

# COLLEGE OF ENGINEERING & BUILT ENVIRONMENT

# SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

# COMP 4604 GUI Design & DB Connectivity DT080/B/4

# Online Sports Utility Design Document

Submitted by

Leonard Benny, Jonathan Kaniki and Hassan Handal

Lecturer

Lejla Rovcanin

# Table of Contents

[COLLEGE OF ENGINEERING & BUILT ENVIRONMENT 0](#_Toc70515853)

[SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING 0](#_Toc70515854)

[COMP 4604 GUI Design & DB Connectivity DT080/B/4 0](#_Toc70515855)

[Online Sports Utility Design Document 0](#_Toc70515856)

[Table of Contents 1](#_Toc70515857)

[System Description 2](#_Toc70515858)

[Use Case Diagram: 3](#_Toc70515859)

[Database diagram 4](#_Toc70515860)

[Relationship Types 4](#_Toc70515861)

[ERD: 5](#_Toc70515862)

[Usability Heuristic 6](#_Toc70515863)

[Error Prevention 6](#_Toc70515864)

[Help and Documentation 6](#_Toc70515865)

[Aesthetic and Minimalist Design 6](#_Toc70515866)

[Flexibility and Efficiency 6](#_Toc70515867)

[User Control and Freedom 6](#_Toc70515868)

[Visibility of System Status 6](#_Toc70515869)

[Consistency and Standards 7](#_Toc70515870)

[Match between the System and the Real World 7](#_Toc70515871)

[Recognition rather than Recall 7](#_Toc70515872)

# System Description

Our database and GUI system are based on online sports utility shop. The user of the system will be able to perform the following tasks:

* Add/remove and maintain the sports utilities.
* Add/remove customers of the shop.
* Allow the users to rent sports utilities.
* Allow the users to return sports utilities.
* Allow the user to check overdue sports utilities rentals.
* Allow the user to print a list of the sports utility’s rental and inventory sports utilities.
* Allow staff to access to the inventory records.

When our applications start, our system gives option to login into it:

1. Customers
2. Shop Keepers

When Shop Keeper login to the system, he can register the customers by filling up the registration form.

Shop Keeper is responsible for adding up different items in his shop.

Registered customers will come up and see the display items and can do:

1. Purchase
2. Rent

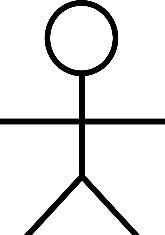
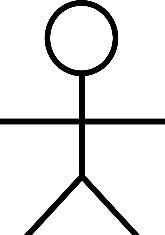
Customer can Rent or Purchase 1 item at the same time and Items which are overdue will be shown to Shop keeper side and he can monitor everything.

Shop keeper can delete, Update and Insert new customers and even Items.

**Tools and Technologies:**

1. JAVA
2. MS Access Database

# Use Case Diagram:

****

Customer

Shop Keeper

# Database diagram

## Relationship Types

* Customer to Rentals is a one to many relationship.
* Customer to Purchase is a one to many relationship.
* Rent to Item is a one to many relationship.
* Purchase to Item is a one to many relationship.
* ShopKeeper to SportsShop is a one to many relationship.
* SportsShop to Item is a many to one relationship

