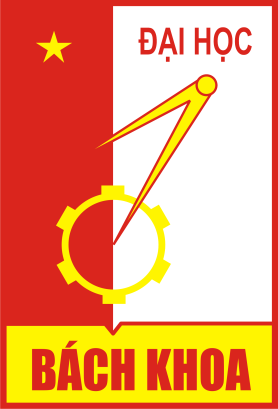
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────



BÁO CÁO MÔN: Cơ sở dữ liệu

**Quản lý siêu thị**

Sinh viên thực hiện : **Trần Mạnh Linh – 20132314**

**Lê Đình Phúc - 20132979**

**Lê Đình Hoàng Sơn - 20133305**

**Nguyễn Tiến Trường- 20134221**

**( Nhóm 2 )**

Lớp : **Việt Nhật B – K58**

Giáo viên hướng dẫn : TS. Trần Việt Trung

***Hà Nội, tháng 11 năm 2015***

**MỤC LỤC**

[Phần 1: Mở đầu 3](#_Toc437459501)

[1.1 Lý do chọn đề tài 3](#_Toc437459502)

[1.2 Đối tượng hướng tới. 3](#_Toc437459503)

[1.3 Phạm vi áp dụng. 3](#_Toc437459504)

[1.4 Nội dung nghiên cứu. 3](#_Toc437459505)

[1.5 Các phần mềm hỗ trợ. 4](#_Toc437459506)

[1.6 Phân công công việc. 4](#_Toc437459507)

[Phần 2: Chi tiết đề tài 5](#_Toc437459508)

[2.1. Thiết kế hệ cơ sở dữ liệu. 5](#_Toc437459509)

[2.1.1. Mô tả: 5](#_Toc437459510)

[2.1.2. Sơ đồ thực thể liên kết. 6](#_Toc437459511)

[2.1.3. Sơ đồ quan hệ. 7](#_Toc437459512)

[2.1.4. Tổng quan các bảng sử dụng: 7](#_Toc437459513)

[2.1.5. Cài đặt với hệ cơ sở dữ liệu mysql. 7](#_Toc437459514)

[2.2. Tăng hiệu suất, bảo mật, tính toàn vẹn của cơ sở dữ liệu. 10](#_Toc437459515)

[2.2.1. Đặc điểm phần mềm. 10](#_Toc437459516)

[2.2.2. Thiết kế kiểm soát dữ liệu. 10](#_Toc437459517)

[2.2.3 Chi tiết cài đặt. 12](#_Toc437459518)

[END $$ 15](#_Toc437459519)

[2.2.4. Tăng hiệu suất: 16](#_Toc437459520)

[2.2.5. Chi tiết cài đặt: 16](#_Toc437459521)

[2.3. Thiết kế giao diện phần mềm. 17](#_Toc437459522)

[2.4. Lập trình ứng dụng. 28](#_Toc437459523)

[2.4.1. Cơ sở các thư viện, kiến thức sử dụng. 28](#_Toc437459524)

[2.4.2. Chi tiết mã nguồn chương trình. 28](#_Toc437459525)

[Phần 3. Chức năng, hướng dẫn sử dụng ứng dụng. 32](#_Toc437459526)

[3.1. Yêu câu môi trường trước khi sử dụng ứng dụng. 32](#_Toc437459527)

[3.2. Chức năng, hướng dẫn sử dụng 32](#_Toc437459528)

[3.2.1. Chức năng, hướng dẫn sử dụng với người quản lý. 33](#_Toc437459529)

[3.2.2. Chức năng, hướng dẫn sử dụng với thu ngân. 39](#_Toc437459530)

[Phần 4. Hướng phát triển phần mềm 40](#_Toc437459531)

[Phần 5. Kết luận 41](#_Toc437459532)

Phần 1: Mở đầu

## 1.1 Lý do chọn đề tài

Hiện nay, các cửa hàng vừa và nhỏ ở Việt Nam phần nhiều vẫn còn quản lý qua giấy tờ, việc thanh toán vẫn còn thông lời nói. Không bảo mật, thiếu pháp lý rõ ràng và hiệu quả thấp. Một phần nguyên nhân của hiện tượng này đó là do những phần mềm quản lý, thu ngân vẫn còn khá ít, chức năng nghèo nàn, giao diện sử dụng thiếu tính thân thiện, cần phải mua hoặc đầu tư cơ sở phần cứng lớn. Nhận thấy điều này, nhóm đã quyết định chọn đề tài nhằm mục đính xây dựng 1 phần mềm quản lý và thu ngân mới. Giao diện thân thiện hơn, cơ sở dữ liệu rõ ràng, có tính an toàn cao.

## 1.2 Đối tượng hướng tới.

Quản lý cơ sở dữ liệu của 1 cửa hàng.

Hỗ trợ tính toán, thu, nhập hóa đơn.

Hỗ trợ tạo các đợt khuyến mại, sự kiện giảm giá.

Quản lý thống kê hiệu năng, doang thu của cửa hàng và nhân viên.

## 1.3 Phạm vi áp dụng.

Tất cả các cửa hàng hoạt động theo phương thức:

* Nhập hàng qua bên thứ 2.
* Bán những hàng hóa đếm được, có giá bán cố định.
* Khuyến mại trực tiếp vào giá tiền hoặc sản phẩm khuyến mại được đính kèm cùng sản phẩm.

## 1.4 Nội dung nghiên cứu.

Thiết kế, xây 1 cơ sở dữ liệu trên hệ quản trị cơ sở dữ liệu MySQL

Thiết kế 1 bộ giao diện phần mềm, thân thiện, dễ tương tác.

Tăng hiệu suất, bảo mật,tính toàn vẹn cho cơ sở dữ liệu.

Xây dựng phần mềm bằng ngôn ngữ lập trình java.

## 1.5 Các phần mềm hỗ trợ.

Eclipse Java EE IDE for Web Developers.

Version: Mars.1 Release (4.5.1)

Build id: 20150924-1200

Workbench 6.3 CE

mysqlserver

DesignerVista 7.0

github

## 1.6 Phân công công việc.

* Nguyễn Tiến Trường: Chọn, nghiên và phân tích đề tài, thiết kế giao diện người dùng, lập trình ứng dụng.
* Trần Mạnh Linh: Thiết kế cơ sở dữ liệu, tối ưu hóa, tăng tính bảo mật, toàn vẹn của cơ sở dữ liệu.
* Lê Đình Phúc: Thiết kế cơ sở dữ liệu, lập trình ứng dụng, viết báo cáo.
* Lê Đình Hoàng Sơn: Lập trình ứng dụng, viết báo cáo, làm slide thuyết trình.

Phần 2: Chi tiết đề tài

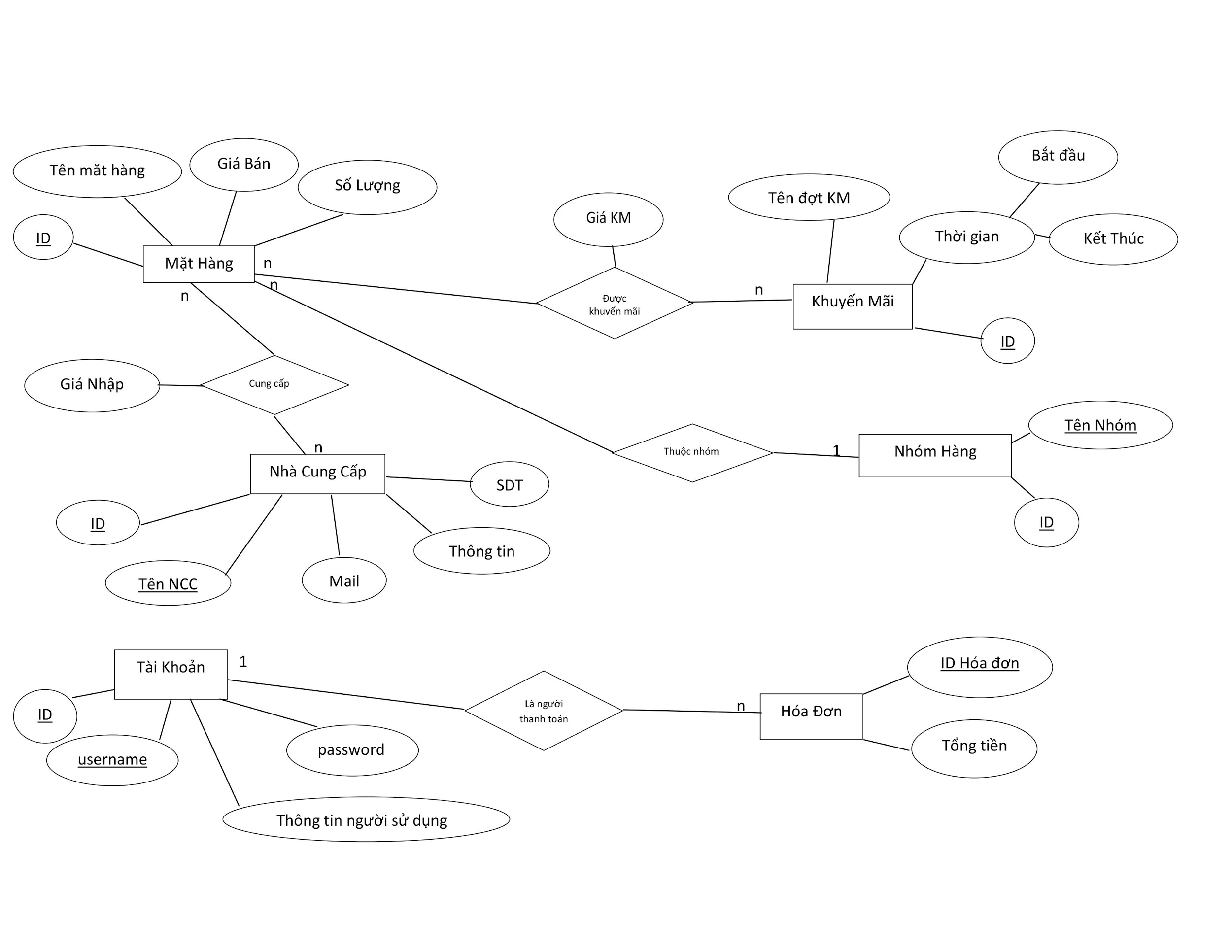
## 2.1. Thiết kế hệ cơ sở dữ liệu.

