Software Engineering

Sport Saloon Application Report

Furkan GÖZÜKARA

Yusuf Can İbişoğlu

195050021

1. Which one will be your project model waterfall, incremental or reuse-oriented development?

I will use incremantal model in project because it will always interact with users and will get feedbacks .

2. Full story of the project features

Yusuf has a Fitness Saloon and it has many subs. He checks how many of subs are using the gym in motion, GYM subs can pay their monthly payment from application. Also in this gym there are many trainer this application provides asystem that can subs and trainer arrange a date for “Personal Trainer” service. GYM has a GYM BAR in this bar provides subs protein based drinks(shakes) and if subs desiree they can buy the powder or ingredients of those drinks to make same thing in home from GYM BAR. Prices of those ingredients are listed on this applications BAR part.

In view of Subs:

Burak is a sub to Yusuf’s GYM and he has the application, he can check how many days left until his subscription is going to end and if he desiree he may pay for monthly payment for the subscription, also check his paid payments on application. He can view the price list of updated GYM BAR. Also Burak can check if the gym it is crowded or not (there will be an status notification)

3. Detailed Test Case of Project Features

1)Login

Test 1
Input text representing nickname(e-mail) text representing password **Tests:** Inputs valid? Nickname exists? Password matches with nickname? **Output** OK or error(Invalid input, user doesn’t exist or password is wrong)

2) Register

Test 2
Input Text Representing; First name, Last name, e-mail, Password, Password conffirmation, Date of birth, Adress **Tests:** Inputs valid? Does e-mail exists? Does Password provide security requiremnets and matches with confrimation password? **Output** OK or Error (Invalid input, Password is not secure, Password doesn’t match, this user is already exist)

3)Logout

Test 3
Input A user **Tests:** Is user authenticated? **Output** **OK or Error**

4)Add-Update-Delete sub(For Admin) (if someone is registred and payed he will automaticly be a sub)

Test 4
Input Choosing the related sub and deleting it from the system. Creating a sub with related credentials Update the credentials of the sub Choosing Related Sub and giving to Sub a Trainer Role **Tests:** Is sub stil exist? Is sub created in database? Cechking the related Sub that have changed the credentials by admin, if the related credentials changed? Checking from the DB if the related Sub turned in to Trainer **Output** **OK, Error**

5)Add-Delete Product and Price regulation (For Admin)

Test 5
Input Delete Product from the BAR ADD product to BAR Change Price to related prodcut **Tests:** Is product stil exist? Is product created in database? Checking if the price changed ! **Output** **OK, Error**

6) Change User Password

Test 6

Input Current Password New Password New Password (Confirmation) **Tests:** Password valid? Do New Passowords and New Password (Confirmation) matches? Do New Password ensure security processdure? **Output** **OK, Error(Passwords doesn't match, Current password is not ture, Please use safer passwords)**

7) Sub Status View

Test 7
Input Crawl related Sub's payments from Database Crawl related Sub's subscription expire date form DB Crawl related Sub's latest date with Trainer Crawl the number of people in the gym at the moment and make the related denisty alert(Calm,Normal,Crowded) **Tests:** Checking the Database payments and amounts of related Sub Checking the Database for expiring date of related Sub Checking if the Sub have a date with any Trainer? Check the number of people from the dataset and check the denisty with already apointed values **Output** **True or false**

8) Trainers Status View

Test 8

Input
Crawl related Trainer's Schedule from the DB
Crawl related Trainer's weekly date amount from the DB
Crawl related Trainer latest date with Sub

Tests:
Checking the Database payments and amounts of related Sub
Checking the Database for how many date Trainer had
Checking the Trainer's upcoming date

Output
True or False

9) Appoitment system(for SUB)

Test 9

Input
Choosing the Trainer in the related time and day
Canceling The PT(SUB Should make a comment in here why he cancelled this)

Tests:
Checking if the Trainer is free in the mentioned day
Check if the appointment deleted from DB

Output

Ok, Error(You can't take appointment in this pirticular date, false)

10) Appoitment system(for Trainer)

Test 10

Input
Locking the days that can't give the PT(physical trainer) service
Canceling The PT (Trainer Should make a comment in here why he cancelled this)

Tests:
Checking if the days locked by Trainer is locked too in the DB.
Check if the appointment deleted from DB

Output

Ok, Error(You can't take appointment in this pirticular date, false)

11) Appoitment system(for Admin)

Test 10

Input
Deleting The Appointment
Creating the Appointment

Tests:
Checking if the related Appointmentis deleted in the DB.
Checking if the Appointment created in the DB.

Output

True or False

4. Full requirements definition of the project

A.User requirements

a. The admin, can add/update/delete a product from the Bar list and the price.

b. The admin, can delete/create appointtment

c. The admin, can add/delete/manipulate the Sub,Trainer (ALL USER)

d. The SUB, can make an appointment.

e. the Sub, can pay the monthly payment and check the bill

f. the Sub, can view the denisity status of the GYM

g. the Sub, can view the expire of subscription

h. the trainer, can declare locked appointment days.

i. the trainer, can view appointments and whole week appointments of himself

j. the trainer can delete appointment

B. System Requierments

a. Authentication

b. accsesing the data

c. user interaction

5. Nonfunctional requirements of the project

A. Product Requirements:

a. Application should be responsive as possible as.

B. Organizational Requirements:

a. Admins should have different pages and menu to manage system

b. Costumers shouldn’t accsess admin area.

C. External Requirements:

a. The products in the BAR, The GYM itself, Trainers should have exist.

6. Nonfunctional requirements metrics table

Proeprty	Measure
Speed	**Processed transactions/second, Internet Speed**
Size	**Mbytes**
Reliability	**Mean time to failure** **Probobility of unavailability** **Rate of failure occurence** **Availability**
Robustness	**Time to restart after failure** **Percentage of events causing failure** **Probability of data corruption on failure**
Portability	**Percentage of target dependent statments** **Number of target systems**

7. Full requirements of each part of the project

a. Login: The system verifies whether the user is present within the system or not. Subsequently, if the user is found, it verifies the accuracy of the password provided. In the event that the password is accurate, the user is granted authentication. Conversely, if the password is incorrect, the system generates an error message along with an explanation.

b.Register: The system verifies the existence of the email in the system. If the user doesn't exist, it checks if the password meets the required criteria. If the password matches the criteria, a user account is created and authenticated.

c. Paying the subscription payment: the system needs a payment system to get payments.

d. Role: In fundementals of this project there is “roles” actually in every projecet does have this. In this case there’s three roles, SUB’s to GYM, Admin, Trainers

e.Appointment System: an Appointment system should have developed to arrange appointments for SUBS with Trainers.

f.Product List in the Bar: Admin will add/delete/regulate the products in the BAR page.

g.Update/Manipulate/Delete/Create user: Admin can manipulate the user list and give them roles and may take it away if it is neccesary.

h.Change password: Sub’s and Trainers should able to change their own passport.

8. Full structured requirements of each part of the project

Population Denisty in GYM	
Function	Calculating the population denisty in the Gym.
Description	Calculating the population denisty in the Gym and giving the related alert for this
Inputs	Crawling instand population from the DB
Source	DB(user Inputs)
Outputs	comparing it with already setted values
aim	Letting Sub’s know if the Gym is too crowded or not.
Action	Giving the related alert such “Crowded,Normal,Calm”
Requirements	Sub’s have to use the app and give the notification that they are in GYM or they should use the Key Card when they get in.
Pre-condition	Using the key card or app
Post-condition	Sub may decide go instantly to Gym or later.

9. Tabular computation of your each function/model of the software

Condition	**Action**
Denistiy level >=4	**Crowded**

4>Denistiy level >=3	**Normal**
3>Denisty level	**Calm**

10. Detailed Scenarios for Project

Login

Initial Assumption	User Wants to sign in to system
Normal	User will write correct e-mail and password, then it will be authenticated.
WHAT CAN GO WRONG:	User can write wrong e-mail. User can write wrong password. User can be lock out.
OTHER ACTIVITIES	Get in the related Sub/Trainer/Admin Interface
SYSTEM STATE ON COMPLETION	User will be authenticated, AccessFailedCount will be zero

Register

Initial Assumption	User wants to sign up in the system

Normal	User will write first name, last name, e-mail, password, date of birth, and address.
WHAT CAN GO WRONG:	User can write invalid data. User can write an e-mail which is already taken. Passwords can't match.
OTHER ACTIVITIES	Get in the related Sub/Trainer/Admin Interface
SYSTEM STATE ON COMPLETION	User will be authenticated, AccessFailedCount will be zero

Add-Update-Delete Sub(Admin)

Initial Assumption	Admin wants to manipulate the SUB's
Normal	Admin can attend and manipulate Subs as he desiree
WHAT CAN GO WRONG:	Admin may delete someone accidentally. Admin may manipulete someone accidentally.
SYSTEM STATE ON COMPLETION	System will be changed.

Add-Delete Product and Price regulation (For Admin)

Initial Assumption	Admin wants to manipulate the Products and prices

Normal	Admin can attend and manipulate Product and prices as he desiree
WHAT CAN GO WRONG:	Admin may delete someone accidentally. Admin may manipulete someone accidentally.
SYSTEM STATE ON COMPLETION	System will be changed.

