Software Engineering and Testing. BSC Year 2, 2020/2021
(Assignment 3 - 20%)

Assessment 3: Design and Draft Implementation

Submitted by: Jake Callow B00152286 / Lukas Stoskus B00152862 / Callum Breen B00152224

Submission date

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: Jake Callow Dated: 17/03/2024

Author: Lukas Stoskus Dated: 17/03/2024

Author: Callum Breen Dated: 17/03/2024

Table of Contents

Contents

Title: Beans and Machines Coffee Supply Store 4

Abstract / Executive Summary 4

1. Project Definitions 4
2. Document Revision 4
3. Methodology 5
4. Requirements 5
5. Case Diagrams 6
6. Conclusions 8

Title: Beans and Machines Coffee Supply Store

Abstract / Executive Summary

This project is an online café supply store for a company called Beans and Machines that wants to expand its product reach. The website will include a list display of products connected to a database, login functionality for the user to store payment information and products in a shopping cart, and a payment page to confirm product orders and store them in the database. The pages will have a clean design and friendly uncomplicated UI for ease of access to streamline the process.

1. Project Definitions

- The purpose of this document is to revise our project requirements and use cases to review how much progress our team has made.
- The project is a website for an online café supply store called Beans and Machines, that will sell a wide range of products from coffee beans to machinery.
- The functional specifications of this project are a connected database to display products and product info, as well as store user authentication details. The website will also include a shopping cart to store products users have selected, and a payment page to read and store payment information. Finally, it will include database storage for delivery details, and a confirmation page when the user has confirmed payment.
- The main components of the software system are a database to store information, and classes and objects to store more temporary information.

2. Document Revision

Rev. 1.0: 06/03/24 – initial version, explained project definitions and methodology.
Rev. 2.0: 13/03/24 – updated methodology and added class diagram and ERD.
Rev. 3.0: 15/03/24 – changed class diagram and added descriptions for all diagrams.
Rev. 4.0: 18/03/24 – final review of document and addition of conclusion.

3. Methodology

We're using a Use Case diagram and Class diagram to demonstrate the model of the system we are building. We also use a wireframe to get an idea of the website's UI.

OOAD is necessary for creating objects within our code to store instances of information such as products within the shopping cart and user login information.

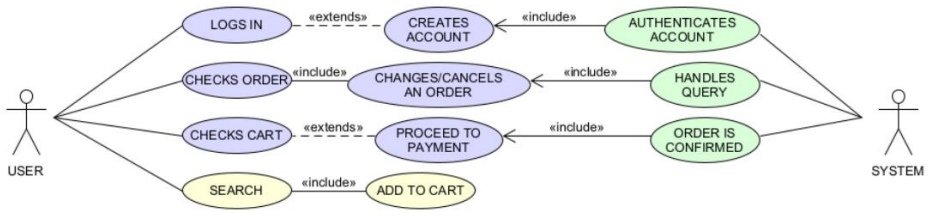
Classes are required for features such as the shopping cart for storing information that is not relevant to store within the database itself, which is an example of volatile storage as it is intended to be temporary and relies on the user to exist.

We're using an ERD to demonstrate how data is being stored within our database.

We chose a simple and clean design for our UI to put an emphasis on the products that the page will be selling.

4. Requirements

4.1 Use Cases:



The user will be able to log in or create an account, search the products and add products to their cart. The user can then access the cart, and if they are logged in, proceed to a payment page to enter payment details. If the user confirms payment, they will receive a confirmation of their order that lists the products to them.

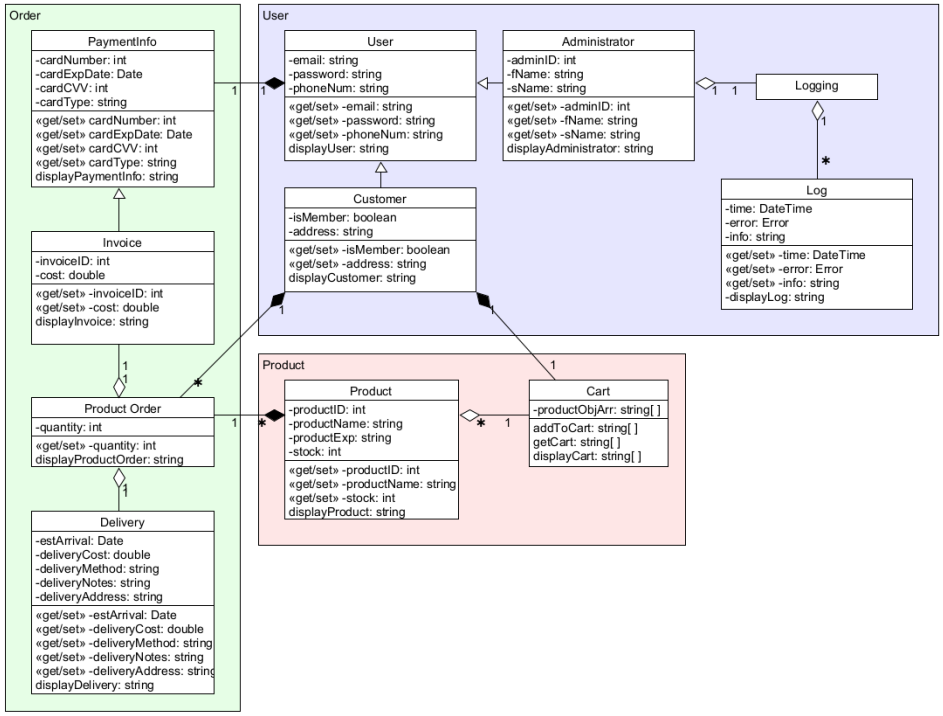
The system administrator will be able to log in to their administrator account and edit or remove product entries. They will also be able to view user product order details.

