Fitness Club Management Software (FCMS)

Team members (*filled by PM, Team Leader*)**:**

No	Name	Surname	Student ID	Role
1	Mert	Savaş	251643	*Backend Programmer.*
2	Ali Tahsin	Saribudak	239555	*Software Tester.*
3	Albert	Forner	250061	*Backend*
4	Chanchal	Kumari	239592	*Frontend Programmer.*
5	Thomas	Cionek	245586	*Frontend Programmer.*
6	Kainar	Kulzhin	250366	*Documentation and reporting.*
7	Anastasiia	Bilonizhko	236963	*UI Designer. Frontend.*
8	Raul	Nuñez	251632	*Frontend Designer.*
9	Mehmet	Isikhan	228931	*Database Designer.*
10	Ernesto	Vieira Manzanera	251663	*PM, Team Leader, Backend.*
11	Bogdan	Novikov	245873	*Frontend Programmer.*
12	Kutay	Guler	245906	*Documentation and reporting.*

Team members activity (*filled by PM, Team Leader after each class*)**:**

Team member	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	S	S	S	S	S	S	S	S	S	S	S	O	-	
2	S	S	O	S	S	S	S	S	S	S	S	S	S	
3	S	S	S	S	S	S	S	S	S	S	O	O	O	
4	S	S	S	S	S	S	S	S	S	S	S	-	S	
5	S	S	S	S	S	S	S	-	S	S	-	-	-	
6	S	S	S	S	S	S	S	-	S	S	-	-	S	
7	S	S	S	S	S	S	S	S	S	S	-	S	-	
8	S	S	S	S	S	S	S	S	S	S	S	S	S	
9	S	S	S	S	S	S	S	S	S	S	S	S	S	
10	S	S	S	S	O	S	S	S	S	S	S	S	S	
11	S	S	S	S	S	S	S	S	S	S	S	S	O	
12	S	S	S	S	S	S	S	-	S	S	-	-	S	

Activity symbols:

Symbol	Description
-	*Absent*
S	*Standard*
O	*Outstanding*

1. Elaboration of application concept (F1)

1.1. Aim of the Project

The project is meant to be the main support of a Fitness Center, taking care of the organization and administration as well as allowing the end-user to interact with an appropriate interface in order to make the most of the facilities. The project aims to make the processes including administrative and customer servicing simpler while being more efficient, time effective and fluid. This is in whole achieved by making a friendly customer interface that helps, on the one hand, with keeping track of personalized training programs and a schedule of all booked trainings, while also allowing administration to check the status and information about the customer and managing the classes & trainings.

With all the above, the fitness club will be armed with management tools that will allow it to not only be more user friendly & customer appealing, but also more efficient in terms of HR & resources management.

1.2. General Assumptions

- A fitness club has more than one room for their activities.
- A fitness club has at least 2 employees.
- A fitness club does not have a WEB-APP with an interface for users.
- A fitness club is having mismanagement and inventory issues which leads to file loss and damage.
- A fitness club has a time consuming financial and performance review process.
- A fitness accepts more than one type of payment methods.
- A fitness club has at least 1 type of activities (training session ex: zumba…).
- A fitness club sells products (Food, Drinks, material for training…).
- A fitness club has at least one manager with whole control over the system.
- A fitness club has too much paperwork which slows down the administrative procedures.
- Trainer records are saved in a space-consuming storage.
- Records need to be updated quickly and easily.
- Admission of both Employees and customers must be done with ease and swiftly.
- Orientation towards cost-reducing.
- Avoid data redundancy.

1.3. Stakeholders Description

The planned system differentiates among three types of users, each of them with predefined limitations and possibilities:

- **Customer**: the average user of the Fitness Club, who uses the facilities and pays for it. Their actions are very restricted to browsing and book activities, and very basic administration of their personal data and payment options. Does not take part in the club's administration and the system must be aware of that.
- **Employee**: employees of the fitness centre with basic administration features oriented to activity planning. As well as the normal User, they are able to edit and consult their personal information on the system.
- **Manager**: users with maximum privileges in the system. Allowed to modify any information related to any user or activity. Capable to deny or permit the access to the resources of the system to low-privilege users, within some constraints.

All these three identities may be referred to in this document as **Users** when there is no need to make a distinction among them.

Furthermore, we can identify a fourth stakeholder which is the **Product Owner** itself which will receive the revenues of the activity of the Fitness Club and thus have an interest on it.

