TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

VIỆN KỸ THUẬT CÔNG NGHỆ



**ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ ỨNG DỤNG DI ĐỘNG**

Đề tài

**THIẾT KẾ ỨNG DỤNG DI ĐỘNG**

**QUẢN LÍ QUÁN CAFÉ**

**RABBIT’S HOUSE**

GVHD: Ths NGUYỄN KIM DUY

SVTH: Nguyễn Anh Hào

1824801030064

D18PM01

Trần Tiến Đạt

1824801030161

D18PM03

Tháng 12/2020

**VIỆN KỸ THUẬT – CÔNG NGHỆ**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

Họ và tên giảng viên: **Nguyễn Kim Duy**

Tên đề tài: **QUẢN LÍ QUÁN CAFE RABBIT’S HOUSE**

Nội dung nhận xét:

**Điểm:**

Bằng số:

Bằng chữ:

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN CHẤM**  *(Ký, ghi rõ họ tên)*  **NGUYỄN KIM DUY** |
|  |  |

MỤC LỤC

**THÀNH VIÊN NHÓM VÀ VAI TRÒ**

**Nhóm: 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ Tên** | **Vai Trò** | **Nội dung thực hiện** |
| 1 | 1824801030161 | Trần Tiến Đạt | Nhóm Trưởng | * Tổng hợp * Hoàn tất chương trình |
| 2 | 1824801030064 | Nguyễn Anh Hào | Thành Viên | * Thiết kế giao diện * Xây khung chương trình |

# **LỜI MỞ ĐẦU**

Trong cuộc sống hiện nay, không khó để có thể tìm kiếm một quán nước, quán café, ngoài nhu cầu giải khát, thưởng thức những món nước lạ thì còn có thể trò chuyện cùng bạn bè, gia đình, việc học nhóm tại quán café cũng rất quen thuộc đối với các bạn trẻ. Và để đáp ứng nhu cầu đó của con người thì ngày nay việc kinh doanh quán café không còn quá xa lạ với mọi người. Các quán café mọc lên như nấm, đủ thể loại, đủ mọi hình thức thu hút khách hàng. Ngoài ra, thức uống cũng là một phần quan trọng dẫn đến lượng khách nhiều hay ít. Có nhiều yếu tố để phát triển một quán café nhưng bên cạnh đó, yếu tố quản lý quán café như thế nào? Làm thế nào để quản lý một cách nhanh chóng và ít tốn thời gian? Phần mềm quản lý quán café ra đời nhằm đáp ứng cho những câu hỏi trên và giúp cho người sử dụng đỡ tốn công thức và thời gian.

Phần mềm được hình thành nhờ các công cụ hỗ trợ như: Draw.io được sử dụng để thiết kế mô hình dữ liệu, áp dụng hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2019 để trao đổi dữ liệu và công cụ phát triển phần mềm Microsoft Visual Studio 2019.

Tài nguyên đồ án:

**CHƯƠNG 1. TỔNG QUAN VỀ ỨNG DỤNG**

1. Định nghĩa bài toán
2. **Tổng quan về tính hình nghiên cứu**
   1. **Kinh doanh thương mại là gì?**

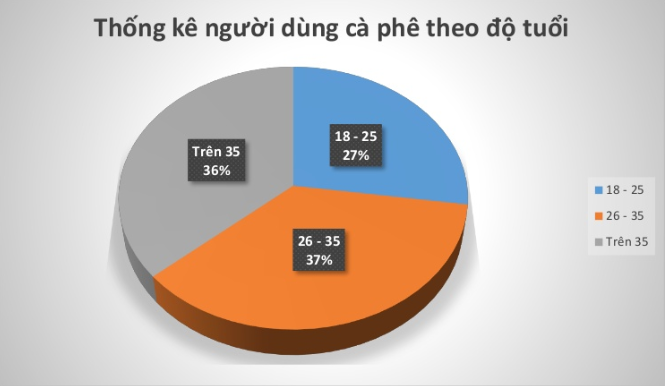
Một trong những ngành nghề đang hấp dẫn nhất hiện nay, đang ngày một phát triển, giúp đem lại nguồn thu lớn cho mỗi cá nhân, tổ chức doanh nghiệp. Ngành nghề kinh doanh rất đa dạng: kinh doanh bất động sản, chứng khoáng, kinh doanh các loại dịch vụ, kinh doanh shop quần áo, quán cà phê…

Kinh doanh thương mại là hoạt động trao dồi, mua bán sản phẩm hàng hóa vật chất trong nền kinh tế tạo tiền đề và cơ hội cho sự hình thành và phát triền. Kinh doanh thương mại là sự đầu tư tiền của, công sức của một cá nhân hay một tổ chức vào việc mua bán hàng hóa và kiếm lời. Kinh doanh thương mại trong thời buổi ngày nay thì có rất nhiều cơ hội nhưng cũng có rất nhiều rủi ro. Vì vậy dù hoạt động như thế nào thì các doanh nghiệp cũng phải đảm bảo an toàn trong kinh doanh. Để đảm bảo an toàn thì doanh nghiệp phải giữ được mức ổn định trong việc tăng doanh thu qua các năm.Do vậy hoạt động thúc đẩy bán hàng nó quyết định đến việc tăng doanh thu và đảm bảo an toàn trong kinh doanh của các doanh nghiệp. Thị trường luôn luôn biến động, thay đổi không ngừng vì thế bán hàng không còn là vấn đề mới mẻ nhưng nó luôn mang tính thời sự cấp bách, và là mối quan tâm hàng đầu của các doanh nghiệp.

* 1. **Kinh doanh quán cafe**

Trong cuộc sống hiện nay, không khó để có thể tìm kiếm một quán nước, quán coffee, ngoài nhu cầu giải khát, thưởng thức những món nước lạ thì còn có thể trò chuyện cùng bạn bè, gia đình, việc học nhóm tại quá coffee cũng rất quen thuộc đối với các bạn trẻ. Và để đáp ứng nhu cầu đó của con người thì ngày nay việc kinh doanh quán coffee không còn quá xa lạ với mọi người. Các quán cà phê mọc lên như nấm, đủ thể loại, đủ hình thức thu hút khách hàng. Ngoài ra, thức uống cũng là một phần quan trọng dẫn đến lượng khách đến quán nhiều hay ít.

Lâu nay cà phê là thức uống dành cho người đi làm, dân công sở nhưng dạo gần đây thì cà phê được giới trẻ yêu thích không kém. Sự du nhập của cà phê đã hình thành một sở thích uống cà phê mới trong giới trẻ.

  
 Biểu đồ 1-1 Thống kê lượng người dùng cà phê theo độ tuổi

1. **Lí do chọn đề tài**

Việc mở một quán cà phê đòi hỏi phải có nhiều yếu tố: tài chính, vật chất, địa lý thuận lợi để giúp cho việc buôn bán phát triển. Bên cạnh đó thì việc quản lý quán cà phê của mình như thế nào cũng là một câu hỏi lớn cần giải quyết.Vậy việc quản lý quán cà phê như thế nào và bằng cách gì? Quản lý về cái gì? Quản lý như thế nào được gọi là tiện lợi? Ít tốn công sức nhưng lại cho ra kết quả một cách hiệu quả.

Đó là một trong những lý do lớn nhất việc các phần mềm quản lý bán hàng ra đời và trả lời cho những câu hỏi trên nhằm đáp ứng cho việc kinh doanh của cá nhân hoặc một tổ chức nào đó thuận lợi và tiết kiệm thời gian hơn cho việc quản lý “đứa con tinh thần” của mình.

1. **Mục tiêu nghiên cứu**

Phần mềm hỗ trợ trong việc quản lý các nghiệp vụ như tính tiền, in hóa đơn, thống kê, quản lý nhân viên, lương bổng…v.v. Phần mềm quản lý bán hàng quán cà phê giúp cho người sử dụng tiết kiệm được thời gian, nhanh chóng, thao tác đơn giản, dễ dàng sử dụng và quản lý.

* Xây dựng một hệ thống mới phù hợp, dể hiểu, dể sử dụng cho người dùng phần mềm. Hiểu được hệ thống quản lý bán hàng, cần quản lý những mảng nào để xây dựng ứng dụng cho phù hợp nhu cầu thực tiễn.
* Giúp quản lý hệ thống của quán trở nên tối ưu hơn, công tác quản lý dễ dàng và tiện lợi hơn.
* Giúp bản thân có thêm khả năng sáng tạo, tư duy thông qua thiết kế giao diện tương tác với người dùng, vận dụng kiến thức bản thân vào thực tiễn.

1. **Nhiệm vụ nghiên cứu**

Phần mềm này được tạo ra nhằm hy vọng đem đến sự thuận tiện cho người sử dụng cũng như giúp cho các chủ quán có thể xem xét một cách tổng quát tình hình kinh doanh của quán mình.

Phần mềm còn mang lại lợi ích kinh tế khi là giải pháp giúp việc quản lý trở nên đơn giản và thân thiện với mọi người.

1. **Đối tượng và khách thể nghiên cứu**

* Đối tượng nghiên cứu: Phần mềm quản lý café Rabbit’s House.
* Khách thể nghiên cứu:
* Những ngưởi có nhu cầu kinh doanh quán cà phê cần phần mềm quản lý quán.
* Nghiệp vụ bán hàng.
* Quy trình hoạt động của một quán cà phê thông thường.

