SOFTWARE DESIGN DESCRIPTION

PREPARED by:

Izmir Institute of Technology

Computer Engineering Department

Project Group 1:

Çağatay İBA

Buket ARSLAN

Mert KARACA

Ayşegül ÜNLÜER

Can ROLLAS

Umutcan CEYHAN

İbrahim Halil ÖZTÜRK

Barış ERDAL

TABLE OF CONTENTS

1. OVERVIEW

1.1 Scope

This article describes the main entity of the `*Fish auction Project*` which has been designed in the project management course. Software engineering has a vast majority in various projects. The governance of a software project is significant. Insomuch as Planned project is not enough to revive the project without software management and design. The design of software and governance must be certain.

This article, which is **Software Design Description**, describes the structure of the Fish Auction Project and implementation of the project via precise steps according to Software Engineering Manner/Literature. Besides, this Software Description Document includes quotes from Software Requirements Specification (**SRS**) and includes diagrams, schemes (UML, IDL, etc.) which are commonly used in Software Engineering and Computer Science.

1.2 Purpose

This SDD article designates the software system design of the `*Fish Auction Project*`. The software design of this project is based on the **SRS** document submitted in the last period of the 2021 Project Management Course. Regardless of how effortless or problematic it is to divert an idea into a project, it will make this process more manageable for the developers to progress together and understand. This document intends to be the guidance of developers, project team, testers, etc. Throughout development, implementation period.

Over time, the DevOps team concludes the main structure and framework of software via this **SDD** document.

1.3 Intended Audience

A) Market Audience:

With reviewing the market competitors there are many websites and organizations in cyberspace, but there are no applications or web pages that sell with the auction system. Fisher put their fish images on social media (Facebook) for

an online auction. On the other hand, bigger companies sell fish with just online shopping and stable prices. According to *Wikipedia.org* ` A target audience is the intended audience or readership of a publication, advertisement, or other message catered specifically to said intended audience. '[1] An intended audience is the most significant elective process in various projects. The audience is mostly fishermen and their clients around Balıklıova. Fishers can upload their fish to an online auction and sell it for the best price via cooperative.

B) Project Audience:

Basic knowledge of programming must be covered. Target audiences listed below are necessary for completing the project.

- Project Team
- Backend developers
- Frontend developers
- Data management Team
- Review Team

1.4 Summary and Context

This project and article describe the *Fish Auction Project.* Necessary introduction and information have been given in sections 1.1, 1.2, 1.3. Main purpose of this SDD is to give information to the team, programmers, reviewers, mentors etc... This project provides online auctions to the fishers. Eventually, this SDD shows those steps in general:

- To explain structures
- To determine required system resources.
- It will be used to assess the impact of requirement changes.
- To help create test cases.
- It will be used to validate the requirements-related complaint.
- Assisting maintenance activities.

2. Definitions and References

2.1 References

- G02_Software Requirements Specifications Report
- lucidchart.com
- btkakademi.gov.tr
- Researchgate
- app.patika.dev

2.2 Glossary

- **Contract**: A legally binding document agreed upon by the customer, cooperative and fishermen.
- **Customer**: Person or persons, who join the auction to buy fish by making bids.
- **Fishermen**: Person, or persons, who sell their fish using the online auction system.
- **Cooperative**: An establishment that organizes the auction and that interacts with customers and fishermen for various purposes.
- **Database:**Collection of all the information monitored by the system.
- **User**: Customer,fishermen or a cooperative member.
- **IEEE:** The Institute of Electrical and Electronics Engineers, Inc.
- **GMT:** Greenwich Mean Time
- **MTBF:** Mean Time Between Failure
- **DBMS:** Database Management System
- **mongoDB :** NoSQL database structure and engine
- **CyberSpace :** According to the World Internet Consortium, it is a world wide web.
- **JSON** : Model of structure in universal data structures.

3. Conceptual model for software design descriptions

3.1 Software design in context

The task of the project is to bring the fish auction online via a web application. This web application is aimed to reduce the process time and workload, as well as to increase the number of buyers reached. This project will be implemented with Java and Javascript Frameworks (for Server side: Java Spring, for Storage: MongoDB and for the Front-End we will use ReactJS),CSS and Bootstrap as stylesheet languages for the templating we will use HTML.

3.2 Software design descriptions within the life cycle

3.2.1 Influences on SDD preparation

The software requirements specification is typically the key software life cycle product that drives a software design. The project's design is determined by the requirements in the SRS (product perspective, functional and non-functional requirements, and interface requirements), as well as the demands of the stakeholders.

3.2.2 Influences on software life cycle products

During the preparation phase of the SDD and/or the project's implementation phase, some decisions may change as needed.

3.2.3 Design verification and design role in validation

Verification and validation will be tested after preparation of the test cases. All system parts will be tested. It will be checked for whether the requirements are fulfilled or not.

4. **Design Description Information Content**

4.1 Introduction

This documentation provides basic knowledge of design of the software. Necessary tools, etc. has been provided in this documentation for implementation e.g. user interfaces, sequence diagrams etc..

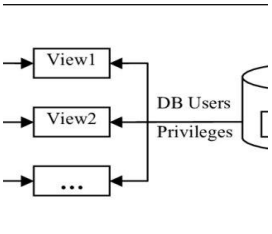
4.2 Sdd Identification

This auction provides knowledge for fisherman's can upload their fish and sell via cooperative.There are three types of interfaces that can be examined in OOP manner.This SDD describes software design for whoever reads this (programmer etc.).

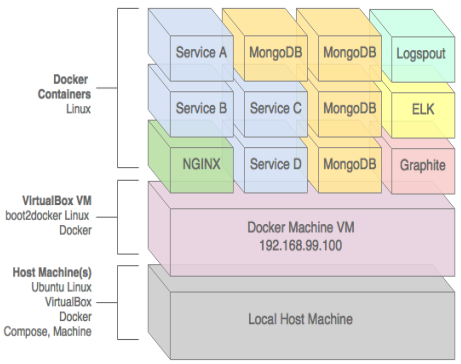
4.3 Design stakeholders and their concerns

Stakeholders of this project are cooperative members, customers and head-customers. Application of the online auction system will satisfy basic roles of the stakeholders via computer etc. It provides an online sale and payment system with basic secure algorithms. Additionally the fisherman does not see the customer and when purchase is completed whether or not according to money transfer from the system.

4.4 Design view



Design of this application based on database interaction (Update, insert, delete) systems and containerized basic application MVC visualized by a markup language dynamically or not dynamically.



4.5 Design viewpoints

This part of the (4.4) document describes and indexes what kind of viewpoints are used in this document and their explanation for this project under the 5th content.

- Logical Viewpoint
- Information Viewpoint
- Interface Viewpoint
- Interaction Viewpoint

4.6 Design rationale

Our design system is based on responsive and containerized MVC. Moreover, we use the system of the JSON - noSQL structure for our design. With those implementations our design will be responsive for every server or deployment.Eventually our database interaction system makes three main roles which are Insert, Update, Delete with our restful api. In next chapters or contents viewpoints will be examined under those definitions.

- ☐ Logical Viewpoint will examine the relations between the classes and the logic points.
- ☐ Information Viewpoint will examine the relations between database microservices and storing.
- ☐ Interface Viewpoint will examine the basics of the interfaces and their roles.
- ☐ Interaction Viewpoint will examine the interaction of web applications.

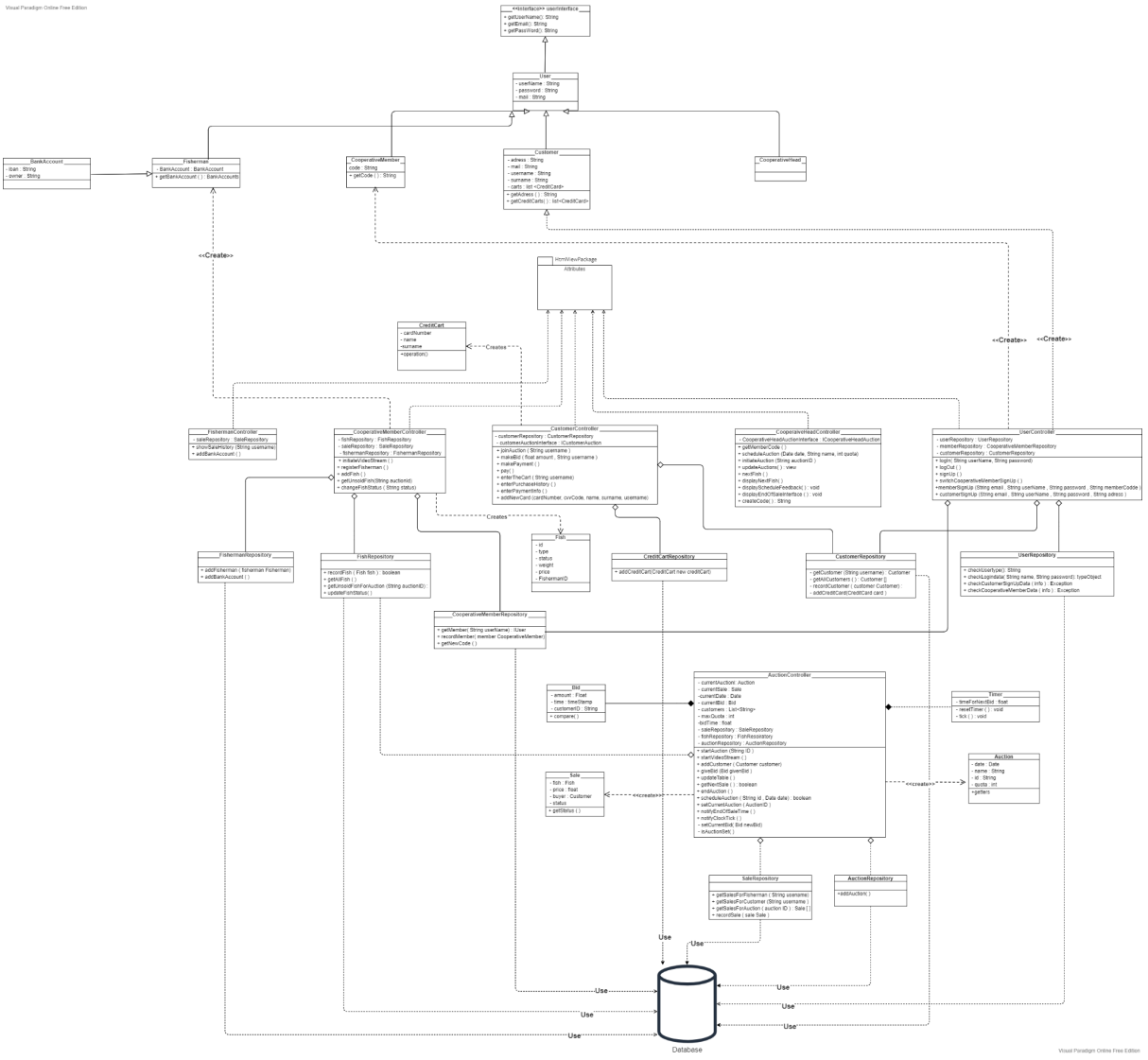
4.7 Design languages

- ☐ **UML (UNIFIED MODEL LANGUAGE)**
- ☐ **ER DIAGRAM (ENTITY RELATIONSHIP)**
- ☐ **UML COMPONENT DIAGRAM**
- ☐ **UML SEQUENCE DIAGRAM**

5.DESIGN VIEWPOINTS

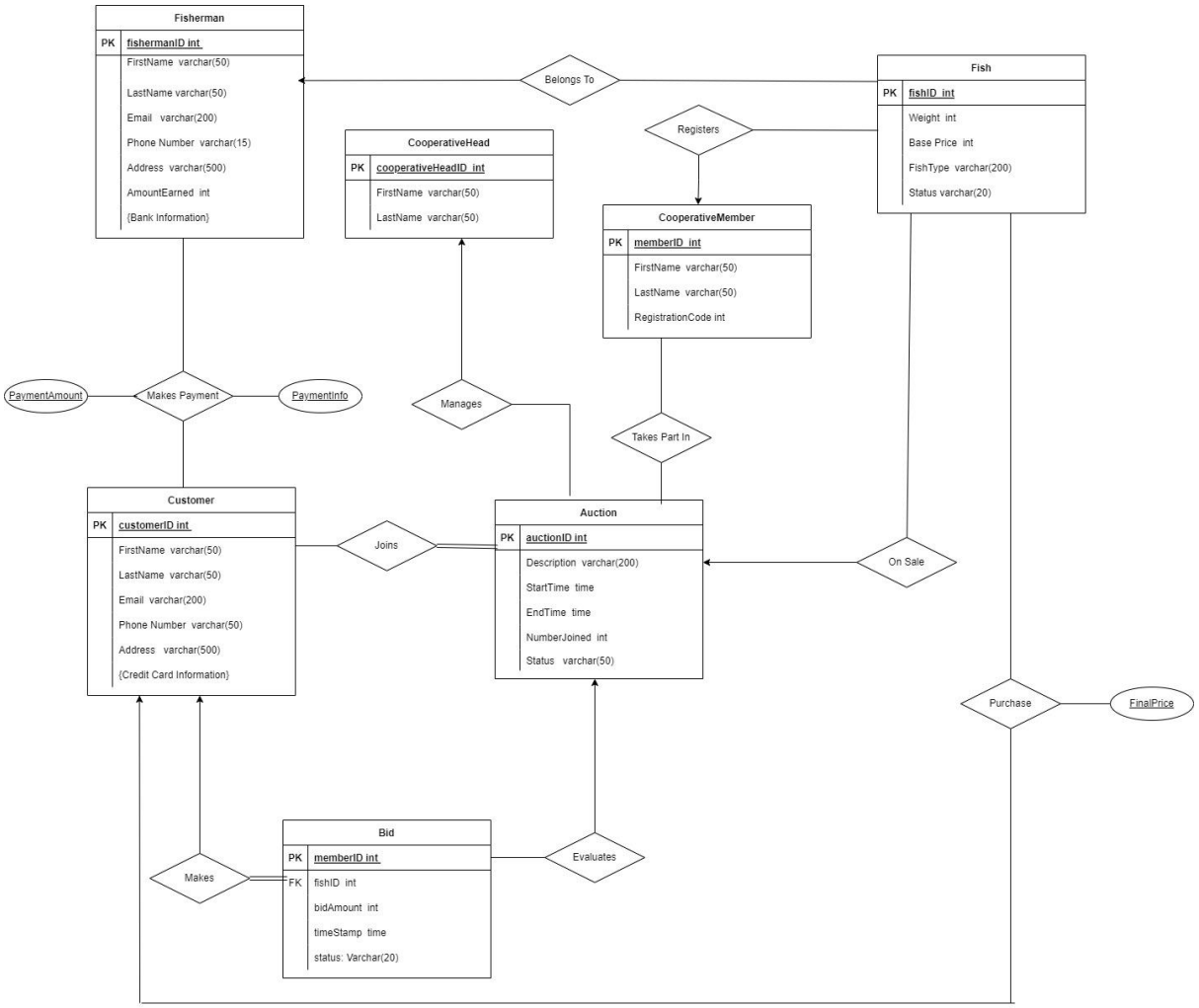
5.1 Logical Viewpoint

5.1.1 UML CLASS DESIGN



5.2 Information Viewpoint

5.2.1 ER Diagram



5.2.2 Relational Schema

Fisherman(<u>fishermanId</u>, name, surname, email, phoneNumber, address, amountEarned)

Bank(<u>bankId, fishermanId</u>)

fishermanId references fishermanId in Fisherman table

CooperativeHead(<u>cooperativeHeadId</u>, firstName, lastName)

CooperativeMember(memberId, firstName, lastName, registrationCode)

Fish(fishId, weight, basePrice, fishType, status, fishermanId, memberId, auctionId, customerId)

fishermanId references fishermanId in Fisherman table

memberId references memberId in CooperativeMember table

Auction reference auction in Auction table

customerId references customerId in Customer table

Customer(customerId, name, surname, email, phoneNumber, address)

CreditCard(creditCardId, customerId)

customerId references customerId in Customer table

Bid(bidId, timeStamp, bidAmount, status, customerId, auctionId, fishId)

customerId references customerId in Customer table

auctionId references auctionId in Auction table

fishId references fishId in Fish table

Auction(auctionId, description, startTime, endTime, numberJoined, status, cooperativeHeadId)

cooperativeHeadId references cooperativeHeadId in CooperativeHead table

TakesPartIn(auctionId, memberId)

memberId references memberId in CooperativeMember table

auctionId references auctionId in Auction table

Joins(customerId, auctionId)

auctionId references auctionId in Auction table

customerId references customerId in Customer table

Payment(fishermanId, customerId, paymentInfo, paymentAmount)

fishermanId references fishermanId in Fisherman table

customerId references customerId in Customer table

Purchase(fishId, customerId, finalPrice)

fishId references fishId in Fish table

customerId references customerId in Customer table

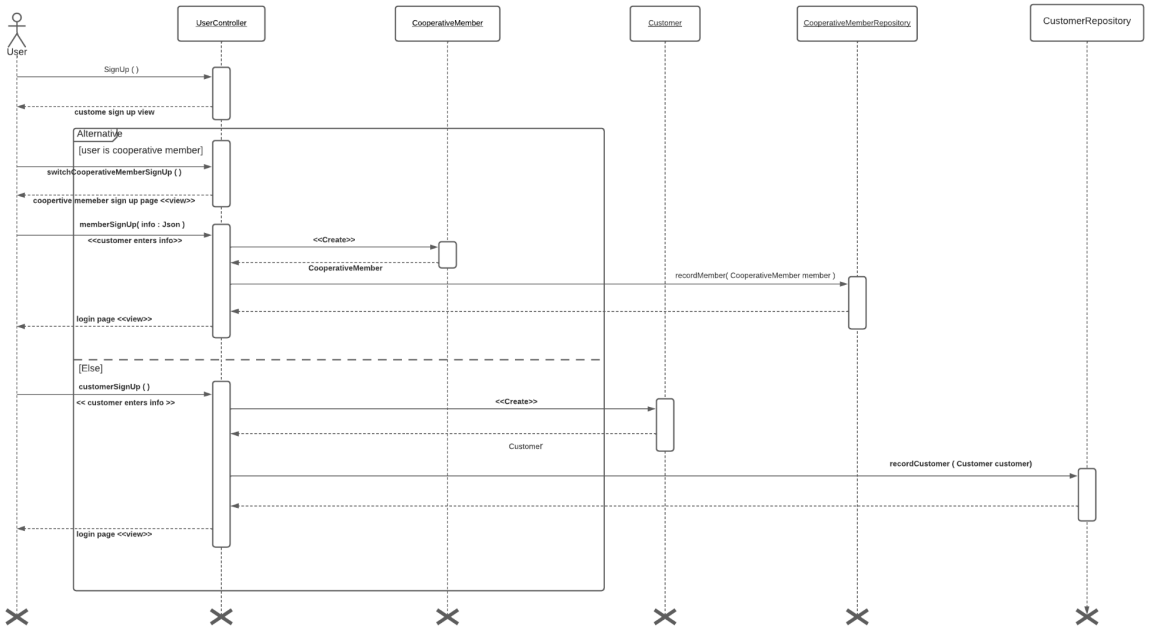
5.2.2 Database Design Notes

- This database is designed according to BCNF and 3NF.

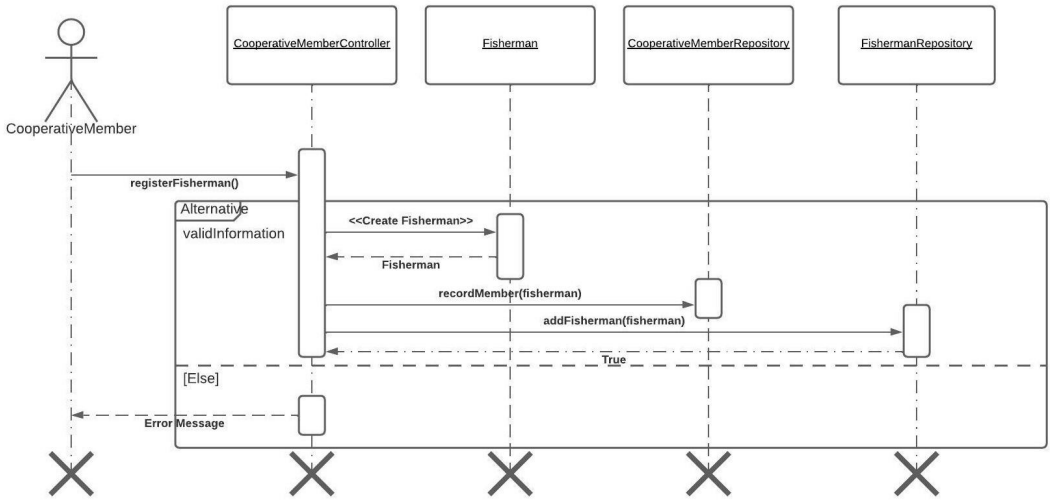
5.5 Interaction Viewpoint

5.5.1 General Sequences

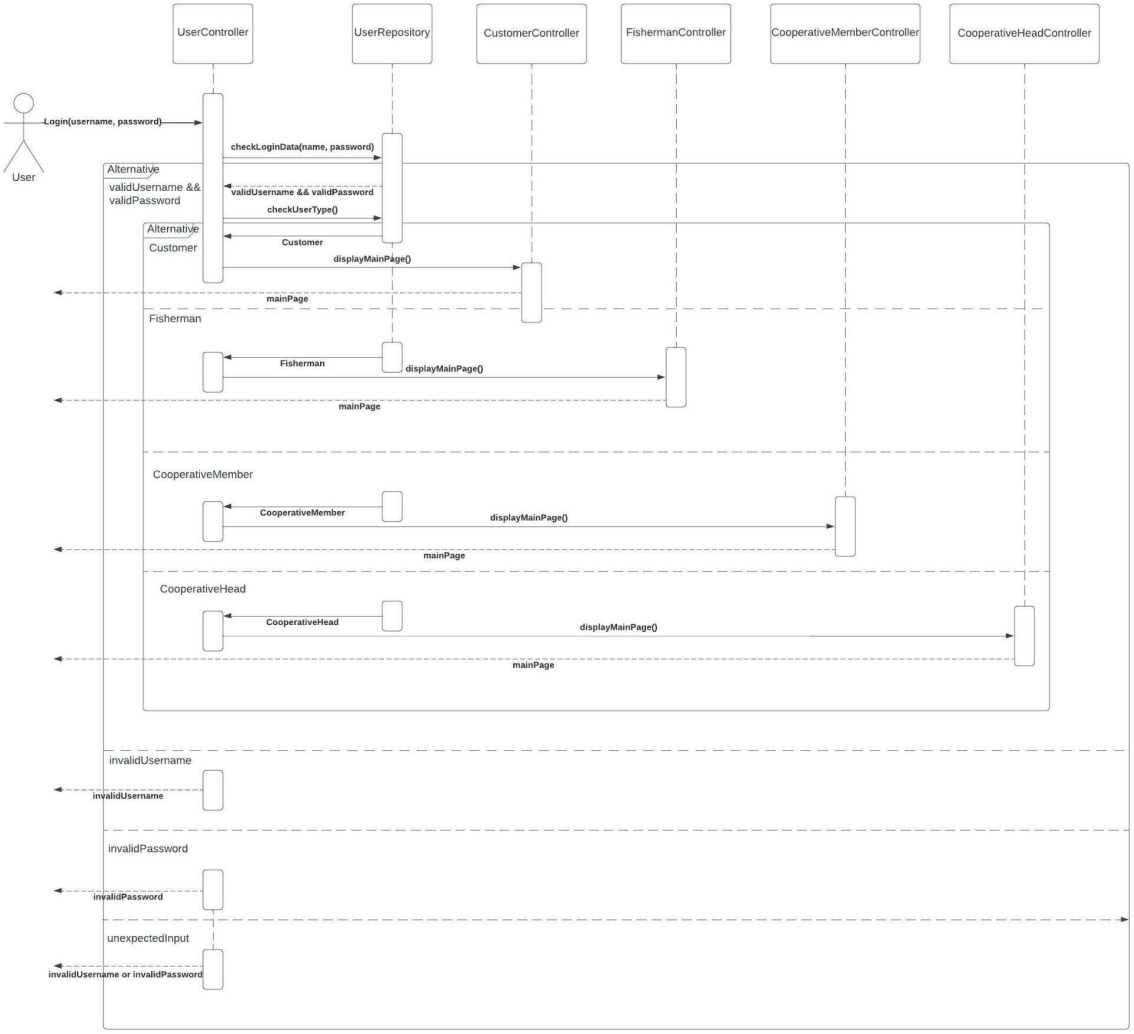
5.5.1.1 Sign Up



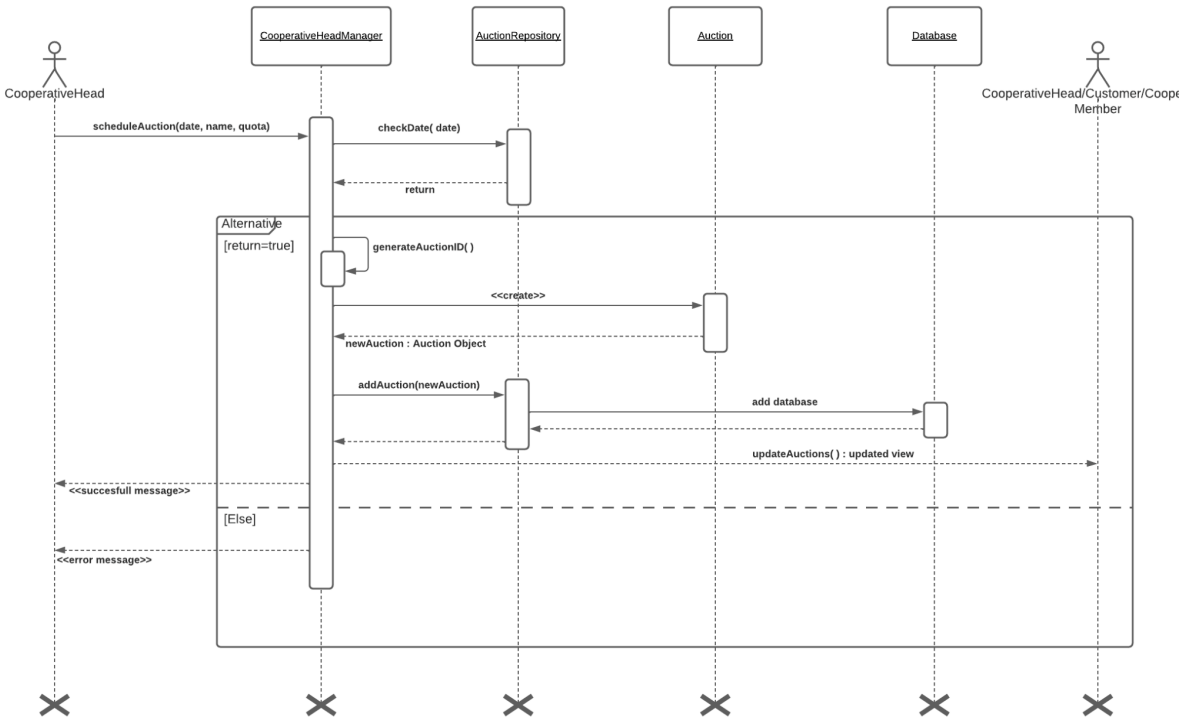
5.5.1.2 Register Fisherman



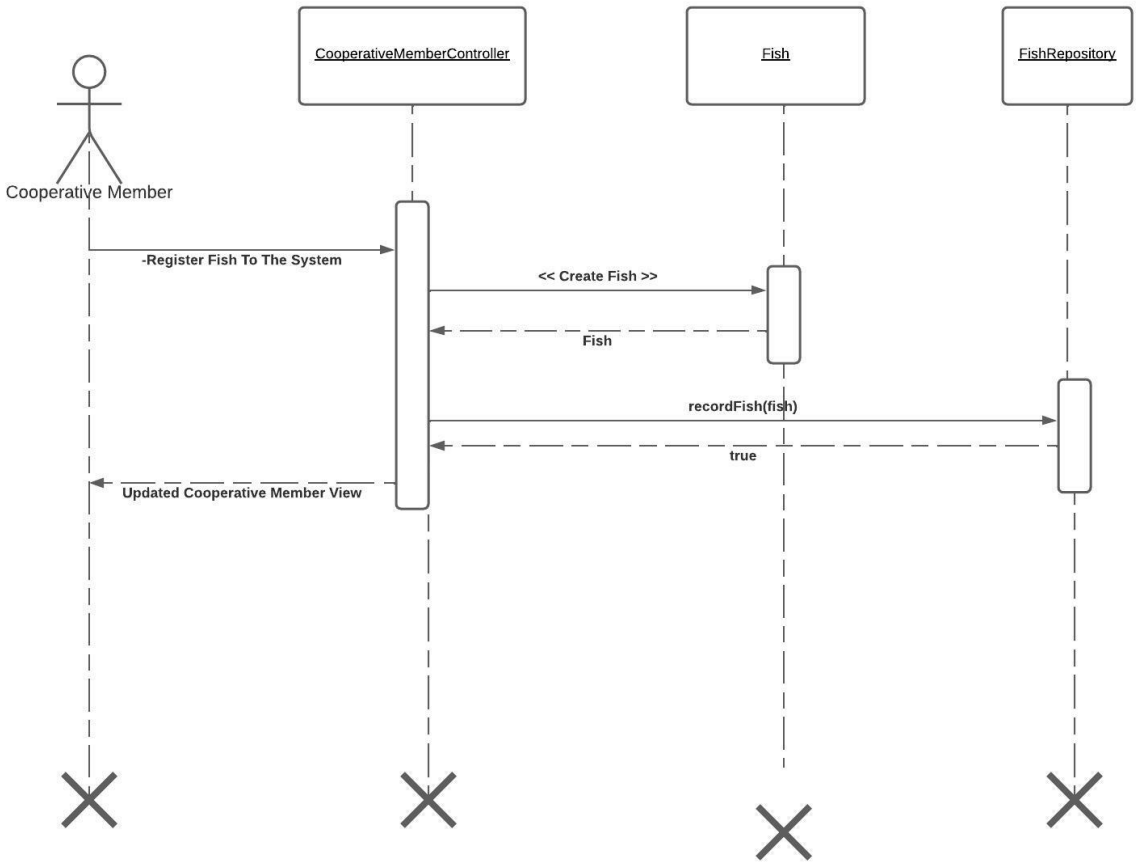
5.5.1.3 Login The System



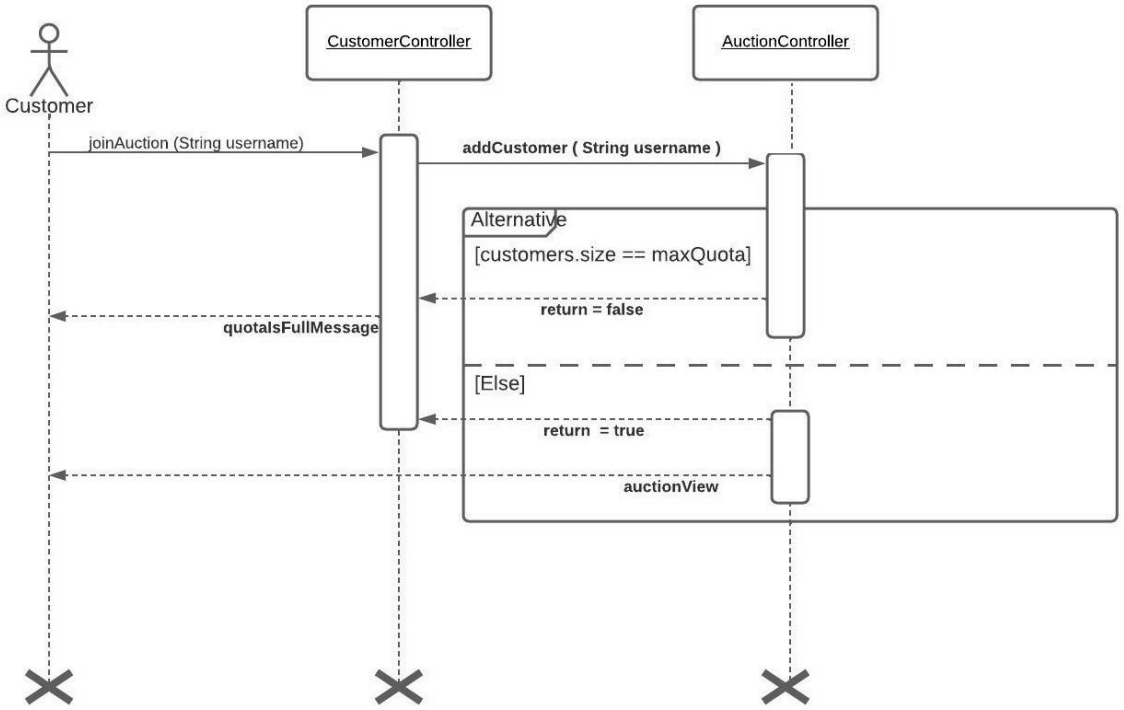
5.5.1.4 Schedule Auction



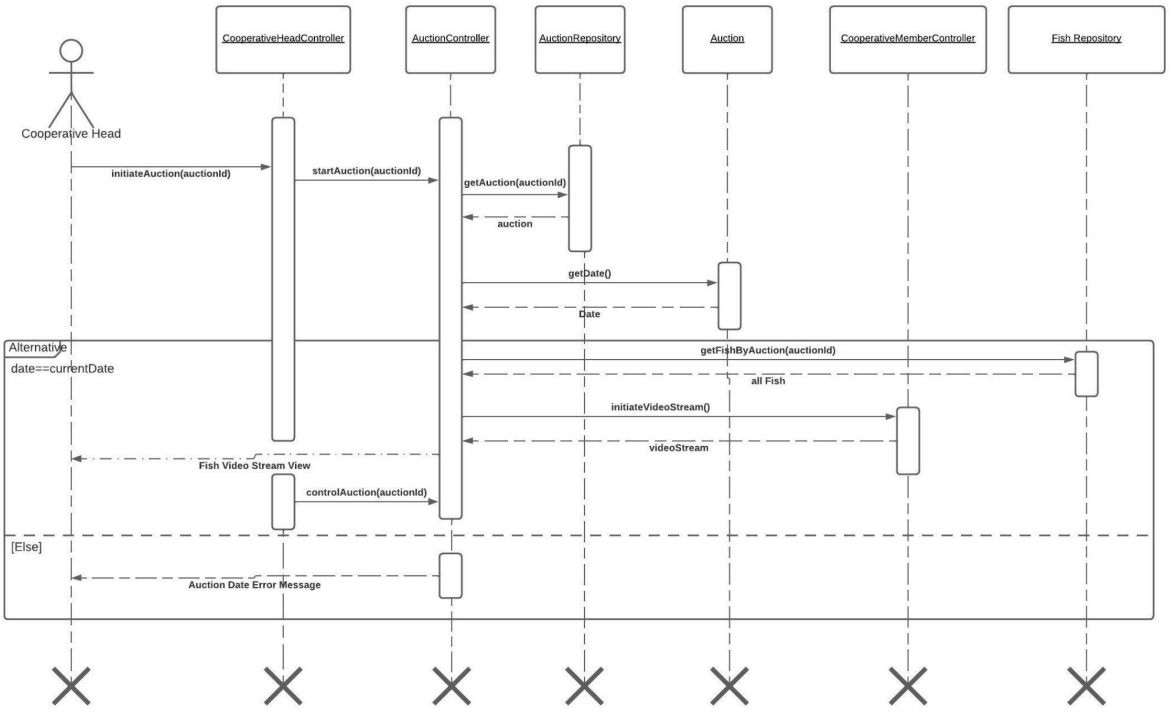
5.5.1.5 Register Fish



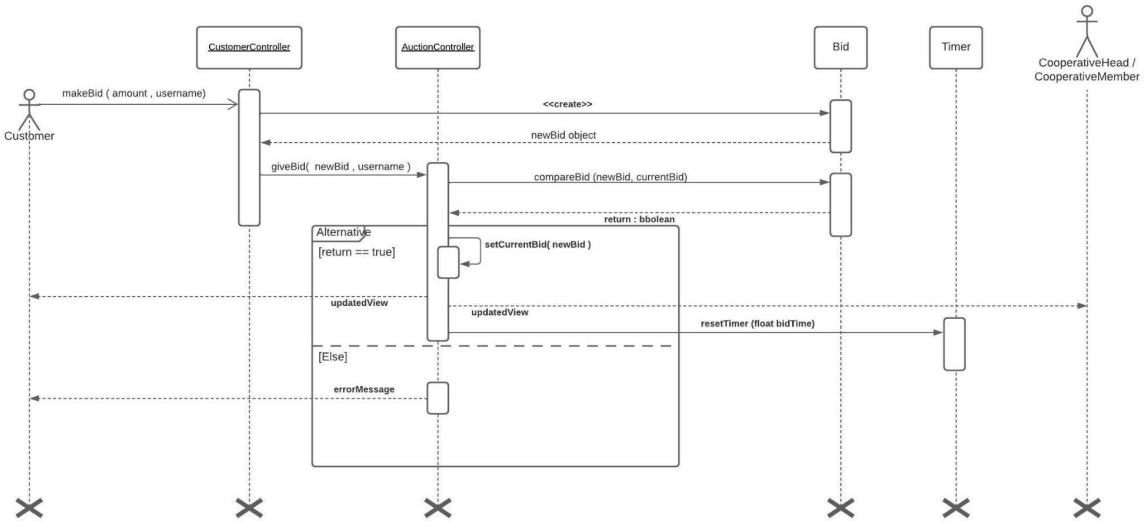
5.5.1.6 Join Auction



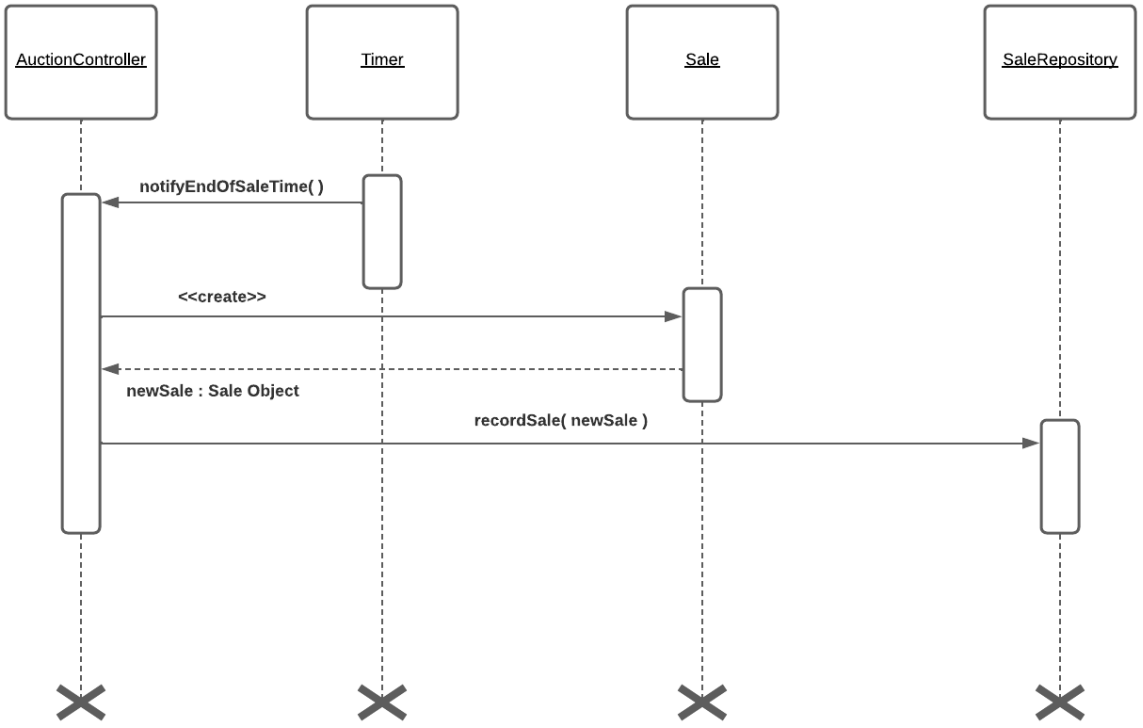
5.5.1.7 Start Auction



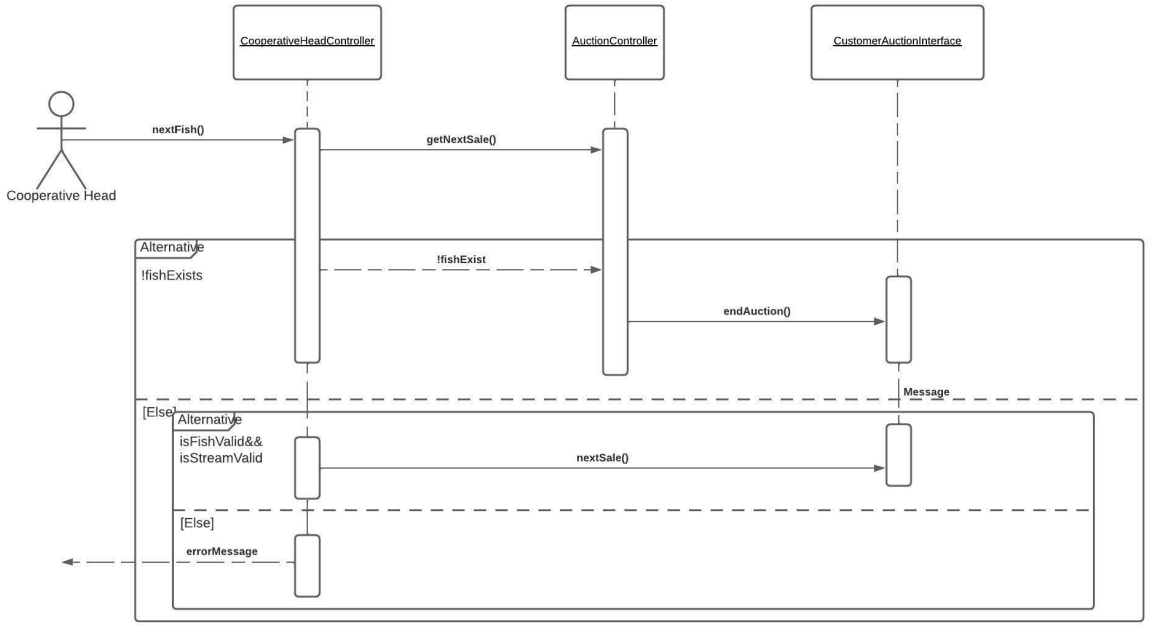
5.5.1.8 Make Bid



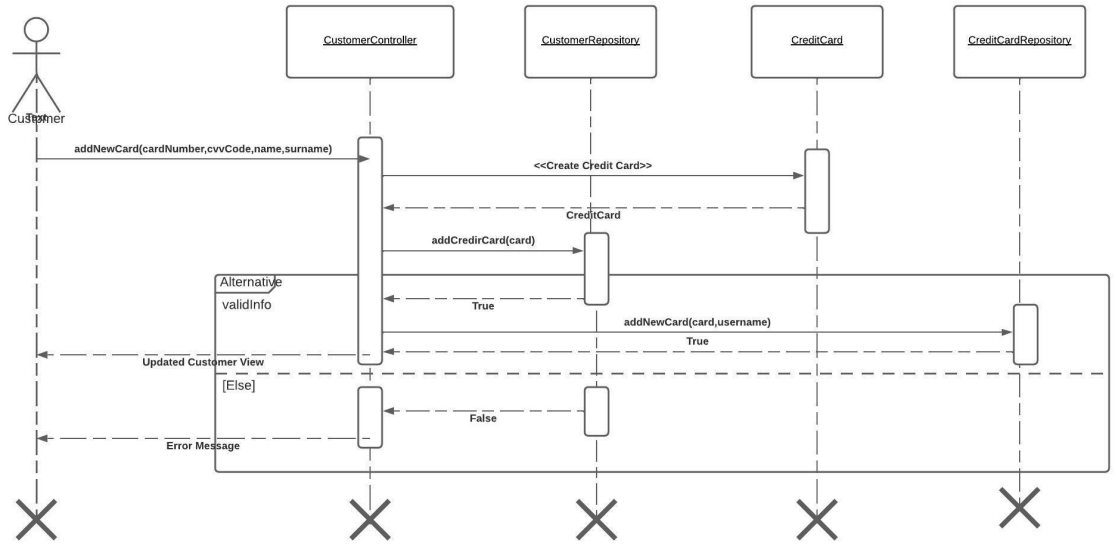
5.5.1.9 End Of Sale



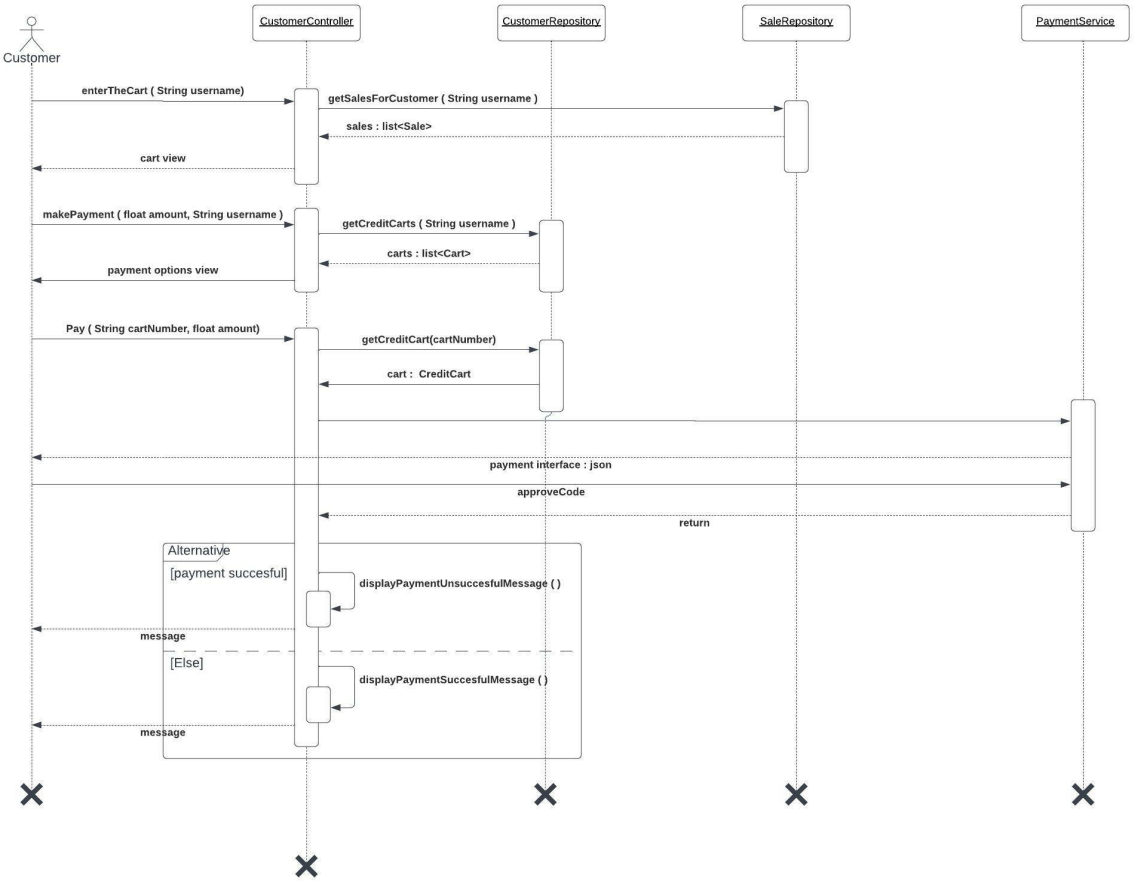
5.5.1.10 Move on to the Next Fish



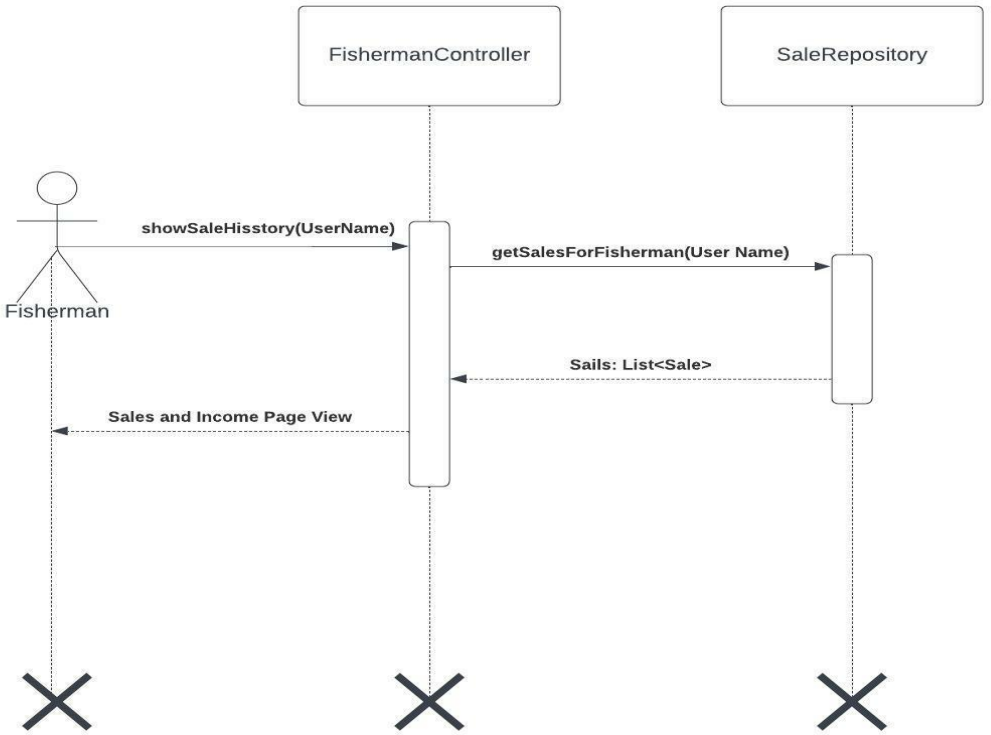
5.5.1.11 Register Card



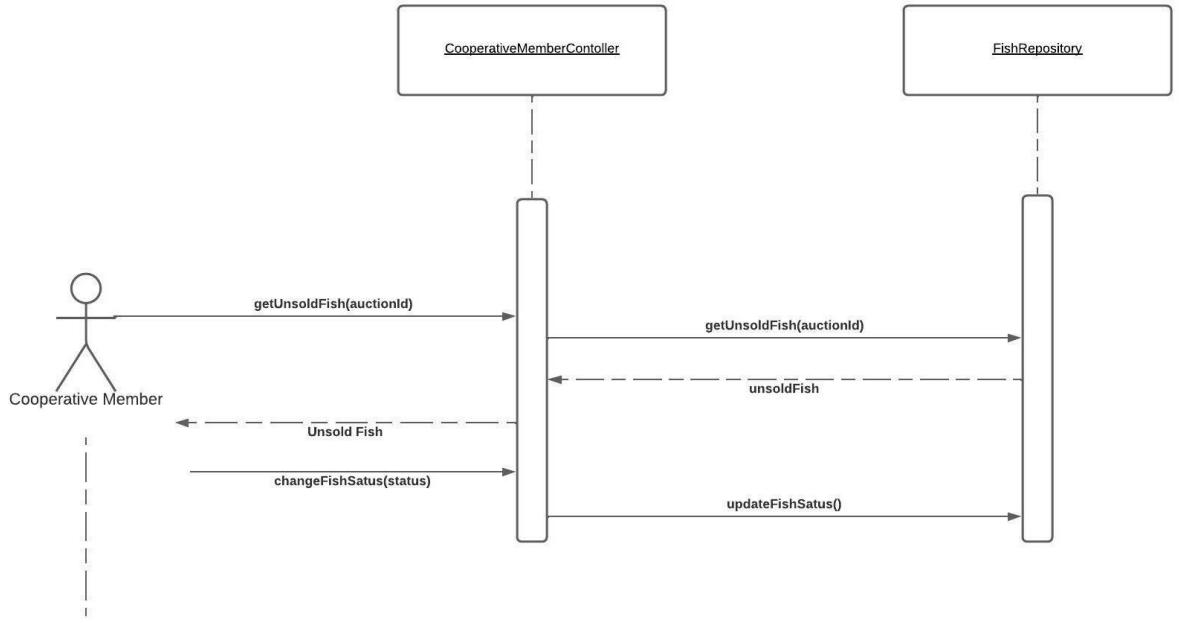
5.5.1.12 Make Payment



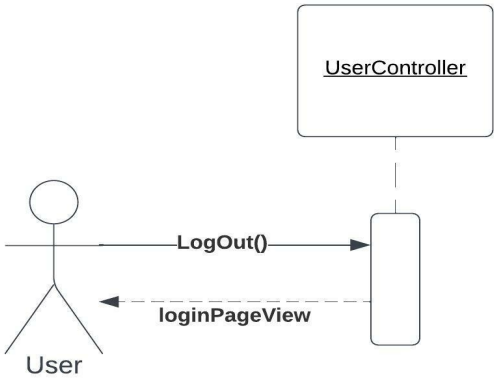
5.5.1.13 Keep Track of Fisherman's Sales and Cash Situation



5.5.1.14 Return Unsold Fish



5.5.1.15 Logout The System



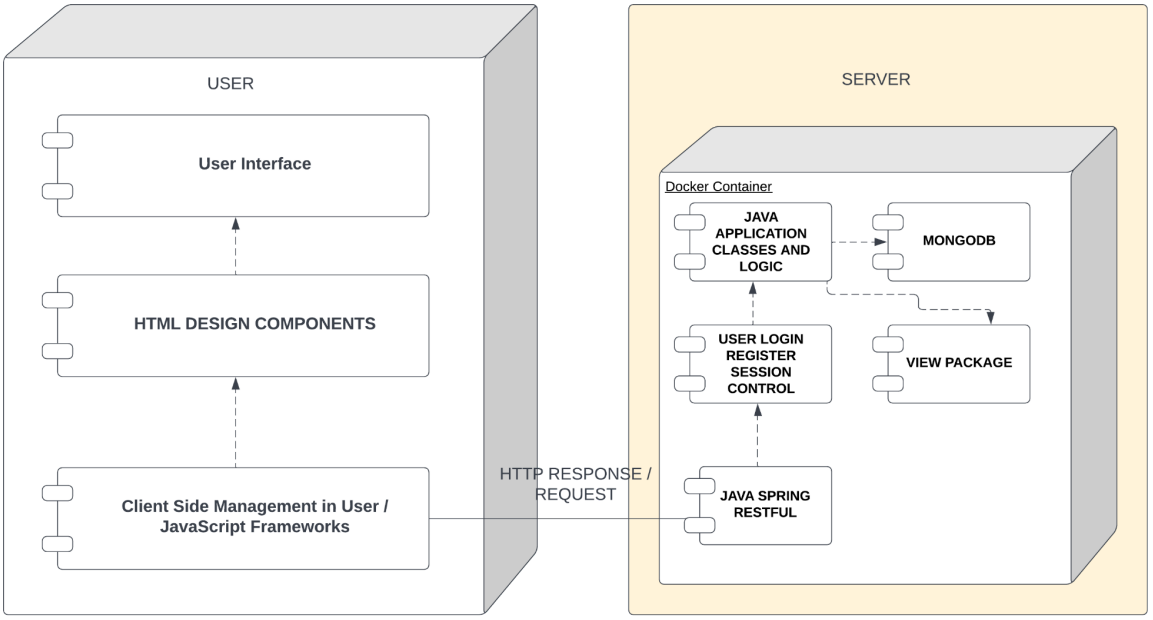
5.8 Interface Viewpoint

Design Concerns

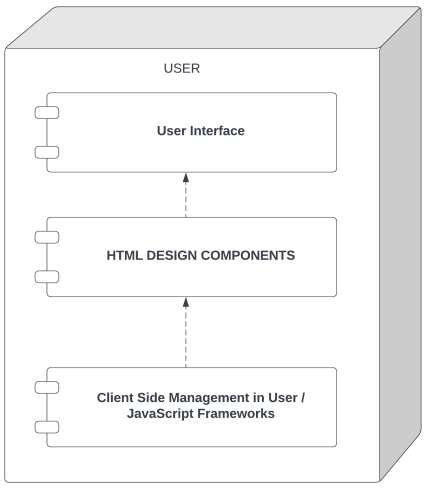
This part of the document provides general information for Programmers and Designers about what they must know to develop the project. On the other hand this part of the document provides view descriptions to designers, programmers etc. There are common types of interfaces used in this project. The other interfaces are the branches of those common interfaces.

5.8.1 SYSTEM INTERFACE

- This point gives information about correct usage of services provided by a design object. This design object contains basic knowledge of basic web application components which are *USER and SERVER.* If the project completed with this design application is ready for deployment.Component Diagram of the system is downside below.



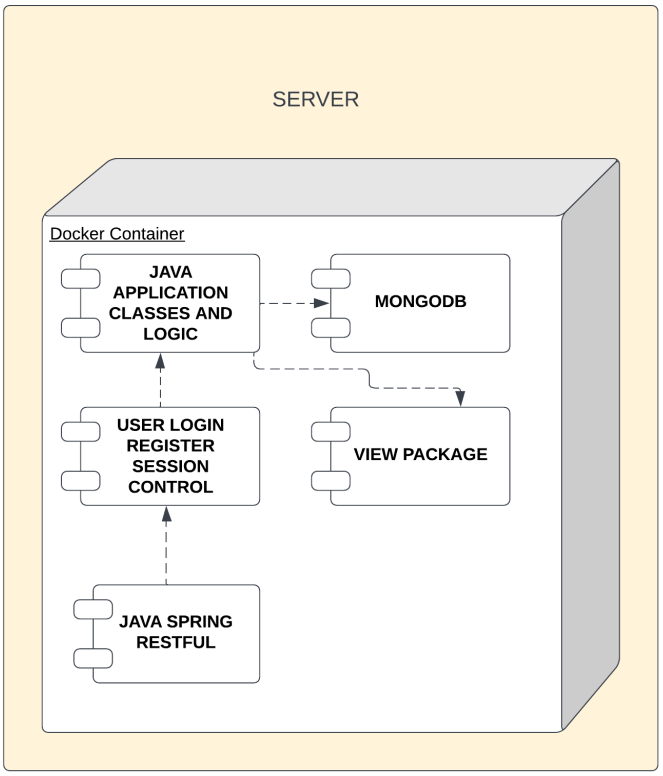
5.8.1.1 USER COMPONENT



- In this part of the document it describes the communication between interfaces from the user component.User is basically a client which is Browser from Computer,Phone,TV etc. Users make requests to the server via http and get responses.Received response will be processed with javascript or javascript frameworks and creates dynamic HTML files for user and client can see the interactive compiled HTML as a view. For example, the user (Fisherman) tries to login into the system and send requests to the system via

username and password.The server sends a response and client gets it and user logs into system and interface via dynamic html will be compiled in browser html engine.Generally *JavaScript has browserside embedded methods which based on async and states.*

5.8.1.2 SERVER COMPONENT



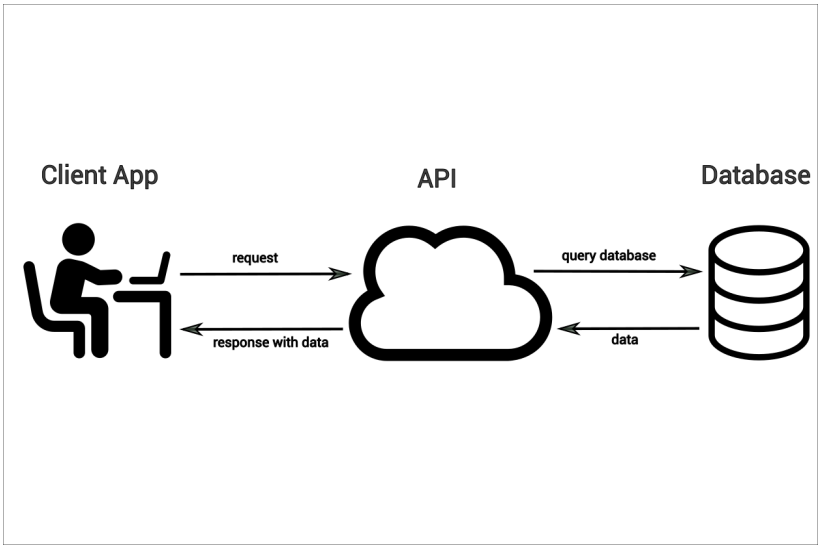
- Server components are based on MVC communicating with each other via interface or classes.MongoDB is a database for storing necessary data in application. View Package consists of html templates communicating from one side which is java application classes. The Java application part communicates db , viewPackage and others. Java spring generally executes the server and http protocols which serve outside of the docker container via 0.0.0.0 ip.Methods of application logic based on uml and it can be found on logic viewpoint.Mongodb has query methods sth like SQL delete, insert ,update etc. Java spring framework has serverside embedded methods like routing Via http. View pack generally includes the templates and static files for application.

5.8.2 How user interfaces communicating and User Types

- Users communicate with other classes without direct access and information leak. Which is based on the API system of JAVA. Users generally use a class for comm. which is based on user type in uml diagrams. There are 4 types of user
    - Fisherman
    - Cooperative Member
    - Cooperative Head
    - Customer

- Every type of user can access data managed according to restrictions about the users. For example Fishermen can not start auctions but cooperative members can.With the system design individuals can access the system with clients and communicate.

-> Scribble Drawing for user communication description figure according to UML class diagram (Note that this is scribble for individuals it does not gives exact solution for system it provides basic insight)



5.8.3 User Interfaces Description

5.8.3.1 CUSTOMER INTERFACE DESCRIPTION

- **makeBid**
    - This method takes one argument as float named as Amount and it delivers to the necessary objects or directories.
- **joinAuction**
    - This method takes one argument which is Customer type object and method provides the joining to the live auction.
- **enterTheCart**
    - This method provides showing the card and making operations.

    Method takes a string argument ID of the customer.
- **EnterPurchaseHistory**
    - This method shows the history of the customer.
- **EnterPaymentInfo**
    - This method provides entering the paymentInfo.
- **AddCreditCard**
- This method provides entering creditCard infos as a string from the user.

5.8.3.2 Cooperative Member Interface Description

- **InitiateAudioStream**
    - Start a stream from the cooperative member.

5.8.3.3 Cooperative head interface Description

- **GetMemberCode**
    - Get memberCode from the System.
- **ScheduleAuction**
    - This is scheduling the auction from the system.
- **InitiateAuction**

- Starting the auction on the system.

- **NextFish**

  - This method passes to the other fish for auction.

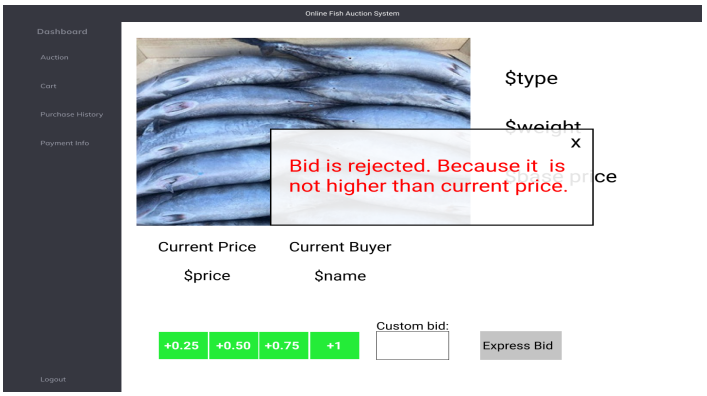
5.8.3.4 Fisherman Interface Description

- **showSaleHistory**
  - Shows the sales history from the database to the fisherman user.
- **addBankAccount**
  - Add bank account to the system

5.8.4 User interfaces examples with GUI

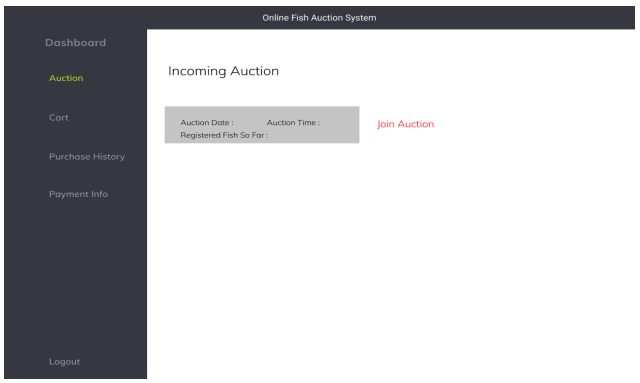
5.8.4.1 Customer making bid

- Express bid button activates the makeBid method from the user interface.



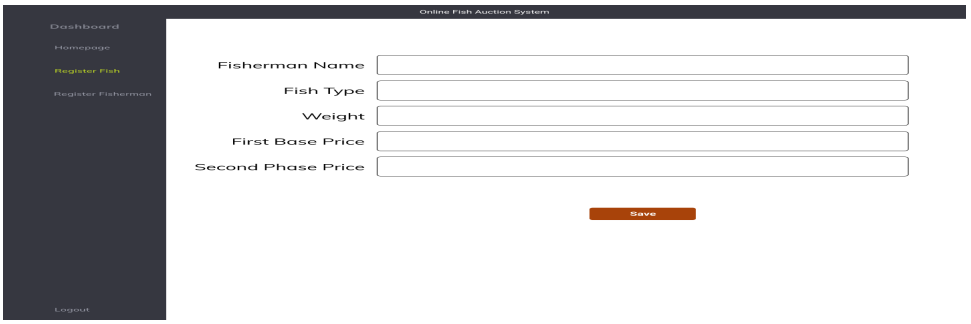
5.8.4.2 Customer Joining the auction

- The Join Auction button activates the joinAuction method via interface.



5.8.4.3 Cooperative Member register the fish

- This form and button provides adding data to the database via user interface.



5.8.4.4 Cooperative Head start the auction

- This button provides a live auction stream to the customer user interfaces via cooperative head interface.