Name	Description	Needs
Product owner	A representative of the organization that is buying or acquiring the software being developed.	● Sets and communicates the requirements of the system. ● Informs about work priorities, updates, and issues that emerge during the development process. ● Able to provide timely responses to queries ● Project financing. ● Project progress and course monitoring. ● Consulting/advising the project team on the timing and deadlines.
Gym Manager	Administrating the overall functioning of the fitness club, regarding both human and material resources. Furthermore, manages access of the clients to the club, and ensures the correct development of the fitness club.	● Watch over the human resources department of the centre, managing the addition and removal of workers into the system, and establish their role in the club. ● Manage employees' schedules. ● Manage activities offered to the clients by the club. ● Access clients information in order to add, modify, and remove users from the gym.

		● Overlook the finances of the fitness club. ● Look after the quality standards of the infrastructures with regards to the clients, keeping track of incidences. ● Notifies incidences regarding the application. ● Receive feedback from employees and clients to keep clients and employees' satisfaction. ● Guarantee the maximum productivity of the club.
Gym Employee	Part of the working staff of the fitness club presented by the software. They are going to use the platform in order to better manage their courses and stay in contact with the customers.	● Browses the services and resources that are available in the club. ● Notifies availability of the services & resources. ● Has to be able to see the list of customers that are enrolled in a specific service. ● Has to be able to see the history of sign-ups and payments of the customers. ● Has to be able to log in / log out of the system. ● Provides feedback about the system and reports bugs. ● Enrolls / Unenrolls users from the services.
Customer	Person who is going to use the application in order to take the facilities offered by the gym in the application.	● Have to be able to make a log in / log out the system. ● Pay the fee ● Browse courses ● Enroll the course ● Book a personal training with the trainer that he wants. ● Check and edit his profile ● Make a feedback of the courses that he did. ● Report a problem in the gym. ● Check the payments done
Maintenance Team	Group of people who collectively are responsible for maintenance after the system has been deployed.	● Receive reports of bugs & feedback about the system ● Answer queries ● Monitoring the system. ● Tracking and fixing bugs. ● Implementing functionalities for preventing new bugs.

