Cairo University  
Faculty of Computers and Artificial Intelligence



**CS251**

**Introduction to Software Engineering**

YAO

Software Design Specifications

Version 0.0

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20210502 | Yusuf Elsayed Abdelrahman Badr | [yusufbadr@yahoo.com](mailto:yusufbadr@yahoo.com) | 01063033085 |
| 20210251 | Alialdin Muhammad mostafa | Alialdin.mohamad@gmail.com | 01120765911 |
| 20210060 | Osama maher masoued | Osama392maher@gmail.com | 01020083229 |

May 2023

Contents

[Instructions [To be removed] 3](#_Toc133519910)

[Team 3](#_Toc133519911)

[Document Purpose and Audience 3](#_Toc133519912)

[System Models 3](#_Toc133519913)

[I. Architecture Diagram 3](#_Toc133519914)

[II. Class Diagram(s) 5](#_Toc133519915)

[III. Class Descriptions 6](#_Toc133519916)

[IV. Sequence diagrams 7](#_Toc133519917)

[Class - Sequence Usage Table 9](#_Toc133519918)

[V. State Diagram 9](#_Toc133519919)

[Tools 10](#_Toc133519920)

[Ownership Report 10](#_Toc133519921)

# Instructions [To be removed]

* **IMPORTANT. Rename this document to**

**CS251-2023-SectionNumber-TAName-LeaderID-DraftToffeeSDSv0.0.pdf for draft version**

**CS251-2023-SectionNumber-TAName-LeaderID-FinalToffeeSDSv1.0.pdf for final version**

* **Include it in a zip file with the code of the project**
* **Remove the following notes and any red notes**
* **This document is the template document for your Software Design.**
* **For further guidelines and information, READ homework 3, document, project description and sample SRS.**

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20210502 | Yusuf Elsayed Abdelrahman Badr | [yusufbadr@yahoo.com](mailto:yusufbadr@yahoo.com) | 01063033085 |
| 20210251 | Alialdin Muhammad Mostafa | Alialdin.mohamad@gmail.com | 01120765911 |
| 20210060 | Osama ?aher Masoued | Osama392maher@gmail.com | 01020083229 |

# Document Purpose and Audience

This document is about the software design specifications of the Toffee Project system. This document is intended for software developers so that they can gain a better understanding of how the system is supposed to work. Furthermore, various diagrams have been included in the document for the aforementioned purpose. A clear class diagram has been included too to facilitate a better understanding of the different components of the system. If you have any concerns about this design specifications document, please do not hesitate to contact the software design team responsible for generating this document

# 

# System Models

## I. Architecture Diagram

1- Main Components or Subsystems:

1. Catalogue Subsystem: This subsystem is responsible for managing the catalog of products that the system will offer. It stores product information such as name, category, description, image, brand, price, and discount percentage (if any). This subsystem will be updated by the admin
2. Authentication and Authorization Subsystem: This subsystem is responsible for user authentication and authorization. It ensures that only authenticated users are allowed to access the system and perform actions based on their role and permissions.
3. Shopping Cart Subsystem: This subsystem is responsible for managing the user's shopping cart. It allows users to add and remove items, update quantities, and view the total cost of their orders.
4. Order Management Subsystem: This subsystem is responsible for managing orders placed by users. It includes features such as order tracking, shipping, and payment processing.
5. Loyalty Points Subsystem: This subsystem is responsible for managing the loyalty points earned by users. It tracks users' points and allows them to redeem points for discounts or other rewards.
6. Gift Voucher Subsystem: This subsystem is responsible for managing gift vouchers. It generates unique codes for each voucher and allows users to redeem vouchers during the checkout process.
7. Reporting and Analytics Subsystem: This subsystem is responsible for generating reports and analytics on various aspects of the system, such as sales, inventory, and user
8. A suitable architectural design for e-commerce systems like Toffee could be a three-tier system consisting of

a) Presentation Tier: This layer provides the user interface for customers to interact with the system. It includes web pages, mobile apps, and other interfaces.

b) Application Layer: This layer is responsible for implementing the business logic of the system. It includes subsystems such as the Catalogue, Authentication and Authorization, Shopping Cart, Order Management, Loyalty Points, and Gift Voucher subsystems.