4.2 Use Case Specifications

A login system and storage will require sufficient tables in the database. The database will also be required to store product information and stock to display on the pages. Classes and objects will be required to store the users shopping cart and login information. Payment details will be stored inside the database.

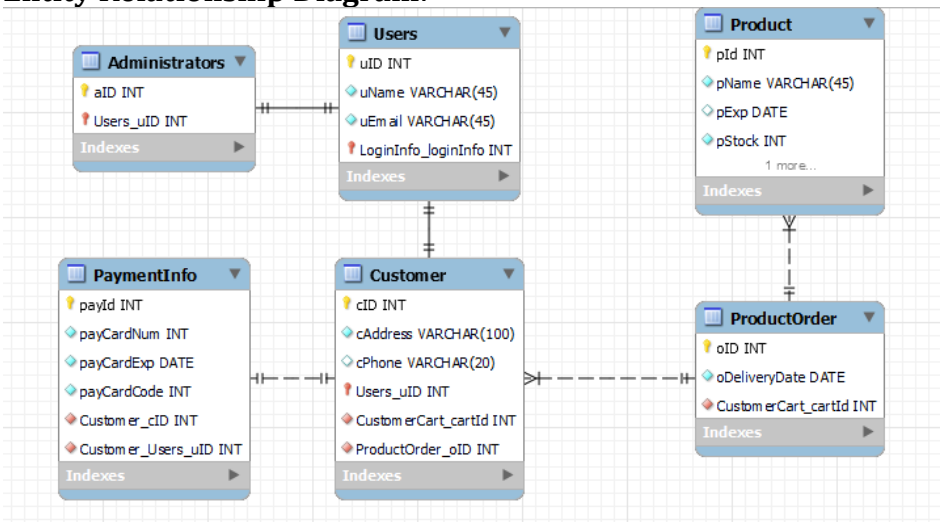
5. Case Diagrams

Class Diagram:



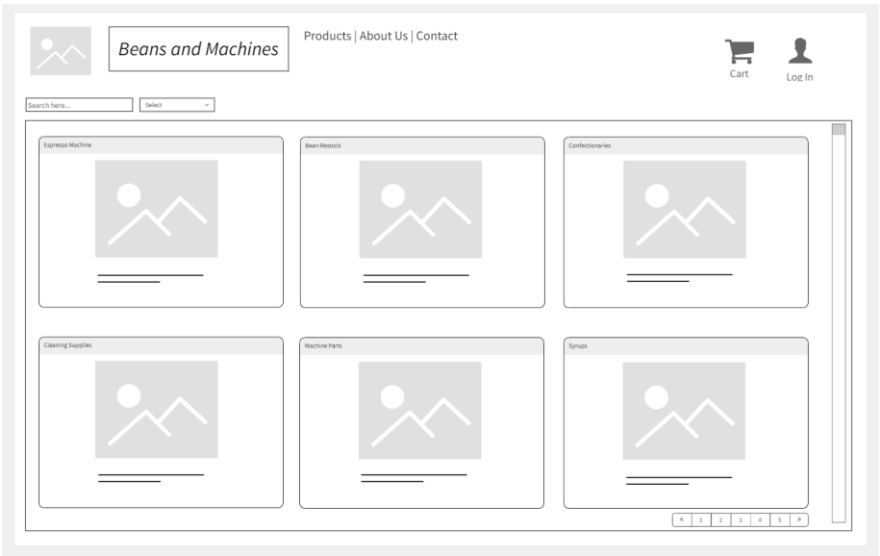
The user class is abstract, and the customer and administrator classes extend from it as they share similar attributes. Cart relies on customer to exist, and somewhat relies on product. A log requires logging, and logging must be done by an administrator. A product order cannot exist without both a product and a customer. Delivery and invoice both somewhat rely on product order, and payment info extends from invoice.

