**Design and Implementation of E-Commerce Site for Online Shopping**



**Design and Implementation of E-Commerce Site for Online Shopping**

**By**

**Xitshembiso Sharlot Baloyi**

**Learnership Program (My First E-Commerce Learnership)**

**Geeks4Learing 2022**

**Design and Implementation of E-Commerce Site for Online Shopping**

Our Project is An E-Commerce Website. The purpose of this Online shopping website is to facilitate the customers to buy products online with ease of clicking in the website page and buy the product of your choice without going to the store it saves time.

**Introduction**

Online shopping is form of electronic commerce which allows customer to directly buy goods or services from a seller over the internet using a web browser.

**Project Overview**

This project provides the Website for ONLINE SHOPPING. The main objective of this website is to provide easy assistance to both the customer and merchant to interact with each other for business purpose through a platform. It provides a system which controls the selling of items and user can buy items sitting anywhere and anytime. The system is user friendly and easy to use. All the data is store in database and avoids any miscalculations.

**Description**

* Any member can register and view available products.
* Only registered member can purchase multiple products regardless of quantity.
* Contact Us page is available to contact Admin for queries.

**There are two roles available: Visitor and User**

* Visitor can view available products.
* User can view and purchase products.

**Name and Logo of the Website:**

**Sharlot Laptop Shop**

**We have the following product**

1. Computers
2. Laptops

**Technology Used:**

1. **Front End:**

* C#
* ANGULAR
* CSS
* HTML/HTML5
* SQL SERVER
* BOOTSTRAP FOR FRAMEWORK

1. **Back End:**

* LUCIDCHARTS FOR DIAGRAMS
* VISIUAL STUDIO CODE/VISIUAL STUDIO
* SQL Server FOR DATABASE STORAGE

**User Stories**

As a customer, I need to authenticate myself so that I can see my account details and past orders.

As a customer searching for a product, I need to be presented the most appropriate choices, so that I am likely to find what I am looking for.

As a customer trying to register for an account, I need to be informed clearly if I am making any errors, so that I can fix them quickly.

As a customer about to make a purchase, I need to be able to submit my credit card details and get authorization so that I can complete my checkout.

**Successful authentication**

When the user enters the correct email address and password and selects “sign-in”

Then route the user to My Account home page and display signed-in status on header

**Unsuccessful authentication**

When the user enters an incorrect combination of email address and password and selects “sign-in”

Then reset credentials fields and display an error message “Incorrect Credentials”

**Forgot credentials**

When the user selects the “forgot credentials”

Then route the user to the “forgot credentials” page

**Register**

When the user selects the “register”

Then route the user to the “registration” page

**Overview**

**Authentication:** User can signup/login/logout

**Items for sale:** User can view lists of items that we have on our website.

**Shopping Cart:** User can add items to shopping cart and the app remembers it next time you login. User can view all the items in their shopping cart. User can delete items in the shopping cart. Shopping cart uses an integer column to store "state".

**Checkout:** User can fill in form and submit billing info. After submitting billing info, items in the shopping cart will move to a different "state".

**Functionality**

* Register
* Login
* View Products on the lists
* Add to Cart Page
* Check out when you are Done

**The Modules used:**

**New User:** this module is for users who don’t have account, here user is allowed to create an account to login. The account creation is done by filling the registration form with user details such as Name, phone no, email Address etc.

**Login:** this module has a login Button where you must enter Username and Password, if the details provided are correct then it is directed to the next page.

**Product:** this module has information regarding Computer and Laptops such as its name, model, colour, price features etc. The user can only view the product, add to cart only those in stock.

**Search:** helps the customer to search based on his budget or interest. The search can be done on different categories like model name, model number, colour, price

**Cart: User** can select any number of computers or laptops and add to cart. He can also remove from the cart if he dislikes it later.

**Payments:** this describes the payment done by the customer. the payment information can include information like the model purchased, quantity, payment methods.