1.4. Stakeholder needs

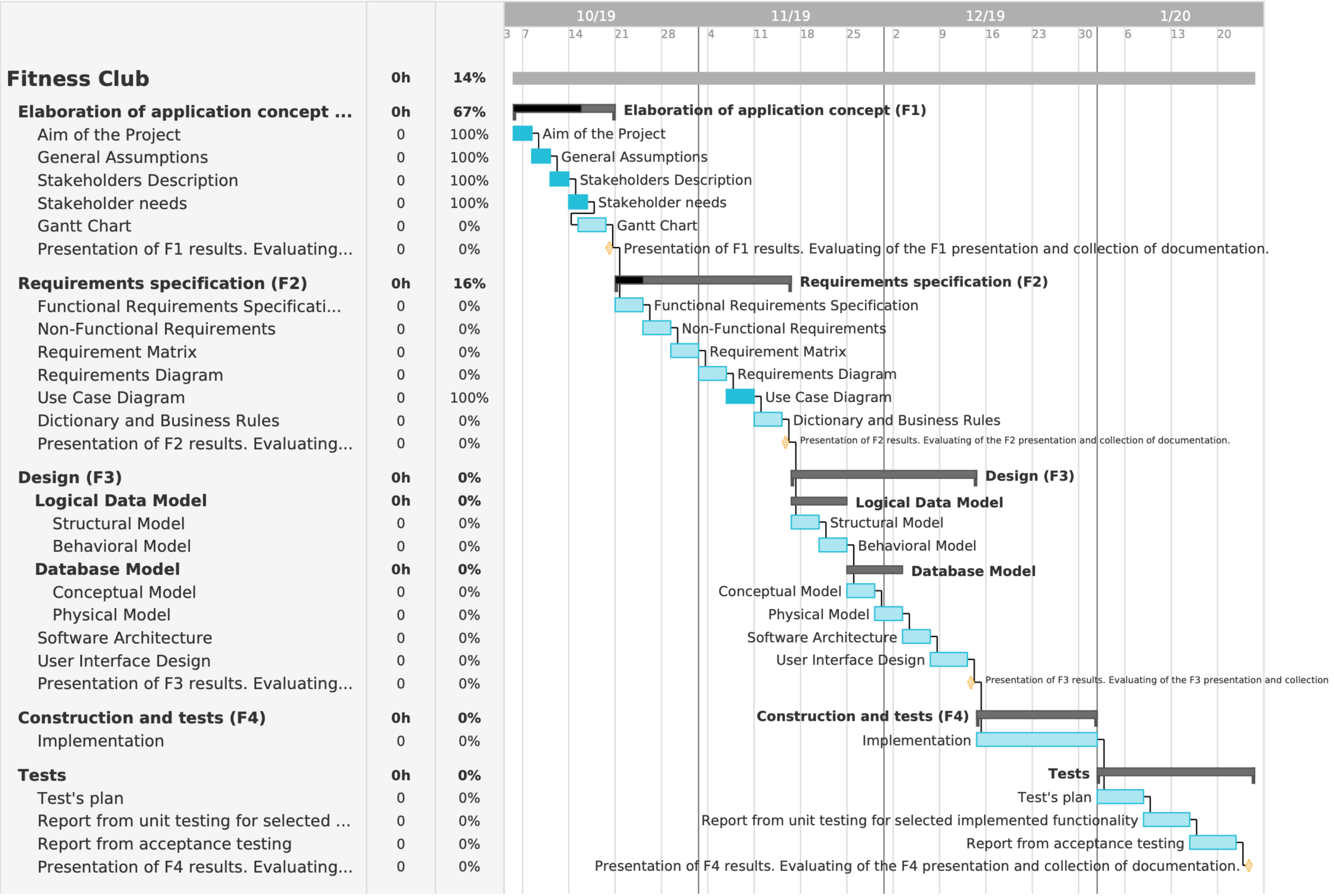
Need	Priority	Features	Stakeholder

Manage Clients	High	Module for check-in, check-out, modifying and deactivating the sign-ups for session, . This module will also provide the possibility to examine the history of payments & sign-ups	Employee
Management of employees (handling of the club's human resources)	High	Module for working with the employee's personal data, job position, salaries etc… This module will give HR managers better control over the information about employees, and allow them to manage this information in an efficient manner.	Manager
Manage expendable resources	Medium	Module for managing the expandable materials and resources of the club. This module will provide the means to create, modify, remove and examine the records of expendable resources. The system will automatically inform the relevant individuals about the lack of a certain resource (e.g. lack of rooms to organize a specific activity, incidences regarding the machines, lack of fitness equipment).	Manager, Employee
Financial Analysis	Low	Module for keeping track of the financial progress of the Club – this module will be available to the managers and the main purposes are as follows: giving the managers a better overview of financial performance, Identifying the most	Manager

		profitable products (Subscriptions included…), identifying the defects in the business model, etc...	
Managing enrolments (from the perspective of the guests)	High	Module for making, modifying and deleting enrolments as well as checking the history of it.	Manager, Employee
Check client information.	Medium	Interface that shows basic information about the client (Payments, Enrolments, days in the club….)	Manager, Employee, Client
Special Offers for Guests	Very Low	Providing special deals to the guests to attract more customers.	Client
Improve the quality of the service	Low	Online polls, and ability to make suggestions, feedback and complaints.	Client, Employee
Managing Enrolments (from the employees perspective)	High	Module for making, modifying and deleting enrolments. Checking the history of all enrolments for all the clients. Examining the history of payments as well as other supporting documents such as invoices, receipts etc.	Manager, Employee
Classes assignment	High	Checking & assigning coaches and trainers to the classes as well as checking available spots.	Employee

1.5. Gantt Chart

Gantt diagram depicting the distribution of planned activities over time, including the allocation of specific resources, including stakeholder groups for planned tasks.



Fitness Club	**0h**	**14%**
Elaboration of application concept ...	**0h**	**67%**
Aim of the Project	0	100%
General Assumptions	0	100%
Stakeholders Description	0	100%
Stakeholder needs	0	100%
Gantt Chart	0	0%
Presentation of F1 results. Evaluating...	0	0%
Requirements specification (F2)	**0h**	**16%**
Functional Requirements Specificati...	0	0%
Non-Functional Requirements	0	0%
Requirement Matrix	0	0%
Requirements Diagram	0	0%
Use Case Diagram	0	100%
Dictionary and Business Rules	0	0%
Presentation of F2 results. Evaluating...	0	0%
Design (F3)	**0h**	**0%**
Logical Data Model	**0h**	**0%**
Structural Model	0	0%
Behavioral Model	0	0%
Database Model	**0h**	**0%**
Conceptual Model	0	0%
Physical Model	0	0%
Software Architecture	0	0%
User Interface Design	0	0%
Presentation of F3 results. Evaluating...	0	0%
Construction and tests (F4)	**0h**	**0%**
Implementation	0	0%
Tests	**0h**	**0%**
Test's plan	0	0%
Report from unit testing for selected ...	0	0%
Report from acceptance testing	0	0%
Presentation of F4 results. Evaluating...	0	0%

