TRƯỜNG ĐẠI HỌC KIÊN GIANG

**KHOA THÔNG TIN & TRUYỀN THÔNG**

--------------------------------

**BÁO CÁO NIÊN LUẬN CƠ SỞ**

**Đề tài:**

**Xây dựng phần mềm quản lý quán cà phê**

SV thực hiện: Tăng Dương Nhật Huy

MSSV: 1501206039

Lớp: B15TT3

Người hướng dẫn: Đào Thị Phấn

*Tháng 10 năm 2018***ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ**

Nhận xét:

Điểm:…../10

……………….., ngày…. tháng ….. năm …..

GIẢNG VIÊN HƯỚNG DẪN

(ký và ghi rõ họ tên)

**Mục lục**

[**Chương 1. Tổng quan** 3](#_Toc526651774)

[**1.1** **Đặt vấn đề**: 3](#_Toc526651775)

[**1.2** **Hướng giải quyết** 3](#_Toc526651776)

[**1.3** **Kế hoạch thực hiện** 3](#_Toc526651777)

[**Chương 2. Cơ sở lý thuyết** 4](#_Toc526651778)

[**2.1** **Giới thiệu về chương trình** 4](#_Toc526651779)

[**2.2** **Giải thích các khái niệm.** 4](#_Toc526651780)

[**Chương 3. Kết quả ứng dụng** 6](#_Toc526651781)

[**3.1** **Đặc tả hệ thống** 6](#_Toc526651782)

[**3.2** **Các sơ đồ** 6](#_Toc526651783)

[**3.3** **Bảng ràng buộc toàn vẹn** 7](#_Toc526651784)

[**3.4** **Các sơ đồ** 8](#_Toc526651785)

[**3.5** **Một số hình ảnh của chương trình** 10](#_Toc526651786)

[**Chương 4. Kết luận và hướng phát triển** 12](#_Toc526651787)

[**Chương 5. Các tài liệu tham khảo** 13](#_Toc526651788)

**Chương 1. Tổng quan**

* 1. **Đặt vấn đề**:

Các quán cà phê nói chung hay các cửa hàng kinh doanh nói riêng đa phần ghi chép các giao dịch trong ngày bằng thủ công, lưu trữ các sổ sách ở dạng trang giấy. Điều này tốn rất nhiều công sức, tiền bạc, cũng như thời gian khi cần các hoạt động như quản lý, kiểm kê các mặt hàng, các giao dịch. Làm ảnh hưởng không nhỏ đến cửa hàng, là cho cửa hàng không thể tối đa được lợi nhuận kinh doanh.

* 1. **Hướng giải quyết**

Để giải quyết vấn về trên thì các cửa hàng nên áp dụng công nghệ thông tin vào công tác quản lý. Mọi dữ liệu về cửa hàng đều được ghi trên máy tính, khi cần có thể xem trực tiếp và nhanh chóng mà không mất nhiều thời gian.

Các nhu cầu cơ bản của một quán cà phê cần giải quyết trước là quản lý được số lượng hóa đơn trong ngày, hóa đơn theo bàn, quản lý được các hạng mục thức ăn mà cửa hàng đang kinh doanh, quản lý được doanh thu của cửa hàng. Khi xử lý được các yêu cầu trên thì phần mềm cơ bản là hoàn thiện cho một quán cà phê có quy mô nhỏ.

* 1. **Kế hoạch thực hiện**

Chương trình dự định sẽ được viết theo mô hình 3 lớp (Three layers) nên công việc đầu tiên là bắt đầu xây dựng hệ thống cơ sở dữ liệu. Mô hình cơ sở dữ liệu được trình bày ở phần sau.

Sau khi thiết kế hoàn thiện cơ sở dữ liệu, ta tiếp được đến phần thiết kê giao diện phần mềm. Giao diện người dùng được thiết kế giản đơn nhất hết mức có thể, làm cho người dùng có thể dễ làm quen với chương trình.

