Daraghmeh Company

Database class project 2021/2022

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student Name in English | Student Name in Arabic | Student ID | Section | Work percentage |
| Mohammad Bdair | محمد أحمد بدير | 12027880 | 1 | 99% |
| Mohammad Ahmad | محمد عزمي أحمد | 12028676 | 1 | 1% |

Date/time

---------------------------------------------This section is intended for the Instructor---------------------------------------

|  |  |
| --- | --- |
| **Topic** | **Mark** |
| Project Requirements and Modeling |  |
| Correctness of Database mapping |  |
| Functional Dependency and Normalization |  |
| Project Tools |  |
| Project Discussion |  |
| Project Completeness |  |
| Project Output Results or reporting (JasperReport, charts, graphs, etc.) |  |
| Project Administration and Management |  |
| Project Report |  |
| Project Idea |  |
| Project Complexity |  |
| Team work |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table of Content

[Abstract 2](#_Toc110005036)

[Team work: 2](#_Toc110005037)

[Requirements 3](#_Toc110005038)

[Reports 4](#_Toc110005039)

[Tools used in the project 4](#_Toc110005040)

[Database Schema 4](#_Toc110005041)

[GUI Overview 5](#_Toc110005042)

[EER Crows Foot notation: 30](#_Toc110005043)

[Mapping 30](#_Toc110005044)

[Existence Dependencies 31](#_Toc110005045)

[Functional Dependencies 31](#_Toc110005046)

[Normalization 32](#_Toc110005047)

[Resources 33](#_Toc110005048)

Abstract

Our project organizes Daraghmeh Company, which is a clothes trading company. The project organizes the company’s departments, warehouses and employees on many aspects, in a way that all information stay connected and related to each other to prevent data loss. The project also has a user-friendly interface and many features to make it easier for each employee to deal with.

Introduction

We chose Daraghmeh clothing center that is located in Nablus as the target of our project’s information source, we visited it and asked one of its supervisors about the nature of its data flow, how each manager is distributed on the department/warehouse that he manages, then the nature assistants’ and cashiers’ work.

We asked them about what program would serve them the most, which turns out to be a program that organizes the managers on each department and warehouse. Which also saves and retrieves suppliers information, and most importantly, saving information about every dress sold or imported.

We collected the information that we needed, then we summarized them into simple and organized points so we can make a full functioning program based on them.

Team work:

Both of us partners, learnt how to use javaFX during the holiday before the summer semester, we learnt it from scratch and then created mini projects and simple ideas and saved them so that we can use them in our main project later, we also learnt CSS Styling, then we started learning about Github as the semester started, we used it to upload our files and merge them, it developed a better understanding of reading and writing a code for us, as we solved the complex conflicts, it made it much easier for us to combine our work together, so we would have a perfect outcome, hopefully it is !

Our Github repository link: <https://github.com/mohamad-ahmmad/Database-project.git>

Requirements

\* The company is organized into **departments and Warehouses.** Each **Department** has a Unique ID, name, single location, **warehouses** that provides it with **clothes**, one **manager, assistants,** and **cashiers** who work in it.

\* Each **Warehouse** has a unique ID, single Location, name, capacity and a **manager**.

**\* Employees** in a department have unique id, name, salary, hire date, birthdate, id card and a CV and can have more than one phone number. The managers have their own office telephone number and a special email address.

**\* Assistants** have their Section that they are assigned to.

**\***Only the **owner** of the company can add new **departments** and **warehouses** and assign **managers** to them.

**\* Department Manager** employs workers and assistants for the department. And he can import **clothes** from a warehouse.

**\* Warehouse Manager** can choose or add suppliers to receive clothes from.

**\* Cashiers in departments** Sell **Clothes, each piece of clothing** has a unique ID(barcode),Whole sale price, price, color, type, brand name, size, preview if available, over all stock and a specific **supplier** who supplied it.

**\*** A **Warehouse** can provide multiple **departments** with **clothes,** and a **department** can receive **clothes** from multiple **warehouses**. We keep track of each piece of clothing provided from a warehouse to a department.

**\* Suppliers** supply **clothes** for the **Warehouses,** we keep track of each shipment that is shipped to every **warehouse.**

**\*Suppliers** have unique ID, name, multiple locations, telephone number and shipment cost.

\*The data base keeps track of all clothes in it, from the moment they are supplied to a warehouse, to when it is imported to department until it is sold.

Reports

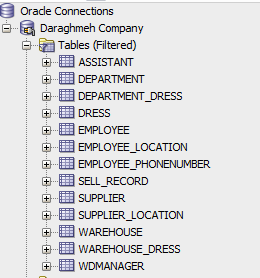
We used Jasper Reports as the main API to make reports, one report is created when a receipt is done, as it contains the information of the sold clothes, the length of this report is determined of how many clothes are sold.

Another report is created when a department is deleted, so that the information of the employees who worked in it will not be lost.

Tools used in the project

* JavaFX Libraries.
* Github.
* Draw.io.
* NetBeans apache.
* SQL developer.
* JasperReport.
* Scene Builder.
* VMware.
* SQL Plus.
* JasperSoft Studio.
* Icons8 (Pichon) for icons.

Database Schema



The schema includes **5** main tables for the Department, Warehouse, Employee, Dress and Supplier, **5** tables for multi value attributes which are : Department\_Dress, Employee\_Location, Employee\_Phonenumber, Supplier\_Location, Warehouse\_Dress,

**2** Tables for sub-enteties: Assistant and WDManager(warehouse-department managers) and **1** table that tracks every dress sold(Sell\_Record).

GUI Overview

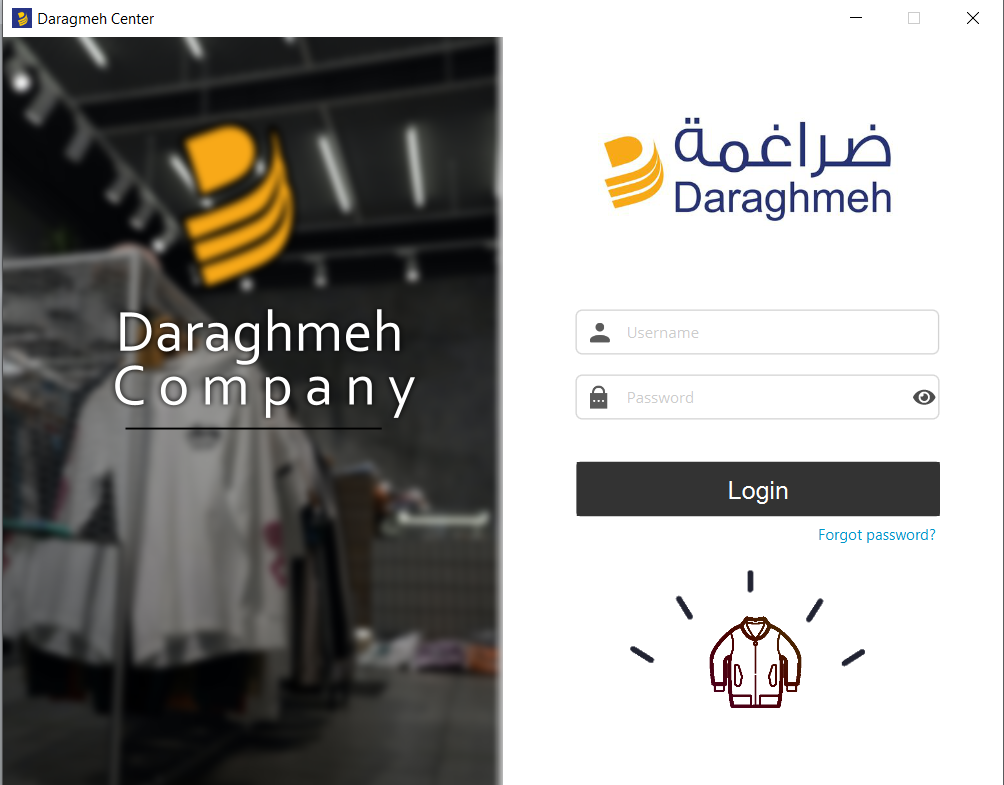
The program is based on JavaFX Library as it has better look-and feel features than Java AWT.

The program starts by showing a log in screen that interacts with the

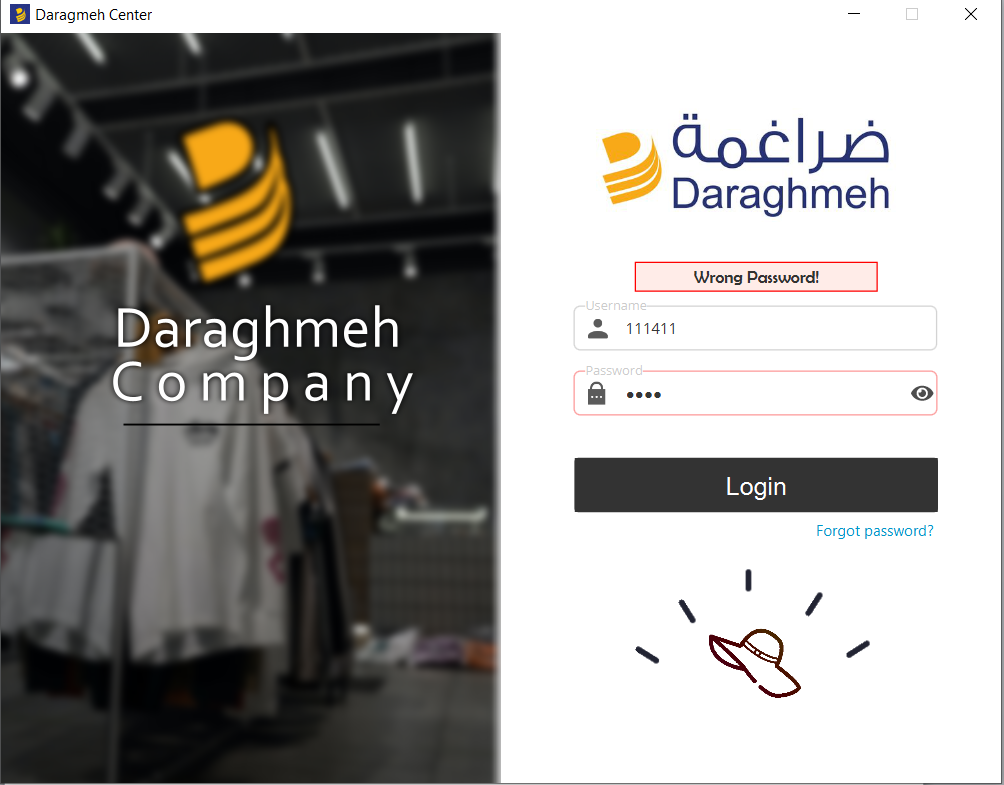
user based on the information that he enters.

Only company owner, warehouse managers, department managers and cashiers can log in into the program, as each of them have his own interface.

Users mentioned above may login by using their ID Card, except for the owner who have his static name and password that are built in the code.



A message is shown if wrong username/password information is entered.

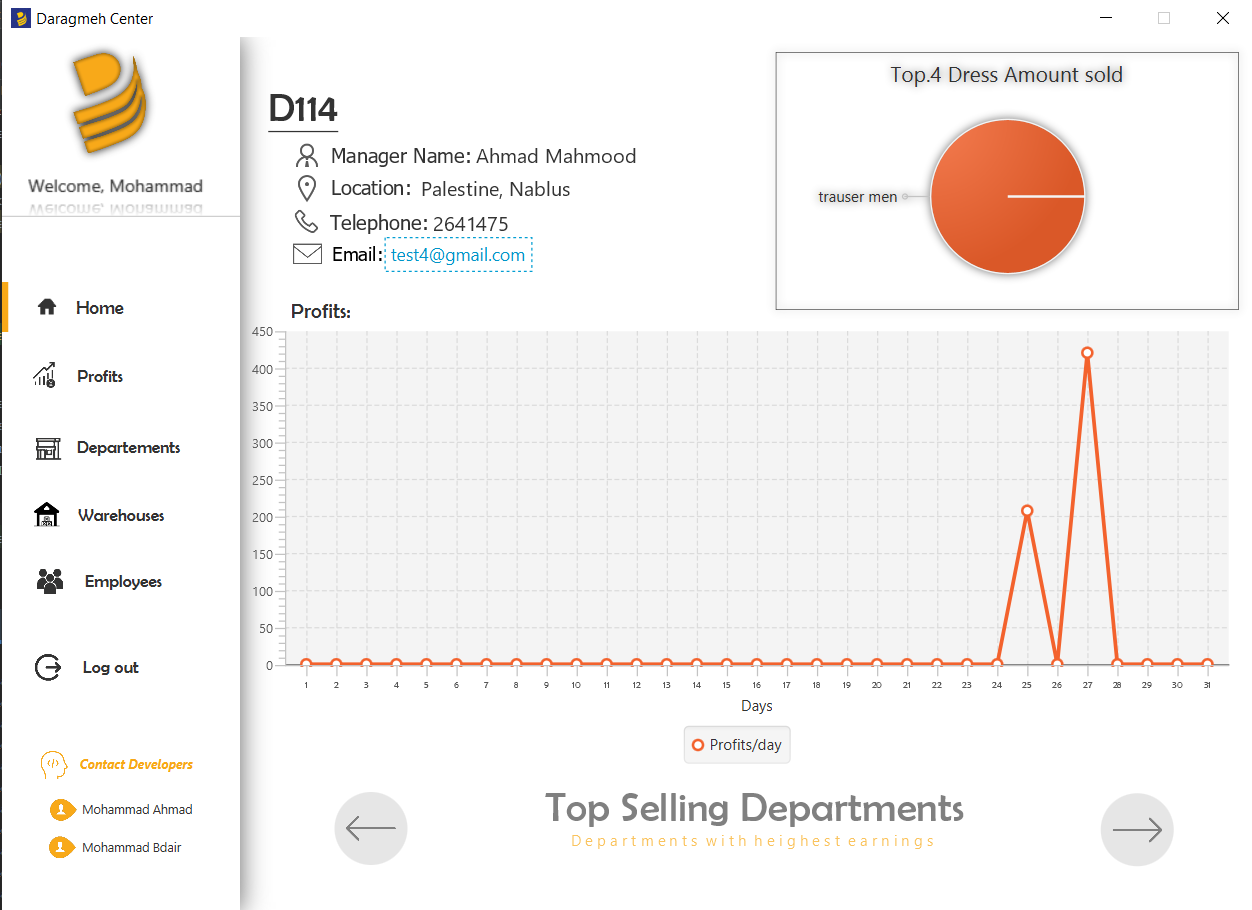


After logging in,

As the owner, Owner page is opened, it has a dashboard that consists of Homepage, Profits Page, Departments, Warehouses and Employees tabs that he can move between easily. In addition to a log out button to log him out.

**Owner homepage:**

This page shows the top 3 highest selling department over the last month. Data is displayed in an area chat.

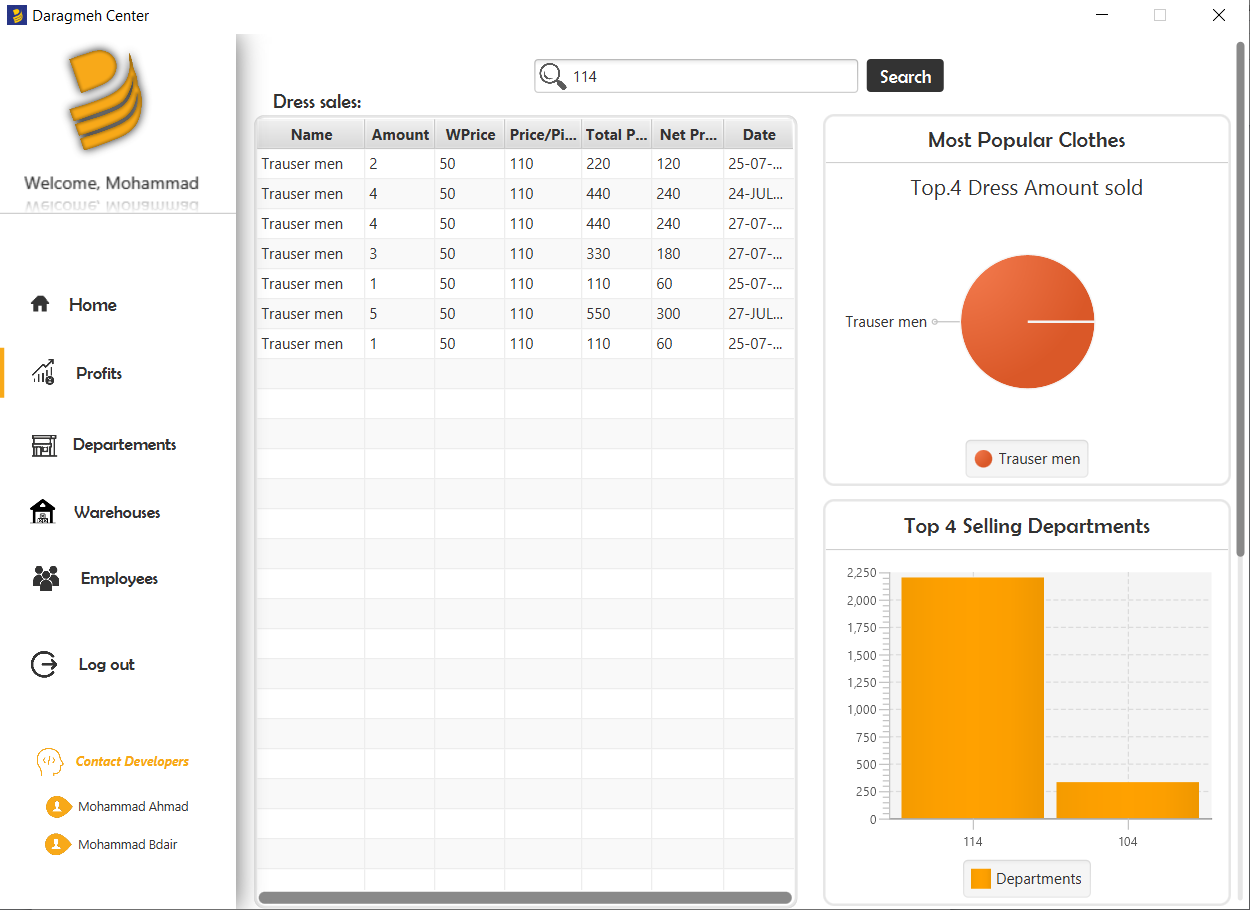


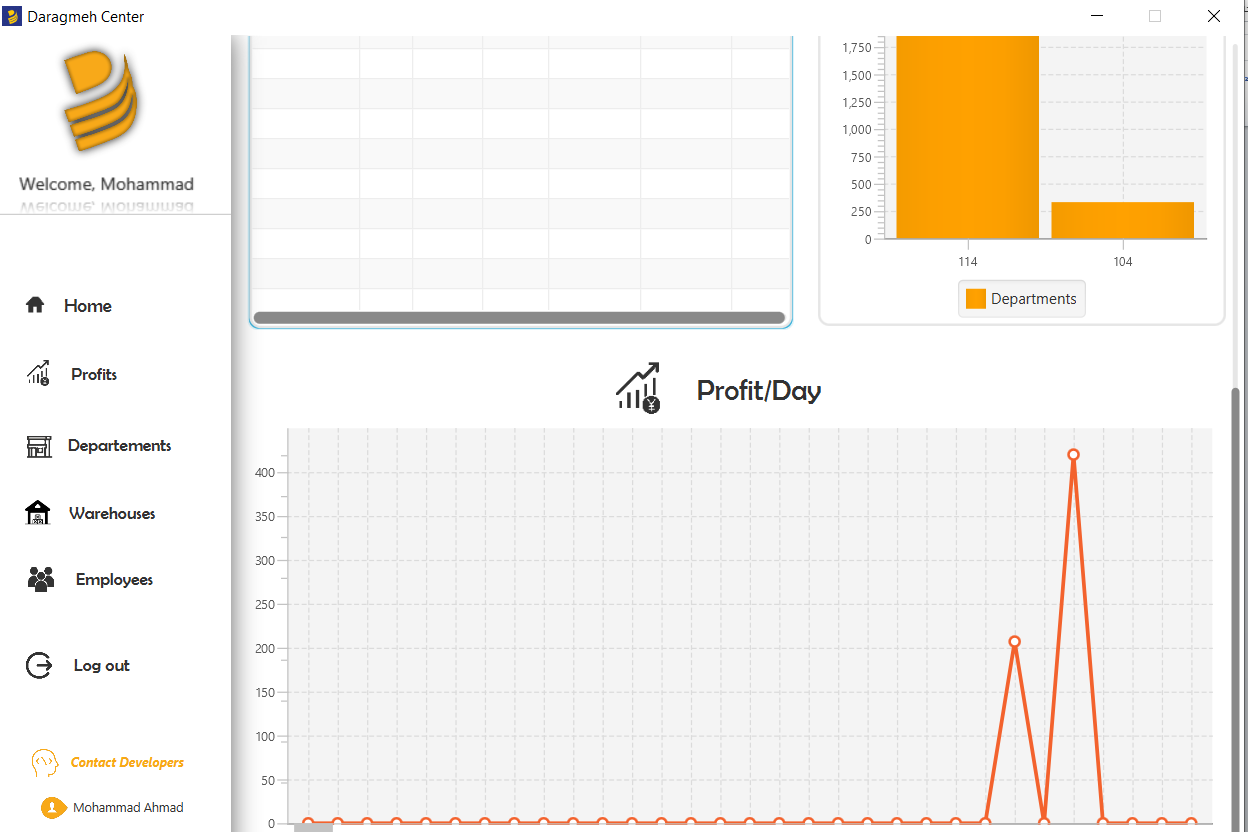
This page displays the information of the location of the highest selling departments, the manager on it and his telephone and email address.