Sub Status View

Initial Assumption	Sub wants to check the Sub status view page
Normal	Sub can see the info's given in the page.
WHAT CAN GO WRONG:	There may be mistake in the DB There may be a discconnection between app and DB
SYSTEM STATE ON COMPLETION	Renewing the Sub Status View page

Trainers Status View

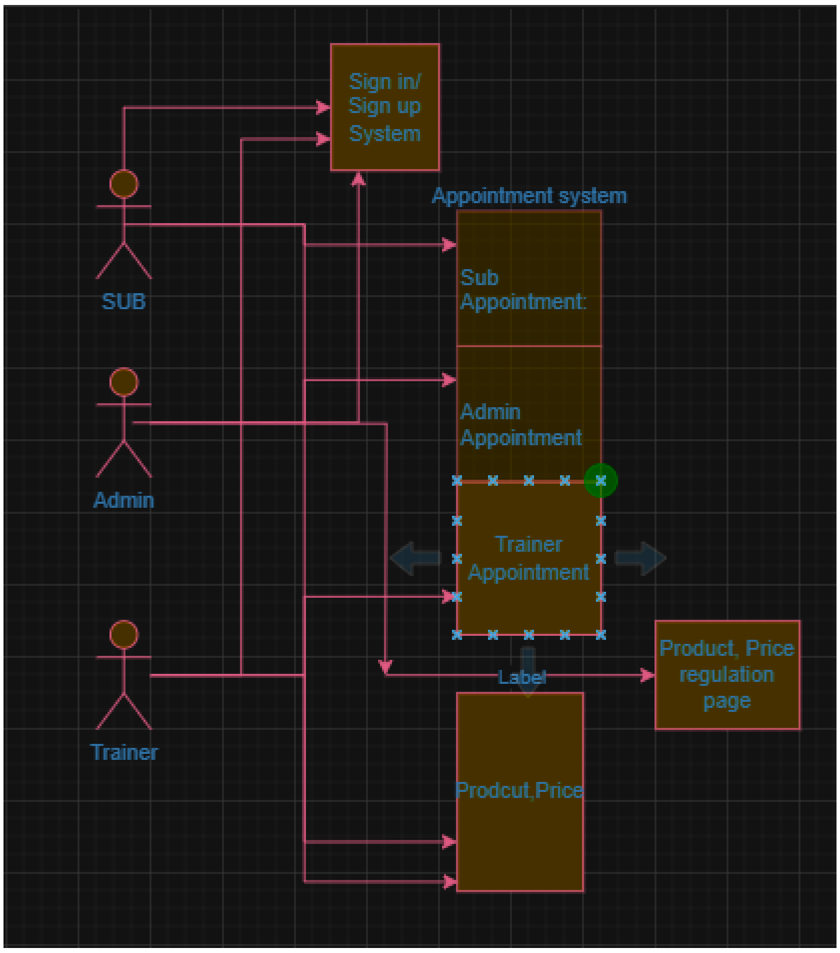
Initial Assumption	Trainer wants to check the Trainer status view page
Normal	Trainer can see the info's given in the page.
WHAT CAN GO WRONG:	There may be mistake in the DB There may be a discconnection between app and DB
SYSTEM STATE ON COMPLETION	Renewing the Trainer Status View page

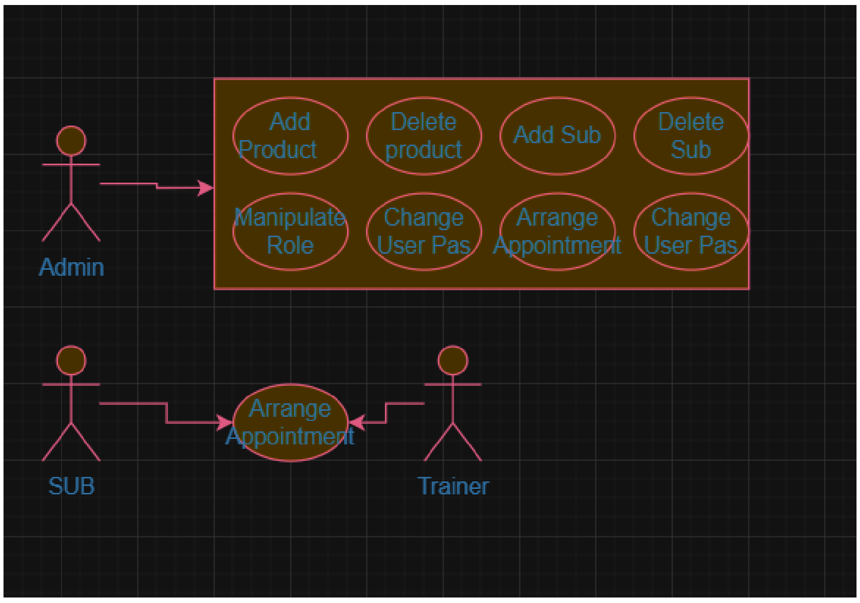
Appoitment system(for SUB/Admin/Trainer)

Initial Assumption	Related User wants to check appointment

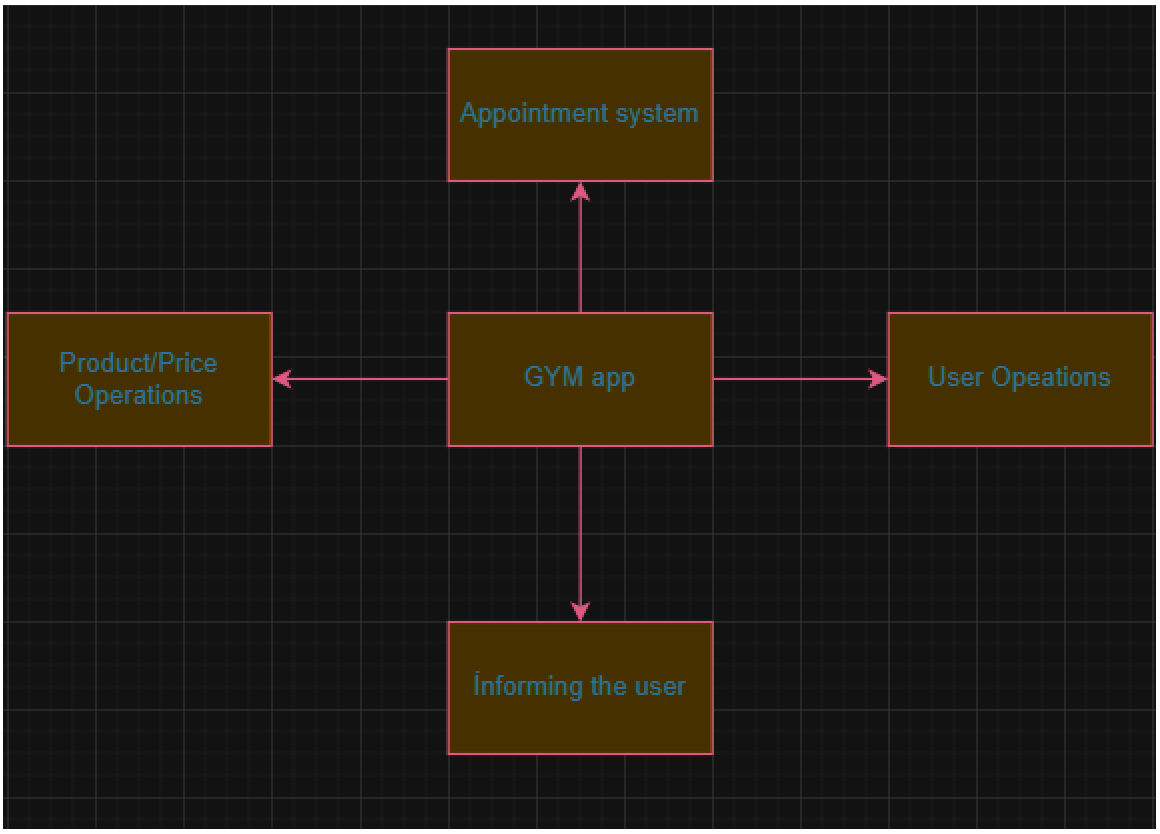
Normal	Related user gets the desired info
WHAT CAN GO WRONG:	There may be mistake in the DB There may be a discconnection between app and DB May Admin cancel an appointment with accidentialty May Trainer cancel an appointment with accidentialty May Sub cancel an appointment with accidentialty
SYSTEM STATE ON COMPLETION	Renewing the Appointment System