**Non-Functional / Operational Requirements**

**Security**

Pages of the website must be access in the way they were intended to be accessed.

**Efficiency and Maintainability**

Page loads should be returned and formatted in a timely fashion depending on the request being made

**Future Scope of Our Project:**

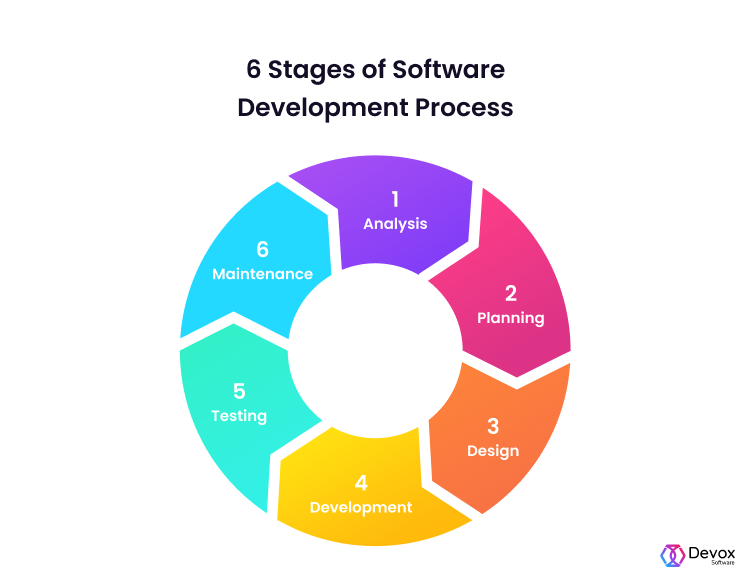
Purchasing and selling products and services over the internet without the need of going physically to the market is what online shopping all about that is the main objective of our project. In addition to this, a lot more options are available when shopping is done online. Our customer will get a lot of information on the pricing and quality of products. The online transactions are more secure our system will ensure that.

**Software Development Life Cycle (SDLC)**

**What is SDLC?**

SDLC (software development lifecycle) is the process that a software project follows, and which consists of a detailed plan describing how to develop, maintain, replace, change or improve specific software. The lifecycle defines a methodology for improving software quality and the overall development process.

The software development life cycle (SDLC) is a process used for structuring the development of any software system, from initiation through to implementation. An increase in demand for software to meet customer needs effectively but with less cost and faster delivery has put tremendous pressure on modern organizations.

**1. Analysis**

During this software development lifecycle phase, the specialists meticulously collect precise requirements from the customer to present a solution fine-tuned to their needs. Any unclarities must be elucidated in this stage only.

The analysis phase also gathers business requirements and identifies any potential risks. This step in SDLC also includes a feasibility study, which defines all fortes and weak points of the project to assess the overall project viability.

**Analysis**

* Gather the business requirements for the system

1. **Planning**

The purpose of the second stage is to outline the scope of the problem and identify solutions. Resources, costs, time, and other aspects should be considered here. The planning phase of the SDLC is also when the project plan is developed that identifies, prioritizes, and assigns the tasks and resources required to build the structure for a project

**Planning**

* Define the system to be developed
* Set the project scope
* Develop projects plan including tasks ,resources and timeframes

