. Việc ăn nhà hàng đã trở nên phổ biến. Thực tế là những nhà hàng, quán ăn với quy mô từ nhỏ đến lớn đang mọc lên như nấm sau mưa.

Ngành ăn uống tại Việt Nam đang phát triển mạnh mẽ và đa dạng, từ các quán ăn đường phố đến những nhà hàng sang trọng. Thói quen ăn uống ngoài gia đình ngày càng gia tăng, không chỉ ở các đô thị lớn mà còn lan rộng đến các vùng nông thôn. Theo các báo cáo, thị trường dịch vụ ăn uống tại Việt Nam đã đạt mức tăng trưởng hàng năm ấn tượng, phản ánh sự thay đổi trong lối sống và nhu cầu của người tiêu dùng. Tuy nhiên, cùng với sự bùng nổ của ngành này, việc quản lý và vận hành hiệu quả nhà hàng trở thành một thách thức không nhỏ đối với các chủ nhà hàng.

Sự phổ biến của smartphone đã tạo ra một sự chuyển đổi đáng kể trong thói quen tiêu dùng của người dân. Hiện nay, hầu hết người dân Việt Nam, đặc biệt là giới trẻ, đều sở hữu ít nhất một chiếc smartphone. Điều này mở ra cơ hội lớn cho các nhà hàng trong việc tiếp cận khách hàng thông qua các ứng dụng di động và mạng xã hội. Khách hàng có thể dễ dàng tìm kiếm, đặt bàn, đặt món ăn và thanh toán trực tuyến chỉ với vài thao tác trên điện thoại. Đây là một xu hướng không thể bỏ qua đối với các doanh nghiệp trong ngành ăn uống.

Việc áp dụng công nghệ thông tin (CNTT) vào quản lý nhà hàng đã và đang trở thành xu hướng tất yếu. Các hệ thống quản lý nhà hàng thông minh giúp tự động hóa nhiều quy trình từ đặt chỗ, gọi món, thanh toán đến quản lý kho, báo cáo doanh thu và đánh giá hiệu quả kinh doanh. Những ứng dụng này không chỉ giúp tối ưu hóa hoạt động của nhà hàng, giảm thiểu sai sót mà còn nâng cao trải nghiệm khách hàng. Nhiều nhà hàng tại Việt Nam đã đầu tư vào các hệ thống quản lý hiện đại, góp phần nâng cao chất lượng dịch vụ và khả năng cạnh tranh trên thị trường.

Tóm lại, trong bối cảnh nhu cầu ăn uống ngày càng cao và sự phát triển mạnh mẽ của công nghệ, việc xây dựng một hệ thống quản lý nhà hàng hiệu quả là vô cùng cần thiết. Điều này không chỉ giúp các nhà hàng vận hành trơn tru, tiết kiệm chi phí mà còn đáp ứng tốt hơn nhu cầu của khách hàng, từ đó tạo nên lợi thế cạnh tranh vững chắc trên thị trường.

**LỜI CAM ĐOAN**

Tôi xin cam đoan rằng đồ án tốt nghiệp này là kết quả nghiên cứu và làm việc của riêng tôi. Mọi số liệu và kết quả trình bày trong đồ án đều trung thực và chính xác. Đồ án này không sao chép dưới bất kỳ hình thức nào. Nếu có sử dụng tài liệu tham khảo, tôi đã trích dẫn và ghi nguồn đầy đủ theo quy định.

**LỜI CẢM ƠN**

Em xin chân thành cảm ơn T.S Trần Thị Minh Hoàn, người đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đồ án tốt nghiệp này. Sự chỉ dẫn quý báu và sự giúp đỡ của cô đã giúp em hoàn thành đồ án một cách tốt nhất.

Em cũng xin gửi lời cảm ơn sâu sắc đến tất cả thầy cô và bạn bè tại Trường Đại học Thủy Lợi, những người đã đồng hành và hỗ trợ em trong quá trình học tập và nghiên cứu tại trường.

Đặc biệt, em xin bày tỏ lòng biết ơn tới các thầy cô trong Khoa Công Nghệ Thông Tin, Đại học Thủy Lợi, những người đã truyền đạt cho em những kiến thức và kỹ năng quý báu, giúp em có được nền tảng vững chắc để hoàn thành đồ án này.

Cuối cùng, do thời gian có hạn và kiến thức còn hạn chế, đồ án của em chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ quý thầy cô và bạn bè để có thể hoàn thiện đồ án tốt hơn trong tương lai.

Em xin chân thành cảm ơn!

1. **Giới thiệu đề tài**
   1. **Lý do chọn đề tài**

Ngành dịch vụ ăn uống ở Việt Nam đang có tiềm năng phát triển rất lớn. Ngày nay, nhiều nhà hàng và quán ăn được xây dựng, và việc ăn nhà hàng đã trở nên phổ biến, không còn xa xỉ với nhiều người dân. Tuy nhiên, phần lớn các nhà hàng, quán ăn vẫn thực hiện quy trình phục vụ một cách thủ công, sử dụng sổ sách và giấy tờ, gây khó khăn và tốn kém trong công tác quản lý cũng như vận hành. Mặc dù việc áp dụng công nghệ thông tin (CNTT) trong quản lý nhà hàng đang ngày càng được mở rộng, vẫn còn nhiều nhà hàng chưa áp dụng các công nghệ này.

Bên cạnh việc phát triển ngành ăn uống Việt Nam còn là một quốc gia đang phát triển với số lượng lớn người sử dụng thiết bị di động. Theo thông tin từ Bộ Thông tin và Truyền thông, đến tháng 5-2023, cả nước có khoảng 123,26 triệu thuê bao điện thoại di động. Thống kê cũng cho thấy, lượng người sử dụng điện thoại thông minh ở Việt Nam đứng top đầu Đông Nam Á.

Để khắc phục các hạn chế trong việc quản lý và vận hành của nhà hàng, đồng thời tận dụng lợi thế về việc sử dụng smartphone, em xin đề xuất giải pháp xây dựng ứng dụng di động quản lý việc đặt món ăn và thanh toán cho nhà hàng PORO Food.

* 1. **Tóm tắt đề tài**

Đồ án hướng tới xây dựng một ứng dụng quản lý nhà hàng nhỏ gọn. Với các tính năng chỉ tập trung vào phần thiết yếu của một nhà hàng đó là “Quản lý đặt món”. Mỗi nhân viên sẽ được cấp một tài khoản cá nhân được dùng để thực hiện công việc của mình. Người quản lý sẽ có thêm các chức năng quản lý tài khoản, hóa đơn, mặt hàng. Mọi thông tin sẽ được lưu trữ nhằm phục vụ việc thống kê.

* 1. **Mục tiêu chính**
* Xây dựng Cơ sở dữ liệu bằng MySql
* Viết API cho ứng dụng bằng Laravel
* Thiết kế hệ thống phù hợp với yêu cầu thực tiễn
* Xây dựng ứng dụng di động bằng Flutter
  1. **Kết quả dự kiến**
* Xây dựng thành công ứng dụng quản lý đặt món “PORO Food”
* Xây dựng thành công Cơ sở dữ liệu phù hợp yêu cầu
* Phát triển thành công API cho ứng dụng bằng Laravel
* Nắm rõ kiến thức và thiết kế thành công hệ thống quản lý đặt món

1. **Khảo sát và phân tích nghiệp vụ**
   1. **Đánh giá hiện trạng**

Nhà hàng PORO Food là một nhà hàng nhỏ nằm trên địa bàn Hà Nội, với quy mô không lớn, được chia thành các bộ phận sau:

* Quản lý nhà hàng
* Nhân viên bếp
* Nhân viên phục vụ
* Nhân viên thu ngân

Hiện tại, công tác quản lý tại nhà hàng chủ yếu dựa trên ghi chép giấy tờ, mọi công việc đều được thực hiện một cách thủ công. Điều này bao gồm các công việc như thống kê doanh thu, cập nhật danh sách món ăn, ghi nhận đơn đặt món từ khách và chuyển đơn đến các bộ phận khác. Việc vận hành và quản lý thủ công này gây ra nhiều khó khăn và tốn nhiều thời gian, nhân lực.

Một trong những khó khăn lớn nhất là trong công tác thống kê do sử dụng hóa đơn giấy. Việc ghi chép, lưu trữ và tổng hợp thông tin doanh thu trên giấy không chỉ mất nhiều thời gian mà còn dễ dẫn đến sai sót. Những sai sót này có thể gây ra các vấn đề trong việc quản lý tài chính, ảnh hưởng đến hoạt động kinh doanh của nhà hàng.

Bên cạnh đó, việc giao tiếp giữa nhân viên bếp và nhân viên phục vụ cũng gặp nhiều khó khăn. Khi sử dụng hệ thống ghi chép thủ công, các đơn đặt món từ khách phải được ghi nhận và chuyển tới bếp qua giấy tờ. Quá trình này không chỉ làm tăng khả năng nhầm lẫn, sai sót mà còn làm chậm quá trình phục vụ khách hàng, ảnh hưởng đến trải nghiệm của khách tại nhà hàng.

Tóm lại, việc quản lý chủ yếu làm thủ công, không dựa vào sự hỗ trợ của phần mềm quản lý, đang gây ra nhiều thách thức và khó khăn cho nhà hàng PORO Food, từ việc thống kê doanh thu đến giao tiếp và phối hợp giữa các bộ phận.

* 1. **Khảo sát nghiệp vụ**
* **Quy trình nghiệp vụ của nhân viên phục vụ**

Khi có khách đến nhà hàng, nhân viên phục vụ sẽ hỏi số lượng khách đi cùng để có thể sắp xếp bàn một cách hợp lý. Sau khi khách được dẫn đến chỗ ngồi, nhân viên sẽ đưa menu để khách chọn món và ghi lại thông tin về các món ăn và yêu cầu từ khách hàng. Thông tin đặt món này sau đó sẽ được chuyển giao tới nhân viên bếp để bắt đầu quá trình chế biến.

Khi món ăn đã được chuẩn bị xong, nhân viên phục vụ sẽ mang ra cho khách và đồng thời chuyển tiếp thông tin đặt món này cho nhân viên thu ngân. Nhân viên thu ngân sẽ lập hóa đơn mới hoặc cập nhật thêm vào hóa đơn có sẵn, đảm bảo mọi thông tin về các món ăn đã được ghi nhận đầy đủ.

Khi khách yêu cầu thanh toán, nhân viên phục vụ sẽ lấy hóa đơn từ thu ngân và đưa cho khách kiểm tra. Sau khi khách hàng đã kiểm tra và xác nhận hóa đơn, nhân viên phục vụ sẽ dẫn khách ra quầy thu ngân để thanh toán. Cuối cùng, nhân viên phục vụ sẽ tiễn khách ra về, đảm bảo rằng họ đã có một trải nghiệm hài lòng tại nhà hàng.

* **Quy trình nghiệp vụ của nhân viên bếp**

Khi tiếp nhận thông tin đặt món từ nhân viên phục vụ, nhân viên bếp bắt đầu quy trình chế biến món ăn. Thông tin đặt món bao gồm các yêu cầu cụ thể từ khách hàng, giúp nhân viên bếp chuẩn bị món ăn đúng theo yêu cầu.

