“Online Jewellery Store”

Software Design Specification

<27/3/2023>

< Yahya Al-Araby 201-800-643 >
< Mohamed Nasr 201-801-675 >
< Amr Abdelaziz 201-801-041 >
< Amr Elsherbiny 201-800-301 >

Prepared for
Instructor: Doaa Shawky, Ph.D.
Spring 2023

Table of Contents:

1- Introduction:

The project that we proposed is talking about an Online Jewellery store which is a market that aims to sell and buy Gold and Silver jewelry and Accessories. The purpose of this document is to define the software requirements for the development of an online jewelry store. The online store will allow customers to browse and purchase jewelry products from the comfort of their own homes. The system will be developed to provide a user-friendly interface, efficient and secure payment methods, and an easy-to-use product catalog.

2- System Features and Requirements:

2.1: Functional Requirements:

1- Registration and login
2- Add product (seller)
3- view product
4- delete product (seller)
5- add product to cart
6- search for product
7- customize profile
8- checkout
9- place order
10- approve order (admin)
11- add to favourites
12- similar products
13- contact us
14- FAQ page
15- Order confirmation by phone
16- Rate seller and buyer
17- view purchasing history
18- customize product (add engraving)
19- shipping & delivery
20- select amount
21- set price (seller)

2.2 Non-functional Requirement:

1- Security: user info such as credit cards and payment details
2- Performance & reliability: any mistake in the purchasing process would cost a fortune. The website must have less to zero errors

3- Timeline and Goals:

3.1- Timeline:

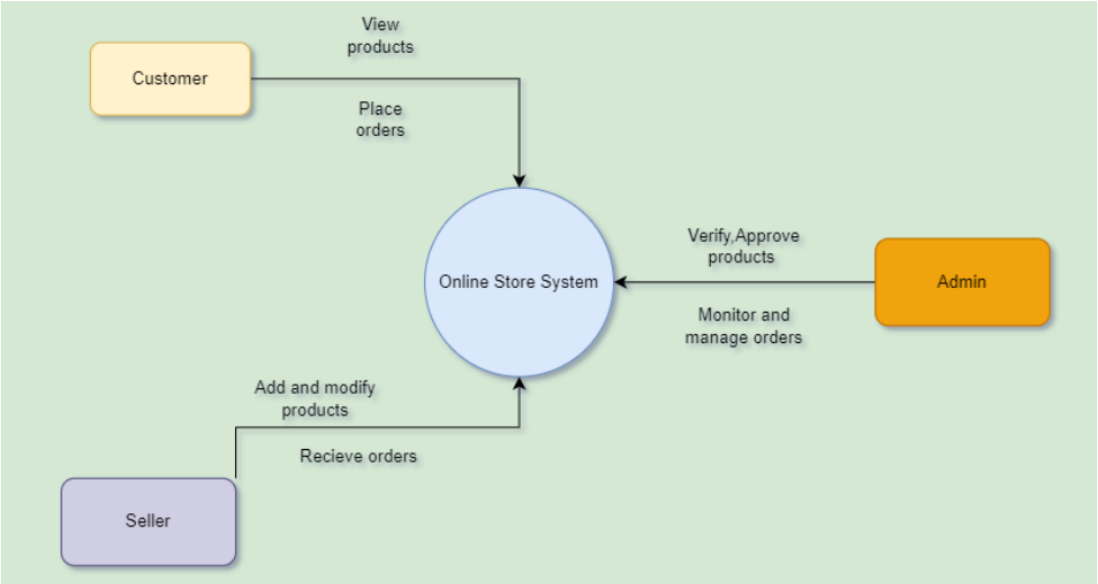
- Start date: March 27, 2023
- End date: May 18, 2023
- Milestones:
    - Design and development completed: August 31, 2023
    - Product catalog uploaded: September 30, 2023
    - Payment and shipping integration completed: October 31, 2023

3.2- Goals:

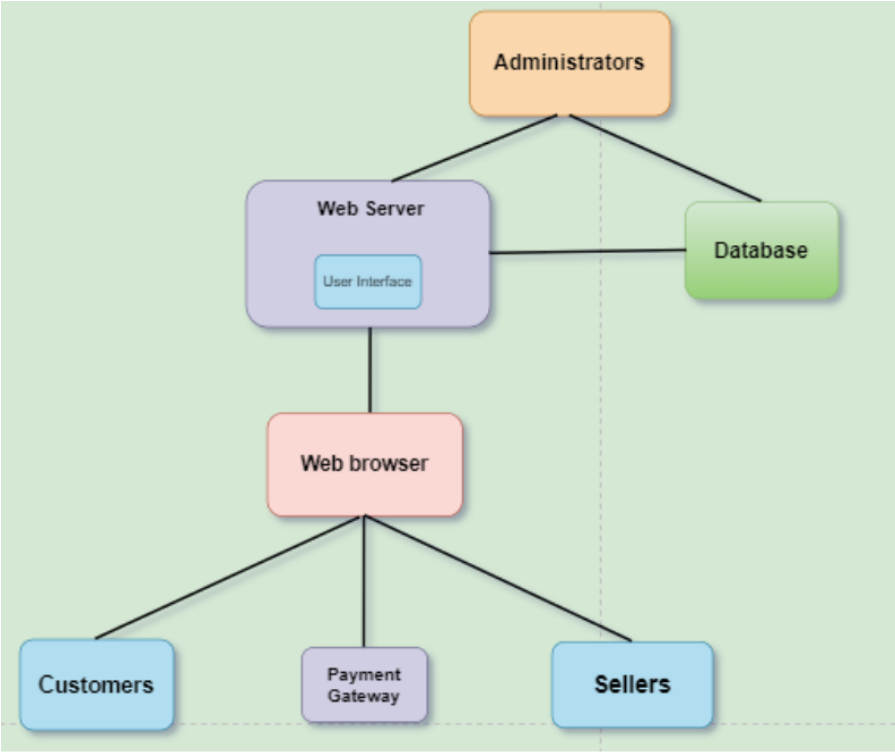
- Primary goal: To create an online jewelry store that provides a seamless shopping experience for customers and generates revenue for the business.
- Secondary goals:
    - Offer a wide variety of high-quality jewelry products that appeal to a diverse customer base.
    - Provide detailed product descriptions and high-quality images to help customers make informed purchasing decisions.
    - Ensure secure payment and shipping options to build trust with customers.

4- System Related Diagrams:

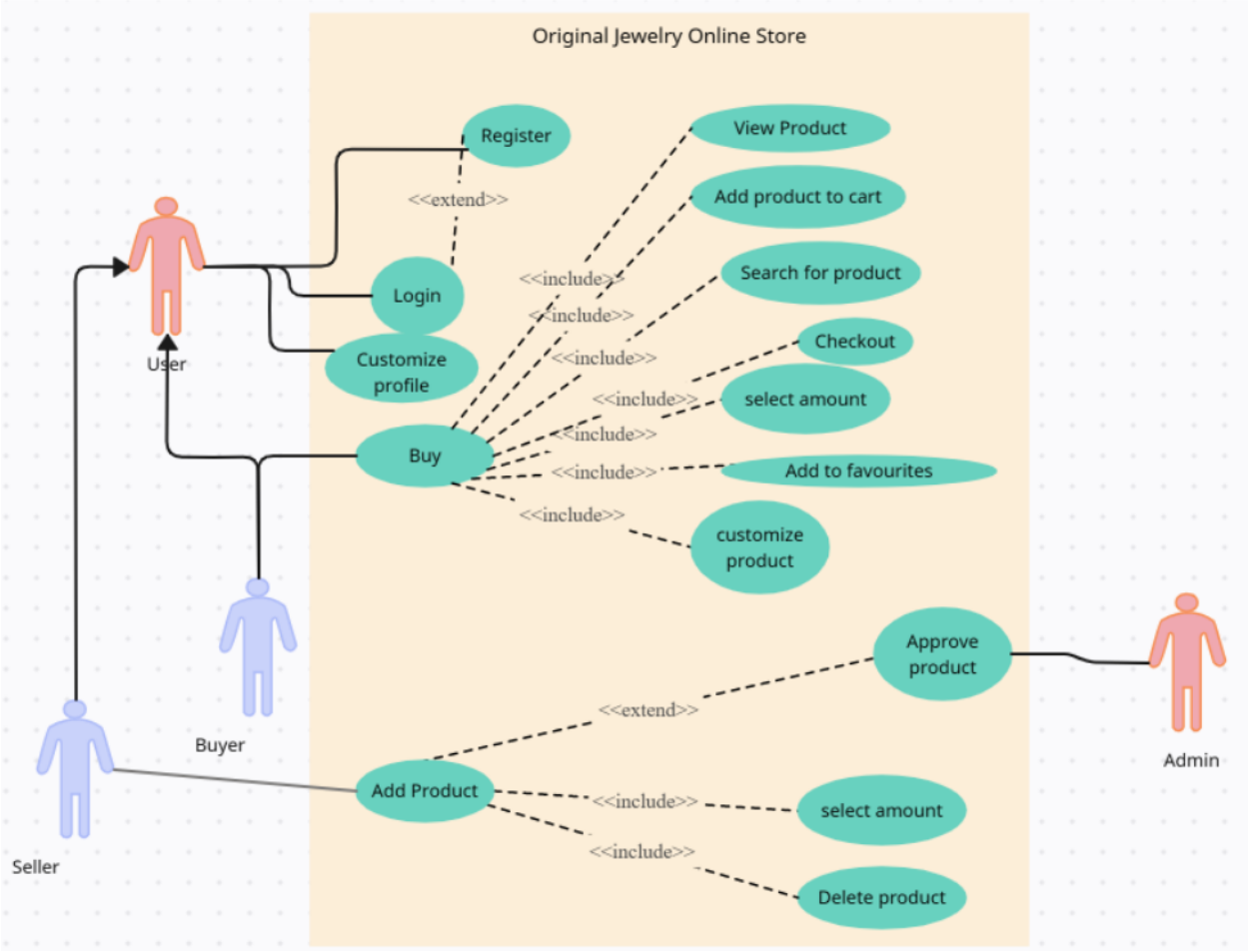
4.1-Context Diagram:



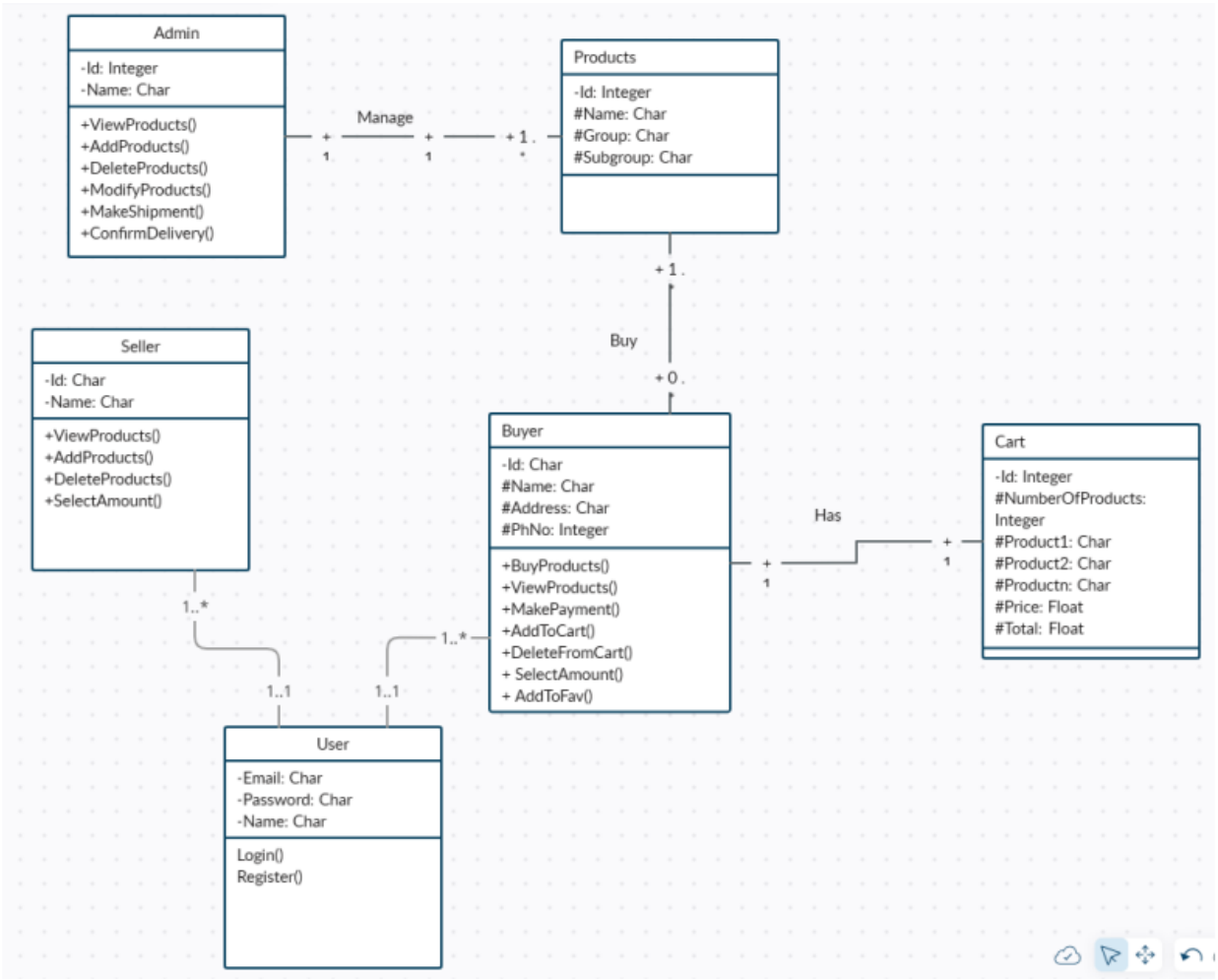
4.2- Concept Diagram:



4.3- Use Case Diagram:



4.4- Class diagram:



4.5 CRC cards for the two implemented sprints:

Product	
• Represent a product in the store's catalog • Store information about the product (name, description, price, image, etc.) • Handle product variations such as size, color, material, etc.	• Cart • Order • User

Edit card #1 × ↓

Cart	
• Contains user's purchases • Allow to add or remove products • Calculates the total price	• Product • User • Order

Edit card #2 × ↑ ↓

User	
• Store Info about user • Represent User on the website • Allows user to change his/her info	• Order • Cart • Product

Edit card #3 × ↑ ↓

Order	
• Represent an order made by user • Store info about order (Total price, products, shipping information) • Track the status of the order (processing, shipped, delivered)	• Product • Cart • User

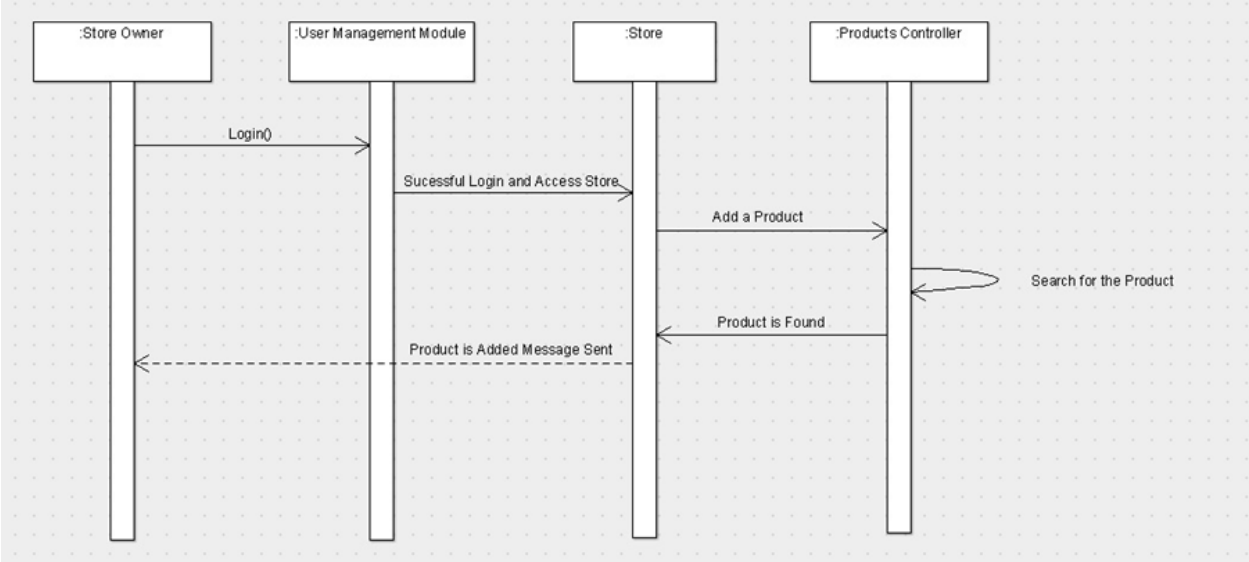
Edit card #4 × ↑

4.6 Product Backlog Items diagram for the three implemented sprints:

PBI
1- **A seller adds a Product to the store**
2- **A Customer Buying a Product**
3- **An administrator adding a product**