### 2.1.1. Mô tả:

Cơ sở dữ liệu bao gồm thông tin về 1 cửa hàng gồm: các mặt hàng, các nhà cung cấp, nhân viên, các sự kiện khuyến mại, hóa đơn thanh toán).

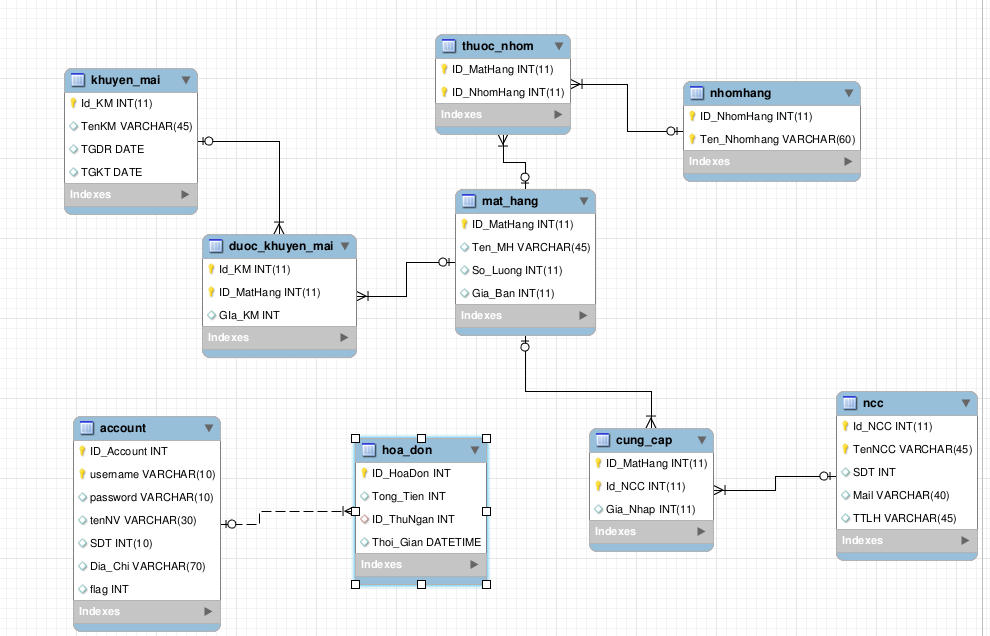
* Thông tin của một mặt hàng bao gồm: mã số, tên, giá bán.
* Các sản phẩm sẽ được cung cấp qua nhà cung cấp. một mặt hàng có thể được cung cấp bởi nhiều nhà cung cấp và giá khác nhau.
* Thông tin của một nhà cung cấp bao gồm: mã số, tên, thông tin liên hệ.
* Mỗi sản phẩm thuộc vào một hoặc nhiều nhóm hàng. Một nhóm hàng có thể có nhiều mặt hàng.
* Cửa hàng có các đợt khuyến mại. Mỗi đợt khuyến mại được đánh dấu bằng một mã duy nhất, thời gian bắt đầu và kết thúc mỗi đợt sự kiện. Trong mỗi đợt sự kiện, một vài mặt hàng sẽ được giảm giá, và các mặt hàng cũng có thể được giảm giá nhiều lần trong các sự kiện khác nhau.
* Để quản lý và phân quyền làm việc cho các nhân viên, ta tạo ra các tài khoản đăng nhập vào hệ thống. Gồm các thông tin: mã nhân viên, tên tài khoản, mật khẩu sử dụng, tên nhân viên, thông tin liên hệ, trạng thái hoạt động.

### 2.1.2. Sơ đồ thực thể liên kết.



*Hình 1: Sơ đồ thực thể liên kết của cơ sở dữ liệu.*

### 2.1.3. Sơ đồ quan hệ.



*Hình 2: Sơ đồ quan hệ của cơ sở dữ liệu.*

### 2.1.4. Tổng quan các bảng sử dụng:

khuyen\_mai( Id\_KM , Ten\_KM , TGDR , TGKT);

mat\_hang( ID\_Mathang, Ten\_MH, So\_Luong, Gia\_Ban);

nhomhang( ID\_NhomHang, Ten\_NhomHang);

ncc( Id\_NCC, TenNCC, SDT, Mail, TTLH);

account ( ID\_Account, username, password, tenNV, SDT, Dia\_Chi, flag);

hoa\_don( ID\_HoaDon,Tong\_Tien, ID\_ThuNgan, Thoi\_Gian);

duoc\_khuyen\_mai( Id\_KM, ID\_MatHang, Gia\_KM);

cung\_cap (ID\_MatHang, Id\_NCC, Gia\_Nhap);

thuoc\_nhom( ID\_MatHang, ID\_NhomHang);

### 2.1.5. Cài đặt với hệ cơ sở dữ liệu mysql.

* *Bảng khuyến mại.*

CREATE TABLE IF NOT EXISTS `khuyen\_mai` (

`Id\_KM` INT(11),

`Ten\_KM` VARCHAR(45),

`TGDR` DATE,

`TGKT` DATE,

PRIMARY KEY (`Id\_KM`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng mặt hàng*

CREATE TABLE if not exists `mat\_hang` (

`ID\_MatHang` int(11),

`Ten\_MH` varchar(45),

`Gia\_Ban` int(11),

`Soluong` int(11),

PRIMARY KEY (`ID\_MatHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng nhóm hàng:*

CREATE TABLE IF NOT EXISTS `nhomhang` (

`ID\_NhomHang` INT(11) ,

`Ten\_Nhomhang` VARCHAR(60) ,

PRIMARY KEY (`ID\_NhomHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng nhà cung cấp:*

CREATE TABLE IF NOT EXISTS `ncc`(

`Id\_NCC` INT(11),

`Ten\_NCC` VARCHAR(45),

`SDT` INT,

`Mail` VARCHAR(40),

`TTLH` VARCHAR(45),

PRIMARY KEY (`Id\_NCC`,`Ten\_NCC`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng tài khoản nhân viên.*

CREATE TABLE if not exists `account` (

`ID\_Account` int ,

`username` varchar(10),

`password` varchar(10) ,

`tenNV` varchar(30),

`SDT` int(10),

`Dia\_Chi` varchar(60),

`flag` int default 0,

PRIMARY KEY (`ID\_Account`,`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng hóa đơn.*

CREATE TABLE if not exists `hoa\_don` (

`ID\_HoaDon` int not null AUTO\_INCREMENT,

`Tong\_Tien` int,

`ID\_ThuNgan` int,

`Thoi\_Gian` datetime ,

PRIMARY KEY (`ID\_HoaDon`),

FOREIGN KEY (`ID\_ThuNgan`) REFERENCES `account`(`ID\_Account`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng mặt hàng được khuyến mại.*

CREATE TABLE if not exists `duoc\_khuyen\_mai` (

`Id\_KM` int(11),

`ID\_MatHang` int(11),

`Gia\_KM` int,

FOREIGN KEY (`Id\_KM`) REFERENCES `khuyen\_mai`(`Id\_KM`) ,

FOREIGN KEY (`ID\_MatHang`) REFERENCES `mat\_hang`(`ID\_MatHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng cung cấp*

CREATE TABLE if not exists `cung\_cap` (

`ID\_MatHang` int(11) ,

`Id\_NCC` int(11) ,

`Gia\_Nhap` int(11) ,

PRIMARY KEY (`ID\_MatHang`),

FOREIGN KEY(`ID\_MatHang`) REFERENCES `mat\_hang`(`ID\_MatHang`),

FOREIGN KEY(`Id\_NCC`)REFERENCES `ncc`(`Id\_NCC`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

* *Bảng thuộc nhóm.*

CREATE TABLE if not exists `thuoc\_nhom` (

`ID\_MatHang` int(11) ,

`ID\_NhomHang` int(11),

PRIMARY KEY (`ID\_MatHang`),

FOREIGN KEY(`ID\_MatHang`) REFERENCES `mat\_hang`(`ID\_MatHang`),

FOREIGN KEY(`ID\_NhomHang`) REFERENCES `nhomhang`(`ID\_NhomHang`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## 2.2. Tăng hiệu suất, bảo mật, tính toàn vẹn của cơ sở dữ liệu.

### 2.2.1. Đặc điểm phần mềm.

* Phần mềm sẽ phải truy suất, tìm kiếm thông tin rất nhiều từ bảng mặt hàng, thông qua mã số mặt hàng.
* Phần mềm sẽ hỗ trợ truy xuất thêm, sửa, xóa dữ liệu từ tất cả các bảng. Dữ liệu đàu vào là dạng chuỗi kí tự, khó kiểm soát.
* Cần phải bảo mật, tăng tính chính xác của các thông tin liên quan đến hóa đơn, số lượng hàng hóa, giá cả nhập vào, bán ra.
* Cần phải tăng hiệu năng các của các biểu thức truy vấn qua mã hoặc tên sản phẩm.

### 2.2.2. Thiết kế kiểm soát dữ liệu.

**Tổng quan:**

Trong qúa trình tạo bảng, tại mỗi trường đều có 2 lựa chọn : 1 là thêm not null vào cuối trường, 2 là để gía trị mặc định cho trường là null.

  Điều này tạo ra một số lỗ hổng trong câu lệnh truy vẫn cũng như câu lệnh thêm dữ liệu vào bảng.

ví dụ bảng bảng ncc:

Nếu thêm not null vào cuối mỗi trường thì khi sử dụng câu lệnh "insert into ncc (`ID\_NCC`) values (1);" ta sẽ nhận được 1 hàng gía trị mới với ID\_NCC = 1 và các trường còn lại là " ".

Còn nếu ngược lại không thêm not null vào cuối mỗi trường thì sẽ nhận được 1 hàng gía trị với ID\_NCC =1 còn các trường khác là NULL.

-> việc này có ảnh hưởng không nhỏ đến việc thực hiện chương trình cũng như việc quản lý các sản phẩm.

Ngoài ra, khi thưc hiện câu lệnh insert có một số trường hợp sinh ra cảnh báo, nhưng vẫn cho phép thêm dữ liệu vào bảng.( ví dụ như thêm ngày 60 tháng 30 năm 2015 vào bảng thì sẽ có cảnh báo, và khi thêm thành công vào bảng thì trường này có gía trị 00-00-0000 ) chính vì thế việc quản lý các cảnh báo này cũng rất quan trọng.