Trước tiên, nhân viên bếp sẽ ghi nhận và phân loại các món ăn cần chế biến, sắp xếp thứ tự ưu tiên dựa trên thời gian đặt món. Nhân viên bếp sẽ theo dõi chặt chẽ các đơn đặt món để biết được món nào đang được nấu, món nào đã hoàn thành, và món nào đang chờ chuẩn bị. Điều này giúp đảm bảo rằng không có món ăn nào bị bỏ quên hoặc chậm trễ.

Khi một món ăn đã hoàn thành, nhân viên bếp sẽ báo lại với nhân viên phục vụ thông qua hệ thống thông báo hoặc trực tiếp, để họ đến nhận món ăn và đưa ra cho khách hàng.

* **Quy trình nghiệp vụ nhân viên thu ngân**

Khi tiếp nhận yêu cầu thanh toán từ nhân viên phục vụ, nhân viên thu ngân sẽ bắt đầu quy trình xuất hóa đơn. Dựa trên thông tin từ các món ăn đã được phục vụ, nhân viên thu ngân sẽ kiểm tra,lập hóa đơn chi tiết và bản sao hóa đơn, đảm bảo rằng tất cả các món ăn và yêu cầu của khách hàng đã được ghi nhận đầy đủ và chính xác.

Sau khi hóa đơn được lập xong, nhân viên thu ngân sẽ chuyển hóa đơn này cho nhân viên phục vụ để khách hàng kiểm tra. Nếu khách hàng có bất kỳ thắc mắc hoặc yêu cầu chỉnh sửa nào về hóa đơn, nhân viên thu ngân sẽ phối hợp với nhân viên phục vụ và bộ phận bếp để xử lý kịp thời.

Khi khách hàng chấp nhận và tiến hành thanh toán, nhân viên thu ngân sẽ tiếp nhận tiền thanh toán. Sau đó, nhân viên thu ngân sẽ giữ lại bản sao hóa đơn đã thanh toán để phục vụ cho công tác thống kê và quản lý tài chính của nhà hàng. Sau khi hoàn tất quá trình thanh toán, nhân viên thu ngân sẽ thông báo với nhân viên phục vụ để dọn dẹp bàn vừa được thanh toán, chuẩn bị sẵn sàng cho lượt khách tiếp theo..

* **Quy trình nghiệp vụ quản lý nhà hàng**

Vào đầu mỗi ngày, quản lý nhà hàng sẽ thực hiện việc nhập nguyên liệu cần thiết. Quá trình này bao gồm nhận hàng từ nhà cung cấp, kiểm tra chất lượng và số lượng của từng loại nguyên liệu để đảm bảo rằng tất cả đều đạt tiêu chuẩn. Sau khi hoàn tất việc nhập nguyên liệu, quản lý sẽ tiến hành bàn giao số lượng và loại nguyên liệu cần thiết cho nhân viên bếp để bắt đầu ca làm việc.

Cuối ngày, quản lý sẽ tiến hành thống kê số lượng các món ăn đã được bán trong ngày. Việc này không chỉ giúp xác định doanh số bán hàng mà còn cung cấp dữ liệu về mức độ tiêu thụ của từng món ăn, từ đó có thể điều chỉnh menu và lượng nguyên liệu nhập vào cho phù hợp. Đồng thời, quản lý cũng sẽ kiểm kê lại số lượng nguyên liệu còn thừa sau một ngày làm việc. Điều này giúp đánh giá hiệu quả sử dụng nguyên liệu và xác định những nguyên liệu cần được bổ sung hoặc điều chỉnh trong những ngày tiếp theo.

Cuối cùng, quản lý sẽ tổng kết doanh thu bán hàng trong ngày. Thông tin này bao gồm số lượng hóa đơn, tổng số tiền thu được và các khoản chi phí phát sinh. Việc tổng kết doanh thu giúp quản lý có cái nhìn tổng quan về hiệu quả kinh doanh và chuẩn bị báo cáo tài chính.

* **Đánh giá quy trình**

Ưu điểm:

* Đáp ứng yêu cầu cơ bản về việc quản lý một nhà hàng, quán ăn nhỏ: Quy trình hiện tại cho phép nhà hàng vận hành và phục vụ khách hàng mà không cần đầu tư vào các hệ thống phức tạp.
* Không tốn chi phí cho việc mua, cài đặt phần mềm quản lý, đào tạo nhân viên sử dụng: Sử dụng phương pháp thủ công giúp tiết kiệm chi phí ban đầu cho phần mềm và đào tạo nhân viên, giảm bớt áp lực tài chính cho nhà hàng nhỏ.

Nhược điểm:

* Nhân viên sẽ bị quá tải khi lượng khách hàng đông do tất cả công việc đều phải xử lý thủ công bằng tay: Khi có nhiều khách hàng, việc ghi chép, chuyển giao thông tin và quản lý hóa đơn bằng tay có thể làm nhân viên quá tải, dẫn đến giảm chất lượng phục vụ.
* Quá trình bảo quản, tìm kiếm tốn nhiều thời gian và công sức do sử dụng nhiều giấy tờ: Việc lưu trữ và quản lý giấy tờ một cách thủ công đòi hỏi nhiều không gian và thời gian, gây khó khăn trong việc tìm kiếm và bảo quản thông tin.
* Dễ xảy ra sai sót trong quá trình quản lý do nhiều yếu tố như: viết sai, tính toán nhầm: Sai sót có thể xảy ra do chữ viết khó đọc, ghi chép không chính xác hoặc nhầm lẫn trong tính toán, ảnh hưởng đến hiệu quả quản lý và trải nghiệm khách hàng.
* Thiếu khả năng theo dõi và báo cáo chi tiết: Việc sử dụng giấy tờ không cho phép tạo ra các báo cáo chi tiết và nhanh chóng về doanh thu, tồn kho, hoặc hiệu suất làm việc của nhân viên, gây khó khăn trong việc ra quyết định quản lý.
* Khó khăn trong việc kiểm soát tồn kho và đặt hàng: Quản lý tồn kho thủ công dễ dẫn đến tình trạng hết hàng hoặc dư thừa nguyên liệu, làm ảnh hưởng đến hoạt động bếp và phục vụ khách hàng.
* Khả năng bảo mật thông tin thấp: Giấy tờ dễ bị mất, hư hỏng hoặc bị lộ thông tin, ảnh hưởng đến bảo mật dữ liệu khách hàng và hoạt động kinh doanh của nhà hàng.

Quy trình thủ công tuy có thể đáp ứng được các nhu cầu cơ bản của nhà hàng nhỏ và tiết kiệm chi phí, nhưng đi kèm với đó là nhiều hạn chế và khó khăn trong quản lý và vận hành, đặc biệt là khi lượng khách hàng tăng cao.

* 1. **Xác định yêu cầu**
     1. **Yêu cầu chức năng:**
* Cho phép nhân viên đăng nhập vào hệ thống
* Cho phép tra cứu danh sách món ăn
* Cho phép nhân viên phục vụ tạo đơn đặt món, xem hóa đơn
* Gửi thông báo tương tác giữa nhân viên phục vụ và nhân viên bếp
* Cho phép quản lý thêm/sửa/xóa bàn trong quán
* Cho phép quản lý thêm/sửa/xóa món ăn, mặt hàng
* Cho phép quản lý thêm/sửa/xóa tài khoản nhân viên
* Cho phép quản lý thêm/sửa/xóa hóa đơn
* Cho phép quản lý thống kế doanh thu theo ngày, tháng, năm, món ăn bán được nhiều nhất
  + 1. **Yêu cầu phi chức năng:**
* Phần mềm hoạt động tốt, truy xuất dữ liệu nhanh
* Phần mềm có giao diện đẹp, dễ sử dụng
  1. **Dự kiến hệ thống mới**

Dựa theo thực trạng, tính chất nghiệp vụ và quy mô nhà hàng, hệ thống mới sẽ chỉ tập trung những chức năng chính để phục vụ việc vận hành và hoạt động của nhà hàng. Các chức năng cụ thể như sau:

Quản lý sẽ nhập thông tin món ăn, mặt hàng, thông tin bàn trong quán, tạo tài khoản để nhân viên đăng nhập vào hệ thống.

Nhân viên phục vụ có thể chọn bàn, chọn món ăn, tạo đơn đặt món và chuyển thông tin trực tuyến đến nhân viên bếp.

Nhân viên bếp có thể quản lý danh sách đơn đặt món, cập nhật tình trạng món ăn đến nhân viên phục vụ.

Ngoài ra hệ thống còn có những chức năng như:

* Thống kê doanh thu, hóa đơn, món ăn bán nhiều nhất
* Tra cứu thông tin món ăn
* Tra cứu thông tin hóa đơn

1. **Thiết kế hệ thống**
   1. **Xác định tác nhân**

**Quản lý:** Tác nhân này chịu trách nhiệm quản lý toàn bộ các hoạt động của nhà hàng.

**Phục vụ:** Tác nhân này chịu trách nhiệm về việc nhận đơn đặt món, phục vụ khách hàng và xử lý các thông tin liên quan đến đặt món.

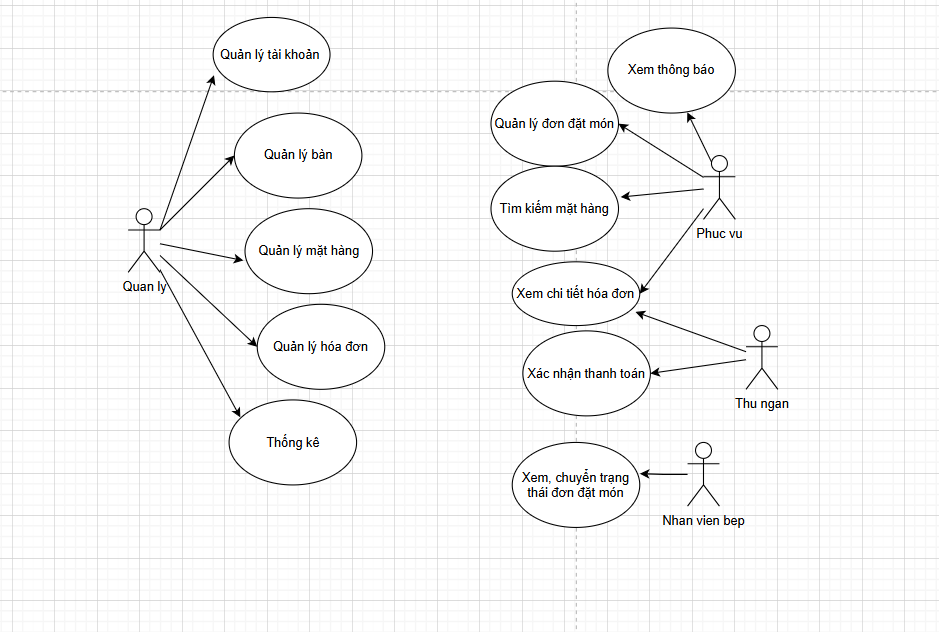
**Thu ngân:** Tác nhân này chịu trách nhiệm về việc xác nhận thanh toán hóa đơn.