10/19
11/19
12/19
1/20
3
7
14
21
28
4
11
18
25
2
9
16
23
30
6
13
20
Elaboration of application concept (F1)
Aim of the Project
General Assumptions
Stakeholders Description
Stakeholder needs
Gantt Chart
Presentation of F1 results. Evaluating of the F1 presentation and collection of documentation.
Requirements specification (F2)
Functional Requirements Specification
Non-Functional Requirements
Requirement Matrix
Requirements Diagram
Use Case Diagram
Dictionary and Business Rules
Presentation of F2 results. Evaluating of the F2 presentation and collection of documentation.
Design (F3)
Logical Data Model
Structural Model
Behavioral Model
Database Model
Conceptual Model
Physical Model
Software Architecture
User Interface Design
Presentation of F3 results. Evaluating of the F3 presentation and collection
Construction and tests (F4)
Implementation
Tests
Test's plan
Report from unit testing for selected implemented functionality
Report from acceptance testing
Presentation of F4 results. Evaluating of the F4 presentation and collection of documentation.

2. Requirements specification (F2)

2.1. Functional Requirements Specification

Name	Source	Significance	Type
R1: Employee requirements.			
R1.1: The system shall allow manager to view the Employee's data provided their Employee ID or Name	Stakeholder: Manager.	MUST	Data Exchange., Database.
R1.2: The system shall allow the manager to view whole data about the Employee.	Stakeholder: Manager, Employee.	MUST	Data Exchange.
R1.3: The system shall allow the manager to add Employee data.	Stakeholder: Manager, Employee.	MUST	Data Exchange.
R1.4: The system shall allow the manager to edit Employee data	Stakeholder: Manager, Employee	MUST	Data Exchange.
R1.5: The system shall allow the manager to delete Employee's data when they resign or fired.	Stakeholder: Manager.	MUST	Interface. Database.
R2: Login			
R2.1: The system shall allow users access the system using their personal account data: email and password.	Stakeholder: Customer, Company.	MUST	Authentication. Database.
R2.2: The system shall allow users to change their passwords to achieve security.	Stakeholder: Customer.	MUST	Authentication.
R2.3: The system shall allow up to 4 failed login attempts before blocking the access to the account.	Stakeholder: Customer, Company.	SHOULD	Authentication.
R2.4: The system shall allow password recovery.	Stakeholder: Customer, Employee, Manager, Company.	SHOULD	Authentication. Interface.
R3: Accounting			
R3.1: The system shall allow Customers to create their personal account.	Stakeholder: Customer, Employee, Manager, Company**.**	MUST	Authentication.

R3.2: The system shall allow manager to create Customer accounts.	Stakeholder: manager	SHOULD	Interface.
R3.3: The system shall allow manager to create Employee accounts.	Stakeholder: manager	MUST	Interface.
R3.4: The system shall allow manager users to deactivate any account.	Stakeholder: manager, Customer, Employee.	MUST	Interface.
R3.5: The system shall allow manager users to activate any account.	Stakeholder: manager, Employee, Customer.	SHOULD	Interface.
R3.6: The system shall allow manager users to delete any account.	Stakeholder: manager	SHOULD	Interface.
R4: Activity Scheduler			
R4.1: The system shall allow the manager to add new activities into the schedule.	Stakeholder: Manger, Customer.	MUST	Interface. Data Exchange.
R4.2: The system shall allow the manager to delete punctual activities.	Stakeholder: Manager.	MUST	Interface.
R4.3: The system shall allow the manager to schedule activities on a weekly repeated basis.	Stakeholder: Manager.	MUST	Interface.
R4.4: The system shall allow Employee to view activity details in the schedule.	Stakeholder: Employee.	MUST	Interface. Data Exchange.
R4.5: The system shall allow Customers to view activity details in the schedule.	Stakeholder: Customer.	MUST	Interface. Data Exchange.
R5: Activity Enrollment			
R5.1: The system shall allow Employee to view number of Customers enrolled in the activity.	Stakeholder: Employee.	SHOULD	Interface. Data Exchange.
R5.2: The system shall allow Customers to enroll and unenrol in activities.	Stakeholder: Customer.	MUST	Interface.
R5.3: The system shall allow manager set which activities have a maximum of enrolled Customers.	Stakeholder: Manager, Employee.	MUST	Interface.
R5.4: The system shall not allow Customers to enrol in fully occupied activities.	Stakeholder: Customer.	MUST	Interface.
R6: Plan and Membership			
R6.1: The system shall allow manager to	Stakeholder:	MUST	Data Exchange.