  Từ những điều trên, nhóm đã thống nhất sử dụng trigger để quản lý việc nhập dữ liệu cho bảng để đảm bảo tính thống nhất của dữ liệu.

**Thực hiện:**

* **Trigger** quản lý việc insert dữ liệu vào bảng khuyen\_mai với các điều kiện :
  + Các trường không được để trống.
  + TGDR và TGKT theo đúng dạng năm/tháng/ngày với ngày phải trong khoảng ngày của tháng hiện tại, tháng trong khoảng 0 đến 12.

(ví dụ: chỉ cho phép ngày 29/2 của năm nhuận )

* + TGDR < TGKT

Vi phạm 1 trong các điều kiện trên sẽ rollback về lúc trước khi insert giá trị mới vào bảng.

* **Trigger** quản lý việc insert dữ liệu vào bảng ncc với điều kiện :
  + Các trường không được trống.
  + Id\_NCC > 0
  + Trường mail phải có dạng "....@.... .com ...."

Vi phạm một trong các điều kiện trên sẽ rollback về trước khi insert dữ liệu.

* **Trigger** quản lý việc insert dữ liệu vào bảng nhomhang  với điều kiện :
  + Các trường không được trống và không được trùng nhau.
* **Trigger** quản lý việc insert dữ liệu vào bảng mat\_hang với các điều kiện:
  + Các trường không được rỗng.
  + ID\_MatHang >0, SoLuong >0, Gia\_Ban >=1000

Nếu 1 điều kiện không thỏa mãn, sẽ ném ra 1 lỗi và tự động rollback về trạng thái bảng lúc trước khi insert dữ liệu mới.

* **Trigger** quản lý việc insert vào bảng cung\_cap với các điều kiện:
  + Các trường ID\_MatHang và Id\_NCC phải tồn tại theo ràng buộc khóa ngoại.
  + Các trường không được trống.
  + Gia\_Nhap >= 1000
  + ID\_MatHang không được trùng lặp.

Rollback nếu vi phạm 1 trong các điều kiện trên.

\* Ngoài ra còn có ràng buộc khóa ngoại ở ID\_MatHang và Id\_NCC nên nếu giá trị nhập vào mà không tồn tại trong các bảng mà nó tham chiếu tới thì sẽ không được thêm vào bảng cung\_cap.(trigger xử lý  thêm để chặt chẽ)

\* Ràng buộc về khóa chính để đảm bảo 1 mặt hàng có thể được cung cấp bởi nhiều nhà cung cấp.(trigger xử lý thêm để chặt chẽ)

* **Trigger** quản lý việc nhập dữ liệu vào bảng thuoc\_nhom.
  + Nếu không tồn tại một trong 2 trường thì bỏ qua insert.
  + Nếu có 2 mặt hàng cùng tên thì ID\_MatHang sẽ khác nhau, tuy nhiên nếu ID\_NhomHang không giống nhau thì loại bỏ.

\* Ngoài ra còn có các ràng buộc về khóa chính để đảm bảo mỗi mặt hàng chỉ được thuộc 1 nhóm hàng.

* **Trigger** quản lý việc nhập dữ liệu vào bảng duoc\_khuyen\_mai.
  + Không có ràng buộc  về khóa chính để đảm bảo trong một đợt khuyến mãi có nhiều sản phẩm được khuyến mãi và một sảm phẩm có thể được khuyến mãi trong nhiều đợt.
  + Tuy nhiên sẽ xảy ra trường hợp nhập sai, vd như nhập 2 giá khuyến mại khác nhau cho cùng 1 sản phẩm trong 1 đợt khuyến mãi. Chính vì thế nên phải sử dụng trigger để giải quyết.
  + Trigger đảm bảo cho không trường nào được trống, và trong cùng 1 đợt khuyến mại, mặt hàng đó chỉ có 1 giá khuyến mại duy nhất.
* **Trigger** quản lý việc nhập vào bảng account với các điều kiện. Chỉ cho insert nếu dữ liệu đầu vào thỏa mãn các điều kiện.
  + Riêng các trường ID\_Account và username không được phép trùng nhau.
  + các trường không được trống
  + Tên tài khoản và mật khẩu phải trên 6 kí tự.

### 2.2.3 Chi tiết cài đặt.

* Trigger bf\_khuyenmai\_insert sẽ quản lý việc insert dữ liệu vào bảng khuyen\_mai.

CREATE TRIGGER bf\_khuyenmai\_insert

BEFORE INSERT ON khuyen\_mai

FOR EACH ROW

BEGIN

        IF (NOT DAYOFMONTH(new.TGDR) OR NOT DAYOFMONTH(new.TGKT)OR         new.TGKT IS NULL OR new.TGDR  IS NULL  OR new.TGKT <= new.TGDR OR   new.Id\_KM <=0 OR new.Ten\_KM IS NULL) THEN SIGNAL SQLSTATE '45001';

        END IF;

END $$

* Trigger bf\_thuocnhom\_insert quản lý việc nhập dữ liệu vào bảng thuoc\_nhom.

CREATE TRIGGER bf\_thuocnhom\_insert

BEFORE INSERT ON thuoc\_nhom

FOR EACH ROW

BEGIN

declare total int default 0;

select tn.ID\_NhomHang from thuoc\_nhom tn natural join mat\_hang mh where mh.Ten\_MH = all (select Ten\_MH from mat\_hang where ID\_MatHang = new.ID\_MatHang) and mh.ID\_MatHang != new.ID\_MatHang into total;

if(total != NULL and total != new.ID\_NhomHang) then signal sqlstate '45001';

        IF(new.ID\_MatHang IS NULL OR new.ID\_NhomHang IS NULL) THEN SIGNAL SQLSTATE '45001';

        END IF;

END $$

* Trigger bf\_ncc\_insert sẽ quản lý việc insert dữ liệu vào bảng ncc.

CREATE TRIGGER bf\_ncc\_insert

BEFORE INSERT ON ncc

FOR EACH ROW

BEGIN

        IF (new.Mail NOT LIKE "%@%.com%" OR new.Id\_NCC IS NULL OR new.Ten\_NCC IS NULL OR new.SDT IS NULL OR new.Mail IS NULL OR new.TTLH IS NULL OR new.Id\_NCC <=0) THEN SIGNAL SQLSTATE '45001';

        END IF;

END $$

* Trigger bf\_nhomhang\_insert sẽ quản lý việc insert dữ liệu vào bảng nhom\_hang.

CREATE TRIGGER bf\_nhomhang\_insert

BEFORE INSERT ON ncc

FOR EACH ROW

BEGIN

        IF (ID\_NhomHang IS NULL OR ID\_NhomHang <= 0 OR Ten\_NhomHang IS NULL) THEN SIGNAL SQLSTATE '45001';

        END IF;

END $$

* Trigger bf\_mathang\_insert sẽ quản lý việc insert dữ liệu vào bảng mat\_hang.

CREATE TRIGGER bf\_mathang\_insert

BEFORE INSERT ON mat\_hang

FOR EACH ROW

BEGIN

        IF(new.ID\_MatHang IS NULL OR new.Gia\_Ban IS NULL OR new.SoLuong IS NULL OR new.ID\_Mathang <= 0 OR new.Ten\_MH IS NULL OR new.Gia\_Ban <=1000 OR new.Soluong <=0) THEN SIGNAL SQLSTATE '45001';

        END IF;

IF (new.Ten\_NhomHang in(select Ten\_NhomHang from nhomhang) )THEN SIGNAL SQLSTATE '45001';

END IF;

IF(new.ID\_NhomHang in(select ID\_NhomHang from nhomhang)) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

* Trigger bf\_cungcap\_insert quản lý việc insert vào bảng cung\_cap.

CREATE TRIGGER bf\_cungcap\_insert

BEFORE INSERT ON cung\_cap

FOR EACH ROW

BEGIN

IF(NOT new.ID\_MatHang OR NOT new.Id\_NCC OR  new.Gia\_Nhap IS NULL OR new.Gia\_Nhap < 1000) THEN SIGNAL SQLSTATE '45001';

END IF;

IF(new.ID\_MatHang in (select cc.ID\_MatHang from cung\_cap cc)) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

* Trigger bf\_thuocnhom\_insert quản lý việc nhập dữ liệu vào bảng thuoc\_nhom.

CREATE TRIGGER bf\_thuocnhom\_insert

BEFORE INSERT ON thuoc\_nhom

FOR EACH ROW

BEGIN

         IF(new.ID\_MatHang IS NULL OR new.ID\_NhomHang IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

declare total int;

 select tn.ID\_NhomHang from thuoc\_nhom tn natural join mat\_hang mh where mh.Ten\_MH = all (select Ten\_MH from mat\_hang where ID\_MatHang = new.ID\_MatHang) and mh.ID\_MatHang != new.ID\_MatHang into total;

   if(total != NULL and total != new.ID\_NhomHang) then signal sqlstate '45001';

END IF;

END $$

* Trigger  bf\_duockhuyenmai\_insert quản lý việc nhập dữ liệu vào bảng duoc\_khuyen\_mai.

CREATE TRIGGER bf\_duockhuyenmai\_insert

BEFORE INSERT ON duoc\_khuyen\_mai

FOR EACH ROW

BEGIN

DECLARE TOTAL INT;

declare total int;

SELECT a.Gia\_KM FROM duoc\_khuyen\_mai a WHERE a.Id\_KM = new.Id\_KM AND a.ID\_MatHang = new.ID\_MatHang INTO total;

IF(new.Id\_KM IS NULL OR new.ID\_MatHang IS NULL OR new.Gia\_KM IS NULL OR new.Gia\_KM <1000) THEN SIGNAL SQLSTATE '45001';

END IF;

IF(total is not NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

END $$

* trigger bf\_account\_insert quản lý việc nhập vào bảng account.

CREATE TRIGGER bf\_account\_update

BEFORE INSERT ON account

FOR EACH ROW

BEGIN

if(new.ID\_Account in (select account.ID\_Account from account)) then signal sqlstate '45001' set message\_text = "Trung ID\_Account";

end if;

if(new.username in (select account.username from account)) then signal sqlstate '45001' set message\_text = "Trung username";

end if;

IF(new.ID\_Account IS NULL OR new.ID\_Account <=0 OR new.username IS NULL OR new.password IS NULL OR new.tenNV IS NULL OR new.SDT IS NULL OR new.Dia\_Chi IS NULL) THEN SIGNAL SQLSTATE '45001';

END IF;

IF(length(new.username) <6 OR length(new.password) <6) THEN SIGNAL SQLSTATE '45001';

end if;

if(new.flag != 0) Then SIGNAL SQLSTATE '45001' set message\_text ="flag != 0";

END IF;

END $$

* trigger bf\_hoadon\_insert quản lý việc nhập vào bảng hoa\_don.