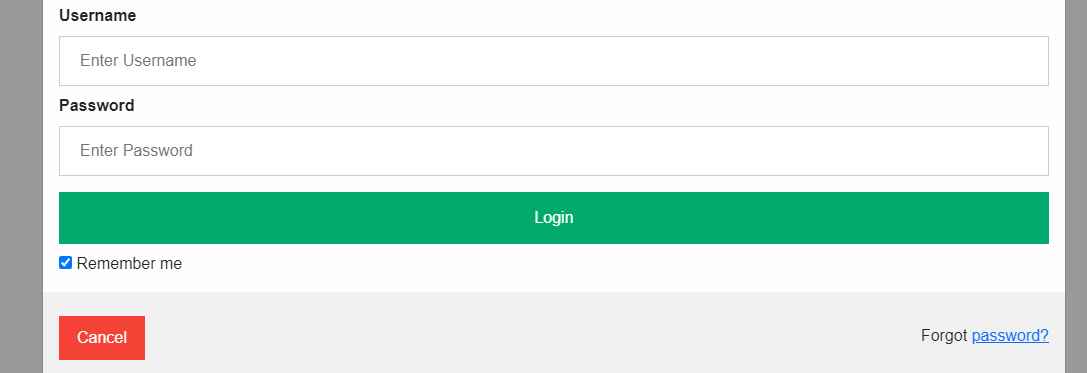
**4.Design**

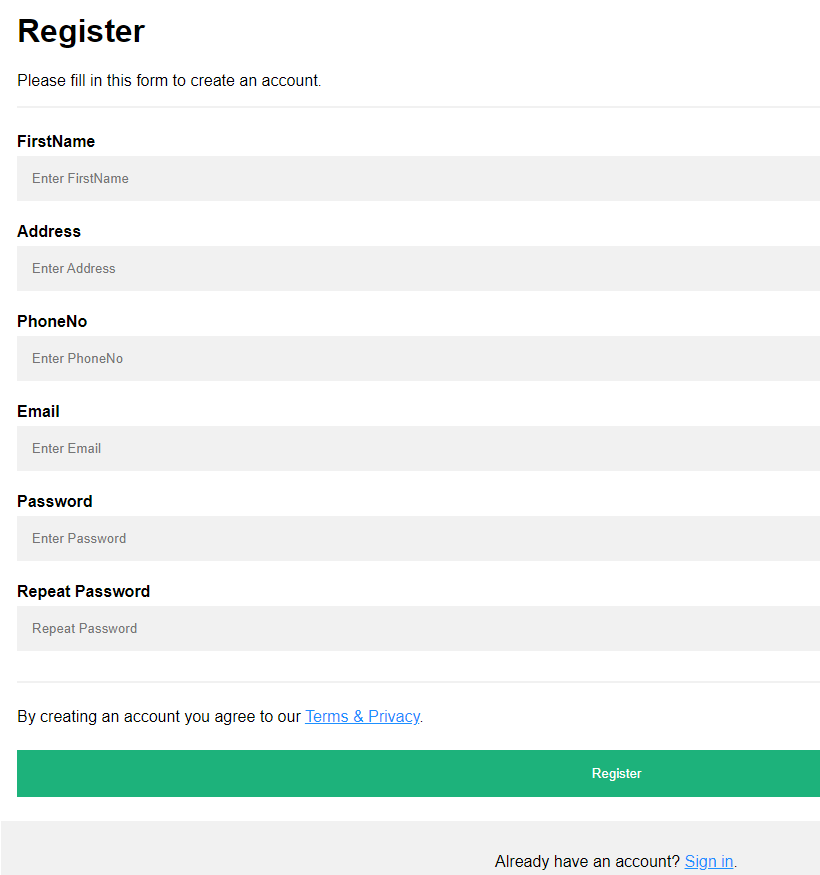
The third phase entails two further steps – High-Level Design (HLD) and Low-Level Design (LLD). The result of the former is the future architecture of a software product, whereas the LLD step describes how each and every feature in the product should work. It’s also in this phase when the database specification is developed to decide on data management and storage for future processing, retrieval, or evaluation.

**Design**

* Design the technical architecture required to support the system
* Write the test cases and design the system models

**LOGIN**





**4. Development**

Software development turns your project’s requirements and prototypes into a tangible solution. Thus, engineers start creating the entire system by crafting code using the required technology.

During this software development lifecycle phase, clients will be able to have a first look at your future product. And by the end of the building process, clients will have an operating feature to share with the customers.