Sau phần thiết kế giao diện là đến phần lập trình các chức năng cho chương trình. Chương trình viết theo mô hình 3 lớp nên sẽ đi qua các công đoạn chung như: mô hình hóa các bảng trong cơ sở dữ liệu thành các lớp với các thuộc tính là các header của các bảng trong cơ sở dữ liệu, mỗi một lớp được mô hình sẽ đi cùng 1 lớp chức các phương thức để tương tác với cơ sở dữ liệu. Việc còn lại chỉ là xử lí các yêu cầu đơn giản trên giao diện đã được thiết kế từ trước. Sau khi hoàn thành chương trình thì sẽ bắt đầu công việc tìm các lỗi có thể xảy ra, sửa hoặc giấu nó giúp cho chương trình hoàn thiện hơn.

**Chương 2. Cơ sở lý thuyết**

* 1. **Giới thiệu về chương trình**

Ngôn ngữ lập trình: C#.

Công nghệ lập trình: Winfrom.

Kỹ thuật lập trình: Lập trình hướng đối tượng – OOP.

Mô hình chương trình: Mô hình 3 lớp – Three layers.

Mẫu thiết kế phần mềm: Singleton.

* 1. **Giải thích các khái niệm.**

**Winform** là công nghệ lập trình của Microsoft cho phép lập trình viên dễ dàng viết ra các chương trình dưới dạng đồ họa. Công nghệ này ra mắt năm 2006 và đã bị khai tử 2013 tuy nhiên vẫn còn được sử dụng rộng rãi do sự tiện dụng và mạnh mẽ của nó.

**Lập trình hướng đối tượng** là kỹ thuật lập trình lấy các đối tượng làm trọng tâm. Các đối tượng trong chương trình bao gồm thuộc tính và phương thức. Lập trình hướng đối tượng giúp tăng năng suất, giảm độ phức tạp của phần mềm cũng như giúp phần mềm dễ dàng bảo trì và nâng cấp tính năng.

**Mô hình 3 lớp** (Three layers) là mô hình thiết kế phần mềm, trong đó chương trình được chia ra 3 lớp tách biệt nhau, 3 lớp đó gồm:

+ Tầng hiển thị (Presentation): hiển thị các thành phần giao diện để tương tác với người dung.

+ Tầng xử lí (Business logic): thực hiện các phép tính, xử lí thông tin do tầng hiển thị gửi xuống, nếu yêu cầu xử lí cần tương tác với cơ sở dữ liệu thì yêu cầu đó sẽ được chuyển xuống tầng tiếp theo..

+ Tầng dữ liệu (Data): tương tác và lưu trữ dữ liệu mà phần mềm tương tác với các hệ quản trị cơ sở dữ liệu. Tầng này trả về dữ liệu cho tầng xử lý để tầng xử lý trả kết quả cho tầng hiển thị.

Mô hình 3 lớp thể hiện được ưu điểm khi được áp dụng trong các phần mềm quản lý do:

Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.

Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.

Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo.

Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.

Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.

**Mẫu thiết kế phần mềm** (Design pattern): Là các dạng mẫu thiết kế phần mềm được tạo ra nhằm giải quyết các vấn đề chung khi thiết kế phần mềm. Số lượng Design pattern hiện tại là 23 và còn đang tăng. Trong chương trình này có sử dụng một Design pattern là Singleton.

Khi thiết kế phần mềm, có một vấn đề thường gặp phải là các lớp tạo ra các đối tượng quá nhiều, khi muốn sử dụng một lớp thì phải khởi tạo đối tượng, điều này làm tốn thời gian, tăng số lượng dòng code, bộ nhớ chưa được tối ưu. Singleton sinh ra để giải quyết vấn đề bằng cách hạn chế lại số lượng các lớp có thể tạo ra các đối tượng, các lớp trong chương trình chỉ có thể tạo ra 1 và duy nhất chỉ 1 đối tượng (đối tượng này thường được gọi là Instance). Việc này giúp code trở nên vô cùng ngắn gọn, dễ hiểu, cái thiện được rất nhiều thời gian lập trình. Ngoài Singleton còn có rất nhiều Design pattern khác giúp khắc phục một số điểm bất cập khi phát triển phần mềm, tuy nhiên trong phạm vi chương trình này các vấn đề đó chưa xuất hiện.