11. Draw use cases diagram for all use cases of the project

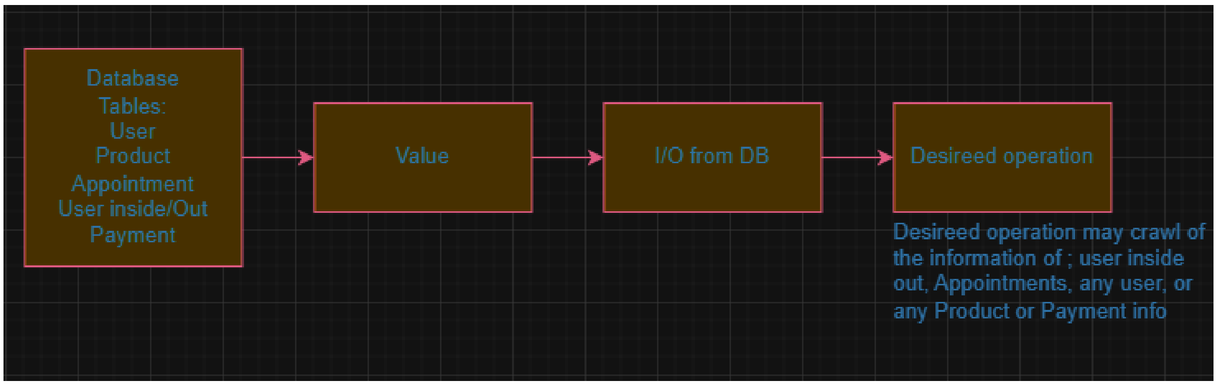




12. Draw full details context UML diagram of the project

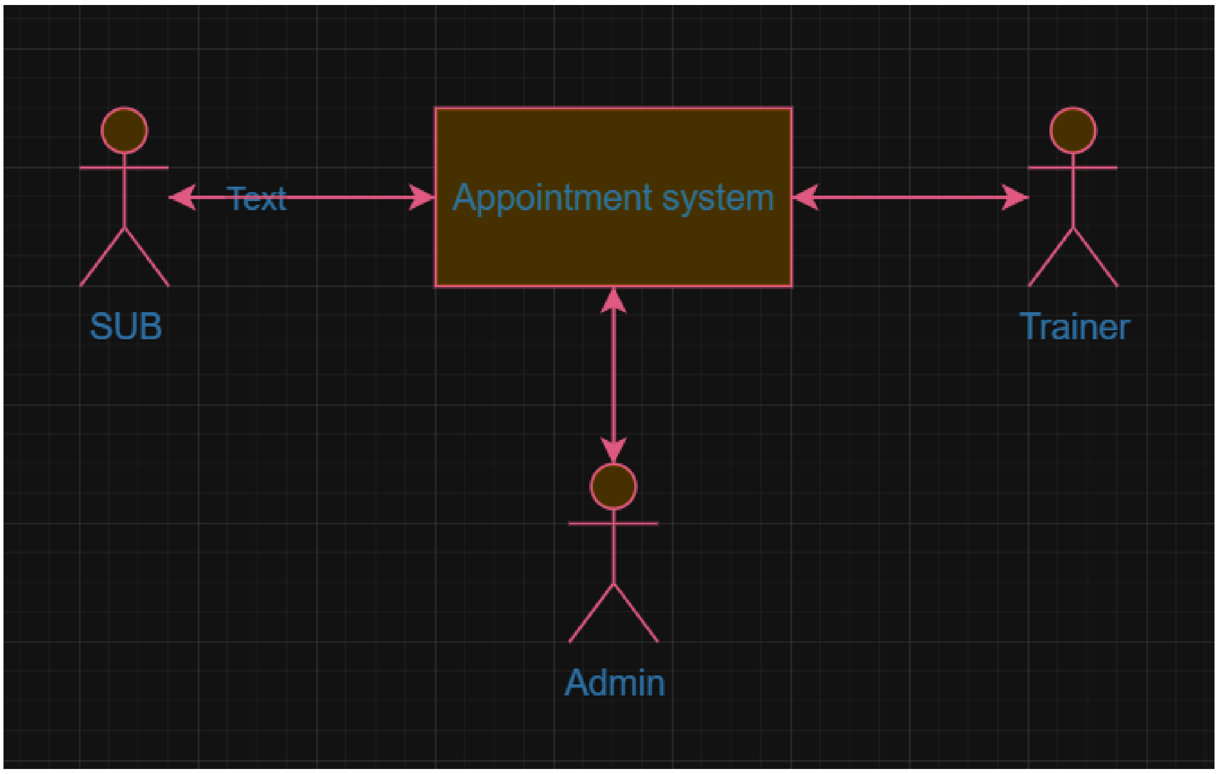


13. Fully Detailed Process Model UML Diagram

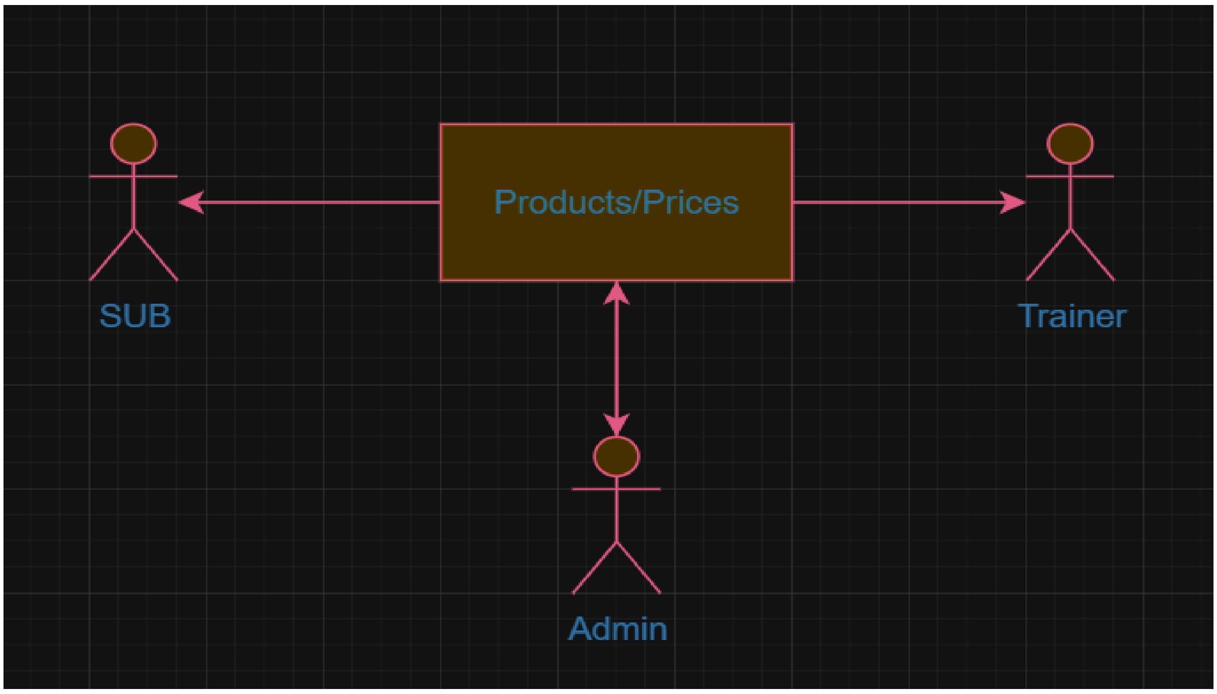

Database Tables:
User
Product
Appointment
User inside/Out
Payment
Value
I/O from DB
Desireed operation
Desireed operation may crawl of the information of ; user inside out, Appointments, any user, or any Product or Payment info