**Nhân viên bếp**: Tác nhân này chịu trách nhiệm về việc xem và chuyển trạng thái đơn đặt món.

* 1. **Xác định các usecase**

Hệ thống có các usecase chính như sau:

* Quản lý tài khoản
* Quản lý bàn
* Quản lý mặt hàng
* Quản lý hóa đơn
* Thống kê
* Quản lý đơn đặt món
* Tìm kiếm mặt hàng
* Xem chi tiết hóa đơn
* Xem thông báo
* Xác nhận thanh toán
* Xem, chuyển trạng thái đơn đặt món
  1. **Biểu đồ usecase tổng quát cho toàn bộ hệ thống**



Biểu đồ usecase tổng quát

* 1. **Thiết kế chi tiết từng chức năng**
     1. **Chức năng đăng nhập**

**A diagram with a black and white text

Description automatically generated**

Hình: Biểu đồ usecase đăng nhập

A diagram of a flowchart

Description automatically generated

Biểu đồ hoạt động chức năng đăng nhập

A diagram of a program

Description automatically generated

Biểu đồ tuần tự chức năng đăng nhập

* + 1. **Chức năng đổi mật khẩu**

**A diagram of a person with a diagram

Description automatically generated**

Biểu đồ usecase đổi mật khẩu

A diagram of a flowchart

Description automatically generated

Biểu đồ hoạt động chức năng đổi mật khẩu

A diagram of a company

Description automatically generated

Biểu đồ tuần tự chức năng đổi mật khẩu

* + 1. **Chức năng tạo đơn đặt món**

A diagram of a diagram

Description automatically generated

Biểu đồ usecase tạo đơn đặt món

|  |  |
| --- | --- |
| Tên Use case | Use case tạo đơn đặt món |
| Tác nhân chính | Phục vụ |
| Mức | 1 |
| Tiền điều kiện | Phục vụ đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin đã thêm và quay lui lại bước trước |
| Đảm bảo thành công | Hệ thống tạo đơn đặt món |
| Kích hoạt | Phục vụ đăng nhập và bấm vào chọn bàn |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách bàn 2. Phục vụ chọn bàn trống 3. Hệ thống hiển thị danh sách mặt hàng 4. Phục vụ chọn mặt hàng và bấm “Thêm vào đơn” 5. Hệ thống hiển thị chi tiết đơn đặt món 6. Phục vụ kiểm tra lại, chỉnh sửa số lượng và bấm “Thông báo” | |

Mô tả usecase Tạo đơn đặt món

|  |  |
| --- | --- |
| Tên Use case | Use case chỉnh sửa đơn đặt món |
| Tác nhân chính | Phục vụ |
| Mức | 1 |
| Tiền điều kiện | Phục vụ đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin đã thêm và quay lui lại bước trước |
| Đảm bảo thành công | Hệ thống cập nhật đơn đặt món |
| Kích hoạt | Phục vụ đăng nhập và bấm vào chọn bàn |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách bàn 2. Phục vụ chọn bàn đang phục vụ 3. Hệ thống hiển thị đơn đặt món chưa được chế biến 4. Phục vụ điều chỉnh số lượng món 5. Phục vụ chọn xem hóa đơn 6. Hệ thống hiển thị món ăn đã/đang chế biến 7. Phục vụ chọn “Thêm món” 8. Hệ thống hiển thị danh sách mặt hàng 9. Phục vụ chọn mặt hàng và bấm “Thêm vào đơn” 10. Hệ thống hiển thị chi tiết đơn đặt món 11. Phục vụ kiểm tra lại, chỉnh sửa số lượng và bấm “Thông báo” | |

Mô tả usecase chỉnh sửa đơn đặt món/thêm đơn đặt món

A diagram of a flowchart

Description automatically generated

Biểu đồ hoạt động chức năng tạo đơn đặt món

A diagram of a project

Description automatically generated

Biều đồ tuần tự chức năng tạo đơn đặt món

* + 1. **Chức năng Cập nhật trạng thái đơn đặt món**

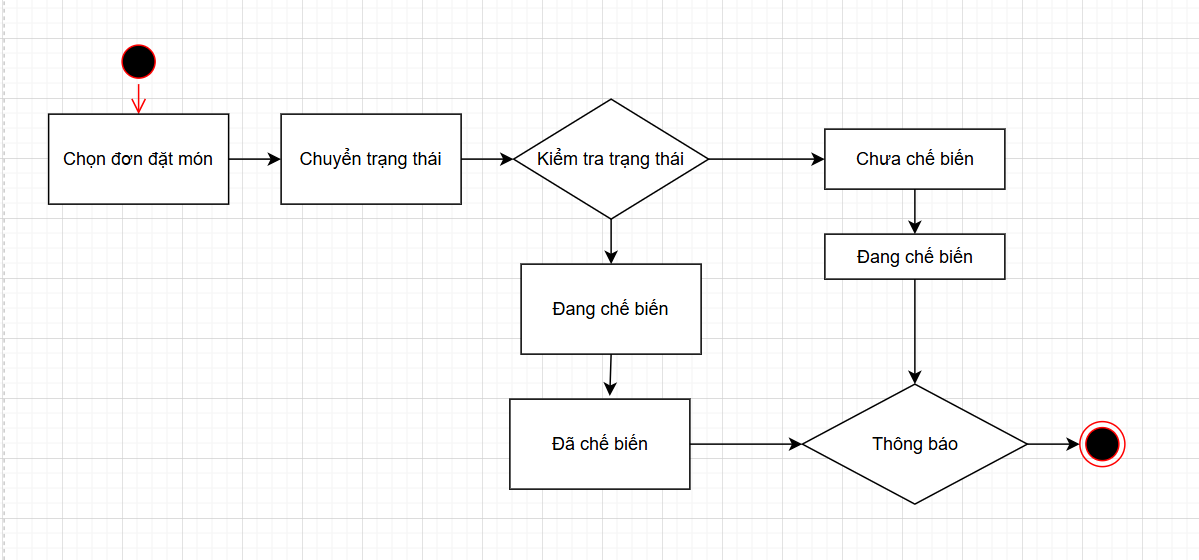
A white grid with black text

Description automatically generated

Biểu đồ usecase Cập nhật trạng thái đơn đặt món

|  |  |
| --- | --- |
| Tên Use case | Use case cập nhật trạng thái đơn đặt món |
| Tác nhân chính | Nhân viên bếp |
| Mức | 1 |
| Tiền điều kiện | Nhân viên bếp đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin đã thêm và quay lui lại bước trước |
| Đảm bảo thành công | Hệ thống cập nhật trạng thái đơn đặt món |
| Kích hoạt | Nhân viên bếp đăng nhập vào hệ thống |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách đơn đặt món 2. Nhân viên bếp chọn 1 đơn đặt món 3. Nhân viên bếp chuyển trạng thái đơn đặt món (Đang chế biến/Đã chế biến) 4. Hệ thống cập nhật trạng thái đơn đặt món 5. Hệ thống gửi thông báo tới phục vụ | |

Mô tả usecase cập nhật trạng thái đơn đặt món



Biểu đồ hoạt động chức năng chuyển trạng thái đơn đặt món

A diagram of a building

Description automatically generated

Biểu đồ tuần tự chức năng chuyển trạng thái đơn đặt món

* + 1. **Chức năng xác nhận thanh toán**

A diagram with text in a circle

Description automatically generated

Biểu đồ usecase xác nhận thanh toán

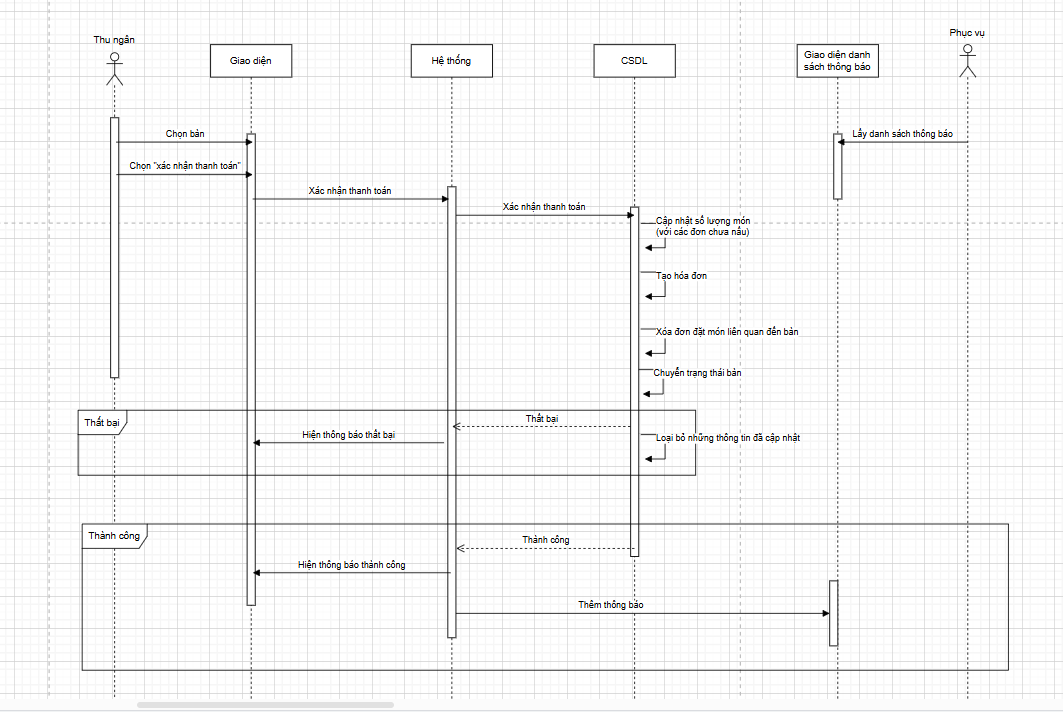
|  |  |
| --- | --- |
| Tên Use case | Use case xác nhận thanh toán |
| Tác nhân chính | Thu ngân |
| Mức | 1 |
| Tiền điều kiện | Thu ngân đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin và quay lại bước trước |
| Đảm bảo thành công | Hệ thống cập nhật trạng thái của bàn |
| Kích hoạt | Thu ngân đăng nhập và bấm chọn bàn |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách bàn 2. Thu ngân chọn bàn đang phục vụ 3. Hệ thống hiển thị hóa đơn chi tiết 4. Thu ngân chọn xác nhận thanh toán 5. Hệ thống hiển thị thông báo xác nhận 6. Thu ngân bấm “Xác nhận” 7. Hệ thống chuyển trạng thái bàn về “Trống” | |
| Ngoại lệ:   1. Thu ngân bấm “Hủy”   1.1 Hệ thống đóng thông báo xác nhận | |