1. **Phương tiện nghiên cứu**

* Nghiên cứu cơ sở lý thuyết về phân tích và thiết kế hệ thống thông tin.
* Hệ quản trị CSDL và ngôn ngữ lập trình Java.
* Sử dụng các công cụ như Eclipse để lập trình, thiết kế giao diện.
* Sử dụng Draw.io để vẽ các sơ đồ chức năng.

1. **Phương pháp và phạm vi nghiên cứu**
   1. **Phương pháp nghiên cứu**
   2. ***Phương pháp nghiên cứu tài liệu***

* Tham khảo, nghiên cứu thêm các tài liệu trên trang web, trang báo, tìm hiểu tình hình thực trạng của đề tài.
* Tìm xem các đồ án, khóa luận tốt nghiệp của các anh chị khóa trước đã làm hoặc tìm hiểu qua sách báo liên quan đến đề tài nghiên cứu.
  1. ***Phương pháp nghiên cứu thực tiễn*** 
     + Phương pháp quan sát: quan sát việc quản lý của một số quán café tại thành phố Thủ Dầu Một.
     + Phương pháp phỏng vấn: Lấy thông tin thực tế từ chủ quán, nhân viên, khách hàng…
  2. ***Phương pháp tổng hợp và phân tích số liệu***
     + Thu thập thông tin từ việc phỏng vấn.
     + Thống kê, tổng kết số liệu.
     + Phân tích, đưa ra kết quả và thực hiện.
  3. **Phạm vi nghiên cứu**

Phạm vi quay quanh các nghiệp vụ bán hàng, tuân theo nghiệp vụ quán lý của một quán cafe, đảm bảo đầy đủ các chức năng chính của phần mềm và khai thác các chức năng mới có liên quan đến quản lý quán café nhằm giúp cho phần mềm thêm phần tiện ích.

1. **Đóng góp của đề tài**

Mở ra một hướng đi mới trong việc quản lý một quán café tiện ích hơn.

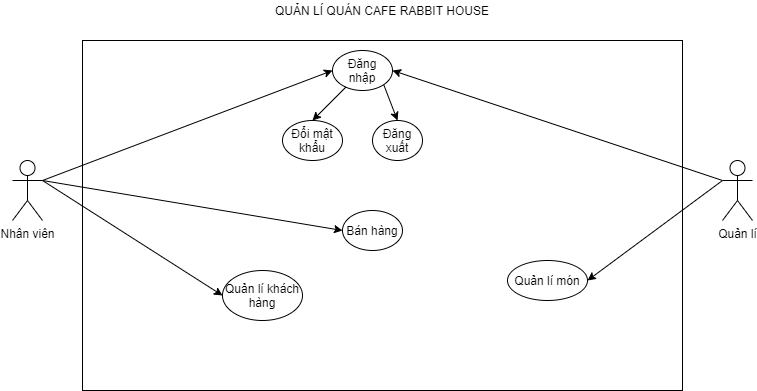
Tổng hợp các chức năng cần thiết để quản lý một quán café góp phần giúp ích cho việc kinh doanh và quản lý thu chi, nhân viên, hàng hóa ... một cách nhanh chóng và chính xác hơn.

1. Phân tích thiết kế hệ thống
2. **Tổng quan về bài toán**

Công ty Rabbit’s House muốn mở một quán café để phục vụ khách hàng là các sinh viên và các nhân viên. Việc mở quán diển ra bình thường. Sau 1 thời gian bán bình nhưng rất cực nên chủ quán quyết định xây dựng một hệ thống chương trình giúp dể dàng hơn cho nhân viên làm việc.

Chương trình được viết ra phải phù hợp các tiêu chí như đơn giản dể sử dụng, nhanh gọn, thông minh có thể tính lương và doanh thu của quán, quản lí có thể quản lí được tiếng trình làm việc quả quán.

1. **Sơ đồ Use Case**



1. **Đặc tả**
   1. ***Đăng nhập***

|  |  |
| --- | --- |
| **Tên use case: Quản lý đăng nhập** | |
| Actor: khách hàng, chủ cửa hàng, nhân viên | |
| Mô tả: Use case này mô tả các bước đăng nhập của các actor vào hệ thống. | |
| Precondition (điều kiện tiên quyết): Nhập tên đắng nhập và mật khẩu. | |
| Poscondition(vị trí): +Nếu đăng nhập thành công thì hệ thống sẽ hiển thị thông tin người nhập tương ứng tài khoản đăng nhập, sản phẩm. Nếu sai thì hệ thống sẻ yêu cầu nhập lại.  +nếu khách hàng chưa có tài khoảng, thì hệ thống sẻ yêu cầu khách hàng đăng ký tài khoản. | |
| Basic flow | |
| Actor | Hệ thống |
| 1. Tạo form kết quả tìm kiếm của hệ thống, người dùng click vào nút đăng ký. | 1. Hệ thống hiển thị form yêu cầu nhập tên đăng nhập và mật khẩu. |
| 1. Người dùng nhập mã thẻ, tên sản phẩm và số lượng mua. | 1. Hệ thống kiểm tra mã thẻ, kiểm tra thông tin. 2. Nếu thông tin chính xác thì hệ thống sẻ bắc đầu tìm kiếm thông tin sản phẩm. |
| 1. Người dùng xác nhận thông tin sản phẩm và đơn giá | 1. Hệ thống sẻ in hóa đơn sản phẩm. Khách hàng có thể thanh toán bằng tiền hoặc sử dụng hệ thống vận chuyển. |
| Alternale flow | |
| Người dùng nhập lại thông tin | +Nếu thông tin không chính xác thì hệ  thống yêu cầu nhập lại.  + Hệ thống thực hiện lại bước 4, số lần  lặp tối đa là 3. Hệ thống sẽ tự động kết thúc  việc đăng ký. |

* 1. ***Đăng xuất***

|  |  |
| --- | --- |
| **Tên use case: Đăng xuất** | |
| Actor: các thành viên trong use case đăng nhập. | |
| Mô tả: Use case này mô tả các bước đăng xuất của các actor vào hệ thống. | |
| Precondition(điều kiện tiên quyết): Đăng xuất ra khỏi hệ thống | |
| Poscondition(vị trí): +actor thực hiện các chức năng đăng xuất khỏi hệ thống.  +hệ thống yêu cầu xác nhận từ actor  +xác nhận đăng xuất.  + Hệ thống đăng xuất tài khoản actor khỏi hệ thống. Nếu Actor không xác nhận đăng xuất thì hệ thống sẽ giữ nguyên hiện trạng. | |
| Basic flow | |
| Actor | Hệ thống |
| 1. Tạo form hiển thị chức năng đăng xuất của hệ thống, người dùng click nút đăng xuất | 1. Hệ thống hiển thị form yêu cầu xác nhận đăng xuất |
| 1. Người dùng kiểm tra lại thông tin trước khi xác nhận đăng xuất. 2. Nếu người dùng xác nhận | 1. Hệ thống tự động kiểm tra và đăng xuất khỏi hệ thống. |
| Alternale flow | |
| Người dùng xác nhận lại thông tin trước khi đăng xuất. | +nếu người dùng đã xác nhận thông tin xong và đăng xuất thì hề thống tự động đăng xuất tài khoản.  +ngược lại nếu người dùng không xác nhận thống tin thì hệ thống sẻ giữ nguyên hiện trạng. |

* 1. ***Quản lý nhân viên***

|  |  |
| --- | --- |
| **Tên use case: Quản lý nhân viên** | |
| Actor : Nhân viên quản lý | |
| Mô tả: Use case này cho phép actor (nhân viên quản lý) duy trì thông tin nhân viên trong hệ thống quản lý quán cafe. Bao gồm thêm, hiệu chỉnh và xóa nhân viên ra khỏi hệ thống. | |
| Precondition(điều kiện tiên quyết): Nhập tên đăng nhập, mật khẩu và chọn chức năng “quản lý nhân viên”. | |
| Poscondition(vị trí):  + Use case này bắt đầu khi actor muốn thêm, thay đổi, và/hoặc xóa thông tin nhân viên trong hệ thống.  + Hệ thống yêu cầu actor chọn chức năng muốn thực hiện (Thêm nhân viên, Cập nhật thông tin nhân viên, hoặc Xóa nhân viên).  \* Sau khi actor cung cấp thông tin được yêu cầu, một trong các luồng phụ sau được thực hiện.  + Nếu actor chọn “Thêm thông tin nhân viên”, luồng phụ Thêm nhân viên được thực hiện.  + Nếu cán bộ đào tạo chọn “Cập nhật thông tin nhân viên”, luồng phụ Cập nhât thông tin nhân viên được thực hiện.  + Nếu cán bộ đào tạo chọn “Xóa thông tin nhân viên”, luồng phụ Xóa nhân viên được thực hiện. | |
| Basic flow | |
| Actor | Hệ thống |
| 1. Nhấn vào nút “Quản lý nhân viên”. | 1. Hệ thống yêu cầu actor nhập vào các thông tin của nhân viên. |
| 1. actor cung cấp thông tin được yêu cầu. | 1. Hệ thống sẽ phát sinh và gán một số ID độc nhất cho nhân viên này. Nhân viên này được thêm vào hệ thống . 2. Hệ thống cung cấp cho actor số ID của nhân viên mới. |
| **Luồng phụ** | |
| **Chức năng “Sửa”** | |
| 1. Actor nhấn vào nút “sửa”. | 1. Hệ thống yêu cầu actor nhập vào số ID của nhân viên hoặc tìm kiếm nhân viên. |
| 1. Actor thay đổi một số thông tin của nhân viên. Gồm bất cứ thông tin nào được chỉ ra trong luồng phụ Thêm nhân viên | 1. hệ thống cập nhật mẩu tin của nhân viên này. |
| **Chức năng “xóa”** | |
| 1. Actor nhấn vào nút “xóa” | 1. Hệ thống yêu cầu actor nhập vào số ID của nhân viên hoặc tìm kiếm nhân viên |
| 1. Actor nhập số ID nhân viên. Hệ thống truy xuất và hiển thị thông tin của nhân viên này. | 1. Hệ thống nhắc người dùng xác nhận thao tác xóa nhân viên. |
| 1. Actor xác nhận xóa. | 1. Hệ thống xóa thông tin của nhân viên này ra khỏi hệ thống. |
| Alternale flow | |