CREATE TRIGGER bf\_hoadon\_insert

BEFORE INSERT ON hoa\_don

FOR EACH ROW

BEGIN

        IF( new.Tong\_Tien IS NULL OR new.ID\_ThuNgan IS NULL OR new.Thoi\_Gian IS NULL) THEN SIGNAL SQLSTATE '45001';

        END IF;

## END $$

### 2.2.4. Tăng hiệu suất:

- Với một cơ sở dữ liệu lớn như quản lý siêu thị thì việc sử dụng index sẽ giups cải thiện đáng kể tốc độ truy suất bảng, hay duyệt tìm kiếm các gía trị.

- Cụ thể, đánh index tại các bảng account, mat\_hang, nhomhang và ncc. Tất cả các câu lệnh thêm index được lưu trữ trong 1 store procedure.

- Ngoài ra, một tài khoản người sử dụng có 3 trạng thái ( gồm online, offline, deleted) nên trước và sau mỗi phiên làm việc của từng account, việc update lại trạng thái tài khoản diễn ra thường xuyên. Câu lệnh này được đặt vào 1 store procedure để tăng hiệu suất sử dụng.

### 2.2.5. Chi tiết cài đặt:

- Procedure setindex nhận đầu vào là một chuỗi kí tự, nếu trùng với tên của 1 trong 4 bảng account, ncc, mat\_hang, nhomhang thì sẽ thực hiện lệnh add index.

CREATE PROCEDURE setindex( in tablename varchar(20))

BEGIN

IF(tablename ='account') THEN ALTER TABLE account ADD INDEX (ID\_Account, username);

END IF;

IF(tablename = 'ncc') THEN ALTER TABLE ncc ADD INDEX (Id\_NCC);

END IF;

IF (tablename ='mat\_hang') THEN ALTER TABLE mat\_hang ADD INDEX (ID\_MatHang);

END IF;

IF (tablename = 'nhomhang') THEN ALTER TABLE nhomhang ADD INDEX (ID\_NhomHang);

END IF;

END $$

- Procedure updateflag nhận đầu vào là 2 tham số kiểu nguyên. Tham số đầu tiên đại diện cho id của tài khoản và tham số thứ 2 là trạng thái mới của tài khoản. (1 là online, 0 là offline, -1 là deleted)

CREATE PROCEDURE( in id int, in status int)

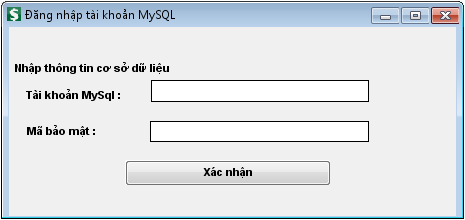
begin

if(status in (1,0,-1)) then update account set flag = status where ID\_Account = id;

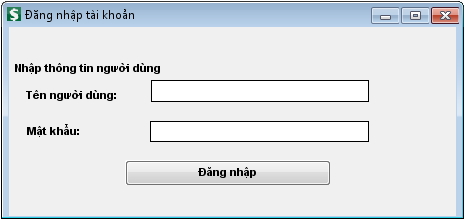
end if;

end $$

## 2.3. Thiết kế giao diện phần mềm.



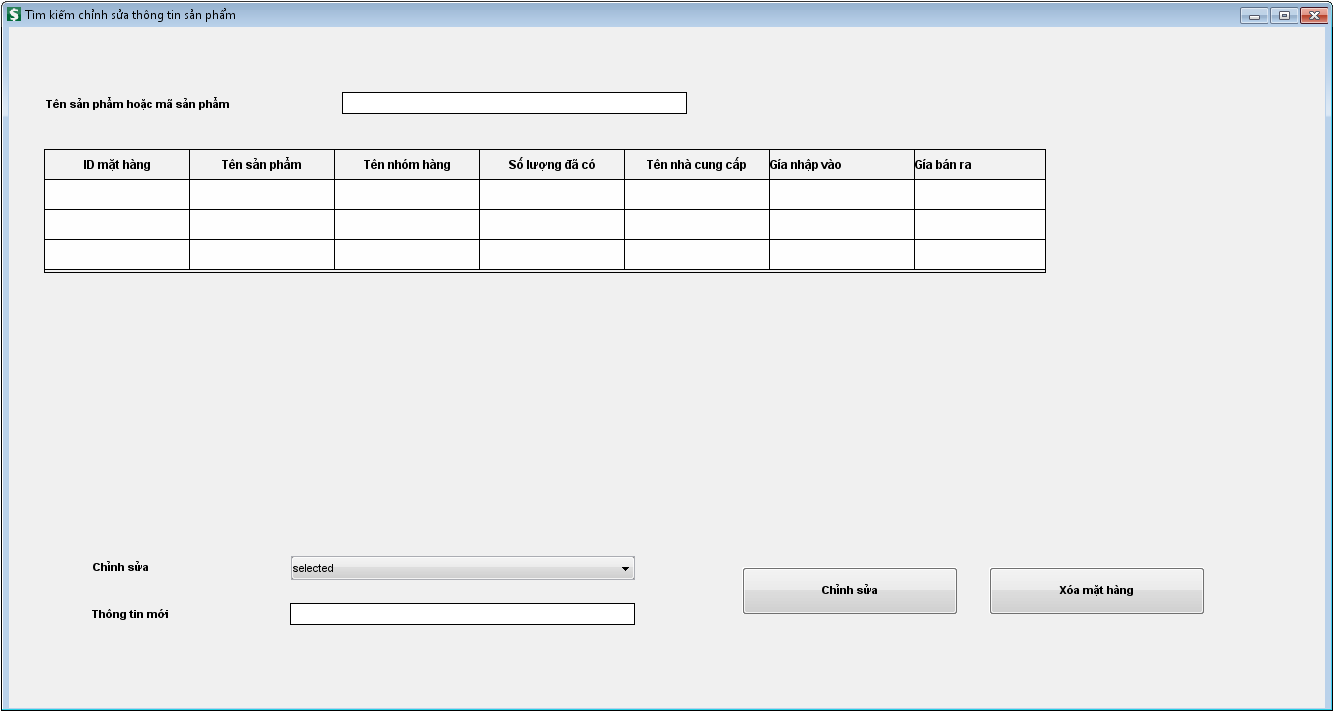
*Hình 1: màn hình đăng nhập Mysql*



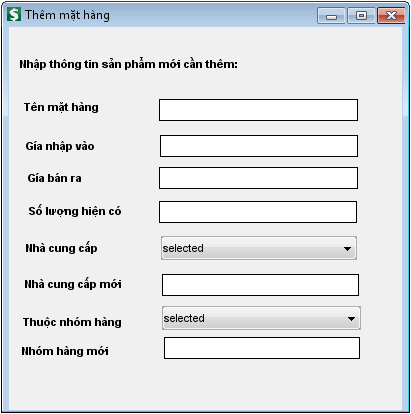
*Hình 2: Màn hình đăng nhập tài khoản*



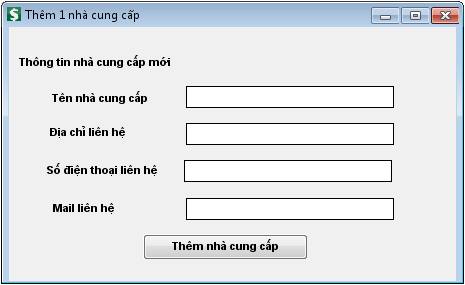
*Hình 3: Màn hình quản lý chính*



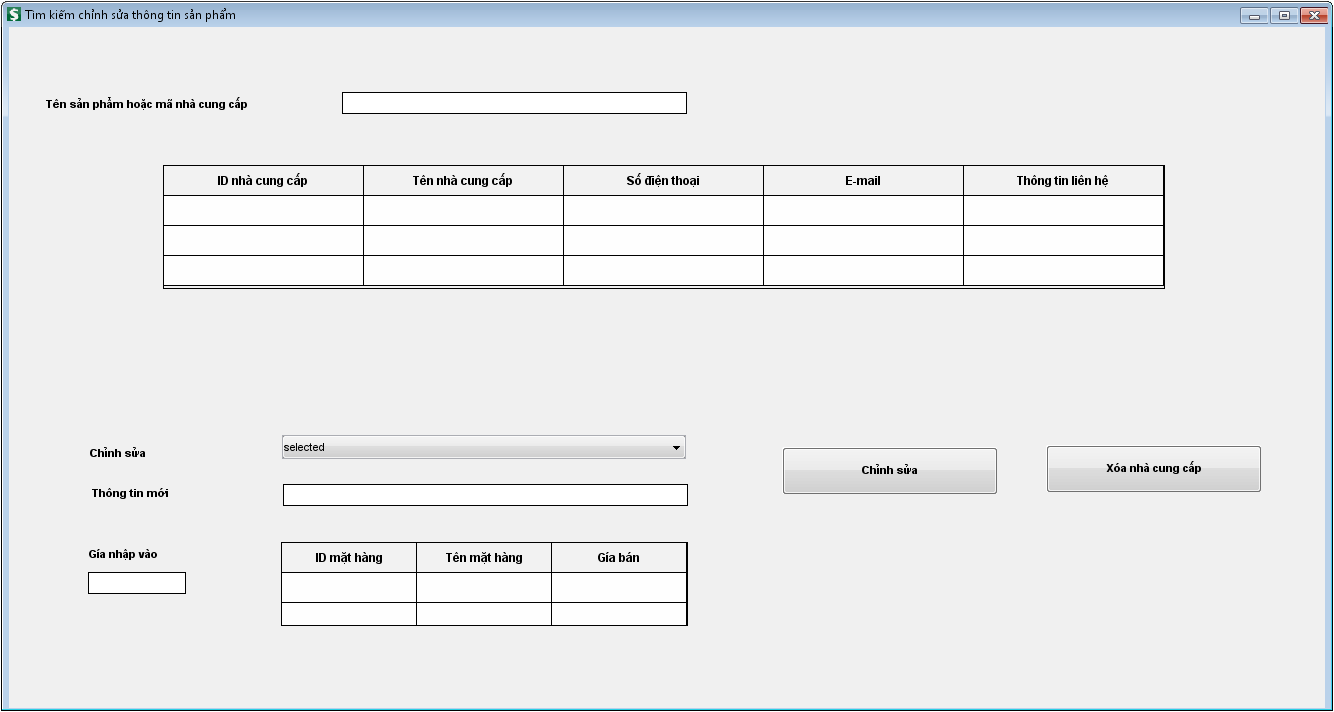
*Hình 4: Màn hình tìm kiếm chỉnh sửa thông tin sản phẩm*



