CSC 3326

Database Systems

Final Report

Team Members:

Ali Elouafiq

Ayoub Khobalatte

Wassim Benhallam

Mohammed Wael Khobalatte

Mohammed Redouane Khrifi

Supervised By:

Dr. Nasser Assem

Spring 2010

Thursday, March 18th 2010

Table of Contents

Project Proposal

Introduction

The requirements of the deliverables dictate that we should have an understanding of our product based on a modeling that embraces all aspects of real products. This way we ensure our working system is properly functioning. Attempts to understand similar systems were successful as we made extensive use of the internet to research the subject. Such measures enabled us to make some enhanced modifications to our project proposal so as to respond to incoming issues raised by these research results. Among the issues raised were business rules and redundancy occurrences, factors that were heavily discussed in class as well. The team has made some adaptations to the project to incorporate these notions. This report describes every step that our team followed in order to set a theoretical framework for our project that will ensure a smooth explanation for different aspects of our project and careful transition to the implementation part.

Project Overview

Prior to any subsequent steps, our team thought about several ideas for database applications to adopt in our project. Ideas were pretty much inspired from real world models and varied from the classical school, pharmacy, or library applications to more challenging ones such as business-related applications. After analyzing the advantages and drawbacks of each one in terms of difficulty, appropriateness, and usefulness, we decided to work on a B2C application (Business to Consumer) for e-commerce named “PIRKT”.

PIRKT, which means “buy” in Letton, is an online selling platform vending computer components, electronic products, and domestic appliances. In fact, we do not have a customer for the application; Pirkt is only the fruit of our imagination and assumes the existence of physical point of sales distributed all around Morocco that will materialize the online operations and perform delivery services.

Project Objectives

The main objective of this project is to create a software with an integrated database. The software can be beneficial to Al Akhawayn community and to outsiders in two different ways:

1. It will facilitate the process of checking the products' availability without the need of actually going to a certain store.
2. The billing and payment can be done in a quicker manner. There will be no need to make tangible transaction.

In addition, this project will be a great opportunity for us, developers, to practice the set of skills acquired in database class and familiarize us with how real-world applications work is conducted.

Requirements' gathering

Online shopping is an important activity that started propagating in the Moroccan market and growing at a frightening rate. Nevertheless, the number of web shops is still small in the country. The time this business starts ravaging comes near, so, as a team, we predict that engineers in morocco will be demanded to design online/web shops more than ever.
As a preparation for what the future holds, we agreed upon working on PIRKT, the first web shop designed by AUI students.

Since Ifrane's shops don't offer a satisfying variety of products, and often, to go shopping AUI students and faculty feel obliged to travel to Fez or Meknes to do their shopping and in order to facilitate the shopping experience and buying of goods in AUI. AUI students and faculty won't have to travel to other cities anymore since products will be delivered by post to the university; all what will be need is a PayPal or a bank account and a PIRKT account.

Project Time Summary:

The database project is expected to begin on February 2010 and conclude with implementation and close-out by May 2010.

Requirements' Specification

The Software will have many functional capabilities, ones that are designed for normal users: students, faculty and staff. These functionalities are mainly about the searching and purchasing of products. These functionalities will be accessible via the web based application.

The Desktop application will allow the administrator to perform his/her tasks separately from the web based one. The admin functionalities are mainly about the control over accounts, products (stock). The desktop application will only be used by the admin. In Addition, more other functionalities will be offered by PIRKT. Hence, the software will include a database that is going to be accessed by administrators and users.

The users will not have the same privileges and the same access levels. The admin will have full access to all the functionalities, while normal users have access to only some specific data and functionalities.

The main modules handled by the software are:

User Management

This Module will deal with user accounts. As seen in many international selling websites, there are three types of users: administrators, registered users and visitors. In order to categorize users of our web application based on their activities, we should create accounts that will be assigned different roles.

A visitor can create his/her account, modify it or remove it. If added, the visitor becomes a registered user and will be granted the corresponding access level and privileges. On the other hand, an administrator has control over all the functionalities including blocking a registered account.

Product and category Management

This module deals with the different types of products available in the database and the different functionalities applied on them by different users. Products are categorized according to different criteria. An administrator can add, modify, remove a product as well as add, modify, remove a category, and assign a product to a category. A visitor can only search for a product and view it whereas a registered user can, in addition, add it to his/her cart and purchase it.

Stock and statistics Management

This module concerns mainly administrators who access the database of products and keep track of stock statistics (how many items sold and bought last month, in which month revenue was highest…) which will help them make appropriate business decisions.

Invoice, Shipping and *Payment Management*

This module involves the process of issuing bills to registered users whenever a purchase occurs. It works in parallel with the stock and statistics module. Payments can be done in various forms (e.g. Paypal, Mastercard…) and trigger the delivery services.

Non functional requirements

- Design an easy and friendly graphical User Interface in French (and in English if possible).
- Apply the adequate methodologies proposed by our instructor.
- Meet the deadlines.

Project Management Plan

Dates and Tasks:

As agreed upon among the team members and according to the course's syllabus, our team will make sure to follow the time schedule specified below, starting from this week.

Date	Tasks	Member (s) Responsible
02/07/2010	Choosing the project	All
02/12/2010	Proposal: Requirements gathering	Wassim Benhallam
02/12/2010	Proposal: Requirement Specification	Ayoub Khobalatte

		Mohammed Wael Khobalatte Mohammed Redouane Khrifi
02/12/2010	Proposal: Project Management Plan	Ali Elouafiq
02/15/2010	Proposal: Peer-review and final changes	All
03/18/2010	Project Midterm Report: ER diagram	Wassim Benhallam
03/18/2010	Project Midterm Report: Tables' Definitions and Normalization.	Ayoub Khobalatte
03/18/2010	Project Midterm Report: Logical Schema	Redouane Khrifi
03/18/2010	Project Midterm Report: Creation and population of the database.	Ali Elouafiq
03/18/2010	Project Midterm Report: Queries	Mohammed Wael Khobalatte
03/18/2010	Project Midterm Report: Peer-review and final changes	All
05/04/2010	Application Development and Report: GUI design and implementation	All
05/04/2010	Application Development and Report: Implementation of all functionalities	All
05/04/2010	Application Development and Report: Testing	All
05/04/2010	Application Development and Report: Final Report	All
Last Week	Final Presentation and Demo	All

The Tasks Assignment:

GENERAL PROCEDURE

Concerning the first weeks the members of the team will decide upon a meeting time, and will gather to work on the same task, since the first tasks are about design decisions and requirement engineering, then the design decisions and the requirements will be clear for all the members of the team.

After the proposal due date, the team member will decide who is most proficient to be a team leader, the one who will distribute the tasks among the team, coordinate between them and make sure everything is going as smoothly as possible.

Our Strategy

In order to be efficient and effective in our work, the team members have decided to assign a team leader. The other four members will be divided to two groups of two people; each group (the team leader included) will work on a specific task to save time and meet the deadlines. Upon completion, the whole team should hold a meeting to review what have been done, change what is it that needs to be changed so as to produce a final report or product that is as accurate and effective as possible.

Below is a detailed explanation of the different strategies that will be followed for each specific task.

1. Requirement Analysis

   In this part, the team's members are supposed to fully understand what is needed to be done before starting the design and implementation phase, that is, gather as much information as possible regarding the requirements.

2. Database Design

   This is the most important part in this course, since the project is conducted for the database class. The team will make sure to provide a very organized and complete database, which covers all data of our software. In order to do this, our team will need to provide an accurate Entity Relation diagram.

3. Implementation

   The purpose of this part is to transform the design done earlier into a high level language (C#), integrating the database by using a suitable DBMS and therefore, create a proper software. The web based side of the project will be programmed .Net framework (with SQL queries embedded) and XHTML.

4. Testing

After implementation is complete, a series of tests will conducted to insure that the software works properly.

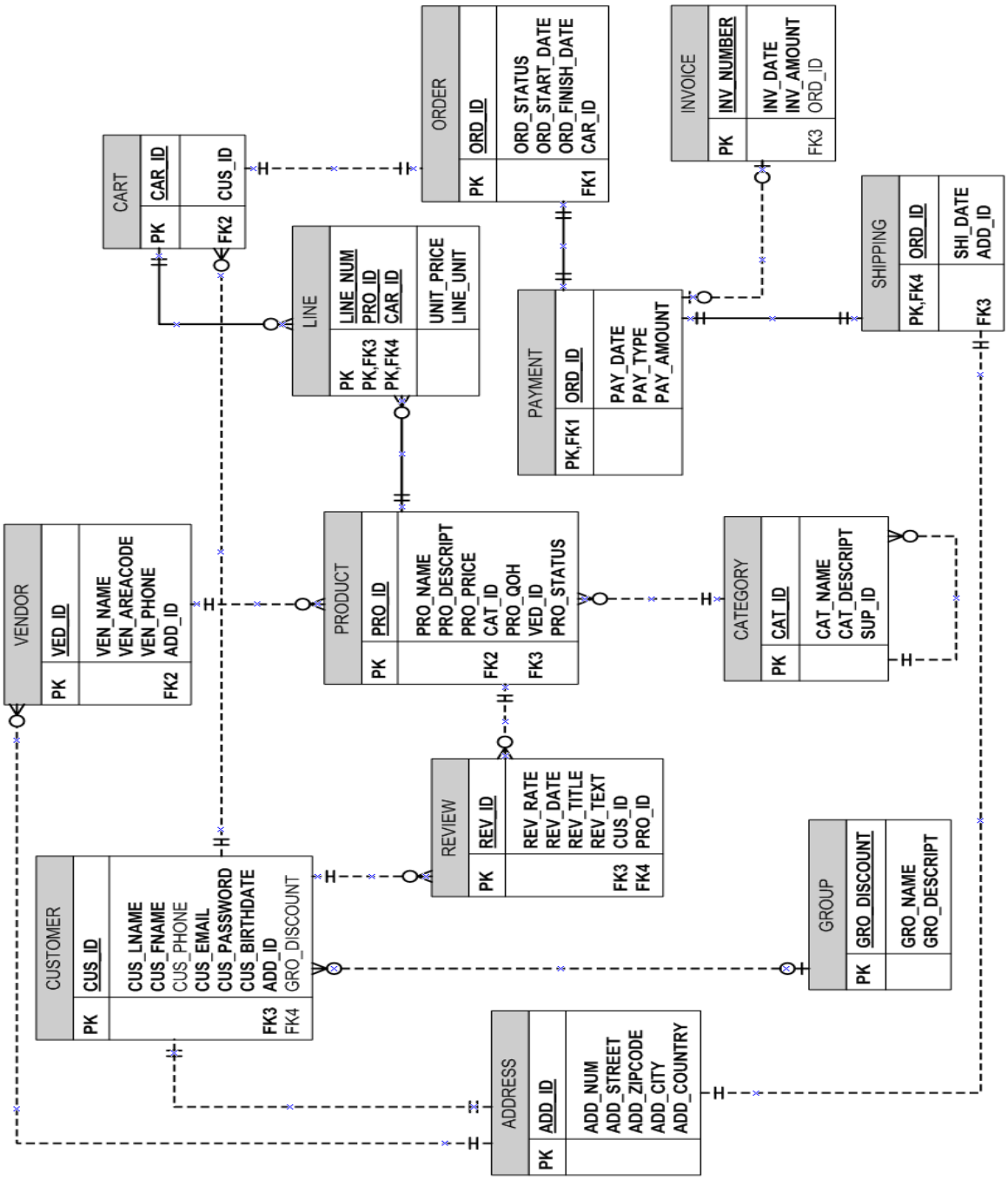
5. MAINTENANCE

Should any error occur and the person using the system report it, the team's members will make sure to devote time to find a solution and update the current version of the software.

Project Design and Database Creation

Conceptual Design

The Entity Relationship Diagram



Description of the Entity Relationship Diagram (ERD)

In software engineering, an entity-relationship model (ERM) is an abstract and conceptual representation of data. It represents the entities to be used in a certain database and the relationships among them including their cardinality and type.

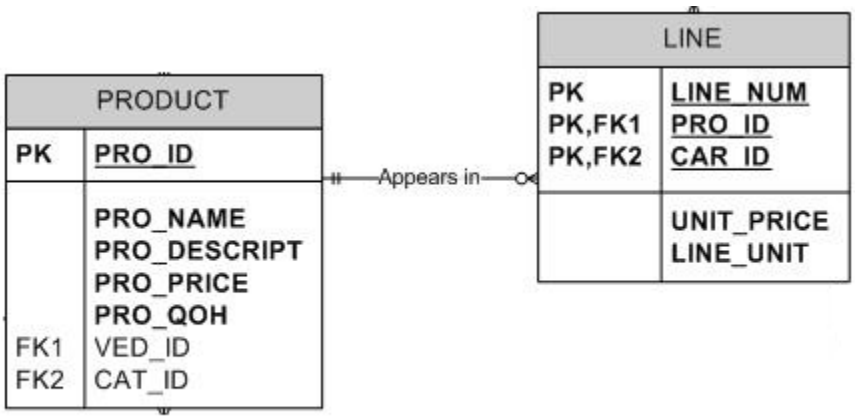
Defining the entity Relationship diagram is one of the most essential in designing a database and even in software. Usually, databases' accuracy and degree of professionalism is assessed through the entity Relationship diagram. For this reason, our team members did their best to optimize our project's ERD as much as possible. In fact, we have spent four hours discussing the entities that should be used and the relationships that exist between them. The final outcome is show hereby above.

For our project, Pirkt, we will be using thirteen different entities: PRODUCT, CUSTOMER, ADDRESS, VENDOR, CART, LINE, INVOICE, REVIEW, GROUP, CATEGORY, PAYMENT, ORDER and finally SHIPPING. Each of these entities is useful to our project in a specific way. For instance, the CUSTOMER entity is essential obviously, since each electronic selling system depend on customer to achieve success. The VENDOR, on the other hand, helps the system keeps track of which companies supplies the products.
More information about the entities separately is below

Business rules:

Business rules are a set of constraints and regulations that are intended to assert business structure or to control or influence the behavior of the business. The process of identifying the business rules starts by the definition of the different entities and their respective attributes; this part was already done and explained in the previous paragraphs.
The business rules were used along the process of creating the tables and their relationships. In fact, the business rules defined the type of relationships to adopt between the tables; one-to-one, one-to-many or many-to-many relationship.

Product Line Relationship



a) Description of The Relationship

In real life, when a customer receives an invoice, it contains many lines and each one of these lines includes a product id, the quantity shipped and so. For this reason, our team members decided that the system should include a relationship between line and product. (Please note that line can be considered as a bridge entity between the product and cart. For more emphases, Please read the relation between cart and line below)

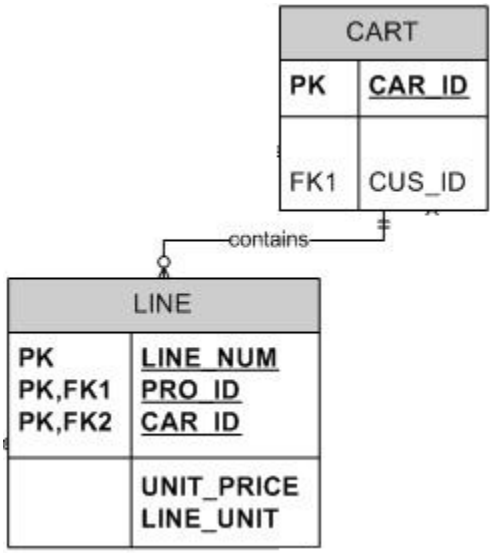
b) Type of The Relationship

The relationship between the line and product is a strong relationship; one of the two entities' primary keys exists in the other entity and the line entity cannot exit independently of the product entity (i.e. a product may not appear in any line and a line should have a product's id)

c) Cardinality of The Relationship

This relationship is of type one-to-many. The reason we chose this type is because in real life business, a production may appear in many lines (provided that the lines should be in different invoices) and each line may contain only one product.

Line Cart Relationship



a) *Description of The Relationship*

When a customer finished picking items from the application and putting them in the cart and decides to purchase them, ll the items in the cart should be transformed into lines which will be contained in the invoice issued afterwards.

b) *Type of The Relationship*

The relationship between the line and cart is a strong relationship; the primary key of the cart exists in cart and an instance of line entity cannot exit independently of the cart

c) *Cardinality of The Relationship*

This a one-to-many relationship because a line contains only one cart whereas a cart may appear in many lines.

Note:

The two relationships discussed above show implicitly that there is a many to many relationship between product and cart. This relationship is expressed through the bridge entity line.

Customer Cart Relationship

a) Description of The Relationship

When a customer decides to start purchasing items or products, he or she is asked first to put the products in a cart. For this reason, there exists a relationship between cart and customer.

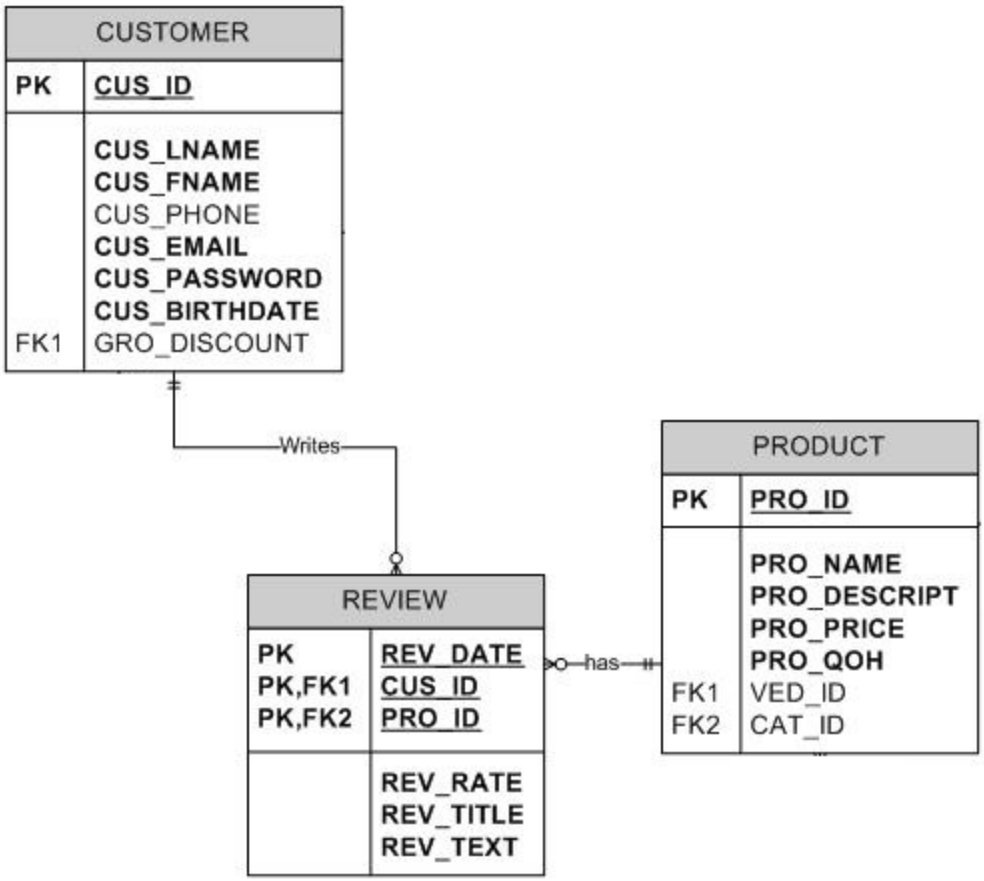
b) Type of The Relationship

This is a weak relationship even though the primary key of customer exists as a foreign key in cart, because a customer can access the application and still have no cart (as long as he or she does not start purchasing products). The two entities are therefore strong since each one of them can exist independently of the other.

c) Cardinality of The Relationship

This relationship is one-to-many because each is allowed to have 1 or more cart, while each cart can only have one (exactly one) customer responsible for it.

Customer Review Relationship and Product Review Relationship



a) *Description of the Relationship:*

The relationship between Customer and Product is about the customer reviewing products. In fact, a customer can review many products, and a product can be reviewed by many customers. This, being a many-to-many relationship, was broken down into two one-to-many relationships through the creation of the new bridge entity Review. Therefore, a customer can write many reviews each of which concerning a specific product. This makes the primary key of the review entity composed of the customer id, the product id, and the review date. The review date was introduced as part of the primary key in order to allow customers to review many times the same product without affecting the uniqueness of the review entity and rules of entity integrity.

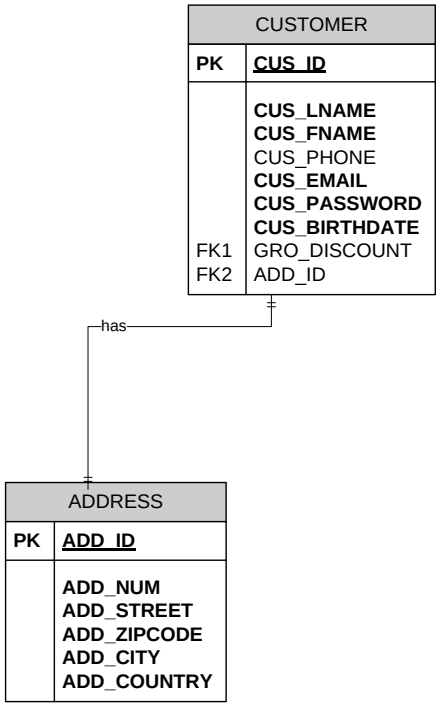
b) *Type of the relationship:*

The relationship between customer and review is a strong relationship because Review is existence dependent on the customer entity. This is further confirmed by the primary key of the customer entity being part of the primary key of the review entity. Similarly, the relationship between the product and review entity is a strong relationship because the primary key of the product entity is part of the primary key of the review entity.

c) *Cardinality of the relationship:*

As said previously, there is a one-to-many relationship between customer and review, and a one-to-many relationship between product and review.

Customer Address Relationship



a) *Description of The Relationship:*

The relation enables the system to link each customer to its address information, thereby enabling the each customer to have its own unique

address, which will enable the administrator of the system to sort customers according to their address information to make the shipping logistic easier, as well as searching efficiently the address information about his customers.

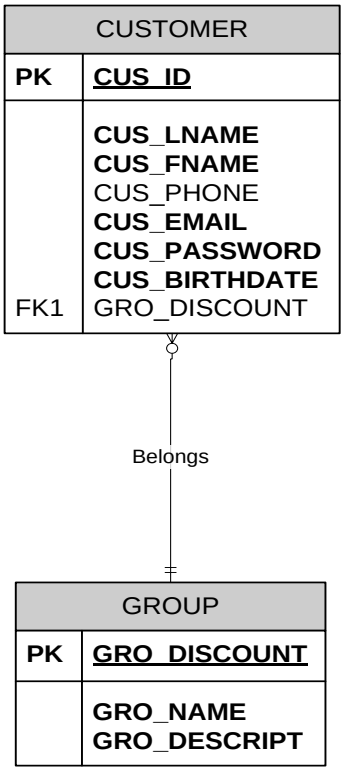
b) *Type of The Relationship*

This relationship is a "has a" relationship, each customer has an address. We can denote that this is a strong relationship, for the reason that the existence of the address itself depends deadly on the existence of the customer existence.

c) *Cardinality of The Relationship*

The Cardinality of the relationship is 1 to 1 relationship.

Customer Group Relationship



a) *Description of The Relationship:*

The relation enables the customers of the platform to belong to one discount group at a time, thereby making the discount process easy when the customer wants to make an order. Moreover this relationship enables the administrator of the system to monitor the groups and their belonging members.

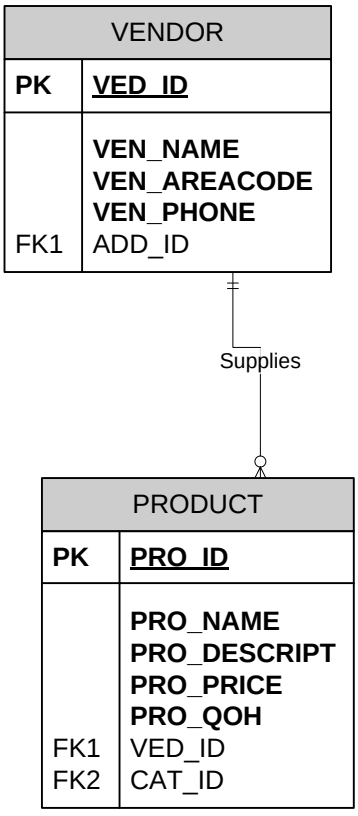
b) *Type of The Relationship*

The relationship is a “belongs to” relationship, and it is considered weak, since a customer can change from a group to another easily, as well as a group can have no member. The existence of one of the entities is completely independent of the other.

c) *Cardinality of The Relationship:*

The Relationship is a 1 to Many Relationship.

Product Vendor Relationship



a) *Description of The Relationship:*

Vendor supplies a product. The foreign key ved_id that appears in the product entity illustrates that relationship. Note that some products may not be necessarily supplied by a vendor. Therefore the foreign key can be null.

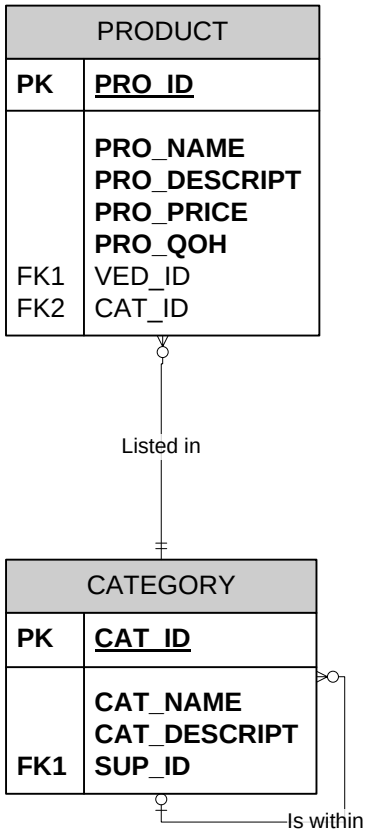
b) *Type of The Relationship:*

The type of the relationship is weak.

c) *Cardinality of The Relationship:*

One to many relationship: A vendor can supply from many products, though in some cases a vendor may not supply any product. A product is supplied by one vendor. A product might not have a vendor.

Product Category Relationship



a) *Description of The Relationship*

Products are organized in a category. The foreign key of the category that appears in a product indicates what category holds that specific product.

b) *Type of The Relationship*

This relationship is of type weak because category can exist without the necessity of hosting products and a product may not be placed in a category.

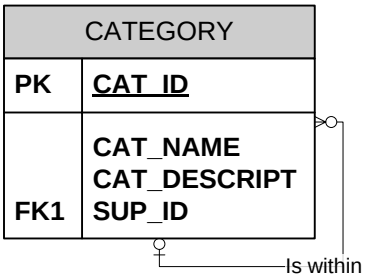
c) *Cardinality of The Relationship*

One to many relationship

A product may appear in one category.