**Chương 3. Kết quả ứng dụng**

* 1. **Đặc tả hệ thống**

Phần mềm cho phép người chủ có thể quản lý cửa hàng một cách dễ dàng, các mục có thể quản lý như:

Quản lý và ghi hóa đơn trên máy tính

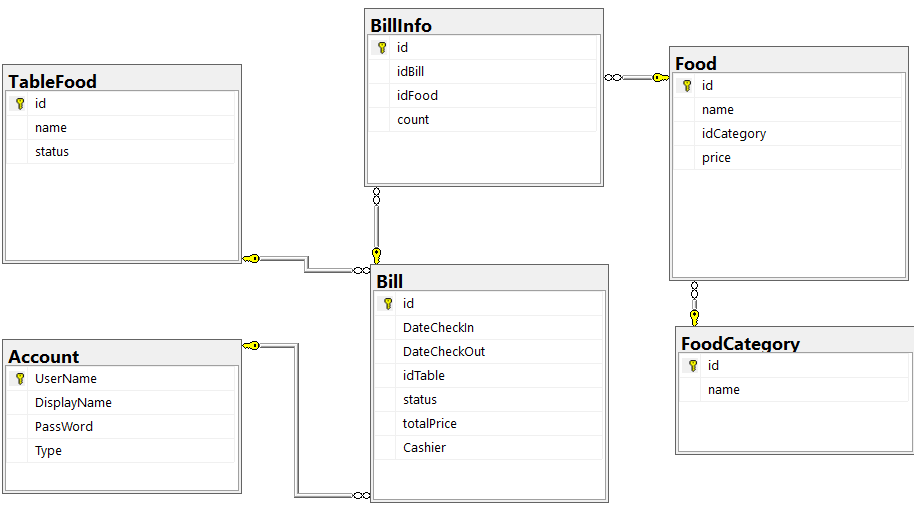
Quản lý menu bao gồm các danh mục, các món ăn có trong danh mục.

Quản lý các bàn có trong cửa hàng.

Tạo báo cáo doanh thu theo các mốc thời gian.

Chủ quán có thể tạo và phân quyền tài khoản cho nhân viên.