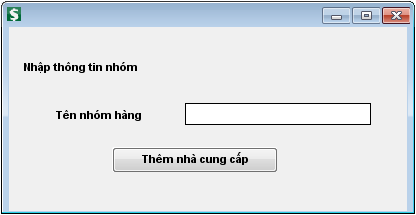
*Hình 5: Màn hình thêm 1 mặt hàng mới*



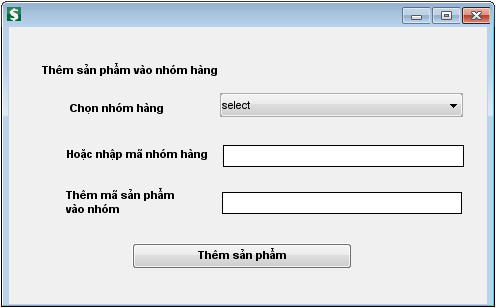
*Hình 6: Màn hình thêm 1 nhà cung cấp*



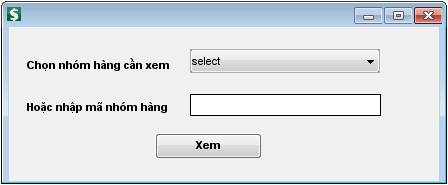
*Hình 7: Màn hình xem, chỉnh sửa nhà cung cấp*



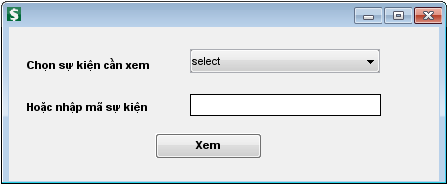
*Hình 8: Màn hình xác nhận nhà cung cấp*



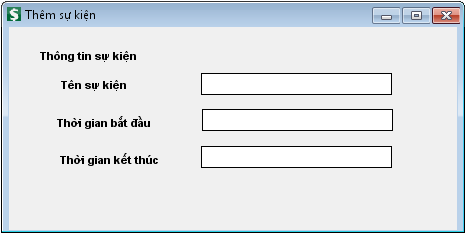
*Hình 9: Màn hình thêm sản phẩm vào nhóm hàng*



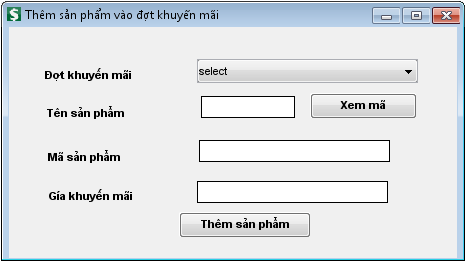
*Hình 10: Màn hình xem các mặt hàng của 1 nhóm hàng*



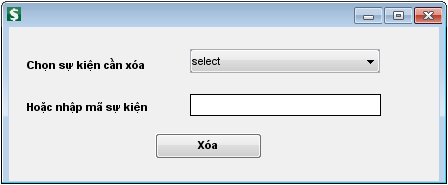
*Hình 11: Màn hình xem chi tiết 1 sự kiện*



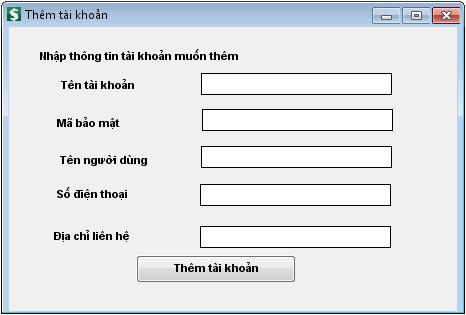
*Hình 12: Màn hình thêm 1 sự kiện*



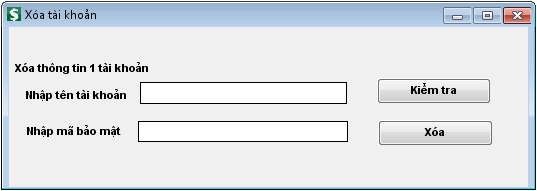
*Hình 13: Màn hình thêm sản phẩm vào đợt khuyến mãi*



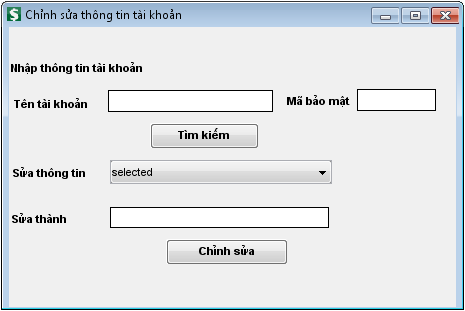
*Hình 14: Màn hình xóa 1 sự kiện*



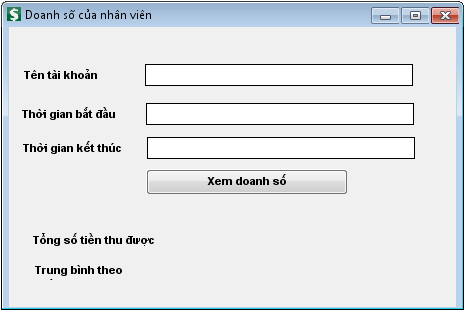
*Hình 15: Màn hình thêm 1 tài khoản*



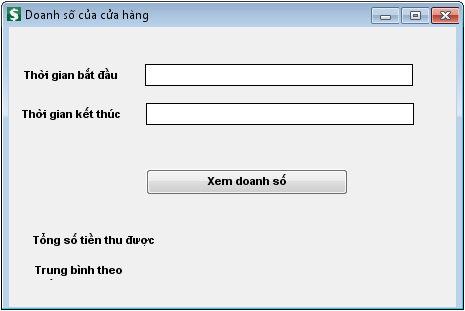
*Hình 16: Màn hình xóa 1 tài khoản*



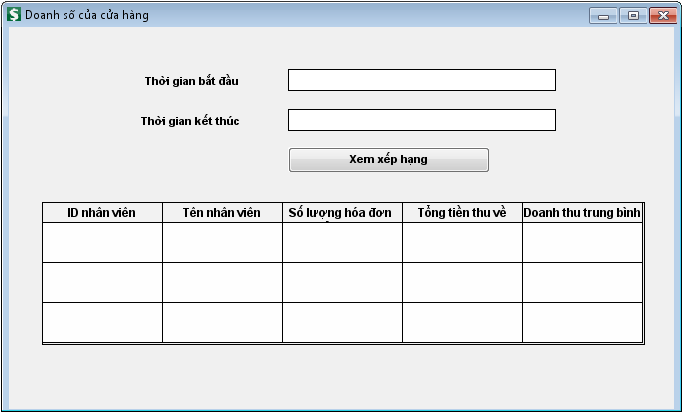
*Hình 17: Màn hình chỉnh sửa thông tin 1 tài khoản*