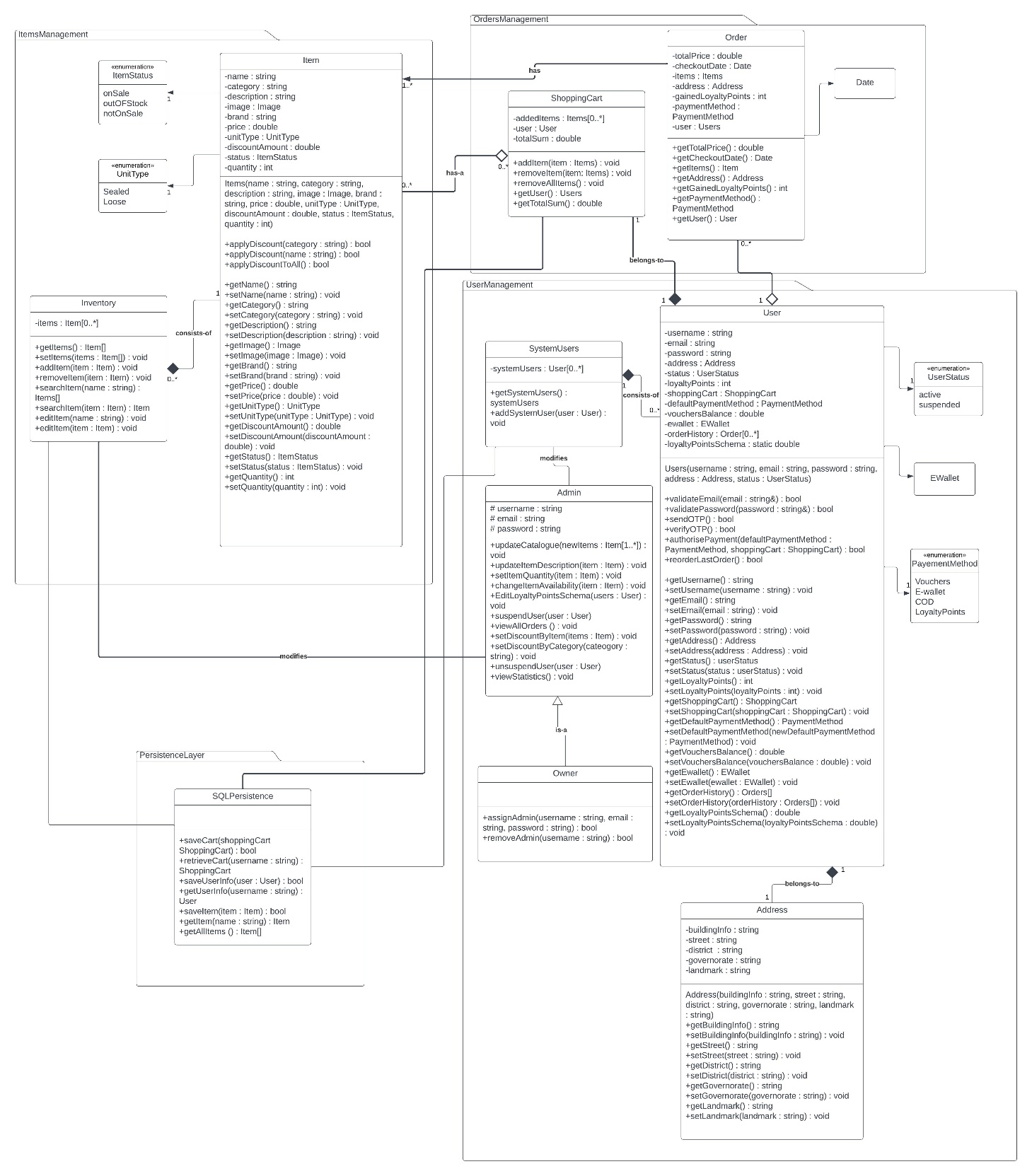
c) Database Layer: This layer stores and manages customer data, product catalogues, orders, and other system information.