14. Every Use Cases UML Diagram

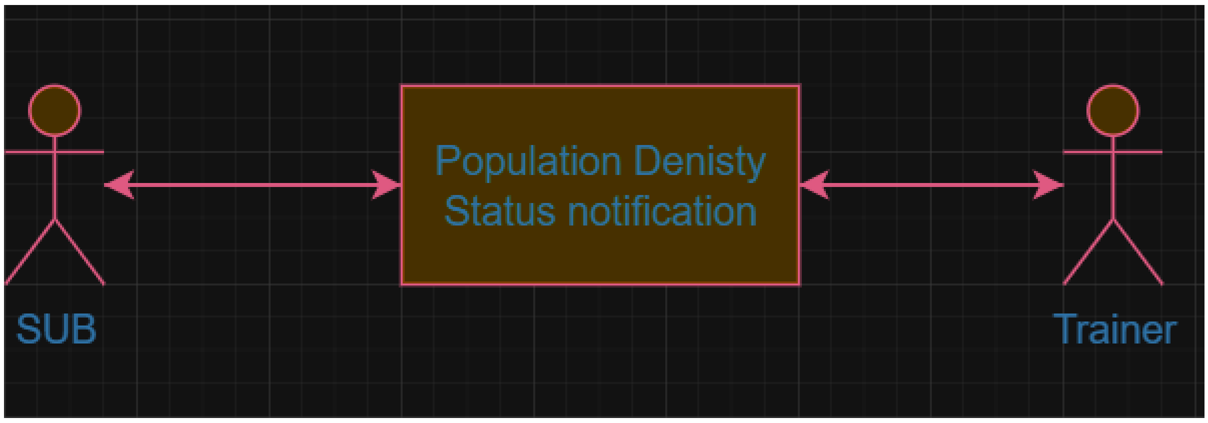
14.1 Appointment system



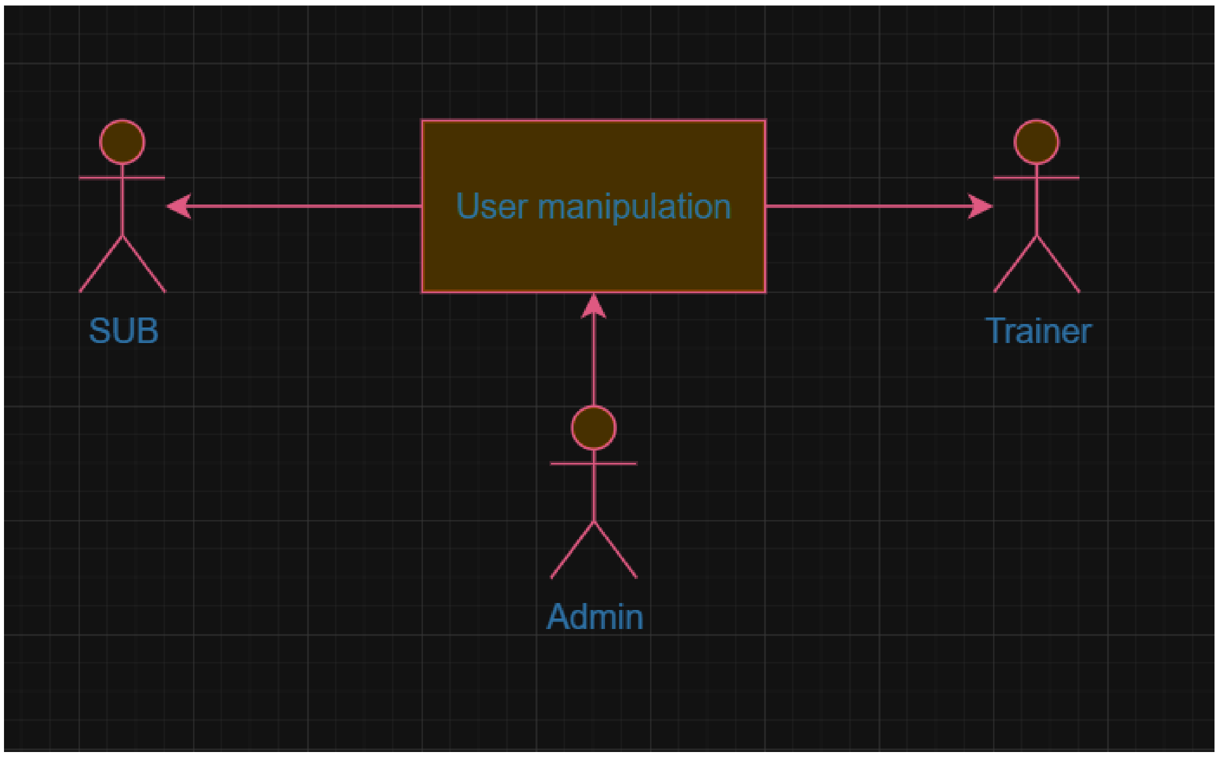
14.2 Product List



14.3 Sub View (aslo trainer view) Status notification



14.4 User manipulation(Roles)



15. Tabular Description of Projects' use cases

15.1 Add-Delete Product and Price regulation.

Sport Saloon App	
Actors	Admin, Trainer,SUB
Description	**Admin adds and manipulates prices and products, SUB's and Trainers are just viewing it(no buying in the app in real life there is)**
Data	Product info
Stimulus	**User can get in to this page and stimulate himself**
Response	Confirmation that Product information has been updated
Comments	The admin should be carefull while updating the products

15.2 Add-Update-Delete sub

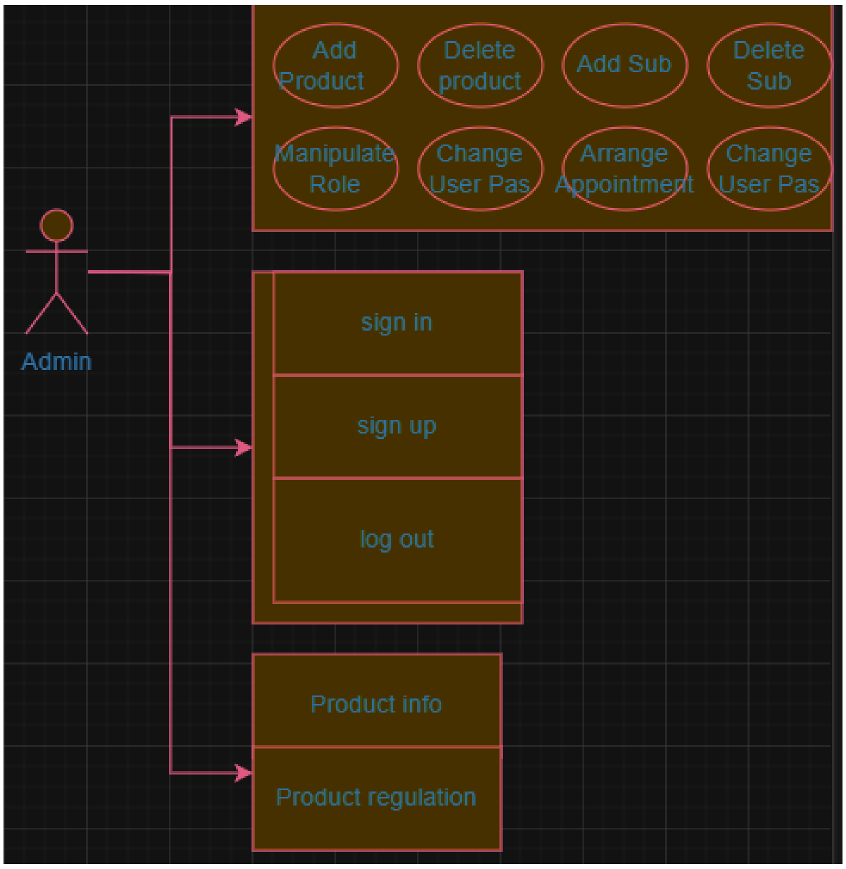
Sport Saloon App	
Actors	Admin, Trainer,SUB
Description	Admin can manipulate roles of users as SUB and Trainer, also may update or delete them. Another case creating a new user
Data	User info
Stimulus	**Trainer,SUB**
Response	Related users info will change.
Comments	Admin should be carefull while updating credentials of users.

15.3 Appoitment system

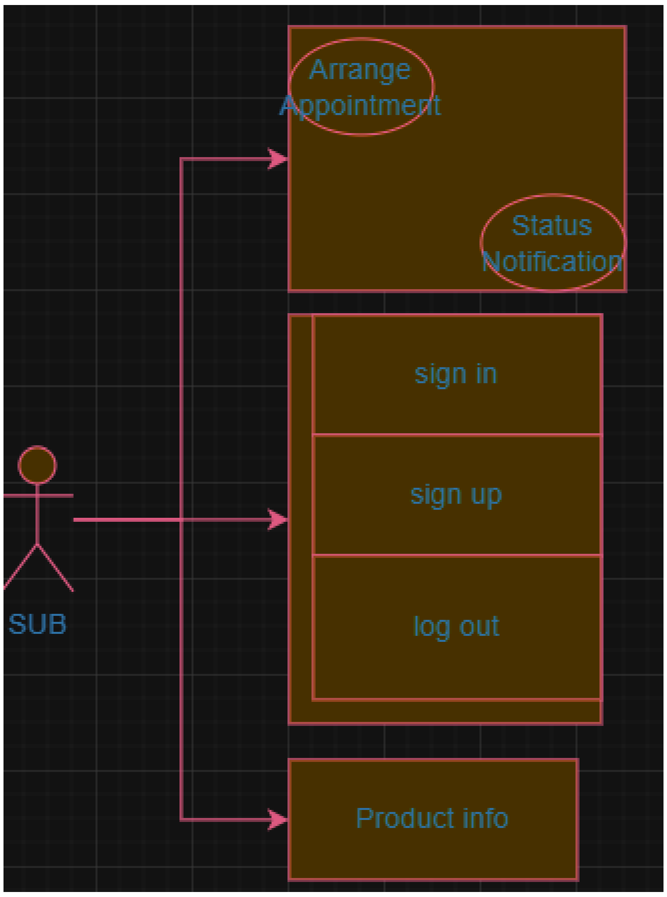
Sport Saloon App	
Actors	Admin, Trainer,SUB
Description	Admin can create or delete appointment Sub can create or delete appointment Trainer can delete appointment, and lock non work days of his
Data	Appointment date and related users appointment date data
Stimulus	**Trainer,SUB**
Response	Related users appointment data will change.
Comments	All users should check the dates.