Note: These relationships always work vice versa

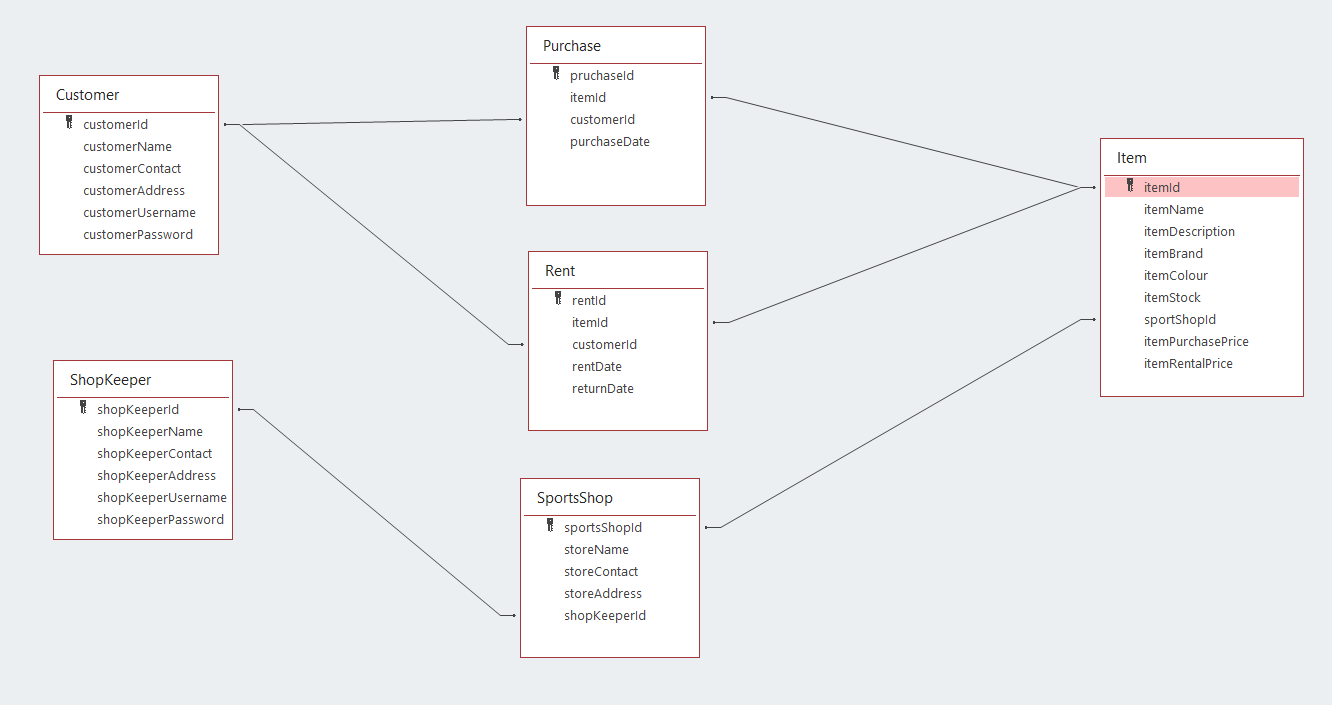


Figure 1: EER diagram

# ERD:

-> sportsShopId

-> storeName

-> storeContact

-> storeAddress

-> shopKeeperId

-> shopKeeperName

-> shopKeeperContact

-> shopKeeperAddress

-> shopKeeperEmail

-> shopKeeperPassword

Manage

Sports Shop

Shop Keeper

-> rentId

-> itemId

-> customerId

-> rentDate

-> returnDate

Has

Rent

-> itemId

-> itemName

-> itemDescription

-> itemBrand

-> itemColour

-> itemStock

-> shopKeeperId

-> itemPurchasePrice

-> itemRentalPrice

Will be

Items

-> customerId

-> customerName

-> customerContact

-> customerAddress

-> customerEmail

-> customerPassword

-> pruchaseId

-> itemId

-> customerId

-> purchaseDate

-> itemColour

-> returnDate

Will be

By

To

Customers

Purchase

**Figure 4: ERD**

# Usability Heuristic

Usability heuristic is an inspection method used by computer software to determine any usability issues within the user interface design. It commonly requires evaluators investigating the interface and judging its compliance with the usability heuristic.

The ten concepts that make up the usability heuristic are as follows::

## Error Prevention

Within the 10 principles, error prevention is arguably the most important feature in the user interface. This can be established by creating a design that prevent problems from occurring. We ensured error prevention by eliminating error prone conditions and embedding error checks with the system before an error is executed by the user.

In the case that a user makes an error, the user will be prompted with an error message detailing exactly what the error is and the solution.

## Help and Documentation

Documentation and help are often considered optionally within the principles. However, it may be required to provide some form of documentation and help. The documentation and help must be easy to search for, easy to understand and focused on the user’s task. The documentation and help should include conclusive instructions to be performed by the user. End users can easily utilize the functionality of the application.

## Aesthetic and Minimalist Design

Within the principle, it is vital that we create a pleasing aesthetic and also have minimalist design. The design should appeal to the user and all dialogue should contain only the relevant information. The design should only contain exactly what is required and no extra irrelevant information.

## Flexibility and Efficiency

Our program consumes less memory space, utilizes the least amount of processing time and the workload of the CPU, thus creating a more efficient system.

Our program is highly flexible as we can use the code across other platforms. The program can handle changes without having to rewrite the entire program.

## User Control and Freedom

Within our system, the user has complete control and freedom. The user submits queries using SQL to query the database. Within our system, the user is able to undo/redo and correct any mistakes which may have occurred by their own doing.

## Visibility of System Status

This principle states that it is vital that we keep the user informed on everything that’s occurring. We have implemented this principle into our program by providing our users with appropriate feedback and information.

## Consistency and Standards

The principle states that within our system, we should have text all the font and size the same. The user shouldn’t have to doubt whether the actions, words and context all mean what they’re meant to mean.

## Match between the System and the Real World

Our system is compatible with the real world in regard to the user’s concepts, language and phrases. Our system minors feature that user is familiar to. This principle is vital as the user will find the system understandable.

## Recognition rather than Recall

The program is written in such a way that makes other users to follow the logic of the program without much effort. It has a user-friendly approach which helps in processing the database application.