A picture containing text, screenshot, font, diagram

Description automatically generated

## 

## II. Class Diagram(s) – For a higher quality please find the attached image in the compressed folder.



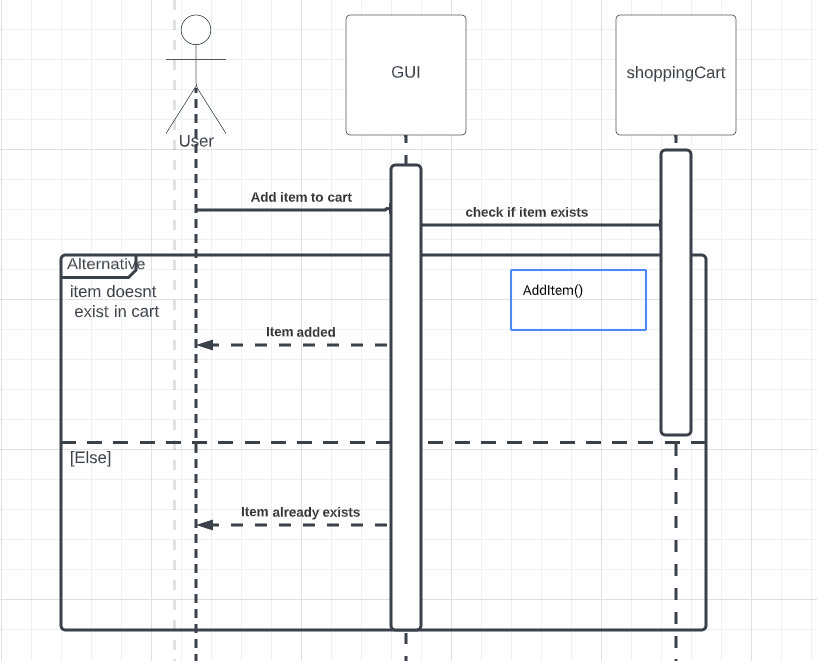
## III. Class Descriptions

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1 | Item | Store all details about an item and can apply discounts or modify details |
| 2 | ItemStatus | Enumeration |
| 3 | UnitType | Enumeration |
| 4 | Inventory | To keep track of all items available |
| 5 | User | Stores data about a single user while giving the option of changing these details. Payment authorization occurs here. Validation of email, password and OTP happens here as well. Reorder last order also available |
| 6 | PaymentMethod | Enumeration |
| 7 | Address | Stores address of the user |
| 8 | SystemUsers | Keeps track of all users registered on the system |
| 9 | Order | Details about previous orders |
| 10 | ShoppingCart | Shopping cart of each user with the option of changing its contents |
| 11 | Admin | Gives certain privileges to admins assigned by owner |
| 12 | Owner | Full privileges of admins with the extra privilege of assigning or removing admins |
| 13 | EWallet | A class for an electronic wallet (EWallet) |
| 14 | SQLPersistence | For handling the retrieving and saving of the data to and from a database using SQL |

