A close up of a logo

Description automatically generatedDESIGN DOCUMENT  
COMP 4604 GUI Design & DB Connectivity

By

Ricardo Gutierrez

Eric Behan

April 2022

# System description

The system created for this project is designed to be used as a management system for a company that rents out cinema equipment to its customers. the system is comprised of a software application developed in Java code using the NetBeans integrated development environment (IDE) and a relational database made in Microsoft Access. When using the system, the user is first prompted to provide login credentials, this is done to require user authentication before accessing the system. The provided credentials, username and password, are then used to query the Admin table to look for a record with the username and password provided. If there is a record the user can then access the system. If not a message is displayed to the user that the credentials provided are incorrect.

Then the main menu of the system is displayed, from this menu a user can access one of the five different functions of the system. The system allows the user to manage the equipment of the business through the equipment section. From this section the stock levels of each equipment type can be updated, new equipment can be added to the table and equipment can be removed from the table. The customers section allows for the management of the various customer records of the business. Customer records can be added, updated and removed from this section. The orders section allows the user to view all of the orders the business has had, view all active orders i.e. orders which haven’t been returned yet but are still within the rent period and view overdue orders which are orders that haven’t been returned and have gone past there return date.

The rent section allows the user to add new rent orders. The rent order can only be added to an existing customer record so first a customer record must be added. Once the rent record is put into the system the equipment table will be updated to change the quantity of equipment to be rented. The return section allows the user to mark a rent order as returned. If the order that was returned is overdue a fine is added to the rent order and the fine is displayed. All of the information from each section is taken from / entered into the database.

Graphical user interface, application

Description automatically generated

Figure All of the windows of the system

# System design

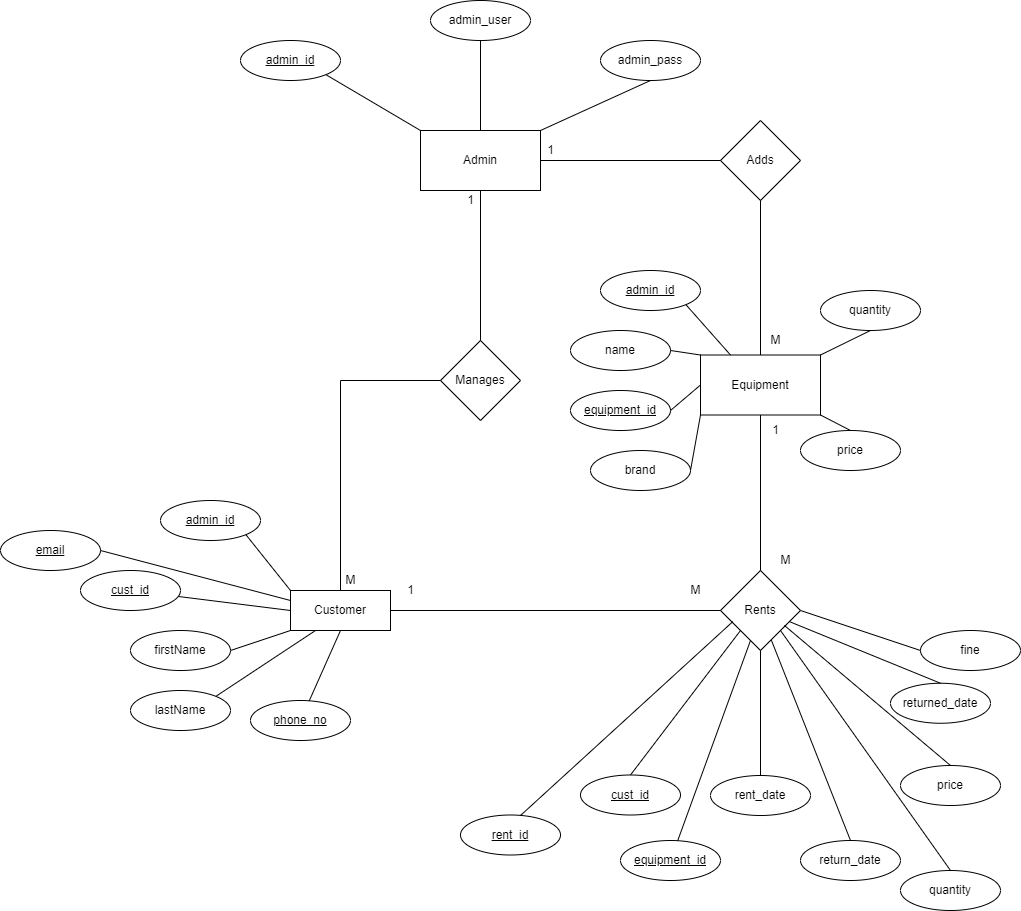


Figure EER Diagram

The enhanced entity relationship (EER) diagram above shows how the entities in the database will relate to each other and it also shows the attributes and primary keys of the entities. Entities are objects that represent important data, the EER diagram is a high-level representation of the database that will be used in this project. When developing an EER diagram and its relationships it is important to first understand the scope of the database, for this system the data will be stored in a relational format to allow for the various parts of an order to all be updated as customer details, rent orders and returns are processed by the system.

# System Flowchart

The below flowchart is a high level flowchart of the system and its operation.

Diagram

Description automatically generated

Figure Flowchart

# Persistent Storage

Diagram

Description automatically generated

Figure 3 Relationship Table

|  |  |
| --- | --- |
| Admin | |
| admin\_id | Auto increment integer |
| admin\_user | String |
| admin\_pass | String |

1.

Table Admin Table

Admin table stores the admin username and password, which manages the creation/deletion of equipment and customers.

|  |  |
| --- | --- |
| Customer | |
| cust\_id | Auto increment integer |
| Email | String |
| f\_Name | String |
| l\_Name | String |
| phoneNo | Integer |
| admin\_id | Integer |

2.

Table Customer Table

In this table, this stores the customer information. There is a column for admin\_id because it will store here which admin added/deleted/updated the customer’s information. Moreover, email and phoneNo must be unique – they cannot be duplicated.

|  |  |
| --- | --- |
| Equipment | |
| equip\_id | Auto increment integer |
| equip\_name | String |
| Brand | String |
| Price | Double |
| Quantity | Integer |
| admin\_id | Integer |

3.

Table Equipment Table

When it comes to storing the equipment information or what others call ‘inventory’, this is the table for it. This is being managed by the admin user as well.

|  |  |
| --- | --- |
| Rent | |
| rent\_id | Auto increment integer |
| cust\_id | Integer |
| rent\_start | Date |
| return\_date | Date |
| Quantity | Integer |
| equipment\_id | Integer |
| Price | Double |
| Fine | Integer |
| returned\_date | Date |

4.

Table Rent Table

This table is the record of customers renting an equipment from our inventory. If the customer rented an equipment, fine and returned\_date are null, this will only be filled after the customer return the equipment rented from our inventory.

# Usability Heuristics

1. **Visibility of System Status** – In our application, an example of the first usability heuristic is when the user entered a wrong user credentials, our application will give the user feedback indicating that the details are wrong.  
   Graphical user interface, text, application

   Description automatically generated
2. **Match between system and the real world –** The interface is very straightforward; it is easy for the user to learn and remember how the interface works. We have ensured that users can understand meaning without having to go look up a word’s definition.

Graphical user interface, application, Word

Description automatically generated

1. **User control and freedom –** User can clear all the textfields with a press of a button and go back to main menu by clicking the home button.

Graphical user interface, text, application

Description automatically generated

1. **Consistency and standards–** We uses the same format in every window – home button on the upper left, combobox for IDs, disabled textfields to maintain consistency in our application.

Graphical user interface, application

Description automatically generated

1. **Error prevention –** To prevent users in committing an error, we only show the available IDs in every combobox ID like the Equipment ID where it only shows ID 12 and 21 instead of the user inputting an ID. Also, the textfields are disabled, it only shows the data from the database.

Graphical user interface, application

Description automatically generated

1. **Recognition rather than recall –** To make life easier for the user, instead of going back to the Orders window to see the name of the customer that rented the specific equipment, the application will automatically display the customer’s name, equipment name, and the return date. The user just need to select which Rent ID from the combobox.

Graphical user interface, application

Description automatically generated

1. **Flexibility and efficiency of use –** This has not been implemented in the application.
2. **Aesthetic and minimalist design –** We have ensured that the visual elements of the interface support the user’s primary goal.

A screen shot of a video game

Description automatically generated with low confidence

1. **Help users recognize, diagnose, and recover from errors –** The example below shows that we have shown the error message in plain language, precisely indicated the problem, and constructively suggested a solution.

Graphical user interface

Description automatically generated

1. **Help and documentation –** Documentation is included in the submitted files.