SPORT_SHOP

	NAME		N.MEC	PERCENTAGE OF WORK
Authors:	Bruno Lopes	MIECT	89179	50%
	Juan Lessa	LEI	91359	50%

PROJECT NAME	CLASS – GROUP
SPORT_SHOP	P4 – G10

This document describes what SPORT_SHOP really is, a system designed for the maintenance of sports stores and all its customers and workers.

Contents:

REQUIREMENTS ANALYSIS

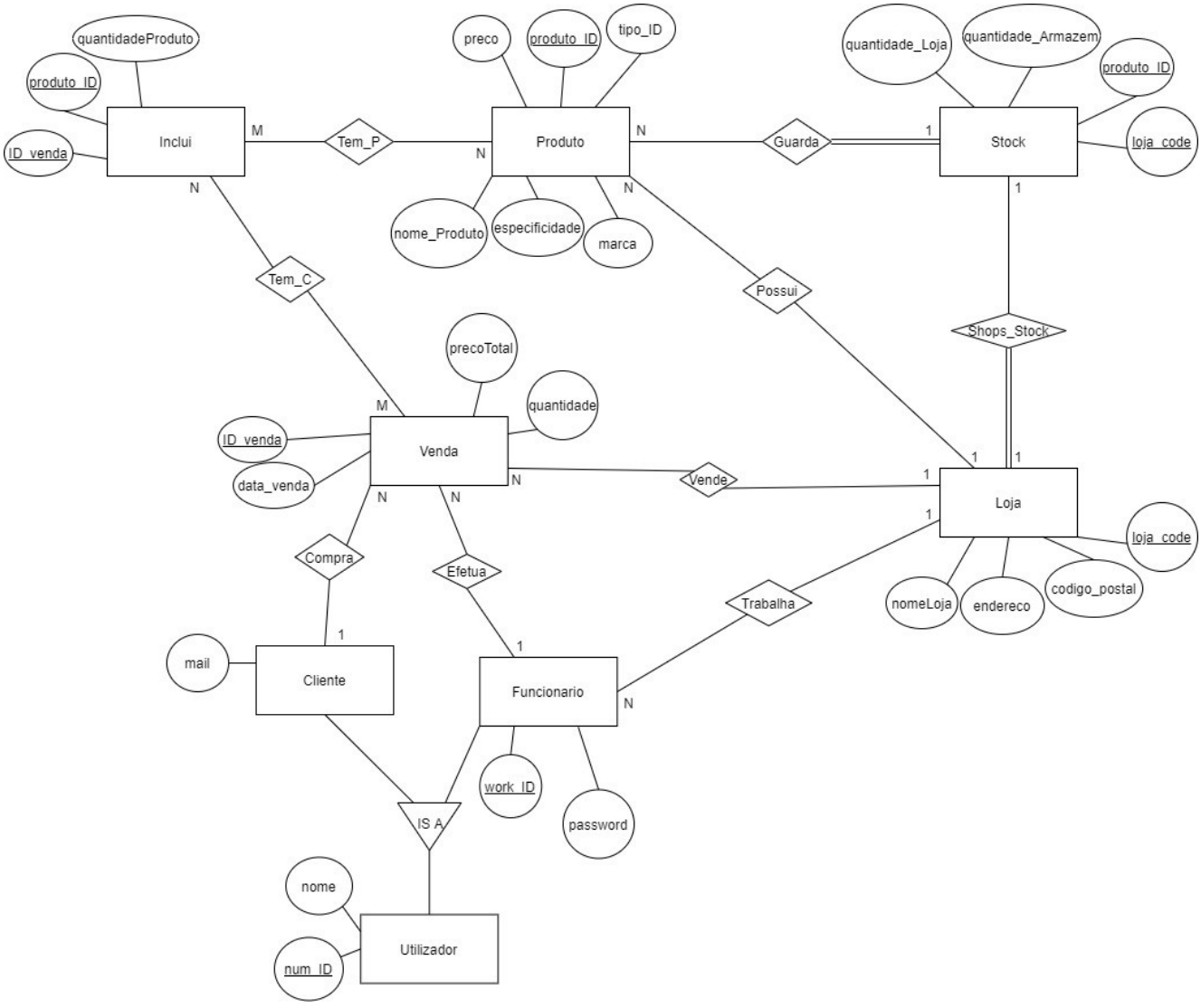
After investigating some of the varied problems in the sports stores area:

- A **problem** was found: It is always very hard to find all kinds of material in large or even small store. Let us look at the example of this already well-known store "Sport Zone", a huge store both in size and in name. In these types of stores it is common to have many types of problems such as taking too long to serve a customer, the same having trouble finding what he wants, etc.. This kind of problems could be easily eradicated just by having a simple but efficient database.
- What gives us an **opportunity**: What if we were able to take most of the workload out of the employees and customers, thus giving them time to be able to focus on their number one priority, a better store management and a more enjoyable shopping trip, respectively. This would prevent not only the amount of work (since, now, the workers would only have to search for the given product in the new system to know whatever they wanted), but also the time spent on endless searches and the number of errors.

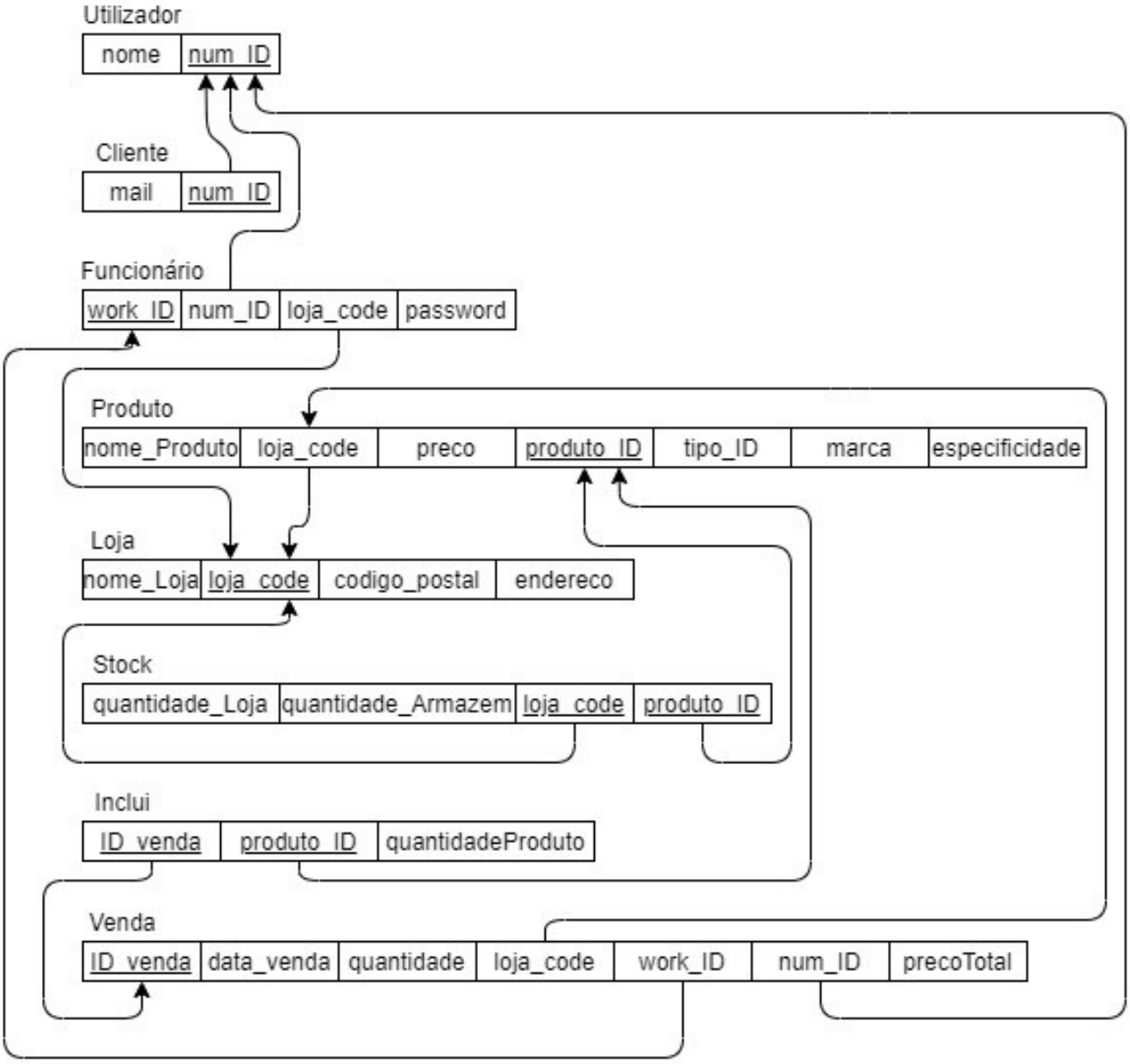
The system should allow users to take advantage of these characteristics:

User	Feature
Everyone	• Be able to see all the products present in the store.
Employee	• Check, insert or remove products from the warehouse stock (for management purposes) • Check, insert or remove products from the store stock (for management purposes) • Access a list of products in the store • Access a list of his sales • Access a list of customers • Create account for a new customer • Remove customer • Access each customer's purchases • Access a list of employees • Create account for a new employee • Remove employee • Access sales of other employees • Access store finances • Make sales
Customer	• Search for products within the store • Shopping

ENTITY-RELANTIONSHIP DIAGRAM (DER)



RELATIONAL MODEL (ER)



SQL DDL

For this access SQL\SQL_DDL.sql.

SQL DML

In the files SQL\SQL_Inserts.sql, SQL\SQL_UpdateStock.sql we can find all the information regarding DML of the system. The Inserts and the Updates were made to add some data to the system at its beginning. After that every Insert and Update is directly called by the interface.

INDEXES

For this access SQL\SQL_Indexes.sql.

In this system there wasn't the need for many indexes since each Entity Primary Keys would, most of the time, solve the problem.

However, we created a single index for the Produto Entity since it is a very consulted table but it isn't modified that much.

TRIGGERS

For this access SQL\SQL_Triggers.sql.

When a new product is created our only trigger (t_Create_Stock) is activated which makes that the newly added product starts with an initial quantity of 0 such in the store as in the warehouse.

STORED PROCEDURES

For this access SQL\SQL_StoredProcedures.sql.

We used Stored Procedures when creating or deleting an object on an Entity derived from another such is the case of creating an employee or a customer, as well as when we wanted to add or remove a given quantity of a given product, to create a product or make a sale.

- Creating or Deleting Employee or Customer (p_Create_Cliente, p_Create_Funcionario, p_Delete_Cliente, p_Delete_Funcionario)
- Adding/Removing Quantities (p_Update_Stock, p_Retira_Stock)
- Create Product (p_Create_Produto)
- Make Sale (p_Create_Venda)

VIDEO FOR THE PRESENTATION

For this access "base de dados APRESENTAÇAO.mp4".

FINAL VIDEO

For this access "Base de Dados Versão final.mp4".

POST PRESENTATION MODIFICATIONS

Before the presentation we had the following features implemented:

- List of Current Employee Sales (Vendas → Lista)
- Details of Sales (Vendas → Quando uma venda estiver selecionada → Detalhes)
- Create Sale (Vendas → Adicionar)
- List of Store Products (Produtos → Lista)

After the presentation, we implemented the features of our system in the interface that were missing or needed to be improved.

These are the features implemented after presentation:

- Create Product (Produtos → Adicionar)
- Filter Products (Produtos → Filtros)
- Filter Sales (Vendas → Filtros)
- Update Stock Quantity (Produtos → Receber Stock)
- Access Store Finances (Finanças)
- List of Customers (Clientes → Lista)
- Add Customer (Clientes → Adicionar)
- Remove Customer (Clientes → Remover)
- List of Employees (Funcionários → Lista)
- Add Employee (Funcionários → Adicionar)
- Remove Employee (Funcionários → Remover)

ATENTION

The script can be accessed on SQL\BD_projeto.sql.

To change the SGBD connection go to Interface\F_login.cs line 71, Interface\F_main.cs line 172 and change the data. As we weren't able to use the connection to our database we left the connection string blank.

CONCLUSIONS

The key aspects we retained from this work are the following:

- We retained the whole process for creating an Application with a proper Database behind – from the Requirements Analysis to the creation of the Conceptual Model and the Relational Model to the creation of an abstraction layer and finalizing in the execution of SQL Queries, Stored Procedures and much more on the application.
- The Relational Model (ER) and the Entity-Relantionship Diagram (DER) were a great help when think and designing the system making it possible to do then a better job.
- Since the beginning of this project a lot of our entities and relationships between them had to be reconsidered and changed for the system to be better built. But that doesn't

mean that every entity is perfect now, it's very probable that some changes in some of them would be very beneficial such as in the Produto entity.

- Encrypting the password should be a thing to do in the future, not just saving it in the Database.