Cairo University  
Faculty of Computers and Artificial Intelligence



**CS251**

**Introduction to Software Engineering**

Project Name

Software Design Specifications

Version 0.0

Month & Year

Contents

[Team 3](#_Toc132125211)

[Document Purpose and Audience 3](#_Toc132125212)

[System Models 3](#_Toc132125213)

[I. Architecture Diagram 3](#_Toc132125214)

[II. Class Diagram(s) 4](#_Toc132125215)

[III. Class Descriptions 5](#_Toc132125216)

[IV. Sequence diagrams 6](#_Toc132125217)

[Class - Sequence Usage Table 7](#_Toc132125218)

[V. State Diagram 8](#_Toc132125219)

[Tools 8](#_Toc132125220)

[Ownership Report 8](#_Toc132125221)

# 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** |
| 20210420 | MinaMagedFaris | Menamaged669@gmail.com | 01140240229 |
|  |  |  |  |
|  |  |  |  |

# Document Purpose and Audience

* **SDS (System Design Specification) is a document that outlines the design of a software system.**
* **SDS helps ensure that the software system is designed and built according to specific requirements, and that it meets the needs of its users.**
* **Audience: software developers, designers, project managers and stakeholders such as product owners, business analysts, and quality assurance personnel.**

# System Models

## I. Architecture Diagram

* **Decide on suitable software architecture for this system. Describe the architecture you chose and why it is suitable for Toffee.**
* **Provide an architecture diagram showing the different components of the system and their relation to each other. Use suitable notation like C4 or arrow and box.**

## II. Class Diagram(s)

Please note that I put the file in the folder for a better view it will open with Visual Paradigm only.

Diagram

Description automatically generated

## III. Class Descriptions

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1. | product | In this class we used to store item information. |
| 2. | Category | Every class product has one category to divide the items and make it easier to be found. |
| 3. | order | In this class we store order details. |
| 4. | Order tracker | From the name of class we track orders and store the number of orders in it and the details of that order to use it in reorder function and to use it to make statistical analysis for the admin or order history. |
| 5. | Admin | In this class we store admin details and give him his responsibilities. |
| 6. | Statistics | In this class we use order tracker class to give us some information about orders to make us able to make the statistics we want. |
| 7. | Customer | In this class we allow non-logged-in customers to view the products and some features, but s/he cannot make an order he must log-in. |
| 8. | logged-in customer | Inherited from customer In this class we make logged-in customer have special features like add to cart and logged-in customer can make an order. |
| 9. | Cart | In cart class we store items that the user wants to buy later. |
| 10. | Payment | We store the payment method in it. |
| 11. | Pay on Delivery | We store some information about the payment like order details and its price |
| 12. | Pay Online | We store the important information about how customer is going to pay in another words what method s/he is going to use like e-wallets |
| 13. | Authentication | Class to make user sign-up or login and it make sure that the user is logged- in and it has an OTP function that sends the message. |

## 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.**





### 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 …. |

## 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.)**

# Tools

* **Write a list of all tools used to develop the design (e.g., ArgoUML, Visual-Paradigm, etc.)**

# 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** |
| **Mina Maged Faris** | **Part of class diagram and sequence diagrams 2 and 3.** |
|  |  |
|  |  |