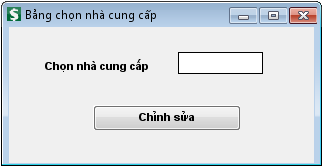
*Hình 18: Màn hình doanh số nhân viên*



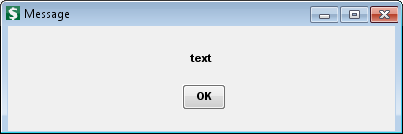
*Hình 19: Màn hình doanh số cửa hàng*



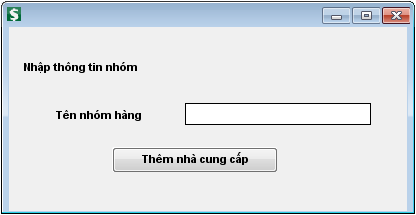
*Hình 20: Màn hình doanh số cửa hàng*



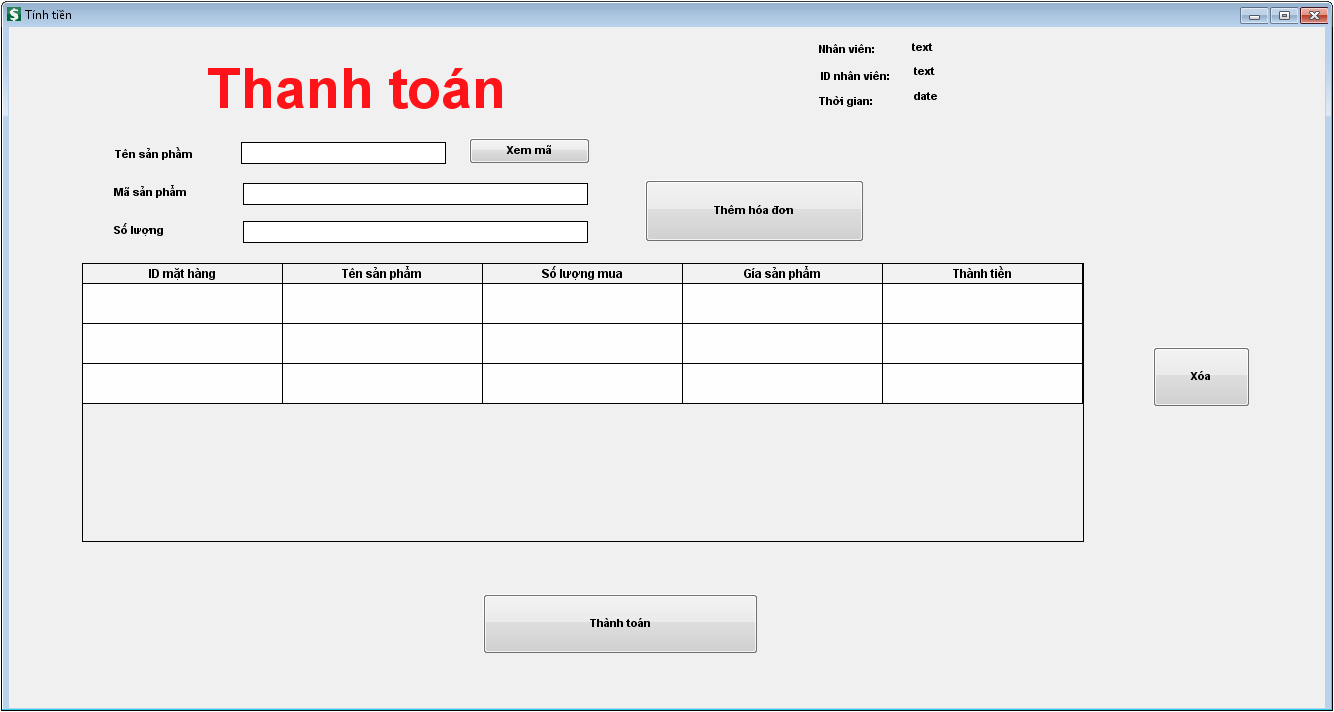
*Hình 21: Màn hình chọn nhà cung cấp*



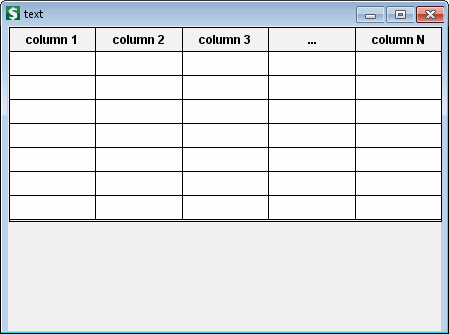
*Hình 22: Màn hình message*



*Hình 23: Màn hình nhập tên nhóm hàng*



*Hinh 24: Màn hình thanh toán*



*Hình 25: Bảng biểu mẫu*

## 2.4. Lập trình ứng dụng.

### 2.4.1. Cơ sở các thư viện, kiến thức sử dụng.

Kiến thức sử dụng:

* Truyền gói tin qua mạng lan sử dụng Socket.
* Kết nối hệ cơ sở dữ liệu bằng thư viện JDBC.
* Thiết kế giao diện đồ họa bằng thư viện Swing.

Các thư viện Java sử dụng:

* java.awt.\*.
* javax.swing.\*.
* java.sql.\*.
* java.util.\*.
* java.text.\*.
* java.io.\*.
* java.net.\*.

### 2.4.2. Chi tiết mã nguồn chương trình.

Bảng chi tiết chức năng của các hàm và class được sử dụng trong chương trình.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Class | Hàm | Chức năng |
| 1 | DoMain |  | Class khởi tạo chương trình. |
| 2 |  | public static void importdata() | Hàm import cơ sở dữ liệu nếu chạy chương trình nếu CSDL chưa tồn tại. |
| 3 |  | public static void run() | Hàm gọi đối tượng đăng nhập người sử dụng. |
| 4 | CreateDatabase |  | Class cài đăt giao diện đăng nhập tài khoản làm việc trong My SQL. |
| 5 | Layer1Interface |  | Class cài đặt đăng nhập tài khoản nhân viên. |
| 6 | Layer3Interface |  | Class cài đặt chức năng thanh toán. |
| 7 |  | private int testIn(String str) | Trả về 1 nếu chuỗi đầu vào là 1 số.  Trả về 0 các trường hợp còn lại. |
| 8 |  | private void getInTable(ResultSet rst ,int ids, int numb, String gia) | Hàm cập nhật dữ liệu vào bảng hàng thanh toán. |
| 9 |  | private static int convertStringToIn(String str) | Hàm trả về giá trị số nguyên được chuyển từ chuỗi đầu vào. |
| 10 | Invoice |  | Class lấy thông tin từ sản phẩm. |
| 11 |  | public int getInvoice(int id) | Hàm trả về giá trị giá tiền của 1 mặt hàng. |
| 12 | TimeSystem |  | Class lấy thông tin thời gian thực của hệ thống. |
| 13 |  | public String fullDate() | Hàm trả về thời gian thực của hệ thống định dạng: yyyy-MM-dd HH:mm:ss |
| 14 | ChooseEven |  | Class cài đặt giao diện cho người dùng chọn đợt khuyến mại, nếu có một măt hàng thuộc nhiều đợt khuyến mại cùng một thời điểm. |
| 15 | Layer2Interface |  | Class cài đặt giao diện quản lý cửa hàng. |
| 16 | ProductManage |  | Class quản lý các chức năng liên quan đến mặt hàng. |
| 17 |  | public void searchProduct() | Hàm sử lý đưa ra những mặt hàng đang kinh doang, cho phép sửa, xóa. |
| 18 |  | public void addNewProduct() | Hàm sử lý thêm mới 1 mặt hàng. |
| 19 |  | public void showProductNeedAdd () | Hàm sử lý đưa ra những mặt hàng đang hết hoặc gần hết cần bổ sung. |
| 20 | SupplyManage |  | Class quản lý các chức năng liên quan đến nhà cung cấp. |
| 21 |  | public void fixInfomationSupply() | Hàm sử lý đưa ra thông tin nhà cung cấp, cho phép sửa, xóa, thêm mặt hàng cung cấp. |
| 22 |  | public void addNewSupply() | Hàm sử lý thêm một nhà cung cấp mới vào dữ liệu. |
| 23 | ProductGroupManage |  | Class quản lý các chức năng liên quan đến nhóm sản phẩm. |
| 24 |  | public void showProdutInGroup() | Hàm sử lý đưa ra những sản phẩm thuộc một nhóm hàng đang có. |
| 25 |  | public void addProductToGroup() | Hàm sử lý thêm một mặt hàng mới vào một nhóm hàng. |
| 26 |  | public void addNewGroup() | Hàm sử lý tạo thêm một nhóm hàng mới. |
| 27 |  | public void ShowAllGroup() | Hàm sử lý đưa ra tất cả những nhóm hàng đang có. |
| 28 | EvenManage |  | Class quản lý các chức năng liên quan đến sự kiện khuyến mại. |
| 29 |  | public void deleteEven() | Hàm sử lý xóa một sự kiện giảm giá. |
| 30 |  | public void addProduceInEven() | Hàm sử lý thêm một mặt hàng vào một sự kiện khuyến mại. |
| 31 |  | public void showOneEven() | Hàm sử lý đưa ra tất cả các mặt hàng được giảm giá trong một đợt sự kiện. |
| 32 |  | public void ShowAllEven() | Hàm sử lý đưa ra tất cả các sự kiện đang diễn ra. |
| 33 |  | public void createEven () | Hàm sử lý thêm mới một sự kiện giảm giá. |
| 34 | AccManage |  | Class quản lý các chức năng liên quan đến tài khoản nhân viên. |
| 35 |  | public void fixAccount() | Hàm sử lý chỉnh sửa thông tin một tài khoản. |
| 36 |  | public void deleteAccount() | Hàm sử lý xóa một tài khoản đang sử dụng. |
| 37 |  | public void addNewAccount() | Hàm sử lý thêm một tài khoản nhân viên mới. |
| 38 | RevenueManage |  | Class quản lý các chức năng liên quan đến doanh số bán hàng. |
| 39 |  | public void showRevenueOf1Account() | Hàm sử lý đưa ra doanh thu của một nhân viên trong một khoảng thời gian. |
| 40 |  | public void showRevenueOfStore() | Hàm sử lý đưa ra doanh thu của cửa hàng trong một khoảng thời gian. |
| 41 |  | public void showBestAccount() | Hàm sử lý đưa ra bảng xếp hạng doanh thu của các nhân viên. |

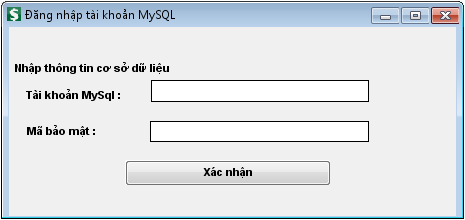
Phần 3. Chức năng, hướng dẫn sử dụng ứng dụng.

## 3.1. Yêu câu môi trường trước khi sử dụng ứng dụng.

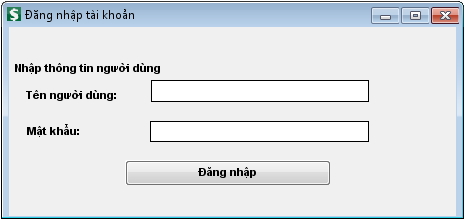
* Hệ điều hành tương thích: Windows.
* Máy tỉnh đã cài đặt mysql.
* môi trường chạy ứng dụng java.

## 3.2. Chức năng, hướng dẫn sử dụng

* Ứng dụng xây đựng cho người quản lý và nhân viên thu ngân với 2 màn hình quản lý khác nhau.

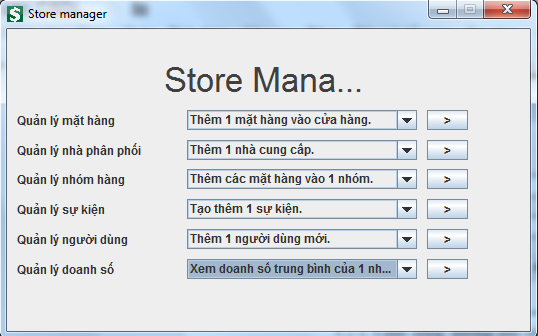


* Đầu tiên bạn cần đăng nhập vào mysql. Điền tên đăng nhập và mặt khẩu vào các ô tướng ứng. Sau khí chọn xác nhân, chương trình sẽ đưa bạn đến màn hình đăng nhập tài khoản. Ở đây bạn sẽ có 2 lựa chọn: 1 là đăng nhập vào tài khoản của người quản lý 2 là đăng nhập vào tài khoản của thu ngân. Sau khi đăng nhập với tài khoàn quản lý hoặc thu ngân mà bạn có, chương trình sẽ đưa bạn đến 2 màn hình quản lý khác nhau.



### 3.2.1. Chức năng, hướng dẫn sử dụng với người quản lý.

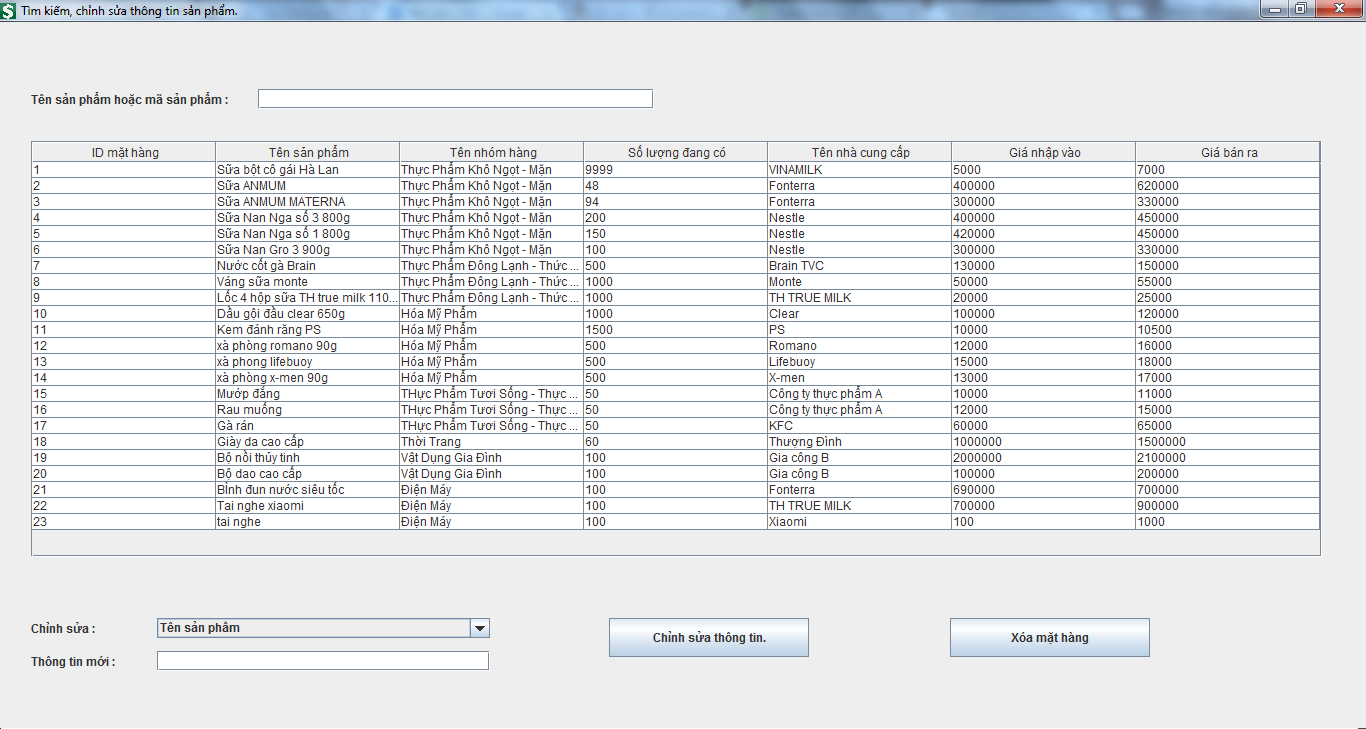
Chương trình dành cho người quản lý gồm 6 chức năng chình được chia thành các phần tách biệt nhau, để chọn chức năng con của mỗi phần, bạn chọn mỗi tên xổ xuống trong trong ô bên cạnh của các chức năng.Như hình dưới:



#### 3.2.1.1. Quản lý mặt hàng

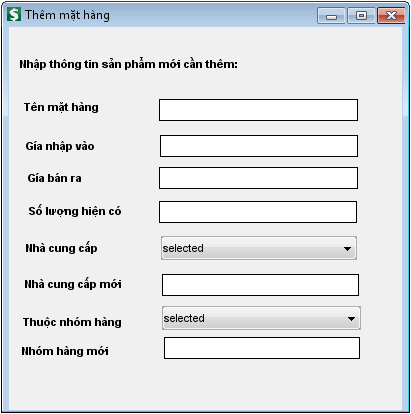
Chức năng này gồm có: các mặt hàng đang kinh doanh, thêm 1 mặt hàng vào cửa hàng, các mặt hàng sắp hết hàng.

* Chức năng các mặt hàng đang kinh doanh: Trong đó tổng hợp cả chức năng tìm kiếm, chỉnh sửa, xóa mặt hàng như hình dưới:



ở đây bạn có thể tìm kiếm sản phầm bằng cách nhập tên hoặc id sản phẩm vào ô tên sản phẩm hoặc mã sản phẩm . Nếu bạn muốn chỉnh sửa, xóa sản phẩm thì click vào sản phẩm tương ứng ở table bên dưới và chọn chức năng muốn chương trình thực hiện ở phần chỉnh sửa rồi điền thông tin và click vào chức năng tương úng là chỉnh sửa thông tin hoặc xóa mặt hàng.

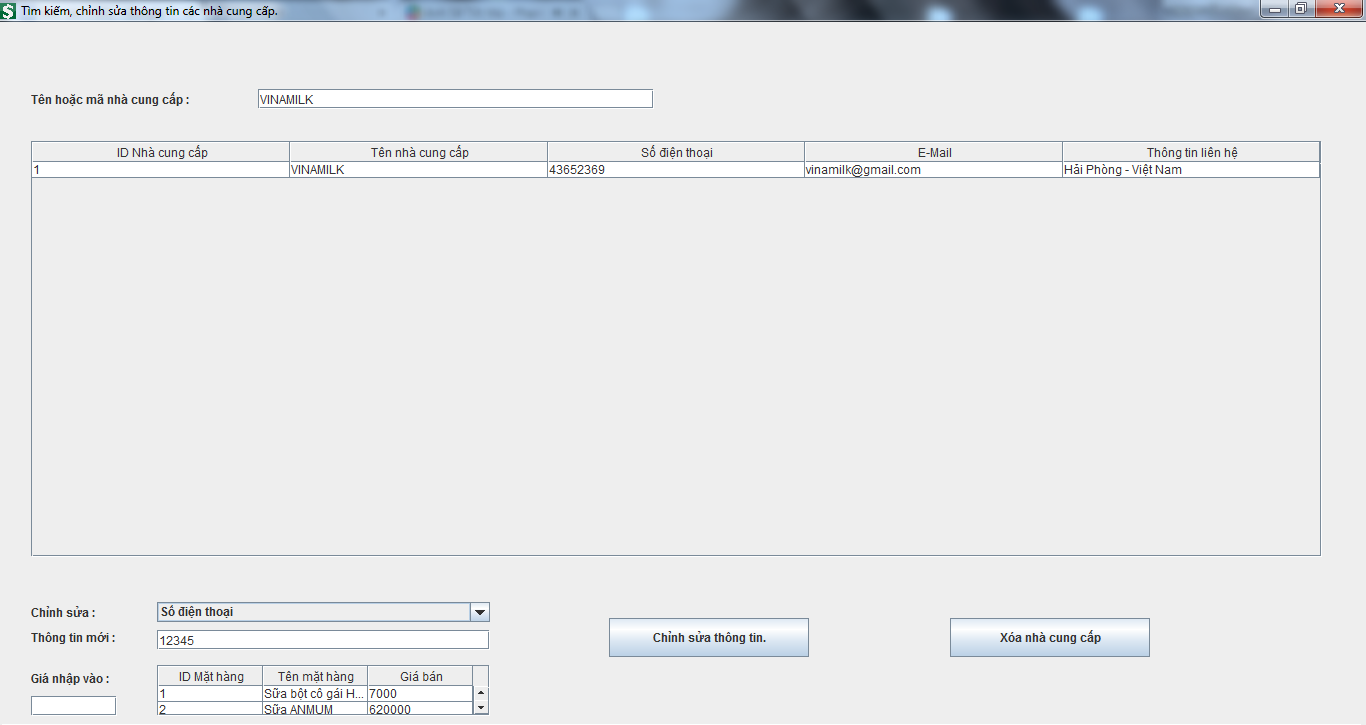
* Chức năng thêm mặt hàng. Khi bạn chọn chức năng này, chương trình sẽ đưa bạn đến phần nhập thông tin cho mặt hàng, bạn cần điền đầy đủ thông tin cho mặt hàng cần thêm như tên sản phẩm, giá nhập, giá bán … Như hình bên dưới:



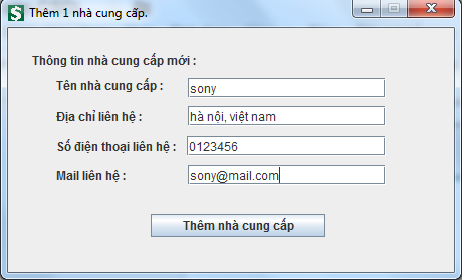
#### 3.2.1.2. Quản lý nhà phân phối

Chức năng này cung cấp cho bạn các khả năng: xem các nhà cung cấp, thêm 1 nhà cung cấp mới, xóa nhà cung cấp và sửa thông tin nhà cung cấp.

* Tìm kiếm. chỉnh sưa thông tin nhà cung cấp: các chức năng này được gộp chúng vào 1 cửa sổ như hình dưới. Ở đây bạn có thể xem các nhà cung cấp ở table, tìm kiếm nhà cung cấp bằng cách nhập tên hoặc ID vào phần: Tên hoặc mã nhà cung cấp, ngoại ra bạn cũng có thể chỉnh sửa thông tin nhà cung cấp ở phần dưới của cửa sổ bằng cách chọn các chức năng tương ứng và nhập thông tin cần thiết. Nếu bạn muốn xóa nhà cung cấp cần chọn nhà cung cấp cần xóa ở trong table và click vào nút: xóa nhà cung cấp.



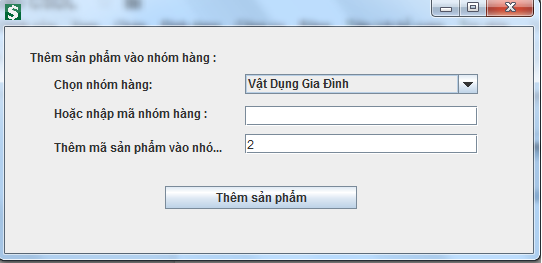
* Thêm 1 nhà cung cấp cung cấp: Khi chọn chức năng này, chương trình sẽ đưa bạn đến cửa số nhập thông tin nhà cung cấp muốn thêm như hình dưới, ở đây bạn chỉ cần điền thông tin của nhà cung cấp muốn thêm rồi click vào nút thêm nhà cung cấp.