* 1. ***Thanh toán***

|  |  |
| --- | --- |
| **Tên use case: thanh toán** | |
| Actor: nhân viên, nhân viên quản lý, khách hàng | |
| Mô tả: Use case này mô tả các bước thanh toán của actor với khách hàng | |
| Precondition(điều kiện tiên quyết): kiểm tra đơn giá và hóa đơn khách hàng. | |
| Poscondition(vị trí): nhân viên kiểm tra mặc hàng, số lượng hàng. Hệ thống hiển thị menu thống kê; ngày tháng năm. Nhân viên kế tooans chọn 1 trong các thư mục. hệ thống sẻ thống kê và in ra giấy. | |
| Basic flow | |
| Actor | Hệ thống |
| 1. Người quản lý chọn chức năng thống kê | 1. Hệ thống yêu cầu chọn hình thức thống kê. |
| 1. Người quản lý chọn hình thức thống kê qua (tháng quý thời điểm hiện tại). | 1. Hệ thống hiển thị bảng thống kê, doanh thu,…. 2. Hệ thống hỏi người quản lý có muốn in bảng thống kê không.. |
| 1. Người quản lý in bảng thống kê | 1. Hệ thống sẻ in cho người quản lý. |
| Alternale flow | |
| Người dùng không in bảng thống kê | +người quản lý sẻ không in bảng thống kê  + use case kết thúc. |

* 1. ***Quản lí nguyên liệu trong kho***

|  |  |
| --- | --- |
| **Tên use case: Quản lý nguyên liệu trong kho** | |
| Actor: người quản lý | |
| Mô tả: Use case này cho phép người quản lý nhập hàng vào cửa hàng khi hàng hóa hết. | |
| Precondition(điều kiện tiên quyết): cập nhật mã hàng vào hệ thống | |
| Poscondition(vị trí): | |
| Basic flow | |
| Actor | Hệ thống |
| 1. Người quản lý chọn chức năng nhập hàng vào hệ thống. | 1. Hệ thống yêu cầu nhập hàng. |
| 1. Người quản lý nhập mã hàng | 1. Hệ thống kiểm tra mã hàng có hợp lệ không. 2. Hệ thống yêu cầu nhập số lượng. |
| 1. Người dùng nhập số lượng và nhà sản xuất | 1. Hệ thống ghi nhận số lượng và kt nhà sản xuất. 2. Hệ thống chuyển thông tin đến chủ kho. |
| Alternale flow | |
| Người quản lý nhập mã không hợp lệ. | +hệ thống sẽ kiểm tra bước 4 ở trên. |

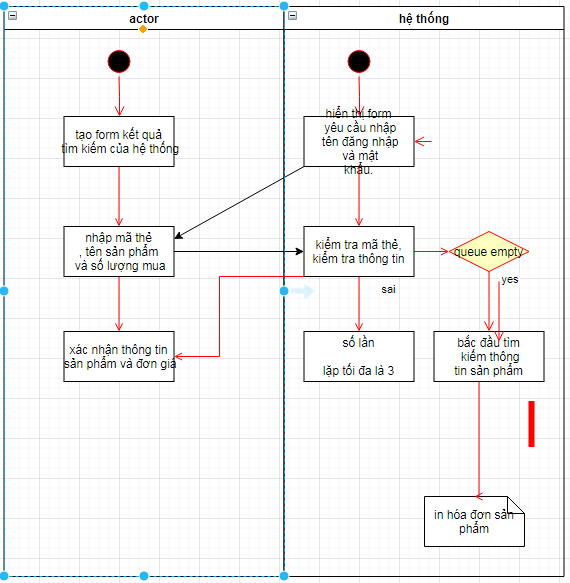
* 1. ***Chọn đồ uống***

|  |  |
| --- | --- |
| **Tên use case: CHỌN ĐỒ UỐNG** | |
| Actor: khách hàng | |
| Mô tả: Use case này cho phép khách hàng chọn đồ uống. | |
| Precondition(điều kiện tiên quyết): khách hàng phải đăng nhập và chọn đặt hàng | |
| Poscondition(vị trí): +Khách hàng đăng nhập vào hệ thống  + Chọn sản phẩm và chọn chức năng đặt hàng  + Hệ thống hiển thị form thông tin khách hàng và ngày nhận. Bao gồm: Số CMND; Họ tên; Địa chỉ; SĐT.  + Hệ thống tự động kiểm tra thông tin sản phẩm mà khách hàng yêu cầu, đồng thời lọc danh sách các loại sản phẩm và các sản phẩm tương ứng mà khách hàng có thể mua vào ngày đó. | |
| Basic flow | |
| Actor | Hệ thống |
| Người dùng click vào nút đặt hàng. | Hệ thống hiển thị form yêu cầu nhập số lượng và đặt hàng. |
| Người dùng chọn sản phẩm, nhập số lượng | Hệ thống kiểm tra thông tin.  Nếu thông tin chính xác thì hệ thống sẻ bắc đầu xuất form hóa đơn. |
| Người dùng xác nhận thông tin sản phẩm và đơn giá | Hệ thống sẻ in hóa đơn sản phẩm. Khách hàng có thể thanh toán bằng tiền hoặc sử dụng hệ thống vận chuyển. |

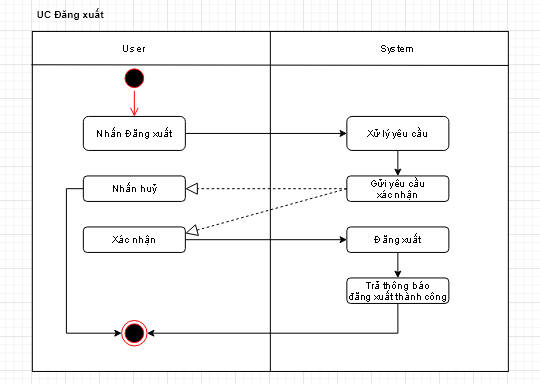
* 1. ***Quản lí order***

|  |  |
| --- | --- |
| **Tên use case: QUẢN LÍ ORDER** | |
| Actor: Khách hàng, nhân viên | |
| Mô tả: Use case cho phép quản lí các đơn hàng order. | |
| Precondition (điều kiện tiên quyết): Khách hàng đã đăng kí | |
| Poscondition(vị trí): + Khách hàng chọn chức năng giao hàng, hình thức giao hàng.  + Nhân viên yêu cầu nhập thông tin người nhận hàng và xác nhận.  + Khách hàng chọn phương thức thanh toán.  + Nhân viên phản hồi lại khách hàng.  + Nhân viên giao hàng đến cho khách và yêu cầu nhận hàng.  + Khách hàng nhận hàng và thanh toán theo phương thức thanh toán nếu cần. | |
| Basic flow | |
| Actor | Hệ thống |
| Nhân viên yêu cầu nhập thông tin người nhận hàng và xác nhận | Yêu cầu lưu thông tin đồng thời cập nhật lại giỏ hàng. |
| Khách hàng chọn phương thức thanh toán. | Hệ thống kiểm tra thông tin.  Yêu cầu nhập phương thức giao hàng và phương thức thanh toán. |
| Nhân viên phản hồi lại khách hàng.  Nhân viên giao hàng đến cho khách và yêu cầu nhận hàng.  Khách hàng nhận hàng và thanh toán theo phương thức thanh toán nếu cần. | Yêu cầu xác nhận lại thông tin và hiển thị mẫu hóa đơn trên giao diện |

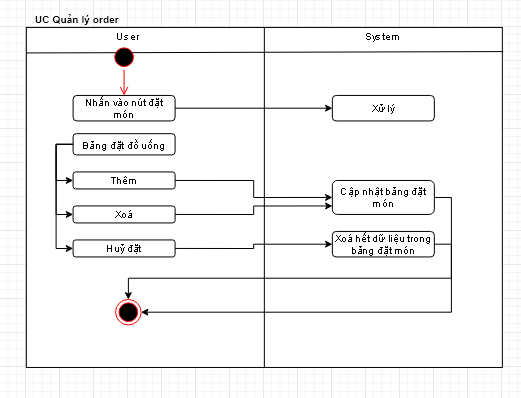
1. **Sơ đồ hoạt động**
   1. ***Đăng nhập***