A category can host zero or more products.

Category Category Relationship



a) *Description of The Relationship*

Since we can have embedded categories, the entity category needs to have a key that references itself.

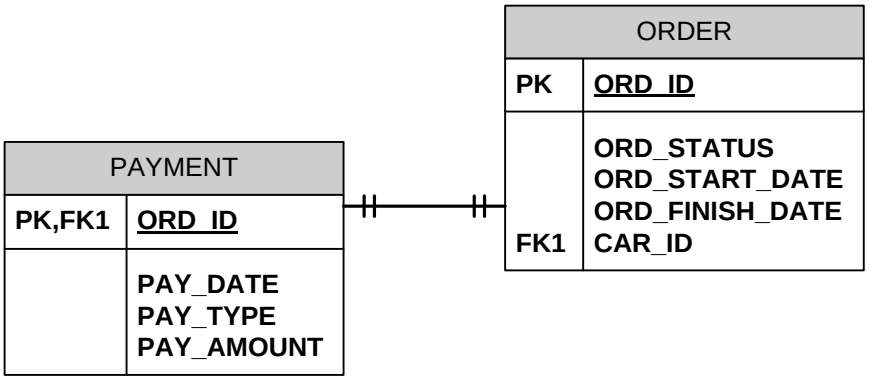
b) *Type of The Relationship*

The type of this category is weak. A category may exist without the need for a parent category.

c) *Cardinality of The Relationship*

One to many unary relationship. Parent category can host many child categories. It can have zero child categories as well. Child categories have zero or one parent category.

PAYMENT Order Relationship



a) Description of The Relationship

When the customer is done with ordering, he should pay

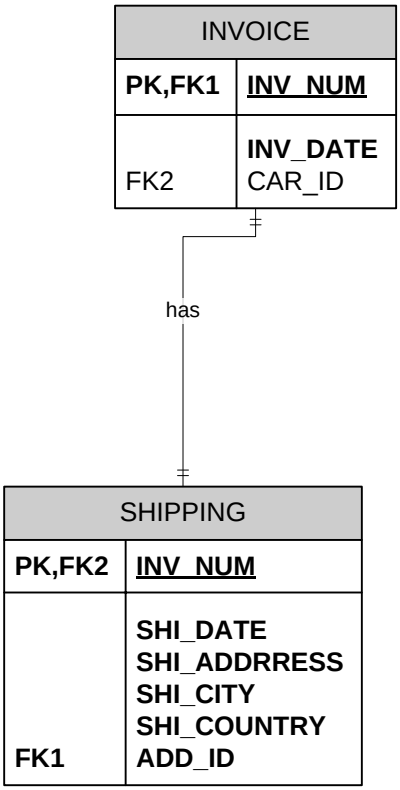
b) Type of The Relationship

The type of this category is strong. A payment cannot exist without the need for a order.

c) Cardinality of The Relationship

One to many relationship. An order issues only one payment.

Invoice Shipping relationship



a) *Description of The Relationship:*

The shipping-invoice relation enables the system to link the customer's shipping information to the generated invoice. Thanks to this relation, the destination of the purchase can be monitored.

b) *b) Type of the Relationship*

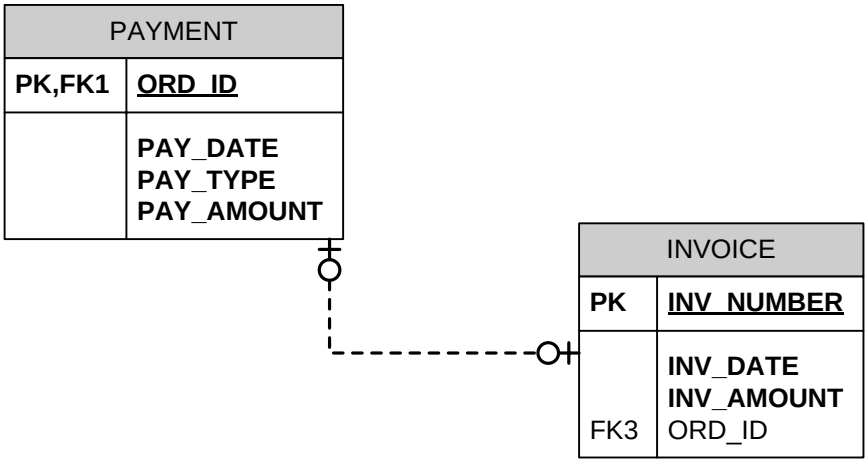
This relationship is a "has a" relationship, each invoice has shipping information. This relationship is strong since an invoice cannot be generated if it has not any shipping information.

c) *Cardinality of The Relationship:*

The relation between invoice and shipping entity is a one-to-one relation. It means that an invoice can have no more and no less than one shipping information. The same from the other side, shipping information can appear

on no more and no less than one invoice. So we have invoice 1-1 relation 1-1 shipping.

Invoice Cart relation



a) *Description of The Relationship:*

When a customer is ready to pay. An invoice should be issued first.

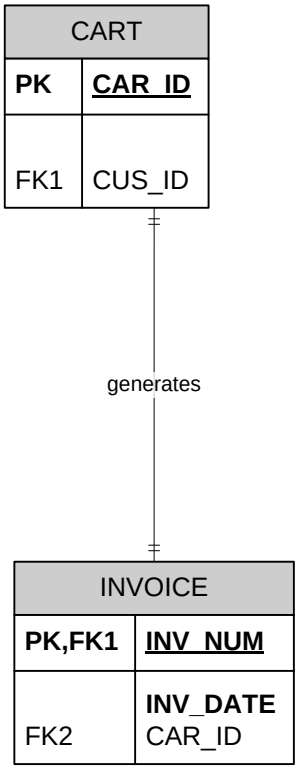
b) *Type of the Relationship*

This relationship is a “generates” relationship, and it’s considered strong

c) *Cardinality of The Relationship:*

The relation between invoice and cart entity is a one-to-one relation. It means that every invoice generates no more and no less than one payment.

Invoice Cart relation



d) Description of The Relationship:

The cart-invoice relation enables the system to link the shopping cart to the invoice entity. Every product that has been added to the shopping cart, its quantity and price will be generated in the invoice. Every purchase details can be monitored thanks to this relation. It also offers to the customer written and detailed information about every purchase.

e) Type of the Relationship

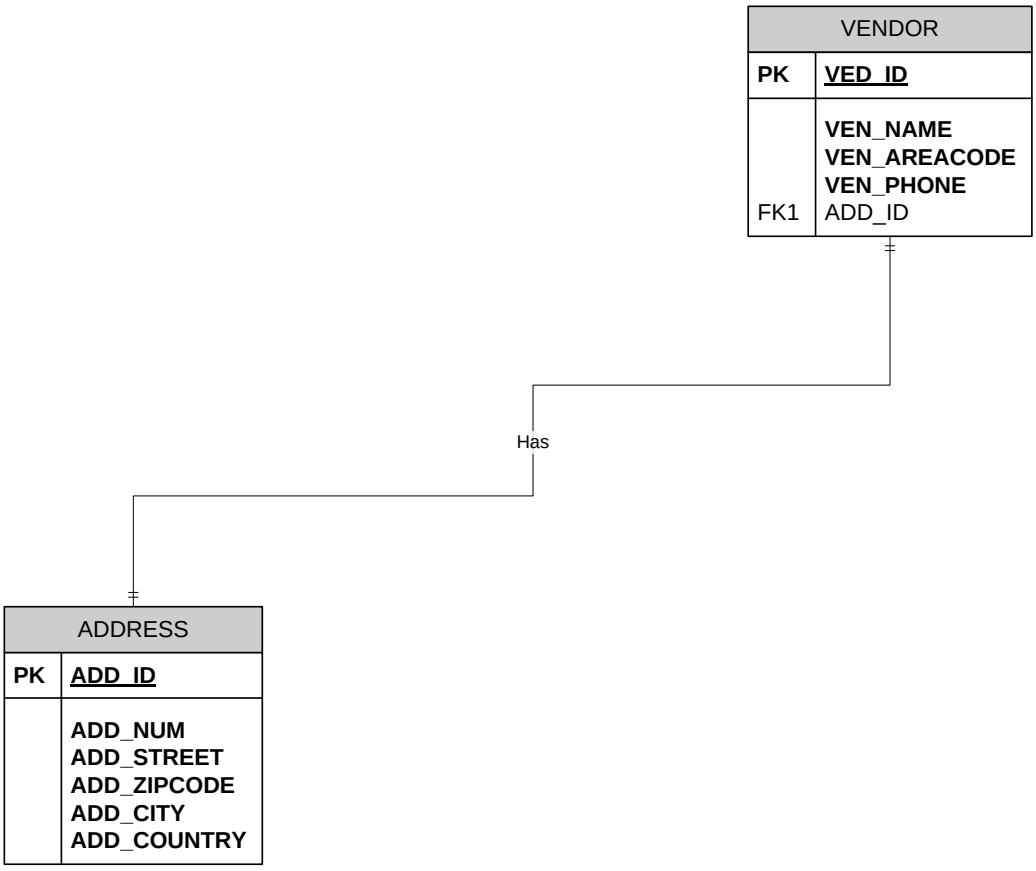
This relationship is a "generates" relationship, and it's considered strong since an invoice cannot be generated without the existence of a shopping cart.

f) Cardinality of The Relationship:

The relation between invoice and cart entity is a one-to-one relation. It means that every shopping cart generates no more and no less than one

invoice. The same goes for the other side, an invoice can be generated by only one cart. So we have invoice 1 to 1 relationship

Vendor Address Relationship



a) *Description of The Relationship:*

The relation enables the system to link each vendor to its address information, thereby enabling each vendor to have its own unique address, which will enable the administrator of the system to sort customers according to their address information to make the shipping logistic easier, as well as searching efficiently the address information about its customers.

b) *Type of The Relationship*

This relationship is a “has a” relationship, each vendor has an address. We can denote that this is a strong relationship, for the reason that the existence of the address itself depends on the existence of the customer existence.

c) *Cardinality of The Relationship*

The Cardinality of the relationship is 1 to 1 relationship.

Shipping Address Relationship

a) *Description of The Relationship:*

The purpose of this relationship is to link between the address of the customer and the address to which he or she wants products delivered. This relationship facilitates calculating the fees of shipping

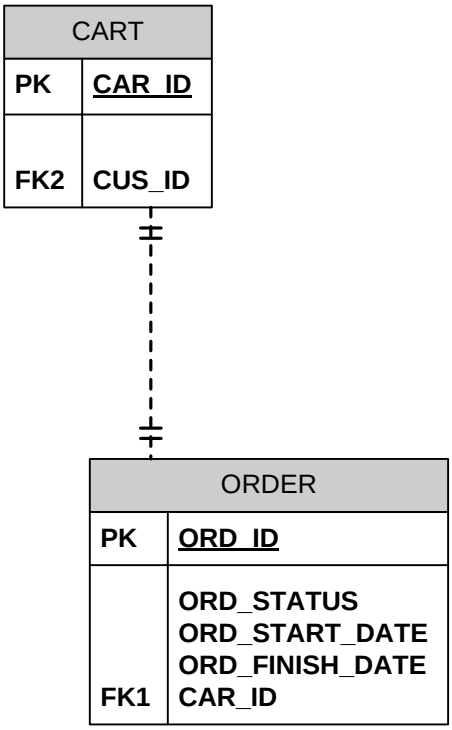
b) *Type of The Relationship*

This relationship is weak because neither of the two entities depend on the other.

c) *Cardinality of The Relationship*

The Cardinality of the relationship is 1 to 1 relationship.