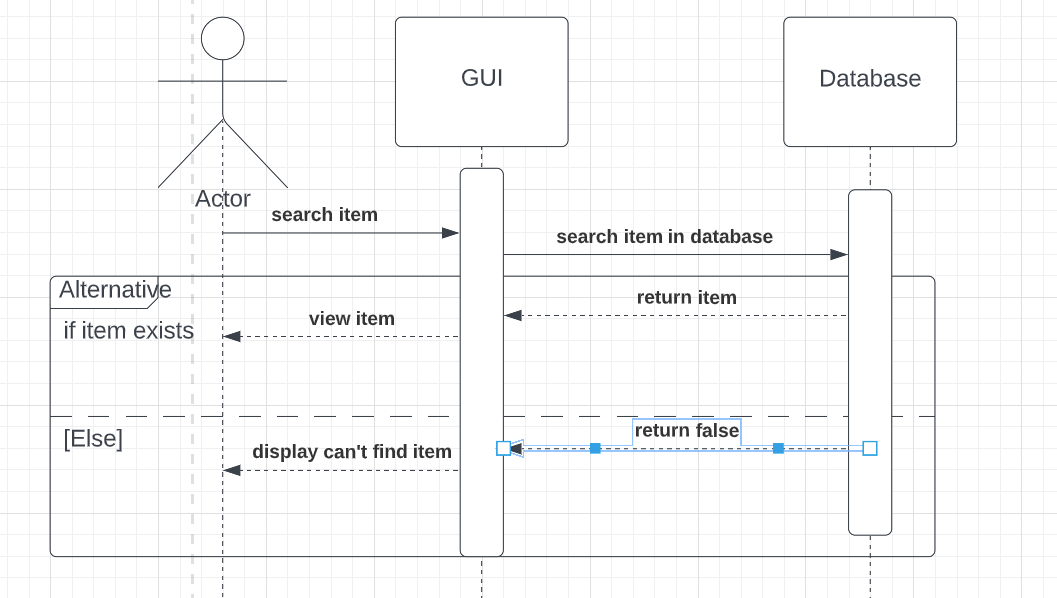
## IV. Sequence diagrams

* **Usually each use case is represented by a sequence diagram or more.**
* **Draw a sequence diagram for the most important SIX use cases (user stories) that have complex interaction.**
* **Overall, all the diagrams should represent all requirements and possible flows for the use case.**
* **Make sure that each object in the sequence diagram has a corresponding class in the class description table above. If not, it will be REJECTED.**
* **Put actual function calls with proper parameters and return types corresponding to class diagrams.**
* **Following are couple of examples for small / medium examples. We expect such diagrams, however there is a missing thing in them. Most of calls don’t have parameters. Please always specify the parameters in the call, matching the class diagram.**

**Adding items to cart**

****

**Searching an item**

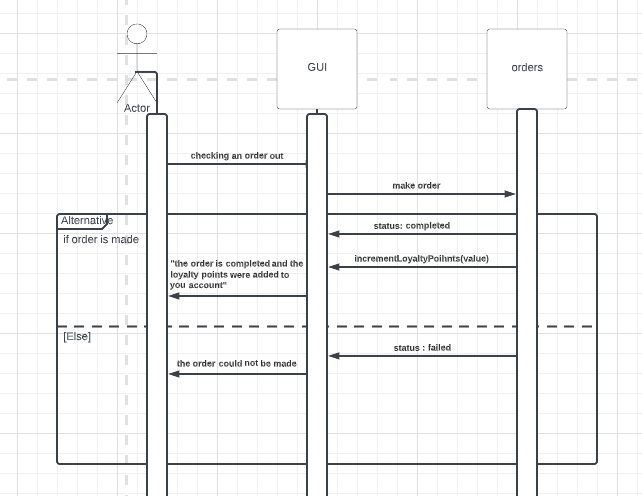
****

### Registration

A picture containing text, diagram, parallel, plan

Description automatically generated

Gaining loyalty points



Displaying items

A picture containing diagram, text, plan, line

Description automatically generated

Checkinh an order out

A picture containing text, diagram, parallel, line

Description automatically generated

### Class - Sequence Usage Table

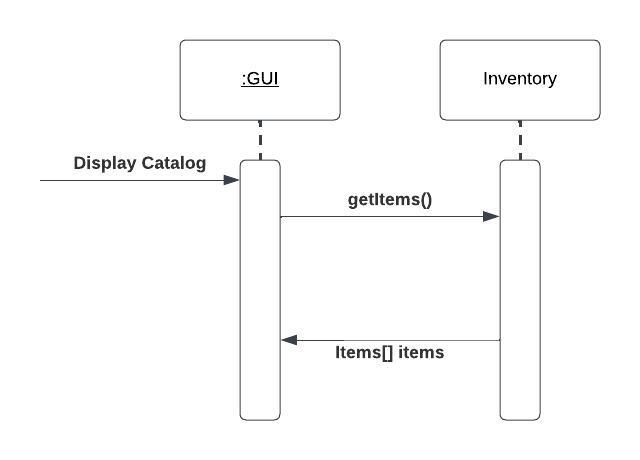
* **In this table, we will list the sequence diagrams you drew. For each one, list all the classes used in this sequence. For each class list all the methods you used in this class. Every method or object on a sequence diagram must belong to an existing class in the class diagram and be shown there. If sequence diagrams do not reflect actual classes and methods, they will be REJECTED.**

| **Sequence Diagram** | **Classes Used** | **All Methods Used** |
| --- | --- | --- |
| 1. Book Field | Class Field  Class Player | Methods …..  Methods …. |
| 1. Displaying items | GUI  Server  Items | Get items |
| 1. registeration | GUI  Server | Verify email  Add user |
| 1. checking an order out | Payment gateway  Orders  Shopping cart | Add order  Get total sum |
| 1. gaining loyalty points | Gui  orders | Increment loyalty points |
|  |  |  |

## V. State Diagram

* **For the order object, draw a state diagram to show the developer the different states it can be in. (for example it is initially created, then it can be shipped, cancelled (if cancelling is possible), …., etc.)**

**Sequence Diagrams - Implementation**

****

**Adding items to cart**

A picture containing text, diagram, parallel, line

Description automatically generated

**Adding items to cart**

A picture containing text, diagram, parallel, line

Description automatically generated

# Tools

# Lucidchart

**Yusuf Elsayed Abdelrahman Badr – 20210502 – Individual BONUS**

**Tool Number 65: ChatGPT**

ChatGPT has proven to be one of the best Artificial Intelligence (AI) model to exist. After reviewing this tool, I can confidently say it is very useful and is proven to be helpful to all people, even the non-tech savvy people.

ChatGPT has the ability to help people with various stuff. Although it does not have the ability to generate pictures or images (as of ChatGPT 3), it can depict what it means by using text characters only. For instance, ChatGPT has the ability to suggest various UML diagrams for a given text problem by illustrating this diagram using text characters.

One particular interesting thing about ChatGPT is that it was able to suggest daily hourly routine to schedule your studying day effectively and productively.

When prompted by “How should I schedule my studying day given <a particular constraint(s)>?”, the AI model was able to suggest a daily schedule which did make sense and was indeed productive and effective while taking in consideration the other constraints. Although this was not always the case, as the AI model did sometimes suggest unrealistic schedules which at times did not follow the given constraints, this was a border case and more often than not the AI model was indeed generating effective schedules.

Another interesting feature is the ability of this AI model to help in composing various emails while taking into account whom is this email intended to. It does so while making sure that the email format matches setting to which this email is intended for.

Not only that, but the AI model is also capable of generating cover letters which makes applying for various jobs easier and can give the applicants new idea that they might want to include in their cover letter.

Another aspect which ChatGPT is great at handling is roleplaying i.e., helping you to simulate the process of you talking to another person as an interviewer for instance. This can be very helpful in for instance, simulating an interview process, which can help you get better prepared for an interview and thus increasing the probability of you getting accepted in that job.

Furthermore, this AI model can help you understand various concept ranging from computer science content to any other random knowledge that you might want to know.

# Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100%.**
* **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Class Diagram + Class Descriptions + DisplayCatalog Sequence Diagram + Classes: Item + Inventory + ShoppingCart + Order | Yusuf Elsayed Abdelrahman Badr |
|  |  |
|  |  |