16 Draw use cases of each agents' use cases UML diagrams of the application

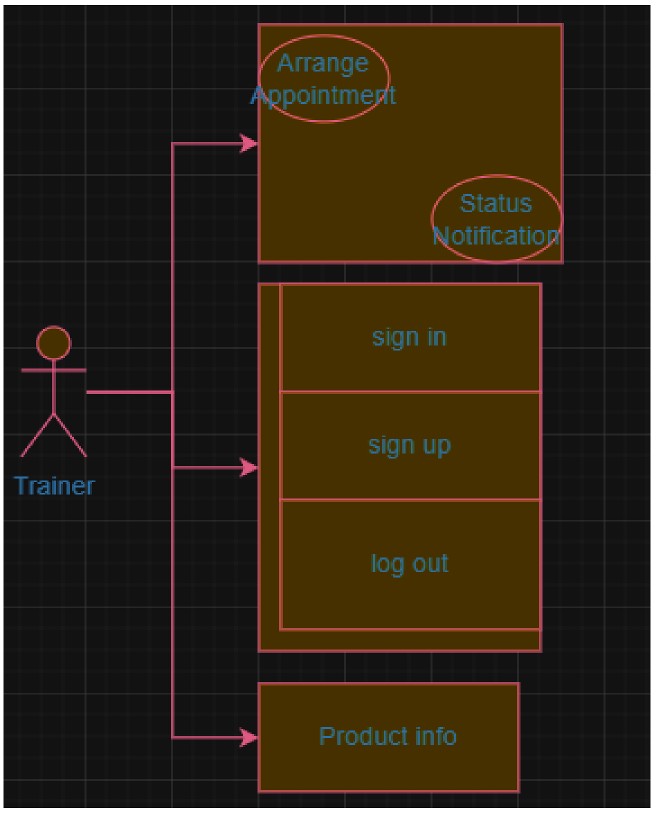
A. Admin



B. Sub

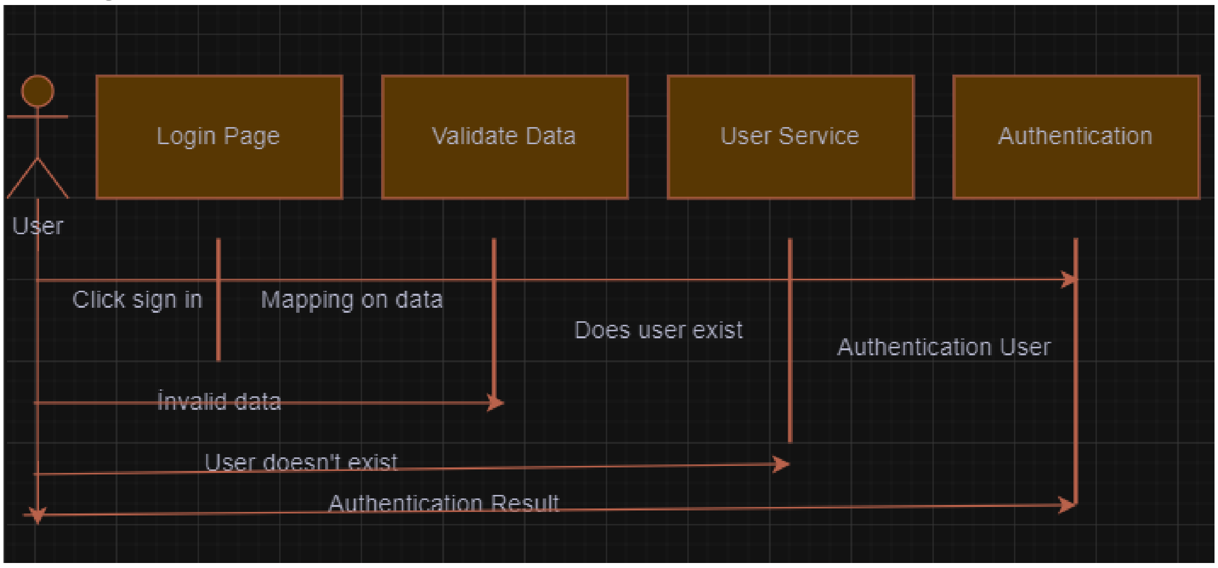


C. Trainer

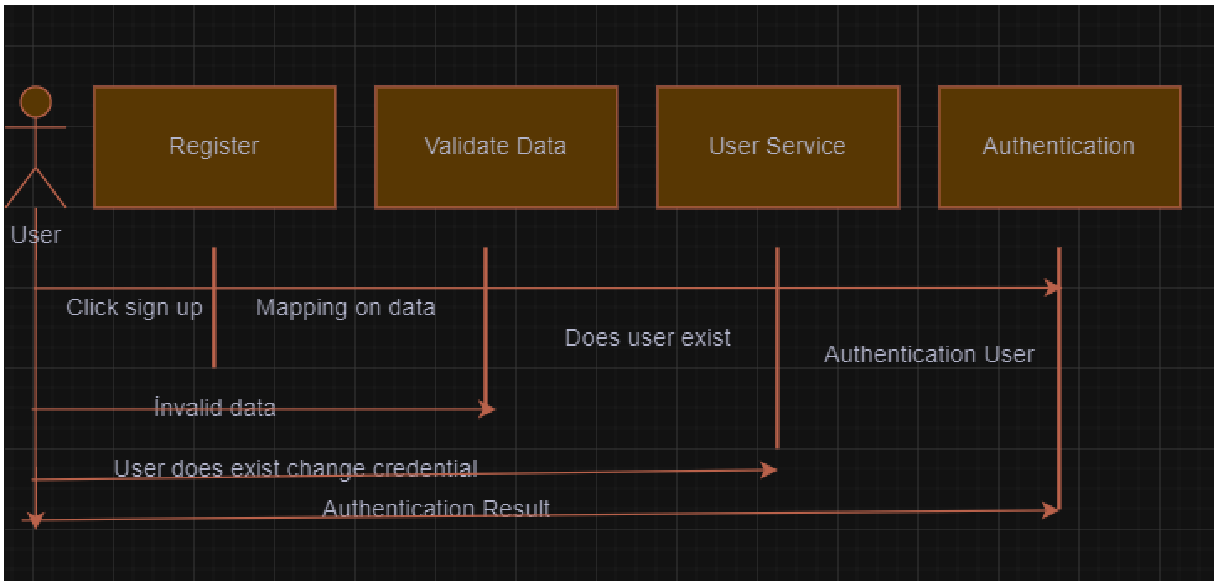


17. Draw Sequence diagrams of every action in the project