Mô tả usecase Xác nhận thanh toán

A diagram of a flowchart

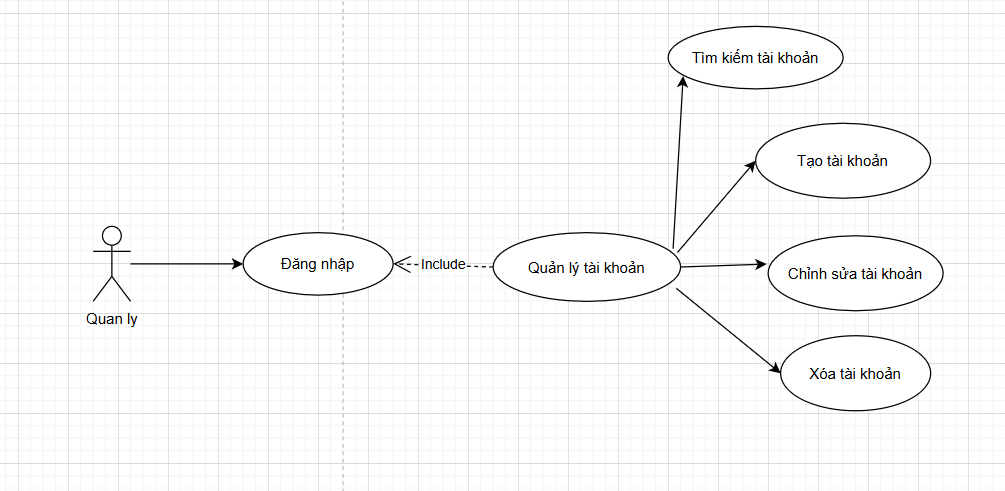
Description automatically generated

Biểu đồ hoạt động chức năng Xác nhận thanh toán



Biểu đồ tuần tự chức năng Xác nhận thanh toán

* + 1. **Chức năng quản lý tài khoản**

****

Biểu đồ usecase chức năng quản lý tài khoản

|  |  |
| --- | --- |
| Tên Use case | Use case tạo tài khoản |
| Tác nhân chính | Quản lý |
| Mức | 2 |
| Tiền điều kiện | Quản lý đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin và quay lại bước trước |
| Đảm bảo thành công | Hệ thống tạo tài khoản mới |
| Kích hoạt | Quản lý đăng nhập và bấm chọn quản lý tài khoản |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách tài khoản 2. Quản lý bấm chọn tạo tài khoản 3. Hệ thống hiển thị form tạo tài khoản 4. Quản lý điền thông tin 5. Quản lý bấm nút “Tạo” 6. Hệ thống tạo tài khoản mới | |
| Ngoại lệ:   1. Quản lý nhập thiếu thông tin    1. Hệ thống hiển thị thông báo nhắc nhở 2. Quản lý nhập mật khẩu quá ngắn   2.1 Hệ thống hiển thị thông báo “Mật khẩu phải lớn hơn 8 kí tự” | |

Mô tả usecase Tạo tài khoản

A diagram of a diagram

Description automatically generated

Biểu đồ hoạt động chức năng Tạo tài khoản

A diagram of a diagram

Description automatically generated

Biểu đồ tuần tự chức năng Tạo tài khoản

|  |  |
| --- | --- |
| Tên Use case | Use case chỉnh sửa tài khoản |
| Tác nhân chính | Quản lý |
| Mức | 2 |
| Tiền điều kiện | Quản lý đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin và quay lại bước trước |
| Đảm bảo thành công | Hệ thống cập nhật thông tin tài khoản |
| Kích hoạt | Quản lý đăng nhập và bấm chọn quản lý tài khoản |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách tài khoản 2. Quản lý chọn tài khoản muốn chỉnh sửa 3. Quản lý chọn “chỉnh sửa” 4. Hệ thống hiển thị form chỉnh sửa tài khoản 5. Quản lý điền thông tin mới 6. Quản lý bấm nút “Lưu” 7. Hệ thống cập nhật thông tin tài khoản | |
| Ngoại lệ:  1. Quản lý nhập thiếu thông tin  1.1 Hệ thống hiển thị thông báo nhắc nhở   1. Quản lý nhập mật khẩu quá ngắn   2.1 Hệ thống hiển thị thông báo “Mật khẩu phải lớn hơn 8 kí tự” | |

Mô tả usecase Chỉnh sửa tài khoản

A diagram of a diagram

Description automatically generated

Biểu đồ hoạt động chức năng Chỉnh sửa tài khoản

A diagram of a diagram

Description automatically generated

Biểu đồ tuần từ chức năng Chỉnh sửa tài khoản

|  |  |
| --- | --- |
| Tên Use case | Use case xóa tài khoản |
| Tác nhân chính | Quản lý |
| Mức | 2 |
| Tiền điều kiện | Quản lý đã đăng nhập vào hệ thống |
| Đàm bảo tối thiểu | Hệ thống loại bỏ thông tin và quay lại bước trước |
| Đảm bảo thành công | Hệ thống xóa tài khoản |
| Kích hoạt | Quản lý đăng nhập và bấm chọn quản lý tài khoản |
| Chuỗi sự kiện chính:   1. Hệ thống hiển thị danh sách tài khoản 2. Quản lý chọn tài khoản muốn xóa 3. Quản lý chọn “Xóa” 4. Hệ thống hiển thị thông báo xác nhận 5. Quản lý bấm “Xác nhận” 6. Hệ thống xóa tài khoản | |
| Ngoại lệ:   1. Quản lý chọn “Hủy”   1.1 Hệ thống đóng thông báo xác nhận | |

Mô tả usecase Xóa tài khoản

A diagram of a flowchart

Description automatically generated

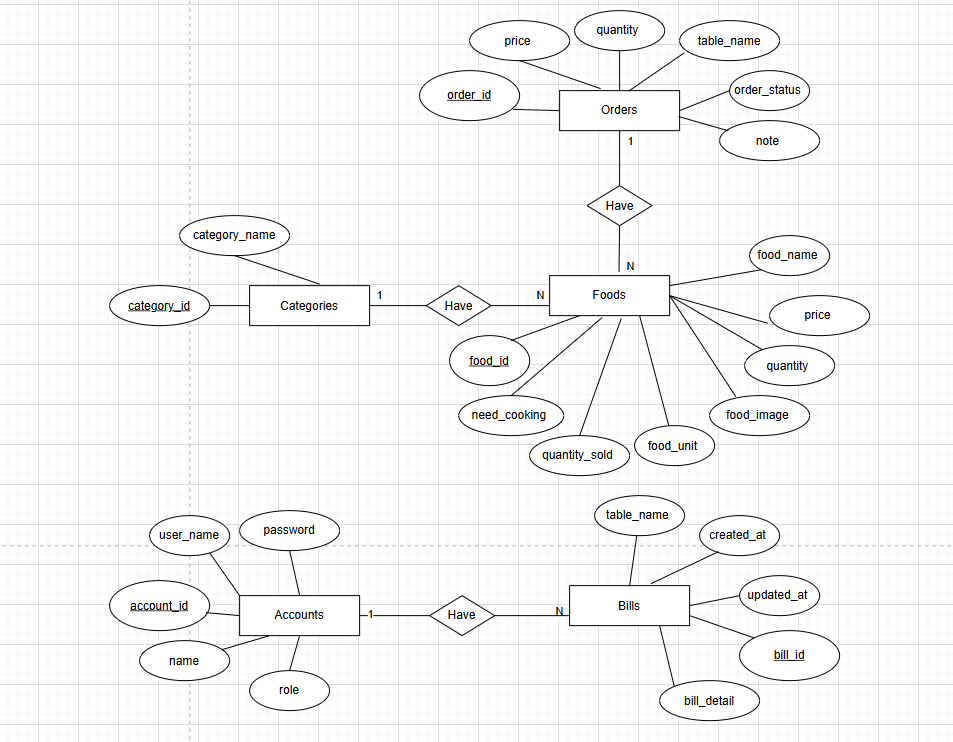
Biểu đồ hoạt động chức năng Xóa tài khoản

A diagram of a project

Description automatically generated

Biểu đồ tuần tự chức năng Xóa tài khoản

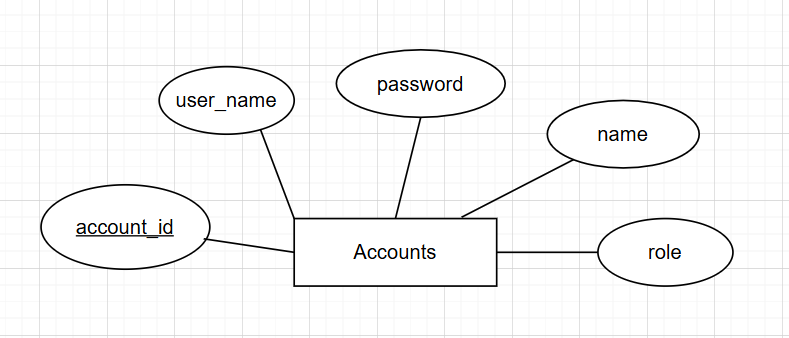
* 1. **Thiết kế cơ sở dữ liệu**
     1. **Mô hình thực thể quan hệ (Entity – Relationship) tổng quát**

****

Mô hình thưc thể liên kết tổng quát

* + 1. **Chi tiết các thực thể và thuộc tính liên quan**
       1. **Thực thể Accounts**

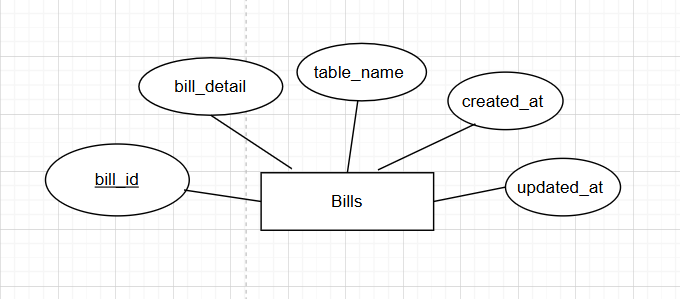
Accounts(account\_id, user\_name, password, name, role)



Thực thể Accounts

* + - 1. **Thực thể Bills**

Bills(bill\_id, bill\_detail, table\_name, created\_at, updated\_at)



Thực thể Bills

* + - 1. **Thực thể Tables**

Tables(table\_id, table\_name, table\_status)

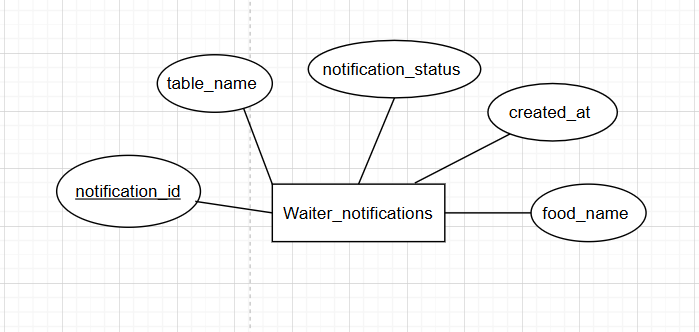
A diagram of tables with text

Description automatically generated

Thực thể Tables

* + - 1. **Thực thể Waiter\_notifications**

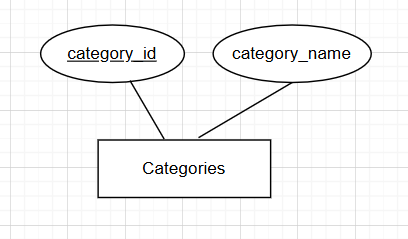
Waiter\_notification(notification\_id, table\_name, notification\_status, created\_at, food\_name)