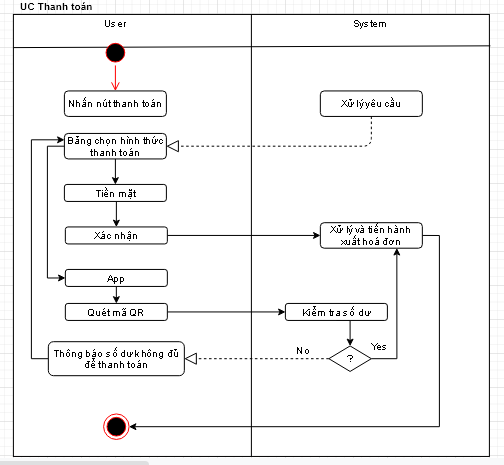
* 1. ***Đăng xuất***



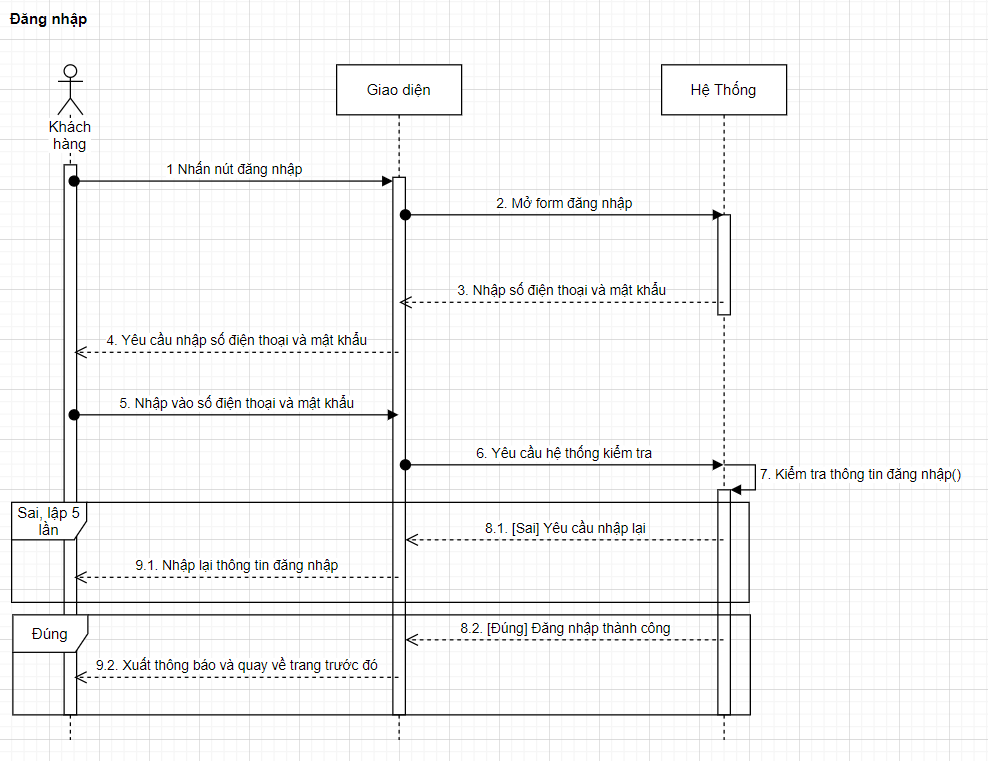
* 1. ***Quản lí order***



* 1. ***Thanh toán***



1. **Sơ đồ tuần tự & biểu đồ cộng tác**
   1. ***Đăng nhập***

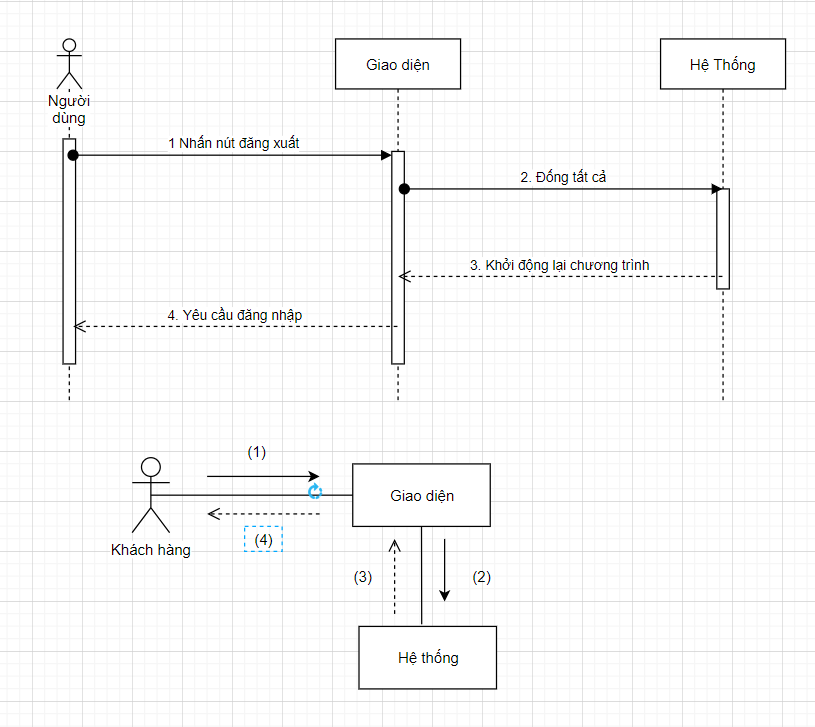


/Nhân viên



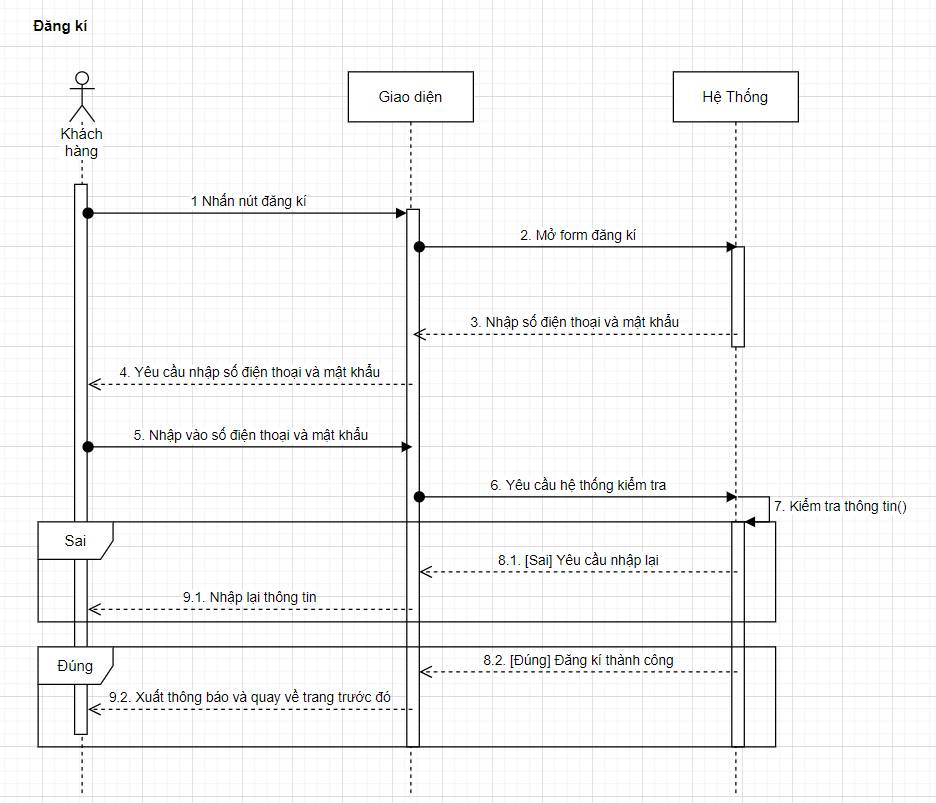
/Nhân viên

* 1. ***Đăng xuất***



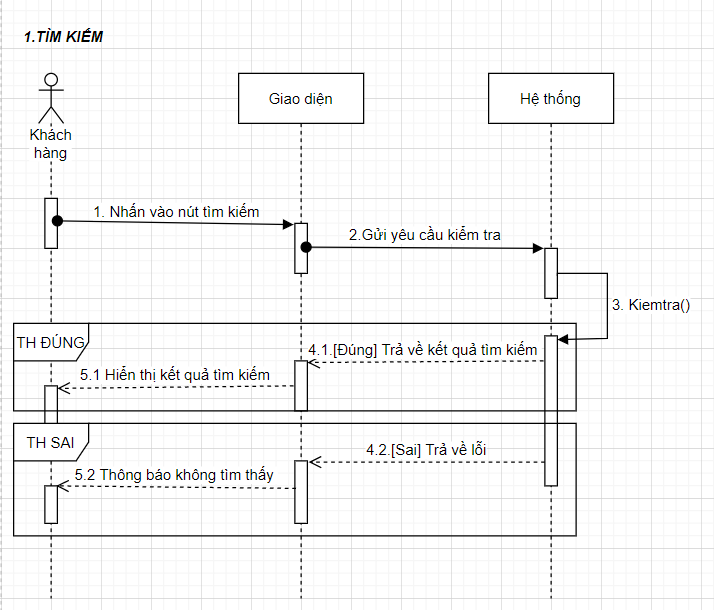
/Nhân viên

* 1. ***Đăng kí***

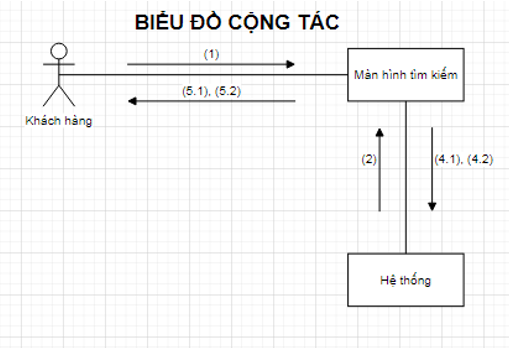




* 1. ***Tìm kiếm***

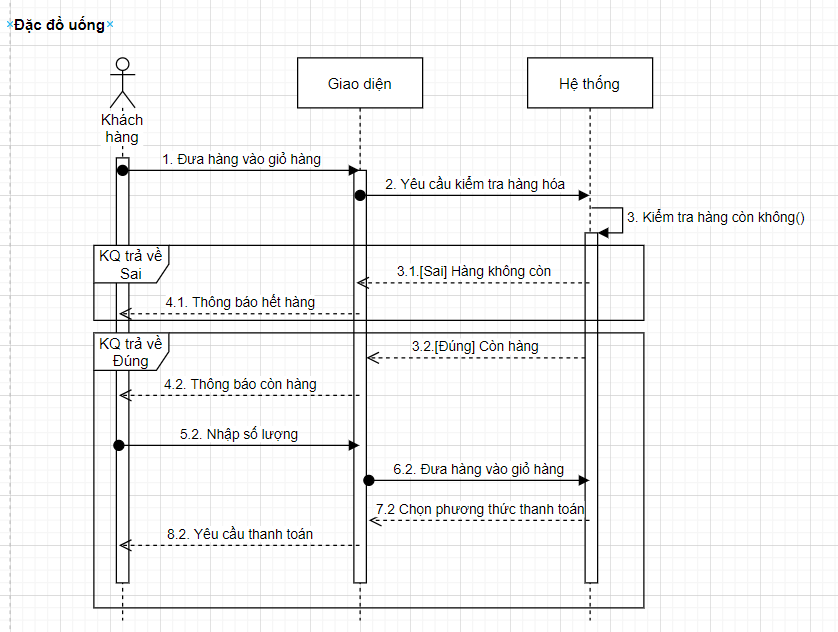


/Nhân viên

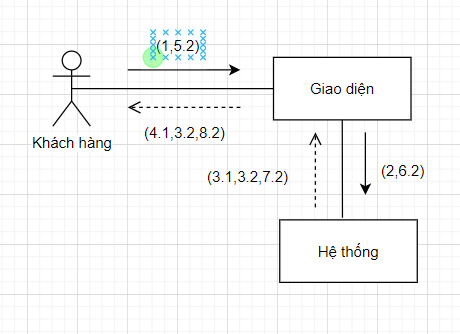


/Nhân viên

* 1. ***Đặc hàng***

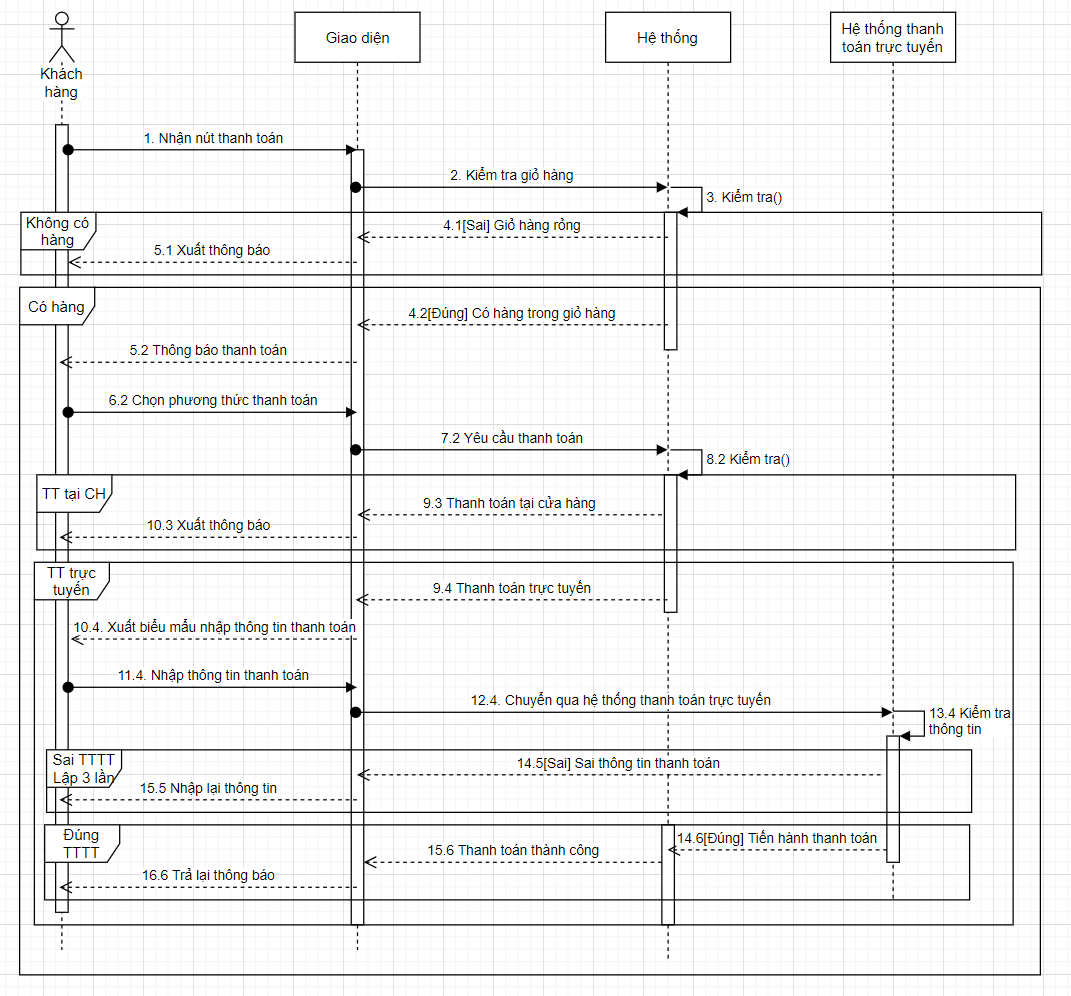


/Nhân viên

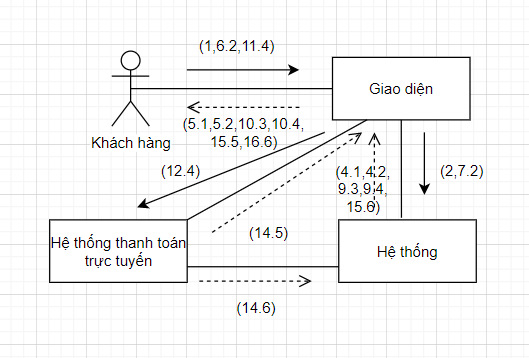


/Nhân viên

* 1. ***Thanh toán***

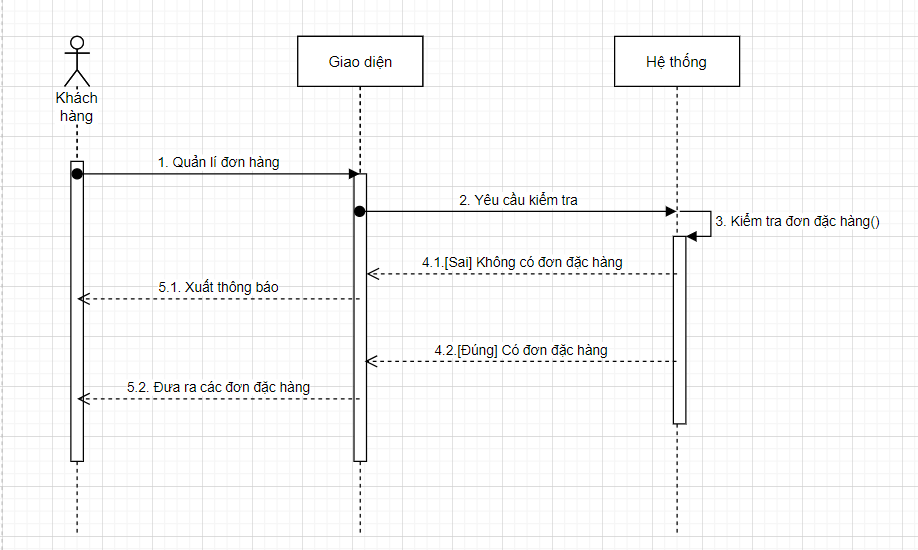


/Nhân viên

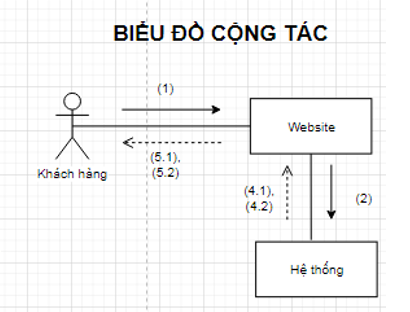


/Nhân viên

* 1. ***Quản lí đơ đặc hàng***



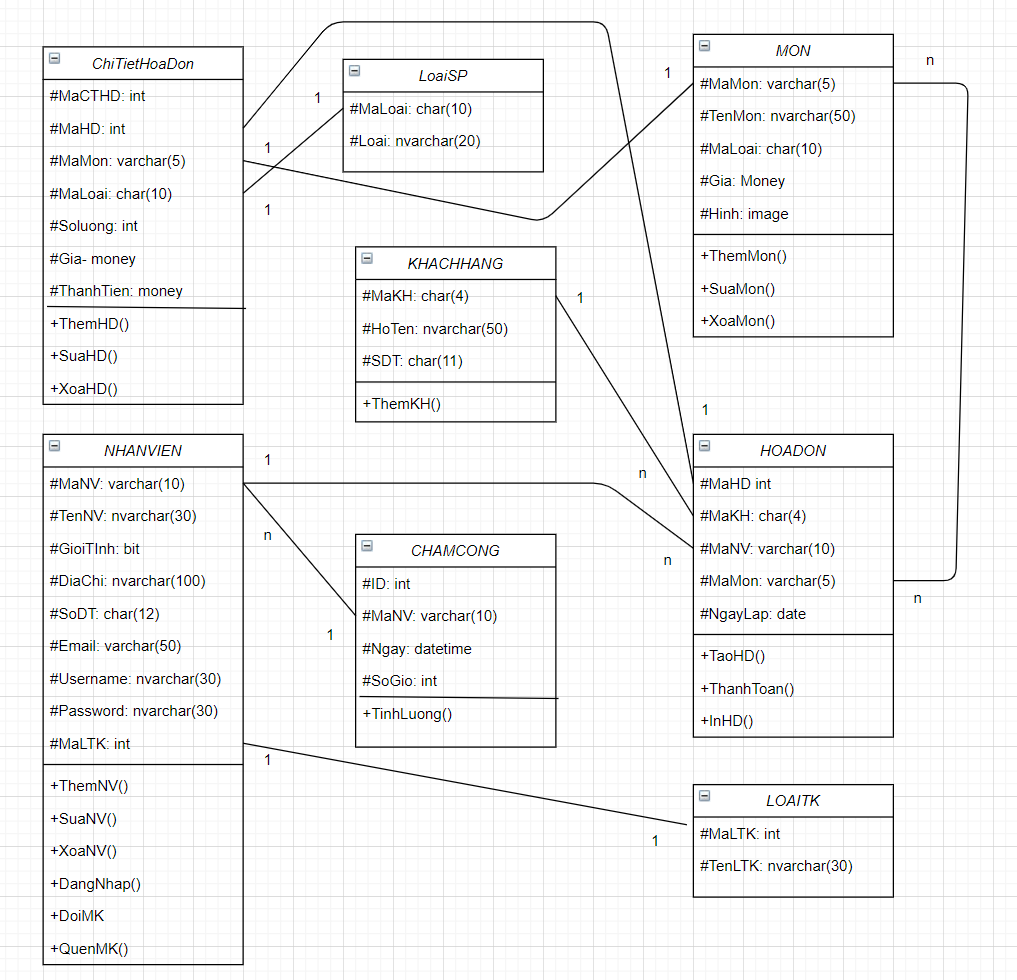
/Nhân viên/ quản lý