* 1. **Các sơ đồ**

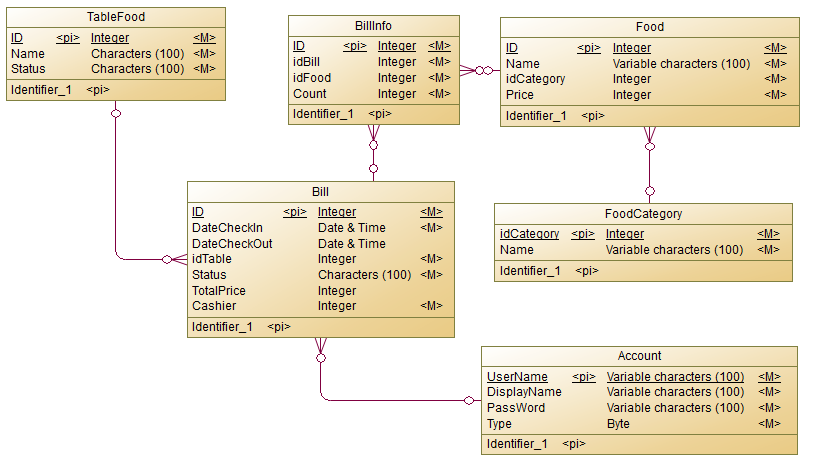


Sơ đồ cơ sở dữ liệu

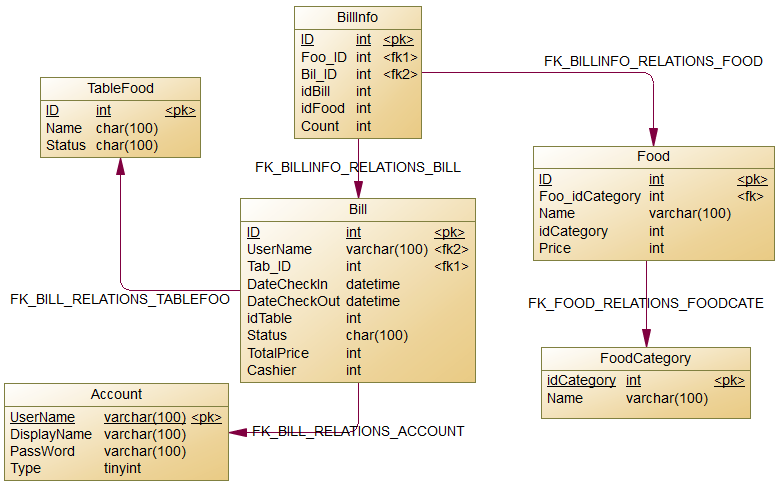
* 1. **Bảng ràng buộc toàn vẹn**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Account | |  |  |  |  |  |  |  |  |
| STT | Tên thuộc tính | Kiểu dữ liệu | Số kí tự | Giá trị mặc định | Khóa chính | Khóa ngoại | Not null | Unique | Diễn giải |
| 1 | Username | nvarchar | 100 |  | x |  |  |  | Tên đăng nhập |
| 2 | Displayname | nvarchar | 100 |  |  |  | x |  | Tên hiển thị |
| 3 | Password | nvarchar | 100 |  |  |  | x |  | Mật khẩu |
| 4 | Type | bit |  |  |  |  | x |  | Kiểu tài khoản |
|  |  |  |  |  |  |  |  |  |  |
| TableFood | |  |  |  |  |  |  |  |  |
| STT | Tên thuộc tính | Kiểu dữ liệu | Số kí tự | Giá trị mặc định | Khóa chính | Khóa ngoại | Not null | Unique | Diễn giải |
| 1 | ID | int |  |  | x |  |  |  | Số id bàn |
| 2 | Name | nvarchar | 100 |  |  |  | x |  | Tên bàn |
| 3 | Status | nvarchar | 100 | Trống |  |  | x |  | Trạng thái bàn |
|  |  |  |  |  |  |  |  |  |  |
| FoodCategory | |  |  |  |  |  |  |  |  |
| STT | Tên thuộc tính | Kiểu dữ liệu | Số kí tự | Giá trị mặc định | Khóa chính | Khóa ngoại | Not null | Unique | Diễn giải |
| 1 | ID | int |  |  | x |  |  |  | Số id của danh mục |
| 2 | Name | nvarchar | 100 |  |  |  | x | x | Tên danh mục |
|  |  |  |  |  |  |  |  |  |  |
| Food | |  |  |  |  |  |  |  |  |
| STT | Tên thuộc tính | Kiểu dữ liệu | Số kí tự | Giá trị mặc định | Khóa chính | Khóa ngoại | Not null | Unique | Diễn giải |
| 1 | ID | int |  |  | x |  |  |  | Số id món ăn |
| 2 | Name | nvarchar | 100 |  |  |  | x |  | Tên món ăn |
| 3 | idCategory | int |  |  |  | FoodCategory |  |  | Thuộc loại danh mục |
| 4 | Price | int |  | 0 |  |  |  |  | Giá bán |
|  |  |  |  |  |  |  |  |  |  |
| Bill | |  |  |  |  |  |  |  |  |
| STT | Tên thuộc tính | Kiểu dữ liệu | Số kí tự | Giá trị mặc định | Khóa chính | Khóa ngoại | Not null | Unique | Diễn giải |
| 1 | ID | int |  |  | x |  |  |  | Số id hóa đơn |
| 2 | DateCheckIn | datetime |  | NOW() |  |  |  |  | Giờ vào |
| 3 | DateCheckOut | datetime |  |  |  |  | x |  | Giờ ra |
| 4 | idTable | int |  |  |  |  |  |  | Bàn ghi hóa đơn |
| 5 | Status | bit | 100 | 0 |  |  |  |  | Trạng thái |
| 6 | TotalPrice | int |  |  |  |  | x |  | Tổng tiền |
| 7 | Cashier | int |  |  |  | Account |  |  | Thu ngân |
|  |  |  |  |  |  |  |  |  |  |
| BillInfo | |  |  |  |  |  |  |  |  |
| STT | Tên thuộc tính | Kiểu dữ liệu | Số kí tự | Giá trị mặc định | Khóa chính | Khóa ngoại | Not null | Unique | Diễn giải |
| 1 | ID | int |  |  | x |  |  |  | Số id |
| 2 | idBill | int |  |  |  | Bill |  |  | Hóa đơn |
| 3 | idFood | int |  |  |  | Food |  |  | Món gọi |
| 4 | Count | int |  |  |  |  |  |  | Số lượng |

