Cairo University  
Faculty of Computers and Artificial Intelligence



**CS251**

**Introduction to Software Engineering**

Toffee

Software Design Specifications

Version 1.0

Section s25-s26

Done By:

Badr Mohamed Ragab El-Said (20210605) badrmohamedragab2003@gmail.com

Mohamed Amir Mohamed (20211079) mohamedamir5050@gmail.com

Omar Rabea Shaaban (20210593) omar.rabea.shaban2002@gamil.com

April - 2023

Contents

[Team 3](#_Toc132125211)

[Document Purpose and Audience 3](#_Toc132125212)

[System Models 3](#_Toc132125213)

[I. Architecture Diagram 4](#_Toc132125214)

[II. Class Diagram(s) 6](#_Toc132125215)

[III. Class Descriptions 6](#_Toc132125216)

[IV. Sequence diagrams 9](#_Toc132125217)

[Class - Sequence Usage Table 10](#_Toc132125218)

[V. State Diagram 11](#_Toc132125219)

[Tools 11](#_Toc132125220)

[Ownership Report 11](#_Toc132125221)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20210605 | **Badr Mohamed Ragab El-Said** | badrmohamedragab2003@gmail.com | 01023457530 |
| 20211079 | **Mohamed Amir Mohamed** | mohamedamir5050@gmail.com | 01110089085 |
| 20210593 | **Omar Rabea Shaaban** | omar.rabea.shaban2002@gamil.com | 01141482782 |

# Document Purpose and Audience

# SDS: is a document that outlines the technical details of a software project. It describes the software's architecture, design, and implementation details.

# The SDS is typically written by software engineers or architects and is used as a reference guide throughout the software development lifecycle.

* (2) it can read by
  + Developers
  + Software architects
  + Project managers
  + Stakeholders

# System Models

## I. Architecture Diagram

**1. The main components or subsystems of the system**

**A.** User Interface: This component is responsible for presenting the website or application to the user, including menus, buttons, and other elements that enable users to interact with the system.

**B.** Business Logic or Application server: This component handles the processing of user requests, such as order processing, inventory management, and payment processing.

**C.** Database or Storage: This component stores all data related to the system, including user information, product information, and transaction history.

**2. Architectural design for Toffee**

A suitable architectural design for an e-commerce system like Toffee is a 3-tier system, which separates the presentation (Client or UI) layer will be in the first tier, application layer will be in the second tier, and database (Storage) layer will be in the third tier.

**3. Architectural diagram**

Diagram

Description automatically generated

## II. Class Diagram(s)

## The link of Class Diagram

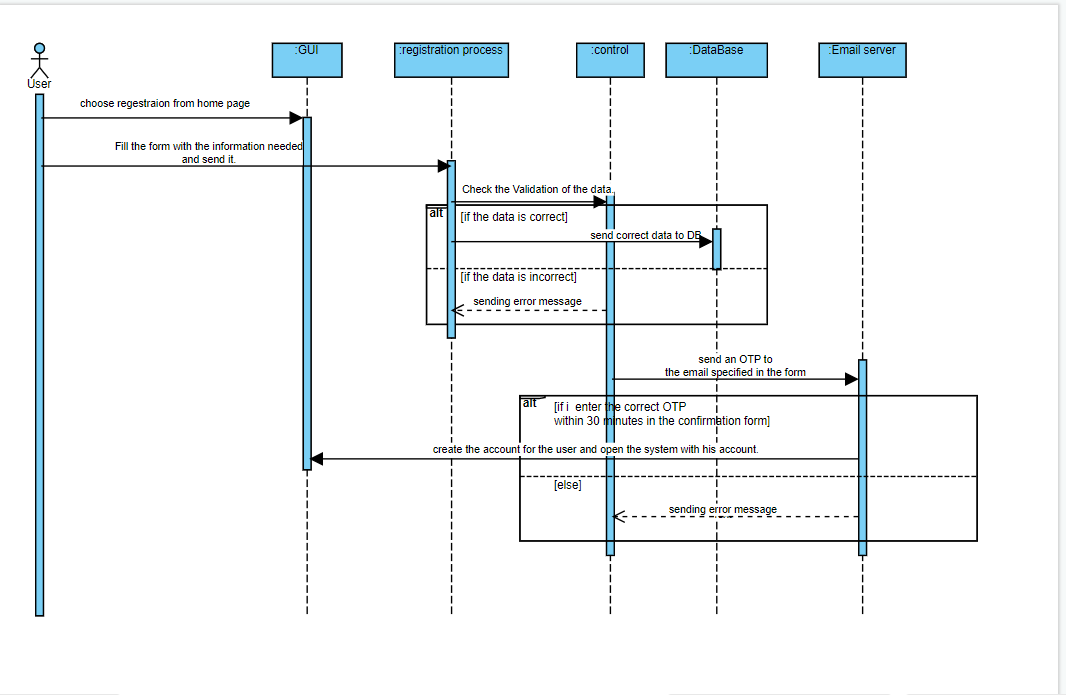
<https://drive.google.com/drive/folders/1kW48ac5mq6GpZRUJytKJxYzSQ-LpUzyb?usp=share_link>