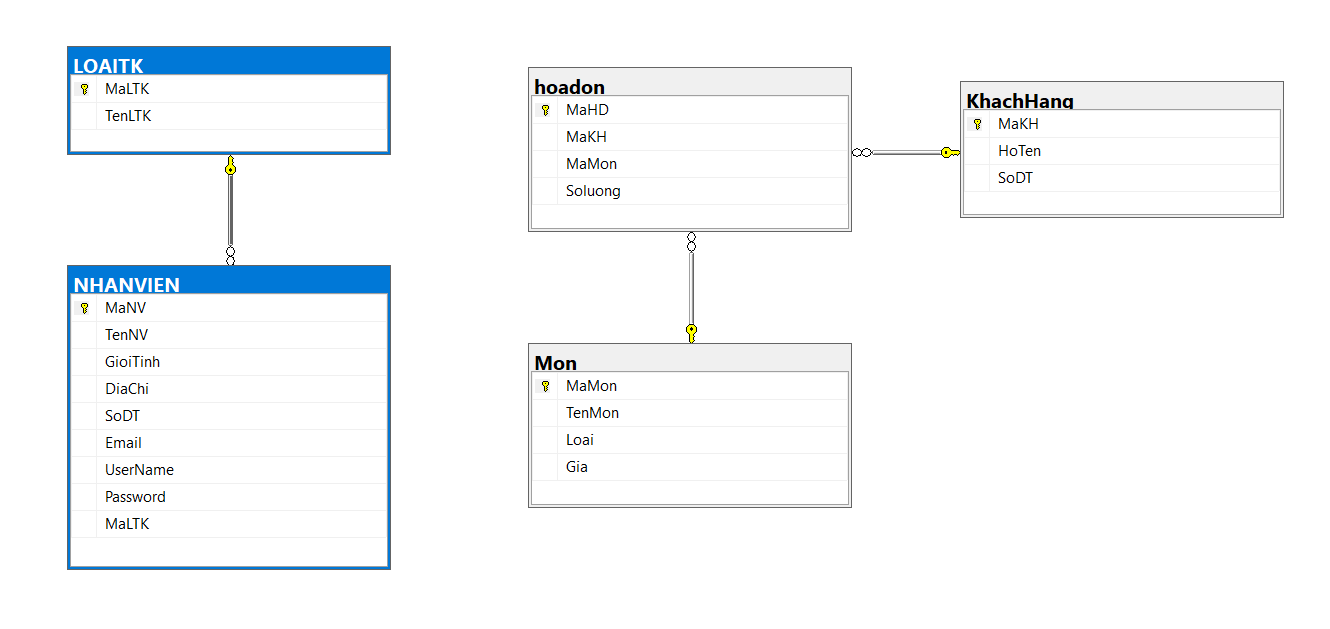
Giao Diện

/Nhân viên/ quản lý

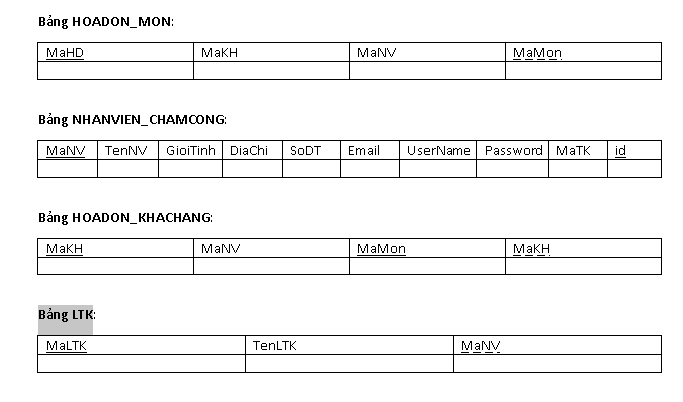
1. **Biểu đồ lớp**



1. Phân tích thiết kế ứng dụng
2. **Class Dinagram cơ sở dữ liệu**



1. **Chuyển đổi thành mô hình dử liệu quan hệ**



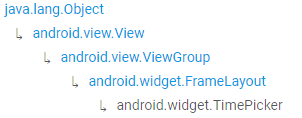
**CHƯƠNG 2. TỔNG QUAN VỀ VIEW DATE-TIME PICKER**

1. Time picker

TimePicker là một ui widget cho phép người sử dụng chọn thời gian trong ngày theo chế độ AM/PM, chọn theo chế độ 24h. Thời gian hiển thị gồm: Giờ, Phút, Giây và theo dạng đồng hồ hiển thị. Nếu chúng ta muốn hiển thị đồng hồ theo dạng dialog thì chúng ta phải sử dụng lớp **TimePickerDialog.**



android.widget.TimePicker là lớp con của FrameLayout.



**TimePicker code trong XML**

<TimePicker

android:id="@+id/simpleTimePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:timePickerMode="spinner"/>