Clicking on the email address copies the email and opens the browser on the gmail copied.

**The profits page:**

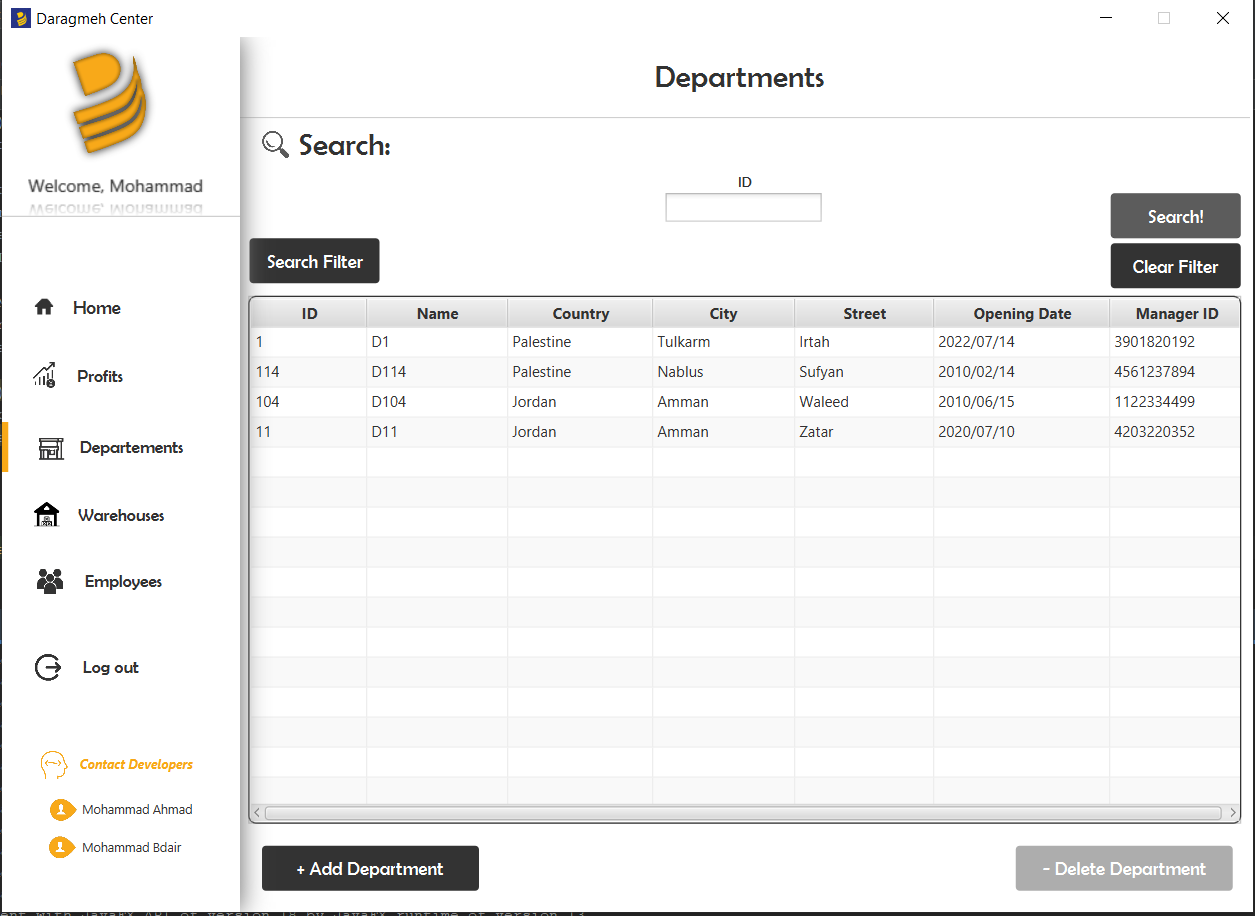
This page shows the total incomes of each dress sold when you search on a specific department



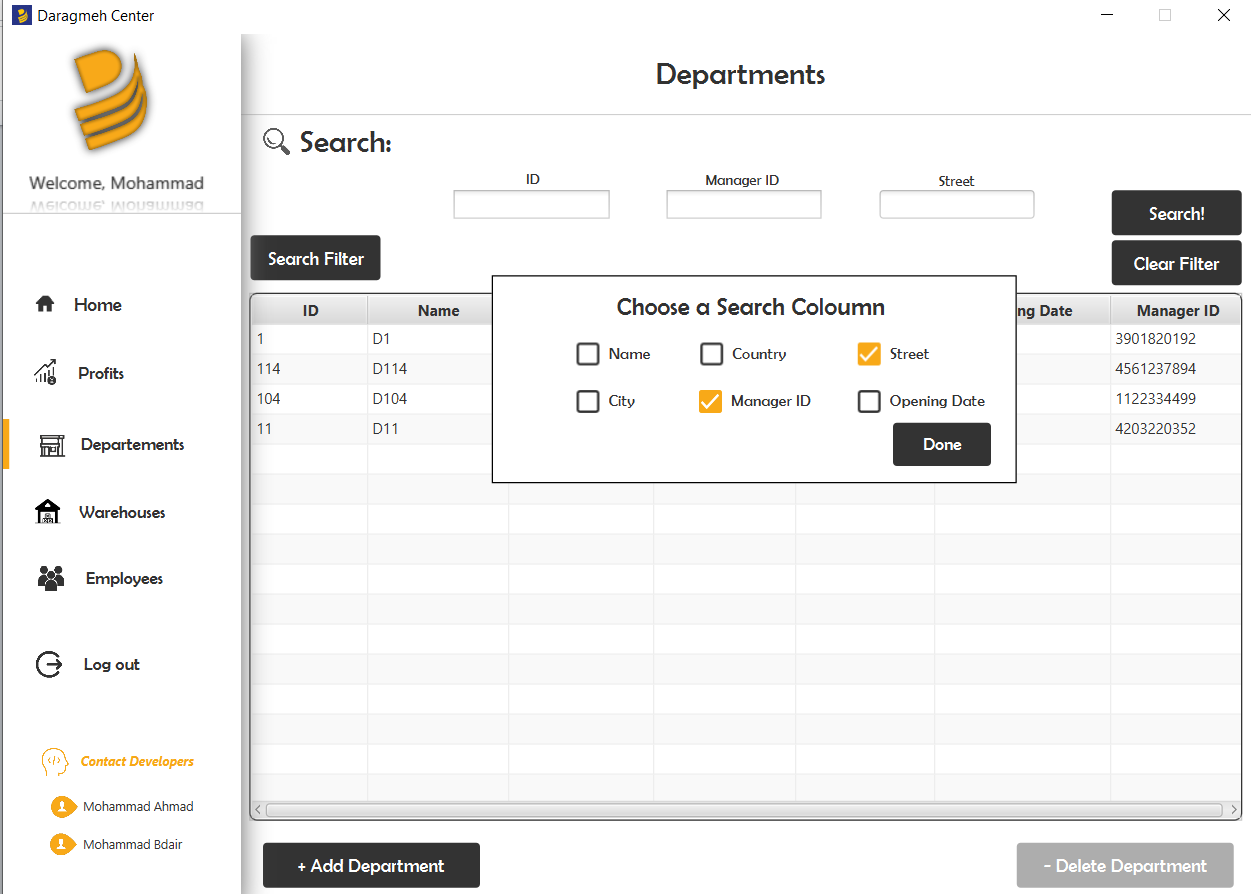


**Departments Page:**

This page has a table of available departments and information about each one:

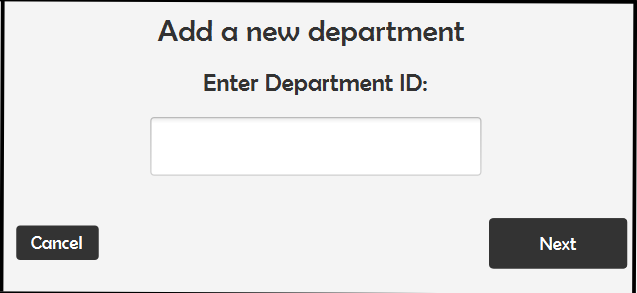


The search filter button shows a filter to select the text field that the owner wishes to search on:

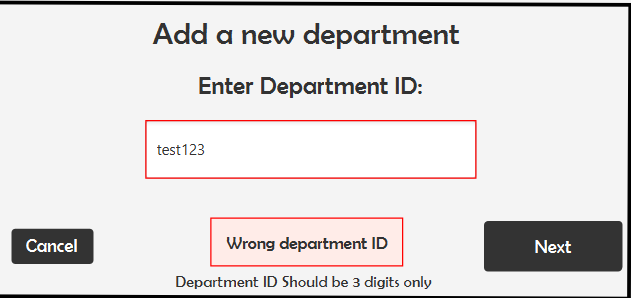


The search button searches the data base for information entered in the fields, the clear filter clears the search

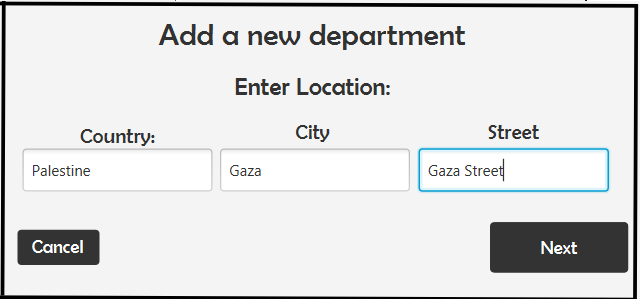
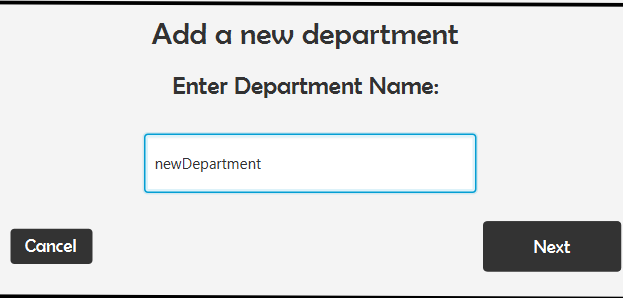
The Add Department button allows the owner to add new departments to his company:



There are few restrictions on the new department ID, as it should be 3 digits only, and must not have the same id as another existing department, it also can not be empty.



If the ID entered is correct, the Next button moves to next field, which is department name which can not be empty aswell, and then to the fields where he enters the department location:



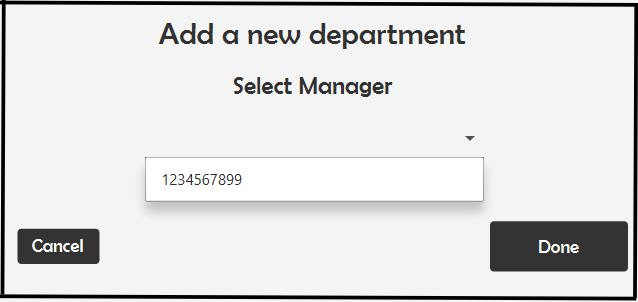
Then you select an available Manager by his SSN.

Availabe manager is detrermined by his ID Card, as IDCards start with 1,2 or 3

1 represents a cashier, 2 represents a department manager, 3 represents a warehouse manager role.

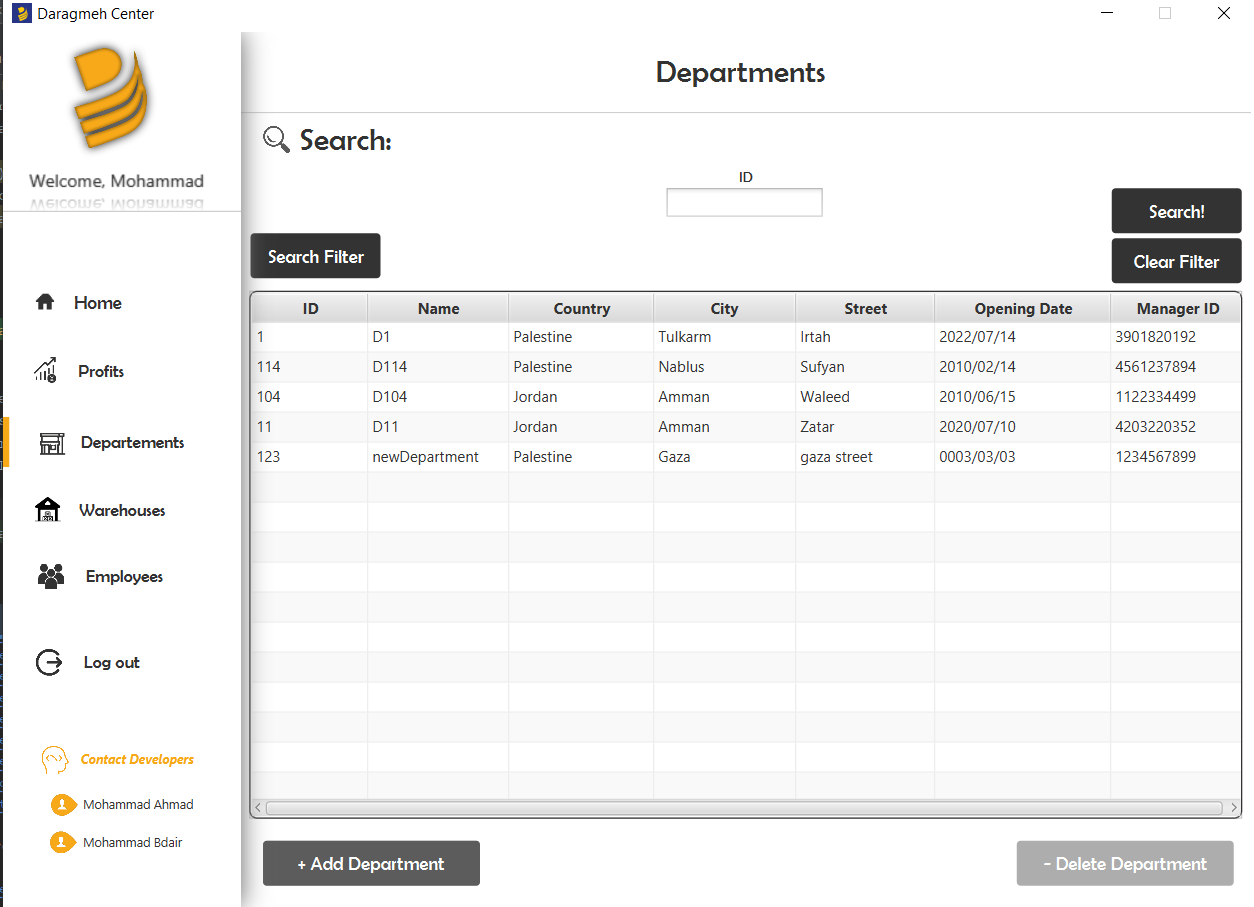
These numbers are then followed by 3 digits that represent the department/warehouse they are assigned into. Then these 3 digits are followed by 3 digits which represent each employees number.

Available managers have 000 as their department/warehouse they are assigned into , for example: 300045;



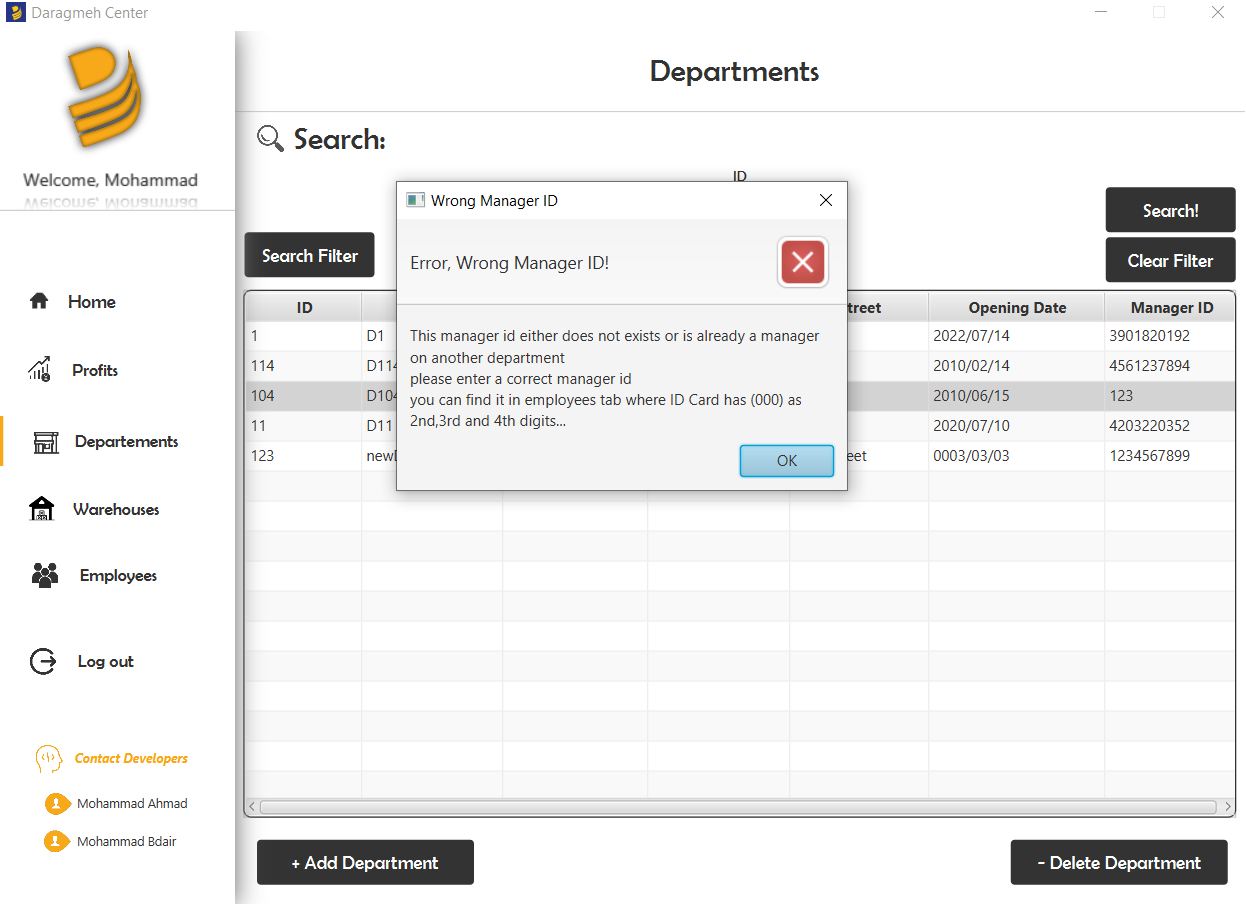
This means that only Manager with SSN 1234567899 is available to manage the new created department, if no managers are selected/avaliable then you cant create a department without a manager.

When done is clicked, a new department will be added to the table as shown:



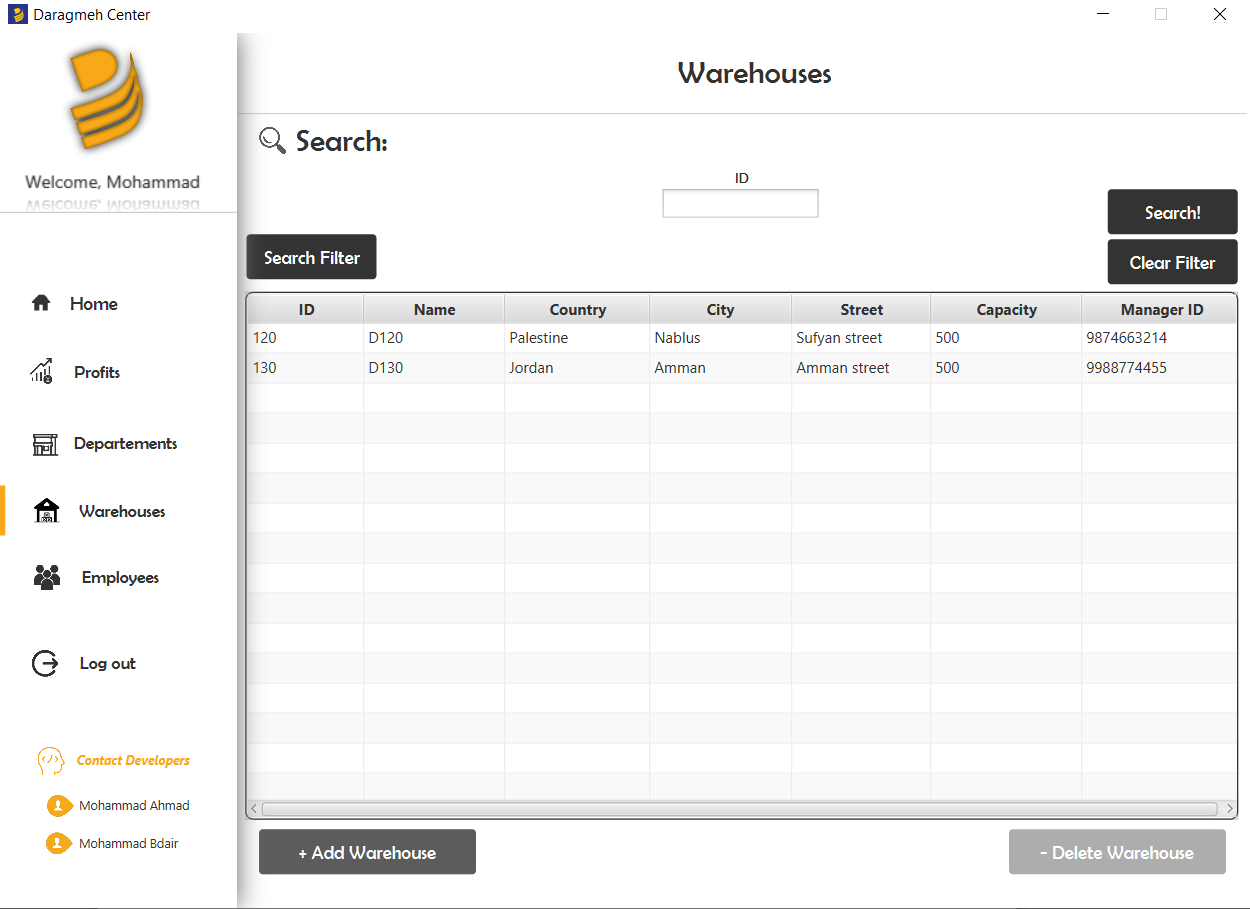
You can delete a department by clicking on it and pressing Delete, this will delete the department from the database and print a report to save the information about it so it does not get lost.

You also can edit any value (except for the ID) in the table by double clicking any cell and entering a new value, note that editing manager ID has restricitions, so that you cant add a manager that does not exist or that manages another department.

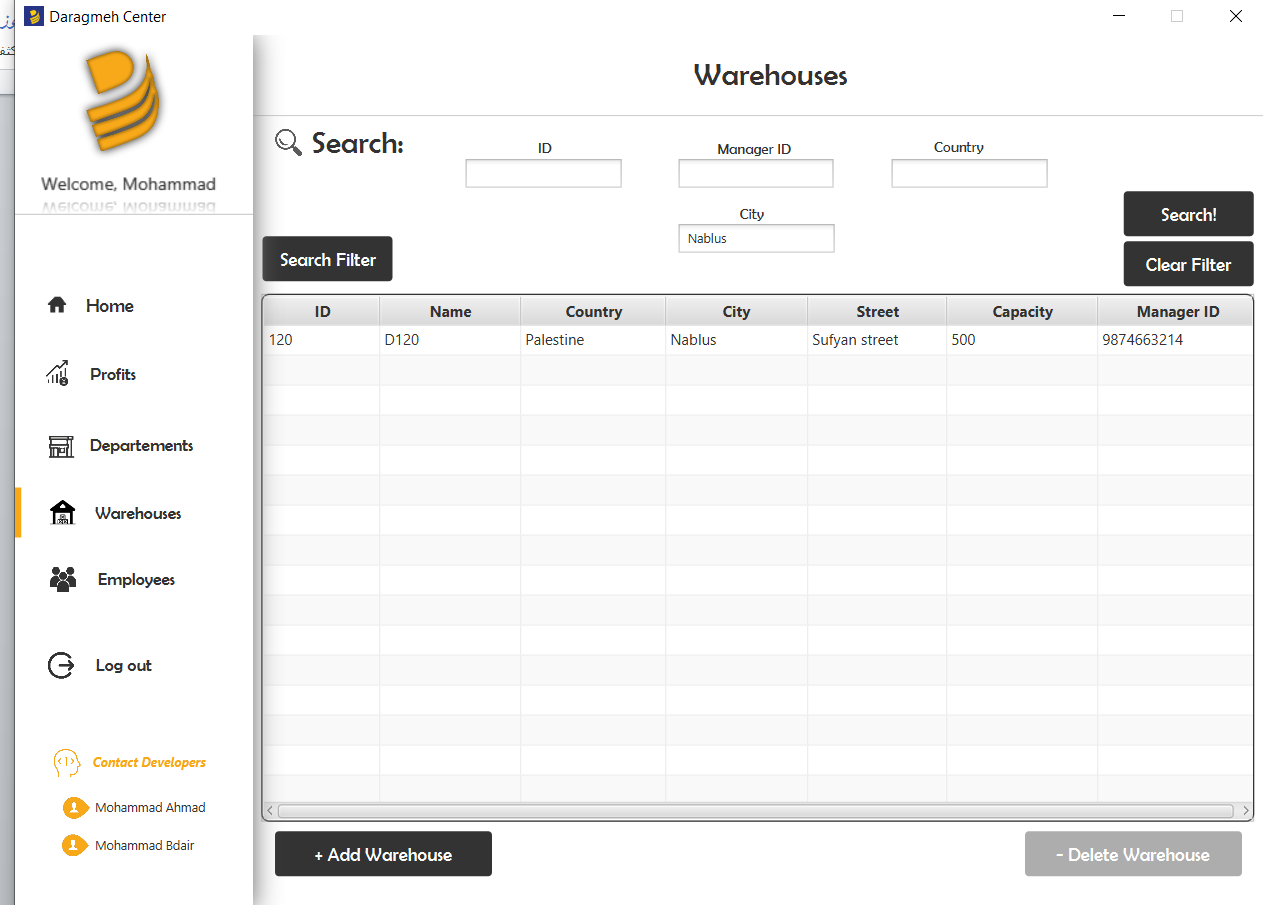


**Warehouses Page**

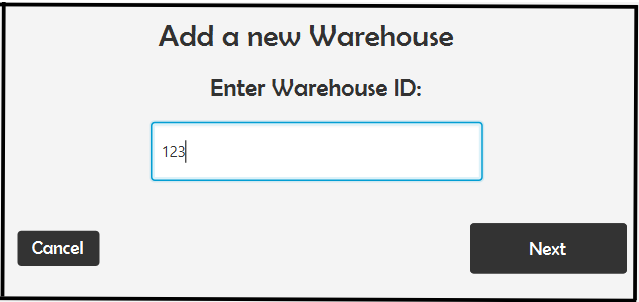
Similar to the Departments page, here the owner can view/search/add/delete all available warehouses.

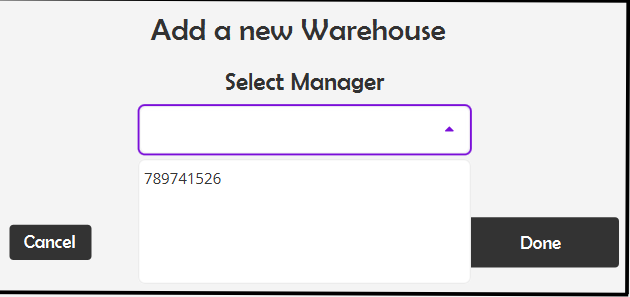
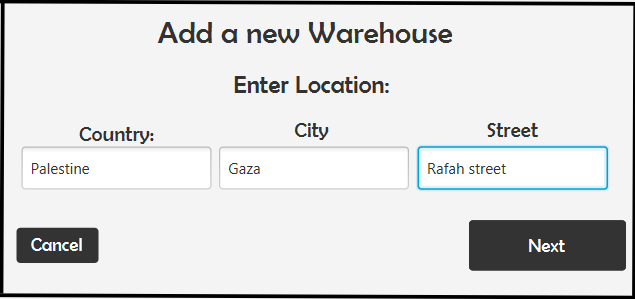


After applying a search filter:

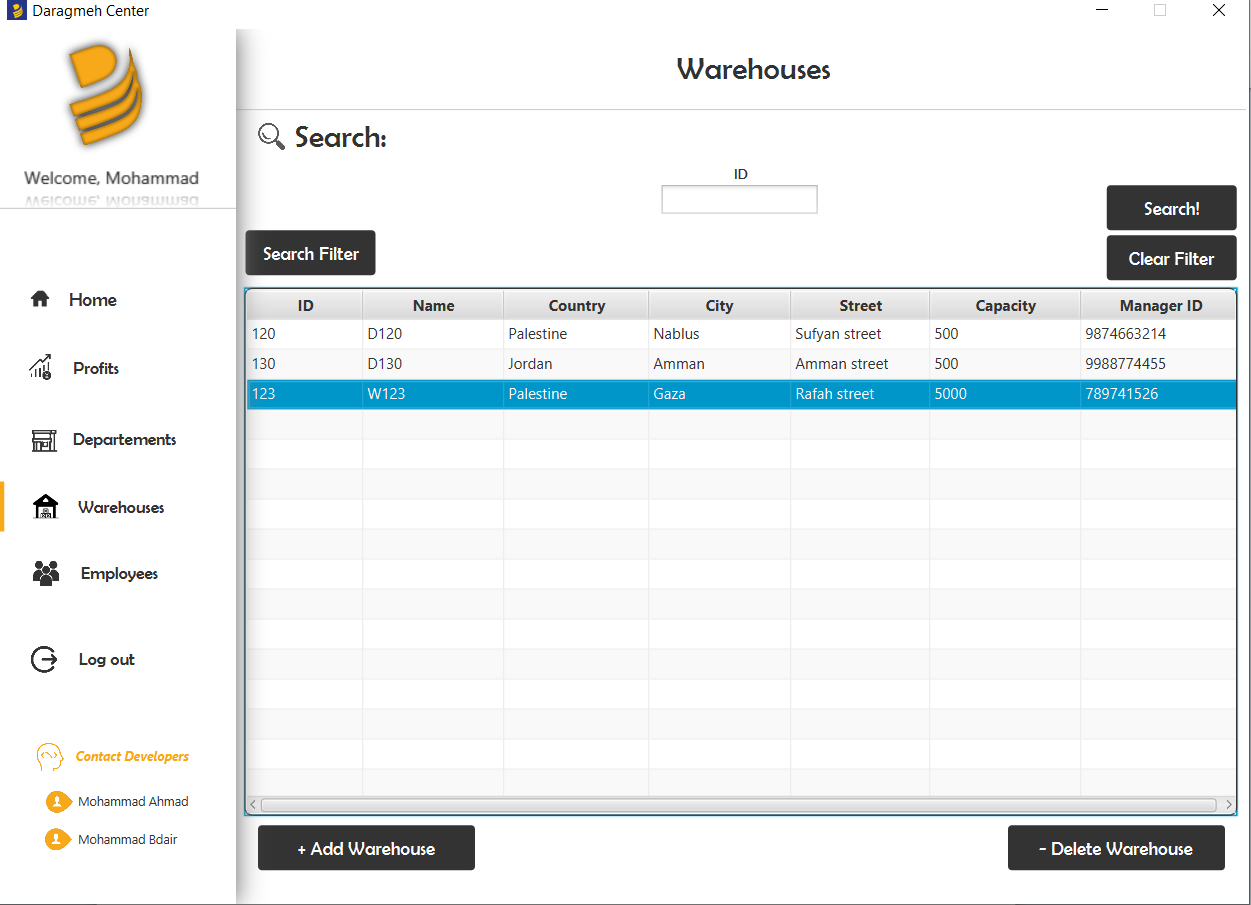


Adding a warehouse:





Selecting a manager follows the same rules as selecting a department manager, the main difference here is that warehouse manager IDCard starts with 3, but department manager ID Card starts with 2.

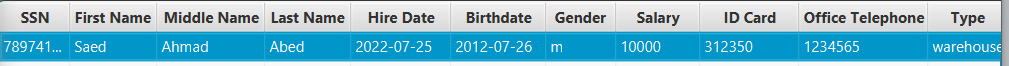


Selecting then deleting a warehouse also resets its owner IDCard to 000 where the 000 indecates the warehouse that he managed.

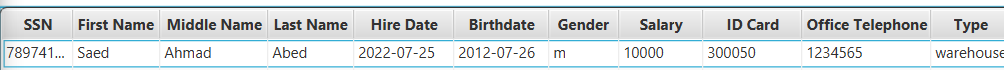
Before adding a warehouse:



After adding a warehouse and selecting this user as the manager:



After deleting the warehouse:

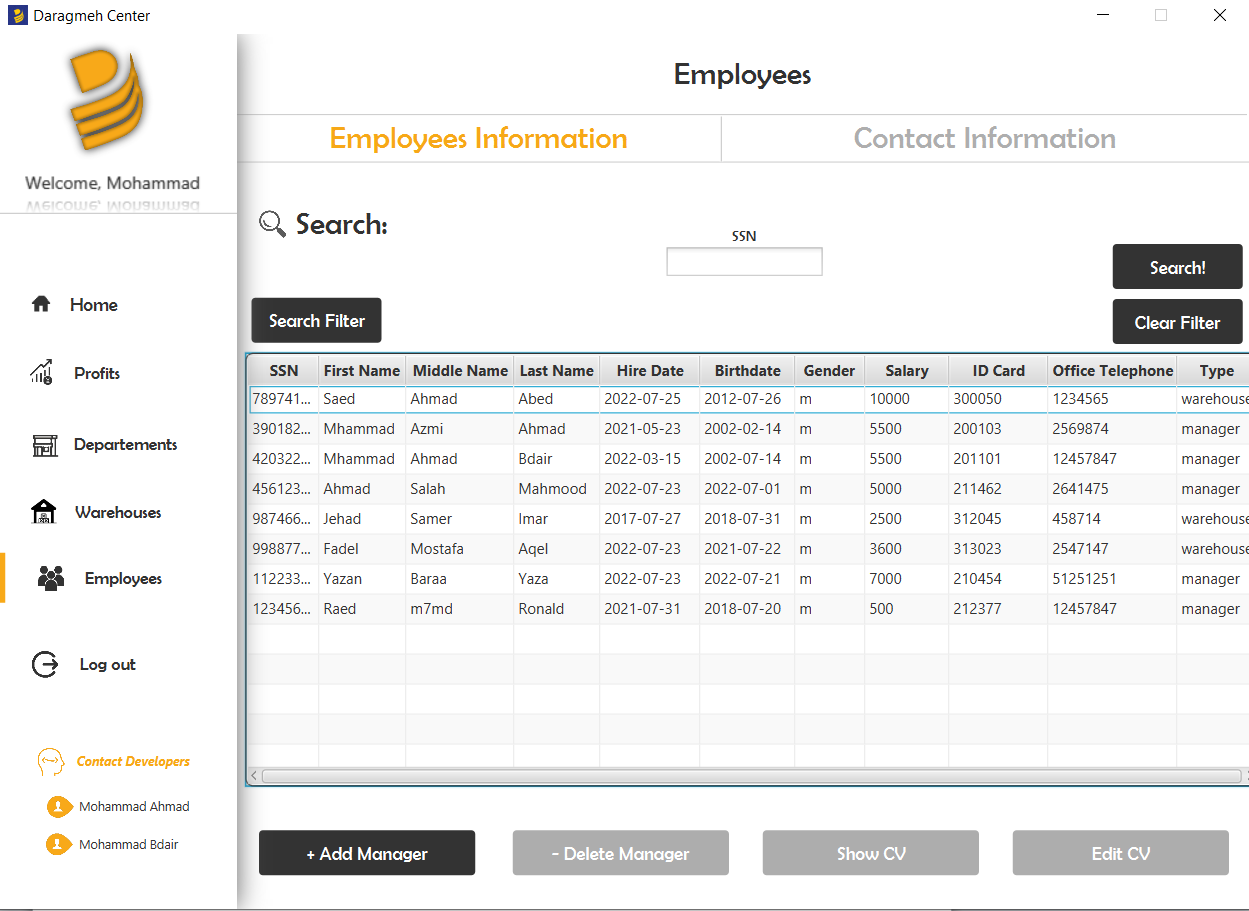


Notice how the IDCard changes.

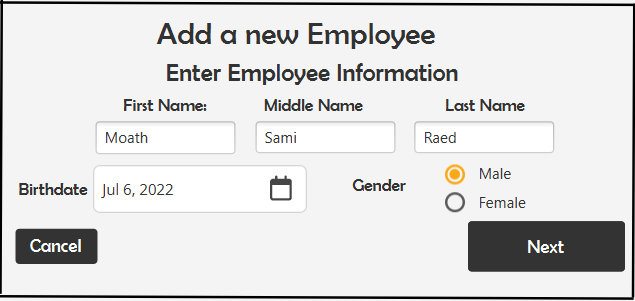
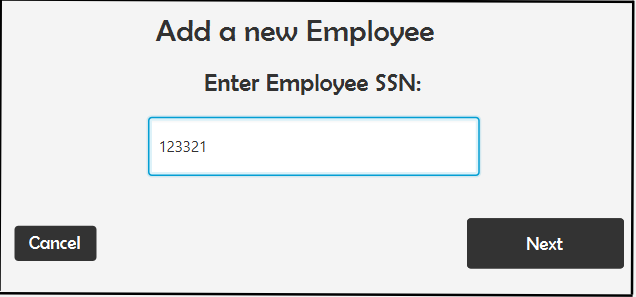
**Employees Page:**

In this page, the owner can view all of the informations about the managers only(Managers that manage a warehouse/department or are not assigned to one yet).

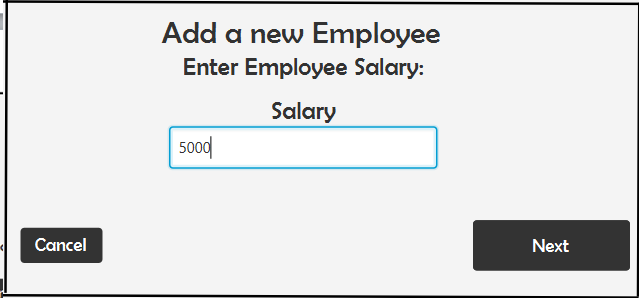
Owner can also move to another tab pane which shows managers and their phone numbers and their locations as the managers may have more than 1 phonenumber/location.



In this pane, he can search using the filter, Add a manager as follows:

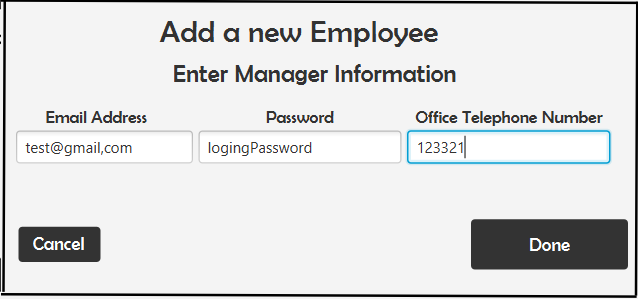


Note that SSN can not already exist.

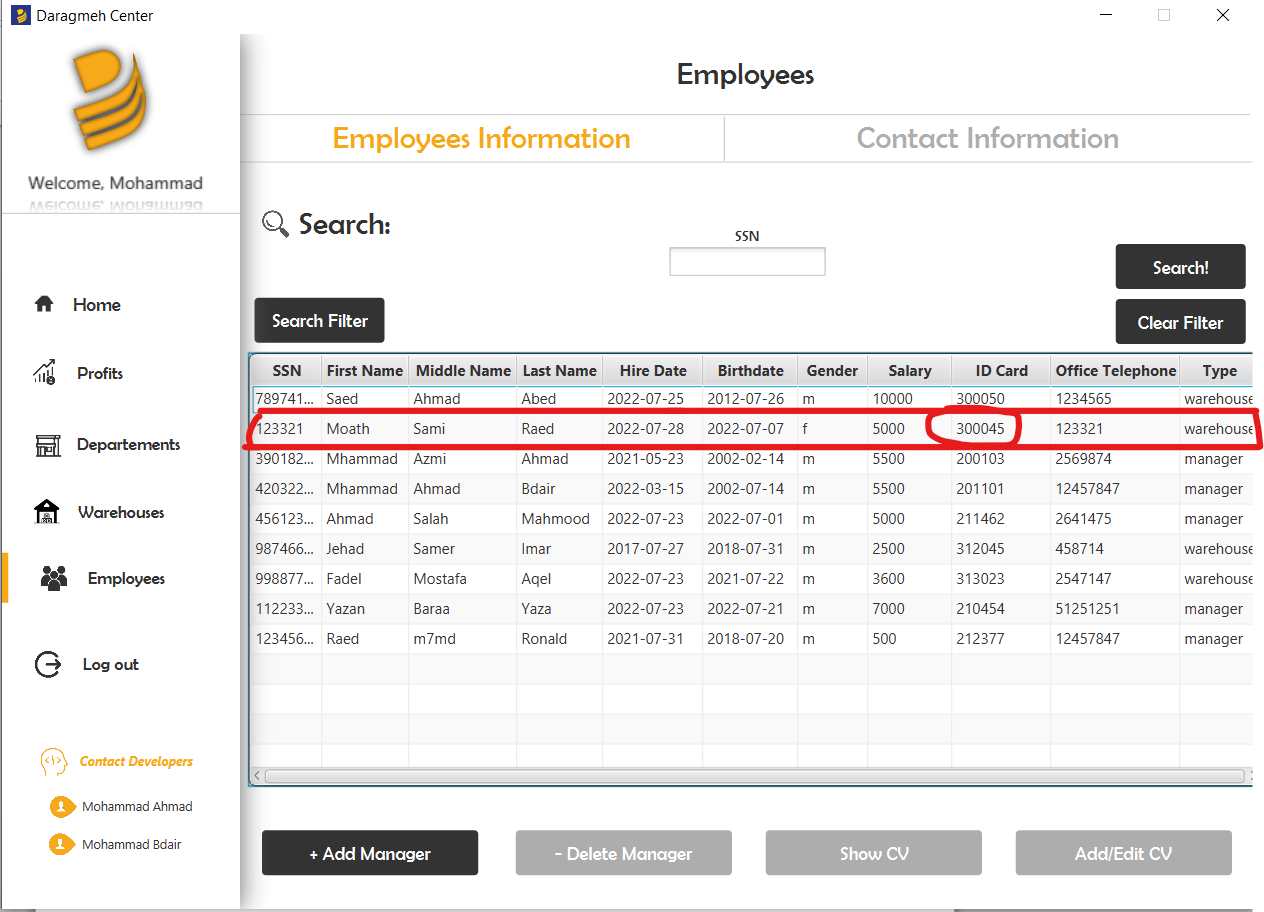


Above, when entering an ID Card, the owner specifies the type of the manager added by the first digit(3 in the screenshot which determines that this manager can only manage warehouses)

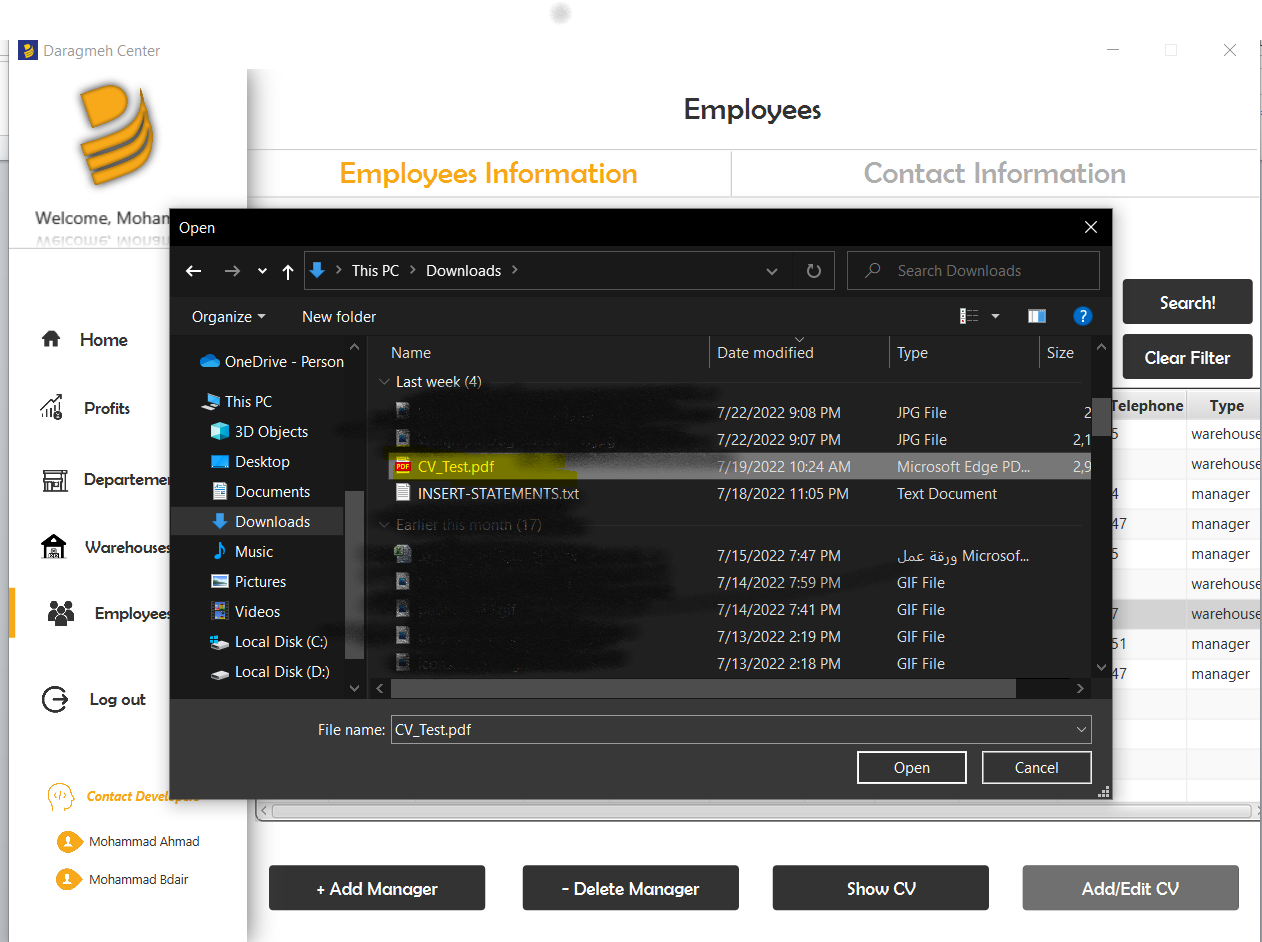
The 123 part will be replaced by 000 at all cases because this managed is not assigned to any warehouse yet, the 45 remains as it represents the special number of this manager.



When added:

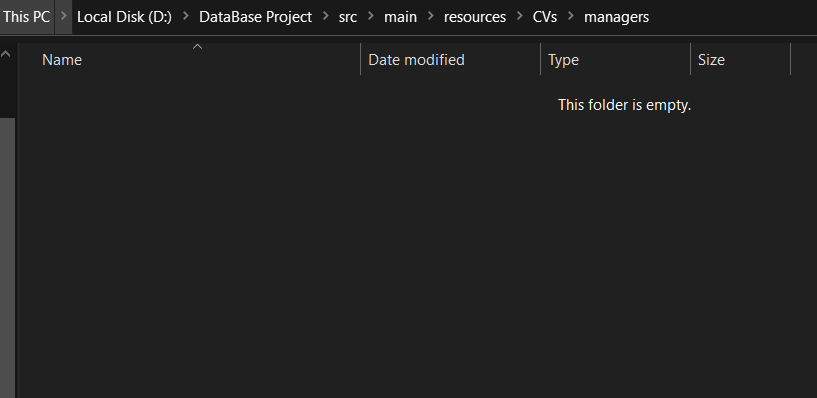


Clicking on Add/Edit Cv opens a window to rowse a PDF File which contains the CV that the manager already submitted.

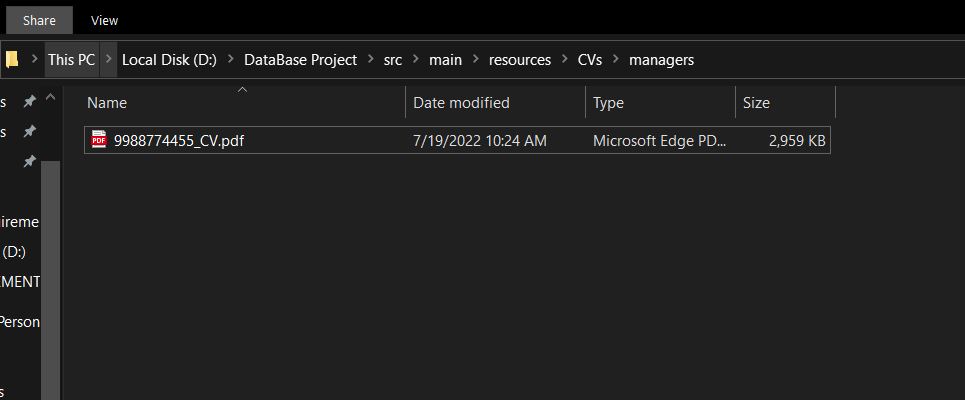


After selecting the pdf, a copy of this pdf will be created and added into the project files, it will also be renamed after the manager’s SSN.

Before Adding:

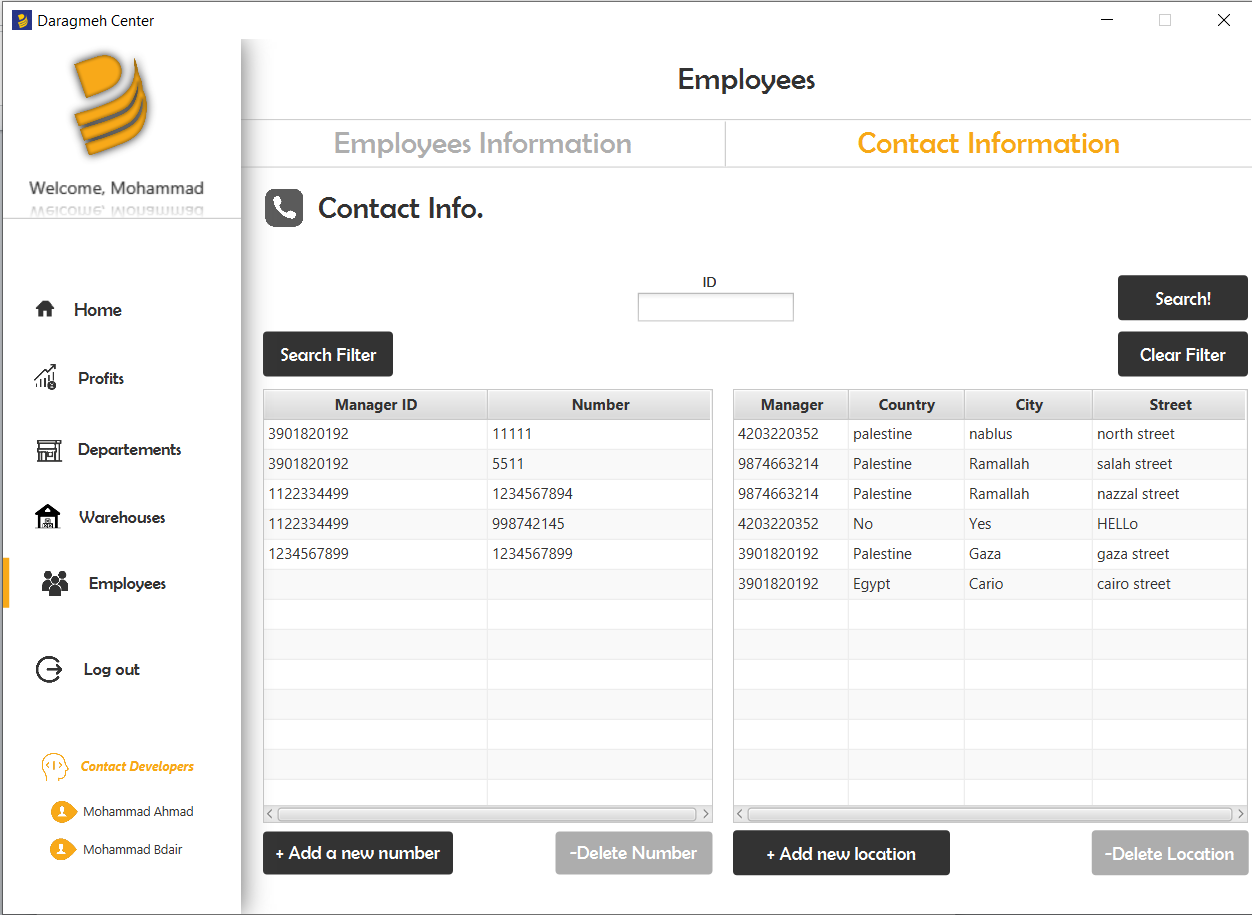


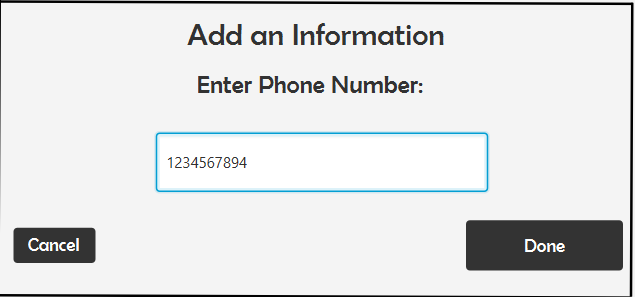
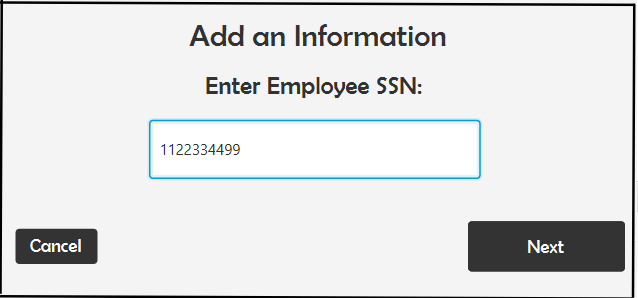
After adding:



Show cv button opens this pdf file if it exists ( note that you should first add a cv then open it)

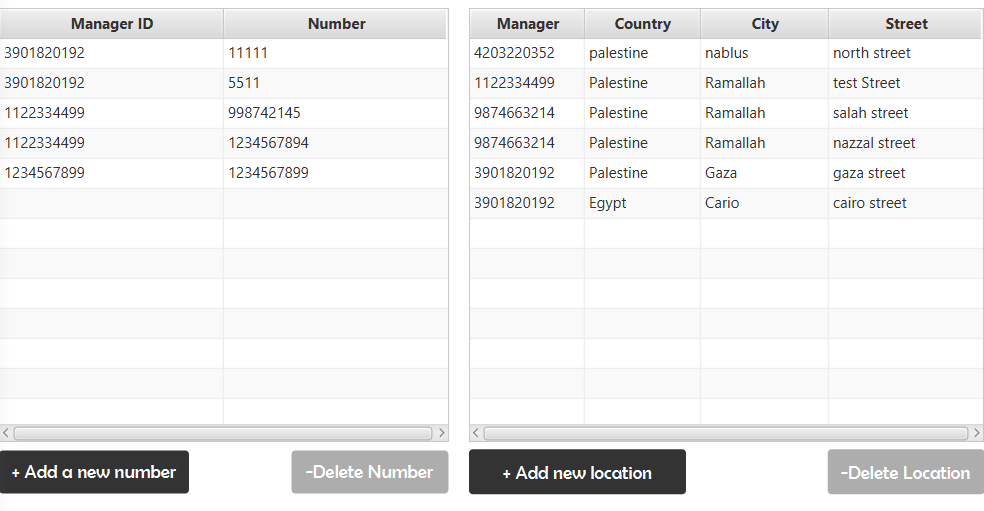
In contact information Pane, you can add new location by entering an SSN that already belongs to an existing manager, then enter the new location/phone number.





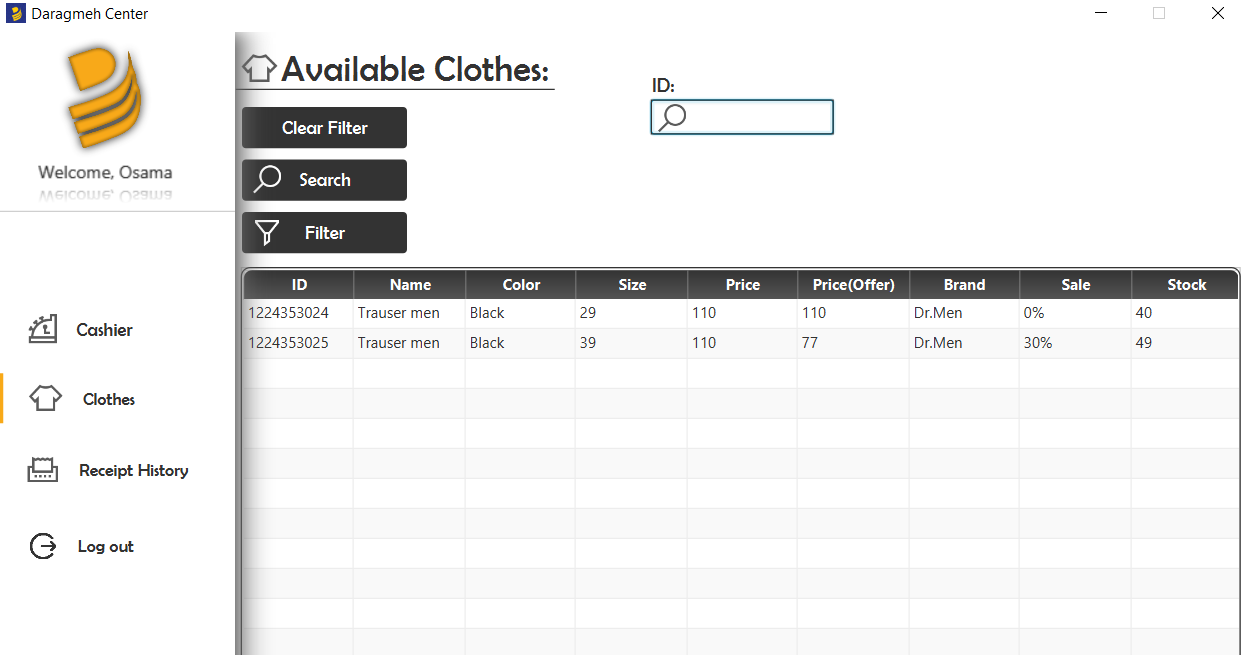
Note that phone number entered must contain 10 digits





**Logging in as a cashier**

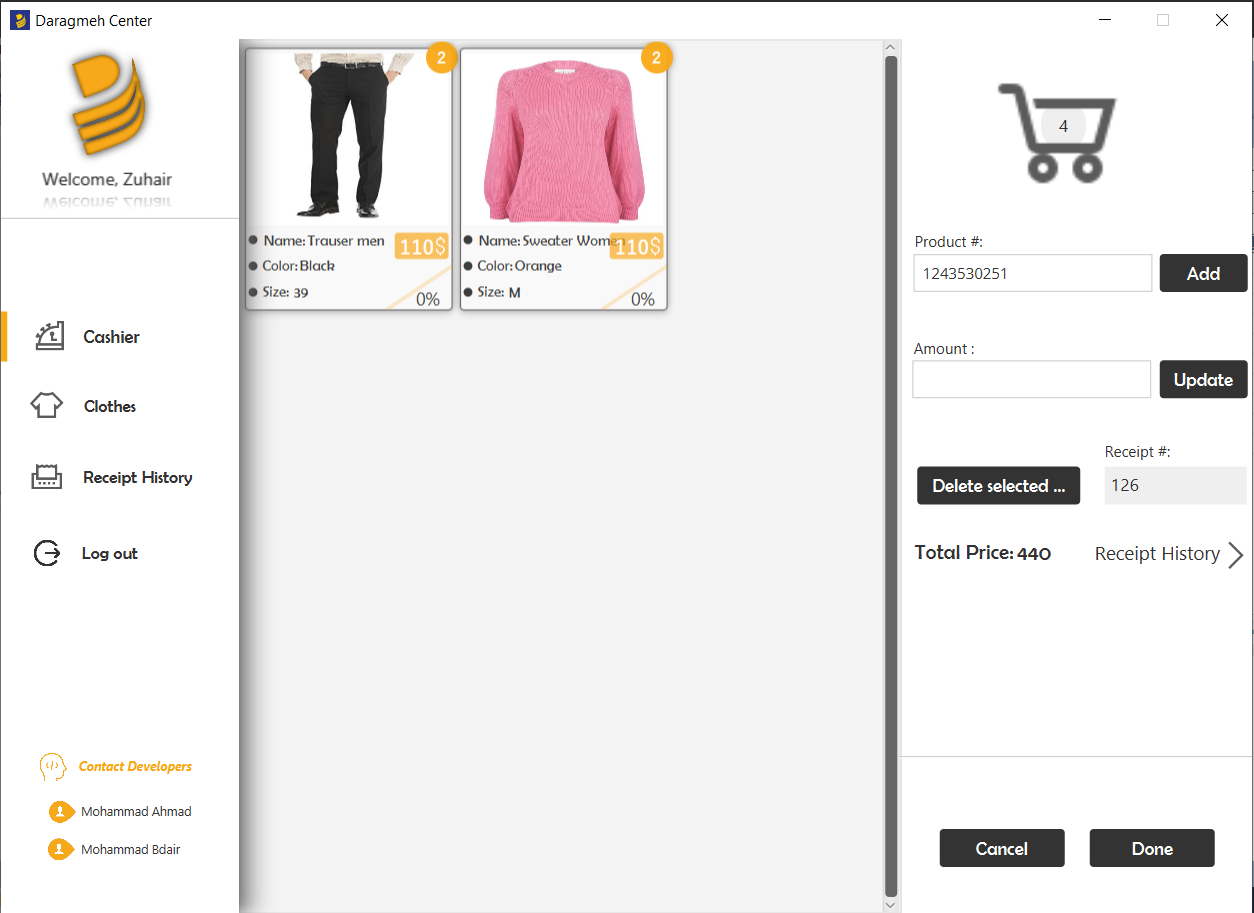
Cashier has a clothes panel which shows all available clothes and their stock in the department that he works at, he can search on any field he wants.



In the main Cashier page, a cashier enters the IDs of the clothes sold per 1 customer, for example a customer boguht 2 clothes, the cashier enter the 2 ids of these dresses and their ammounts and then presses Done to apply the reciept for this customer.

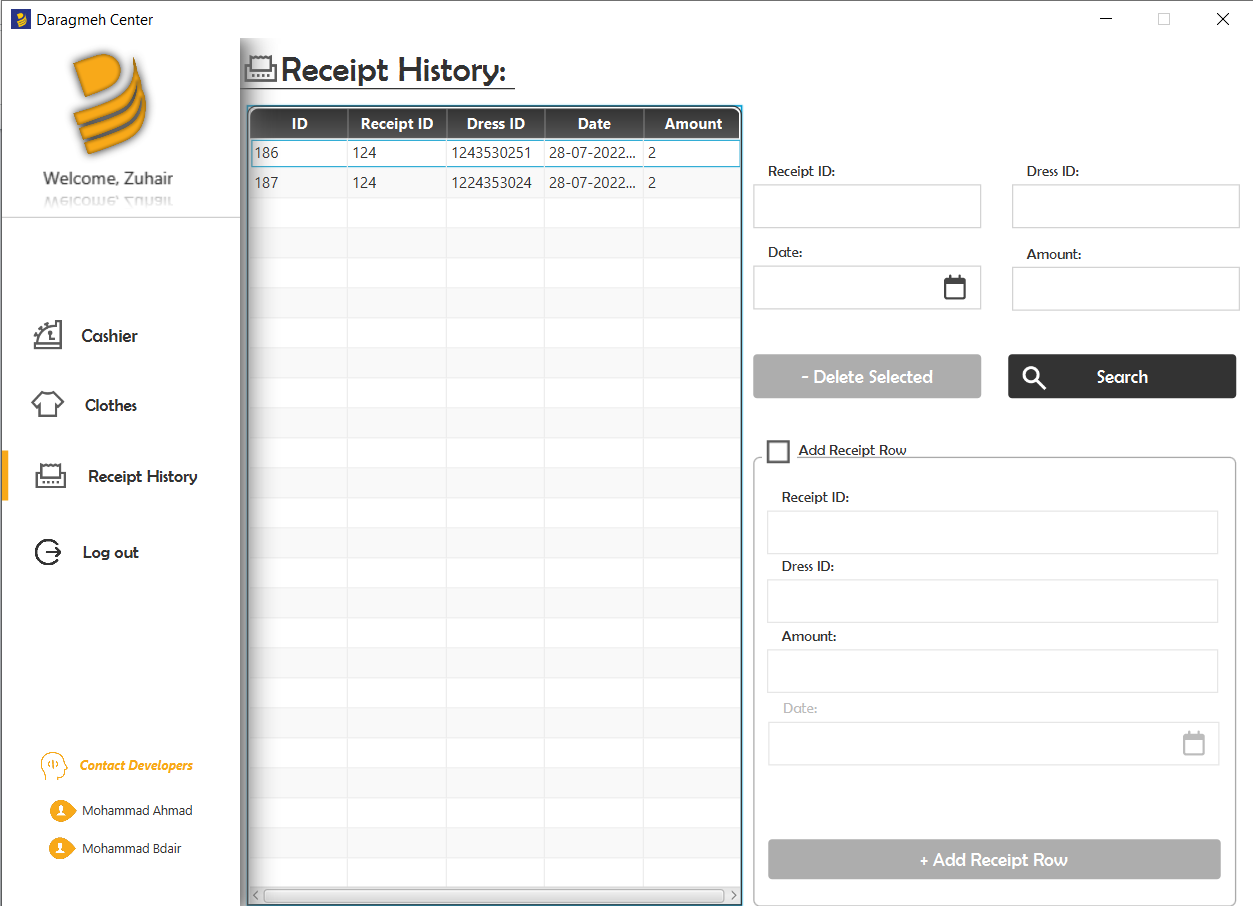
Cashier can control the amount of each piece and delete whatever he wants by selecting the card and then doing the task.

Receipt number and prices are automatically generated.

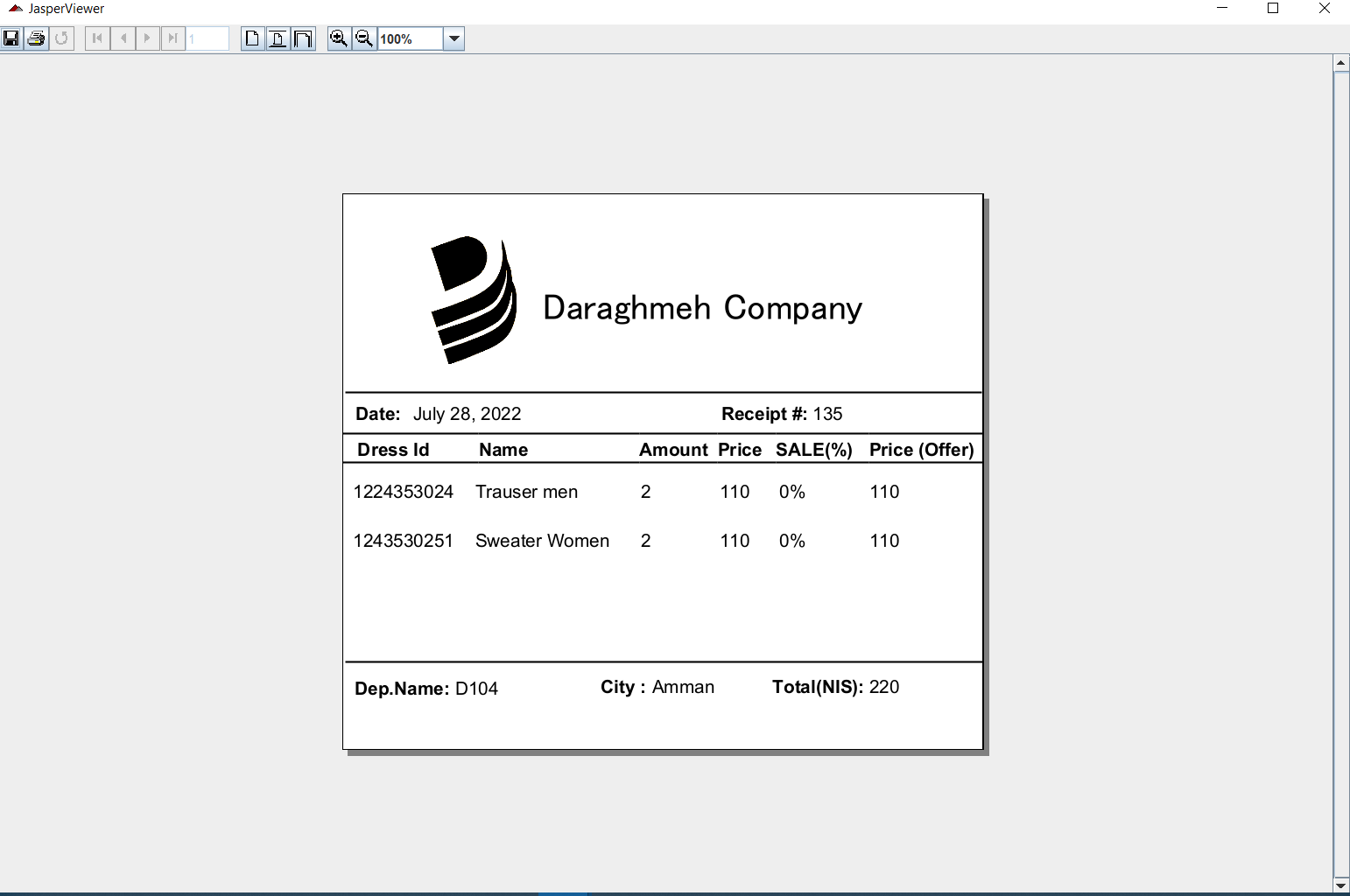


After adding, a new receipt will be added to the receipt history table

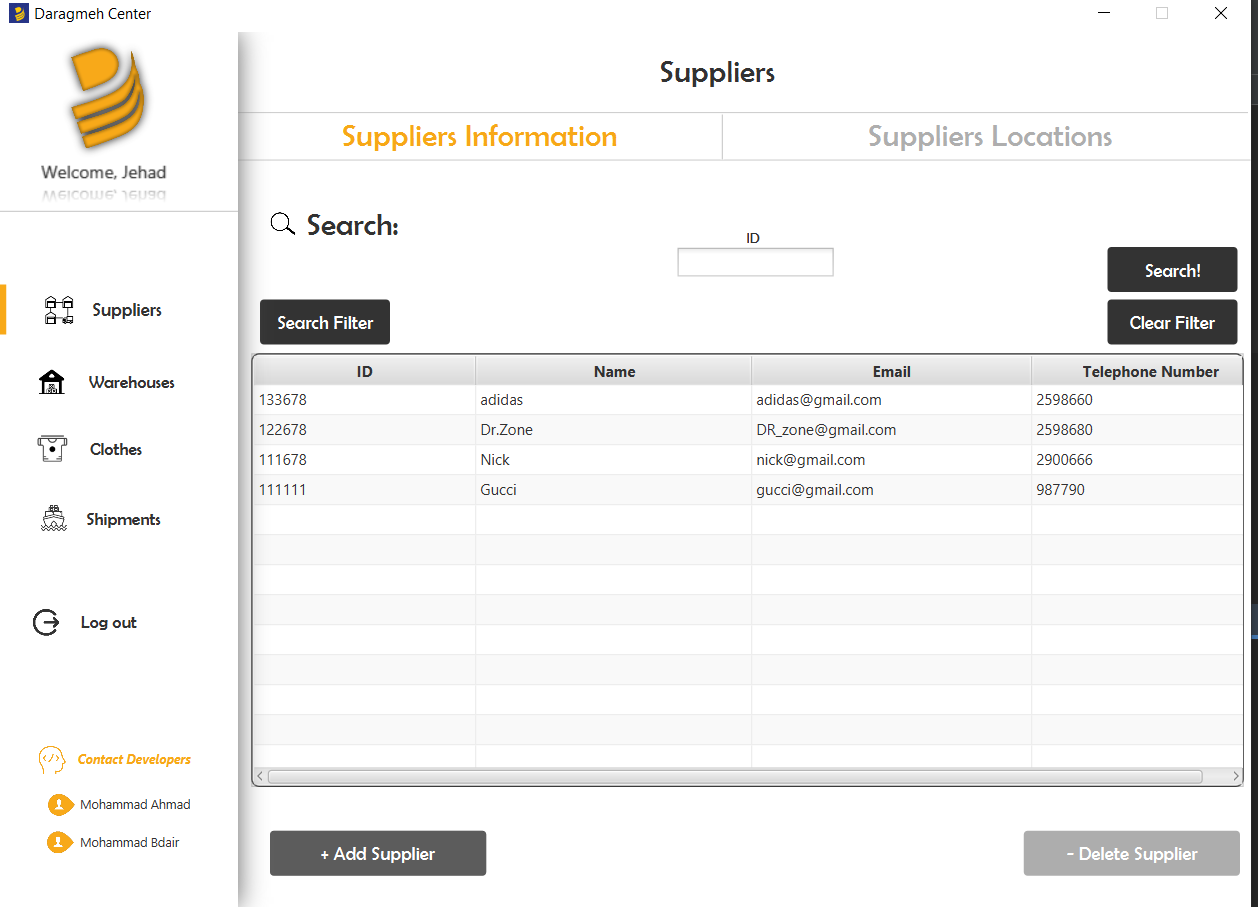
You can delete a selected receipt, or add a receipt manually.



When the receipt is done, the following report is created:

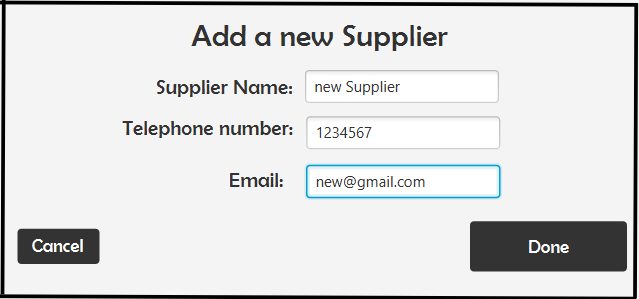
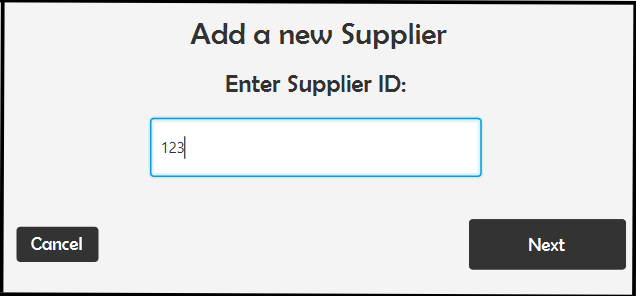


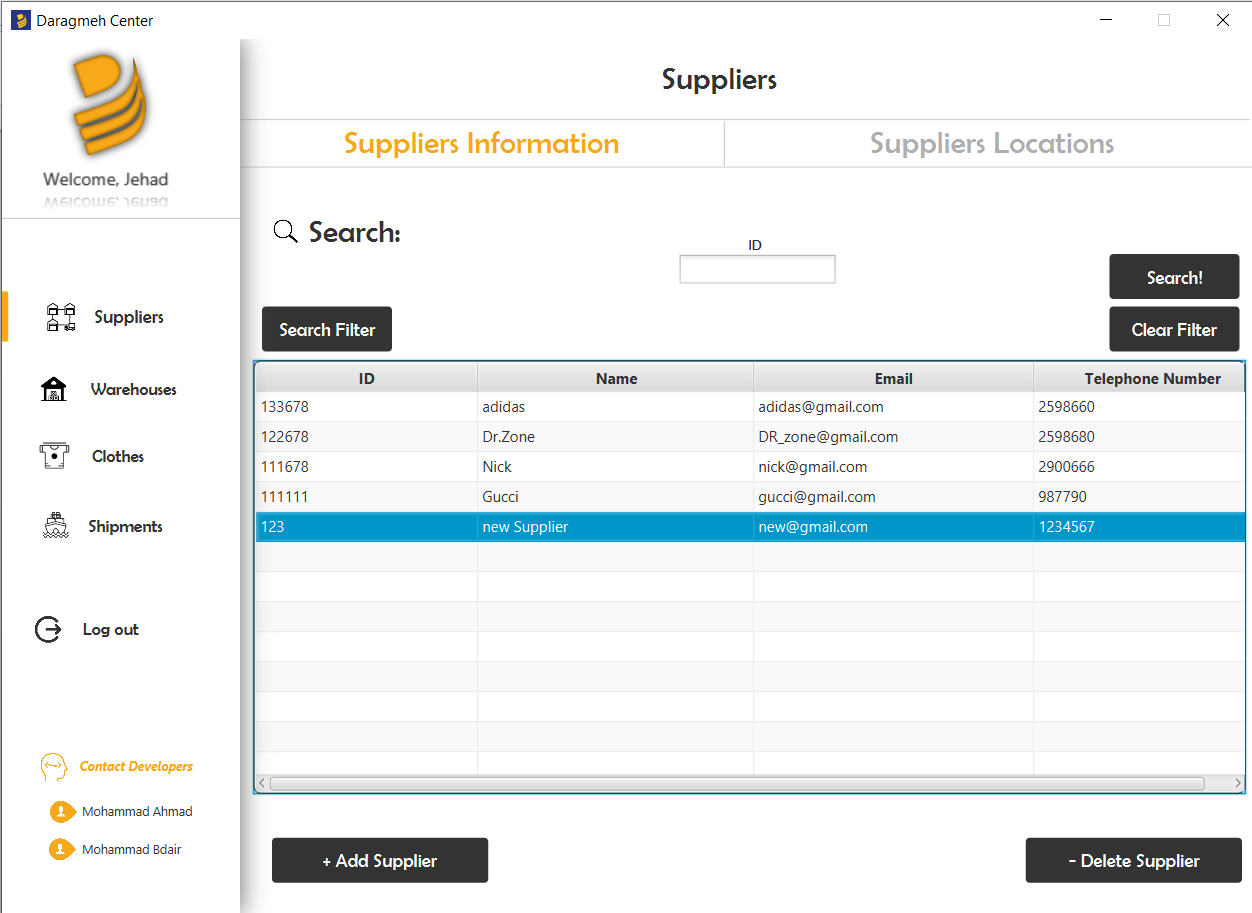
**Logging in as a Warehouse Manager**



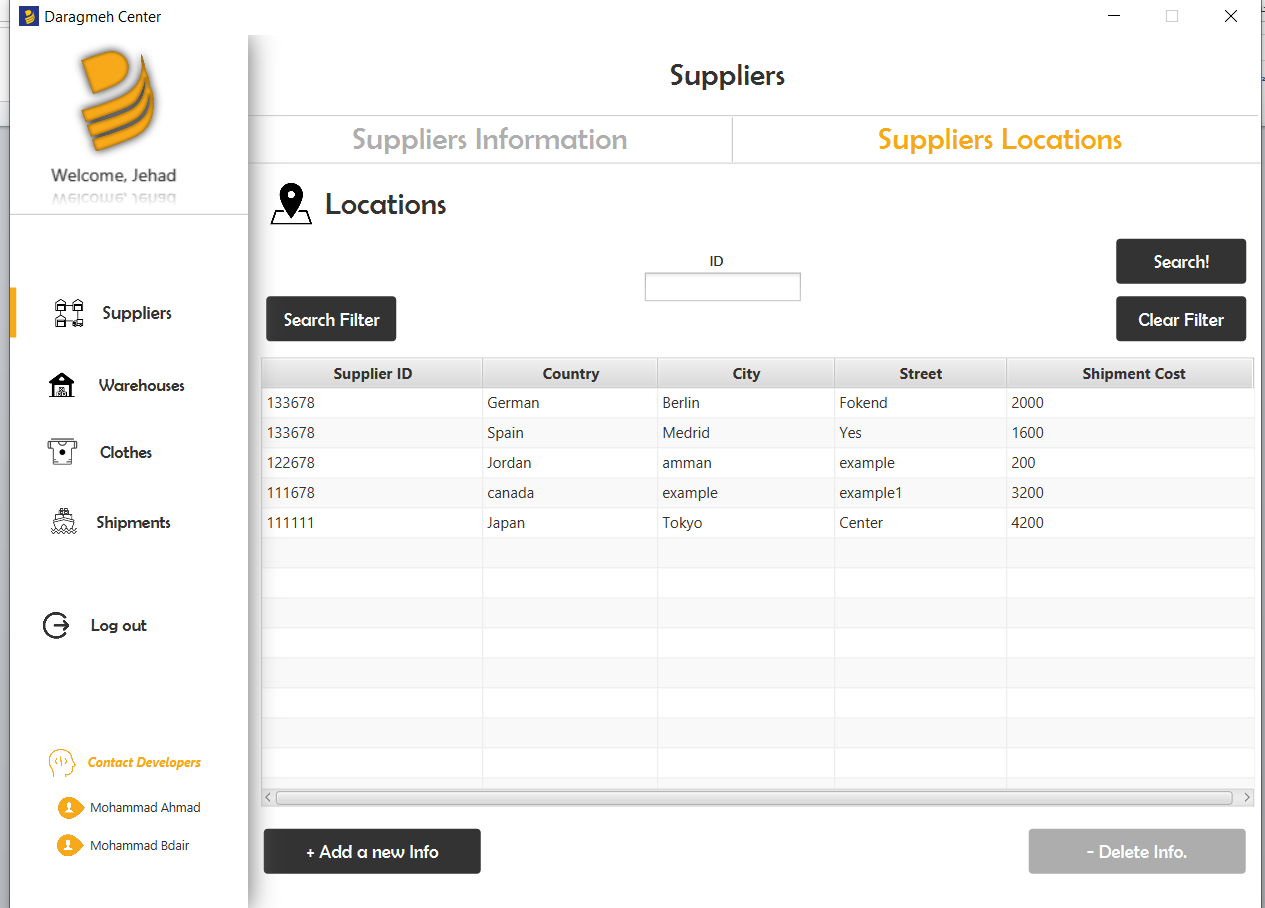
This page shows all of the suppliers for the warehouse manager so he can contact them and ask for whatever clothes he wants.

Adding a supplier:

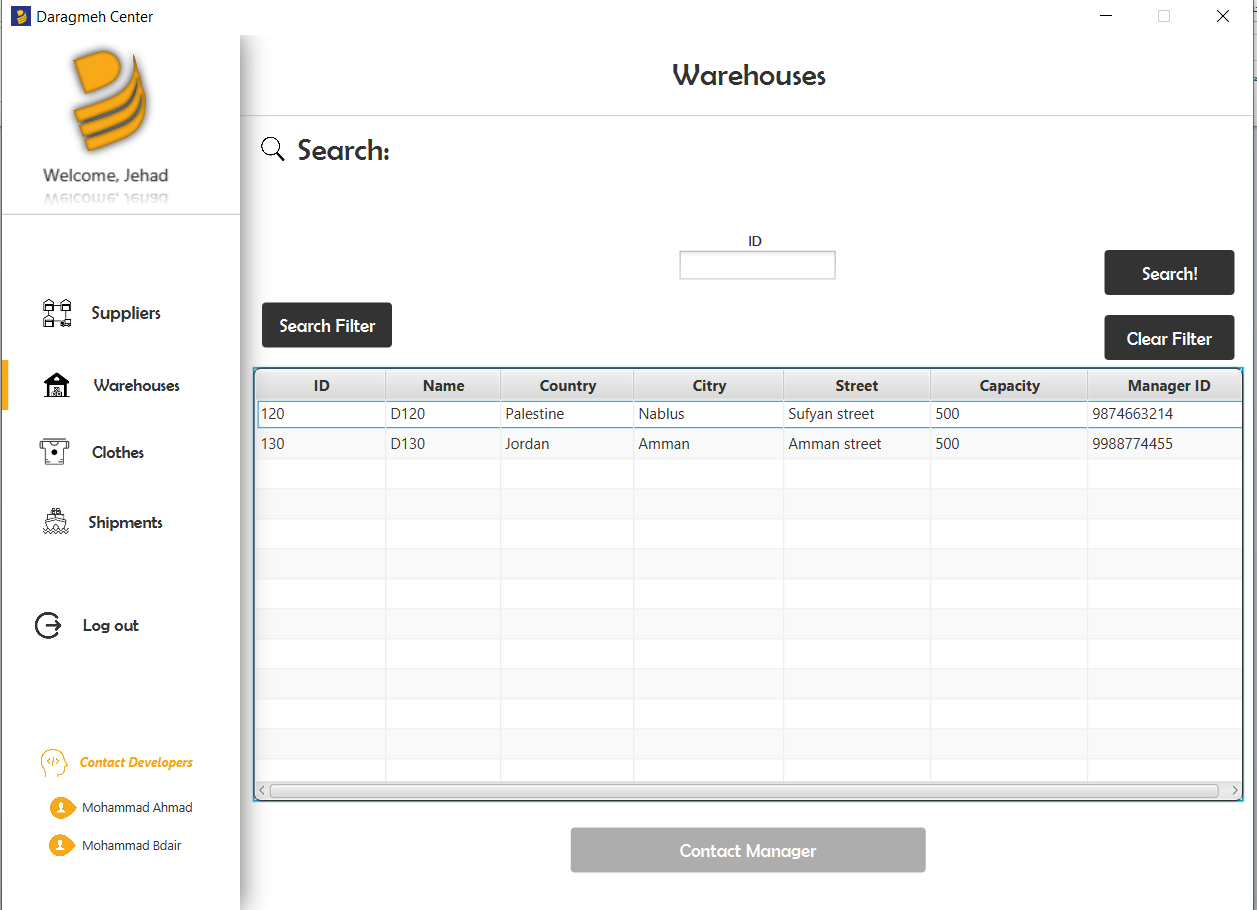




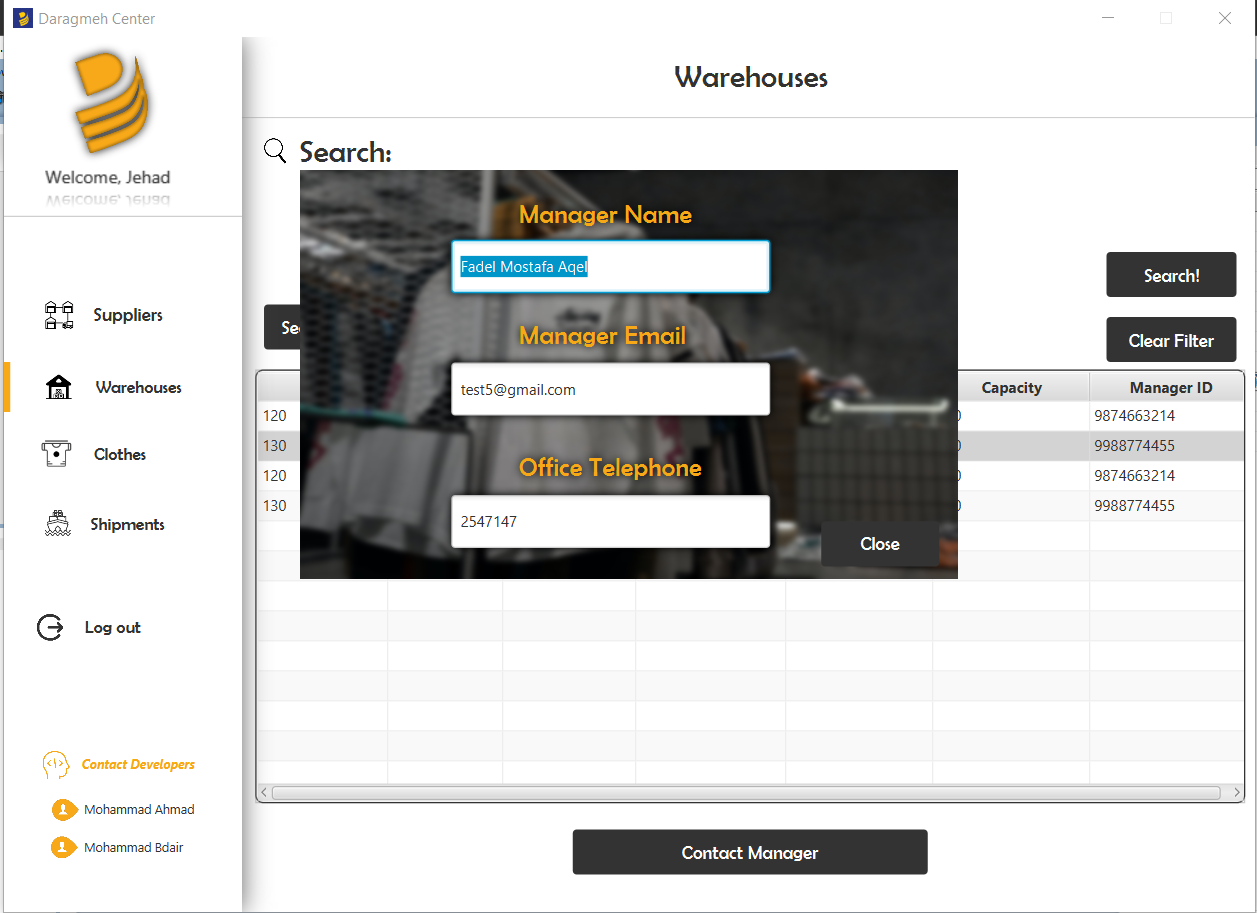
The locations tab pane shows suppliers locatinos and their shipment cost, which depends on the location as the distances varies.



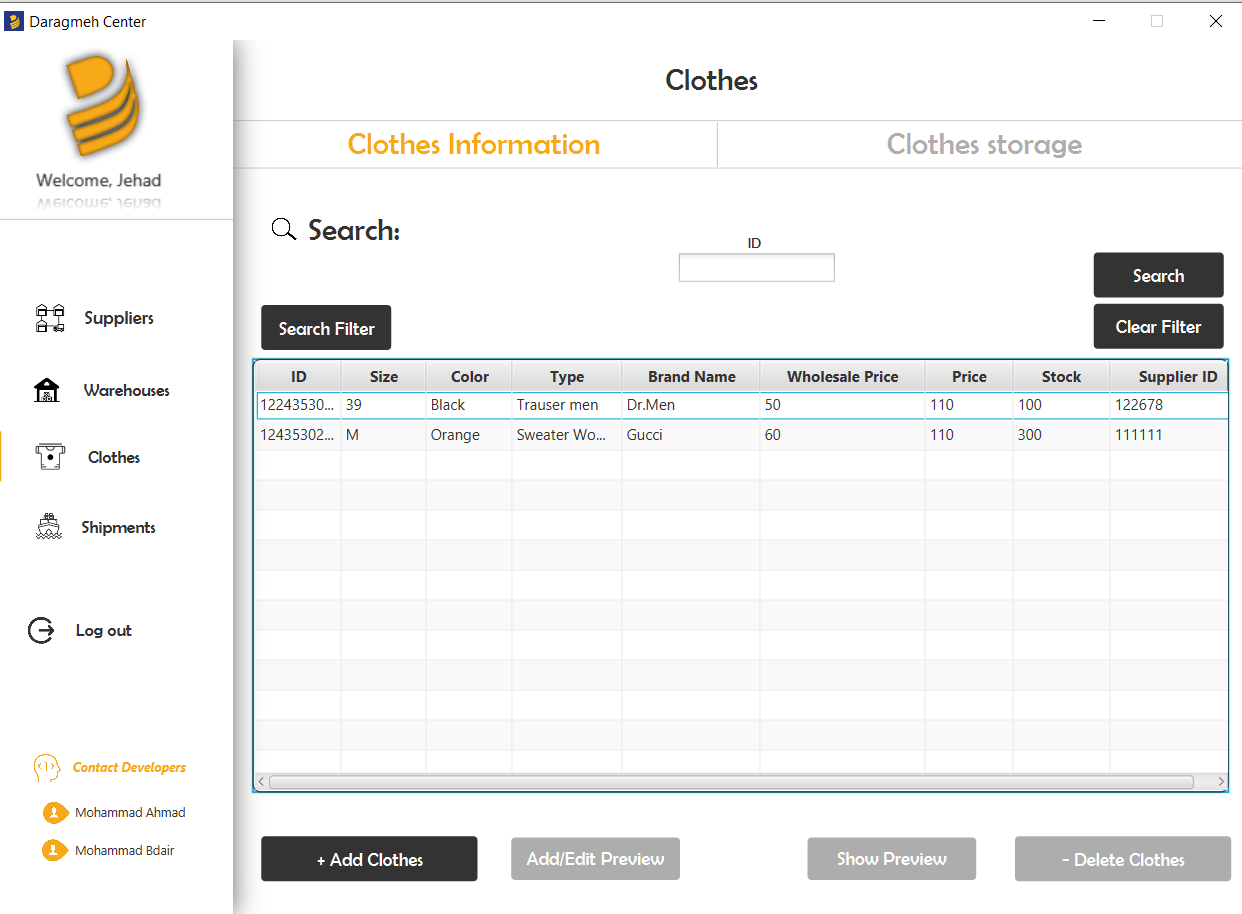
**Warehouses Page:**

****

This page shows all of the warehouses available, selecting one and clicking on the button shows the information of the owner of the selected warehouse, as if this owner wants to contact him or take clothes from him.



**Clothes Page:**

****

This page shows all of the clothes available in the manager’s warehouse and their stock.

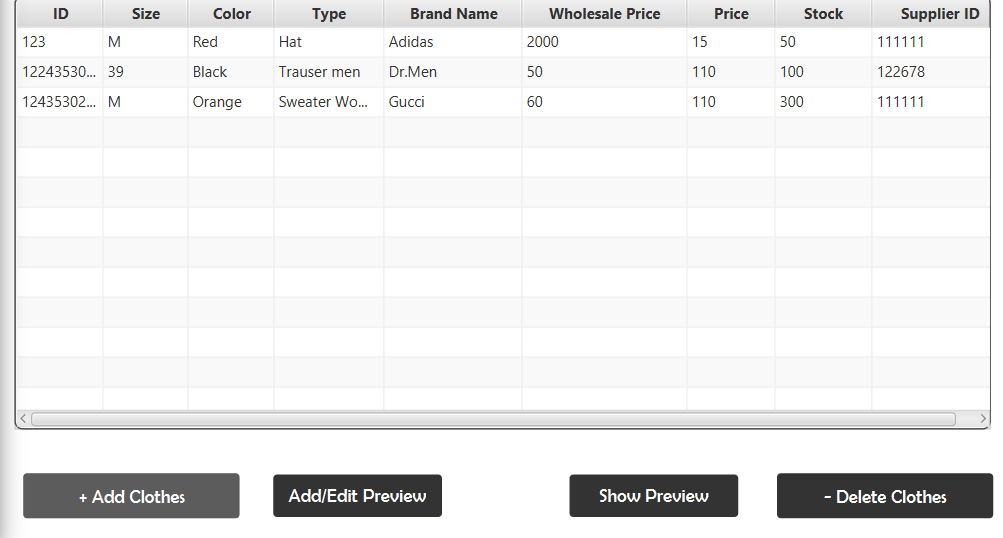
You can add or edit a preview for the clothes, if no previews are available a defualt preview picture will be assigned to the dress.



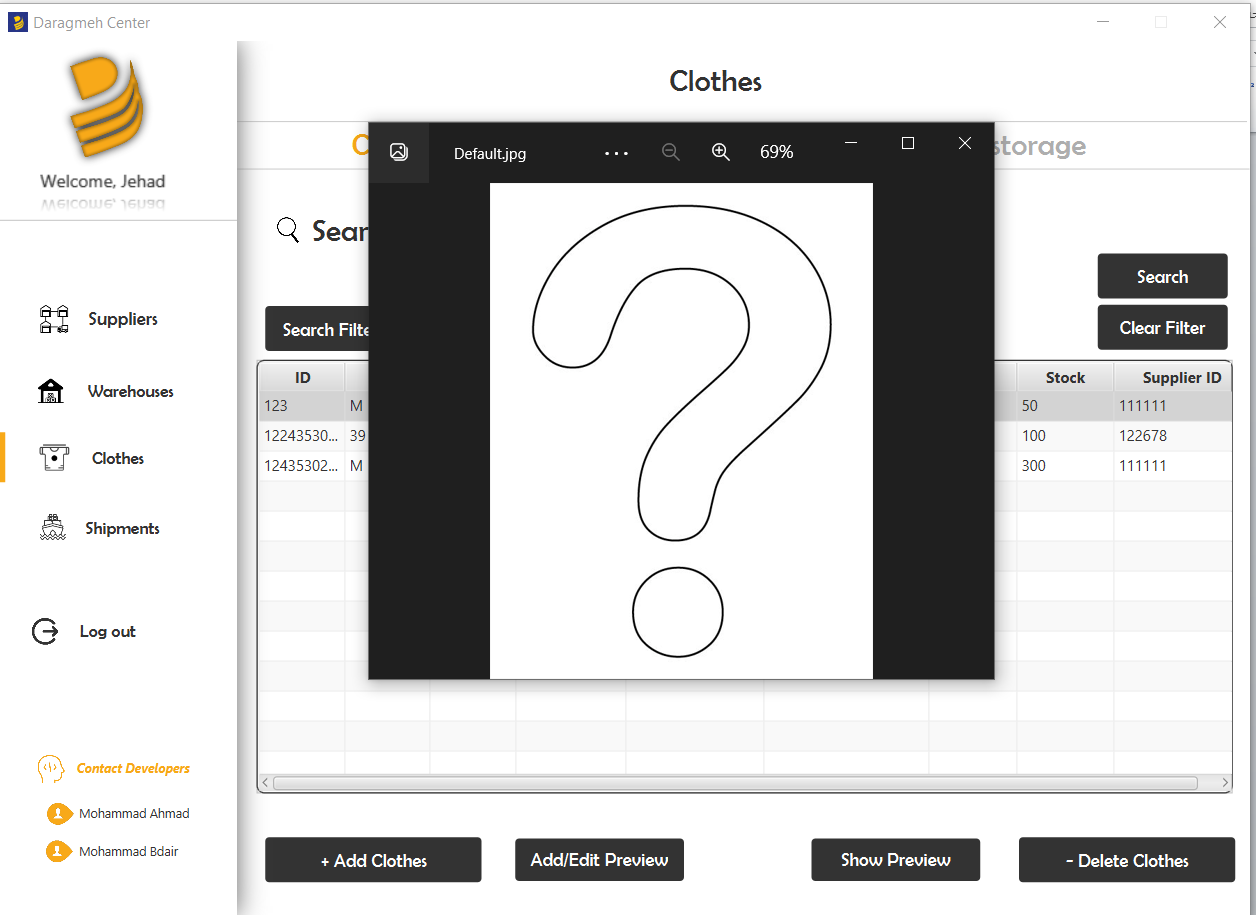


Not that you cant add exessive stock, if the entered stock exceeds the Capacity of the warehouse it won’t be added.