**Development**

* Build the technical architecture
* Build the database and programs

1. **Testing**

In the fifth stage, all the pieces of code are tested to verify and validate a software product. Testers then perform Software Testing Life Cycle activities to monitor the system for bugs, and defects. This is done to check the correspondence between the real and expected behaviour of a program.

The testing stage and the initial SDLC phases can be performed with both internal software development and outsourcing, as they require end-user interaction.

**Testing**

* Perform the testing of the system

**Implementation**

* Install system
* Write detailed user documentation
* Provide training for system users

**6. Maintenance**

Once the system is deployed, any necessary upgrades, enhancements, and changes can be made, implementing new features into the operating software. It is crucial to maintain and modernize the system regularly so it can adapt to future needs.

**Maintenance**

* Build a help desk to support the system users
* Provide an environment to support the system users

**Oop Principles**

**Encapsulation**

Encapsulation is the mechanism of hiding of data implementation by restricting access to public methods. Instance variables are kept private and accessor methods are made public to achieve this. For example, we are hiding the name and dob attributes of person class in the below code snippet. Encapsulation — private instance variable and public accessor methods.

**Abstraction**

Abstract means a concept or an idea which is not associated with any instance. Using abstract class/Interface we express the intent of the class rather than the actual implementation. In a way, one class should not know the inner details of another in order to use it, just knowing the interfaces should be good enough.

**Inheritance**

Inheritances expresses “is-a” and/or “has-a” relationship between two objects. Using Inheritance, In derived classes we can reuse the code of existing super classes. In Java, concept of “is-a” is based on class inheritance (using extends) or interface implementation

**Polymorphism**

It means one name many forms. It is further of two types — static and dynamic. Static polymorphism is achieved using method overloading and dynamic polymorphism using method overriding. It is closely related to inheritance. We can write a code that works on the superclass, and it will work with any subclass type as well.

**SOLID: The First 5 Principles of Object-Oriented Design**

**Introduction**

**SOLID is an acronym** for the first five object-oriented design (OOD) principles. These principles establish practices that lend to developing software with considerations for maintaining and extending as the project grows. Adopting these practices can also contribute to avoiding code smells, refactoring code, and Agile or Adaptive software development.

**SOLID stands for:**

• S - Single-responsibility Principle

• O - Open-closed Principle

• L - Liskov Substitution Principle

• I - Interface Segregation Principle

• D - Dependency Inversion Principle

**The Solid Principles of the object-oriented design:**

**Single responsibility**

An object class should only be responsible for one specific function, and only have one reason to change.

**Open/Closed**

Developer should be able to add new features, functions and extensions to a class while leaving the rest of the existing codebase intact.

**Liskov substitution**

The objects contained in a subclass must exhibit the same behaviour as any higher- level superclass it depends on.