Cart Order Relationship



a) *Description of The Relationship:*

The purpose of this relationship is to link between the cart filled with products with the corresponding order issued by the customer

b) *Type of The Relationship*

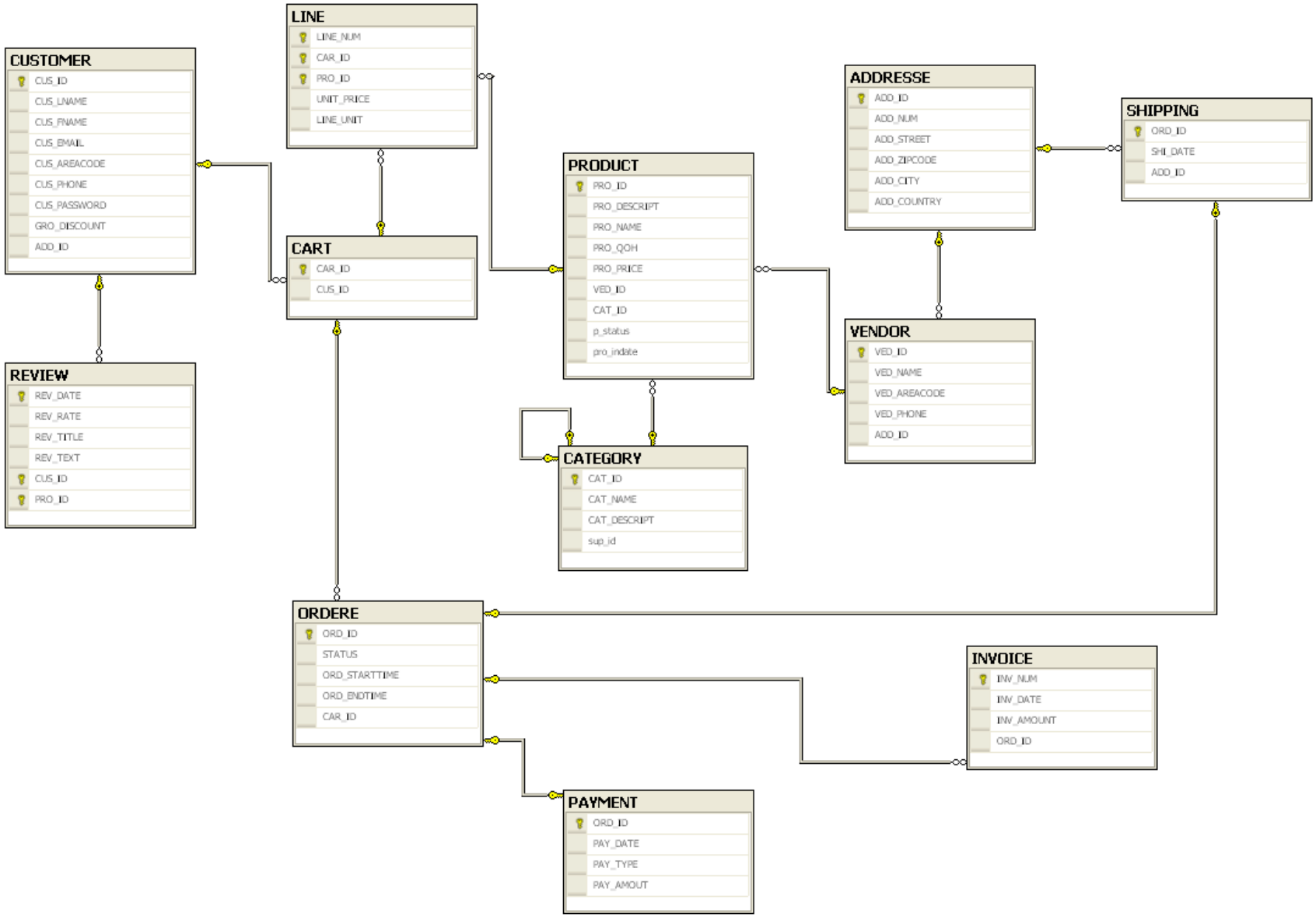
This relationship is weak because neither of the two entities depend on the other.

c) *Cardinality of The Relationship*

The Cardinality of the relationship is 1 to 1 relationship.

Logical Schema

The following figure is generated automatically by Microsoft SQL server 2008


CUSTOMER
CUS_ID
CUS_LNAME
CUS_FNAME
CUS_EMAIL
CUS_AREACODE
CUS_PHONE
CUS_PASSWORD
GRO_DISCOUNT
ADD_ID
LINE
LINE_NUM
CAR_ID
PRO_ID
UNIT_PRICE
LINE_UNIT
CART
CAR_ID
CUS_ID
PRODUCT
PRO_ID
PRO_DESCRIPT
PRO_NAME
PRO_QOH
PRO_PRICE
VED_ID
CAT_ID
p_status
pro_indate
ADDRESSE
ADD_ID
ADD_NUM
ADD_STREET
ADD_ZIPCODE
ADD_CITY
ADD_COUNTRY
SHIPPING
ORD_ID
SHI_DATE
ADD_ID
VENDOR
VED_ID
VED_NAME
VED_AREACODE
VED_PHONE
ADD_ID
REVIEW
REV_DATE
REV_RATE
REV_TITLE
REV_TEXT
CUS_ID
PRO_ID
CATEGORY
CAT_ID
CAT_NAME
CAT_DESCRIPT
sup_id
ORDERE
ORD_ID
STATUS
ORD_STARTTIME
ORD_ENDTIME
CAR_ID
INVOICE
INV_NUM
INV_DATE
INV_AMOUNT
ORD_ID
PAYMENT
ORD_ID
PAY_DATE
PAY_TYPE
PAY_AMOUT

Implementation

Creation of the Database

/*Creating the table of category*/

```
create table CATEGORY (
CAT_ID INTEGER PRImARY KEY not null,
CAT_NAME varchar(30) not null,
CAT_DESCRIPT varchar(100) not null,
SUP_ID INTEGER not null,
constraint SUP_ID foreign key (CAT_ID) references CATEGORY
(CAT_ID)
)
```

/*Creating the table of customer*/

```
CREATE TABLE CUSTOMER (
CUS_ID    INTEGER PRIMARY KEY,
CUS_LNAME VARCHAR(15) NOT NULL,
CUS_FNAME VARCHAR(15) NOT NULL,
CUS_EMAIL varchar(20),
CUS_AREACODE   CHAR(5)  NOT NULL,
CUS_PHONE CHAR(8) NOT NULL,
CUS_PASSWORD VARCHAR(20) not null,
GRO_DISCOUNT INTEGER not null,
ADD_ID INTEGER not null,
CONSTRAINT CUSTOMER_GROUP_FK FOREIGN KEY (GRO_DISCOUNT)
REFERENCES GROUPE (GRO_DISCOUNT),
CONSTRAINT ADDRESS_CUSTOMER_FK FOREIGN KEY(ADD_ID) REFERENCES
ADDRESSE(ADD_ID))
```

/*Creating the table of Groupe (Group is a reserved word)*/

```
CREATE TABLE GROUPE(
GRO_DISCOUNT INTEGER not null primary key,
GRO_NAME varchar(20) not null,
GRO_DESCRIPT varchar(100) not null,
)
```

/*Creating the table of Addresse (address is a reserved word)*/

```
CREATE TABLE ADDRESSE(
ADD_ID INTEGER not null primary key,
ADD_NUM INTEGER NOT NULL,
```

```
ADD_STREET VARCHAR(20) not null,
ADD_ZIPCODE INTEGER not null,
ADD_CITY varchar(20) not null,
ADD_COUNTRY varchar(20) not null,
)
```

/*Creating the table of Review (address is a reserved word)*/

```
create table REVIEW (

REV_DATE DATETIME not null,
REV_RATE INTEGER not null,
REV_TITLE VARCHAR(20) not null,
REV_TEXT VARCHAR(100) not null,
CUS_ID INTEGER not null,
PRO_ID INTEGER not null,
constraint REVIEW_PK primary key(REV_DATE, CUS_ID,PRO_ID),
constraint REVIEW_CUSTOMER_FK foreign key (CUS_ID) references
CUSTOMER (CUS_ID),
constraint REVIEW_PRODUCT_FK foreign key (PRO_ID) references
PRODUCT (PRO_ID)

)
```

/*Creating the table of cart*/

```
CREATE TABLE CART (
CAR_ID    INTEGER PRIMARY KEY,
CUS_ID  INTEGER NOT NULL,
CONSTRAINT CART_CUS_ID_FK FOREIGN KEY (CUS_ID) REFERENCES
CUSTOMER(CUS_ID));
```

/*Creating the table of line*/

```
CREATE TABLE LINE (
LINE_NUM INTEGER NOT NULL,
CAR_ID    INTEGER NOT NULL,
PRO_ID  INTEGER NOT NULL,
UNIT_PRICE INTEGER NOT NULL,
LINE_UNIT INTEGER NOT NULL,
PRIMARY KEY (CAR_ID,PRO_ID,LINE_NUM),
CONSTRAINT LINE_PRO_ID_FK FOREIGN KEY (PRO_ID) REFERENCES
PRODUCT(PRO_ID),
CONSTRAINT LINE_CAR_ID_FK FOREIGN KEY (CAR_ID) REFERENCES
CART(CAR_ID));
```

/*Creating the table of invoice*/

```
CREATE TABLE INVOICE (
INV_NUM INTEGER PRIMARY KEY,
CAR_ID INTEGER,
INV_DATE DATETIME NOT NULL,
CONSTRAINT INVOICE_CART_ID_FK FOREIGN KEY (CAR_ID) REFERENCES CART(CAR_ID));
```

/*Creating the table of shipping*/

```
CREATE TABLE SHIPPING (
INV_NUM INTEGER PRIMARY KEY,
SHI_DATE DATETIME ,
SHI_ADRESSE_ID INTEGER NOT NULL,
FOREIGN KEY (SHI_ADRESSE_ID) REFERENCES ADDRESSE(ADD_ID),
FOREIGN KEY (INV_NUM) REFERENCES INVOICE(INV_NUM));
```

/*Creating the table of vendor*/

```
CREATE TABLE VENDOR (
VED_ID          INTEGER,
VED_NAME        VARCHAR(35) NOT NULL,
VED_AREACODE    CHAR(5) NOT NULL,
VED_PHONE       CHAR(8) NOT NULL,
ADD_ID INTEGER
PRIMARY KEY (VED_ID),
FOREIGN KEY (ADD_ID) REFERENCES ADDRESSE(ADD_ID));
```

/*Creating the table of product*/

```
CREATE TABLE PRODUCT (
PRO_ID    VARCHAR(10) PRIMARY KEY,
PRO_DESCRIPT    VARCHAR(35) NOT NULL,
PRO_NAME        VARCHAR(35) NOT NULL,
PRO_QOH         INTEGER NOT NULL,
PRO_PRICE       NUMERIC(8,2) NOT NULL,
VED_ID               INTEGER,
CONSTRAINT PRODUCT_VED_CODE_FK
FOREIGN KEY (VED_ID) REFERENCES VENDOR(VED_ID));
```

Population of the Database

Below are screen shots of our tables populated.

TABLE OF ADDRESSES

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from addresse
```

Results | Messages

	ADD_ID	ADD_N...	ADD_STREET	ADD_ZIPCODE	ADD_CITY	ADD_COUNTRY
1	1	12	Mohamed V	23450	CasaBlanca	Morocco
2	2	35	St James	26879	London	United Kingdom
3	3	45	DARB BOUAZA	50001	CASABLANCA	Morocco
4	4	83	SIDI BOUZEKRI	46982	MEKNES	Morocco
5	5	666	Wall Street	369581	NEW YORK	U.S.A
6	6	13	L7BASSA	911	ST BERNARD	Morocco
7	7	2912	Rue Camillia	10000	Rabat	Morocco
8	8	1988	Rue La Paix	25000	Paris	France
9	9	1108	Rue al Mouqawama	10000	Rabat	Morocco
10	10	2803	Hay Anarjiss	31000	Fes	Morocco

TABLE OF CART

SSEDB.khobalatte_a - SQLQuery2.sql*

```
select * from cart
```

Results | Messages

	CAR_ID	CUS_ID
1	1	2
2	2	8
3	3	4
4	4	3
5	5	5
6	6	6
7	7	1
8	8	7

TABLE OF CART

SSEDB.khobalatte_a - SQLQuery2.sql*

```
select * from cart
```

Results | Messages

	CAR_ID	CUS_ID
1	1	2
2	2	8
3	3	4
4	4	3
5	5	5
6	6	6
7	7	1
8	8	7

TABLE OF PAYMENT

Table - dbo.PAYMENT | Table - dbo.ORDERE | Table - dbo.SHIPPING

	ORD_ID	PAY_DATE	PAY_TYPE	PAY_AMOUT
▶	2	10/1/2010 12:0...	VISA	10000.00
	4	5/5/2010 12:00:...	MASTERCARD	5000.00
	5	10/1/2010 12:0...	CHEQUE	20000.00
*	NULL	NULL	NULL	NULL

TABLE OF CATEGORY

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from category
```

Results | Messages

	CAT_ID	CAT_NAME	CAT_DESCRIPT	sup_id
1	1	TVs	Different types of TVs	NULL
2	2	LCD	Groups LCDs	1
3	3	Laptops	Compûter Laptops	4
4	4	Computers	Category of all computer	NULL
5	5	Mobile	For Mobile Phones and awesome portable staff	NULL
6	6	Blackberry	Cool Mobile PDAs	5
7	7	CarDesign	All About Car Design	NULL
8	8	Rims&Tires	Cars Rims and Tires	7
9	9	Spoilers	Cars spoilers design and look	7

TABLE OF CUSTOMER

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from customer
```

Results | Messages

	CUS_ID	CUS_LNAME	CUS_FNAME	CUS_EMAIL	CUS_AREACODE	CUS_PHONE	CUS_PASSWORD	GRO_DISCOUNT	ADD_ID
1	1	Khobalatte	Mohamed Wael	m.khobalatte@aui.ma	24350	15690326	xoxx	2	3
2	2	Khobalatte	Ayoub	a.khobalatte@aui.ma	24350	75902645	nuts	7	3
3	3	Benhallam	Wassim	w.benhallam@aui.ma	5550	49360126	pass3	3	3
4	4	Benhallam	Alae	rikitenchu.com	5550	71314906	secret4	4	4
5	5	Elouafiq	Ali	a.elouafiq@aui.ma	20209	54616708	alongpassword	5	5
6	6	Ouaazki	Abdessalam	a.ouaazki@aui.ma	20209	89744723	alongpassword	6	6
7	7	Khrifi	Med Redouane	m.khrifi@aui.ma	Blg38	3079	_PassA	7	7
8	8	Lmalki	El Bachir	e.lmalki@aui.ma	Blg38	3079	_PassB	8	8

TABLE OF GROUPE

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from groupe
```

khobalatte_a - SQLQuery2.sql*

	GRO_DISCOUNT	GRO_NAME	GRO_DESCRIPT
1	1	bronze	This discount concerns our new customer
2	2	bronze	This discount concerns our new customer
3	3	bronze	This discount concerns new customers
4	4	bronze	This discount concerns new customers
5	5	SILVERdiscount	an awesome discount
6	6	SILVERdiscount	an even more awesome discount
7	7	GOLD	Discount of 5% for all customers belonging to th...
8	8	PLATINUM	Discount of 10% for all customers belonging to t...

TABLE OF SHIPPING

Table - dbo.ORDERE | **Table - dbo.SHIPPING**

	ORD_ID	SHI_DATE	ADD_ID
▶	2	1/1/2011 12:00:...	2
	4	5/5/2011 12:00:...	5
*	*NULL*	*NULL*	*NULL*

TABLE OF ORDERE

Table - dbo.ORDERE | Table - dbo.SHIPPING

ORD_ID	STATUS	ORD_STARTTIME	ORD_ENDTIME	CAR_ID
1	0	1/1/2010 12:00:...	NULL	1
2	1	1/1/2010 12:00:...	3/1/2010 12:00:...	2
3	1	2/10/2009 12:0...	3/10/2009 12:0...	3
4	1	5/5/2008 12:00:...	5/5/2007 12:00:...	5
5	0	1/1/2010 12:00:...	NULL	6
11	1	10/23/1990 12:...	NULL	6
123	1	4/15/2010 12:0...	4/15/2010 12:0...	4
654	1	4/28/2010 12:0...	4/28/2010 12:0...	1
NULL	NULL	NULL	NULL	NULL

TABLE OF INVOICE

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from invoice
```

Results | Messages

	INV_NUM	CAR_ID	INV_DATE
1	1	8	2002-01-11 00:00:00.000
2	2	7	2002-08-31 00:00:00.000
3	3	4	2006-05-24 00:00:00.000
4	4	3	2010-09-19 00:00:00.000
5	5	5	2010-04-03 00:00:00.000
6	6	6	2012-02-06 00:00:00.000
7	7	7	2009-12-12 00:00:00.000
8	8	8	2010-04-11 00:00:00.000

TABLE OF LINE

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from line
```

Results | Messages

	LINE_NUM	CAR_ID	PRO_ID	UNIT_PRICE	LINE_UNIT
1	3	3	3	9600	15
2	4	3	4	13500	7
3	1	5	1	15000	3
4	2	5	2	5000	7
5	5	5	5	5800	9
6	6	5	6	7809	3
7	7	7	10	900	5
8	8	8	15	9102	1

TABLE OF PRODUCT

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from product
```

Results | Messages

	PRO_ID	PRO_DESCRIPT	PRO_NAME	PRO_QOH	PRO_PRICE	VED_ID	CAT_ID
1	1	A 21 inches LCD from Sony	LCD Sony	20	15000.00	1	2
2	2	A console from Sony	Playstation	200	5000.00	1	7
3	3	very powerful computer	HP-pavillon dv6	10	9600.00	4	3
4	4	Dual core PC computer	Lenovo 9562	5	13500.00	3	4
5	5	Awesome Gadget	Pearl 8100	100	2580.00	5	6
6	6	Awesome Gadget	7200 series	100	5890.00	5	6
7	10	Double Golden Spoiler	Super Spoiler	12	999.99	8	9
8	15	the new LG LCD screens generation	LG S	2	7125.00	7	2

TABLE OF REVIEW

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from review
```

Results | Messages

	REV_DATE	REV_RATE	REV_TITLE	REV_TEXT	CUS_ID	PRO_ID
1	1999-09-11 00:00:00.000	5	excelleent product	Incredibly excellent	4	3
2	2003-04-18 00:00:00.000	2	Bad LCD Resolution	I was disappointed, HD movies looks horrible on ...	4	2
3	2008-01-06 00:00:00.000	6	birthday Gift	What a beautiful gift, beautiful and cut ^^	6	5
4	2010-01-04 00:00:00.000	5	Question	how to use the user manual?	7	1
5	2010-03-18 00:00:00.000	4	Awesome Snoy product	Playstations are my favorite, Sony proved it is the...	4	1
6	2010-12-01 00:00:00.000	3	PC problem	Lenovo is terribly bad	3	4
7	2010-12-03 00:00:00.000	5	Awsome	that is the best product I have ever had	1	10
8	2012-12-12 00:00:00.000	5	Waming	this expires at the end of the world	5	6

TABLE OF SHIPPING

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from shipping
```

Results | Messages

	INV_NUM	SHI_DATE	SHI_ADRESSE_ID
1	1	2002-01-11 00:00:00.000	1
2	2	2002-08-31 00:00:00.000	2
3	3	2000-01-29 00:00:00.000	4
4	4	2001-08-11 00:00:00.000	3
5	5	2010-02-03 00:00:00.000	5
6	6	2012-04-06 00:00:00.000	6
7	7	2010-10-29 00:00:00.000	7
8	8	2011-01-12 00:00:00.000	8

TABLE OF VENDOR

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select * from vendor
```

Results | Messages

	VED_ID	VED_NAME	VED_AREACODE	VED_PHONE	ADD_ID
1	1	Sony	13256	45896321	4
2	2	Apple	63215	13002574	7
3	3	Hp Company	6545	55589745	4
4	4	Compaq	6545	39685478	3
5	5	vandersar	98985	0	5
6	6	jilali	6666	0	6
7	7	Rooney	lfr	2222	8
8	8	scholes	Brs	1234	7

Manipulating the Database (Queries)

Purpose

Display product name of products purchased by customer whose id is 1

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select p.pro_name from
(product p inner join review r on p.pro_id=r.pro_id) inner join customer c
on r.cus_id = c.cus_id
where c.cus_id=1
```

Results | Messages

	pro_name
1	Super Spoiler

Purpose

Give the list of carts with the number of their respective issued invoices

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select car_id, count(inv_num) as tot_invoice
from invoice group by car_id
```

Results | Messages

	car_id	tot_invoice
1	3	1
2	4	1
3	5	1
4	6	1
5	7	2
6	8	2

Purpose

Give the minimum price for each available category.

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select c.cat_name, MIN(pro_price) as MinPerCategory
from product p inner join category c
on p.cat_id = c.cat_id group by c.cat_name
```

Results | Messages

	cat_name	MinPerCategory
1	Blackberry	2580.00
2	CarDesign	5000.00
3	Computers	13500.00
4	Laptops	9600.00
5	LCD	7125.00
6	Spoilers	999.99

Purpose

Display all the products that supplied by vendor which id is 5

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select pro_id, pro_name, pro_price, pro_qoh
from product where ved_id = 5
```

Results | Messages

	pro_id	pro_name	pro_price	pro_qoh
1	5	Pearl 8100	2580.00	100
2	6	7200 series	5890.00	100

Purpose

Display all categories available with their direct sub-categories.

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select C1.cat_name as parent, C2.cat_name as child
from category C1 inner join category C2
on C1.cat_id = C2.sup_id
order by parent, child
```

Results | Messages

	parent	child
1	CarDesign	Rims&Tires
2	CarDesign	Spoilers
3	Computers	Laptops
4	Mobile	Blackberry
5	TVs	LCD

Purpose

Display product's name and price, its category and, customer's whole name who bought it and to where it's shipped.

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
select C.cus_lname, c.cus_fname,
p.pro_name,p.pro_price,
a.add_city as shipping_city, a.add_country as shipping_country
from customer c ,product p ,shipping s ,line l ,cart m ,invoice i ,addresse a
where c.cus_id = m.cus_id
and l.car_id = m.car_id
and p.pro_id = l.pro_id
and i.car_id = m.car_id
and i.inv_num = s.inv_num
and s.shi_adresse_id = a.add_id
```

Results | Messages

	cus_lname	cus_fname	pro_name	pro_price	shipping_city	shipping_country
1	Khrifi	Med Redouane	LG S	7125.00	CasaBlanca	Morocco
2	Khobalatte	Mohamed Wael	Super Spoiler	999.99	London	United Kingdom
3	Benhallam	Alae	HP-pavillon dv6	9600.00	CASABLANCA	Morocco
4	Benhallam	Alae	Lenovo 9562	13500.00	CASABLANCA	Morocco
5	Elouafiq	Ali	LCD Sony	15000.00	NEW YORK	U.S.A
6	Elouafiq	Ali	Playstation	5000.00	NEW YORK	U.S.A
7	Elouafiq	Ali	Pearl 8100	2580.00	NEW YORK	U.S.A
8	Elouafiq	Ali	7200 series	5890.00	NEW YORK	U.S.A
9	Khobalatte	Mohamed Wael	Super Spoiler	999.99	Rabat	Morocco
10	Khrifi	Med Redouane	LG S	7125.00	Paris	France

Purpose

Create index on review to enforce uniqueness of primary key.

Result

SSEDB.khobalatte_a - SQLQuery2.sql* | Summary

```
create unique index PK on review (cus_id, pro_id, rev_date)
```

Messages

```
Command(s) completed successfully.
```

Languages and Tools

Languages and Tools

The team members have chosen C# as development language to tackle the task of implementing the Pirkt. Reasons for adopting this language are its similarity with the Java language, the positive feedback from former Database Systems students, and the nice graphic capabilities it's capable of. The IDE we're using is Visual Studio 2005, a Microsoft product that cooperates with other Microsoft projects. This IDE eases the level of scalability and linkage required from both the desktop and web applications, since both require close connections with the database, a task that Microsoft didn't ignore. Concerning the DBMS, the team members are used to work with Microsoft SQL Server 2005 since it is the one that we are using in the course Lab. Visual Studio can access some of the functionalities of the SQL Server, which proves efficient in most cases.

It would have been a fantastic journey into computer science if we had the chance to explore some other widely web development tools like PHP and tools of database implementation like MySql; nevertheless, the time constraint does not allow going for this alternative.

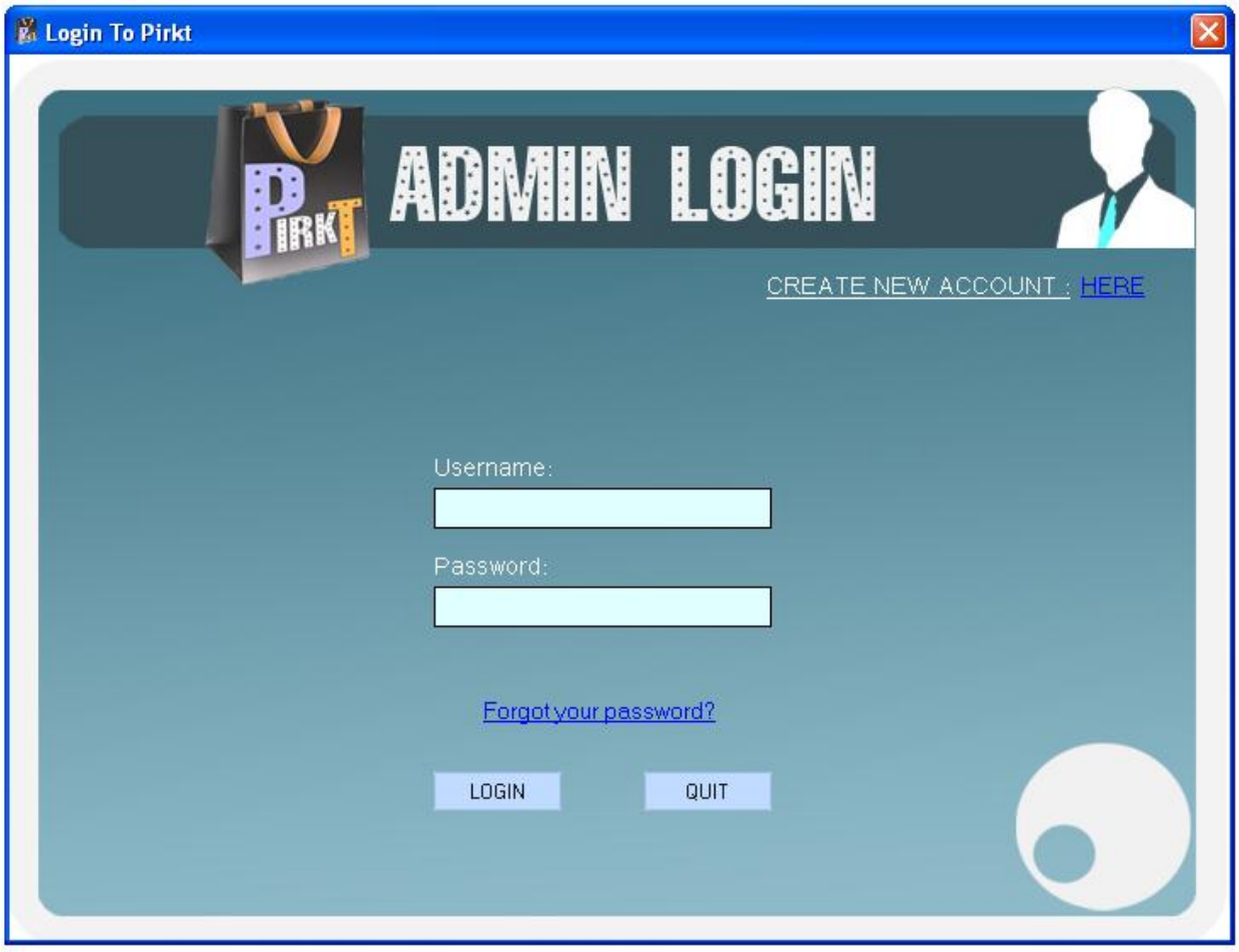
Structure of the Application

The most important piece of the system that the team is developing is surely the user interface (UI); indeed, it is the mean by which customers can make use of all the functionalities of the system in a simple way. The team members made sure, the user interface as user friendly as they could.

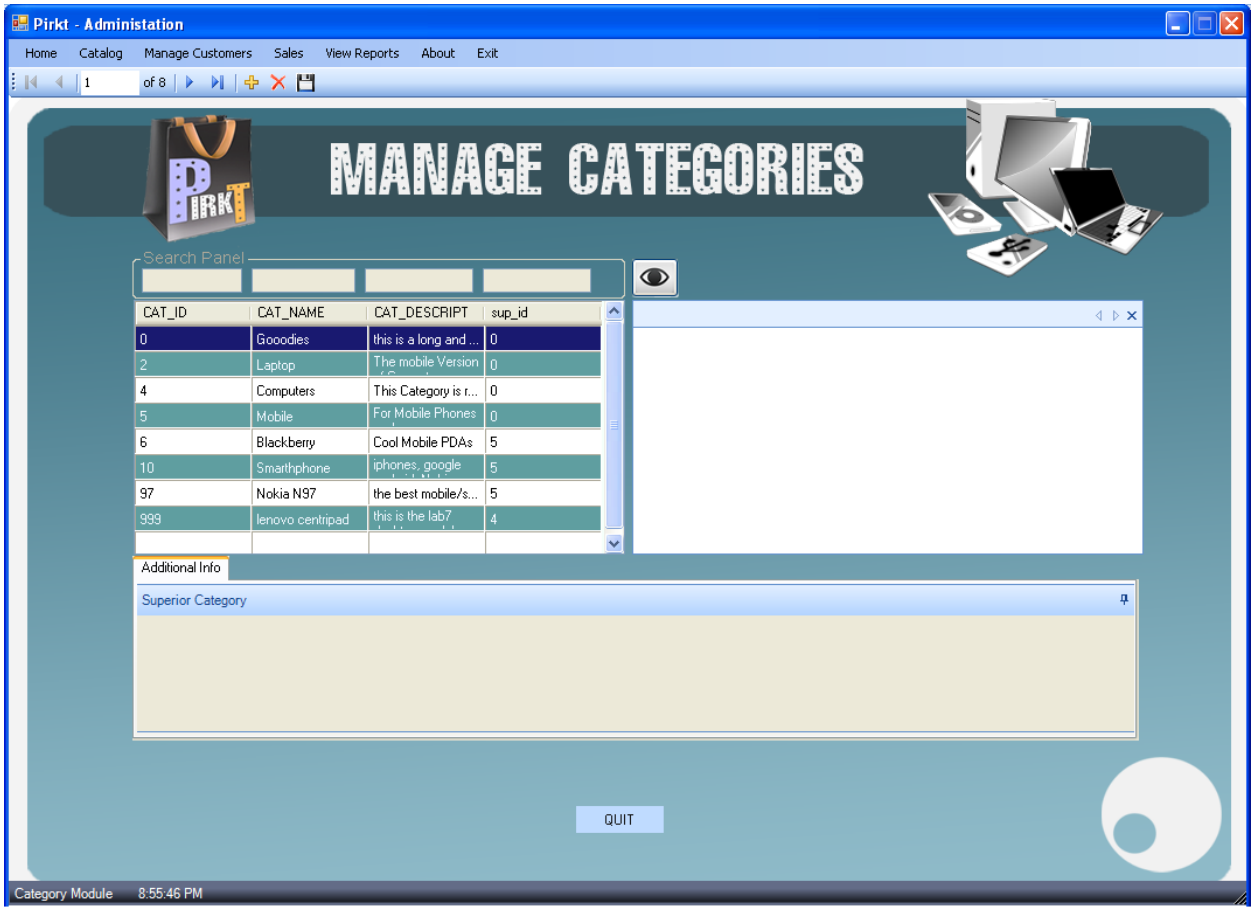
The interface was structured in a way that functionalities are clear to the user which does not mean necessarily displaying contains of the tables as they appear within the database.

The system consists of two separate applications: a desktop application mainly used for maintenance and administration; a web application used as the interface between the PIRKT Organization and customers

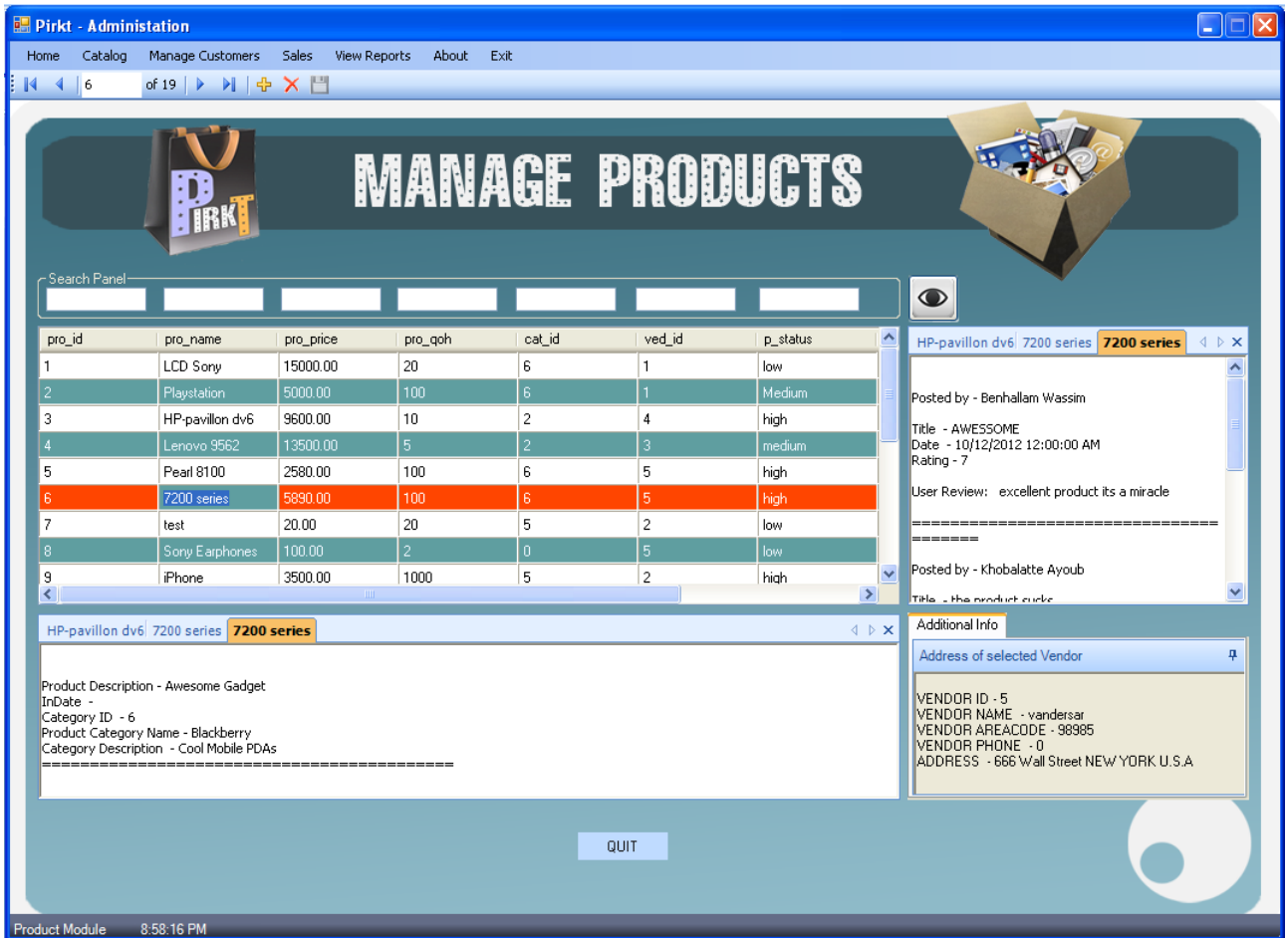
Following are only some screenshots of both the Desktop and the Web applications illustrating some of the functionalities. The team is exploring all the capabilities of C# and the .NET framework in order to provide fully implemented applications.


Login To Pirkt
PIRKT
ADMIN LOGIN
CREATE NEW ACCOUNT : HERE
Username:
Password:
Forgot your password?
LOGIN
QUIT

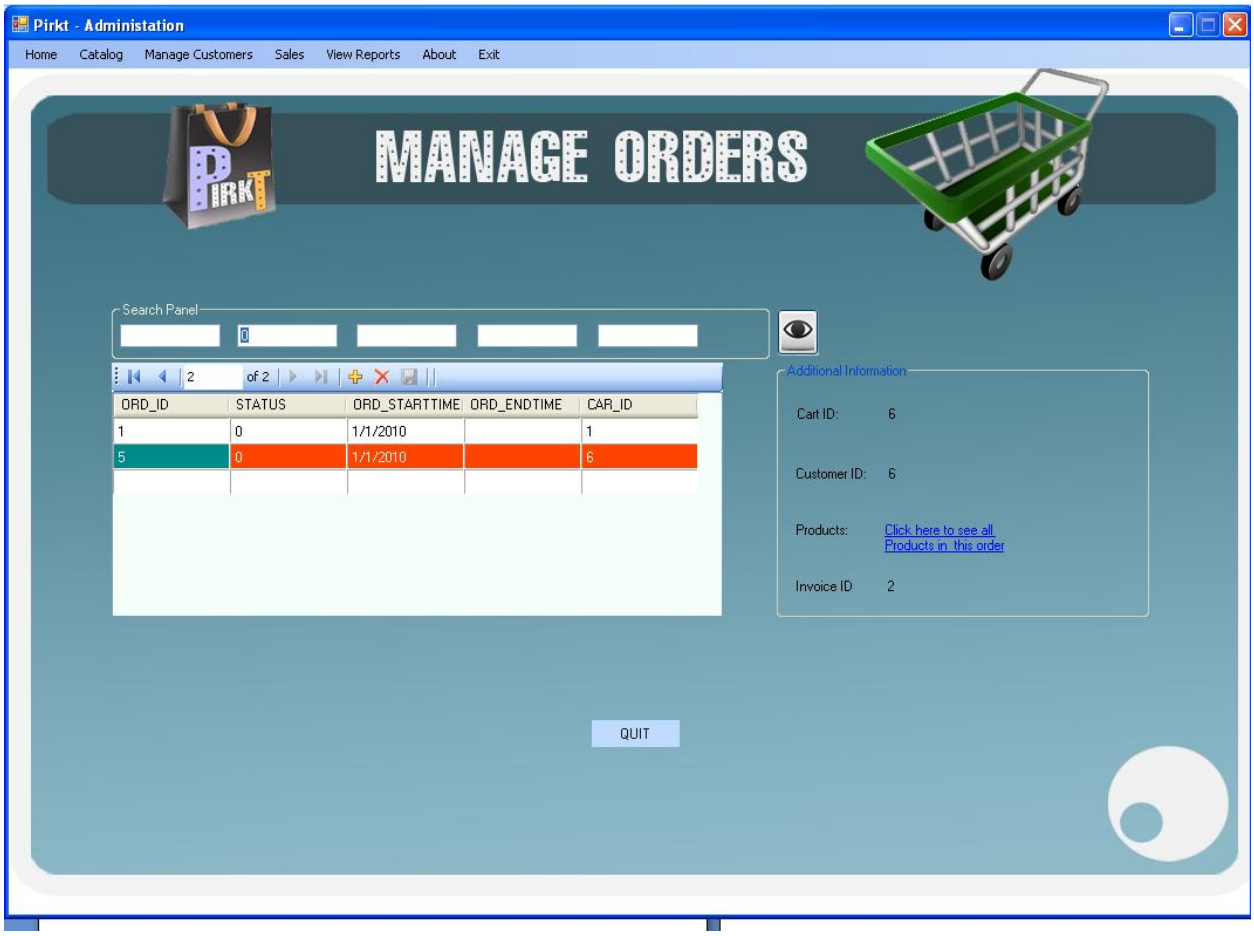
Login Form



Manage Categories Form

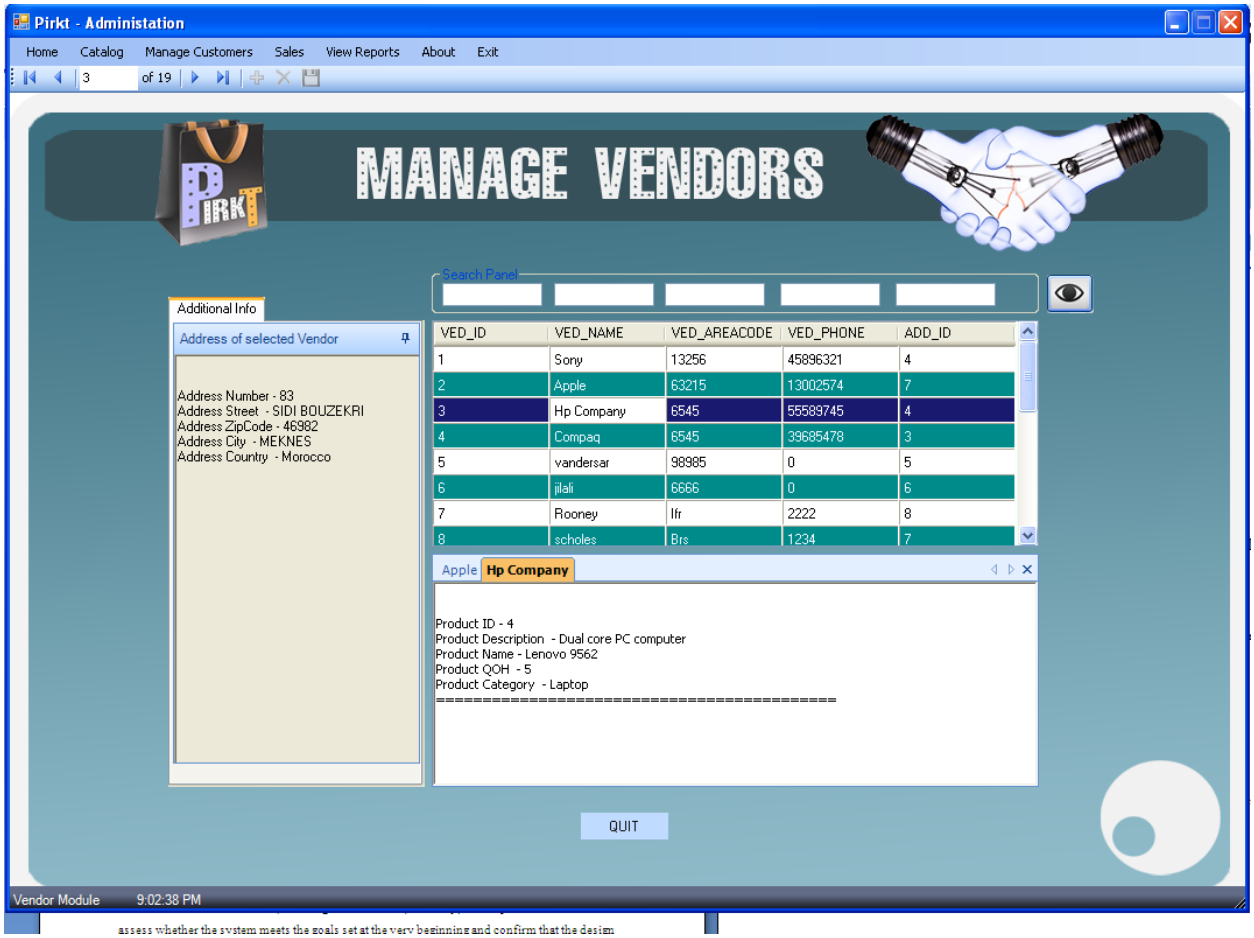


Manage Products Form

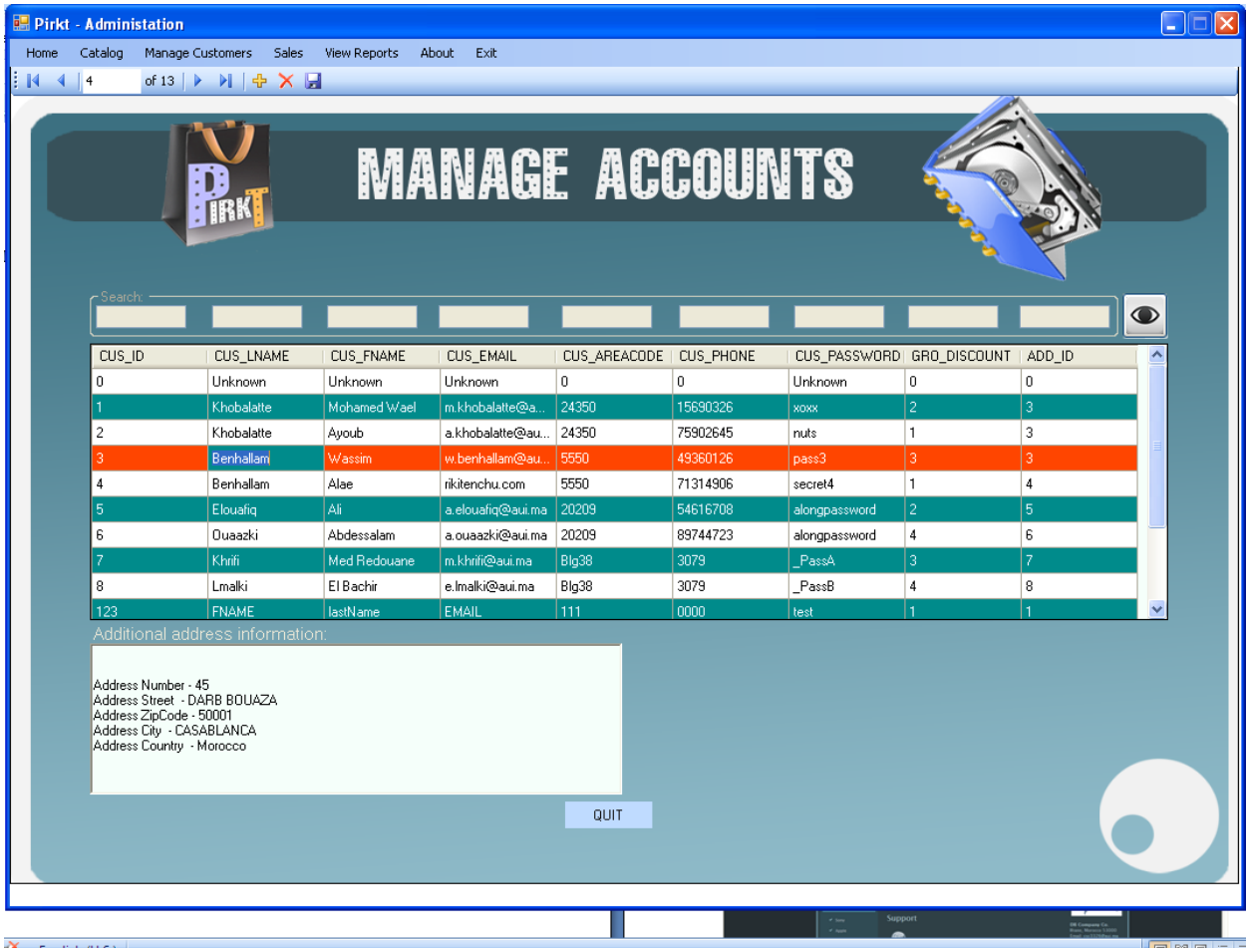


Manage Orders Form

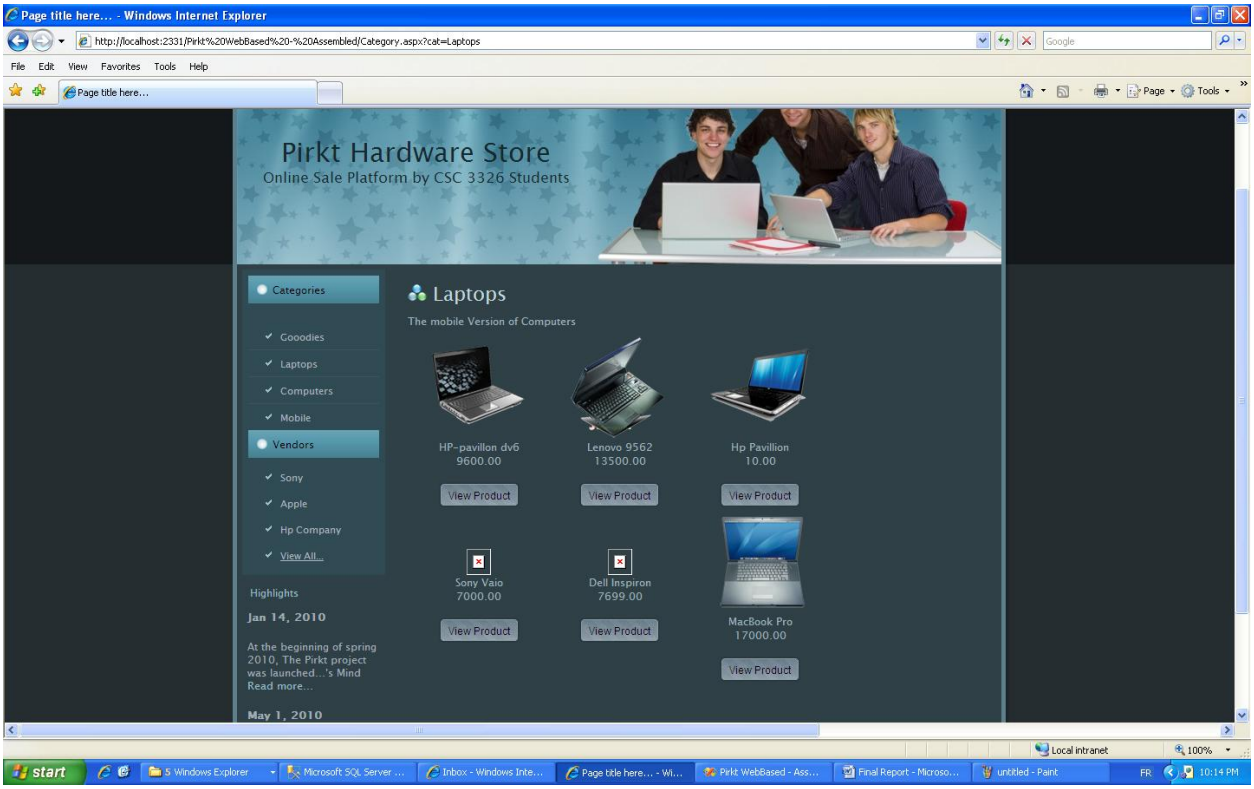
Scenario: when the administrator enters something in the fields above the table, it only shows the rows satisfying the searching criterion



Manage Vendors Form

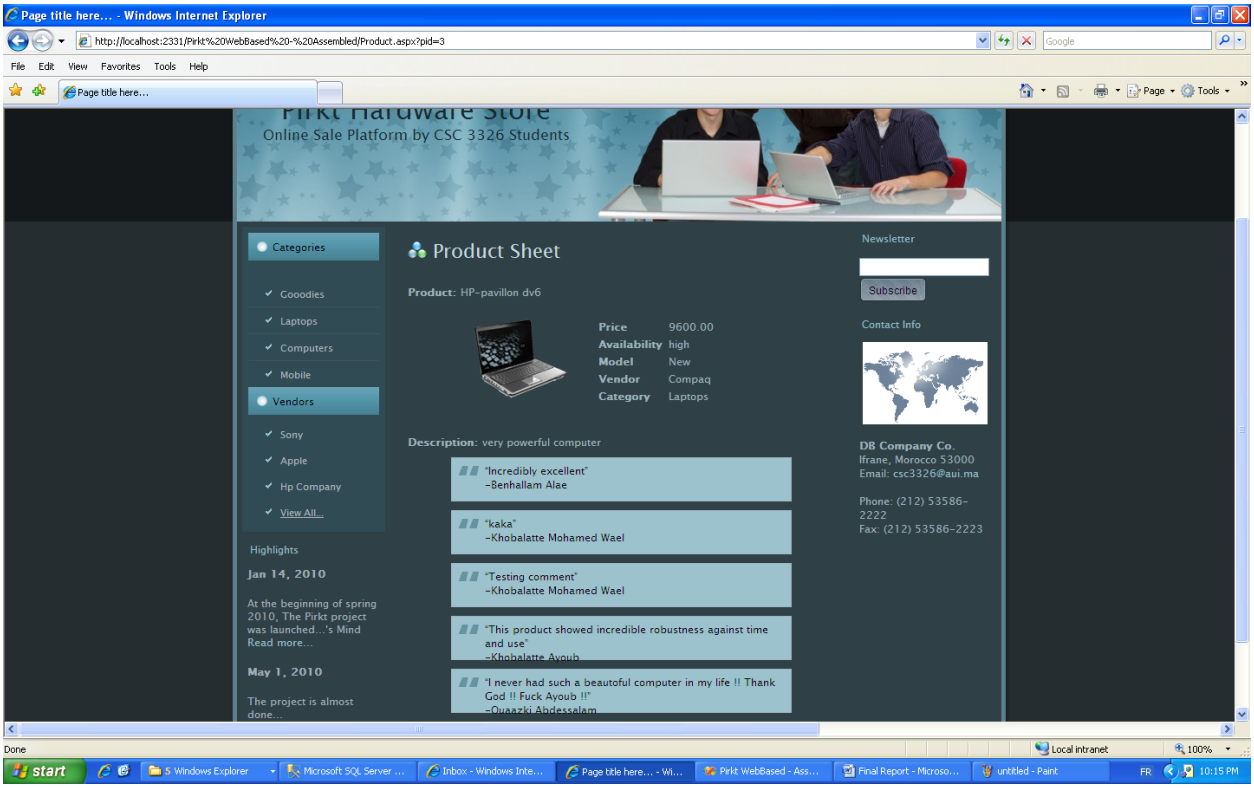

Pirkt - Administation
Home
Catalog
Manage Customers
Sales
View Reports
About
Exit
4
of 13
MANAGE ACCOUNTS
Search:
CUS_ID | CUS_LNAME | CUS_FNAME | CUS_EMAIL | CUS_AREACODE | CUS_PHONE | CUS_PASSWORD | GRO_DISCOUNT | ADD_ID
0 | Unknown | Unknown | Unknown | 0 | 0 | Unknown | 0 | 0
1 | Khobalatte | Mohamed Wael | m.khobalatte@a... | 24350 | 15690326 | xoxx | 2 | 3
2 | Khobalatte | Ayoub | a.khobalatte@au... | 24350 | 75902645 | nuts | 1 | 3
3 | Benhallam | Wassim | w.benhallam@au... | 5550 | 49360126 | pass3 | 3 | 3
4 | Benhallam | Alae | rikitenchu.com | 5550 | 71314906 | secret4 | 1 | 4
5 | Elouafiq | Ali | a.elouafiq@aui.ma | 20209 | 54616708 | alongpassword | 2 | 5
6 | Ouaazki | Abdessalam | a.ouaazki@aui.ma | 20209 | 89744723 | alongpassword | 4 | 6
7 | Khrifi | Med Redouane | m.khrifi@aui.ma | Blg38 | 3079 | _PassA | 3 | 7
8 | Lmalki | El Bachir | e.lmalki@aui.ma | Blg38 | 3079 | _PassB | 4 | 8
123 | FNAME | lastName | EMAIL | 111 | 0000 | test | 1 | 1
Additional address information:
Address Number - 45
Address Street - DARB BOUAZA
Address ZipCode - 50001
Address City - CASABLANCA
Address Country - Morocco
QUIT

Web Application

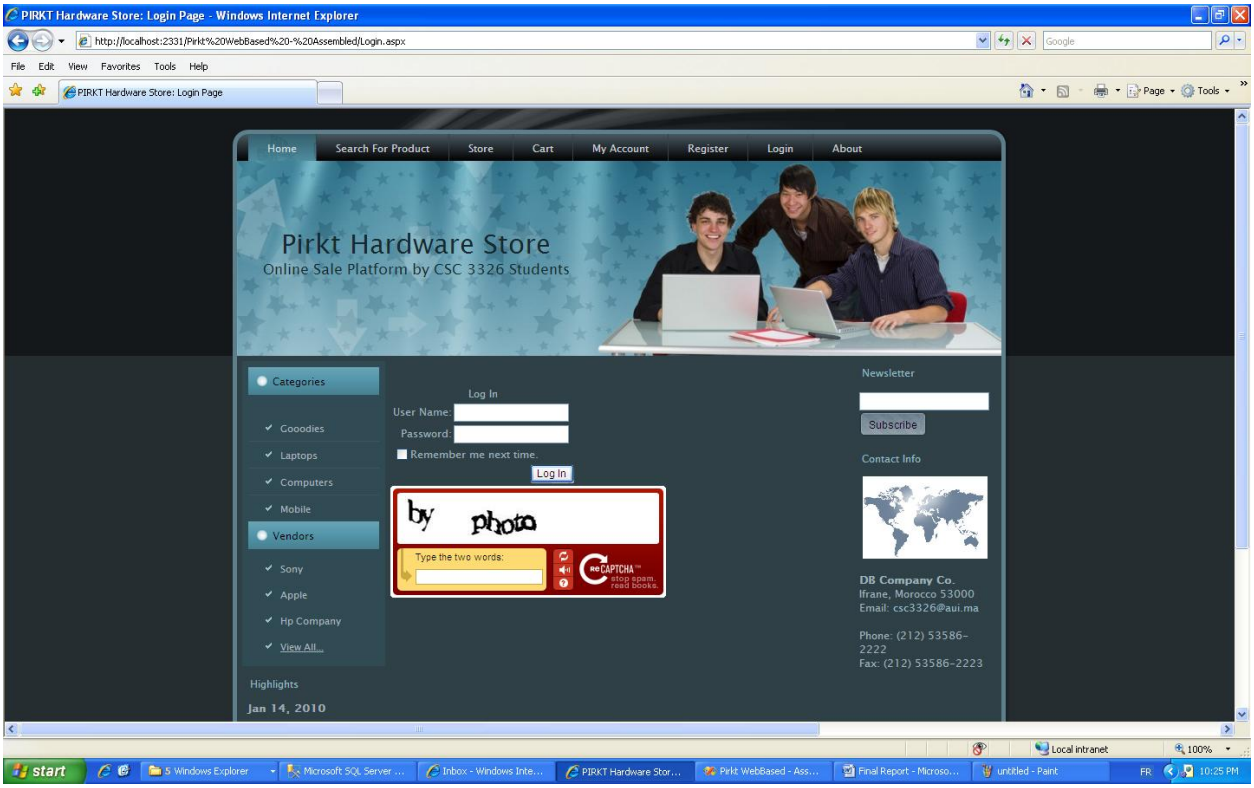


Scenario: When you click on a category button in the menu all the products in that category are shown

Z

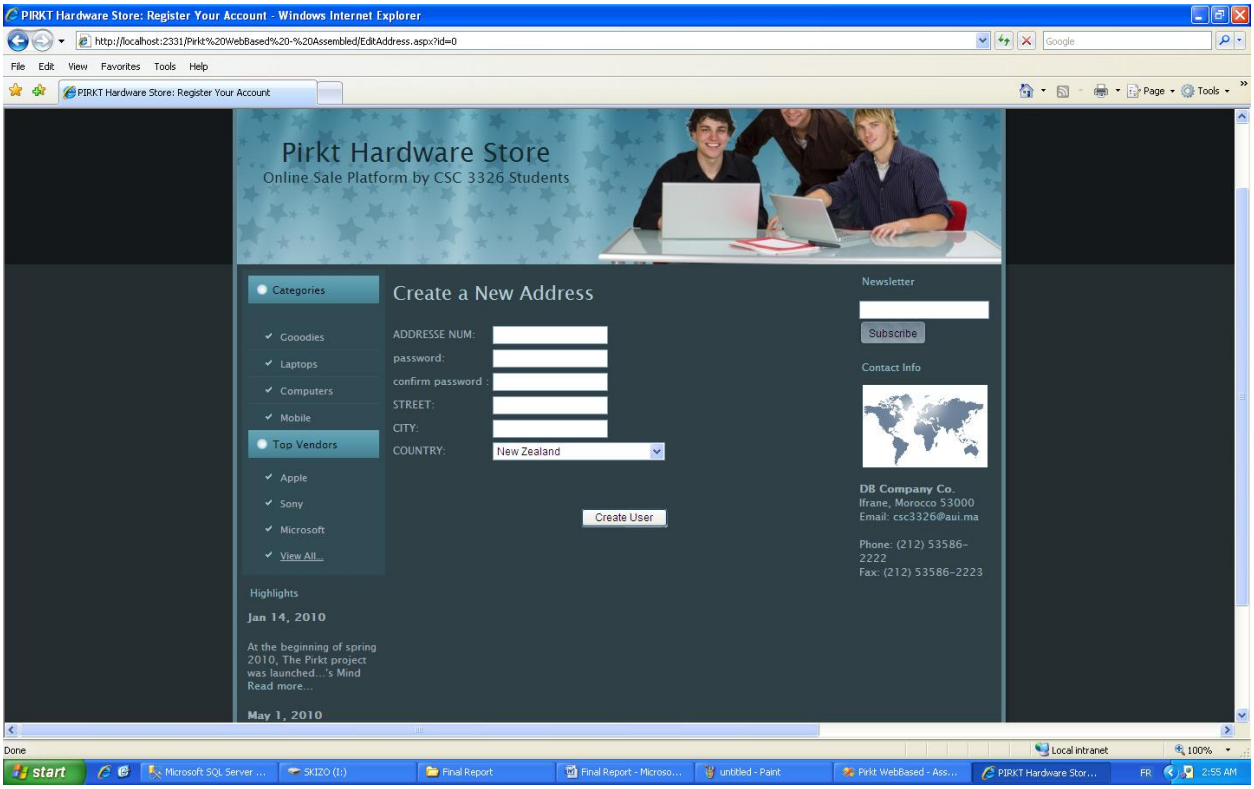


Scenario: When you click on view details button under the product, additional information appears as well as the reviews.

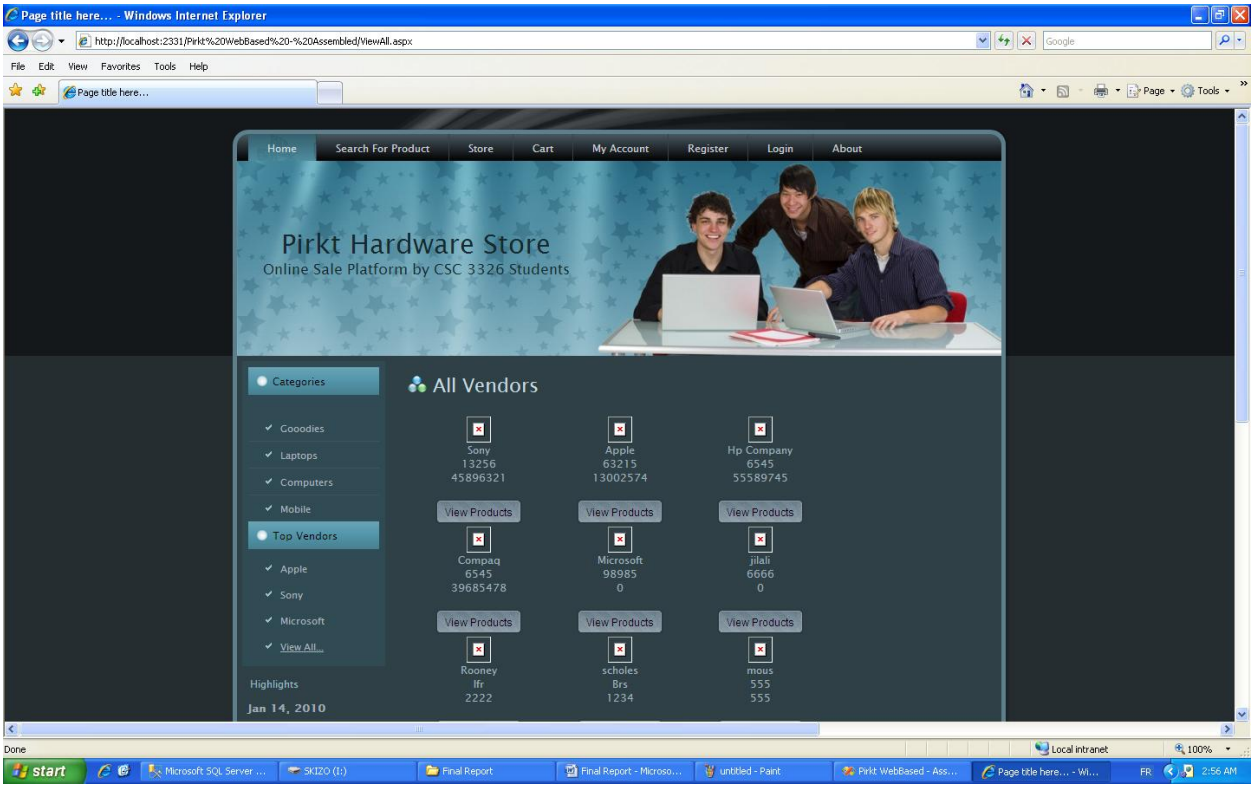


Scenario: this form asks the user to login his or her account following the universal standards in order to purchase an item

Scenario: this form asks the user to register his or her account following the universal standards.



Scenario: A user can add or modify his address after creating the account



Scenario: the list of all vendors

Testing and fine-tuning

Testing is one of the most critical steps in the process of developing a project. We would to ensure, by any means, that our project doesn't run into compiling and runtimes errors. We've been doing this ourselves, that is, we acted as if we were potential customers who are expecting some well developed functionality as well as a flawless and consistent experience.

The testing phase used relevant data that we ourselves have incorporated in our database.

As we've seen in class, data needs some constraints and exceptions to be handled so that we ensure we don't run in some infamous data problems. Our testing mechanism involved, at a later stage, checking numerous scenarios. Feedback was critical, as we ran into some anomalies that successfully overcame later.

User manual

Desktop Application

The PIRKT desktop application is a tool that enables the administrator to manage and monitor the different programs that it offers as well as all the applications and reference letters that have been submitted by customers. The main purpose of the desktop application is to allow the administrator retrieve the needed information easily in order to make some decisions regarding the way the PIRKT organization works.

MAIN FEATURES

- Manage the accounts of the customers registered through the website
- Manage the vendors supplying the products
- Manage the products as well as the categories each product belong to
- Manage the invoices, orders and payment issued by the customers and application.
- View the statistics related to the application
- …

Application Map

The application is composed of the following frames:

- Login frame: a frame asking the administrator to add his or her username and password
- Home Panel
- Manage Customers Panel:
- Manage Invoices Panel
- Manage Order Panel
- Manage Categories Panel
- Manage Products Panel
- Manage Group Panel
- Manage Products Panel
- Manage Vendor Panel
- Statistics and reports

Web Application

The PIRKT hardware store is a website that enables customers to interact with the products offered by PIRKT organization. Its main objective is to make the customers search for their needed products and purchase them.

Main Features

The PIRKT Hardware store Website is made to satisfy the following goals:

- Allow the customers to search for the products that they want to buy
- Add products to line and then deciding whether or not to purchase them

- Create accounts and modify it.
- Be notified whenever a new product is added to the database
- …

SITE MAP

The PIRKT Hardware store Website is made of the following pages:

- Login Page / create new student Profile Page
- Home Page
- Contact Page
- About page
- Customer profile
- products page
- Vendors page
- The buying wizard (adding to cart and confirming the purchase)

Future

The team is aware of the level of involvement as well as improvement the project needs. We plan on extending our website to allow users more functionality. We also would like to support sessions and platform specific user experiences. Mobile version is also a possible outcome depending on how Internet technologies class will handle such concept and its related technologies.

We will ensure that our project is as successful as it should have been if it was adopted by a client.

Conclusion

Though we don’t have a real client, we nonetheless learned some great stuff. Throughout all the project’s different stages, we had the chance to ensure that certain concepts, inherently

important and specific to each phase, were fully grasped. We also learned and developed management and communication skills that are needed in real life circumstances.

As computer science students, we surely recognize then importance of database design and implementation and we can assert that notions related to this discipline were fully understood. We have been introduced to some new tools like the MS SQL Server 2005, one of the strongest DBMS which handles many back office operations. In addition, the team had the chance to use a new programming language which is C# along with ASP.NET to implement both the desktop and the web applications. In order to achieve that was to get familiar with the IDE that Microsoft designed to support c# and ASP.NET, Microsoft Visual Studio.

This project was a unique opportunity for us to see how large scale projects, at least compared to projects implemented in previous computer science classes, should be handled and what are the various skills needed in able to succeed. Different, and certainly constructive, ideas emerged as we went along. We all agreed on the importance of communication, as well management skills and how decisive they are in determining the fate of any project, and certainly ones as detailed and complex as the computers science projects.