Thực thể Waiter\_notifications

* + - 1. **Thực thể Categories**

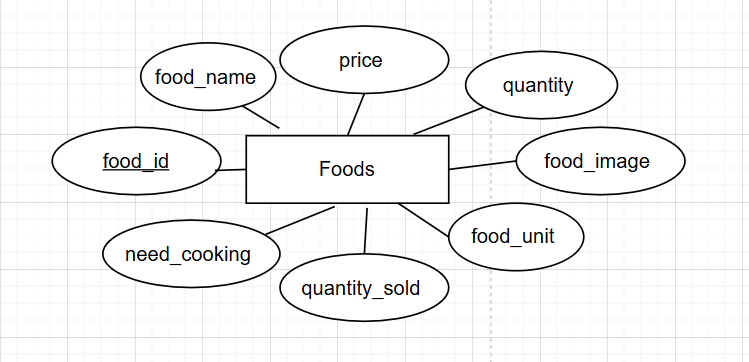
Categories(category\_id, category\_name)



Thực thể Categories

* + - 1. **Thực thể Foods**

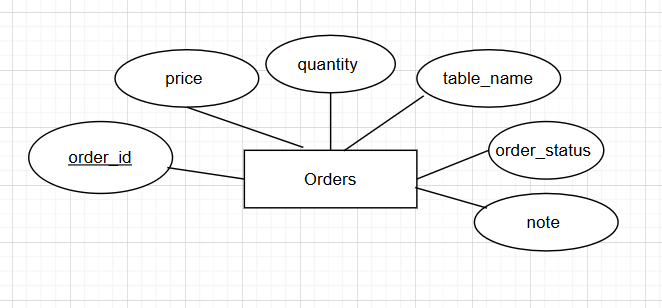
Foods(food\_id, food\_name, price, quantity, food\_image, food\_unit, quantity\_sold, need\_cooking)



Thực thể Foods

* + - 1. **Thực thể Orders**

Orders(order\_id, price, quantity, table\_name, order\_status, note)



Thực thể Orders

* + 1. **Mối quan hệ giữa các thực thể**
       1. **Mối quan hệ giữa Accounts và Bills**

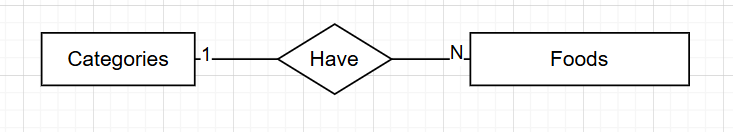
**A black and white diagram with black text

Description automatically generated**

Quan hệ giữa thực thể Account và Bills

Account – Bills là mối quan hệ 1 – N. Một Account có thể tạo nhiều Bill, một Bill chỉ có thể được tạo bởi một Acount. Khóa chính của bảng Accounts sẽ là khóa ngoại trong bản Bills

* + - 1. **Mối quan hệ giữa Categories và Foods**



Quan hệ giữa thực thể Categories và Foods

Categories – Foods là mối quan hệ 1 – N. Một Category có thể có nhiều Food, một Food chỉ thuộc về một Category. Khóa chính của bảng Categories sẽ là khóa ngoại trong bản Foods

* + - 1. **Mối quan hệ giữa Foods và Orders**

A diagram with a black rhombus

Description automatically generated

Quan hệ giữa Foods và Orders

Foods – Orders là mối quan hệ 1 – N. Một Food có thể thuộc nhiều Order, một Order chỉ có một Food. Khóa chính của bảng Foods sẽ là khóa ngoại trong bản Orders

* + 1. **Các bảng có trong cơ sở dữ liệu**
       1. **Bảng Accounts**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Accounts | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | account\_id | Int | 20 | Primary key | Mã tài khoản |
| 2 | user\_name | Char | 50 | Unique | Tên đăng nhập |
| 3 | password | Char | 50 | Not null | Mật khẩu |
| 4 | name | Char | 50 | Not null | Tên nhân viên |
| 5 | role | Char | 20 | Not null | Chức vụ |

* + - 1. **Bảng Bills**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bills | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | bill\_id | Int | 50 | Primary key | Mã hóa đơn |
| 2 | account\_id | Char | 20 | Foreign key | Mã nhân viên |
| 3 | bill\_detail | Text |  | Not null | Chi tiết hóa đơn |
| 4 | Table\_name | Char | 20 | Not null | Tên bàn |
| 5 | created\_at | Datetime |  | Not null | Ngày tạo |
| 6 | updated\_at | Datetime |  |  | Ngày cập nhật |

* + - 1. **Bảng Tables**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tables | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | table\_id | Int | 20 | Primary key | Mã bàn |
| 2 | table\_name | Char | 20 | Unique, Not null | Tên bàn |
| 3 | table\_status | Char | 20 | Not null | Trạng thái bàn |

* + - 1. **Bảng Waiter\_notifications**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Waiter\_notifications | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | notification\_id | Int | 20 | Primary key | Mã thông báo |
| 2 | table\_name | Char | 20 | Not null | Tên bàn |
| 3 | notification\_status | Char | 20 | Not null | Trạng thống báo |
| 4 | created\_at | Datetime |  | Not null | Ngày tạo |
| 5 | food\_name | Char | 50 |  | Tên mặt hàng |

* + - 1. **Bảng Categories**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Categories | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | category\_id | Int | 20 | Primary key | Mã danh mục |
| 2 | category\_name | Char | 50 | Not null | Tên danh mục |

* + - 1. **Bảng Foods**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Foods | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | food\_id | Int | 20 | Primary key | Mã mặt hàng |
| 2 | category\_id | Char | 20 | Foreign key | Mã danh mục |
| 3 | food\_name | Char | 50 | Not null | Tên mặt hàng |
| 4 | price | Int | 10 | Not null | Giá bán |
| 5 | quantity | Int | 10 | Not null | Số lượng |
| 6 | food\_image | Char | 50 |  | Ảnh mặt hàng |
| 7 | food\_unit | Char | 50 |  | Đơn vị |
| 8 | need\_cooking | Int | 1 | Default: 1 | Cần chế biến |
| 9 | quantity\_sold | Int | 20 | Default: 0 | Số lượng đã bán |

* + - 1. **Bảng Orders**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Orders | | | | | |
| STT | Tên trường | Kiểu | Độ dài | Ghi chú | Mô tả |
| 1 | order\_id | Int | 50 | Primary key | Mã đơn đặt món |
| 2 | food\_id | Int | 20 | Foreign key | Mã mặt hàng |
| 3 | price | Int | 10 | Not null | Giá bán |
| 4 | quantity | Int | 10 | Not null | Số lượng |
| 5 | table\_name | Char | 20 | Not null | Tên bàn |
| 6 | order\_status | Char | 20 | Not null | Trạng thái đơn đặt món |
| 7 | Note | Char | 50 |  | Ghi chú |

* + 1. **Mô hình cơ sở dữ liệu**

A screenshot of a computer program

Description automatically generated

Mô hình cơ sở dữ liệu

1. **Xây dựng chương trình**
   1. **Công nghệ sử dụng**
      1. **Ngôn ngữ lập trình Dart**

Dart là một ngôn ngữ lập trình tối ưu cho phía client, được phát triển để xây dựng các ứng dụng nhanh trên mọi nền tảng. Mục tiêu của nó là cung cấp ngôn ngữ lập trình hiệu quả nhất cho phát triển đa nền tảng, kết hợp với nền tảng runtime linh hoạt cho các framework ứng dụng.

Ngôn ngữ lập trình được định hình bởi các lựa chọn kỹ thuật trong quá trình phát triển, những quyết định này sẽ ảnh hưởng đến khả năng và sức mạnh của ngôn ngữ đó. Dart được thiết kế với mục tiêu kỹ thuật đặc biệt phù hợp cho phát triển phía client, ưu tiên cả trải nghiệm phát triển (tải lại trạng thái nhanh chỉ trong vài giây) và chất lượng sản phẩm cao trên nhiều mục tiêu biên dịch khác nhau (web, di động và máy tính để bàn).

Dart cũng là nền tảng của Flutter. Dart cung cấp ngôn ngữ và runtime để vận hành các ứng dụng Flutter, nhưng nó cũng hỗ trợ nhiều tác vụ quan trọng khác của lập trình viên như định dạng, phân tích và kiểm thử mã nguồn.

Ngôn ngữ Dart đảm bảo an toàn kiểu; nó sử dụng kiểm tra kiểu tĩnh để đảm bảo giá trị của một biến luôn phù hợp với kiểu tĩnh của biến đó. Điều này đôi khi được gọi là sound typing. Mặc dù các kiểu là bắt buộc, nhưng việc chú thích kiểu là tùy chọn nhờ vào khả năng suy luận kiểu. Hệ thống kiểu của Dart cũng linh hoạt, cho phép sử dụng kiểu dynamic kết hợp với kiểm tra thời gian chạy, điều này có thể hữu ích trong quá trình thử nghiệm hoặc đối với mã cần đặc biệt linh hoạt.

Dart có tính năng sound null-safety tích hợp. Điều này có nghĩa là các giá trị không thể là null trừ khi lập trình viên cho phép chúng. Với sound null-safety, Dart có thể bảo vệ mã nguồn khỏi các ngoại lệ null tại thời gian chạy thông qua phân tích mã tĩnh. Không giống như nhiều ngôn ngữ null-safe khác, khi Dart xác định rằng một biến không thể là null, biến đó không bao giờ có thể là null. Nếu kiểm tra mã đang chạy trong trình gỡ lỗi, sẽ thấy rằng tính không thể null được giữ nguyên tại thời gian chạy; do đó được gọi là sound null-safety. [[Dart overview | Dart](https://dart.dev/overview)]

Lịch sử phát triển của Dart:

Ngôn ngữ lập trình Dart được phát triển bởi Google từ ngày 10 tháng 10 năm 2011.

Cú pháp của Dart tương tự như C, nghĩa là các lập trình viên quen thuộc với C++, Java và JavaScript sẽ cảm thấy quen thuộc. Ngoài ra, các nhà phát triển Dart đã thêm vào các tính năng như generics, kiểu tĩnh, v.v. Dart được phát triển để giải quyết các vấn đề của JavaScript. Có vẻ như các kỹ sư của Google đã gặp khó khăn khi duy trì các mã JavaScript lớn cho Gmail và Google Maps, và đó là lý do họ bắt đầu làm việc trên ngôn ngữ thay thế. Khi Dart mới được phát triển, nó chưa được sử dụng rộng rãi.

Tuy nhiên, Dart đang dần trở nên phổ biến nhờ sự xuất hiện của Flutter, framework UI của Google dành cho việc xây dựng giao diện gốc trên iOS và Android. Hiện nay, Dart là một trong những ngôn ngữ lập trình được ưa chuộng nhất.



Hình. Logo ngôn ngữ lập trình Dart

* + 1. **Framework Flutter**

Flutter là một framework giao diện người dùng di động miễn phí và mã nguồn mở được tạo bởi Google và ra mắt vào tháng 5 năm 2017. Nói một cách đơn giản, Flutter cho phép tạo ứng dụng di động gốc chỉ với một mã nguồn duy nhất. Điều này có nghĩa là chỉ cần sử dụng một ngôn ngữ lập trình và một mã nguồn duy nhất để tạo hai ứng dụng khác nhau (cho iOS và Android).

Flutter bao gồm hai phần quan trọng:

* **SDK (Bộ công cụ phát triển phần mềm):** Một tập hợp các công cụ giúp phát triển ứng dụng, bao gồm các công cụ để biên dịch mã nguồn thành mã máy gốc (cho iOS và Android).
* **Framework (Thư viện UI dựa trên widgets):** Một tập hợp các thành phần giao diện người dùng có thể tái sử dụng (nút bấm, ô nhập liệu, thanh trượt, v.v.) mà có thể tùy chỉnh theo nhu cầu riêng.

Việc sử dụng Flutter giúp đơn giản hóa quá trình phát triển ứng dụng di động, tiết kiệm thời gian và tài nguyên bằng cách tận dụng mã nguồn chung cho cả hai nền tảng iOS và Android. [[The History And Rules of Dart Language | by Shota Nukumizu | Medium](https://medium.com/@author2000.1225/the-history-and-rules-of-dart-language-f25e09a58530)]

Trong Flutter, widgets là những khối xây dựng cơ bản và là phương tiện chính để xây dựng giao diện người dùng. Widgets định nghĩa diện mạo của ứng dụng, dựa trên trạng thái hiện tại và cấu hình của chúng.

Mọi thứ trong Flutter, từ các nút bấm và ô nhập liệu đơn giản đến các bố cục và hoạt ảnh phức tạp, đều được biểu diễn dưới dạng widgets. Có rất nhiều widgets để lựa chọn, và chúng có thể được phân loại thành 14 nhóm, mỗi nhóm phục vụ các vai trò khác nhau trong ứng dụng Flutter.

* **Accessibility Widgets**: cải thiện khả năng truy cập của ứng dụng, làm cho nó thân thiện hơn với người dùng.
* **Animation and Motion Widgets**: mang các yếu tố động vào ứng dụng bằng cách tích hợp hoạt ảnh vào các widgets khác.
* **Assets, Images, and Icons Widgets**: quản lý các tài sản hình ảnh như hình ảnh và biểu tượng.
* **Async Widgets**: hỗ trợ các hoạt động bất đồng bộ trong ứng dụng Flutter.
* **Basic Widgets**: các widgets cơ bản cần thiết cho việc phát triển bất kỳ ứng dụng Flutter nào.
* **Cupertino Widgets**: các widgets theo phong cách iOS.
* **Input Widgets**: xử lý đầu vào của người dùng trong ứng dụng Flutter.
* **Interaction Models Widgets**: quản lý các sự kiện cảm ứng và điều hướng giữa các view khác nhau trong ứng dụng.
* **Layout Widgets**: hỗ trợ sắp xếp các widgets khác trên màn hình theo nhu cầu.
* **Material Components Widgets**: tuân theo các hướng dẫn thiết kế material của Google.
* **Painting and Effects Widgets**: áp dụng các hiệu ứng hình ảnh cho các widget con mà không thay đổi bố cục hay hình dạng của chúng.
* **Scrolling Widgets**: cung cấp chức năng cuộn cho các widgets không có khả năng cuộn mặc định.
* **Styling Widgets**: quản lý chủ đề, khả năng đáp ứng và kích thước của ứng dụng.
* **Text Widgets**: hiển thị văn bản.

Ngoài các nhóm trên, widgets còn được chia thành hai loại chính:

* **StatelessWidget**: là widget bất biến, không thay đổi trạng thái nội bộ sau khi được tạo. Nó đại diện cho các thành phần giao diện người dùng không yêu cầu cập nhật động. StatelessWidget thường được sử dụng cho các yếu tố tĩnh như nhãn văn bản, nút bấm, biểu tượng hoặc hình ảnh. StatelessWidget được định nghĩa bằng cách triển khai phương thức build của widget, phương thức này trả về một cây widget biểu diễn cấu trúc của thành phần giao diện người dùng.
* **StatefulWidget**: là widget có thể thay đổi trạng thái nội bộ theo thời gian. Nó đại diện cho các thành phần giao diện người dùng yêu cầu cập nhật động hoặc xử lý tương tác người dùng. StatefulWidget bao gồm hai lớp: lớp widget (không thay đổi) và lớp trạng thái tương ứng chứa trạng thái có thể thay đổi. Lớp trạng thái chịu trách nhiệm cập nhật widget khi có sự thay đổi trạng thái. Phương thức build của StatefulWidget được gọi mỗi khi widget cần được xây dựng lại do sự thay đổi trạng thái.

A logo with a checkered background

Description automatically generated

Hình. Logo Framework Flutter

Trong Flutter, Widget Tree là cấu trúc phân cấp của các widgets, tương tự như cây DOM trong phát triển web. Mỗi widget trong Flutter đại diện cho một phần tử giao diện người dùng và có thể chứa các widgets con, tạo nên một cấu trúc cây. Cây widget này xác định cách các thành phần giao diện được sắp xếp và hiển thị trên màn hình.

Widget Tree bắt đầu từ widget gốc, thường là MaterialApp hoặc CupertinoApp, và từ đó phân nhánh thành các widgets con như Scaffold, AppBar, Text, Button, và nhiều loại khác. Mỗi widget trong cây có thể có các thuộc tính và trạng thái riêng, và chúng tương tác với nhau để tạo ra giao diện người dùng hoàn chỉnh.

Cấu trúc phân cấp của Widget Tree giúp quản lý và tổ chức mã nguồn dễ dàng hơn. Khi cần thay đổi giao diện, chỉ cần cập nhật widget tương ứng trong cây. Điều này cũng cho phép Flutter thực hiện việc cập nhật giao diện hiệu quả hơn, vì chỉ những phần của cây bị thay đổi mới cần được vẽ lại.

Một điểm mạnh của Widget Tree là khả năng tái sử dụng các widgets. Các widgets có thể được tạo ra và sắp xếp lại một cách dễ dàng, giúp phát triển ứng dụng nhanh chóng và linh hoạt. Bằng cách tận dụng cấu trúc phân cấp của Widget Tree, các nhà phát triển có thể tạo ra các giao diện phức tạp một cách có tổ chức và dễ bảo trì.

A diagram of a website

Description automatically generated with medium confidence

Hình. Mô hình cây widgets trong Flutter

* + 1. **Ngôn ngữ lập trình PHP**

PHP là một ngôn ngữ lập trình mã nguồn mở phía máy chủ, được sử dụng để tạo các trang web, ứng dụng, hệ thống quản lý quan hệ khách hàng và nhiều ứng dụng khác. Đây là một ngôn ngữ đa mục đích được sử dụng rộng rãi và có thể nhúng vào HTML, giúp đơn giản hóa mã HTML và duy trì sự phổ biến với các nhà phát triển.

PHP là viết tắt của 'PHP: Hypertext Preprocessor', với PHP ban đầu là 'Personal Home Page'. Tên viết tắt đã thay đổi khi ngôn ngữ phát triển kể từ khi ra mắt vào năm 1994 để phản ánh chính xác hơn bản chất của nó.

Tính đến năm 2022, đã có 8 phiên bản của PHP, với phiên bản 8.1 hiện đang là lựa chọn phổ biến cho các trang web. Lập trình PHP có thể được sử dụng để tạo hầu hết các ứng dụng mà một nhà phát triển phần mềm cần. Tuy nhiên, có ba lĩnh vực chính mà PHP phát triển mạnh mẽ:

* **Server-side scripting**: Lập trình kịch bản phía máy chủ là thế mạnh chính của PHP. Để bắt đầu với lập trình kịch bản phía máy chủ bằng PHP, cần có một PHP parser, web server và trình duyệt web.
* **Command-line scripting**: Lập trình kịch bản dòng lệnh lý tưởng cho các kịch bản được tạo bằng cron (Linux) hoặc Task Scheduler (Windows) và rất hữu ích cho việc xử lý văn bản đơn giản.
* **Viết ứng dụng máy tính để bàn**: Mặc dù không phải là ngôn ngữ tốt nhất để tạo ứng dụng máy tính để bàn, nhưng PHP cung cấp nhiều tùy chọn hơn cho các nhà phát triển web nâng cao so với các đối thủ cạnh tranh.

PHP cũng xuất sắc trong việc thu thập dữ liệu từ biểu mẫu, mã hóa dữ liệu người dùng và gửi cũng như nhận cookie. Một trong những tính năng chính của PHP làm cho nó rất hữu dụng là nó tương thích với tất cả các hệ điều hành chính, cho phép viết mã bất kể công nghệ nào đang được sử dụng.

PHP đã trở thành một ngôn ngữ lập trình phổ biến trong gần ba thập kỷ nhờ vào những lợi ích đáng kể mà nó mang lại cho người dùng và các nhà phát triển. Các lợi ích chính của PHP bao gồm:

* **Đa nền tảng**: PHP có khả năng chạy trên mọi hệ điều hành chính, giúp loại bỏ lo ngại về tương thích. Điều này cũng mang lại lợi ích làm việc hiệu quả trong các nhóm dự án mà không cần phải lo lắng về việc các thành viên có thể truy cập mã nguồn hay không.
* **Mã nguồn mở**: Với tính chất mã nguồn mở, PHP là một ngôn ngữ lập trình miễn phí, giúp giảm chi phí xây dựng dự án.
* **Sự hiểu biết rộng rãi của cộng đồng**: Với sự tồn tại lâu dài, PHP đã tạo ra một cộng đồng lớn có sẵn để hỗ trợ và sử dụng lượng mã nguồn có sẵn để khởi đầu các dự án.
* **Tương thích tốt với HTML**: PHP hỗ trợ tốt trong việc tương tác với HTML, giúp đơn giản hóa quy trình phát triển dự án.
* **Dễ học và nắm bắt**: Với thiết kế đơn giản, PHP là ngôn ngữ dễ học và nhanh chóng tiếp cận hơn so với các ngôn ngữ khác.
* **Các công cụ hỗ trợ phong phú**: Có nhiều công cụ hỗ trợ cho PHP, từ tích hợp, gợi ý mã, làm nổi bật cú pháp và nhiều tính năng khác, mang đến sự hỗ trợ đáng kể cho các nhà phát triển.
* **Thời gian tải trang tuyệt vời**: Với yêu cầu hiệu suất trang web ngày càng cao, PHP mang lại khả năng tải trang nhanh chóng, giúp tối ưu hóa trải nghiệm người dùng và thành công của dự án.

Nhờ những lợi ích này, PHP vẫn là một lựa chọn hàng đầu cho nhiều nhà phát triển trong việc xây dựng các ứng dụng và trang web hiện đại.