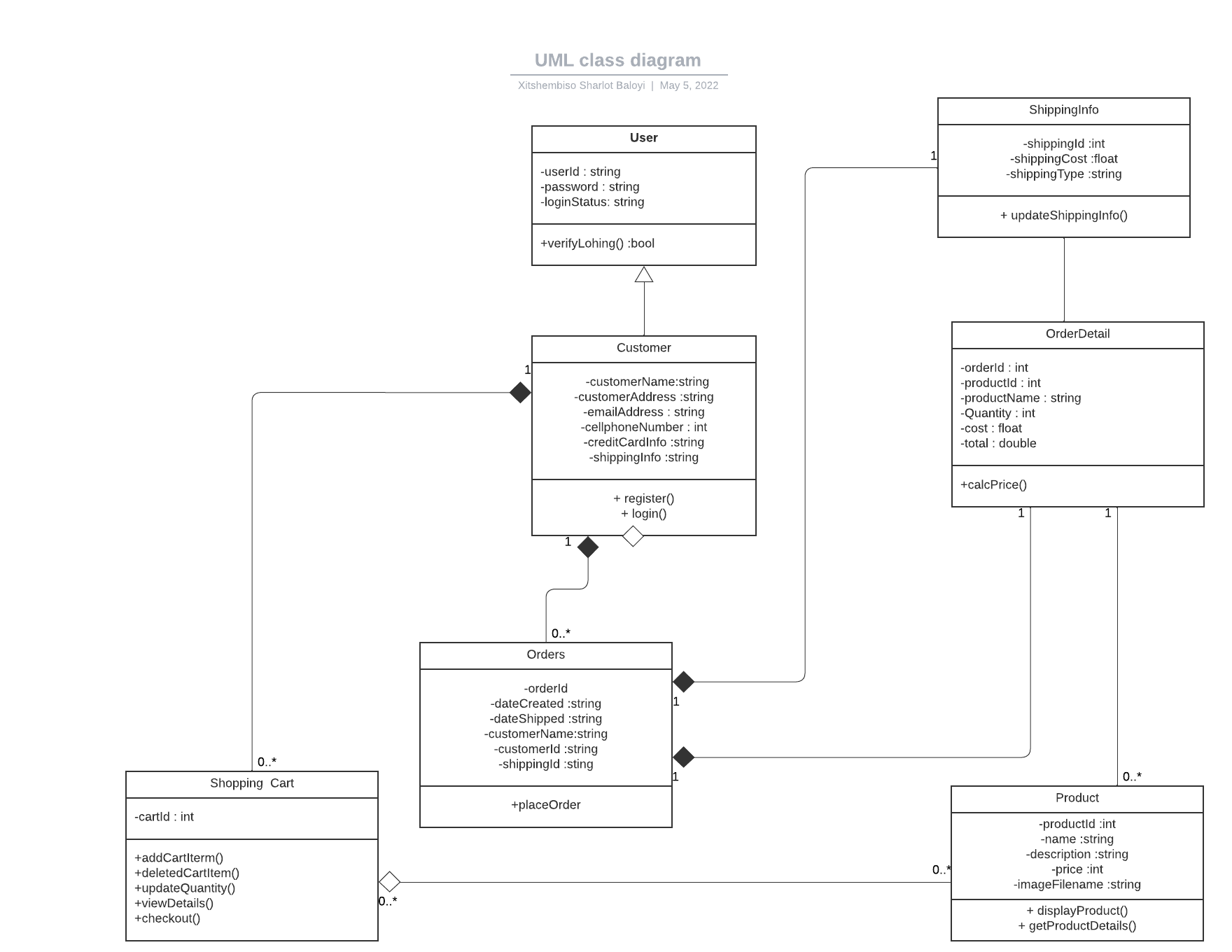
**Interface segregation**

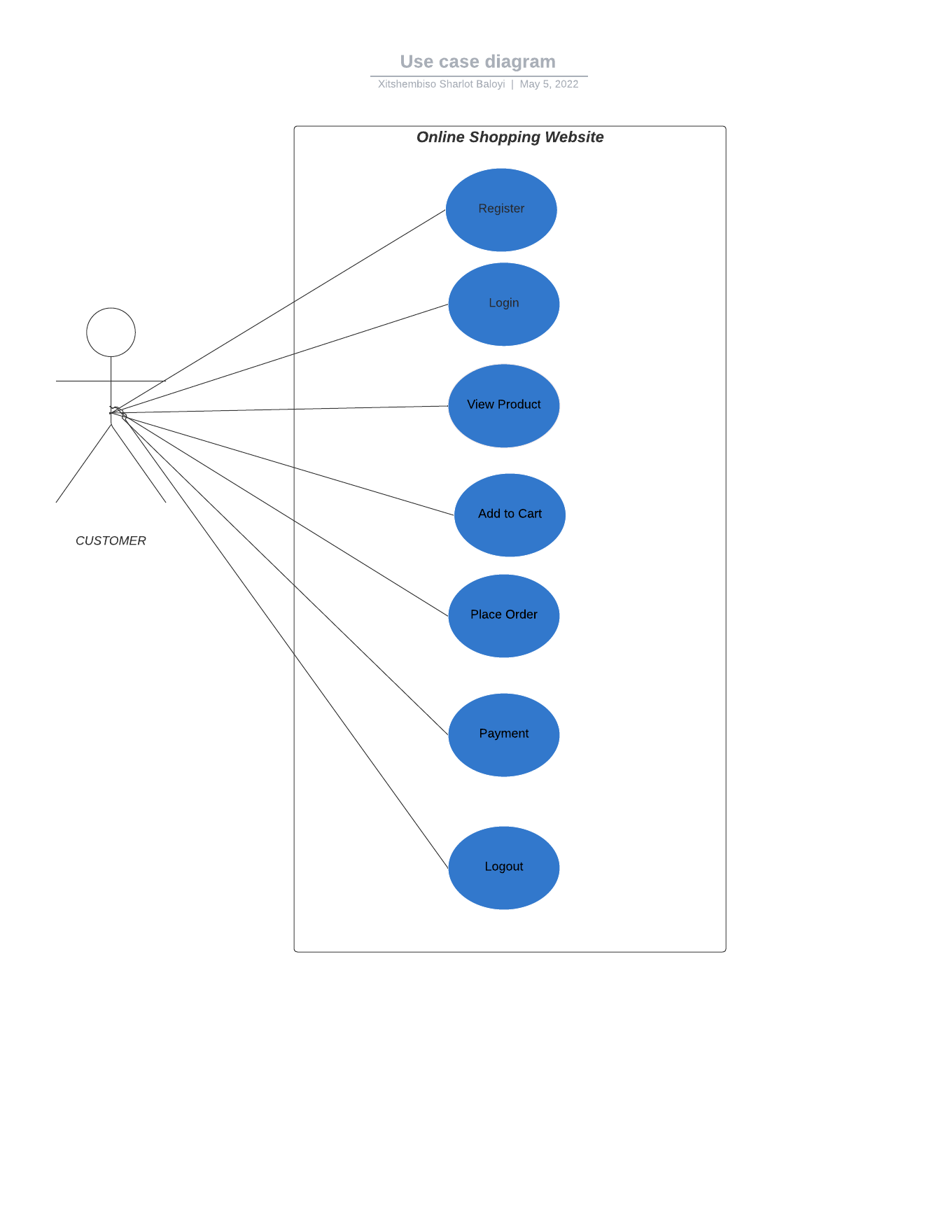
Create a separate client interface for each class within an application, even if those classes share some of the same methods.

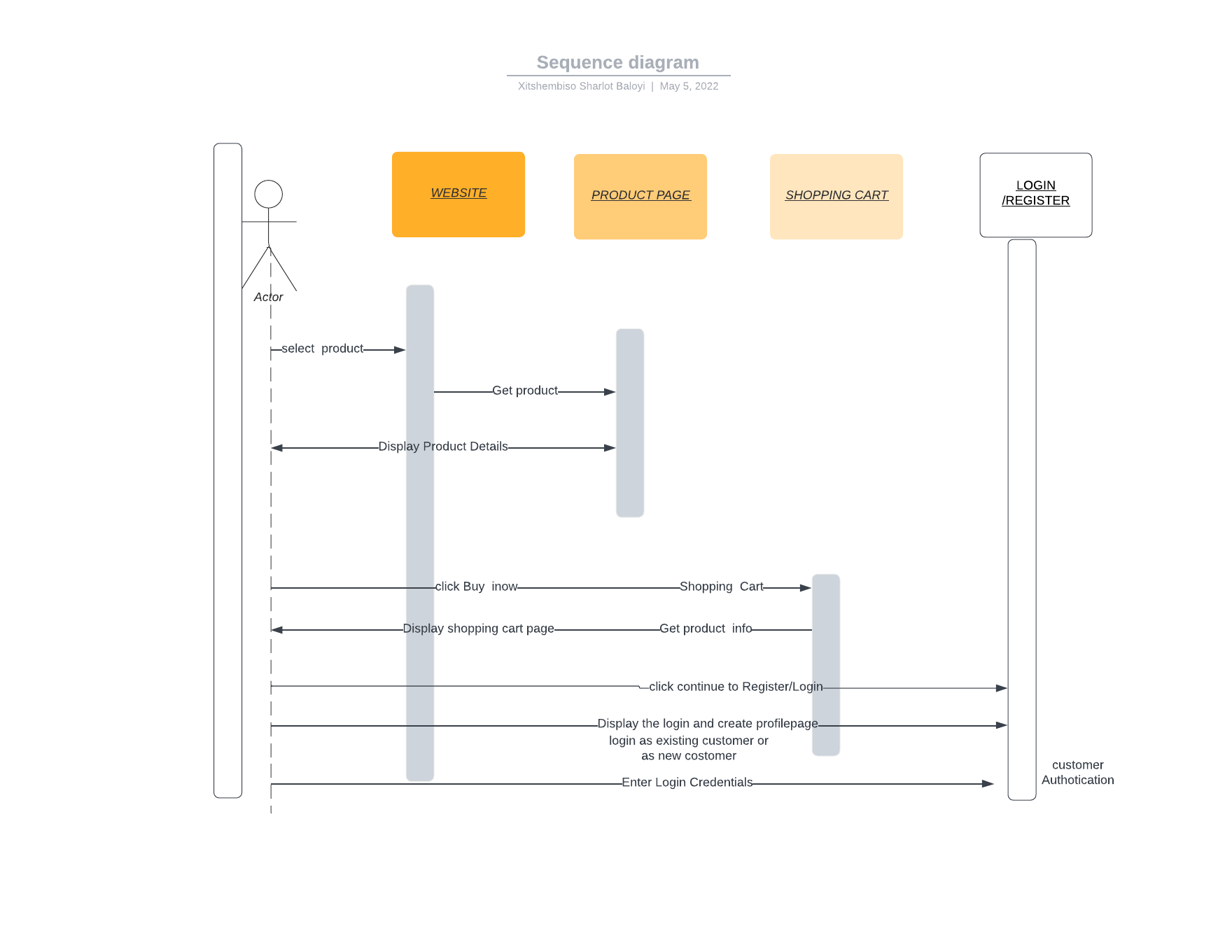
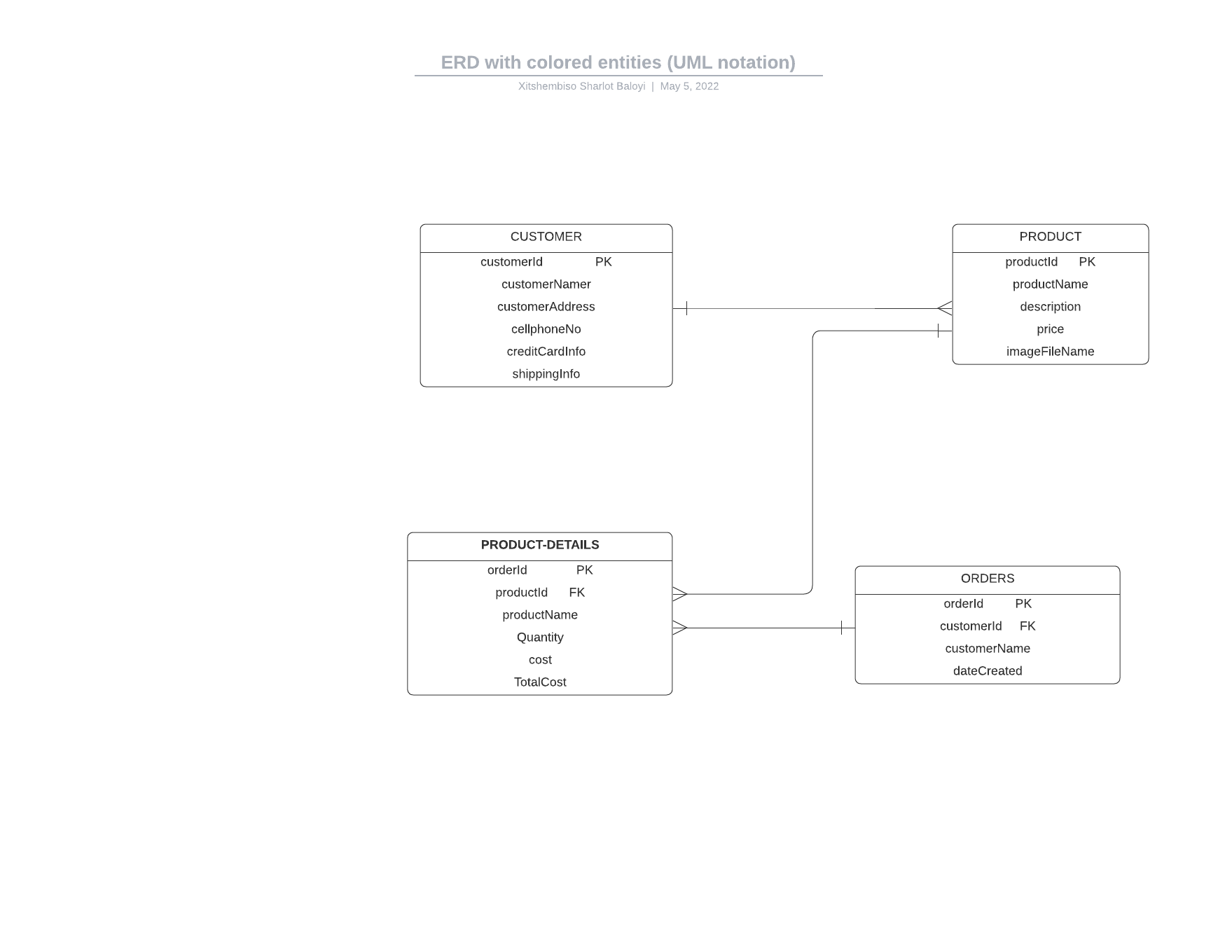
**Dependency Inversion**