add new plans and their details: name, price, conditions, duration, availability.	Manager, Company.		Interface
R6.2: The system shall allow manager to edit plan details.	Stakeholder: Manager, Company.	MUST	Data Exchange. Interface.
R6.3: The system shall allow manager to delete plan.	Stakeholder: Manager.	MUST	Interface.
R6.4:The system shall allow users to view plan details.	Stakeholder: Customer, Company.	MUST	Interface.
R7: Payment			
R7.1: The system shall allow the Customer to perform payment online.	Marketing. Stakeholder: Customer, Manager.	SHOULD	Interface, Authentification
R7.2: Payment shall be performed by an authorised third party.	Legal. Stakeholder: Customer.	MUST	Authentification
R7.3: The system shall automatically deactivate user account when payment is delayed a monthly pay.	Stakeholder: Company.	MUST	Database, Interface
R7.4: The system shall allow manager to view user's payment status.	Stakeholder: Manager, Company.	MUST	Interface, Database.
R7.5: The system shall allow payment by direct debit if bank account number is provided.	Stakeholder: Company.	MUST	Software, Authentification
R7.6: The system shall not allow any user to see user's bank account information.	Stakeholder: customer.	MUST	Security
R8: Logout			
R8.1: The system shall allow users to logout of the system.	Stakeholder: customer.	MUST	Interface. Authentication.

2.2. Non-Functional Requirements

Non-functional requirements are related to the quality and performance of the system under certain conditions, not to the functionalities offered. These qualities can be divided into X groups:

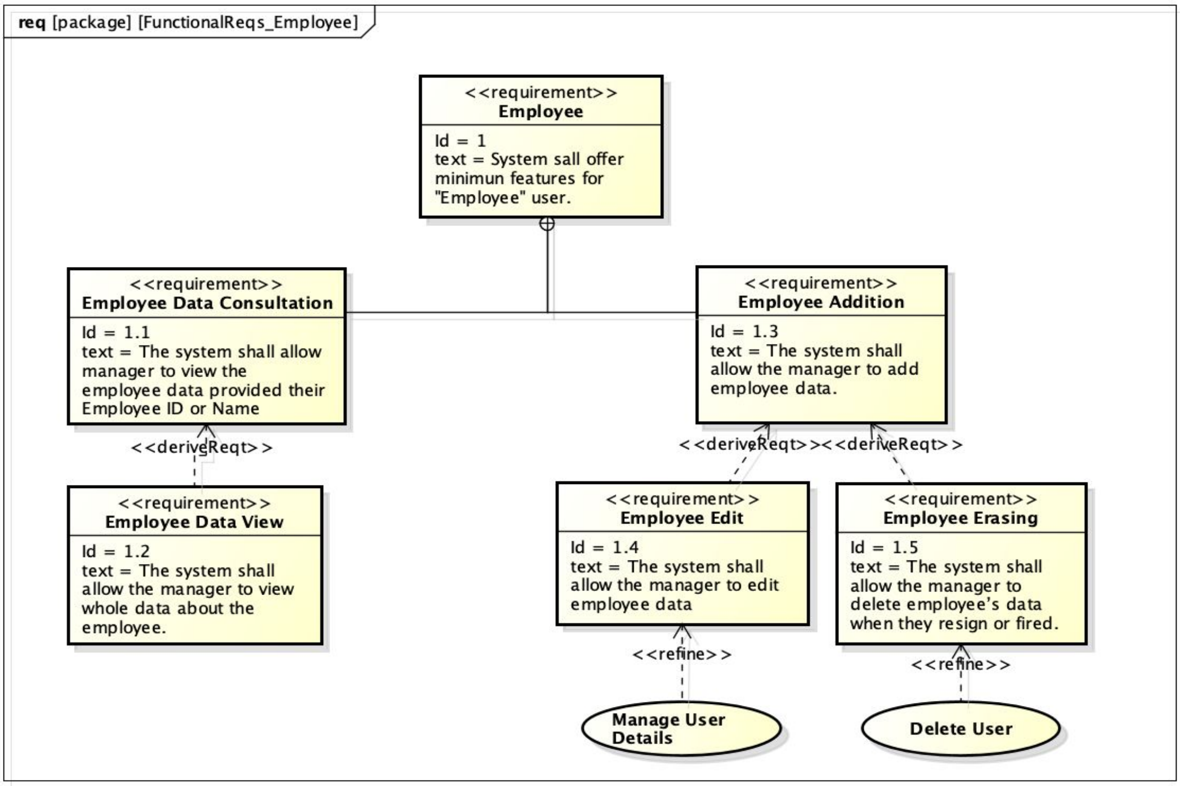
Name	Source	Significance
R9: Security		

R9.1: Data can only be accessed by the user authorised to view it.	Stakeholders.	MUST
R9.2: Data can only be edited by the user authorised to edit.	Stakeholders.	MUST
R9.3: User shall be asked by the system to enter Username and Password to authenticate in the system.	Stakeholders.	MUST
R9.4: Passwords shall be stored encrypted.	Stakeholder. Legal requirements.	MUST
R9.5: Connection should be encrypted end-to-end in the following procedures: Authentication, Signing up, Bank operations.	Stakeholder. Legal requirements.	MUST
R10: Reliability		
R10.1: System should be available with an Internet connection and power supply.	Hardware definitions.	MUST
R10.2: Any maintenance that supposes unavailability of the system shall be communicated to all the users.	Stakeholder.	SHOULD
R10.3: A copy of the user data should be kept for backup purposes.	Stakeholder.	MUST
R10.4: System shall allow fast and swift changes without preventing user to use the application.	Stakeholder: customer.	SHOULD
R11: Portability		
R11.1: System shall run in Windows, Mac and Linux OS.	Stakeholder: customer.	SHOULD
R11.2: System shall run in every browser.	Stakeholder: Customer.	SHOULD
R12: Usability		
R12.1: Documentation for the use of the application shall be included.	Stakeholder: Customer.	MUST
R12.2: System shall complete every procedure within a maximum time of 30 seconds.	Stakeholder: Customer, Manager, User. Marketing.	MUST
R12.3: System shall keep user informed about procedures taking place at the moment.	Stakeholder: Customer, Manager, User.	SHOULD
R12.3: Technical and usability documentation shall be included.	Concept of Vision Document.	SHOULD

	R1.1:The system shall allow manager to view the employee provided their employee ID or Name	R1.3: The system shall allow the manager to add employee data.	R1.5: The system shall allow the manager to delete employee's data when they resign or fired.	R2.1: The system shall allow users access the system using their personal account data: email and password.	R2.2: The system shall allow users to change their passwords to achieve security	R3.1: The system shall allow Customers to create their personal account.	R3.2: The system shall allow manager to create Customer accounts.	R3.3: The system shall allow manager to create employee accounts.	R3.4: The system shall allow manager users to deactivate any account.	R4.1: The system shall allow the manager to add new activities into the schedule.	R4.3: The system shall allow the manager to schedule activities on a weekly repeated basis.	R4.5: The system shall allow Customers to view activity details in the schedule.	R5.2: The system shall allow Customers to enroll and unenrol in activities.	R5.3: The system shall allow manager set which activities have a maximum of enrolled	R5.4: The system shall not allow Customers to enrol in fully occupied activities.
Company	4	4	5	5	3	4	4	4	5	4	4	4	3	3	4
Customer	0	0	0	5	4	4	4	0	0	3	2	5	4	3	4
Employee	3	4	5	5	4	0	0	4	0	3	2	0	3	4	4
Manager	5	5	5	5	4	3	4	4	4	4	5	0	4	5	3

	R6.1: The system shall allow manager to add new plans and their details: name, price,	R7.1: The system shall allow the Customer to perform payment online.	R7.2: Payment shall be performed by an authorised third party.	R7.3: The system shall automatically deactivate user account when payment is delayed	R8.1: The system shall allow users to logout of the system.	R9.1: Data can only be accessed by the user authorised to view it.	R9.2: Data can only be edited by the user authorised to edit.	R9.3: User shall be asked by the system to enter Username and Password to authenticate in the system	R9.4: Passwords shall be stored encrypted.	R9.5: Connection should be encrypted end-to-end in the following procedures: Authentication, Signing up, Bank operations.	R10.3: A copy of the user data should be kept for backup purposes.	R11.1: System shall run in Windows, Mac and Linux OS.	R11.2: System shall run in every browser.	R12.1: Documentation for the use of the application shall be included.	R12.2: System shall complete every procedure within a maximum time of 30 seconds.
Company	5	4	4	5	3	3	3	3	4	5	4	3	3	3	4
Customer	4	5	5	0	4	5	5	4	5	5	3	4	4	4	4
Employee	0	0	0	0	3	5	5	4	5	5	3	2	2	2	4
Manager	5	1	2	3	3	5	5	4	5	5	4	2	2	2	4

2.4. Requirements Diagram



req [package] [FunctionalReqs_Employee]

<<requirement>>
Employee
Id = 1
text = System sall offer minimun features for "Employee" user.

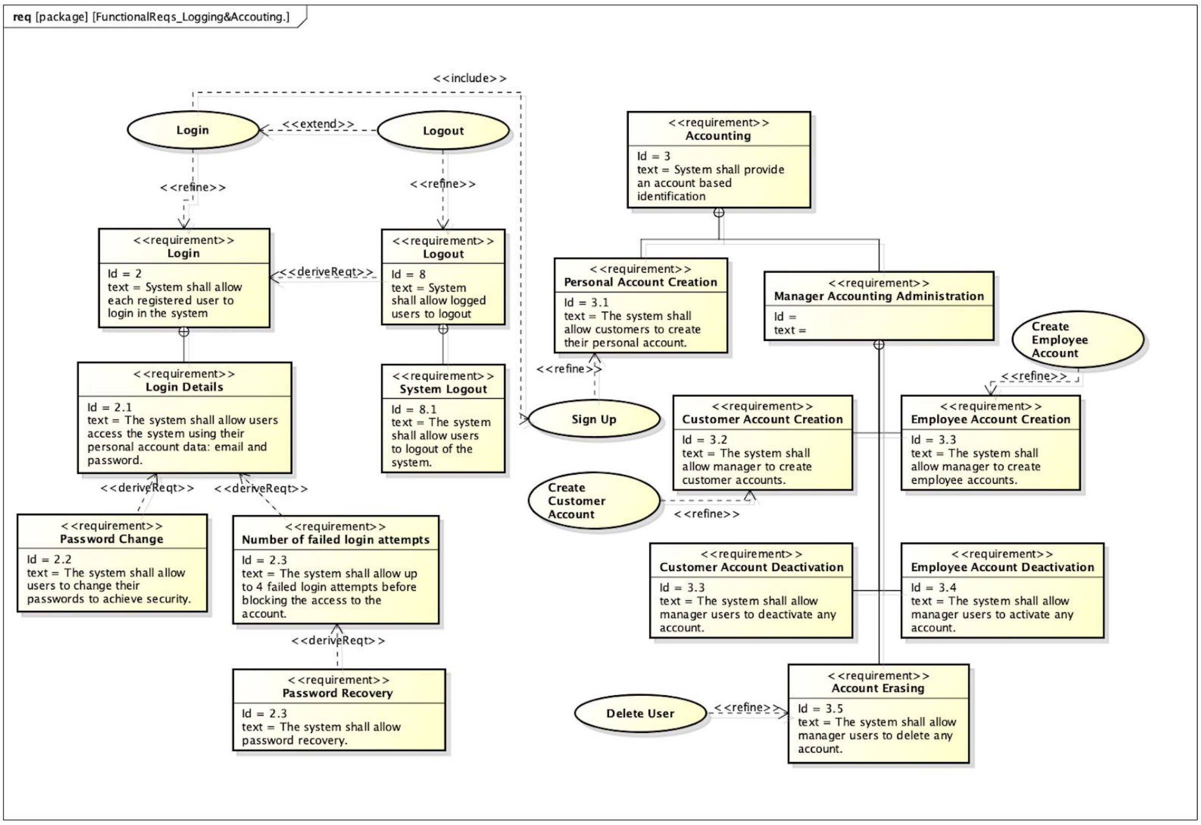
<<requirement>>
Employee Data Consultation
Id = 1.1
text = The system shall allow manager to view the employee data provided their Employee ID or Name

<<requirement>>
Employee Addition
Id = 1.3
text = The system shall allow the manager to add employee data.

<<requirement>>
Employee Data View
Id = 1.2
text = The system shall allow the manager to view whole data about the employee.

<<requirement>>
Employee Edit
Id = 1.4
text = The system shall allow the manager to edit employee data

<<requirement>>
Employee Erasing
Id = 1.5
text = The system shall allow the manager to delete employee's data when they resign or fired.



req [package] [FunctionalReqs_Logging&Accouting.]

<<requirement>>
Login
Id = 2
text = System shall allow each registered user to login in the system

<<requirement>>
Logout
Id = 8
text = System shall allow logged users to logout

<<requirement>>
Login Details
Id = 2.1
text = The system shall allow users access the system using their personal account data: email and password.

<<requirement>>
System Logout
Id = 8.1
text = The system shall allow users to logout of the system.

<<requirement>>
Password Change
Id = 2.2
text = The system shall allow users to change their passwords to achieve security.