[[What is PHP? Uses & Introduction - Code Institute Global](https://codeinstitute.net/global/blog/what-is-php-programming/)]



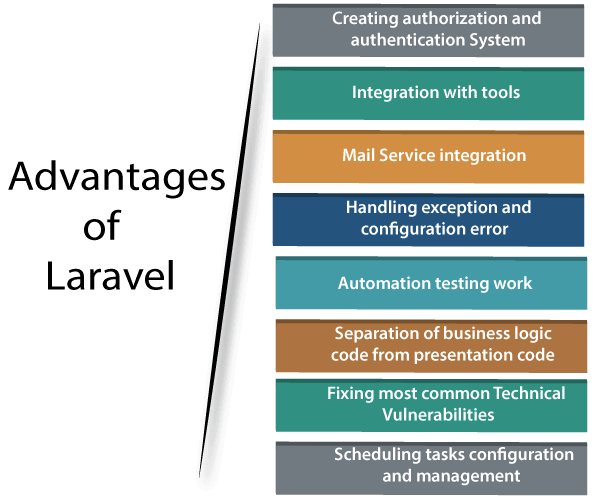
Hình . Logo ngôn ngữ lập trình PHP

* + 1. **Framework Laravel**

Laravel là một framework PHP mã nguồn mở, mạnh mẽ và dễ hiểu. Nó tuân thủ mô hình thiết kế model-view-controller (MVC). Laravel tái sử dụng các thành phần hiện có từ các framework khác, giúp tạo ra các ứng dụng web có cấu trúc rõ ràng và thực tế hơn.

Laravel cung cấp một bộ tính năng phong phú, kết hợp các tính năng cơ bản của các framework PHP như CodeIgniter, Yii và các ngôn ngữ lập trình khác như Ruby on Rails. Laravel có một bộ tính năng rất phong phú giúp tăng tốc độ phát triển web.

Nếu lập trình viên đã quen thuộc với Core PHP và Advanced PHP, Laravel sẽ giúp công việc của họ dễ dàng hơn. Nó tiết kiệm rất nhiều thời gian nếu lập trình viên có ý định phát triển một trang web từ đầu. Hơn nữa, một trang web được xây dựng bằng Laravel là an toàn và ngăn ngừa nhiều cuộc tấn công web.

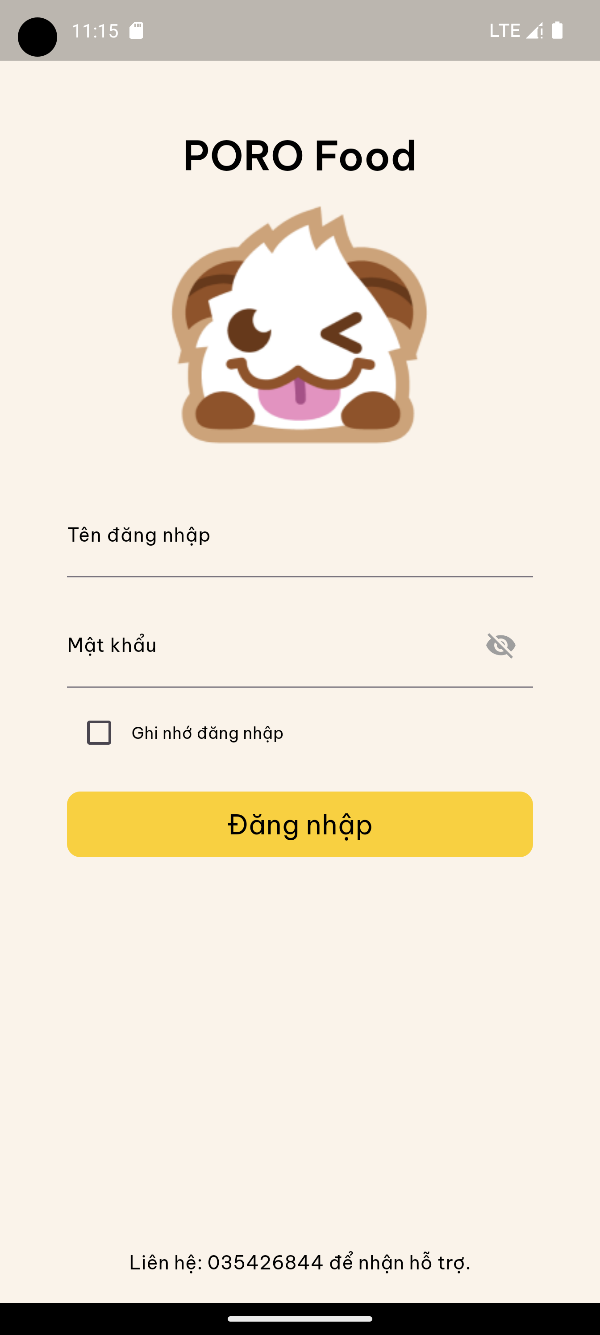
****

Hình . Ưu điểm của Laravel

* + 1. **Laravel Reverb**

Laravel Reverb là một máy chủ WebSocket được xây dựng cho ứng dụng Laravel, cho phép giao tiếp thời gian thực giữa máy khách và máy chủ. Dưới đây là một số ưu điểm của Laravel Reverb:

* **Tích hợp dễ dàng**: Laravel Reverb được tích hợp sẵn với Laravel, điều này giúp việc triển khai và sử dụng trở nên đơn giản và thuận tiện.
* **Giao tiếp thời gian thực**: Nó cho phép các ứng dụng Laravel truyền và nhận dữ liệu trong thời gian thực qua WebSocket, cung cấp trải nghiệm người dùng tốt hơn trong việc cập nhật và đồng bộ dữ liệu.
* **Hiệu suất cao**: Với việc sử dụng WebSocket, Laravel Reverb giúp giảm thiểu độ trễ và tối ưu hóa hiệu suất của ứng dụng, đặc biệt là trong các ứng dụng yêu cầu giao tiếp thời gian thực.
* **Đơn giản và linh hoạt**: Laravel Reverb cung cấp các API và công cụ linh hoạt để quản lý kết nối WebSocket và xử lý dữ liệu thời gian thực, đồng thời giúp phát triển ứng dụng một cách dễ dàng và hiệu quả.
* **Mã nguồn mở và hỗ trợ từ cộng đồng**: Với tính chất mã nguồn mở, Laravel Reverb nhận được sự hỗ trợ từ cộng đồng Laravel rộng lớn, giúp cải thiện và phát triển liên tục tính năng và sửa lỗi.
  1. **Giao diện chương trình**
     1. **Giao diện đăng nhập**

****

Giao diện chức năng đăng nhập

* + 1. **Giao diện chức năng đổi mật khẩu**

****

Hình . Giao diện đổi mật khẩu

* + 1. **Giao diện chức trang chủ (phục vụ)**

****

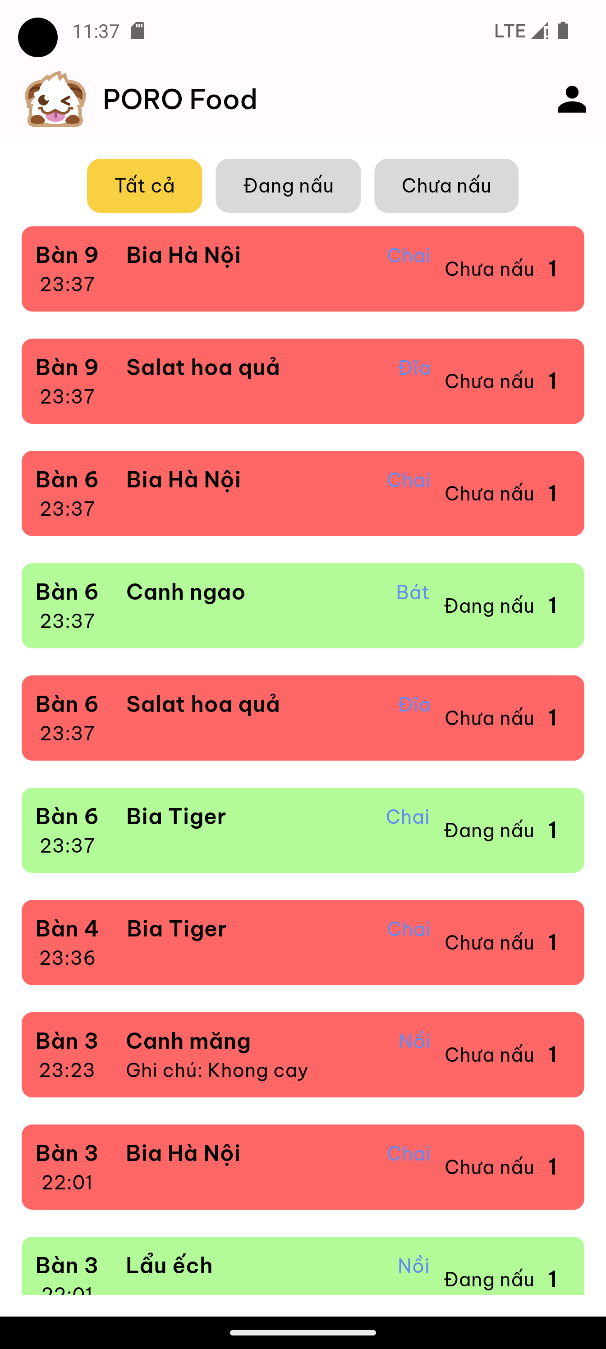
Hình . Giao diện trang chủ nhân viên phục vụ

