BBM384 -Software Engineering Laboratory

Sports Center Membership System
Architecture Notebook

Group 4:
Aycan ÖZMEN – 21427248
Muazzez Şule KARAŞLAR-21427066
İlayda KAYA- 21580836
Mustafa DANYILDIZ – 21426845
Uğurcan ÇİFTÇİ -21526863

Sports Center Membership System
Architecture Notebook

1. Purpose

The main purpose of this document is the general design of the system architectural structures to understand component interactions. These are supported by diagrams. This document describes the philosophy, decisions, constraints, justifications, significant elements, and any other overarching aspects of the system that shape the design and implementation. It explains the design of the system, its architectural mechanisms, its architecturally significant requirements and its constraints.

2. Architectural goals and philosophy

- The goal of this program is to provide a well-designed architecture that can be used easily by the employees who are in sports center and the customers.

- The system is designed as the web application.

- The application will be simple and easy to use.

- System's functionalities will be structured using C# and MySQL.

- Proper user authentication will be provided. Users (the customers and the employees who are in sports center) will be logged with username and password to perform operations.

- The system will respond to the member in less than two seconds from the time of the requested submittal.

- All interface going to be represented in English. But it will be easily modifiable to extend language support.

- Data persistence will be addressed using a relational database.

3. Assumptions and dependencies

- Users are able to buy the courses, follow up their body development in a fast and reliable way.

- The application should be working 24 hours a day, 7 days a week.

- The storage capacity of the system should be high.

- All information should be stored in the database. Passwords must be stored encrypted.

- The system should support quick transaction.

- The system will not be associated with any hardware.

- This system requires the internet.

4. Architecturally significant requirements

- In this project, the system is a web application.

- Users must log in to the system with their username and password in order to be able to carry out operations related to the sports center.

- For security reasons, the password of anyone registered on the system must be encrypted.

- The system should record every change the user makes. These "log" records will be recorded in full.

- All operations performed by other users registered to the system (system manager, branch manager, trainer) are added to the employee records for audit purposes.

- The system must be able to respond to any request within a maximum of 5 seconds.

- The system will run seven days a week, twenty-four hours per day.

- If there is any problem with the system, it will be intervened as soon as possible. The system records all problems that are experienced.

## 5. Decisions, constraints, and justifications

- The system is a web application, so internet connection is required for it to work
- Since the system is web-based, there is no specific operating system limitation.
- The system is trying to be designed in a flexible way, considering that there may be various requirements later.
- The system will be developed using the C# language and the MySQL database will be used.
- Users who are not registered to the system are limited in the operations they can do???
- Manipulating customer and trainer can only be done by branch manager.
- Manipulating employee can only be done by system manager.

6. Architectural Mechanisms

We have benefited from the architectural mechanism to reduce the complexity of our system. Our main mechanism we will use is 3 Layer Architecture mechanism. It is aimed to manage the system faster and easier by using this architecture. The actions performed on each layer of this architecture are designed separately. Thus, if there are situations that need to be intervened, the process is carried out easily and without waste of time.

There are three distinct layers in this mechanism: Data Layer, Business Process Layer, Presentation Layer.

1- **Data Layer:** At this layer, the data is extracted from the database and added to the database.

2- **Business Process Layer:** In this layer, the data extracted from the database is processed in the system. With the business process layer, the generated programmed data is now ready.