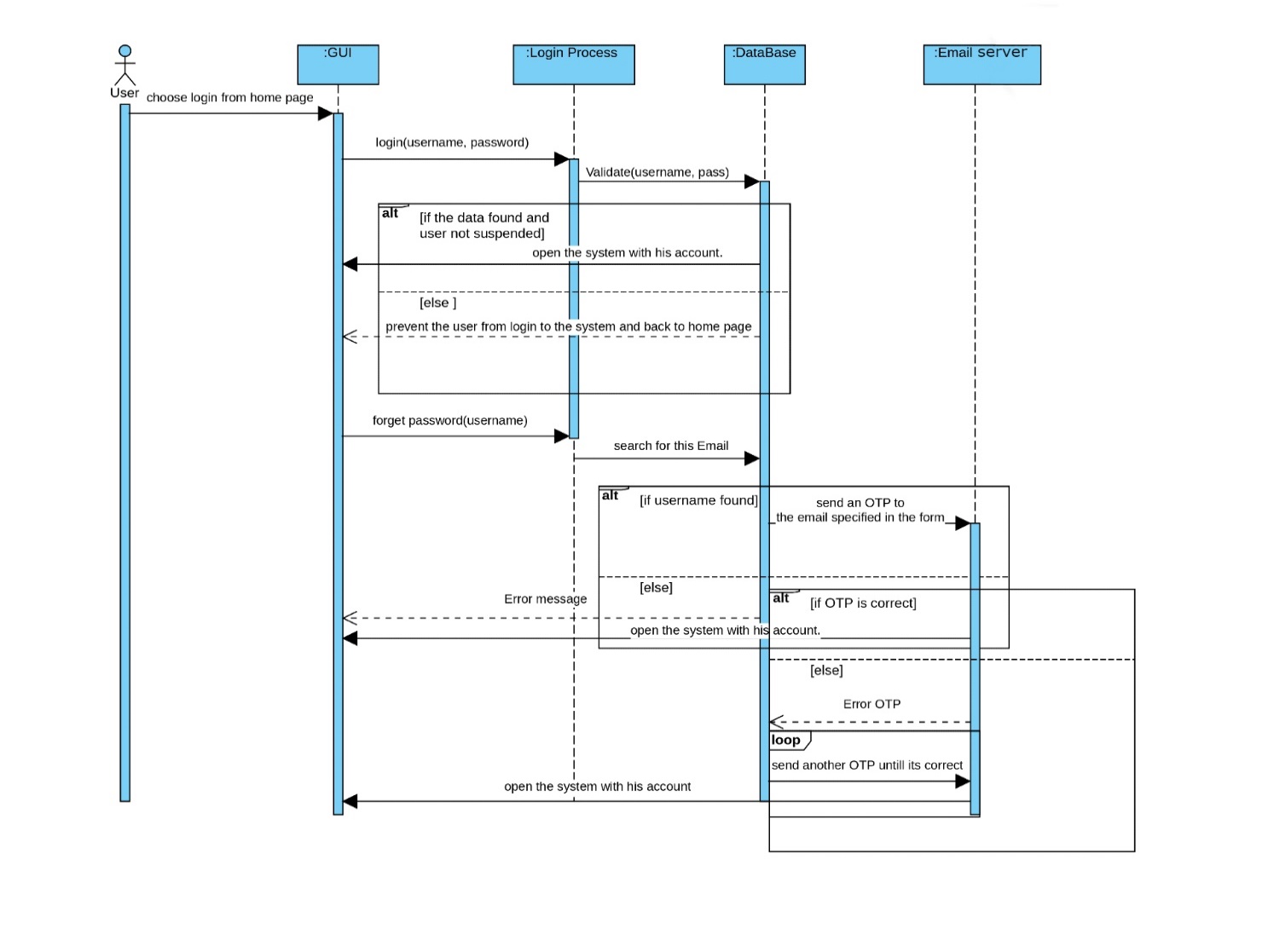
## III. Class Descriptions

| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1. | System | It initializes the database and products and creates user.  It is responsible for running the whole system. |
| 2. | User | It is responsible for doing the functions of general user want to make like:   1. Login 2. Register 3. Add To Cart some products. 4. Check out.   etc. |
| 3. | UserInformation | It is responsible for storing the information of User like:   1. Name 2. ID 3. Orders   etc. |
| 4. | Address | It is responsible for storing the address information of User like:   1. Street 2. Governate 3. Floor Number   etc. |
| 5. | Order | It is responsible for storing the information of orders of User like:   1. Date of the order 2. ID of order 3. Price   etc. |
| 6. | ShoppingCart | It is responsible for storing the products which the user adds to buy and some information of the Shopping cart like:   1. Id 2. CustomerId 3. LastModified   etc. |
| 7. | ShoppingItem | It is responsible for storing the information of the product which the user add to buy like:   1. Id 2. Quantity 3. Price   etc. |
| 8. | PaymentMethod | It responsible of identifying the method of payment and store the information of the method |
| 9. | E-Wallets | It responsible of identifying the method of payment to be using E-Wallets and store the information of the method |
| 10. | Loyalty Points | It responsible of identifying the method of payment to be using Loyalty Points and store the information of the method |
| 11. | Vouchers | It responsible of identifying the method of payment to be using Vouchers and store the information of the method |
| 12. | Cash | It responsible of identifying the method of payment to be using Cash and store the information of the method |
| 13. | FilePersistence | It is responsible for loading, save, create, or remote the shopping cart |
| 14. | PersistenceFactory | It is responsible for doing the processes in the FilePersistence class |
| 15. | Product | It is responsible for storing the information from ProductInformation class |
| 16. | ProductInformation | It is responsible for storing the information of products like:   1. Name 2. Id 3. Quantity   etc. |
