Hacettepe University
Department of Computer Engineering

BBM384
Architectural Notebook

Veli Can AYDIN	21526662 - Software Project Manager
Onur VARSAK	21387485 - Software Analyst
Abdülsamet KALKAN	21427033 - Software Architect
Sefa Seyda ÖZDOĞAN	21483421 - Software Configuration Manager
Hakan İMAL	21426993 - Software Tester
Barkın ATASAY	21590911 - Software Quality Manager

1. Purpose

This document describes the philosophy, decisions, constraints, justifications, significant elements, and any other overarching aspects of the system that shape the design and implementation.

When designing this system, we will explain how it is shaped, what stages it is going through, what steps it will follow, and how we will use restrictions.
This document will serve as a guide between the software and the design at the stage of our project.

To summarize, this document describes how the system works, what systems it needs. This document will play a great role in system design for us and the people who design the system. Each item will be explained in more detail below.

2. Architectural goals and philosophy

The main task of the architect is to design the project on time and prepare a system that will meet all the requirements. The way we are going to follow is to choose robust technologies and reliable technologies.

The main purpose of our site is a fast, reliable, easy-to-use system.

This system will be a Web-based application that will serve users through the web. The need for a working environment will be the internet. We use the database to be more interactive when designing our application, and this database will always be updated. Our system is responsible and mobile phones and mobile phones over the web is designed according to.

Since our system is currently in design phase, a new component is designed to be added because new improvements can be added to our system, which will help to complete the system in a timely and hassle-free manner.

3. Assumptions and dependencies

Our system is designed to be open to new features and is designed to add a new feature to our system immediately.

We've thought about creating a more interactive environment for users, and in this system, users will be able to communicate with their educators more quickly and get support.

The purpose of using a database is to provide a more interactive environment for the users and managers of the system.

We designed our application on a Web-based basis because we saved the user from the trouble of downloading and installing the file to use the system.

Our system works seamlessly thanks to only one Internet browser on all kinds of hardware.

The new features are designed to be easily integrated into our system.

4. Architecturally significant requirements

- Our system runs smoothly on all embedded systems that can run Internet browsers.
- The system needs an Internet browser.
- Users and administrators who wish to log on to the system must enter their user names and passwords.
- The system is designed to run 7/24
- Every step on my site will be backed up in a short time and a precaution will be taken against any system crash.
- The system should provide a more convenient environment between users and educators.

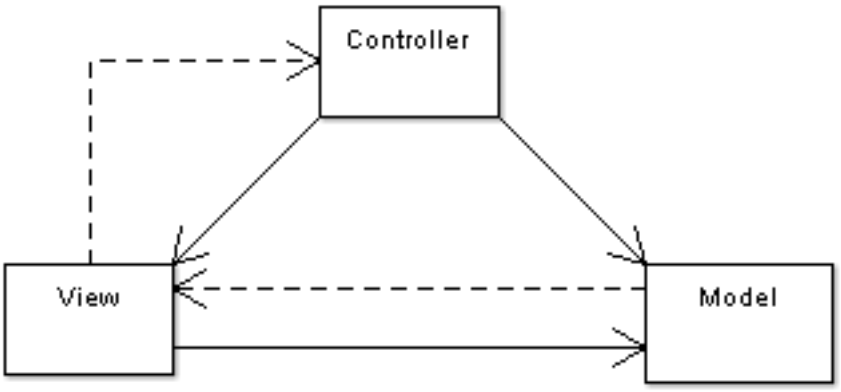
5. Decisions, constraints, and justifications

- Users and administrators in the system will not be exposed to all information for security reasons.
- There will be no restrictions on the number of users.
- Adding users to the course will give the approval to the Administrators and trainers upon request from the user.
- To register to the system, you only need to be a member of the gym.
- Users who are not registered with the system will only be able to view the features of the sports hall and will not be able to contact other people.
- Other processes will be after logon.
- Users will be informed about the system through documentation.
- No access to the system without a connection to the internet.
- The application will be simple and simple to use.
- The system will run simultaneously depending on the speed of the internet rather than being connected to the hardware.

6. Architectural Mechanisms

Architectural mechanisms are needed to reduce the complexity of the system's development process and to create a good product.

In addition, architectural mechanism is used to standardize solutions. The development process can sometimes be complex. We used the MVC structure here.



Architectural Mechanism 1

Model: MySQL Server with hibernate

1) Purpose: To store application's data objects (courses, users, trainers).
2) Attributes: Attributes of the courses (name, trainer name, courseId, UserList) and users (username, name, surname and password), trainer(username, name, surname and password, courseList)
3) Function: When a new create or user is created, it is stored in an instance of the model.

Architectural Mechanism 2

View: User Interfaces

1) Purpose: To interact users with trainer in course.
2) Attributes: Users can be visitor, member or trainer.
3) Function: Informs users about transactions.

Architectural Mechanism 3

Controller: User Controller, Trainer Controller and Course Controller

1) Purpose: To update the view when transaction occurs.
2) Attributes: Users can be visitor, member or trainer.
3) Function: Sends commands to model (course and users) to update the models' state.

7. Key abstractions

Trainers: This is sub type of Person. Trainers can add, delete, and update course. They also can add a customer to the system and remove him/her from system.

Customers: This is sub type of Person. Customers can ask for getting a course from system or return it. They have password for authentication.

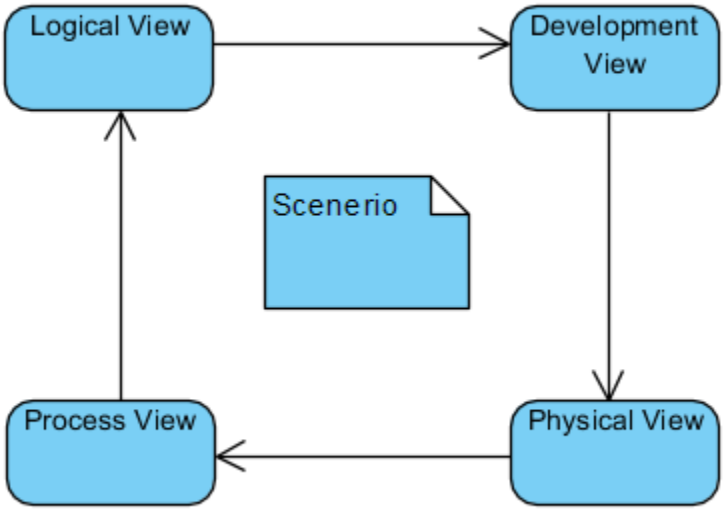
Persons: Describes the Customers and Trainers information that is held by the system.

Manager: This abstraction describes the Controller Class in the MVC pattern. It includes all functions that run the application.

Courses: Is the abstraction of course class that holds information about the course

8. Layers or architectural framework

Provides a common environment for building, interpreting, analyzing, and using architectural frameworks. architecture description 4+1 view model is an example of this. These multiple views define different stakeholders in the system, such as end users, developers, and project management.

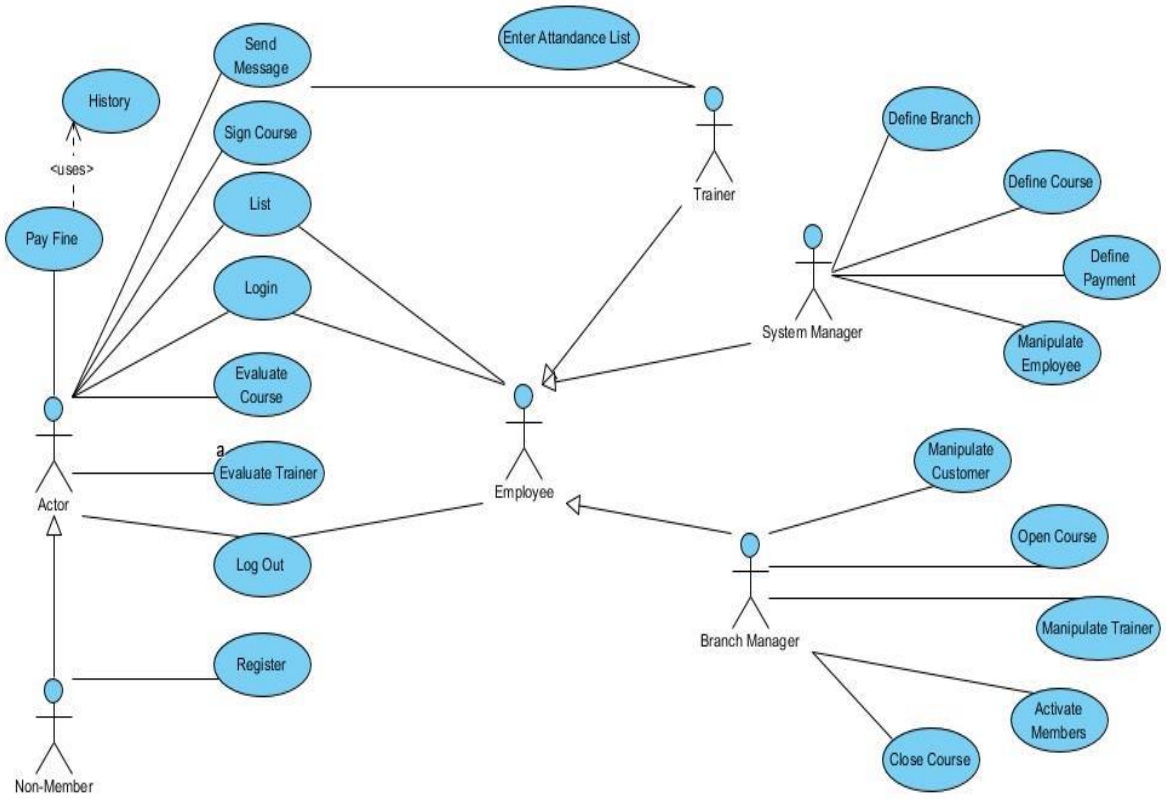


9. Architectural views

Recommended views

- **Logical:** Describes the structure and behavior of architecturally significant portions of the system. This might include the package structure, critical interfaces, important classes and subsystems, and the relationships between these elements. It also includes physical and logical views of persistent data, if persistence will be built into the system. This is a documented subset of the design.
- **Operational:** Describes the physical nodes of the system and the processes, threads, and components that run on those physical nodes. This view isn't necessary if the system runs in a single process and thread.

- **Use case:**


Send Message
Enter Attandance List
History
Sign Course
Trainer
Define Branch
<uses>
Define Course
List
Pay Fine
Define Payment
Login
System Manager
Manipulate Employee
Evaluate Course
Evaluate Trainer
Actor
Employee
Manipulate Customer
Open Course
Log Out
Branch Manager
Manipulate Trainer
Register
Activate Members
Close Course
Non-Member