**Các phương thức thường dùng của TimePicker**

**1. setCurrentHour(Integer currentHour):**Phương thức này dùng để thiết lập ngày hiện tại chomột **TimePicker**

Từ API 23 trở về trước thì sử dụng phương thức**setCurrentHour().**Còn từ API 23 trở về sau sử dụng phương thức ***setHour(Integer hour).***Trong phương thức này chỉ có một tham số là số nguyên được sử dụng thiết lập giá trị giờ.

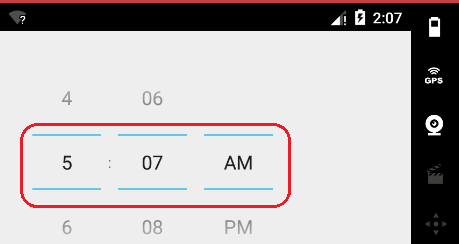
Ví dụ sau thiết lập giờ hiện tại là 5.

TimePicker simpleTimePicker=(TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time picker

// set the value for current hours

simpleTimePicker.setCurrentHour(5); // Trước api 23

simpleTimePicker.setHour(5); // sau api 23



**2. setCurrentMinute(Integer currentMinute):**Phương thức này dùng để thiết lập ngày hiện tại chomột **TimePicker**

Từ API 23 trở về trước thì sử dụng phương thức**setCurrentMinute().**Còn từ API 23 trở về sau sử dụng phương thức ***setMinute(Integer minute).***Trong phương thức này chỉ có một tham số là số nguyên được sử dụng thiết lập giá trị phút.

Ví dụ sau thiết lập phút hiện tại là 35.