#### 3.2.1.3. Quản lý nhóm hàng.

Chức năng này cung cấp cho bạn các khả nằng như xem các nhóm hàng, Thêm 1 nhóm hàng, thêm các mặt hàng vào 1 nhóm hàng …

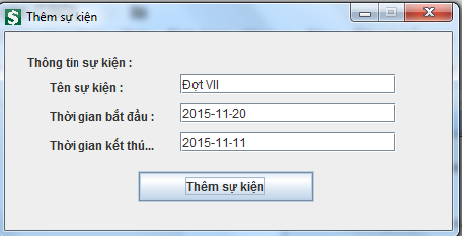
Với các chức năng: Thêm 1 nhóm hàng, thêm các mặt hàng vào 1 nhóm hàng hay xem các mặt hàng trong 1 nhóm: Bạn chỉ cần chọn chức năng mong muốn và điền thông tin đầy đủ cho các sản phẩm hoặc nhóm rồi click chọn chức năng mong muốn. Ví dụ như hình bên dưới demo chức năng thêm các mặt hàng vào nhóm:



#### 3.2.1.4. Quản lý sự kiện.

Cung cấp các chức năng : Xem tất cả các sự kiện, xem 1 sự kiện, tạo 1 sự kiện, thêm sản phẩm vào sự kiện, xóa 1 sự kiện: tương tự như phần quản lý nhóm hàng, bạn cũng chọn các chức năng tương ứng và điền đầy đủ thông tin.

Chú ý trong chức nằng tạo 1 sự kiện có phần điền thời gian bắt đầu sự kiện và kết thúc sự kiện, bạn phải điền ngày tháng cho phù hợp, ví dụ như là: ngày bắt đầu trước ngày kết thúc hoặc 1 thánh có không quá 32 ngày … để tránh mất thời gian phải nhập lại.

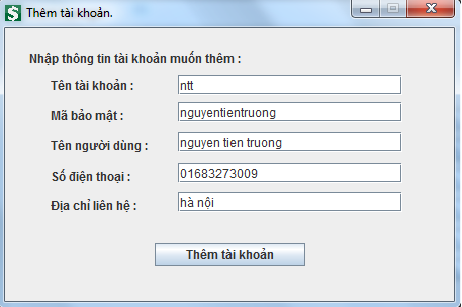


#### 3.2.1.5. Quản lý người dùng

Chức năng này cho phép bạn thêm, sửa hay xóa các tài khoản của thu ngân.

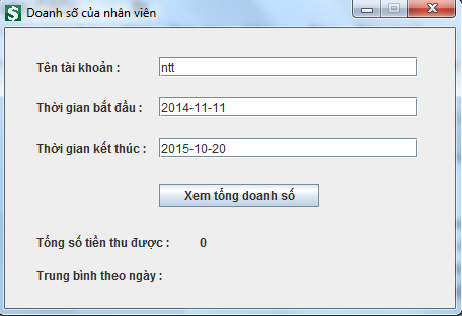
Chú ý: Bạn không thể thao tác các chức năng này đối với tài khoản của người quản lý.

Bạn cũng làm tương tự với các thao tác chọn chức năng tương ứng, điền đầy đủ thông tin và click vào chức năng mong muốn . Như hình demo thêm 1 tài khoản:



#### 3.2.1.6. Quản lý doanh số.

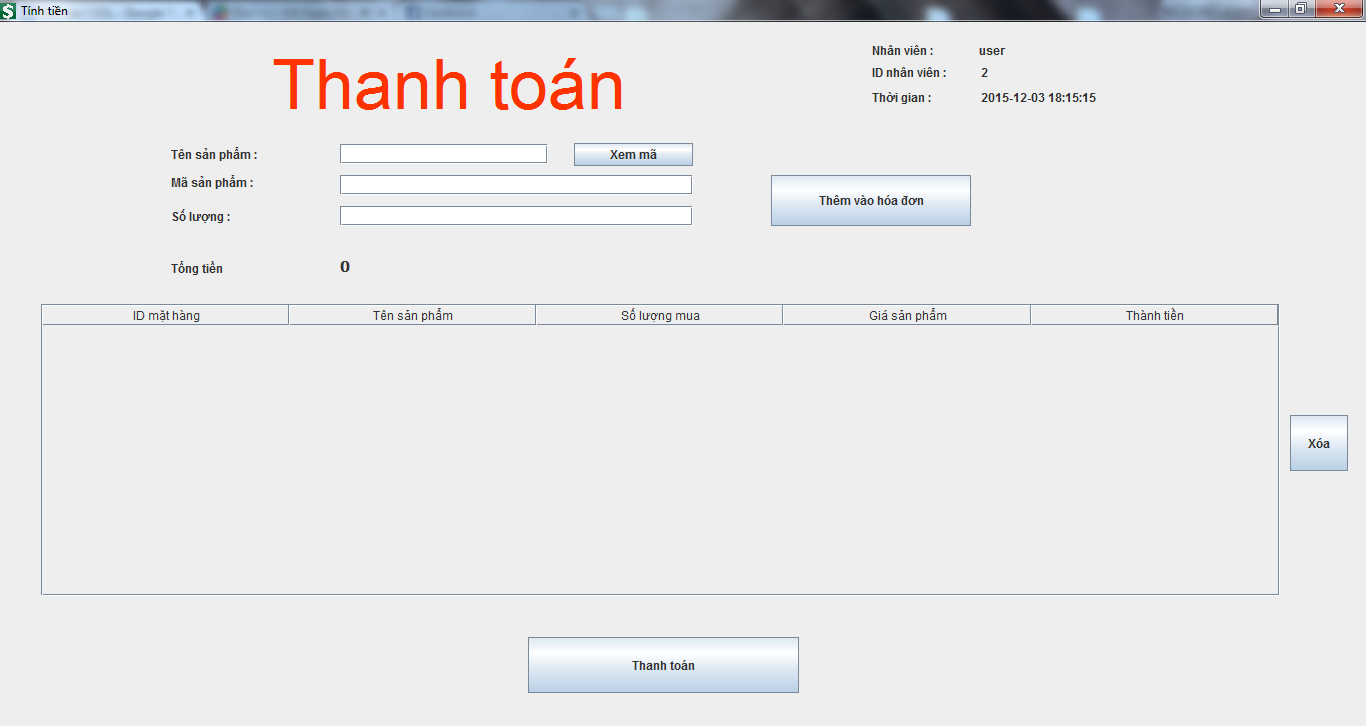
Bạn có thể xem doanh số trung bình của nhân viên để đánh giá khả năng của từng người bằng cách chọn chức năng: xem doanh số trung bình của 1 nhân viên ở màn hình chính phần quản lý doanh số, sau đó điền tên tài khoản của nhân viên tương ứng cùng với thời gian vắt đầu và kết thúc đợt khảo sát và click vào nút xem doanh số.



Ngoài ra bạn có thể xem doanh thu của cửa hàng trong 1 khoảng thời gian hay bảng xếp hạng doanh số của nhân viên để phân loại năng lực cũng như có thưởng, nhắc nhở, cải thiện khả năng bán hàng cho nhân viên. Để sử dụng các chức năng này bạn cũng chọn các phần tương ứng và điền thông tin cũng như ngày tháng vào các ô phù hợp và xem kết quả trong 1 cửa sổ mới.

### 3.2.2. Chức năng, hướng dẫn sử dụng với thu ngân.

Chức năng này chỉ cung cấp cho nhân viên thu ngân 1 chức năng duy nhất là tính hóa đơn cho khách hàng với 1 màn hình chính duy nhất và không có các màn hình phụ kèm theo với đầy đủ các chức năng để tính toán hóa đơn như: Thêm 1 sản phẩm vào hóa đơn, xóa 1 sản phẩm khỏi hóa đơn và thanh toán. Như hình:



Để thêm 1 sản phẩm vào hóa đơn bạn cần nhập tên sản phẩm hoặc mã sản phẩm vào các ô tương ứng và có thể click vào nút xem mã để kiểm tra sự tồn tại của sản phẩm rồi điền số lượng sản phẩm được mua và click vào nút thêm vào hóa đơn. Để xóa 1 sản phẩm khỏi hóa đơn bạn làm như sau: Chọn sản phẩm cần xóa và click vào nút xóa. Cuối cùng bạn click vào nút thanh toán để xem tổng tiền khách hàng cần trả và thực hiên phiên thanh toán khác.

Phần 4. Hướng phát triển phần mềm

Hướng phát triển mở rộng của cơ sở dữ liệu.

* + Tùy thuộc theo mục đích của mỗi của hàng. Cho phép quản lý thêm các trường dữ liệu cho phù hợp.
  + Khi lượng dữ liệu trở nên lớn hơn thì ta có thể thuê sử dụng các máy chủ ảo, điện toán đám mây để làm tăng không gian lưu trữ cũng như hiệu năng phần cứng.

Hướng phát triển chức năng của phần mềm.

* + Phần mềm sẽ cần phải mở rộng them các chức năng để phù hợp với nhiều loại cửa hàng hơn.
  + Phát triển chức năng online để một cửa hàng có nhiều chi nhánh thì vẫn đồng nhất được một cơ sở dữ liệu.
  + Phát triển thêm chức năng để người mua cũng có thể xem sản phẩm.
  + Phát triển chức năng thanh toán trên các nền tảng di động để các nhân viên có thể dễ dàng tăng ca hoặc thu ngân tại bàn.

Phần 5. Kết luận

Việc nghiên cứu, xây dựng, phát hàng ứng dụng quản lý cửa hàng đã giải quyết được các vấn đề sau:

* Cung cấp một phần mềm quản lý cửa hàng hoàn toàn mới.
* Tăng tính bảo mật cho các cơ sở dữ liệu cửa hàng vừa và nhỏ.
* Phát triển một phần mềm nhẹ, có tính tương thích cao, giao diện và cách sử dụng dễ dàng, thân thiện với mọi đối tượng.
* Cơ sở vật chất, đầu tư ban đầu thấp.
* Chức năng khá đầy đủ và thỏa mãn các nhu cầu cơ bản
* Góp phần nâng cao sự chuyên nghiệp trong bán hàng, cách kinh doanh và thanh toán tại Việt Nam.