3- **Presentation Layer:** In this layer, the user interface of the application interacts with the user. In the previous 2 layers, the view of the data which is prepared to the user is determined and the system has become ready.
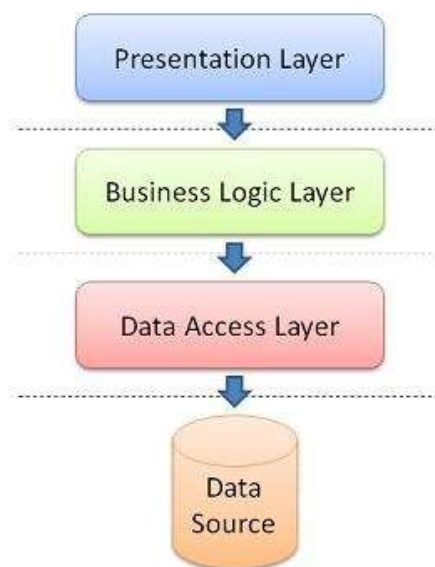
(This image was taken from weblogs.asp.net)
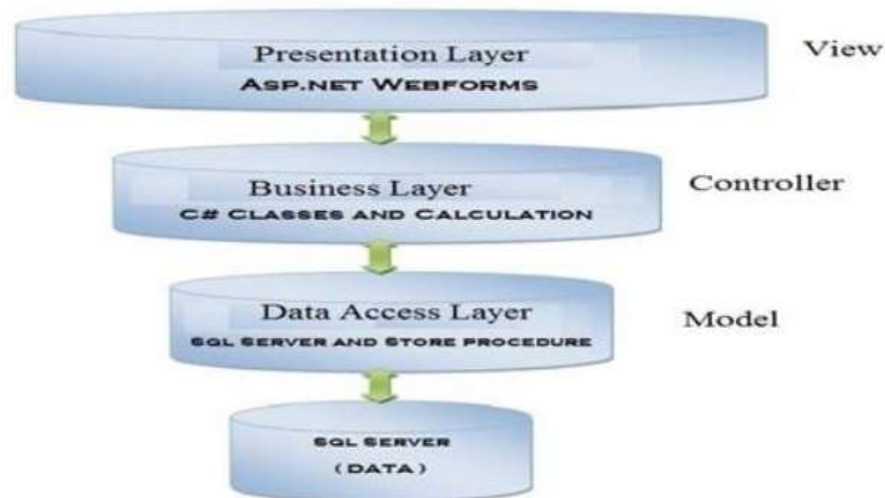
Advantages of 3-layer Architecture
- Reusability: It is possible to modify only the presentation layer without affecting the business and data layers.
- Maintainability: Other layers are not affected when we change any layer.
- Since the job layer is between the data layer and the presentation layer, the database layer is more secure, not directly accessible.
- Easy to apply object oriented concept.

7. Key abstractions

Member: The abstraction of members of the sports center. They can perform the procedures related to the sports hall in the system.

Trainer: The abstraction for arranging the measures of the members and updating the attendance list.

System Manager: System manager can define branch, course and payment. System manager can also manipulate employee. That is, the abstraction for organizing and managing the system in general.

**Branch Manager:** Branch manager can manipulate customer and trainer, open/close courses and activate member.

Interfaces: Interaction point between user and system.

Database: The abstraction for storing gym information and user information.

8. Layers or architectural framework

The main architectural framework in our project is the "4 + 1 view model". The reason we choose this model is that every stakeholder is interested in something different in the system. For this reason, it is logical to decompose architecture to different views.
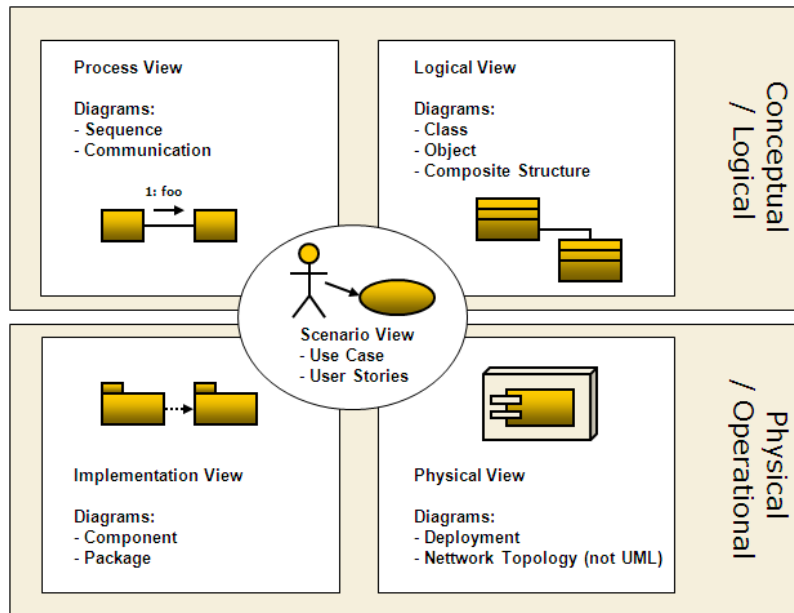
This approach has a total of 5 views and is expressed as 4 + 1. It makes modeling easier and makes it easier to organize. The 4 + 1 approach makes it possible for stakeholders to take relevant parts of the model.

9. Architectural views

In 4+1 Architectural View Model, we have got logical, implementation, process and physical views.

1- **Logical View:** The purpose of the logical view is to specify the functional requirements of the system. It's relevance is to developers. This contains information about the various parts of the system. In UML, logical view is modeled using class, object and composite structure diagrams.