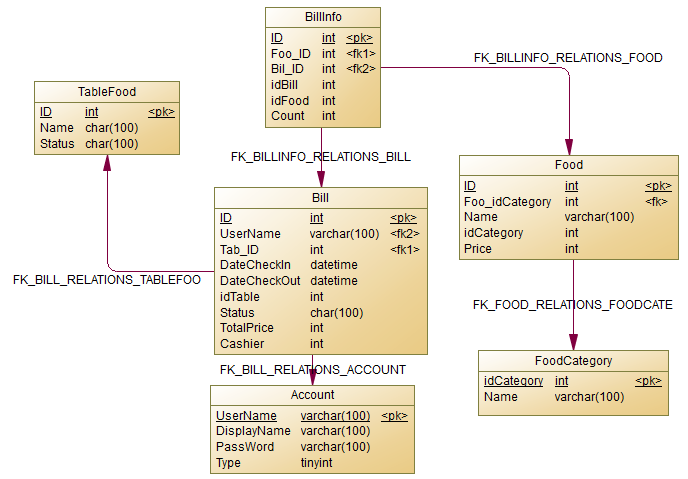
* 1. **Các sơ đồ**



Sơ đồ CDM - Conceptual Data Model

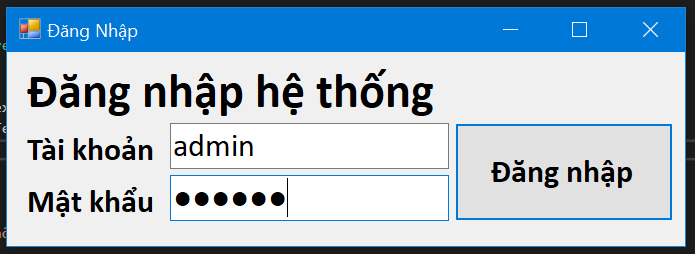


*Sơ đồ LDM - Logical Data Model*



Sơ đồ PDM - Physical Data Model

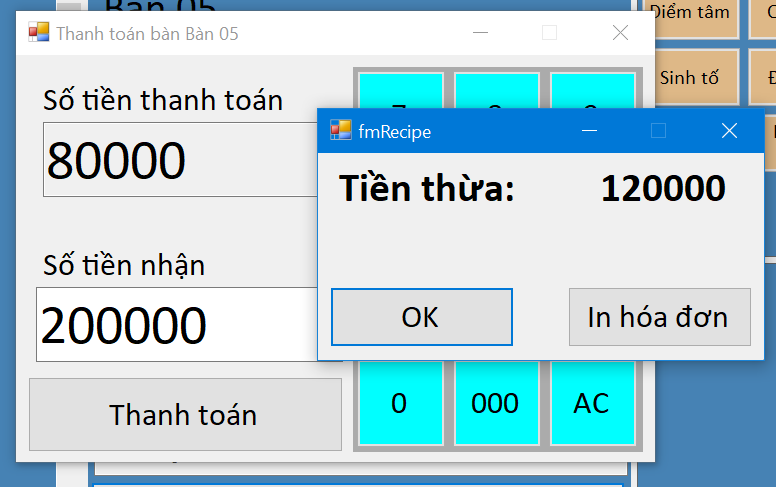
* 1. **Một số hình ảnh của chương trình**