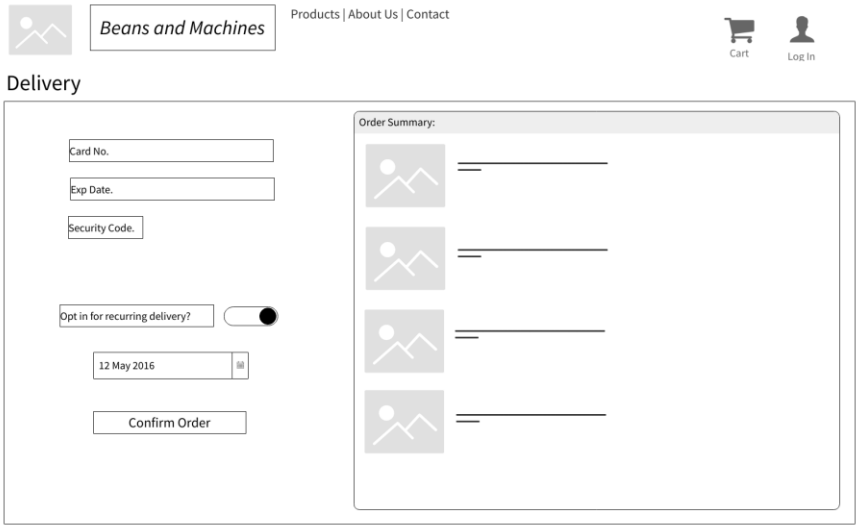
Entity Relationship Diagram:

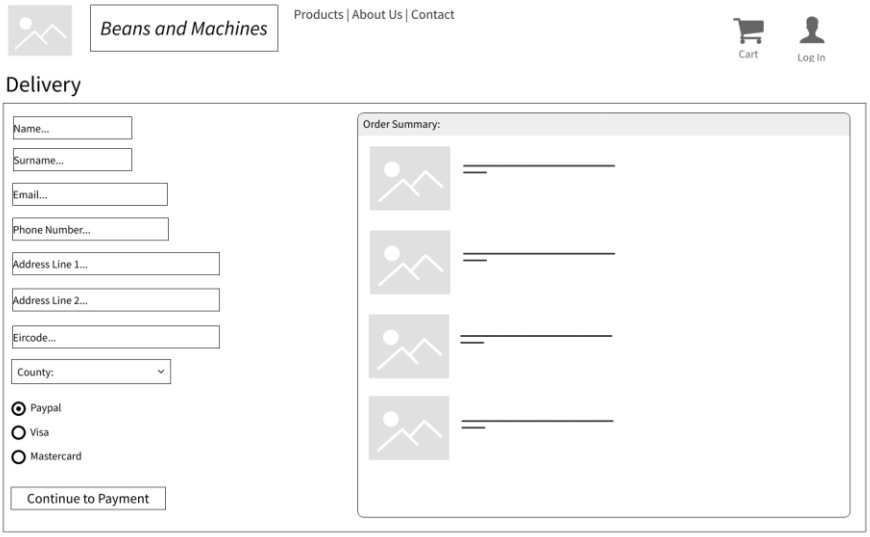


These are all the required tables for the database, anything shown in the class diagram that is not present here will have classes or objects instead.

Wireframes:


Beans and Machines
Products | About Us | Contact
Cart
Log In
Search here...
Select
Espresso Machine
Bean Restock
Confectionaries
Cleaning Supplies
Machine Parts
Syrups


Beans and Machines
Products | About Us | Contact
Cart
Log In
Delivery
Card No.
Exp Date.
Security Code.
Opt in for recurring delivery?
12 May 2016
Confirm Order
Order Summary:


Beans and Machines
Products | About Us | Contact
Cart
Log In
Delivery
Name...
Surname...
Email...
Phone Number...
Address Line 1...
Address Line 2...
Eircode...
County:
Paypal
Visa
Mastercard
Continue to Payment
Order Summary:

6. Conclusions

The project has progressed quite well with all of the required functionalities being either fully or partially implemented. The UI of the website similarly matches the planned wireframes. The website can correctly read and write information to the database and display it on the page.

Checklist: Is your document complete and correct?

Content:

- Does the design include all requirements from the customers' needs
- Are you satisfied with all parts of the document?
- Do you believe all parts have been implemented?
- Have you explained your methodology and design choices?
- Have you clearly articulated your understanding of the purpose of all diagrams created ?
- What are these diagrams? Why you need them? How were they developed?
- Is each part of the document in agreement with all other parts?
- Does the design create a solution for the initial proposal?

Completeness:

- Are all the necessary components specified?
- Are the design specifications precise enough?
- Are all sections from the document template included – if changed, why?

Clarity:

- Is the design reasonable?
- Is the level of details for each design section appropriate?
- Is the design written in a language appropriate to the intended audience of software engineering teams?
- Are all items clear and unambiguous?