Online Clothing Store System

GUI DESIGN & DATABASE CONNECTIONS – FE2 PROJECT

EOGHAN O’BRIEN, THOMAS MURDOCK & AARON

2021

**Table of Contents**

[**Introduction** 1](#_Toc70094038)

[**High-Level Description of System** 2](#_Toc70094039)

[**Description of Persistent Storage** 3](#_Toc70094040)

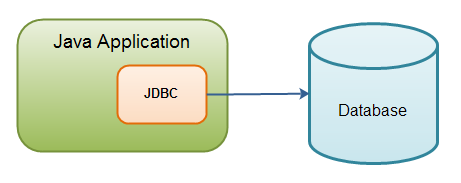
[**Flowcharts** 6](#_Toc70094041)

[**10 Usability Heuristics** 7](#_Toc70094042)

# **Introduction**

This is the design document for our formal element two in the GUI Design & Database Connections. The project idea is an online stores system in which we use many elements such as Java Swing & MS Access databases. The members of this group are Eoghan O’Brien, Thomas Murdock & Aaron Broe. In this report, we will outline the structure of this application and the different elements within this that outlines the GUI, database and Java aspects present within, in order to fully create our system accordingly.

Link to our GitHub: <https://github.com/ClothingOnlineDatabase>



# **High-Level Description of System**

Like mentioned above, this system is an online clothes store system, in which we are taking inspiration from many online clothes vendors on the real market such as ASOS and Boohoo. These online vendors are very popular, especially with young adults, in which they can view many popular brands and what they have to offer, as well as many different types of clothes, such as hoodies, trousers and trainers. With this inspiration in mind, we will be looking towards having a system in the clothing industry in which the user will be able to create orders containing clothing of their choice, in which they can purchase or rent items. The rental of items would apply to the rental of suits typically, which will be aimed to be achieved within this schema.

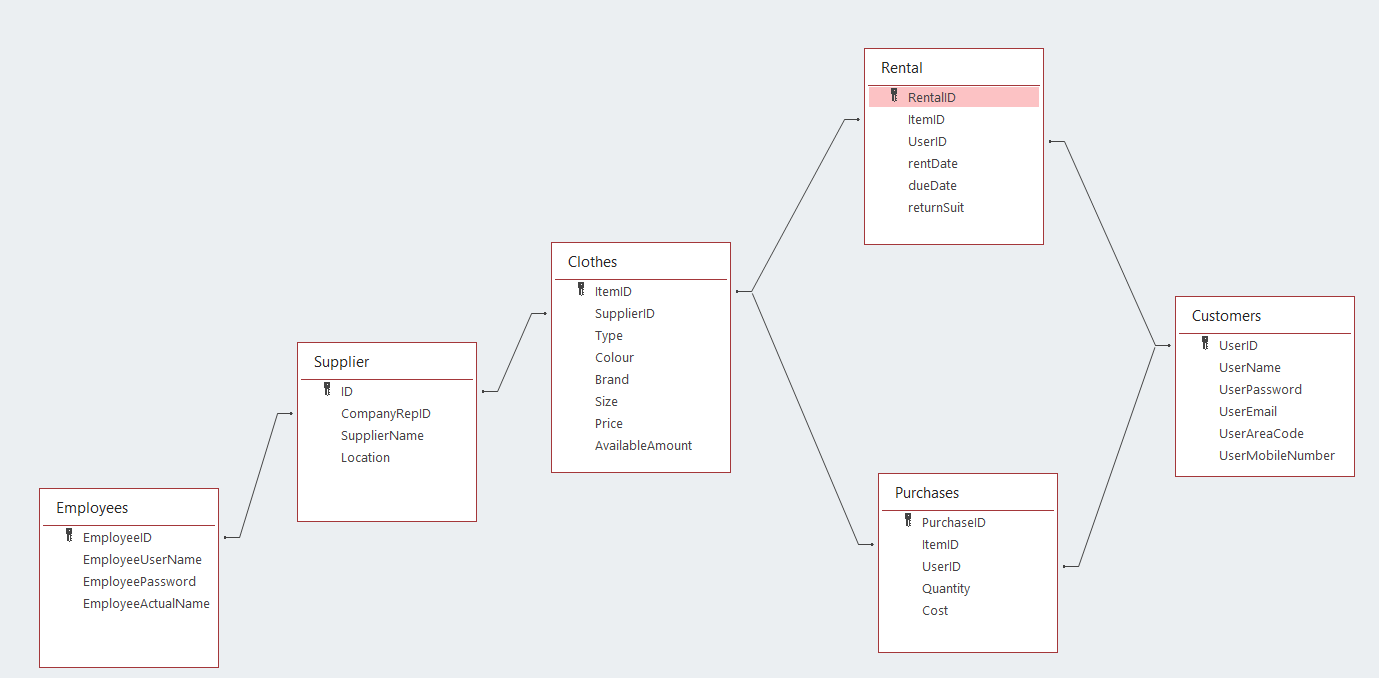
Additionally to the aspect of users being able to view and order clothes within this system, there is a need to be able to maintain the system and view clothing/order status. For example, it is required for us to “allow for stock/inventory control to be maintained”. Therefore, we will need to have an employee in which they have direct contact with a supplier of the goods. This supplier will have their details such as their name and location, and they will be providing the company with clothing, which will be seen in a clothing amount on the table which shows what clothes are present.