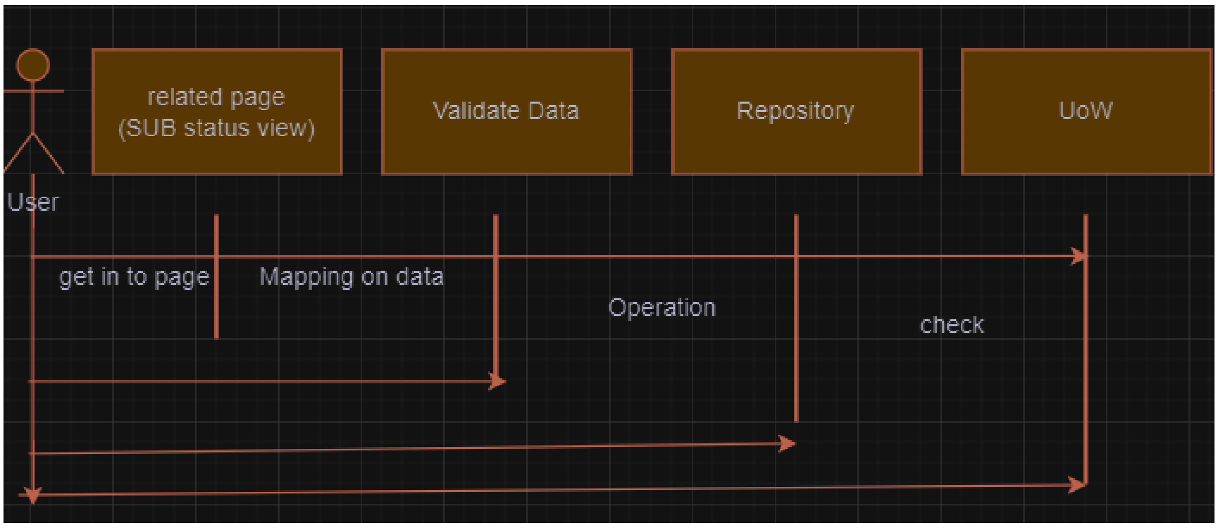
17.1 Login



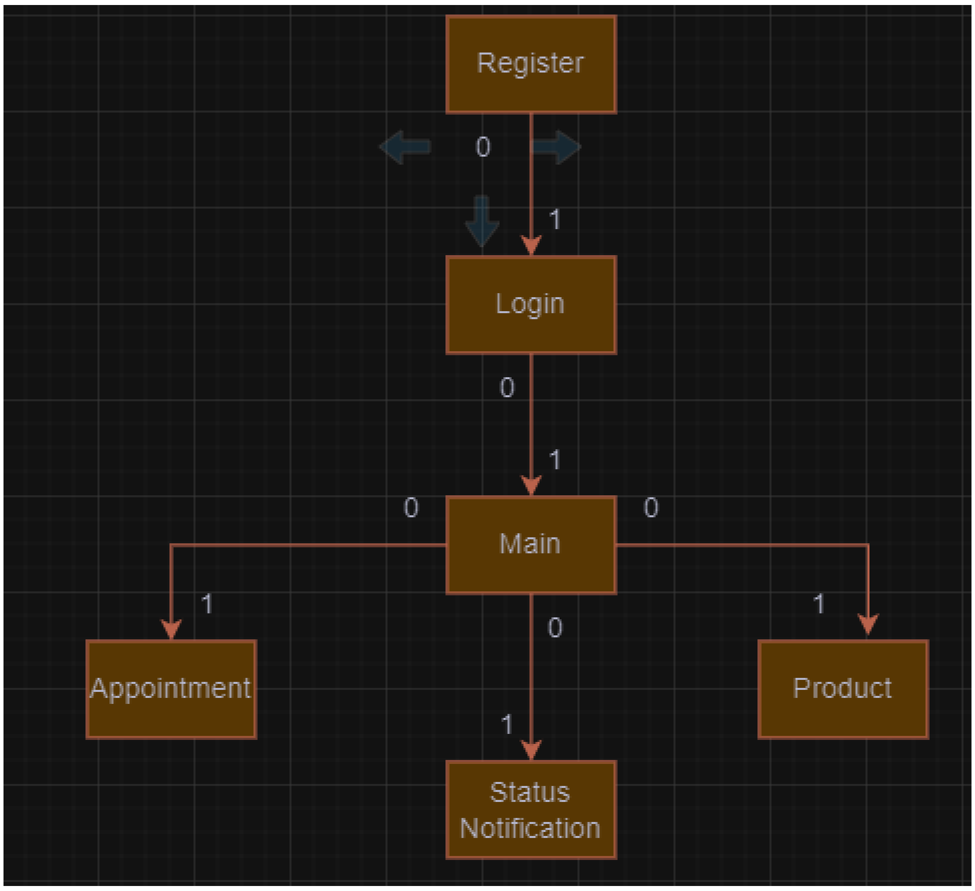
17.2 Register



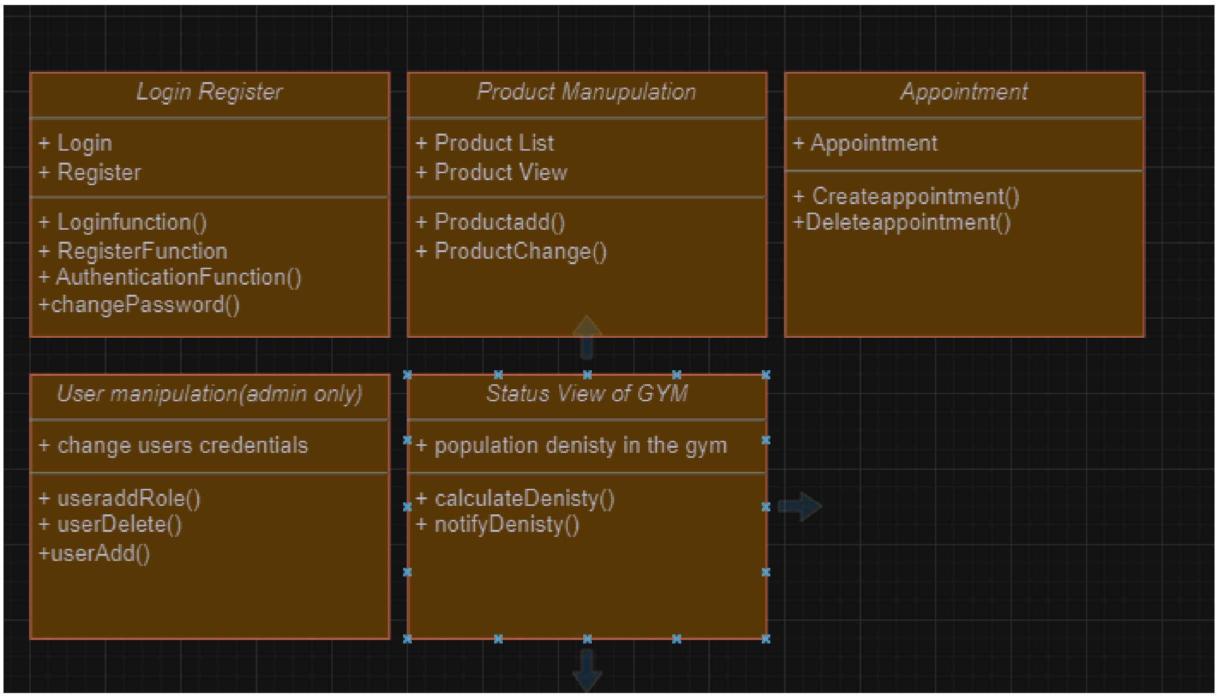
17.3 A general approch other operations in the project



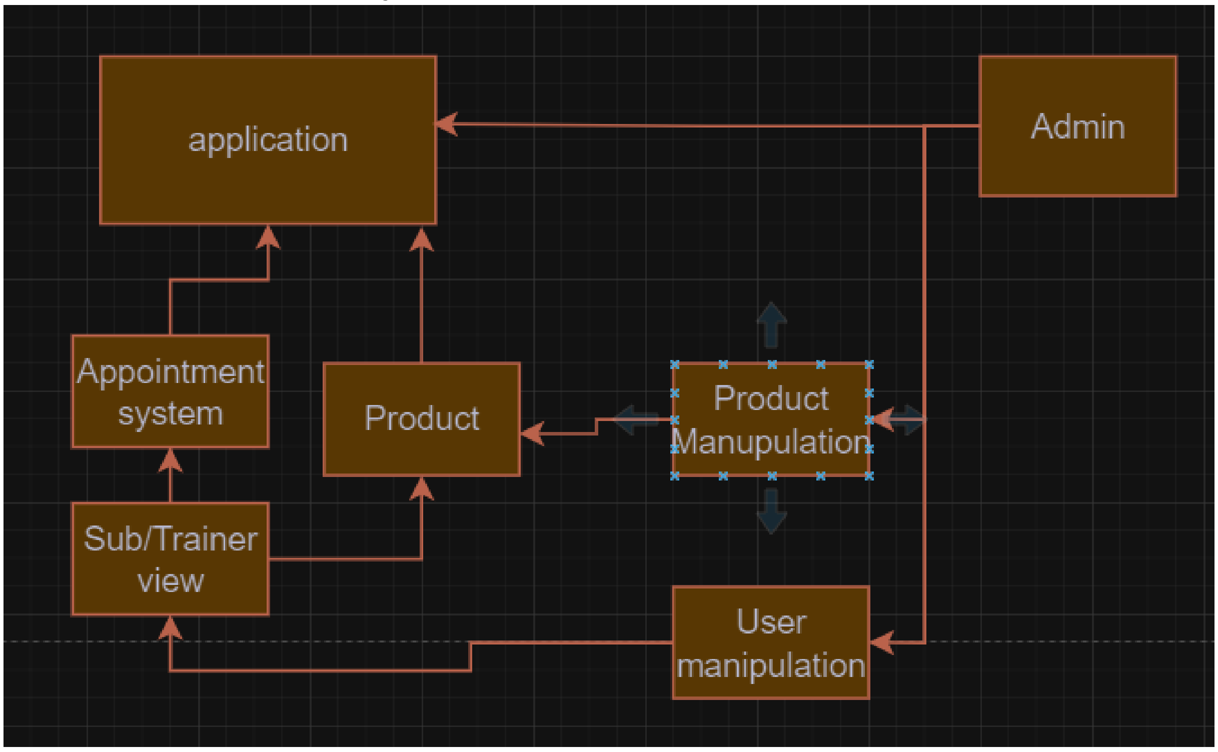
18. Draw UML classes associations of all classes



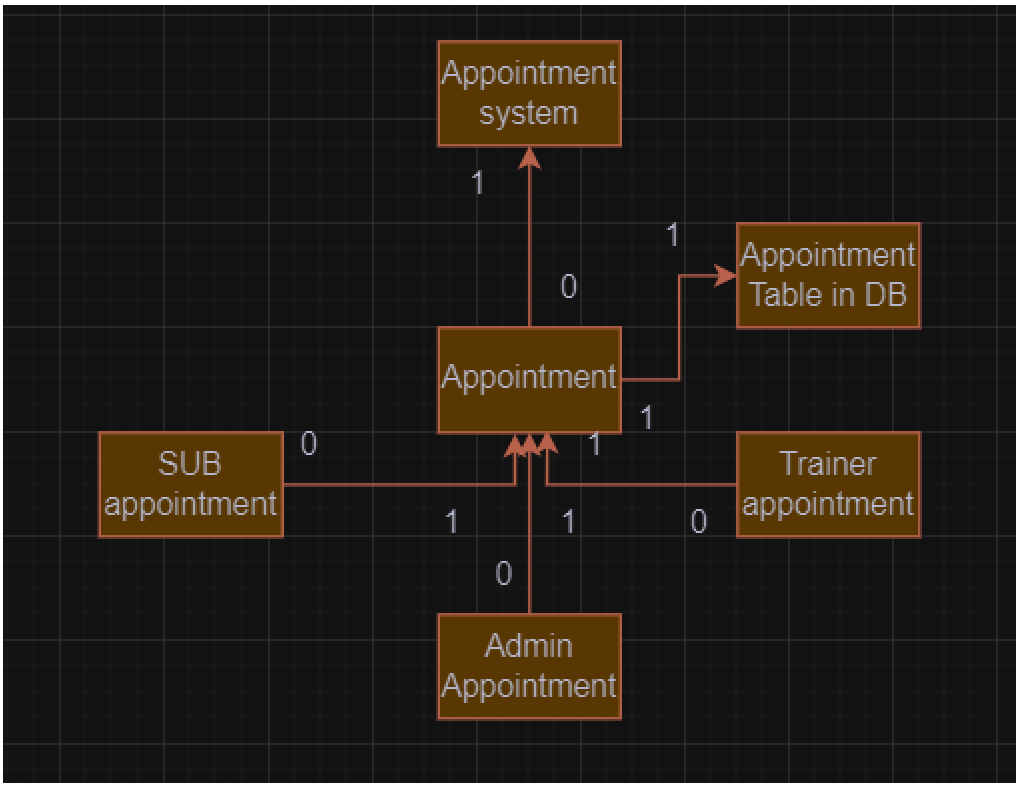
19. Draw Class Models



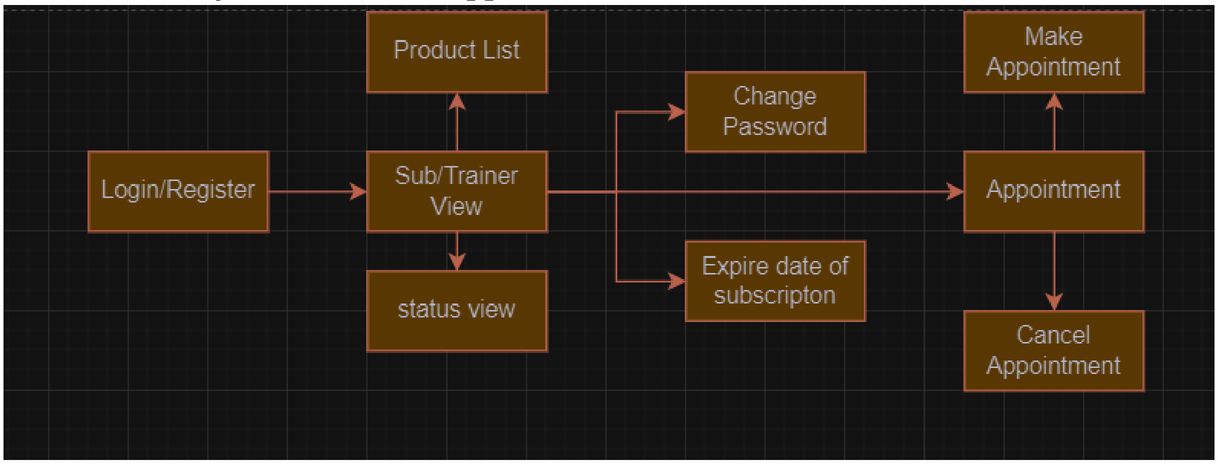
20. Generalization Hierarchy Classes



21. Aggregation Associations of All Classes

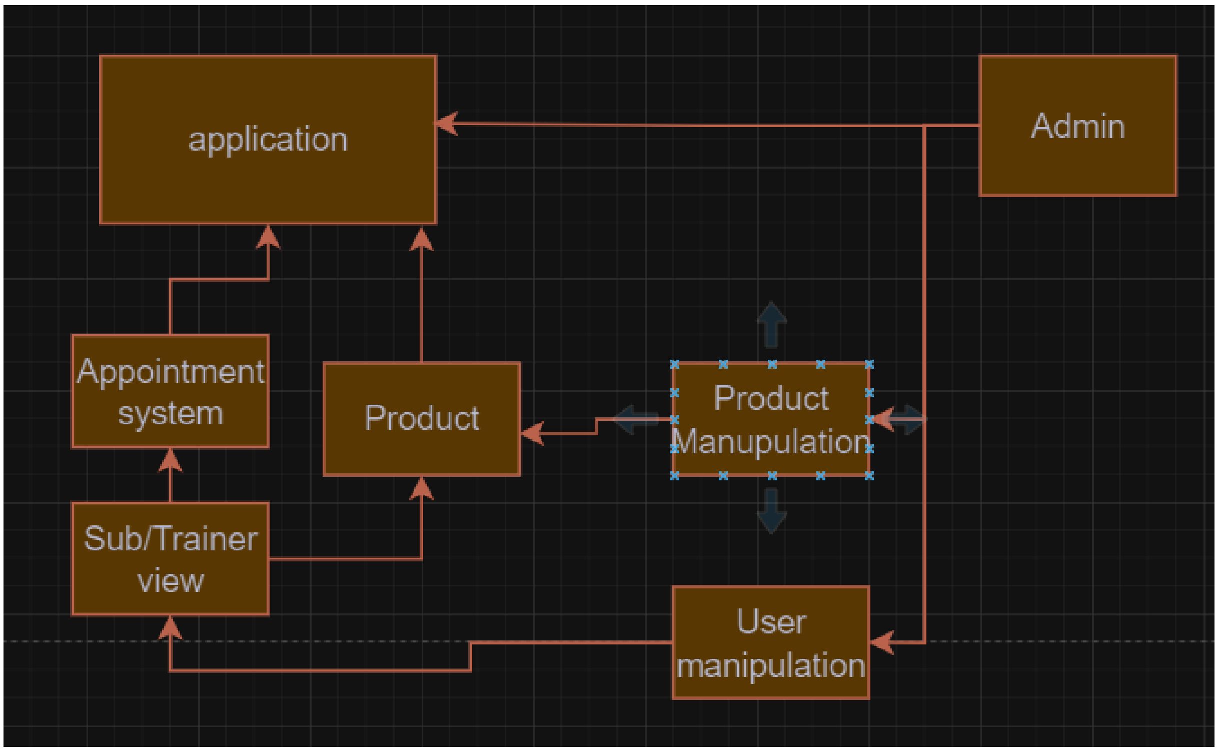


22. Draw Activty of model of the application

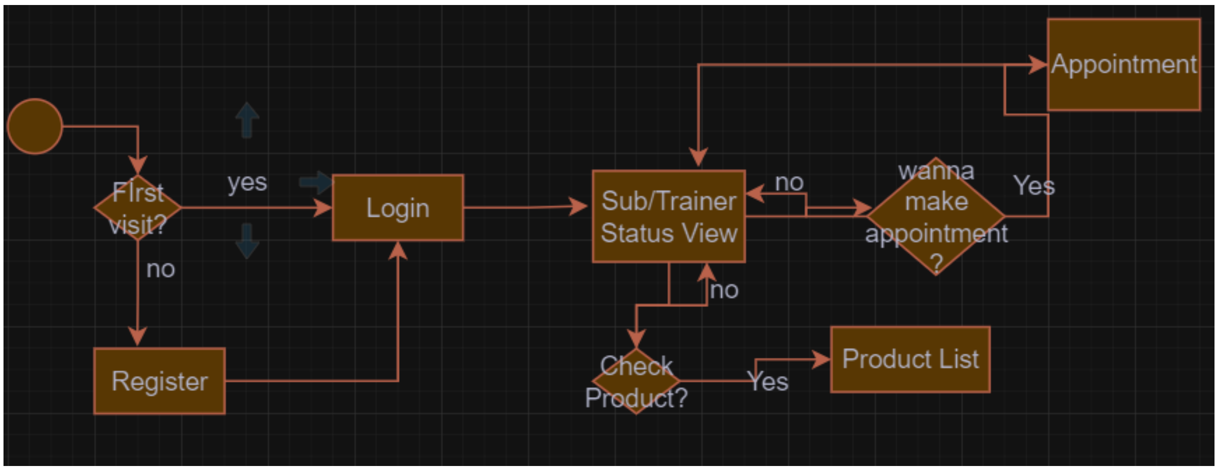


23.Draw the Application process same as 20 "**Generalization Hierarchy Classes**

”



24. State Diagram

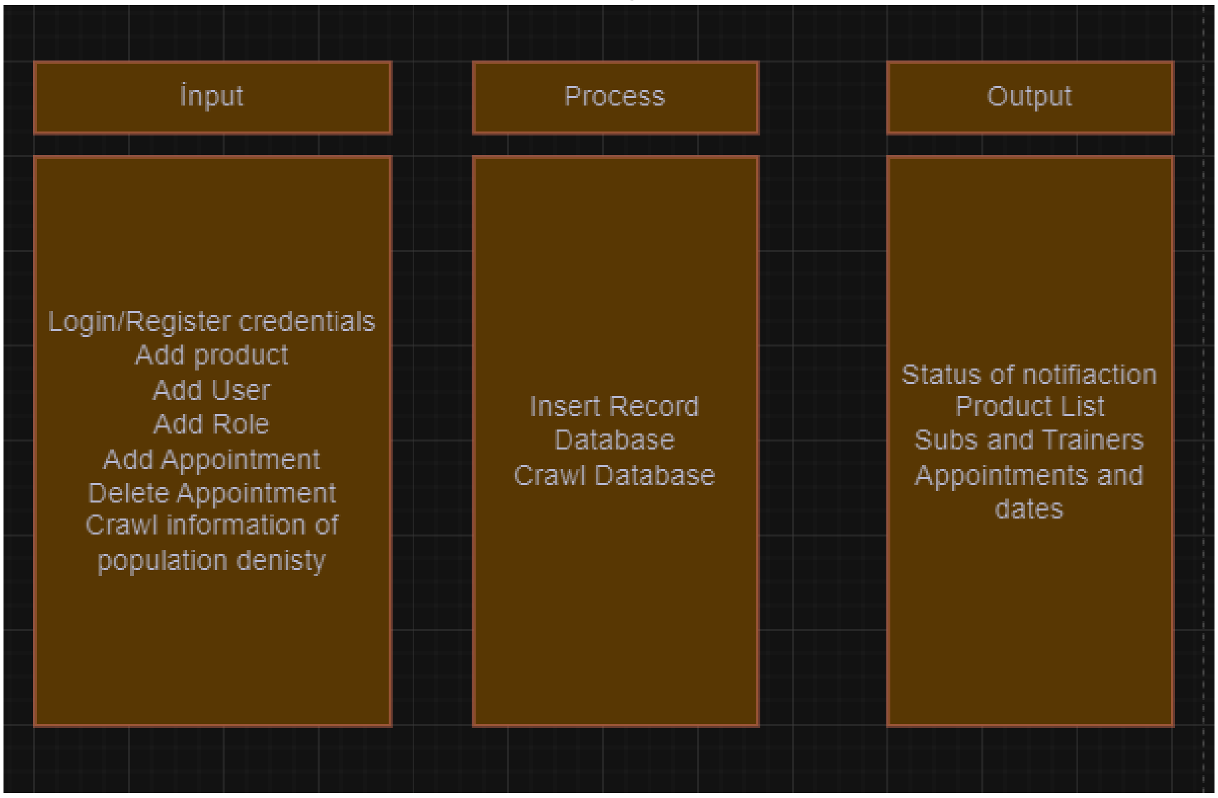


25. Prepare structured forms of the application’s states

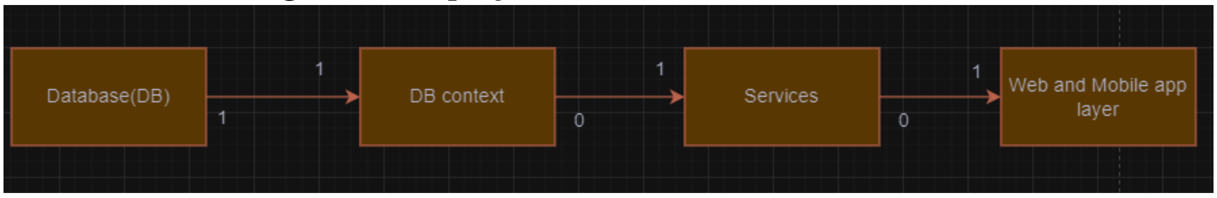
Login	**A page to login**

Register	**A page to register**
satuts view	**A page that SUB or Trainer may check the denisty of the gym**
Appointment	**SUB's may get an appointment for a PT or cancel on of the appointment as well Trainers too.**
Product List	**Listed Products with prices that can be bought in GYM BAR (no selling in application)**

