Cairo University  
Faculty of Computers and Artificial Intelligence



**CS251**

**Introduction to Software Engineering**



Software Design Specifications

Version 1.0

|  |  |
| --- | --- |
| **Name** | **Email** |
| Alan Samir Hakoun | [alanhakoun@gmail.com](mailto:alanhakoun@gmail.com) - 01006891062 |
| Sohaila Abdelazim Khalifa | [sohailakhalifa03@gmail.com](mailto:sohailakhalifa03@gmail.com) |
| Sara Tamer Mohamed Bihery | [sasooelbihery@gmail.com](mailto:sasooelbihery@gmail.com) |

April 2023

Contents

[Team 3](#_Toc134812478)

[Document Purpose and Audience 3](#_Toc134812479)

[System Models 3](#_Toc134812480)

[I. Architecture Diagram 3](#_Toc134812481)

[II. Class Diagram(s) 7](#_Toc134812482)

[III. Class Descriptions 8](#_Toc134812483)

[IV. Sequence diagrams 10](#_Toc134812484)

[Class - Sequence Usage Table 16](#_Toc134812485)

[V. State Diagram 17](#_Toc134812486)

[Tools 18](#_Toc134812487)

[Ownership Report 18](#_Toc134812488)

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
| 20210755 | Alan Samir Hakoun | [alanhakoun@gmail.com](mailto:alanhakoun@gmail.com) | 01006891062 |
| 20210492 | Sohaila Abdelazim Khalifa | [Sohailakhalifa03@gmail.com](mailto:Sohailakhalifa03@gmail.com) | 01278489956 |
| 20210155 | Sara Tamer Mohamed Bihery | [sasooelbihery@gmail.com](mailto:sasooelbihery@gmail.com) | 01274239962 |

# Document Purpose and Audience

This document is meant to illustrate the Software Design Specifications such as the architecture, class, sequence and state designs. It outlines the technical details of the development of the software to help in implementing an accurate application depending on the requirements given earlier. Moreover, the document acts as a reference for the development team throughout the software development life cycle.

Audience:

• Project manager

• Software Development team [software architects, developers, testers]

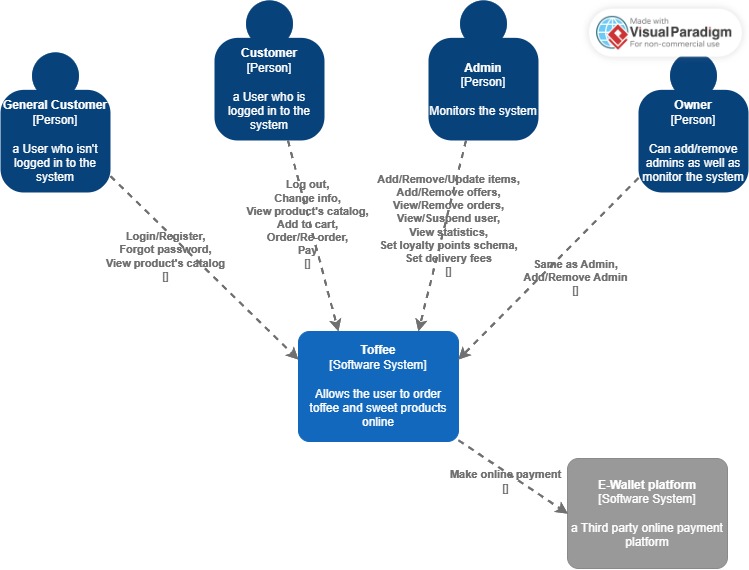
• Stakeholders.

# System Models

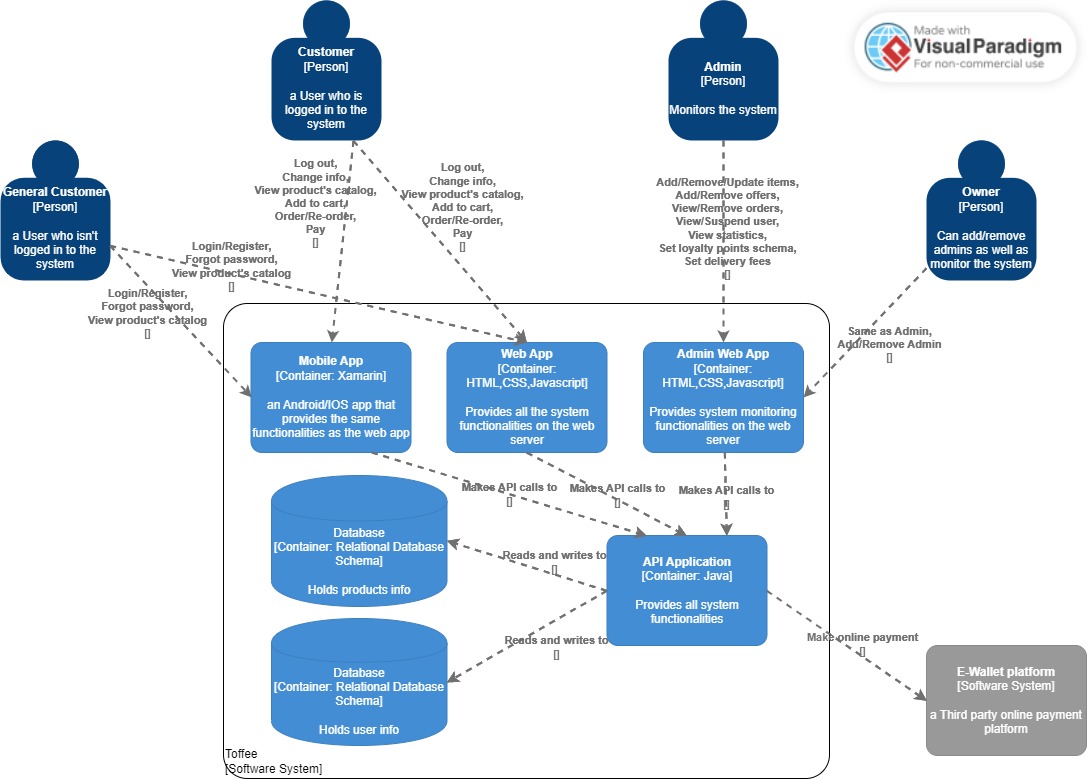
## I. Architecture Diagram

Architecture used: C4 notation.   
Description: We used C4 architecture as its simple abstractions will make it easy to use as well as easy to understand, it gives details on what the system is, showing system components and the technologies used and who is going to use it, as well as what interacts with the database covering presentation, application and data tiers in a more detailed manner than 3-tier architecture.

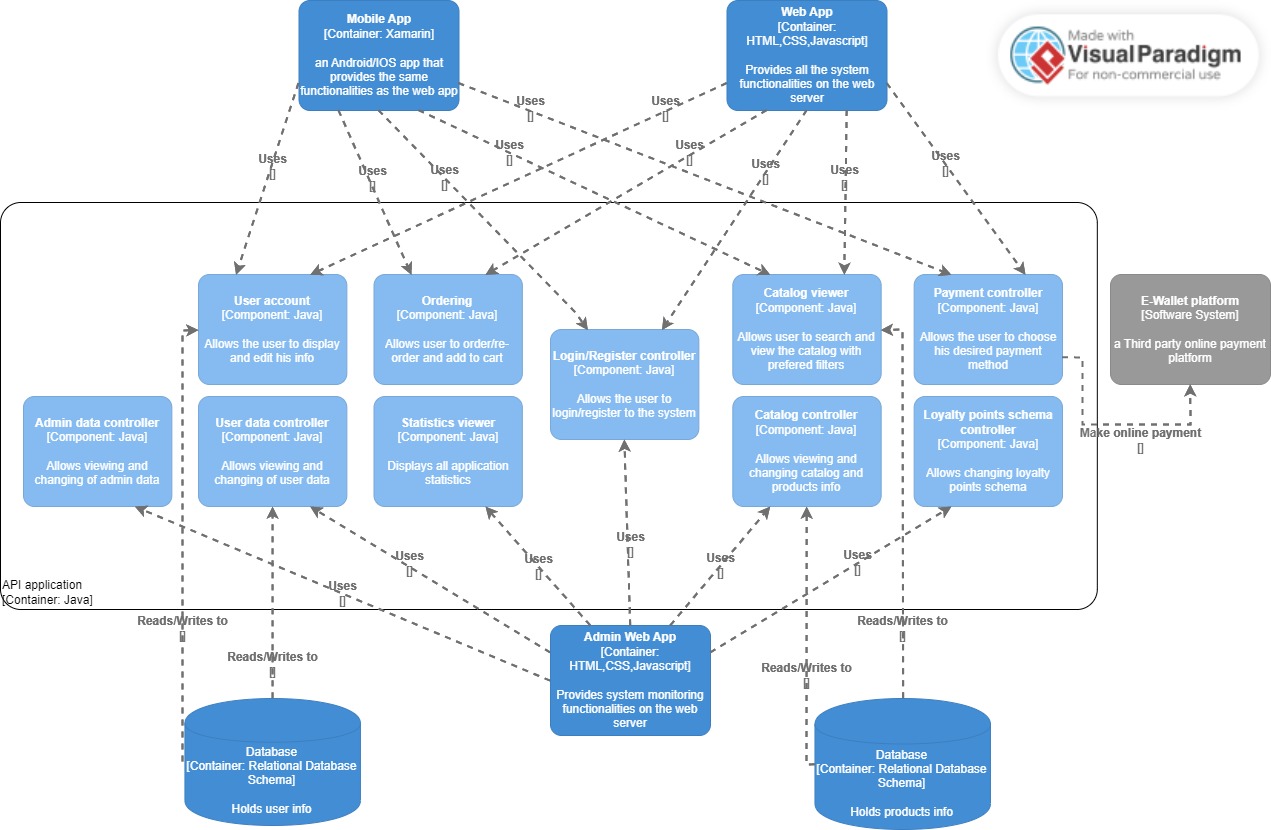
Diagram:

  
**context:**

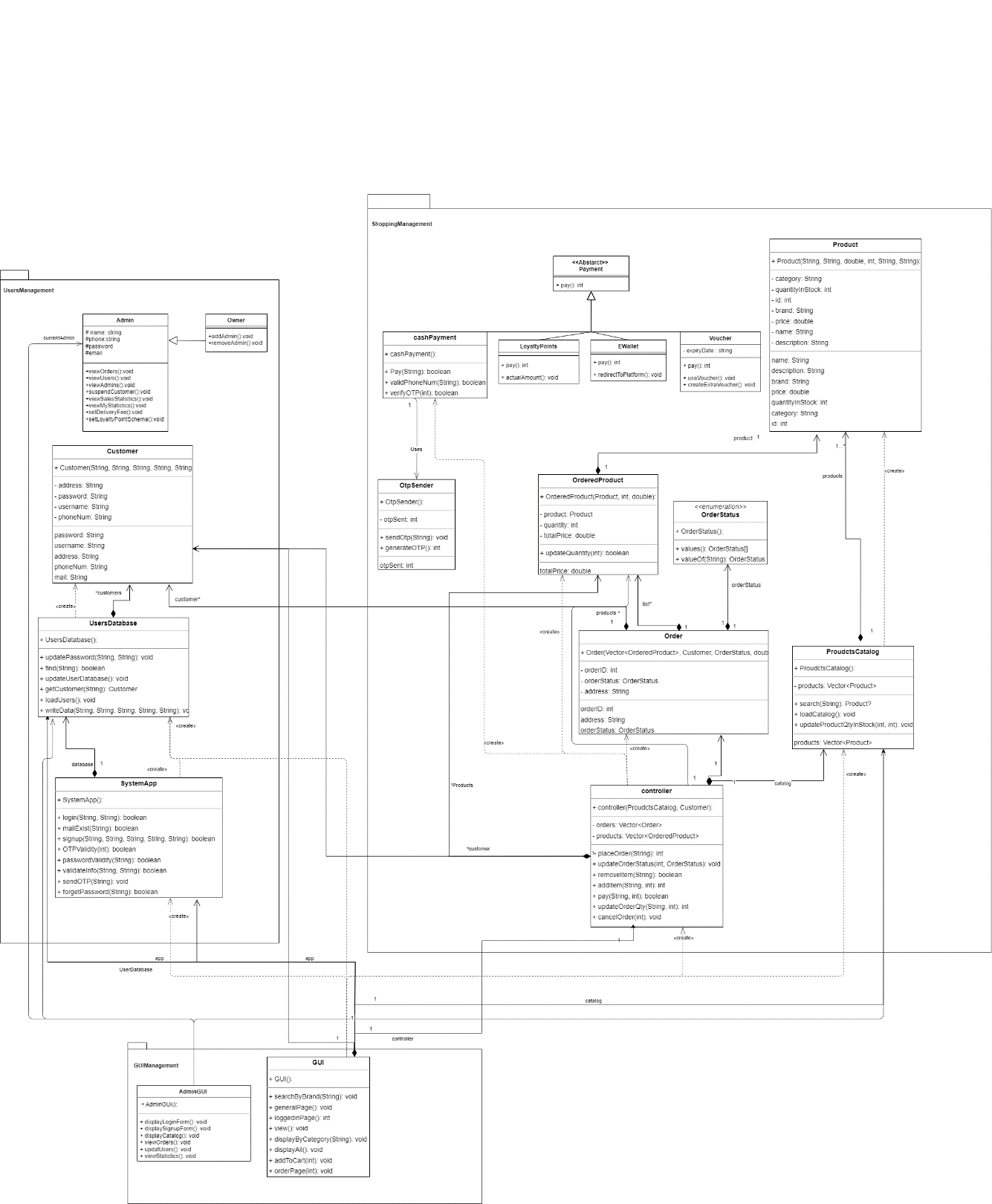
**Container:**



**Component:**



## II. Class Diagram(s)



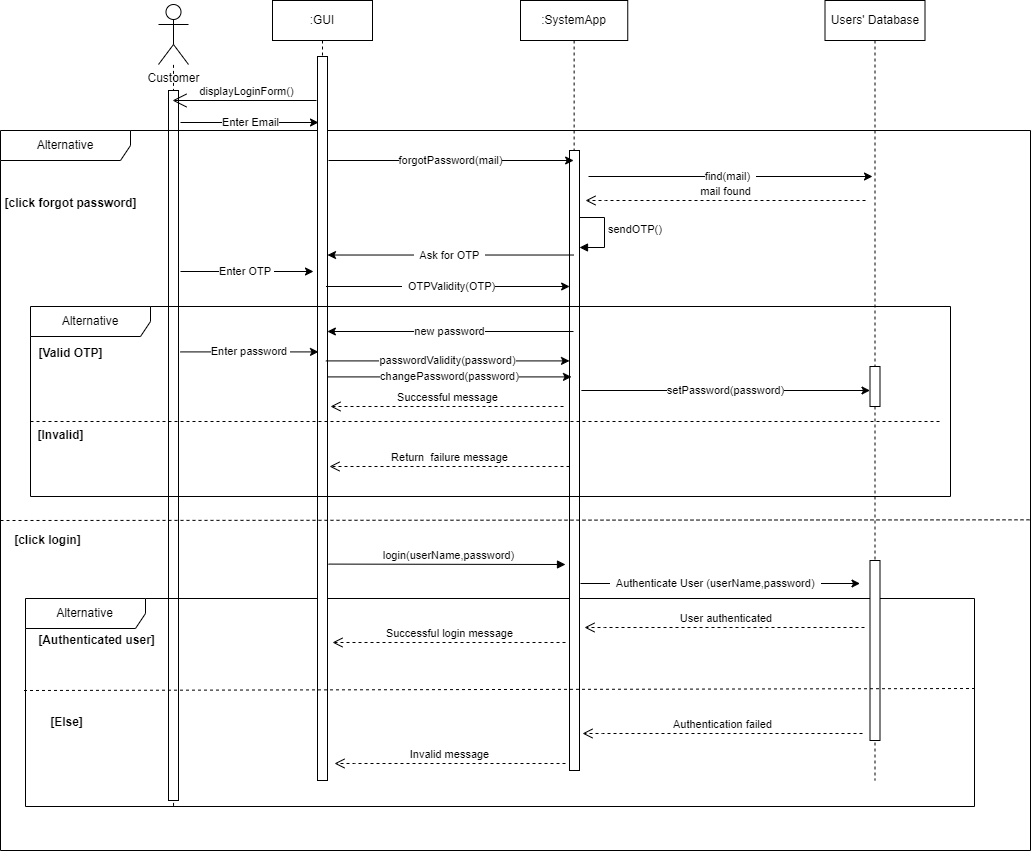
**NOTE: for better visualization please refer to:** [**https://drive.google.com/file/d/1AMZbXmPbUaagHET9g8F\_FYOAMMYphHv9/view?usp=sharing**](https://drive.google.com/file/d/1AMZbXmPbUaagHET9g8F_FYOAMMYphHv9/view?usp=sharing)

## III. Class Descriptions

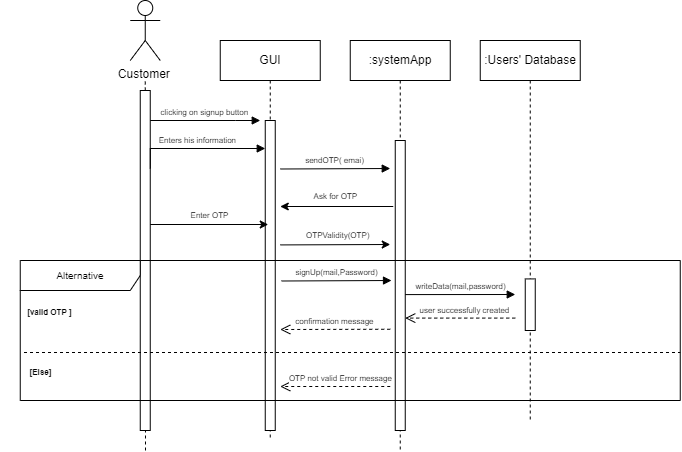
| **Class ID** | **Class Name** | **Description & Responsibility** |
| --- | --- | --- |
| 1. | SystemApp | It’s the controller class that is responsible for managing the registration and accessing app features by its methods such as login and signup. In addition, it has a relationship between the database class to be able to read and write user/system info and connect return the result to the GUI. |
| 2. | Customer | Represents users who have created an account on the application, and therefore have access to additional features and methods such as making an order, adding to cart and paying for order info rather than just viewing the catalog. |
| 3. | Admin | It’s the class that includes all admin’s attributes and methods to allow him/her to control catalog, customers and track sales, the class has relationships with other classes as Order, Payment, and Catalog as admin can access each of them. |
| 4. | Owner | It’s an inherited class from Admin class, as the owner has all admin responsibilities and relationships in addition to some methods like tracking all admin and controlling each of them. |
| 5. | Order | It illustrates an order in the application. Each order has an id and contains information such as the address and status. |
| 6. | OrderdProduct | Acts as a container that saves the product and its quantity to be ordered in cart. |
| 7. | Product | Contain the items that are available for purchase. Each product has a unique id, name, description, quantity in stock and price. |
| 8. | ProductsCatalog | Represents the collection of all products that are available for purchase in the database and update the database after each order. |
| 9. | Payment | Shows the different methods of payment that are available to users when making an order. The Payment class is the abstract class of four inherited classes: Cash, LoyaltyPoints, Voucher, and EWallet. Each inherited class represents a specific method of payment that a user can use to complete the transaction. |
| 10. | Cash | Represents cash payments and its methods including verifying the phone number using mobile OTP. |
| 11. | LoyaltyPoints | Represents methods that use loyalty points as the payment. |
| 12. | Voucher | Represents methods that use gift vouchers as payment. |
| 13. | EWallet | Represents payments made using a digital wallet redirecting to another platform. |
| 14. | GUI | Illustrates the class that deals with the user interface of a customer and connects to application features. It has a direct relationship with the system app, controller and catalog. |
| 15. | AdminGUI | Illustrates the class that deals with the user interface of an admin and connects to its application features. |
| 16. | UsersDatabase | Contains the database of the system including user info and connected to the system app. |
| 17. | Controller | It’s the class that controls the shopping process that begins with adding an item to the cart and ends with paying and shipping this order to its customer and closing the order. It is also connected to the catalog to update it after the order is completed. |
| 18. | OrderStatus | It represents the status of the order between created, ready to ship, shipping, shipped, returned, cancelled, and closed. |
| 19. | OTPsender | It’s the class responsible for sending an OTP to the mobile number in the cash payment case to verify the customer's number. It’s sending SMS to numbers which only verified on Twilio. |

## IV. Sequence diagrams

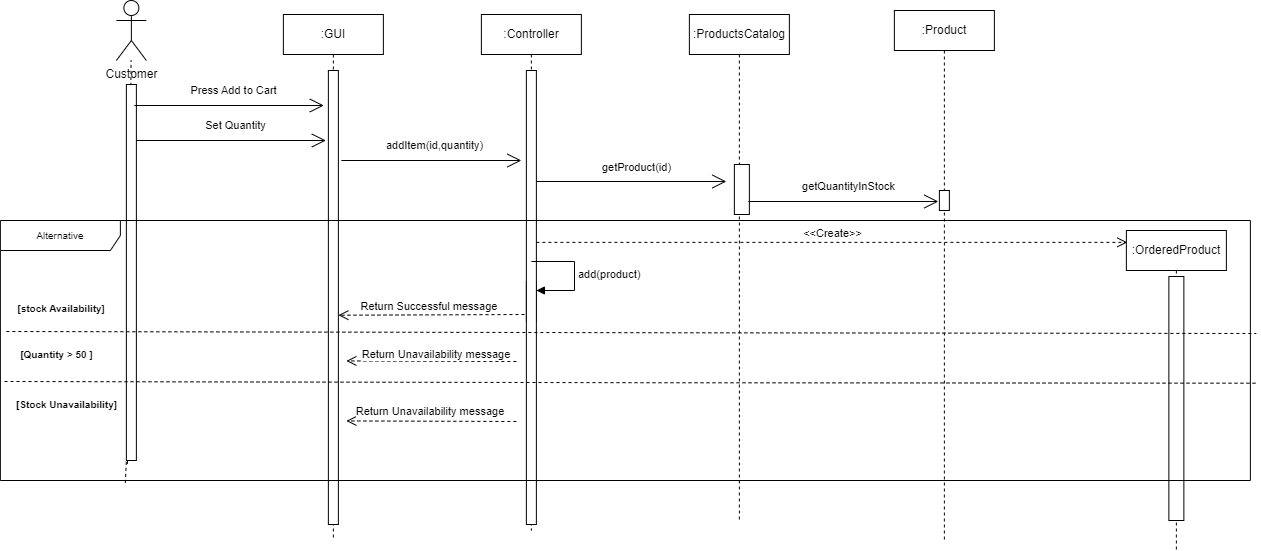
1.login:

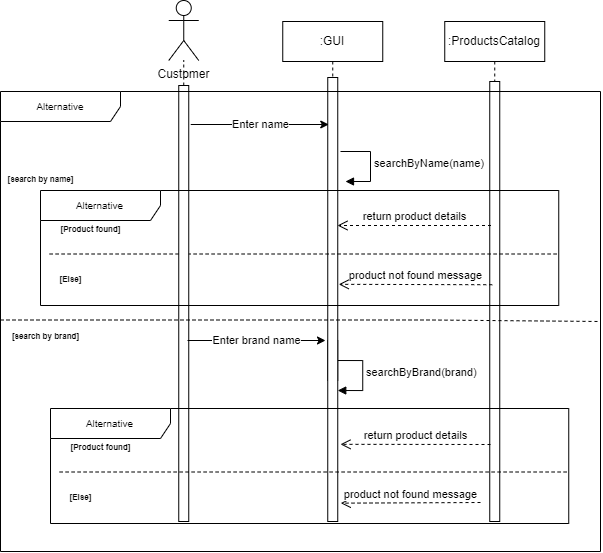


2.Register[signup]:

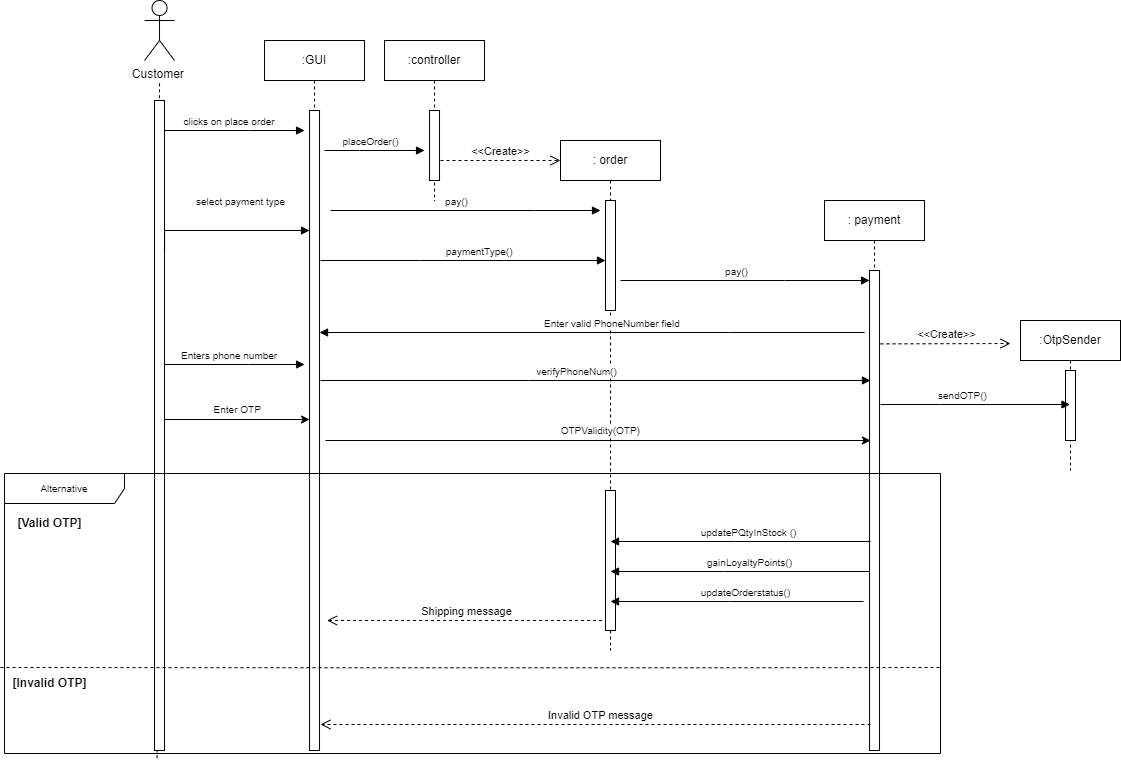


3.Add to Cart:

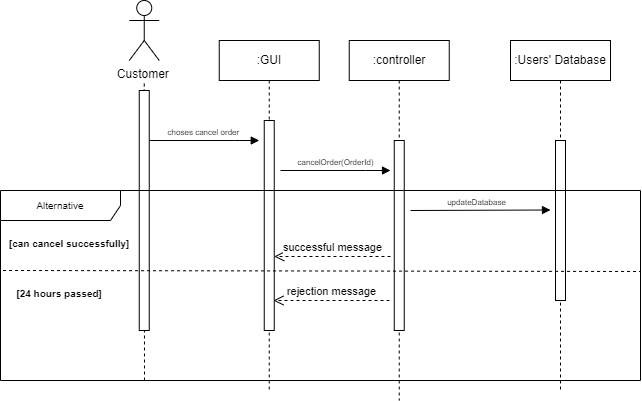


4. Search for an item:

5.Order:



6.Cancel Order:



### Class - Sequence Usage Table

| **Sequence Diagram** | **Classes Used** | **All Methods Used** |
| --- | --- | --- |
| 1. Login | Class **GUI**  Class **systemApp**  Class **Users'Database** | Login  AuthenticateUser.  MailValidity  OTPValidity  PasswordValidity  EditInfo |
| 1. Register | Class **GUI**  Class **systemApp**  Class **Users'Database** | Signup  validateInfo  OTPValidity  SendOtp  writeData |
| 1. Add to Cart | Class **GUI**  Class **Controller**  Class **ProductsCatalog**  Class **Product**  Class **OrderedProduct** | addItem  getProduct  getQuantityInStock |
| 1. Search for an item | Class **GUI**  Class **ProductsCatalog** | searchByName  searchByBrand |
| 1. Order | Class **GUI**  Class **Controller**  Class **Order**  Class **Payement**  Class **OtpSender** | placeOrder checkout  pay  otpValidity  verifyPhoneNum  paymentType  updateQtyInStock  updateOrderStatus |
| 6. Cancel order | Class **GUI**  Class **Controller**  Class **UsersDatabase** | cancelOrder  update |

## 

## V. State Diagram

# 

# Tools

* **Draw.io - class diagram & sequence diagram & state diagram.**
* **Visual paradigm - Architecture Diagram.**

# Ownership Report

|  |  |
| --- | --- |
| **Item** | **Owners** |
| Architecture Diagram, parts of class Diagram. | Alan Samir Hakoun |
| Sequence Diagram, parts of class Diagram. | Sohaila Abdelazim Khalifa |
| Sequence Diagram, parts of class Diagram. | Sara Tamer Mohamed |