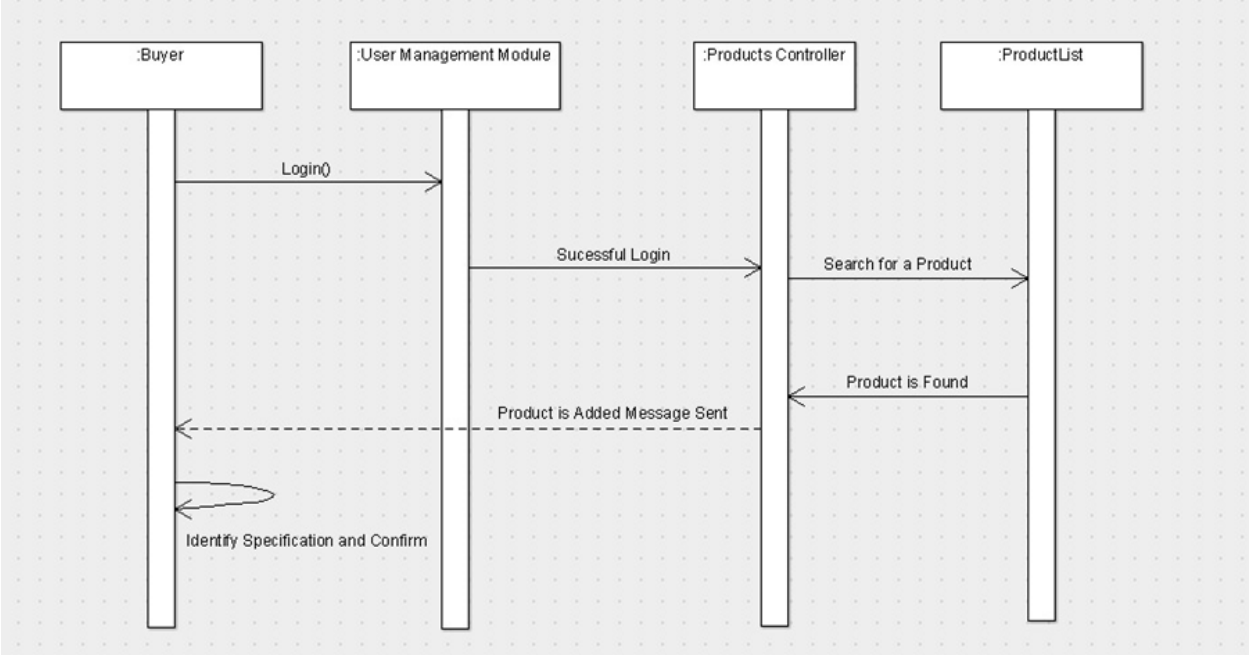
4.7: Sequence Diagram for the first sprint:

A seller adds a product to the store



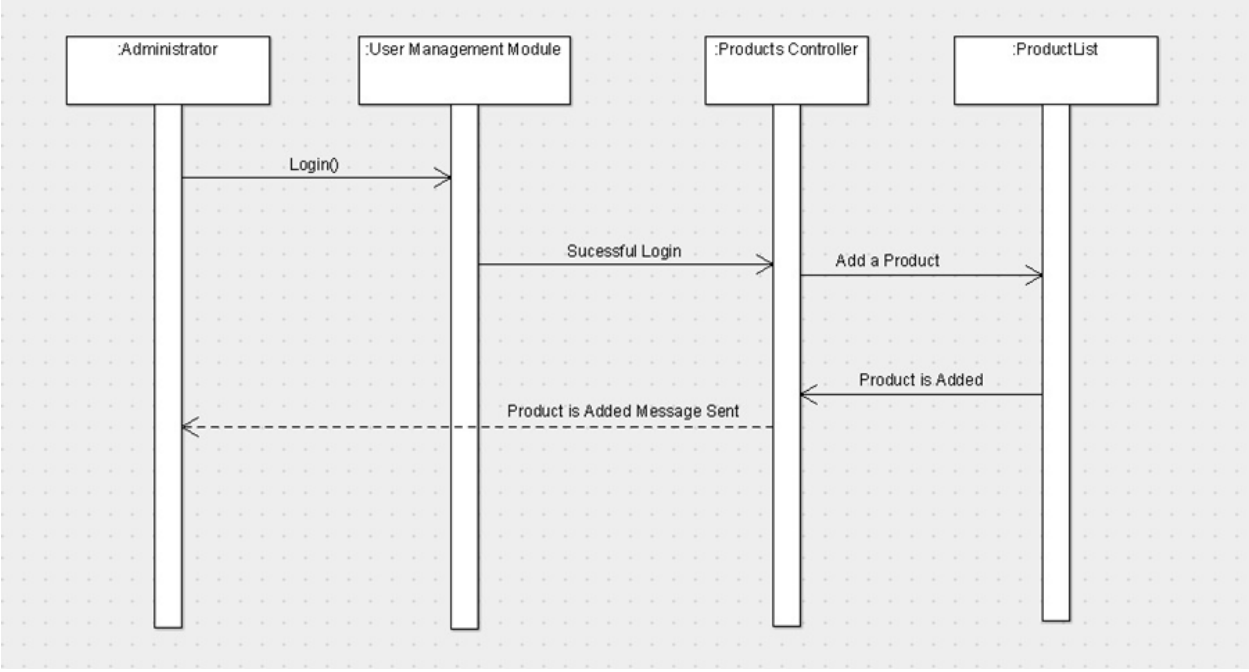
4.8: Sequence Diagram for the second sprint

A customer buying a product



4.9: Sequence Diagram for the third sprint

An administrator adding a product



5- Test Cases:

5.1- Test case for calculating sum of all products in the cart

```python
def test_calculate_total():
    # Create some Product instances
    p1 = Product(1, "Product A", "Group 1", "Subgroup 1", 19.99)
    p2 = Product(2, "Product B", "Group 2", "Subgroup 2", 29.99)
    p3 = Product(3, "Product C", "Group 1", "Subgroup 2", 39.99)

    # Create a Cart instance and add the products to it
    cart = Cart(1, [p1, p2, p3])

    # Check that the total price is calculated correctly
    assert cart.calculate_total() == 89.97
```

5.2- Test case for Register function in user class

```python
def test_register():
    # Create some User instances and a set of registered emails
    u1 = User("User A", "password1", "usera@example.com")
    u2 = User("User B", "password2", "userb@example.com")
    u3 = User("User C", "password3", "userc@example.com")
    registered_emails = set(["usera@example.com", "userb@example.com"])

    # Test registering a new user with a unique email
    assert u3.register(registered_emails) == True
    assert "userc@example.com" in registered_emails

    # Test registering a new user with a duplicate email
    assert u2.register(registered_emails) == False
    assert len(registered_emails) == 3
```

6- conclusion:

In conclusion, the development of an online jewelry store website was a challenging but rewarding project. The website provides a platform for customers to browse and purchase jewelry items, while also offering the ability for sellers to showcase their products. The implementation of user classes for buyers and sellers allowed for personalized experiences, including functions such as adding to cart, making payments, and adding products. Additionally, the use of a product class enabled efficient organization and retrieval of product information. Overall, this project successfully demonstrated the ability to create a functional and user-friendly e-commerce website for the sale of jewelry. Future improvements may include expanding the product offerings and implementing more advanced search and filtering options for customers.

7- Change Log:

May 1, 2023

Added the code for Classes: User, Seller.
Added some functions such as: login, add_to_cart
Updated the Use Case Diagram

May 3, 2023

Make the sequence Diagram
Added the code for classes: Buyer, product
Added some functions such as: Registration, view_product, delete_product, add_to_favourites.

May 5, 2023

Added the Admin Class as well as the functions required for it.

May 6, 2023 Made the Database, as well as some improvements to the GUI.
Finalize phase 3 of the project.