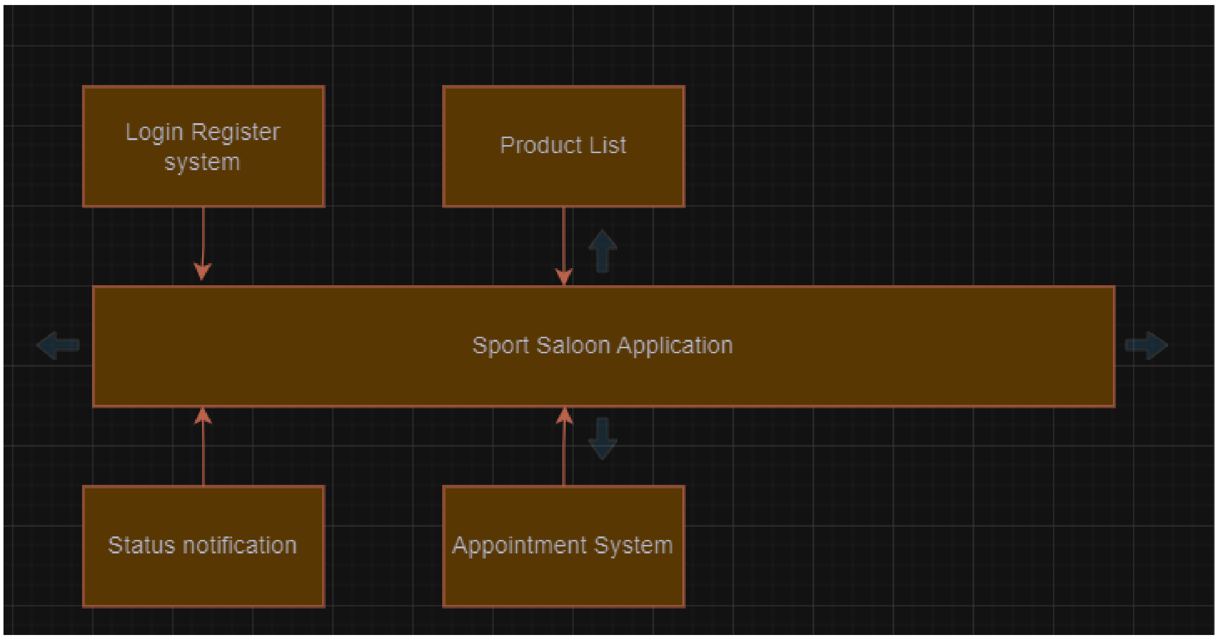
26. Draw the Software Architecture of the Project



27. Draw context diagram of the project



28. Draw high level of architecture of the project



29. Draw all obhects classes of your project
It been driven in 19, 20 and 21

30. Prepare a detailed usage scenario for the project

Yusuf has a Fitness Saloon and it has many subs. He checks how many of subs are using the gym in motion, GYM subs can pay their monthly payment from application. Also in this gym there are many trainer this application provides asystem that can subs and trainer arrange a date for "Personal Trainer" service. GYM has a GYM BAR in this bar provides subs protein based drinks(shakes) and if subs desiree they can buy the powder or ingredients of those drinks to make same thing in home from GYM BAR. Prices of those ingredients are listed on this applications BAR part.

31. Reliability terminology of the software

Human error or mistake	Admin can add wrong price or quantity in the system.
System fault	The hardware might be damaged
System error	An erroneous system state that can lead to system behavior that is unexpected by system users
System failure	An event that occurs at some point in time when the system does not deliver a service as expected by its users

32. Safety terminology of the software

Accident (or mishap)	Leak out user information
Hazard	Member of the system can be harmed
Damage	Unforeseen
Hazard Severity	LOW
Hazard Probability	LOW
Risk	The risk of leak out user information is very high.

33. Security terminology of the software

Asset	User information
Exposure	Possible loss or harm to a computing system. This can be loss or damage to data, or can be a loss of time and effort if recovery is necessary after a security breach
Vulnerability	A weakness in database server that may be exploited to cause harm or loss.
Attack	An exploitation of a system's vulnerability
Threats	Circumstances that have potential to cause loss or harm.
Control	Properly update and patch the system.

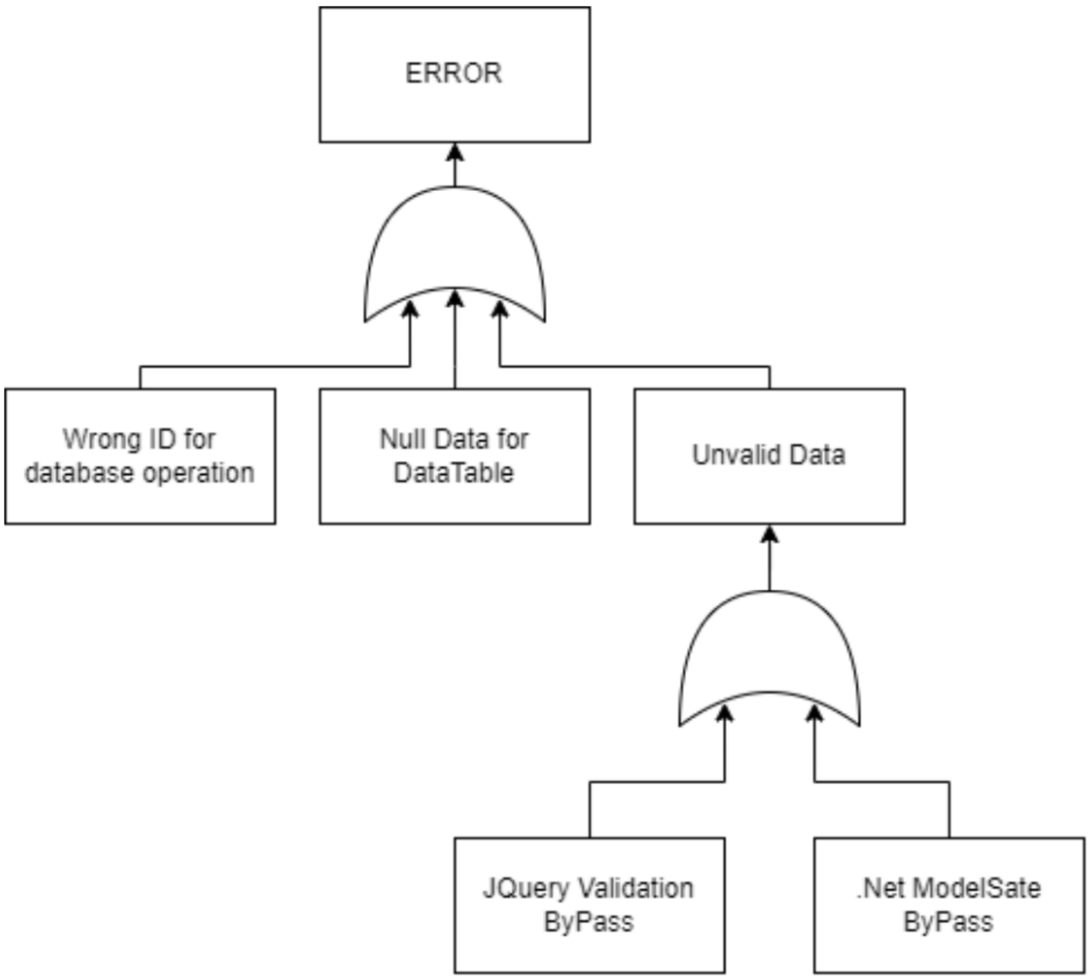
34. Develop some vulnerability avoidance techniques

System is a basic system there's not much layers it is not that secure but stil we will hash our database. It is almost imposibble to obtain data with brute force.

35. Prepare a risk classification tabular view of the system

Identified hazard	Hazard probability	Accident severity	Estimated risk	Acceptability
Wrong User Credentials	VERY LOW	VERY LOW	VERY HİGH	INTOLERABLE
Wrong Product Information	VERY LOW	VERY LOW	VERY LOW	INTOLERABLE
Power failure	High	LOW	LOW	ACCEPTABLE

36. Prepare an example of a software fault tree for the system



37. Prepare an examples of safety requirements for the system

1) The system will prevent empty, or null values.

2) The system will encode and decode html characters to prevent XSS.

3) The system creates own SQL queries to prevent possible SQL injection.

4) The system manages cookie and session to prevent possible CSRF

5) The system calculates and protects necessary value in database to prevent IDOR.

38. Prepare an examples of functional reliability requirements for the system

1) The system has multiple data validation in different layers. (Checking)

2) The system store database backup in different server.

3) The system must be implemented in LTS version of .NET Core

39. Prepare a threat and control analysis in a preliminary risk assessment for the system

Threat	Probability	Control	Feasibility
Threat of unauthorized individuals gaining access to sensitive information stored in the ecommerce system, such as customer information and financial data	High	Using strong authentication and authorization controls to prevent unauthorized access.	These controls are relatively easy to implement and maintain.
Hackers can flood the system with traffic to make it unavailable to legitimate users	High	Use a Web Application Firewall (WAF) to detect and block DDoS attacks.	These controls are relatively easy to implement and maintain.

Thank you for listening