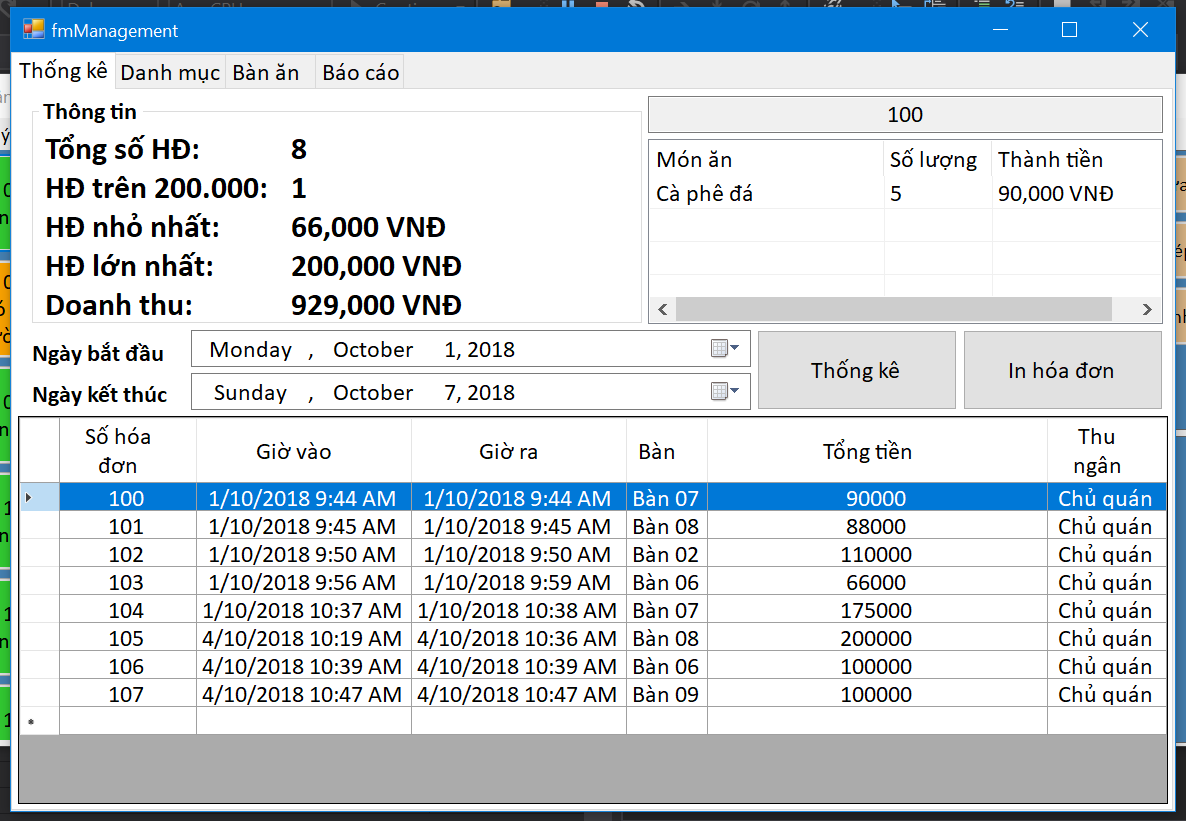
Giao diện đăng nhập



Giao diện chính của chương trình



Giao diện tính tiền



Giao diện thống kê

**Chương 4. Kết luận và hướng phát triển**

Phần mềm cơ bản đáp ứng các yêu cầu của việc quản lý của cửa hàng. Tuy nhiên vẫn còn số tính năng mà chương trình có thể them vào để phần mềm trở nên hoàn hảo hơn, như:

+ Chương trình có thể phát triển bằng cách kết nối vào một kho nguyên liệu, dựa vào số lượng các sản phẩm bán mà có thể tính được số lượng nguyên liệu tương ứng còn tồn tại trong kho, qua đó giải quyết được các vấn đề của nhà kho, giúp tăng hiệu quả của cửa hàng.

+ Các dữ liệu của cửa hàng có thể sao lưu trên hệ thống đám mây, dữ liệu được an toàn hơn và có thể khôi phục lại bất khi nào.

+ Phát triển quy mô của phần mềm lên, ví dụ như tương tác với một hệ thống khác nằm bên khâu pha chế sản phẩm, khi có thức ăn được gọi thì bên khâu pha chế cũng được nhận thông tin, giảm thời gian đợi chờ của khách, tăng tính chuyên nghiệp của cửa hàng.

**Chương 5. Các tài liệu tham khảo**

[1]Stackoverflow.com

[2]Howkteam.com