2- **Implementation View:** The organization of software modules such as source code, data files, executable files in the project development environment. It focuses more on the subsystems. In UML, Package and Component diagrams are used to model the implementation view.
   *Package Diagram is as below:*
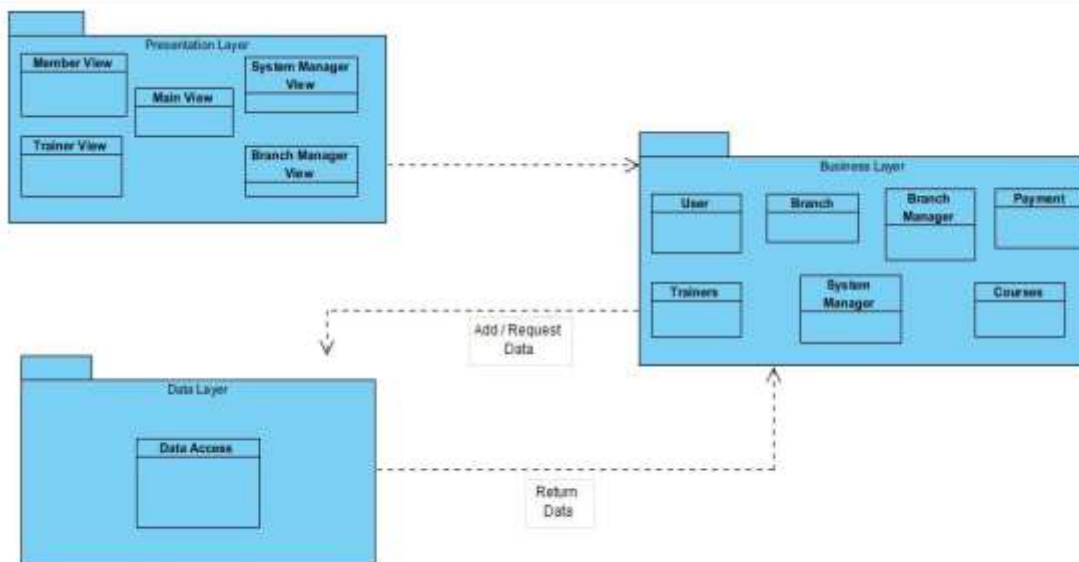
Component Diagram is as below:
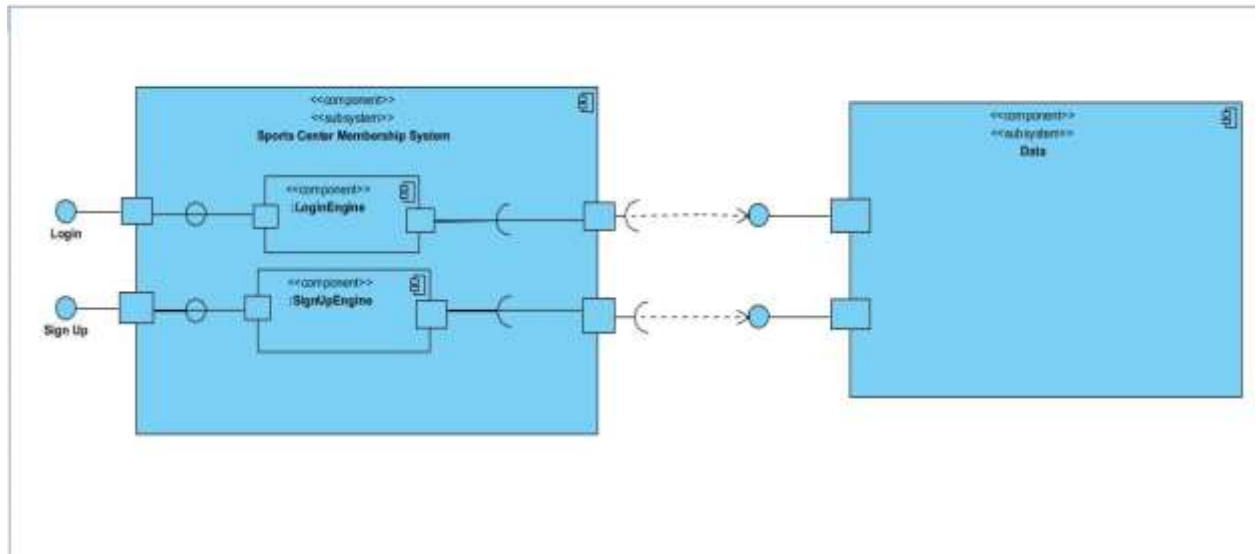
3- **Process View:** This explains the concurrency and synchronization mechanisms of the system. It encompasses some requirements such as performance, fault tolerance, availability and data distribution. In UML, Sequence and Communication diagrams are used to model the development view.