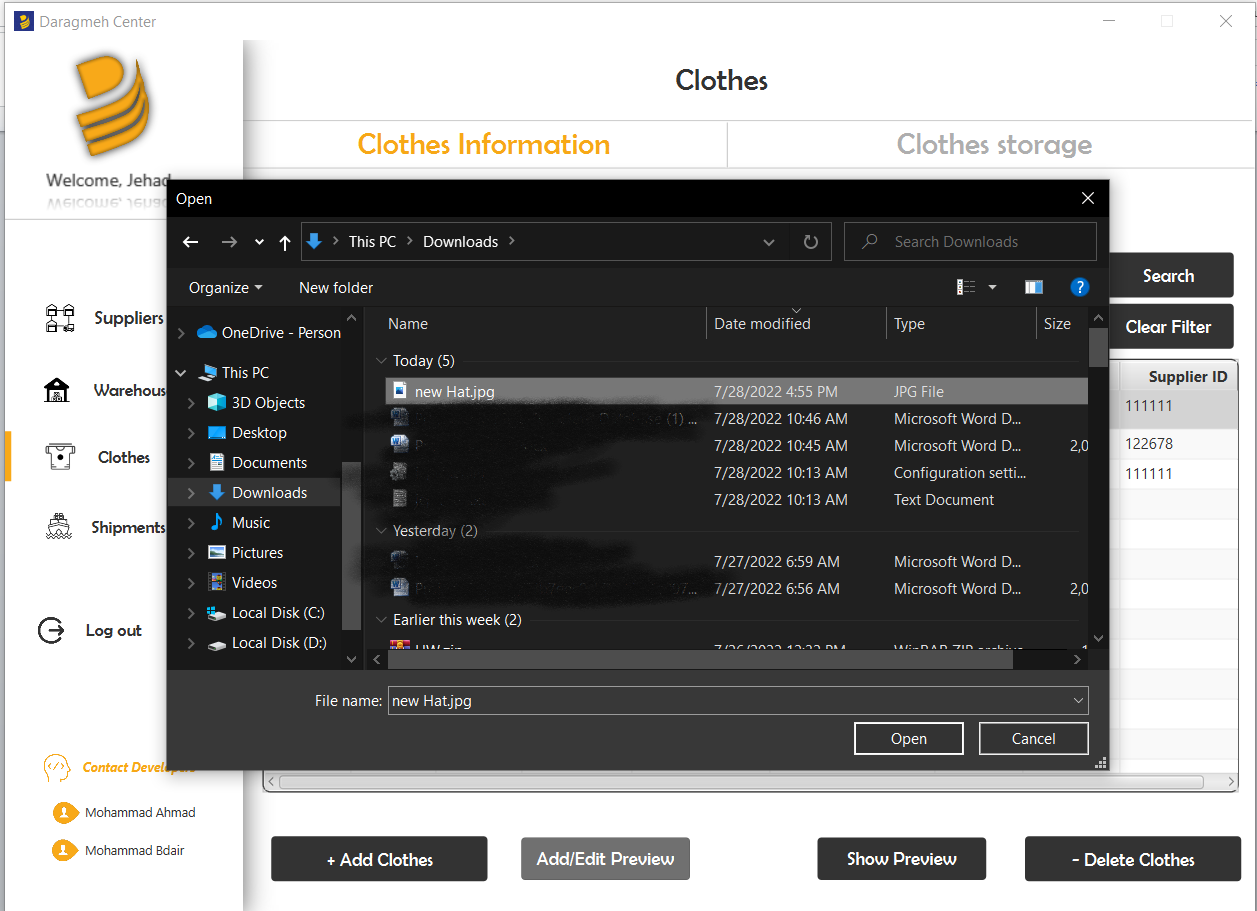




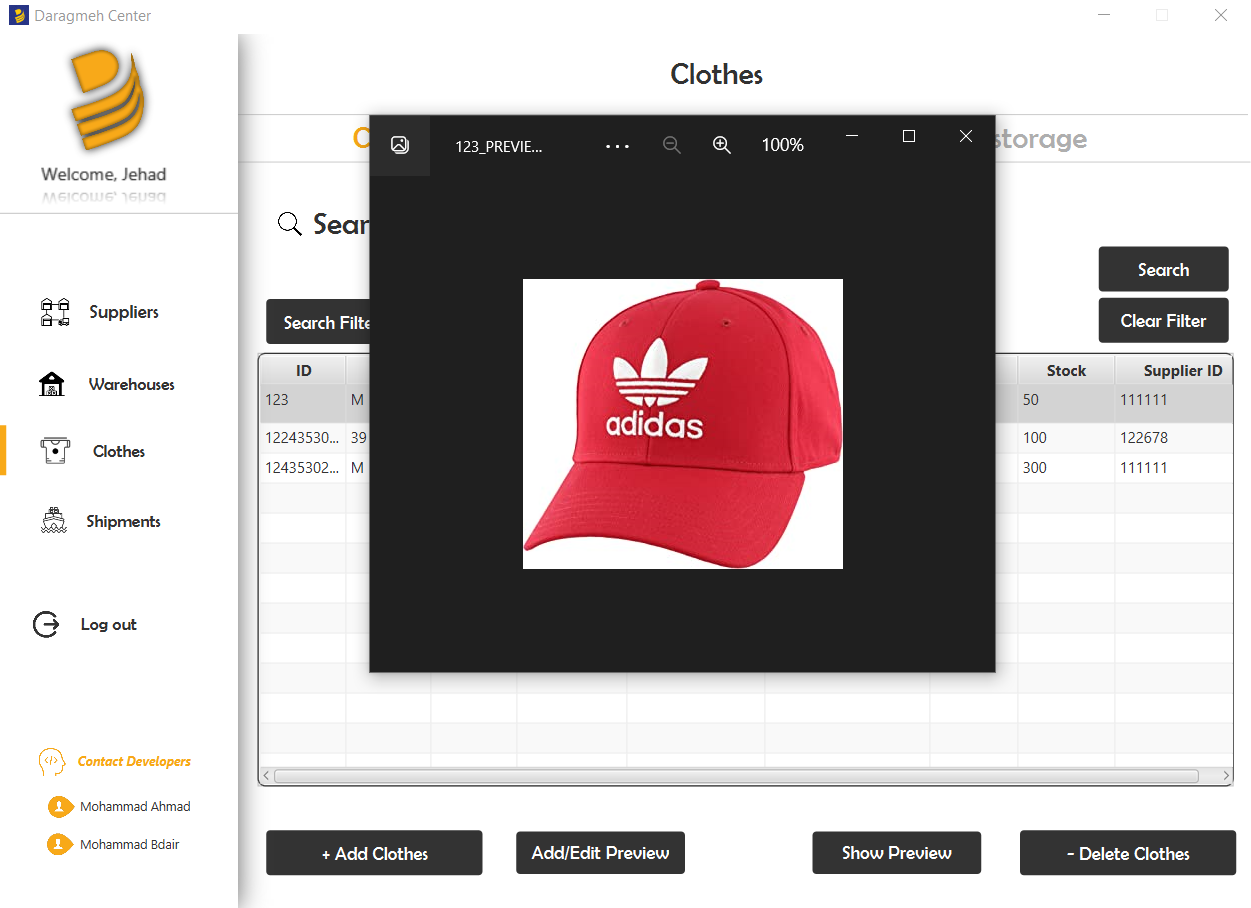
When clicking on show preview on the new imported clothes:



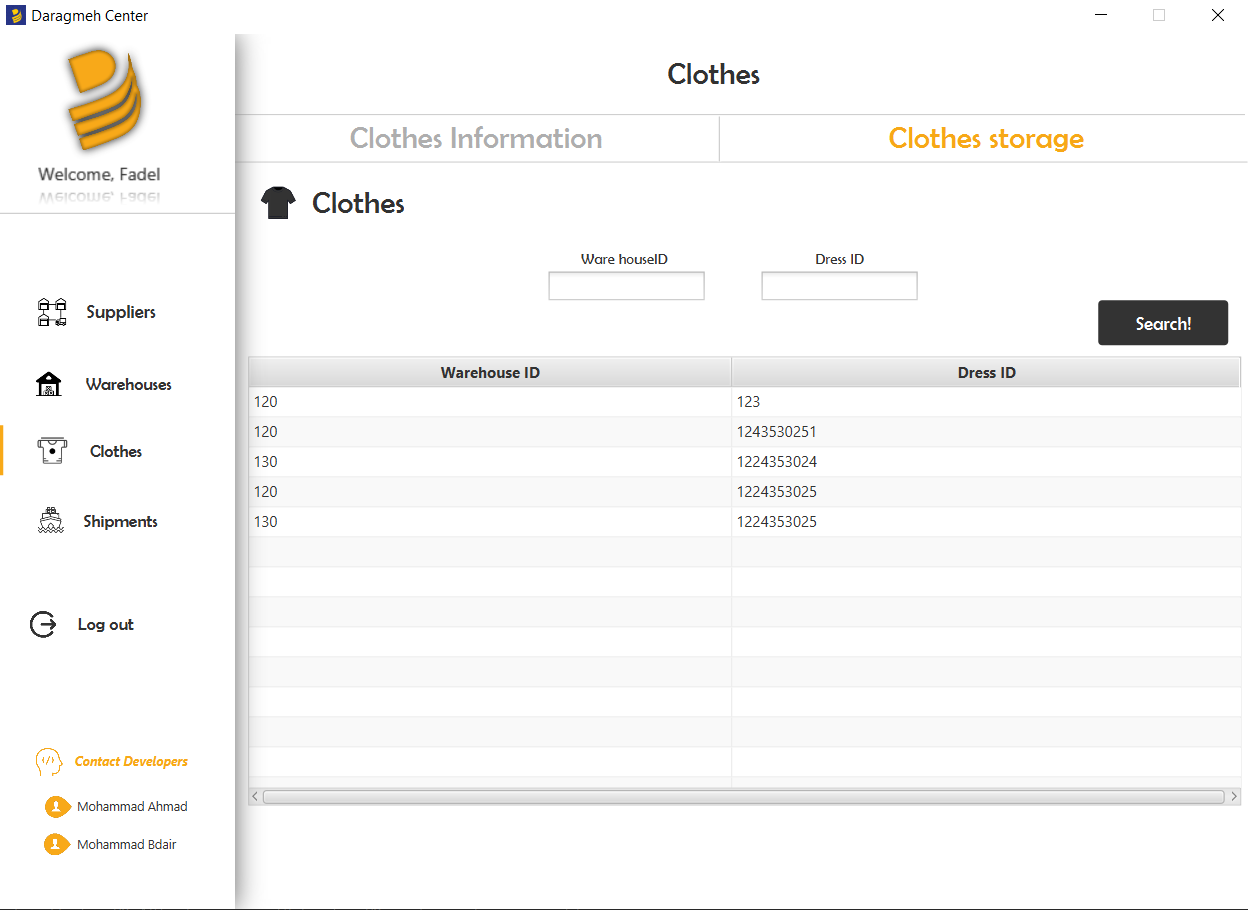
When adding a preview image



Then previewing it, a new imagie with the ID of this dress will be creaetd and viewed.

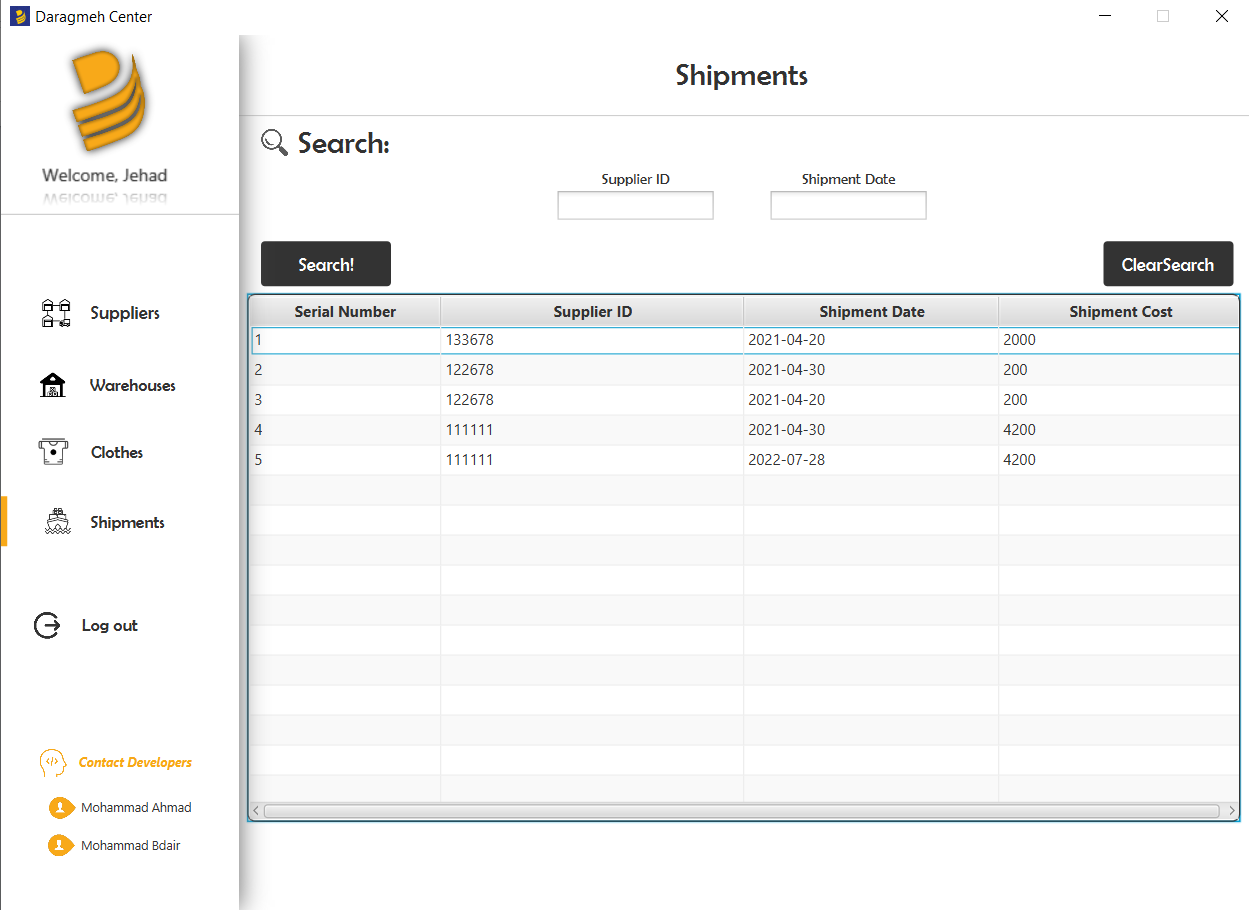


Clothes storage tab pane shows all of the available clothes in the warehouse that the manager manages.



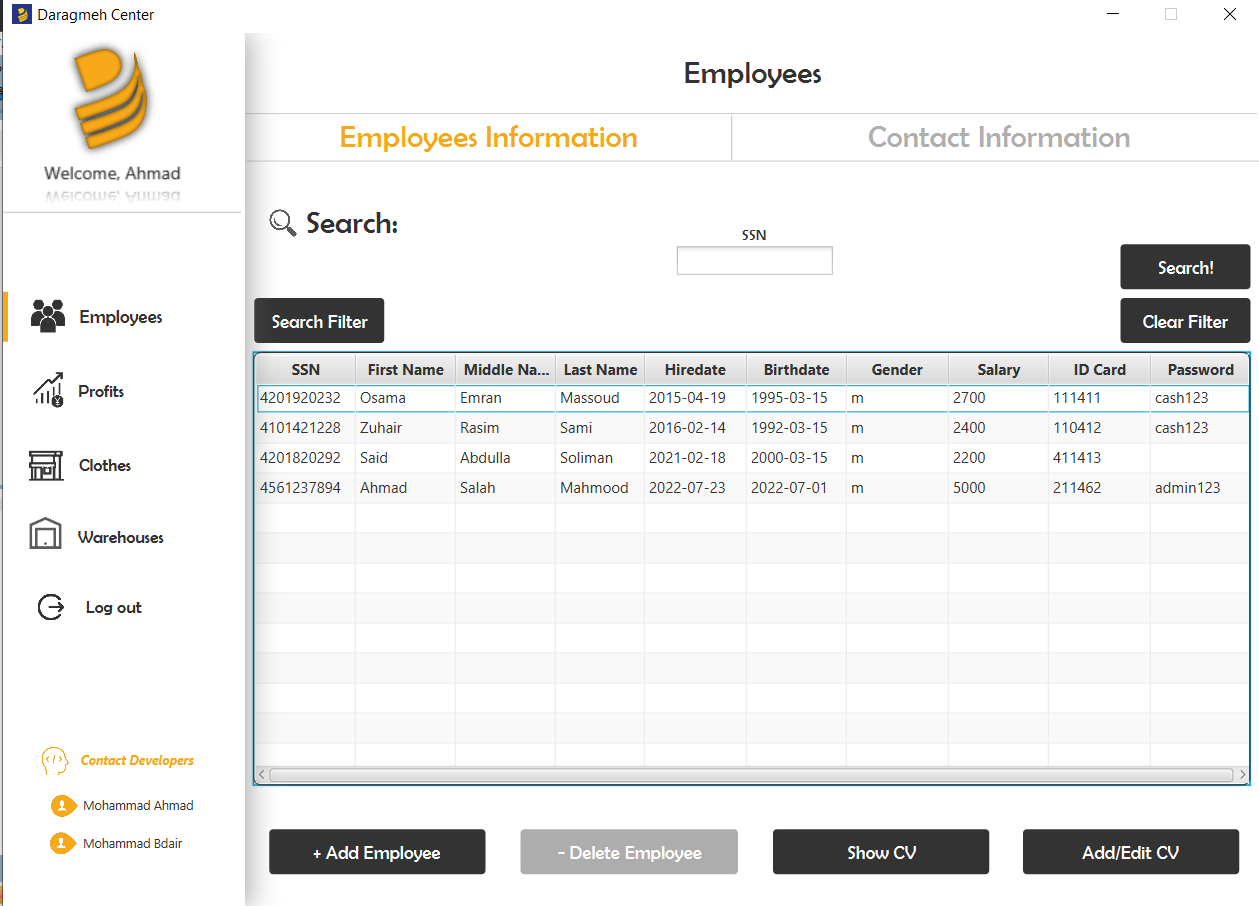
**Shipments page:**

This page shows every shipment that was made, whenever a new dress is imported, information will be stored in this table

****

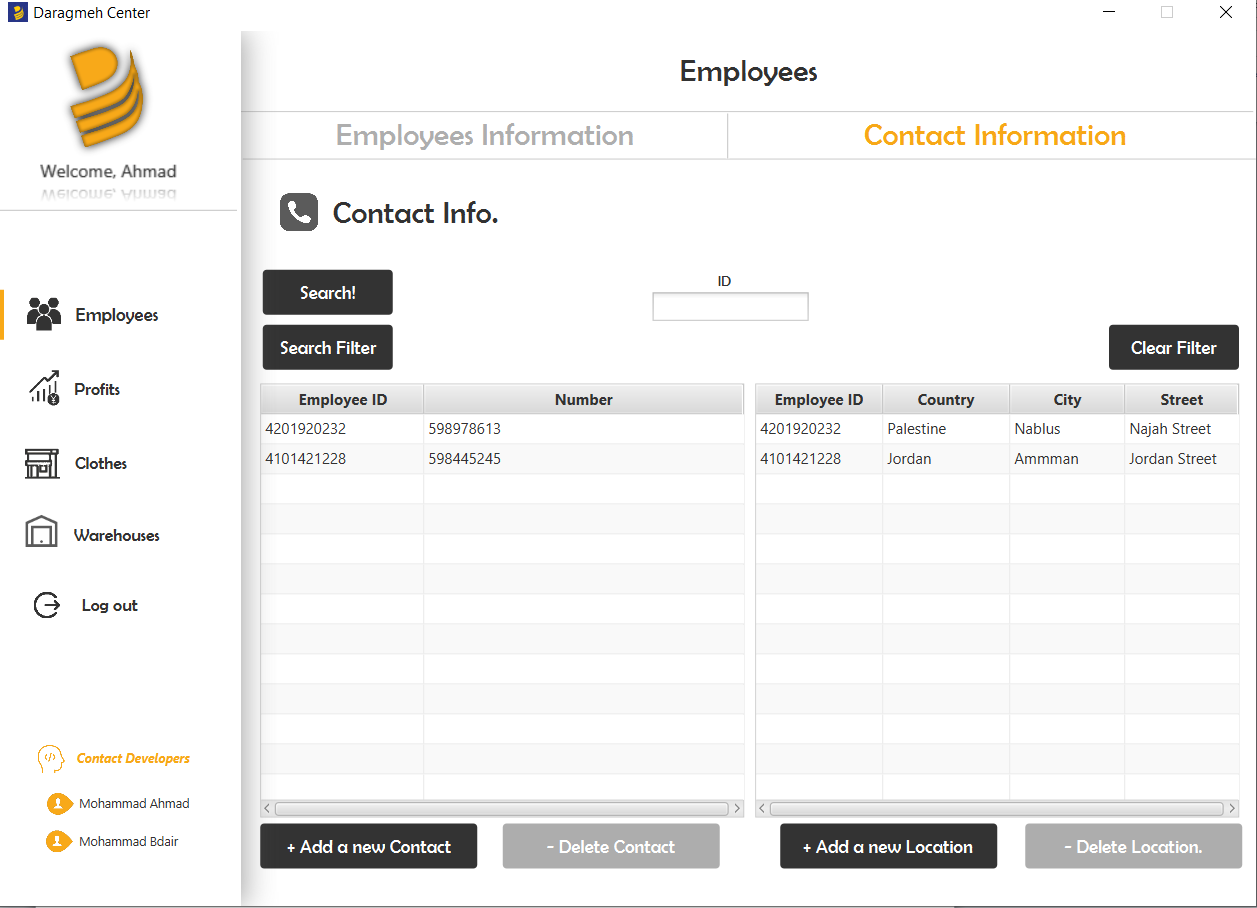
**Loggin in as a department manager**

Employees page: this page contains all the information about the cashiers and assistants that work in this department.



The manager can add or edit or preview CVs of hese employees, and he also can add new employees, notice that assistants don’t have passwords as they can not log in to this program.

**Employees information tap pane:** this pane shows locations and numbers of the employees



**Profits:**

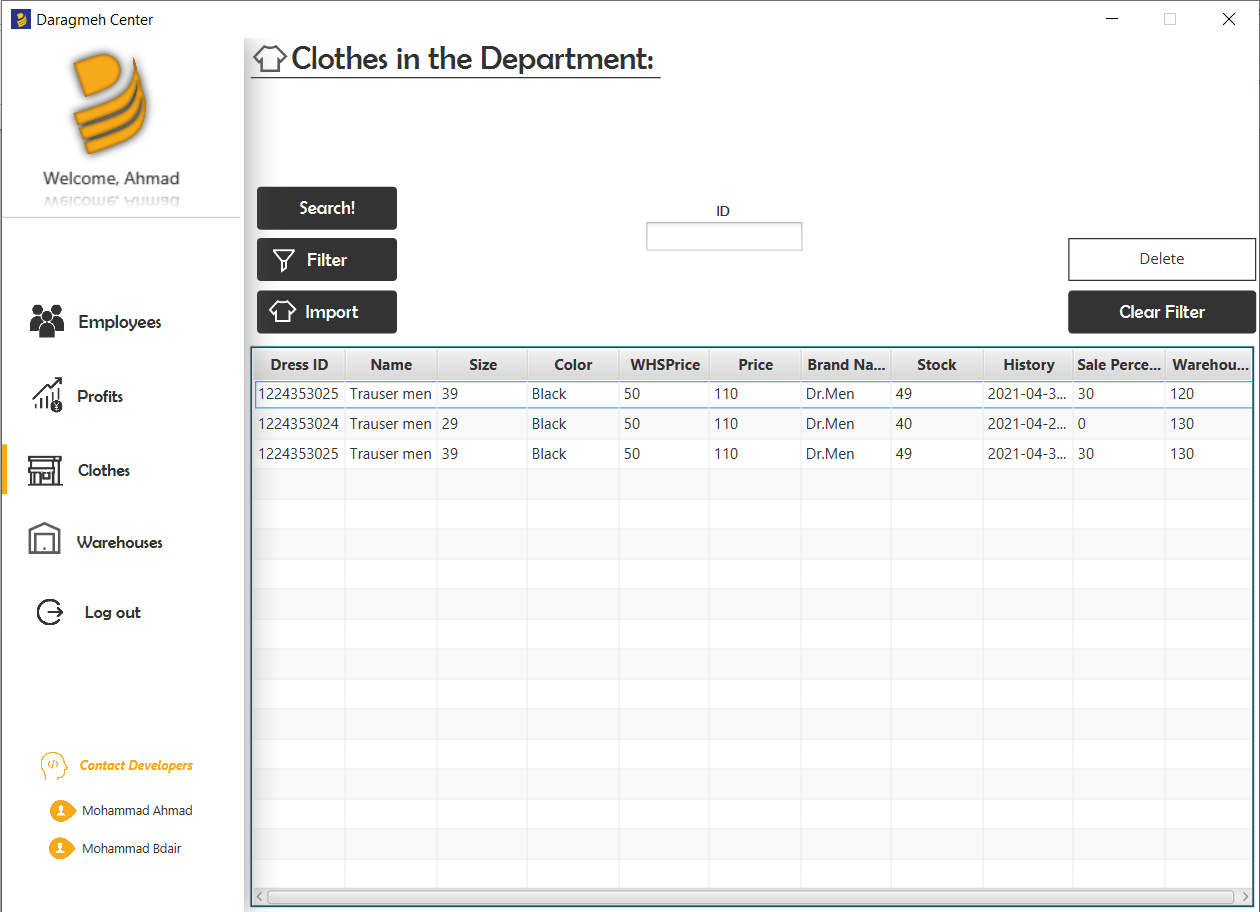
//////////////////

//////////////////

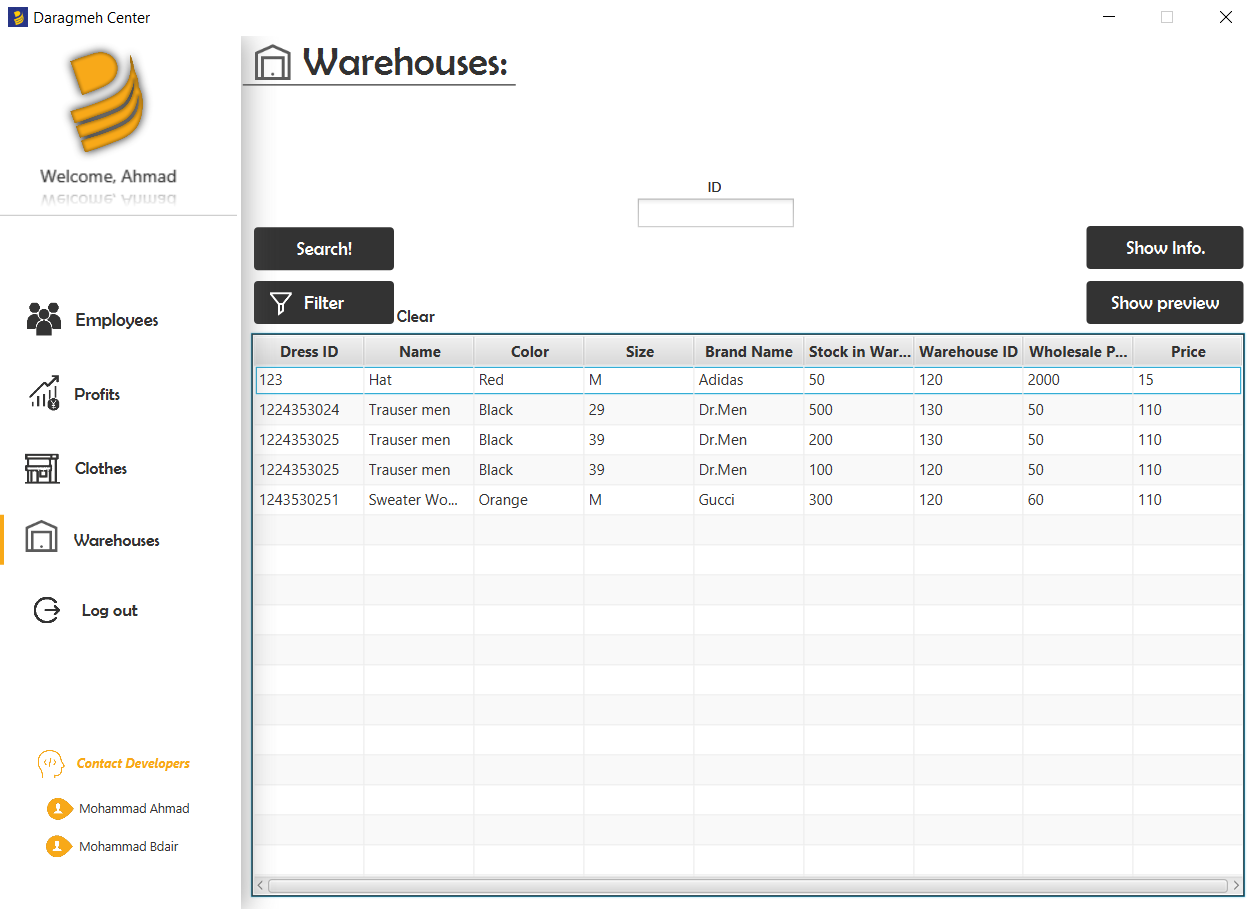
/////////////////

**Clothes Page:**

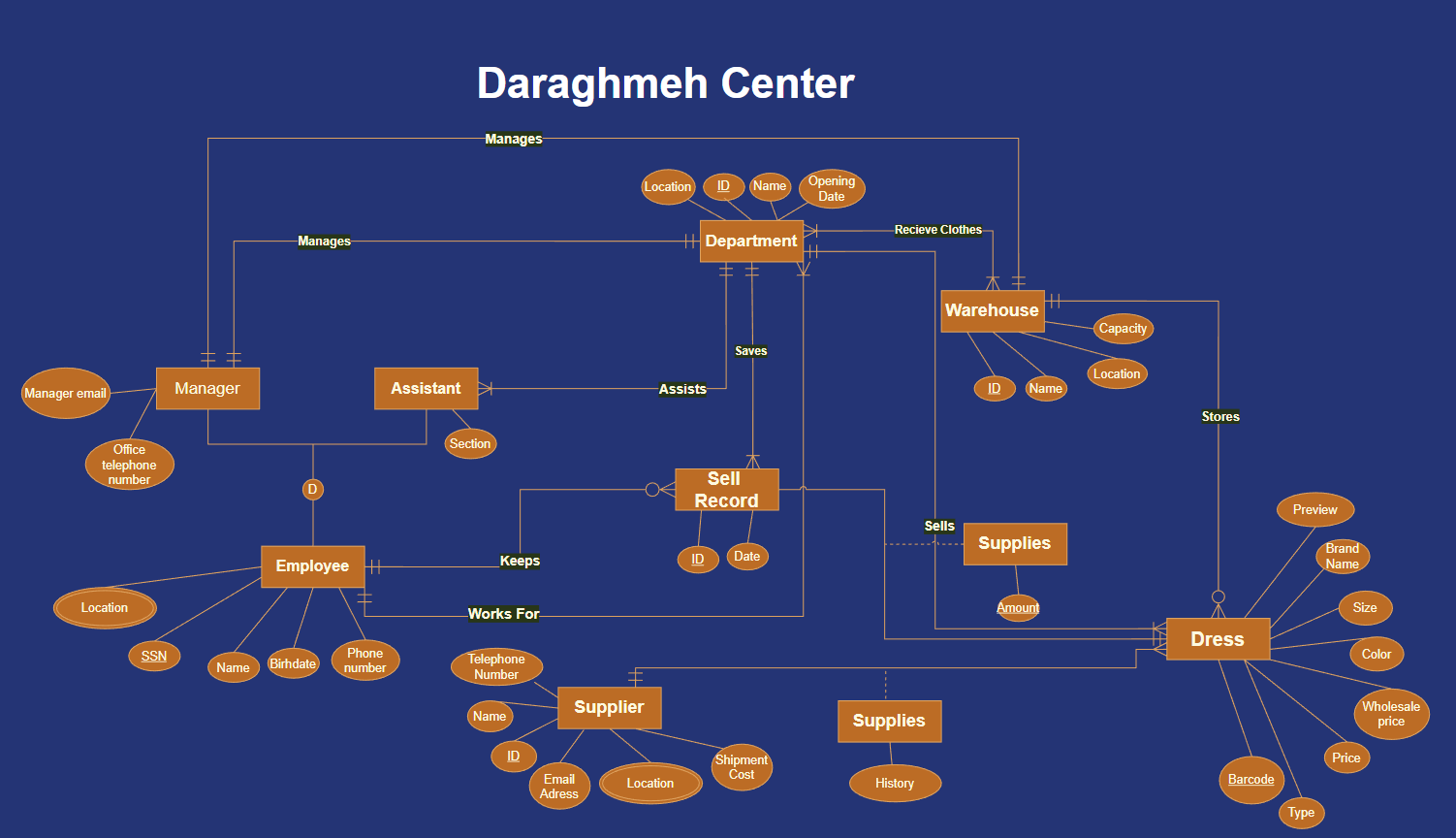
This page shows all available clothes in the department and their stock



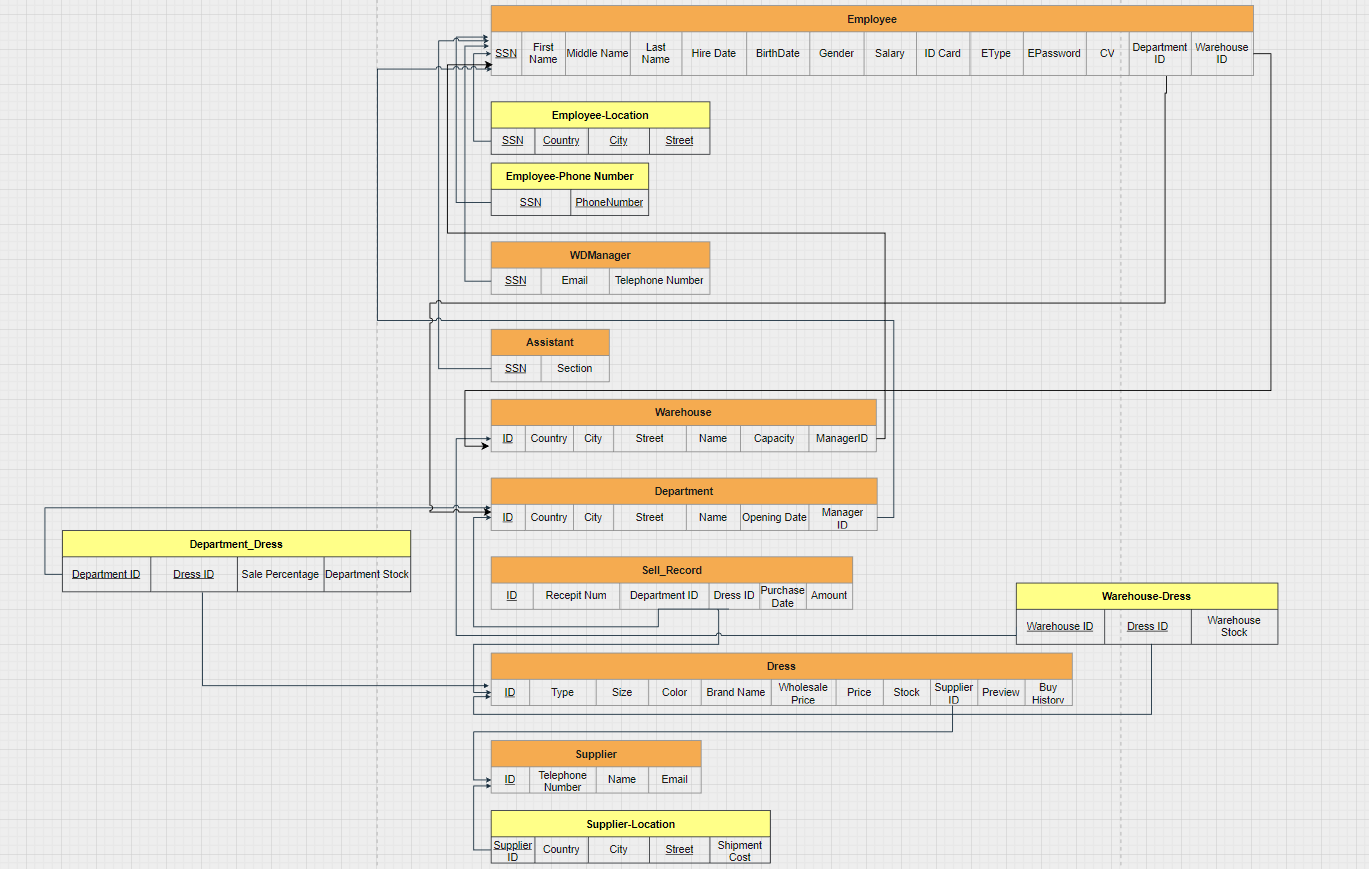
**Warehouses Page:** This page shows all the warehouses so the manager can import clothes from

****

EER Crows Foot notation:



# Mapping



# Existence Dependencies

Shipment ID depends on buy history of a dress.

Shipment Cost depends on Supplier location.

Sell record ID depends on cashier when he sells a dress

# Functional Dependencies

**Assistants table:**

SSN → Section

**Departments table:**

DepartmentId → Country, City, Street, name, Manager ID, Opening Date.

ManagerID → Department ID

**Dress table**:

DressID → Dress Type, Size, Color, Brand name, Wholesale price, price, stock, supplier ID, preview, buy history.

**Department\_Dress table:**

DressID, Department id → Sale percentage, Department Stock

**Employee table:**

SSN → First name, middle name, last name, hire date, birthdate, gender, salary, IDCard, cv, warehouse id.

IDCard → Type, Password. DepartmentID.

**Employee\_Location table**:

SSN,Country, city, street → SSN, Country, city, street

**Employee\_Phonenumber table:**

SSN Phone number→ SSN, Phone number

**Sell\_Record table:**

ReceiptID→ Receipt Number

Receipt number→ Department ID, Dress id, Purchase Date, Amount

**Supplier table:**

SupplierID→ Telephone number, Email, Supplier name

**Supplier\_Location table:**

SupplierID, Country, City, Street→ Shipment Cost

**Warehouse table:**

WarehouseID → Country, City, Street, Warehouse name, Manager ID, Capacity

ManagerID → Warehouse ID

**Warehouse\_Dress table:**

Warehouse ID, Dress ID → Warehouse Stock

**WDManager table:**

Manager SSN → Manager email, Office Telephone Number

# Normalization

**1NF:** All tables satisfy the First normal form because they all have only one value per cell (intersect between a row and a column)

**2NF:** Tables containing composite primary key: Employee\_Location, Employee\_Phonenumber, Supplier\_Location, Warehouse\_Dress, All of these tables are in 2NF because all dependent columns fully depend on the composite primary key. Remaining tables are in 2NF because they do not have composite keys.

**3NF**: Tables not in 3NF:

**1) Employee table**

Reason: (SSN → IDCard) and (IDCard →DepartmentID, Type, Warehouse ID, Password)

So SSN → DepartmentID, Type, WarehouseID, Password

Solution: create 2 tables: 1st table contains SSN, Firstname, middle name, salary, IDCard, with SSN as a primary key and IDCard as a foreign key referencing 2nd table.

2nd table contains IDCard, DepartmentID, warehouse ID, type, password., with IDCard as a primary key.

**2) Sell\_Record**

Reason: (ReceiptID → ReceiptNum) and (ReceiptNum →DepartmentID, DressID, Amount, Purchase Date)

So ReceiptID → DepartmentID, DressID, Amount, Purchase Date

Solution: create 2 tables: 1st table contains ReceiptID, ReceiptNum with ReceiptID as a primary key and ReceipyNum as foreign key referencing 2nd table

2nd table contains ReceiptNum as a primary key and DepartmentID, DressID, Amount, Purchase Date.

**BCNF**

All tables are in BCNF(After applying 3NF), because all determinants are candidate keys.

# Resources

**Books:**

* Database Systems: Introduction to Databases and Data Warehouses 2ed edition by Nenad Jukic , Susan Vrbsky, et al. - latest edition
* Mastering Oracle SQL and SQL\*Plus 1st Corrected ed., Corr. 2nd printing Edition
* Learn JavaFX 17: Building User Experience and Interfaces with Java 2nd ed. Edition for Kishori Sharan ,Peter Späth

**Libraries:**

* [**https://github.com/palexdev/MaterialFX**](https://github.com/palexdev/MaterialFX?fbclid=IwAR0pBDqkRaoJlC97oo5hQy6EQEFFjo1U-3c2w2o-pQucxVtvMU1UFJvwQAQ)
* [**https://github.com/controlsfx/controlsfx**](https://github.com/controlsfx/controlsfx)
* [**https://github.com/sshahine/JFoenix**](https://github.com/sshahine/JFoenix)

**Other resources:**

* W3School
* Youtube
* JavaFX DOCUMENTS
* MaterialFX DOCUMENTS
* Some employees at Daraghmeh center located in Nablus.
* Personal friends and experts