TimePicker simpleTimePicker=(TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time picker

// set the value for current hours

simpleTimePicker.setCurrentMinute(35); // Trước api 23

simpleTimePicker.setMinute(35); // sau api 23

**3. getCurrentHour():**Phương thức này lấy giờ hiện tại của **TimePicker**

Từ API 23 trở về trước thì sử dụng phương thức**getCurrentHour().**Còn từ API 23 trở về sau sử dụng phương thức ***getHour().*** Phương thức này trả về giá trị số nguyên

Code sau lấy giờ hiện tại của **TimePicker**

TimePicker simpleTimePicker = (TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time

pickerint hours =simpleTimePicker.getCurrentHour(); // Trước api 23

int hours =simpleTimePicker.getHour(); // sau api 23

**4. getCurrentMinute():**Phương thức này lấy phút hiện tại của **TimePicker**

Từ API 23 trở về trước thì sử dụng phương thức**getCurrentMinute().**Còn từ API 23 trở về sau sử dụng phương thức ***getMinute().*** Phương thức này trả về giá trị số nguyên

Code sau lấy phút hiện tại của **TimePicker**

TimePicker simpleTimePicker = (TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time picker

int minutes = simpleTimePicker.getCurrentMinute(); // trước api 23

int minutes = simpleTimePicker.getMinute(); // sau api 23

**5. setIs24HourView(Boolean is24HourView):**Phương thức dùng thiết lập chế độ giờ hiển thì là 24h hay là hiển thị dạng **AM/PM.** Trong phương thức này thiết lập chế độ**true/false.** Nếu giá trị **true**chế độ hiển thị theo 24h, nếu **false**hiển thị theo chế độ **AM/PM**

Ví dụ thiết lập chế độ cho một **TimePicker**

TimePicker simpleTimePicker = (TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time picker

simpleTimePicker.setIs24HourView(true); // set 24 hours mode for the time picker



**6. is24HourView():**Phương này kiểm tra xem chế độ hiện tại **24h**hay **AM/PM.**Phương thức này trả về **true**nếu nó đang ở chế độ **24h**ngược lại **false**đang ở chế độ **AM/PM**

Ví dụ sau lấy chế độ hiển thị hiện tại của **TimePicker**

TimePicker simpleTimePicker = (TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time picker

Boolean mode=simpleTimePicker.is24HourView(); // Kiểm tra xem chế độ hiện tại của TimePicker

**7. setOnTimeChangedListener(TimePicker.OnTimeChangedListener onTimeChangedListener):**Phương thức này thiết lập hàm callback mà chỉ rằng thời gian đã được chỉnh sửa bởi người dùng.

Phương thức onTimeChanged(TimePicker view, int hourOfDay, int minute) lắng nghe sự kiện từ người sử dụng, nó gồm 3 tham số: **TimePicker**, **hourOfDay**giờ trong ngày, **minute**dùng để lấy phút sau khi thay đổi thời gian của **TimePicker.**

TimePicker simpleTimePicker = (TimePicker)findViewById(R.id.simpleTimePicker); // initiate a time picker

simpleTimePicker.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {

@Override

public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {

}

});

**Một số thuộc tính thường dùng của TimePicker**

**1. android:id:**Là thuộc tính duy nhất của **TimePicker**.

<TimePicker

android:id="@+id/simpleTimePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/> <!-- id of a time picker -->

Dựa vào Id ta sẽ lấy được control theo đúng Id này, xem code bên dưới để biết cách lấy control theo Id:

TimePicker simpleTimePicker = (TimePicker)findViewById(R.id.simpleTimePicker);

**2. android:timePickerMode:**thuộc tính này thường được sử dụng để thiết lập chế độ hiển thị dạng đồng hồ là **spinner**hay là **clock**. trước API 21 mặc định chế độ là clock, sau API 21 phải thiết lập chế độ là **spinner**

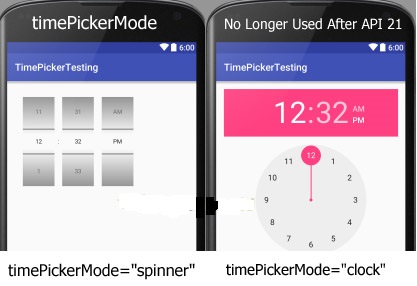
<TimePicker

android:id="@+id/simpleTimePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:timePickerMode="spinner" /> <!-- time picker mode of a time picker -->



**3. android:background:**Thuộc tính này thiết lập màu nền hoặc image trong thư mục drawable cho **TimePicker**.

<TimePicker

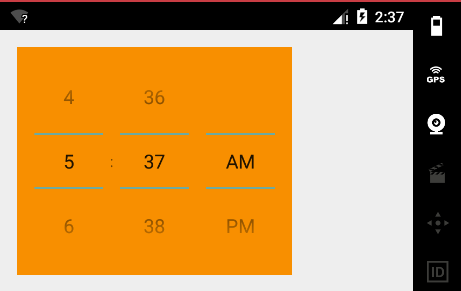
android:id="@+id/simpleTimePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:timePickerMode="spinner"

android:background="#F88F00" /> <!-- orange background color for the time picke



**Thiết lập màu nên trong java class**

TimePicker simpleTimePicker=(TimePicker)findViewById(R.id.simpleTimePicker); //initiate a time picker

simpleTimePicker.setBackgroundColor(Color.YELLOW); //Yellow background color for the background of a time picker

**4. android:padding:**Thuộc tính này xác định khoảng cách từ đường viền của **TimePicker**với nội dung nó chứa:**left, right, top or bottom.** Cũng ví dụ trên bây giờ chúng ta xác định**padding=50dp**từ mọi phía của **TimePicker.**

<TimePicker

android:id="@+id/simpleTimePicker"

android:layout\_width="wrap\_content"

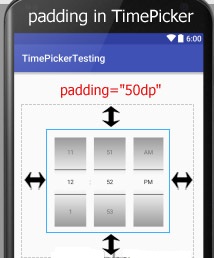
android:layout\_height="wrap\_content"

android:timePickerMode="spinner"

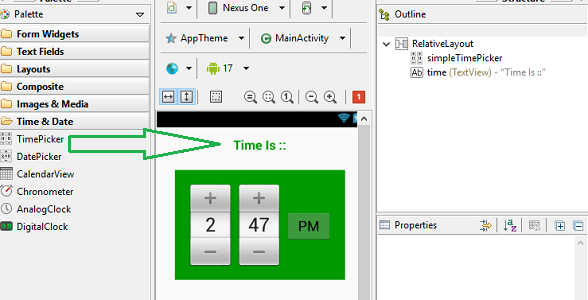
android:layout\_centerHorizontal="true"

android:layout\_marginTop="50dp"

android:padding="50dp"/> <!-- 20dp padding from all the sides of a time picker

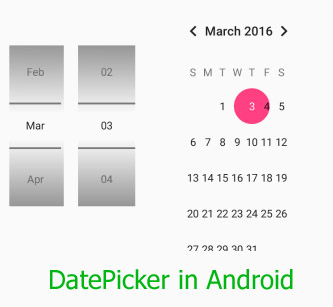


**Ví dụ:**Trong ví dụ này chúng ta sẽ làm ứng dụng gồm có một **TimePicker**và một **TextView**. Khi người sử dụng tùy chỉnh giờ trên TimePicker chúng ta sẽ lấy giá trị của nó rồi hiển thị trên **TextView**và **Toast**. Tiến hành tạo project, vào thư mục  **res /layout -> activity\_main.xml**thiết kế giao diện sau:

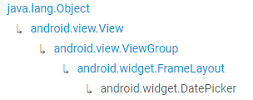


1. Date picker

**DatePicker**là một ui widget thường được sử dụng chọn ngày. **DatePicker**cho phép chọn ngày, tháng và năm.Để hình dung Dialog này như thế nào thì các bạn có thể thấy hình dạng của DatePickerDialog dưới đây.



android.widget.DatePicker là lớp con của lớp FrameLayout.



**DatePicker code trong XML:**

<DatePicker

android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:datePickerMode="spinner"/>

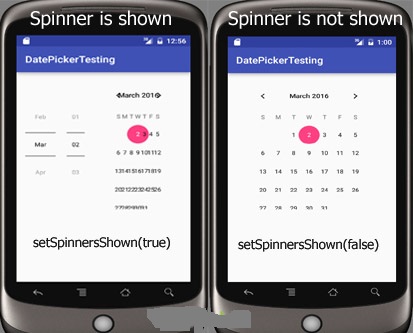
**Các phương thức thường dùng của DatePicker**

**1. setSpinnersShown(boolean shown):**Phương thức này thường sử dụng để hiện thi DatePicker ở dạng spinner hay không. Trong phương thức này chúng ta có thể thiết lập giá trị **true**hoặc **false.**Nếu giá trị là**false** thì spinner không hiển thị. Mặc định là **true**

Code sau cách sử dụng phương thức**setSpinnersShow()**là **false**

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

simpleDatePicker.setSpinnersShown(false); // set false value for the spinner shown function



**2. getDayOfMonth():**Phương thức này được sử dụng để lấy ngày trong  tháng từ **DatePicker.**Phương thức này trả về một số nguyên.

Code sau lấy ngày trong tháng từ **DatePicker**

/\*Add in Oncreate() funtion after setContentView()\*/

DatePicker simpleDatePicker = (DatePicker) findViewById(R.id.simpleDatePicker); // initiate a date picker

int day = simpleDatePicker.getDayOfMonth(); // Lấy ngày trong tháng

**3. getMonth():**Phương thức này được sử dụng để lấy tháng từ **DatePicker.**Phương thức này trả về một số nguyên.

Code sau lấy tháng từ **DatePicker**

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

int month = simpleDatePicker.getMonth(); // Lấy tháng được chọn

**4. getYear():**Phương thức này được sử dụng để lấy năm từ **DatePicker.**Phương thức này trả về một số nguyên.

Code sau lấy năm từ **DatePicker**

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

int year = simpleDatePicker.getYear(); // Lấy năm được chọn

**5. getFirstDayOfWeek():**Phương thức này lấy ngày đầu tiên trong tuần. Phương thức này trả về một số nguyên.

Code sau lấy ngày đầu tiên trong tuần

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

int firstDay=simpleDatePicker.getFirstDayOfWeek(); // Lấy ngày đầu tiên trong tuần

**Một số thuộc tính thường dùng của DatePicker**

**1. android:id:**Là thuộc tính duy nhất của **DatePicker**.

<DatePicker

android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

/>

Dựa vào Id ta sẽ lấy được control theo đúng Id này, xem code bên dưới để biết cách lấy control theo Id:

DatePicker simpleDatePicker = (DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

**2. android:datePickerMode:**Thuộc tính này thường dùng để thiết lập **DatePicker**theo chế độ **spinner**hay **calendar**. Mặc định là **calendar.**Để thiết lập chế độ **spinner**phải sử dụng API từ 21 trở về sau:

Code sau chúng ta thiết lập chế độ **spinner**cho **DatePicker**

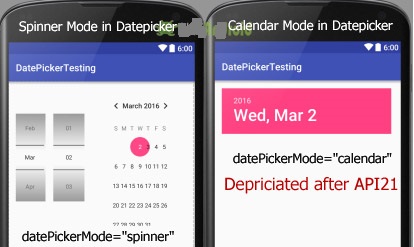
<DatePicker

android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:datePickerMode="spinner" /> <!-- spinner mode of a date picker -->



**3. android:background:**Thuộc tính này thiết lập màu nền hoặc image trong thư mục drawable cho **DatePicker**.

Code sau chúng ta thiết lập màu nền, màu đỏ cho **DatePicker**

<DatePicker

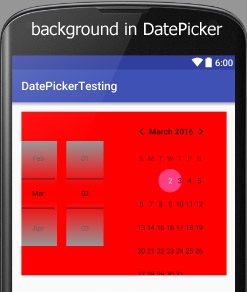
android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:datePickerMode="spinner"

android:background="#f00"/> <!-- red color for the background of the date picker -->



**Thiết lập màu nên trong java class**

DatePicker simpleDatePicker=(DatePicker)findViewById(R.id.simpleDatePicker); // initiate a date picker

simpleDatePicker.setBackgroundColor(Color.RED); // Thiết lập màu nền là màu đỏ

**4. android:padding:**Thuộc tính này xác định khoảng cách từ đường viền của **TimePicker**với nội dung nó chứa:**left, right, top or bottom.** Cũng ví dụ trên bây giờ chúng ta xác định**padding=40dp**từ mọi phía của **DatePicker.**

<DatePicker

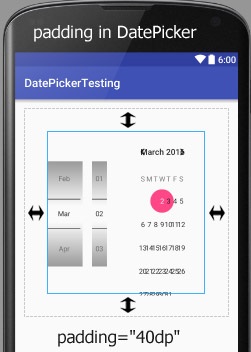
android:id="@+id/simpleDatePicker"

android:layout\_width="wrap\_content"

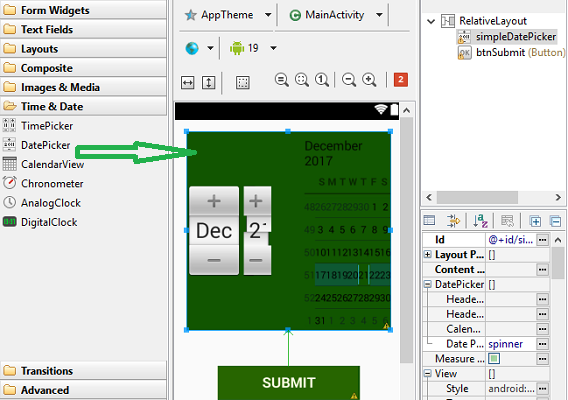
android:layout\_height="wrap\_content"

android:datePickerMode="spinner"

android:padding="40dp"/> <!-- 40dp padding from all the sides of a date picker -->



**Ví dụ:**Trong ví dụ này chúng ta sẽ làm ứng dụng gồm có một **DatePicker**và một **Button**. Khi người sử dụng chọn ngày  trên **DatePicker**và click vào **Button "Submit"** chúng ta sẽ lấy giá trị của **DatePicker**rồi hiển thị lên **Toast**. Tiến hành tạo project, vào thư mục  **res /layout -> activity\_main.xml**thiết kế giao diện sau:



**CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG ỨNG DỤNG**

1. **Giao diện & các trường hợp**
   1. ***Đăng nhập***

*Quên mật khẩu?*

Thoát

Đăng nhập

UserName

Password

* Đây là màng hình bắt đầu khi chương trình được chạy lên.
* Tại đây, tất cả các quyề như admin và nhân viên có thể đăng nhập vào và sử dụng phần mềm.
* Nếu nhân viên hay admin quên mật khẩu, họ chỉ cần nhấp vào “Forgot your password?” để lấy lại mật khẩu
* Nễu là tài khoản mới tạo đăng nhập lần đầu, hệ thống yêu cầu đổi mật khẩu, mango hình đổi mật khẩu xuất hiện.
  1. ***Quên mật khẩu***

Hủy

Kiểm tra

UserName

UserID

*Để có UserID hãy liên lạc với quản trị của bạn.*

* Đây là màng hình xác minh tài khoản với hai ô, ta nhập tên đăng nhập và key nhân viên vào để tiến vào màng hình đổi mật khẩu.
* Key nhân viên chính là mã nhân viên được cấp trong khi tạo tài khoản
* Vì đây là bước xác minh nên yêu cầu chủ tài khoản nhập đúng thông tin. Nếu xác nhận là đúng, chương trình chuyển sang màng hình đổi mật khẩu.
  1. ***Đổi mật khẩu***

Hủy

Đổi & ĐN

Password

N.L.Pass

* Đây là màng hình nhập mật khẩu
* Mật khẩu mới phải được nhập lại hai lần để tránh sai xót.
* Sau chi nhập xong, màng hình đăng nhập sẽ hiện lên cho người dung đăng nhập.
  1. ***Trang chủ***

Xin chào: Aoki Daisuke

K.Tra Kho

Bán Hàng

Q.Lý NV

Q.Lý KH

* Là màng hình xuất hiện sau chi đã đăng nhập thành công
* Tùy theo loại tài khoản đã được đăng nhập, các chức năng của trang chủ sẽ thây đổi cho phù hợp.
  + 1. ***Quản lí: Quản lí tất cả thông tin về quán café Rabbit’s House***
* Đây là màng hình có thể xem tất cả thong tin về nhân viên, số hóa đơn và hang hóa
* Có thể quản lí thông tin nhân viên, tính lương nhân viên,…
* Có thể tổng hợp số hóa đơn, doanh thu
* Xem lượng hàng hóa còn và hết trong kho
* Tại đây, admin có thể thêm sửa xóa menu thực đơn của Rabbit’s House

1. ***Bán hàng***

* Trong tất cả các quyền (cả quyền admin và quyền nhân viên) điều có thể truy cập vào phần bán hàng
* Đây là nơi tạo ra các hóa đơn
* Hướng dẩn sử dụng trang:

+ Nhập số điện thoại khách hàng vào, nếu đã có, thông tin khách hàng sẽ hiện ra, sau khi lưu thông tin khách hàng, ta nhấp vào món mà khách gọi, thông tin món và giá tiền sẽ vào bản. Nếu 1 món đó được nhấp 2 lần có nghĩa là khách gọi 2 phần.

+ Ta có thể in hóa đơn tại đây.

1. ***Thêm khách hàng***

* Tại đây, ta có thể thêm khách hàng vào hệ thống

1. **Các trường hợp giao diện**
   1. ***Giao diện đăng nhập, đổi mật khẩu, quên mật khẩu***

* Tài khoản, mật khẩu không tồn tại hoặc sai
* Chưa nhập 1 trong 2 ô
* Chưa nhập cả 2 ô
* Nhập lại mật khẩu không trùng với mật khẩu đã nhập
  1. ***Giao diện quản lí***
* Chưa đăng nhập vào bằng tài khoản admin
  1. ***Giao diện thêm món***
* Mã món bị trùng
* Thông tin món bị null
  1. ***Giao diện bán hàng***
* Thông tin khách hàng bị chùng
* Không có thông tin khách hàng
  1. ***Giao diện thêm nhân viên, khách hàng***
* Mã nhân viên bị trùng
* Thông tin nhân viên null

1. **Ưu nhược điểm của giao diện**
   1. ***Ưu điểm***

* Giao diện quen thuộc, dể sử dụng
* Trực quan trỏ rang, dể theo dỏi
* Giao diện thiết kế phẳn thông minh
* Tích hợp các thuật toán thông minh bên trong chương trình
  1. ***Nhược điểm***
* Chưa được đẹp
* Thiếu phần cho khách hàng sử dụng hệ thống