| 17. | Catalog | It is responsible for storing the products, showing the catalog and search of a product by Name, Brand, or Category. |
| 18. | Type | It responsible of identifying the type of the product either “**by unity**” (it means user must buy the product by units) or “**by kilo**” (it means user must buy the product by kilos) |
| 19. | ByUnits | It is responsible for storing the amount of the product and Max amount which user can order |
| 20. | ByKilo | It is responsible for storing the amount of the product and Max amount which user can order |
| 21. | Database | It is responsible for storing the users, the information of each user, products in the website and shopping carts |
| 22. | Register Process | It is responsible for doing the register process and save the new user in the database |
| 23. | Login process | It is responsible for doing the login process and validates the information which user entered. |
| 24. | Control | It is responsible for handling the processes which need the database like:   1. Login 2. Register |
| 25. | Admin | It is responsible for storing the information of Admin like:   1. Name 2. ID 3. password   etc. |
| 26. | Type | It responsible of identifying the type of the admin either “**General Admin**” or “**Owner**” |
| 27. | GeneralAdmin | It is an admin |
| 28. | Owner | It is an Admin but can do this assign Admin Privileges and remove Admin Privileges |

## IV. Sequence diagrams

1. Sequence diagram for registration



2.Sequence diagram for Login

### Class - Sequence Usage Table

| **Sequence Diagram** | **Classes Used** | **All Methods Used** |
| --- | --- | --- |
| 1. Registration | Class GUI  Class Registration process  Class control  Class Database | Methods play\_system()  Methods set\_information ()  Methods checkInformation (),  sendOTP (),  confirmOTP(OTP\_num) |
| 1. Login | Class GUI  Class Login process  Class Database | Methods Play\_system()  Methods Login (Username, Password)  validate(Username, Password)  forgotPassword(Username) |

## 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

* **Lucid**
* **Visual Paradigm**
* **Creately**

# Ownership Report

|  |  |
| --- | --- |
| **Owners** | **Item** |
| **Badr Mohamed Ragab El-Said** | Sequence Diagram |
| **Mohamed Amir Mohamed** | Class Diagram |
| **Omar Rabea Shaaban** | Architecture Diagram |