From this, the system will have a database in which a customer and employee can log into this from a “web portal” style login page. The customer will be able to view clothing in which they can add to orders for clothing that they like. The employee will be able to perform the various tasks in which they can perform maintenance tasks such as stock control and removal/addition of users. Following this overview of this system, we will now outline the various different concepts and supporting documentation to help assist with describing this system overall.



# **Description of Persistent Storage**

A huge requirement for this project is to have a database system for persistent storage in which we will have a database working alongside our Java files which will be used to store different users and clothing items as required by the customer or employee. Below is the tables from our database and the relationships between these tables:

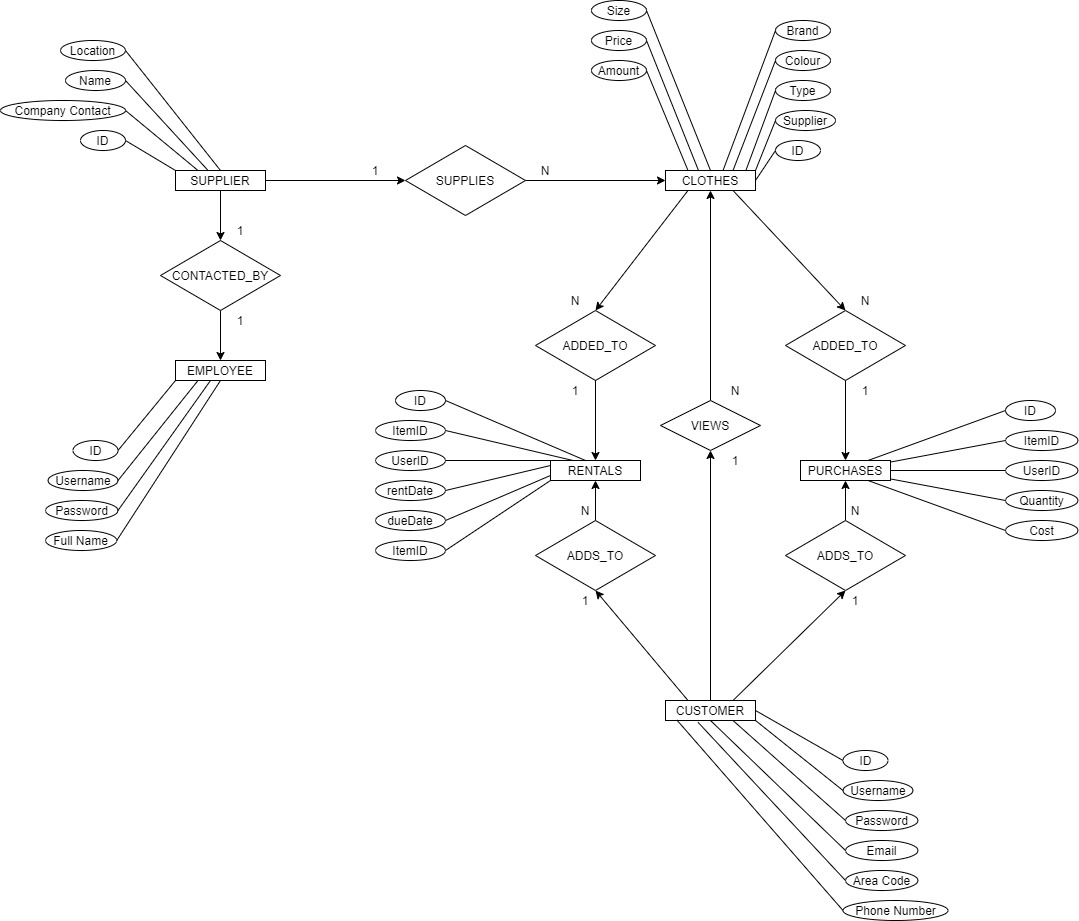


As seen within the above table:

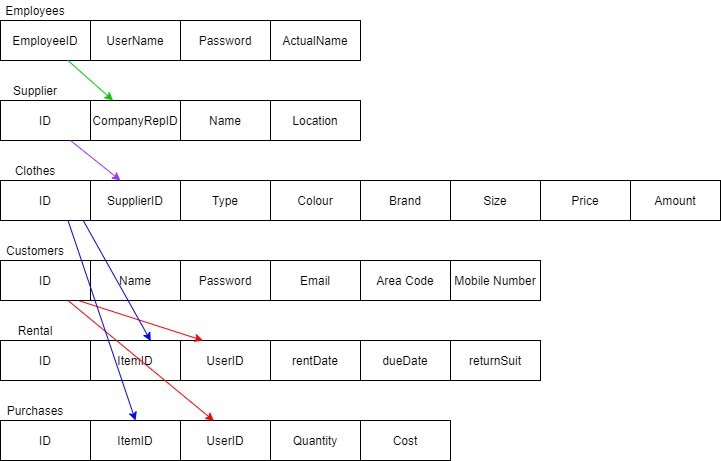
* There are six tables: Employees, Supplier, Clothes, Customers, Rentals & Purchases.