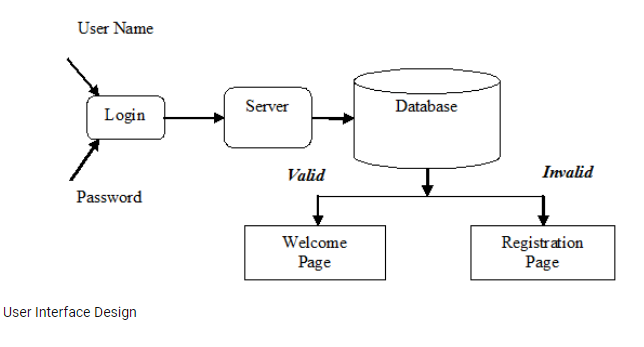
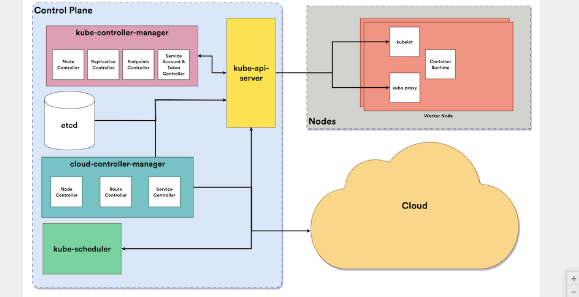
When a subclass is dependent on a superior class, the higher-level class should not be affected by any changes made to the subclass.

**DAIGRAMS**









**Data flow Diagram**