* + 1. **Giao diện chức năng chọn món**



Hình . Giao diện chọn món

* + 1. **Giao diện trang chủ (nhân viên bếp)**



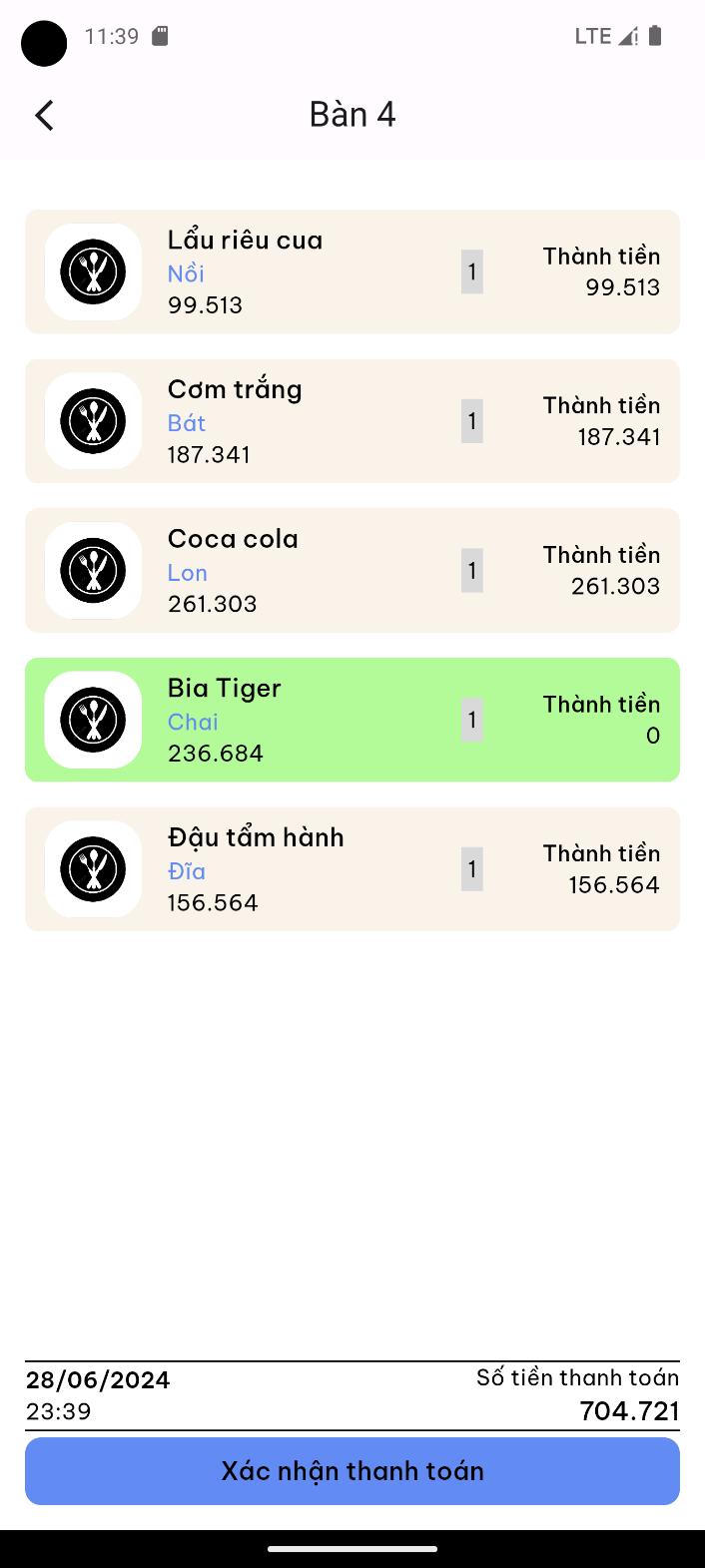
Hình. Giao diện nhân viên bếp

* + 1. **Giao diện trang chủ (thu ngân)**



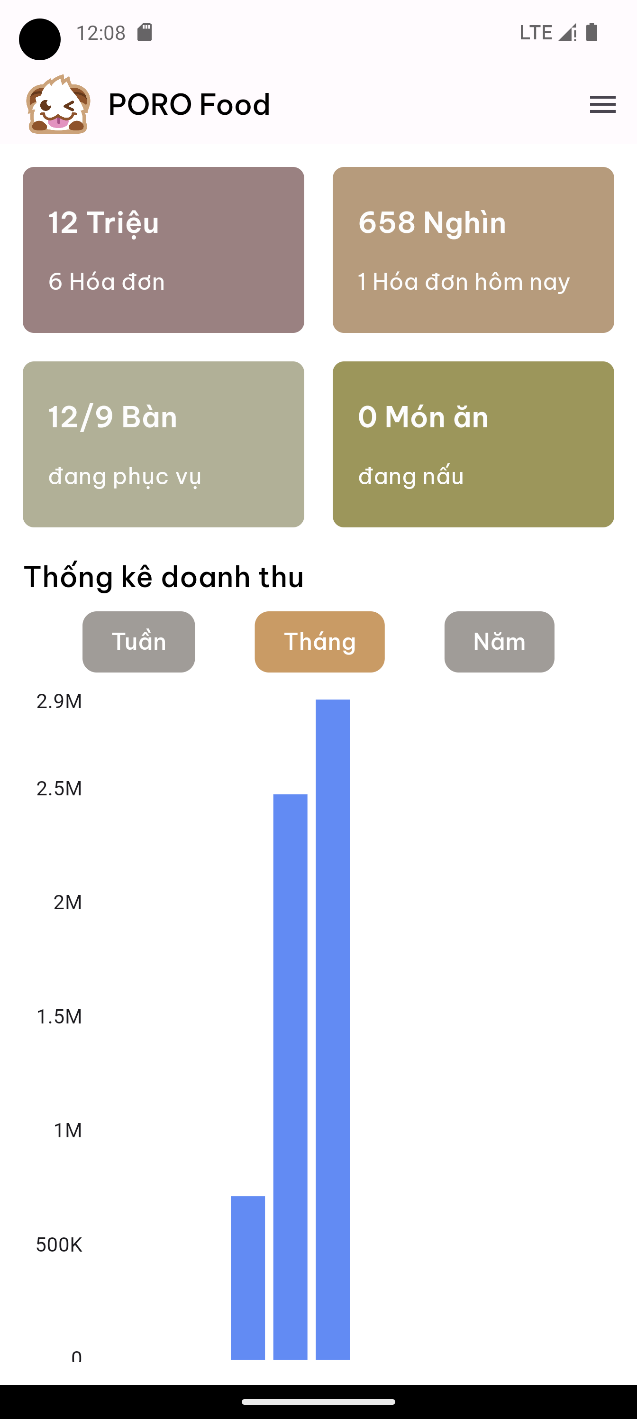
Hình . Giao diện trang chủ (thu ngân)

* + 1. **Giao diện thanh toán hóa đơn**



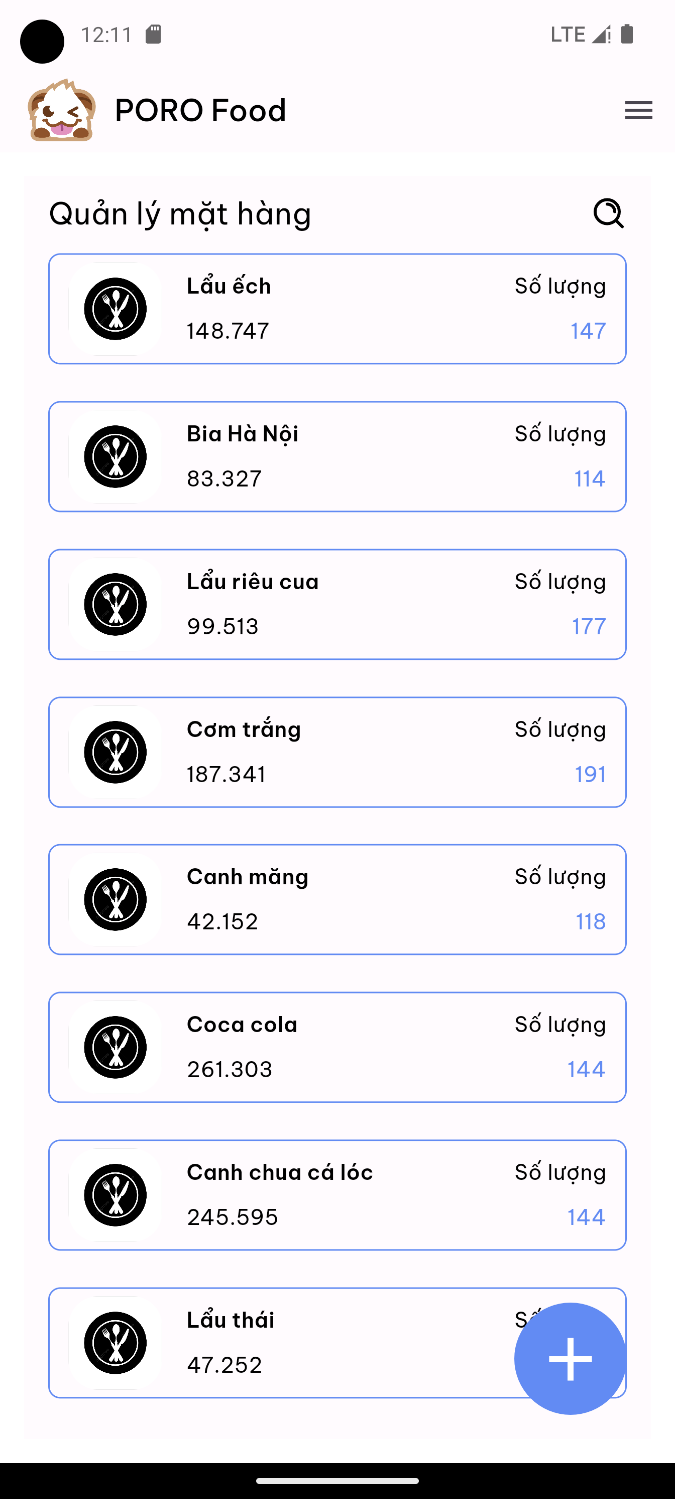
Hình . Giao diện thanh toán hóa đơn

* + 1. **Giao diện trang chủ quản lý**



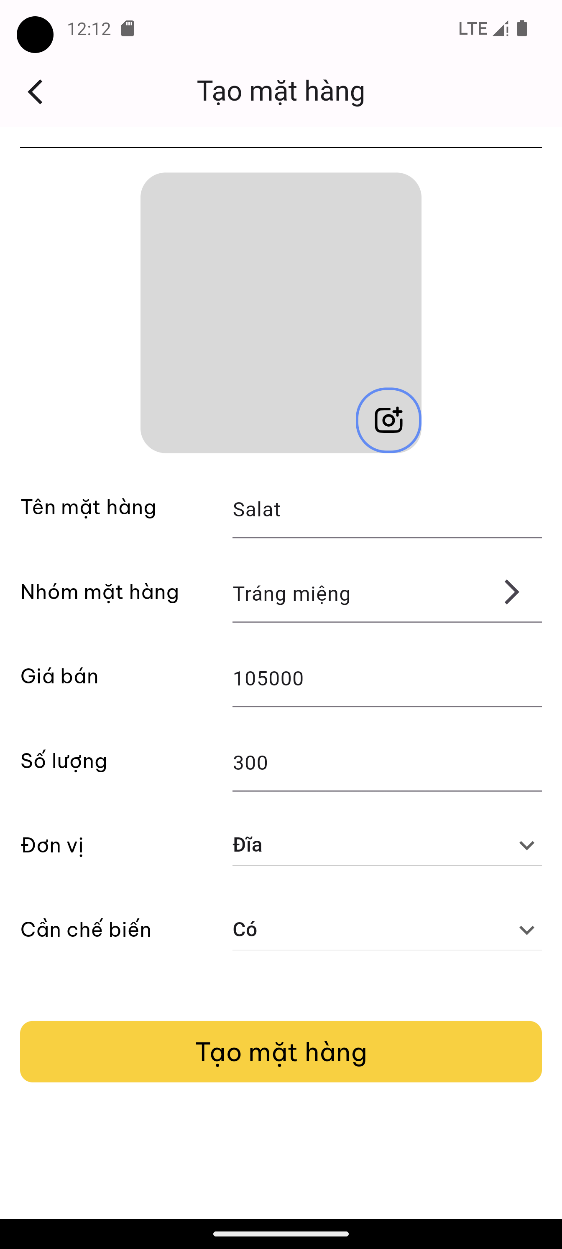
Hình . Giao diện trang chủ quản lý

* + 1. **Giao diện quản lý mặt hàng**

****

Hinh . Giao diện quản lý mặt hàng

* + 1. **Giao diện thêm mặt hàng**



Hình . Giao diện thêm mặt hàng