Sequence Diagram for Login operation is as below:
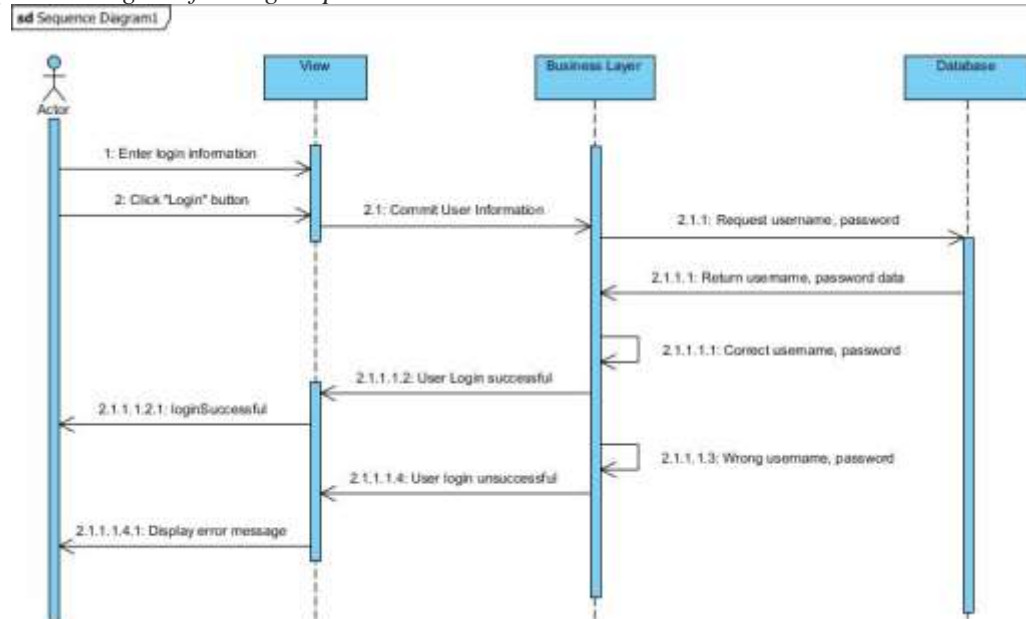
4- **Physical view:** Deployment diagram shows how the system will look in the physical layer with software components and their hardware parts. It shows relations between them.
Deployment Diagram is as below:
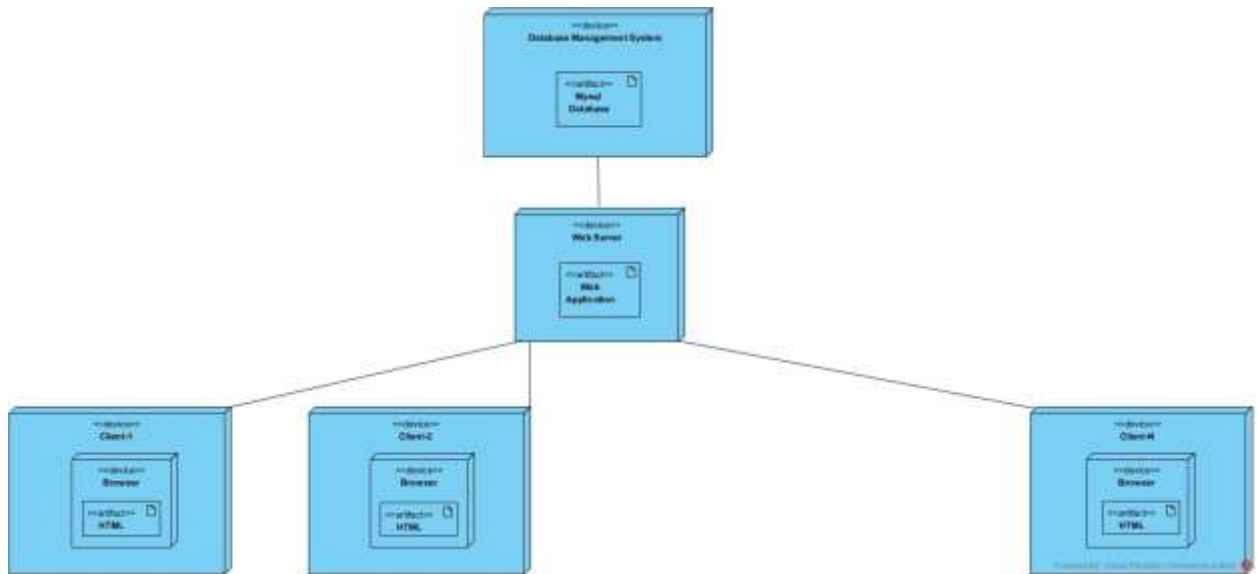
5- **Scenario View:** This view describes the functionality of the system from the perspective from outside world. It captures system functionality as seen by users and built in early stages of development. In UML, Use Case diagrams are used to model the physical view.
Use Case Diagram is as below: