IFS 242 Final Assignment

Due Date : 2 June 2023, 18:00

Name : Zuhar Regal, 4227160

Lecturer : Grant Hearn

Database and Web Application Design for the Shoebox

The Development Process

Problem

1. The Problem Statement : (Analysis Phase)
    - Overview
    - Requirements
    - Entities
2. Use Case Diagrams : (Analysis + Design Phase)
    - Use Case Diagrams
    - Data Entities
    - Database Tables

Solution

3. Entity Related Diagrams : (Design Phase)
4. Database Implementation : (Implementation phase)
    - DDL Queries

---

1. Problem Statement

Overview

The Shoebox is a retail online store that focuses on offering a selection of footwear. The web application developed will rely on a database system that effectively manages information related to the shoes sold, and the orders made by registered customers. Moreover, the database will also encompass customer records, ensuring that only registered individuals can proceed with placing orders.
We have been tasked with creating a database and web application for The Shoebox, an online store specializing in the sale of shoes.

Requirements

Requirements of the Database:

- In terms of product information, each shoe in the database should have a name, price, brand, color, material, description, and an associated picture. This set of attributes provides customers with detailed information about the shoes, assisting them in purchasing the shoes.
- Regarding customer data, the database should store the customer's name, surname, email address, delivery address, cell phone number, and South African ID number. These details enable efficient order-processing as each order is linked to the customer’s email address.
- Each order made by a customer should be stored. The database should record the customer who made the order, as well as the status of the order.
- Each shoe item that has been purchased, for each order, will be stored.

Requirements of the web application:

- A **Registration page**, where customers can register and insert their name, last name, email address, delivery address, cell phone number, and S.A. ID number.Registration should be mandatory for making a purchase.
- A **Registration processing page** that will send the customers information to the database and report to the customer on the success of the registration. Customers must first register on the web application before placing an order. Each order placed from the product page must include the customer's email address which serves as a unique identifier, linking the order to a registered customer in our database. Upon order submission, the web application should process the order if the customer is registered or reject it if the customer is not registered.
- A **Home page** which shows what the shoebox is about.
- An **Orders page**, where customers can view and select the products they want to purchase.
- After a customer has placed an order, they should be directed to the **Ordering Accepting page** where the order is then inserted into the database. This page will display the customer's name,the order price, including the value-added tax (VAT) and where the product will be delivered to.

An employee should be able to:

- Maintain records of the registered customers, including their name, email address, delivery address, cell phone number, and SA ID number.
- Maintain information regarding the shoes that are sold.
- Maintain the orders placed by different customers as well as the status of the order.
- Maintain the shoes that are purchased in each order.
- Access a page or functionality that will enable them to view all the unfilled orders that have been placed by customers.

A customer should be able to:

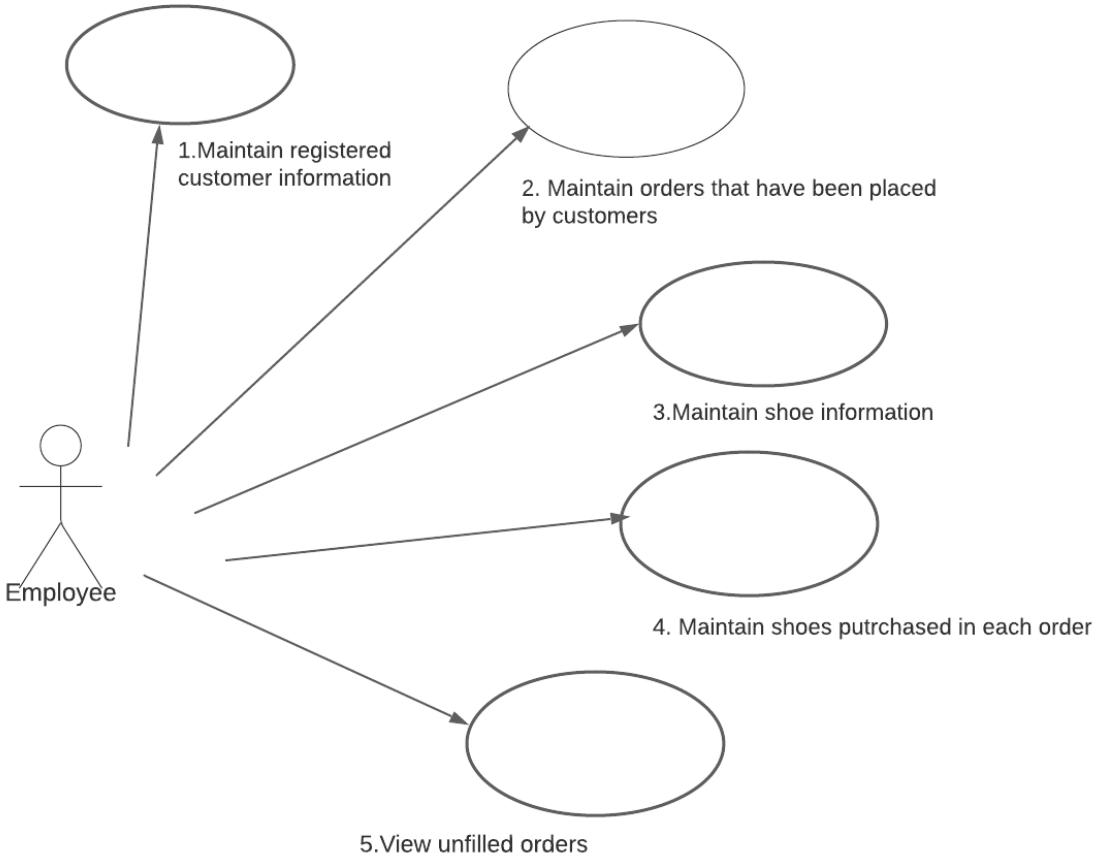
- View the shoes available for sale: see a list of shoes, along with their details (name, price, brand, color, material, description, and an associated picture) in the orders page.
- Select one or more shoes: choose one or more shoes they want to purchase.
- Add selected shoes to the order: select one or more pairs of shoes and add to their order.
- Provide personal information: enter their personal information, including first and last names, email address, delivery address, cell phone number, and S.A. ID number.
- Register as a customer: register on the website using the provided personal information.
- Receive registration confirmation: after registering, the customer should be directed to the process registration page which will indicate whether they are registered or whether they have to try again.
- Place the order: submit their order by inserting their email.
- Receive order confirmation: after submitting the order, the customer should receive a message indicating whether or not their order has been processed or rejected. Unregistered customers should have their order rejected by the system. Registered customers should be directed to the order accepting page which shows their order summary.
- View order summary: view the list of shoes selected in their order.

Entities

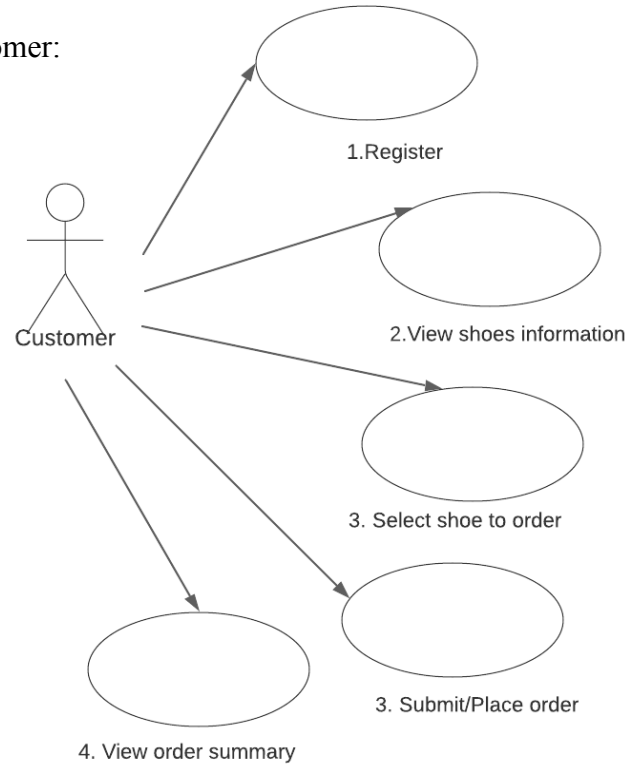
- Customers: this entity represents customers who have registered on the website.
- Shoes: this entity represents the shoes available for purchase.
- Orders: this entity represents the order a customer has placed on the web application..

2. Use Case Diagrams

Employee:


1.Maintain registered customer information
2. Maintain orders that have been placed by customers
3.Maintain shoe information
Employee
4. Maintain shoes putrchased in each order
5.View unfilled orders

Customer:


1.Register
Customer
2.View shoes information
3. Select shoe to order
3. Submit/Place order
4. View order summary

Data Entities

- Customers: this entity represents customers who have registered on the website. It includes attributes such as customer ID, name, surname, email address, delivery address, cell phone number, and South African ID number.
- Shoes: this entity represents the shoes available for purchase. It includes attributes such as shoe ID, name, price, brand, color, material, description, and an associated image.
- Orders: this entity represents the orders a customer has placed.. It includes attributes such as order ID, customer ID (linking to the Customers entity) and order status, which indicates whether an order is filled or unfilled.
- Order_items: this entity represents the individual shoes within an order. It includes attributes such as shoe ID (linking to the Shoes entity), and order ID (linking to the Orders entity).

Database tables

Database tables for customers, shoes, orders and order_items:

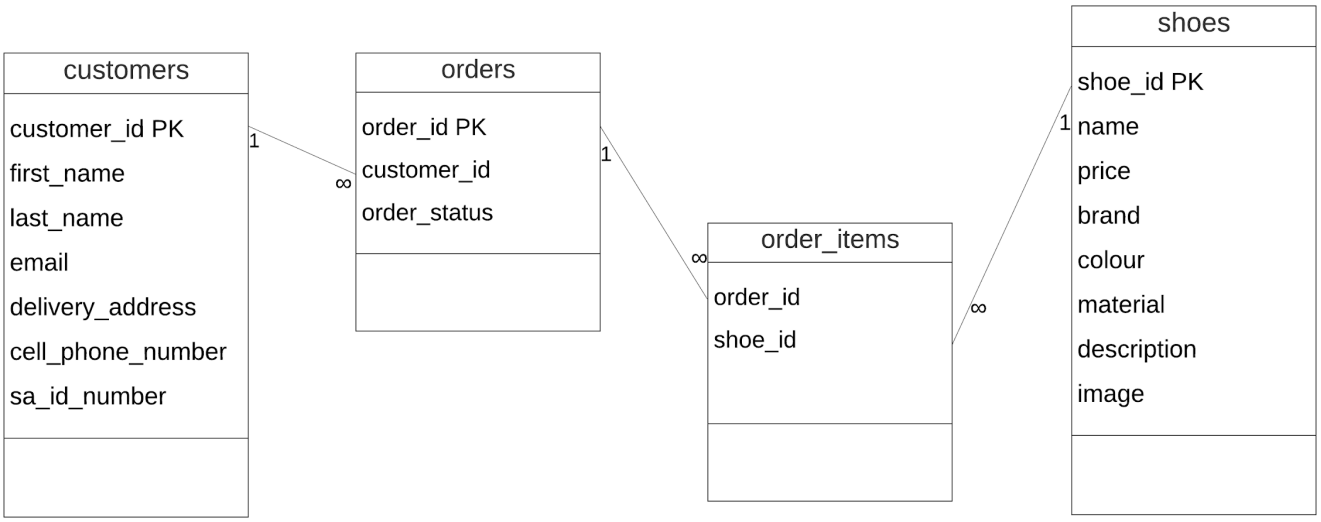
orders	
order_id	*PK INT(11) NOT NULL*
customer_id	*FK INT(11) NOT NULL*
order_status	*VARCHAR(15) NOT NULL*

customers	
customer_id	PK INT(11) NOT NULL
first_name	VARCHAR(50) NOT NULL
last_name	VARCHAR(50) NOT NULL
email	VARCHAR(70) NOT NULL
delivery_address	VARCHAR(100) NOT NULL
cell_phone_number	VARCHAR(15) NOT NULL
sa_id_number	CHAR(13) NOT NULL

shoes	
shoe_id	PK INT(11) NOT NULL
name	VARCHAR(50) NOT NULL
price	DECIMAL (8,2) NOT NULL
brand	VARCHAR(30) NOT NULL
colour	VARCHAR(20) NOT NULL
material	VARCHAR(30) NOT NULL
description	TEXT NOT NULL
image	VARCHAR(50) NOT NULL

order_items	
order_id	FK INT(11) NOT NULL
shoe_id	FK INT(11) NOT NULL

3. Entity Related Diagram



4. Database Implementation

DDL Queries

```
CREATE DATABASE the_shoebox;

CREATE TABLE shoes (
  shoe_id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  name VARCHAR(50) NOT NULL,
  price DECIMAL(8, 2) NOT NULL,
  brand VARCHAR(30) NOT NULL,
  colour VARCHAR(20) NOT NULL,
  material VARCHAR(30) NOT NULL,
  description TEXT NOT NULL,
  image VARCHAR(50) NOT NULL
);
```

```
CREATE TABLE customers (
  customer_id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50) NOT NULL,
  email VARCHAR(70) NOT NULL,
  delivery_address VARCHAR(100) NOT NULL,
  cell_phone_number VARCHAR(15) NOT NULL,
  sa_id_number CHAR(13) NOT NULL
);

CREATE TABLE orders (
  order_id INT PRIMARY KEY AUTO_INCREMENT NOT NULL,
  customer_id INT NOT NULL,
  order_status VARCHAR(15)
);

CREATE TABLE order_items (
  order_id INT NOT NULL,
  shoe_id INT NOT NULL,
);

CREATE USER 'ifs242user'@'localhost' IDENTIFIED BY 'password';
GRANT SELECT,INSERT,UPDATE,DELETE ON the_shoebox.* TO 'ifs242user'@'localhost';
```

END