Cairo University  
Faculty of Computers and Artificial Intelligence



**CS251**

**Introduction to Software Engineering**

Toffee’s SDS

Software Design Specifications

Draft Version

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20210501 | Laila Hesham Kandil | lailakandil64@gmail.com |
| 20211034 | Doaa Ali El-Sayed | Doali246885@gmail.com |
| 20210458 | Youssef Ahmed Zakaria | youssefahmed052@gmail.com |

Contents

[Team 3](#_Toc133443605)

[Document Purpose and Audience 3](#_Toc133443606)

[System Models 5](#_Toc133443607)

[I. Architecture Diagram 5](#_Toc133443608)

[II. Class Diagram(s) 7](#_Toc133443609)

[III. Class Descriptions 8](#_Toc133443610)

[IV. Sequence diagrams 9](#_Toc133443611)

[Class - Sequence Usage Table 10](#_Toc133443612)

[V. State Diagram 11](#_Toc133443613)

[Tools 11](#_Toc133443614)

[Ownership Report 11](#_Toc133443615)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20210501 | Laila Hesham Kandil | lailakandil64@gmail.com | 01012960328 |
| 20211034 | Doaa Ali El-Sayed | Doali246885@gmail.com | 01019271286 |
| 20210458 | Youssef Ahmed Zakaria | youssefahmed052@gmail.com | 01120700202 |

# Document Purpose and Audience

**Purpose:**

The purpose of an SDS document for the project called Toffee, which is an e-commerce platform, is to provide a comprehensive overview of the system's design and functionality. The system architecture will depict how Toffee's e-commerce platform works and what its parts are, such as the user interface, payment gateway, order processing system, and database. It will help people understand how Toffee's e-commerce platform behaves and what it can do. The architecture will cover various aspects of Toffee's e-commerce platform, like how it interacts with users, manages orders, and ensures data privacy and security.

The sequence diagram for Toffee's e-commerce platform will illustrate the interactions between its components during a particular sequence of events, such as a user searching for a product, adding it to the cart, and placing an order. It will show how different components of Toffee's e-commerce platform, such as the search engine, product catalog, cart management, and payment gateway, interact with each other to fulfill the user's request.

The state diagram for Toffee's e-commerce platform will show the different states that the system can be in, such as idle, searching, cart management, and order processing, and how it transitions between those states based on user actions and system events.

Together, these diagrams will provide a clear representation of Toffee's e-commerce platform's design and behavior, which can be used by developers, testers, and other stakeholders to understand and evaluate its functionality and make informed decisions about its development and deployment.

**Audience:**

The audience of 3-tier architecture includes people who are involved in the design, development, and deployment of software systems.

* software developers.
* software architects.
* project managers.
* technical decision-makers.

# System Models

## Diagram Description automatically generatedI. Architecture Diagram

**Advantages of Three-Tier Architecture :**

* The key three-tier benefit is improved scalability since the application servers can be deployed on many machines. Also, the database does not make longer connections with every client – it only requires connections from a smaller number of application servers.
* It improves data integrity. Here, all the updated information goes through the second tier. The second tier can ensure that only important information is allowed to be updated in the database and the risk of unreliable client applications corrupting information is removed.
* Security is improved since the client does not have direct access to the database; it is more difficult for a client to obtain unauthorized data. Business logic is more secure because it is stored on a secure central server.
* High performance, lightweight, persistent object.
* Better to reuse.
* Easy to maintain and modify.
* The added modularity makes it easier to modify or replace one tier without affecting the other tier.

## II. Class Diagram(s)

Diagram

Description automatically generated

## III. Class Descriptions

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1. | User’s Account | This class contains all the necessary information about the user which includes his name, email, address, password and his status on the application. There are setter functions that are used in the registration and the getter functions are used in the login process to check if the user is available or not and the final function in this class checks if the user entered a valid password or not. |
| 2. | Item | This class contains all the necessary information about the item in the catalog which includes name, category it is in, description, brand, price, and unit type which can either be kg or lbs, status, the discount value, and whether it’s available or not. The functions that are in this class are used to set the status of the item (on sale - out of stock - not on sale) and the set quantity function shows how much of this item is left before it needs to be restocked. |
| 3. | Catalog | The class Catalog has an array of items of type items and it has the following functions: we can search for a certain item, the user can filter items by category, the user can display all the items regardless of whether they are in the same category or not, the discount value on the item or category and finally if the admin is the one using the application, he can update the catalog by adding new items or removing items. |
| 4. | Address | This class contains all the necessary information about the user’s address which includes governorate, district, street, building information (number, floor, flat), and landmark. This class is called when the user is about to make an order and is entering his details. |
| 5. | Order | This class contains the total price of the items that the user has added to the shopping cart, the date of purchase, the address to that the user wants the items to be delivered, his loyalty points, his available vouchers, and his e-wallet information. |
| 6. | Shopping Cart | This class contains the items that the user has added to his shopping cart and the user can add more items to the shopping cart, remove items from the shopping cart, and the total price is added up automatically as the user adds items to the cart |
| 7. | Payment | This class includes all the different payment methods that the user can pay whether by cash or credit or using his loyalty points or a voucher. |
| 8. | Admin’s Account | This class contains all the necessary information about the system admin which includes his name, email, password .There are setter functions that are used in the registration and the getter functions are used in the login process to check if the admin is available or not and the final function in this class checks if the admin entered a valid password or not. |

## 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 a couple of examples for small/medium examples. We expect such diagrams, however, there is a missing thing in them. Most of the 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

* Visual Paradigm
* LucidChart

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Laila Hesham Kandil |  |
| Youssef Ahmed Zakaria |  |
| Doaa Ali ElSayed |  |