* There are primary keys in every table which is necessary as there are various foreign keys required for tables. For example, two large elements of this schema are the rentals & purchases tables as the customer will be looking at the clothes as part of this system. There are foreign keys in this to indicate what customer is purchasing/renting, and what clothing is being purchased/rented, which is identified by the “ID” element, acting as foreign keys from other tables

To assist in describing the above schema, we have created an EER Diagram & relational model, in which they take an overall view of this schema and show the different allocations and relationships within.

**EER Diagram**

****

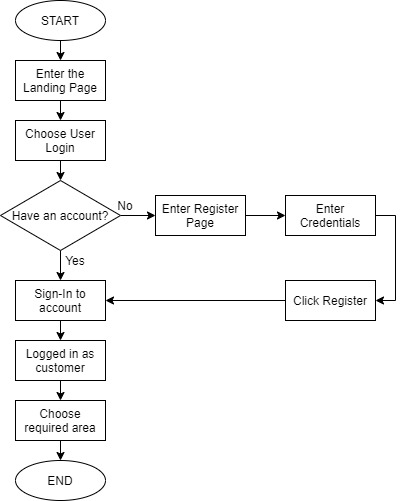
**Relational Model**



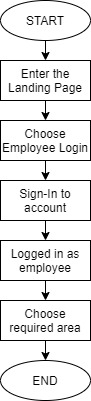
# **Flowcharts**

To show some of the inner workings of the code created for this project, we will present to you flowcharts which further show the workings of the login pages for this system as there is two different types of individuals that can be present within this system, being a customer or an employee. Below are flowcharts for these two separate routes: employee flowchart & customer flowchart.

**Customer “Route” Flowchart**

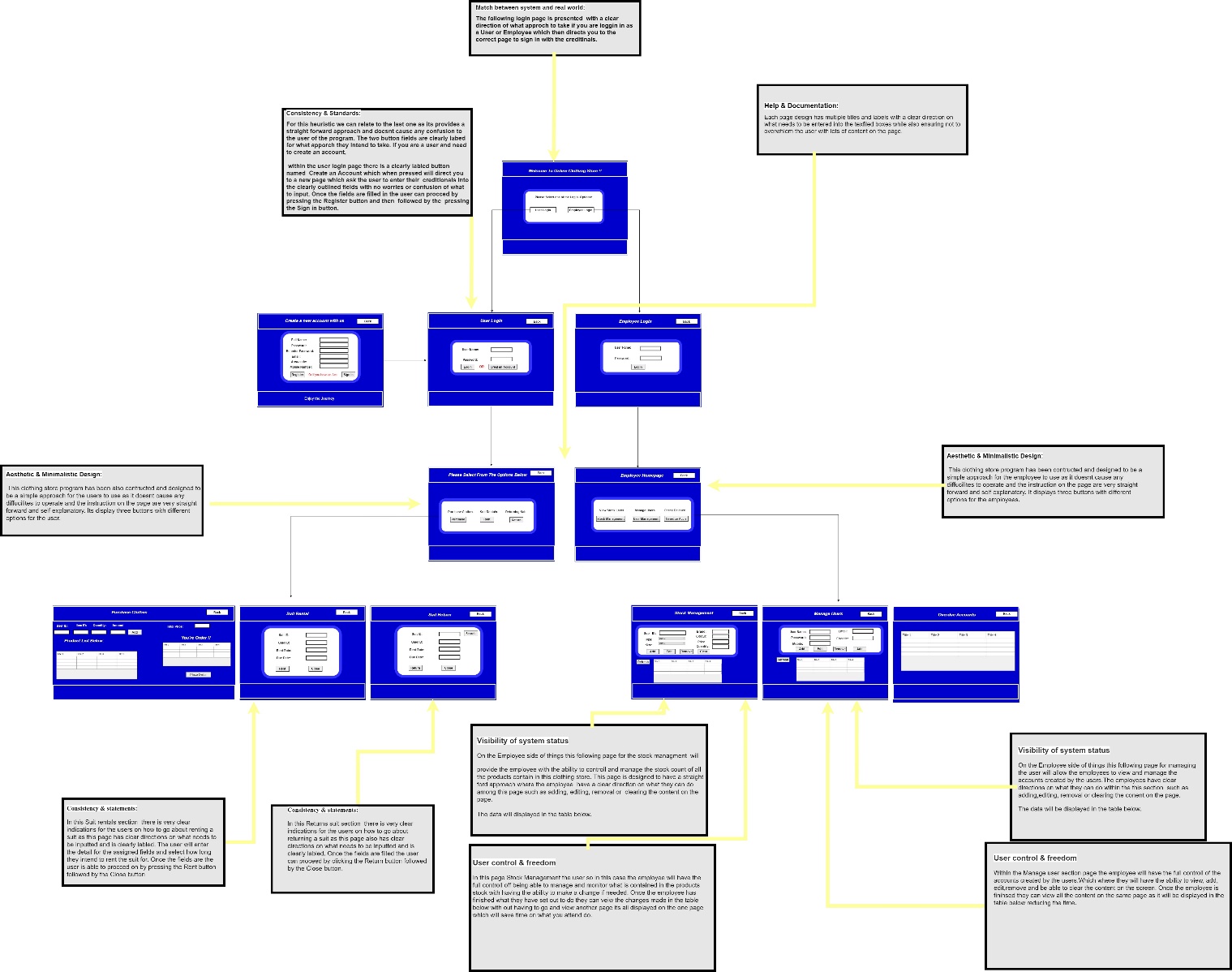


**Employee “Route” Flowchart**

****

# **10 Usability Heuristics**

Below are the usability heuristics for this application. We have attached the file in which this was created for further inspection.

****