<<requirement>>
Number of failed login attempts
Id = 2.3
text = The system shall allow up to 4 failed login attempts before blocking the access to the account.

<<requirement>>
Password Recovery
Id = 2.3
text = The system shall allow password recovery.

<<requirement>>
Accounting
Id = 3
text = System shall provide an account based identification

<<requirement>>
Personal Account Creation
Id = 3.1
text = The system shall allow customers to create their personal account.

<<requirement>>
Manager Accounting Administration
Id =
text =

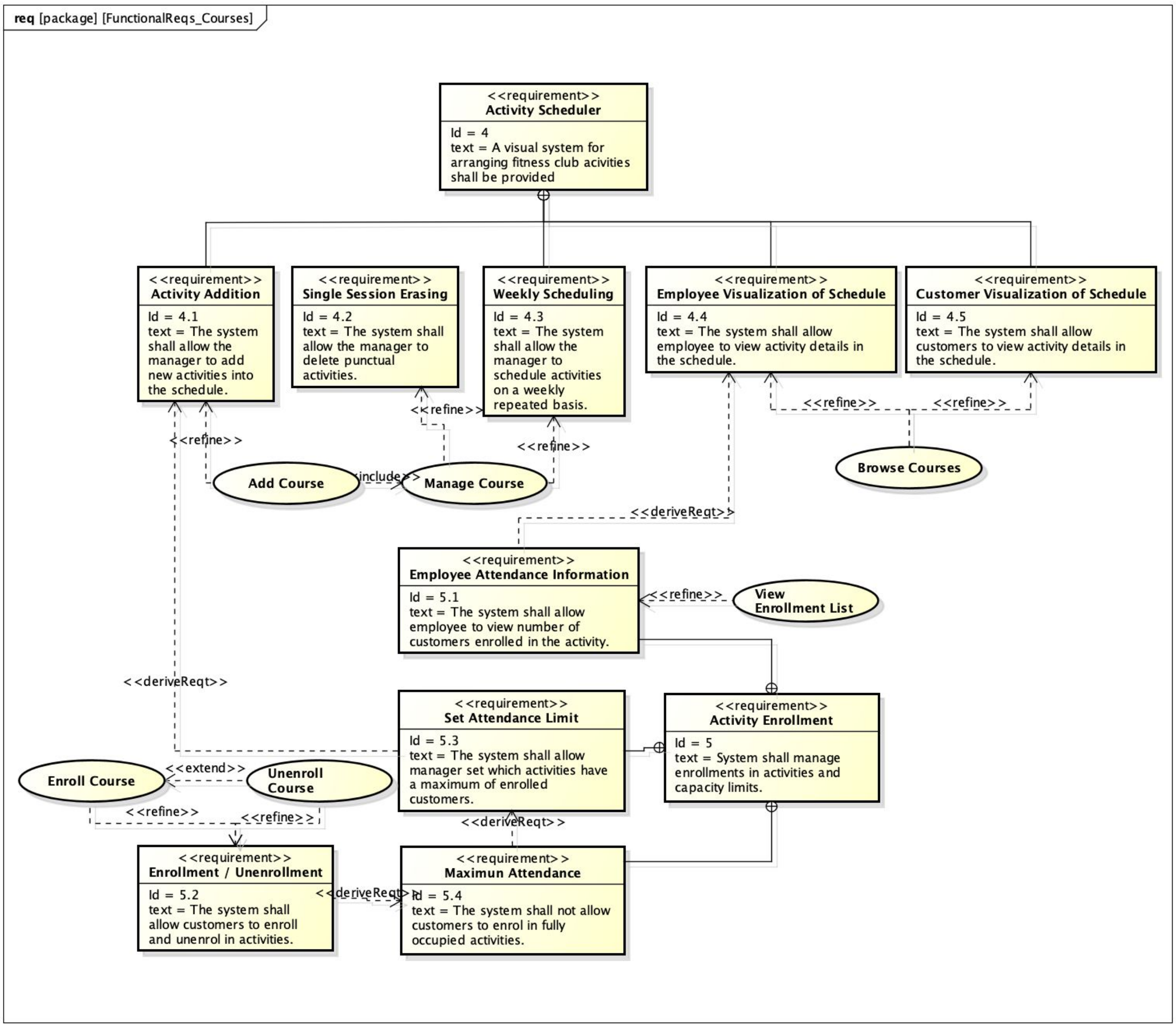
<<requirement>>
Customer Account Creation
Id = 3.2
text = The system shall allow manager to create customer accounts.

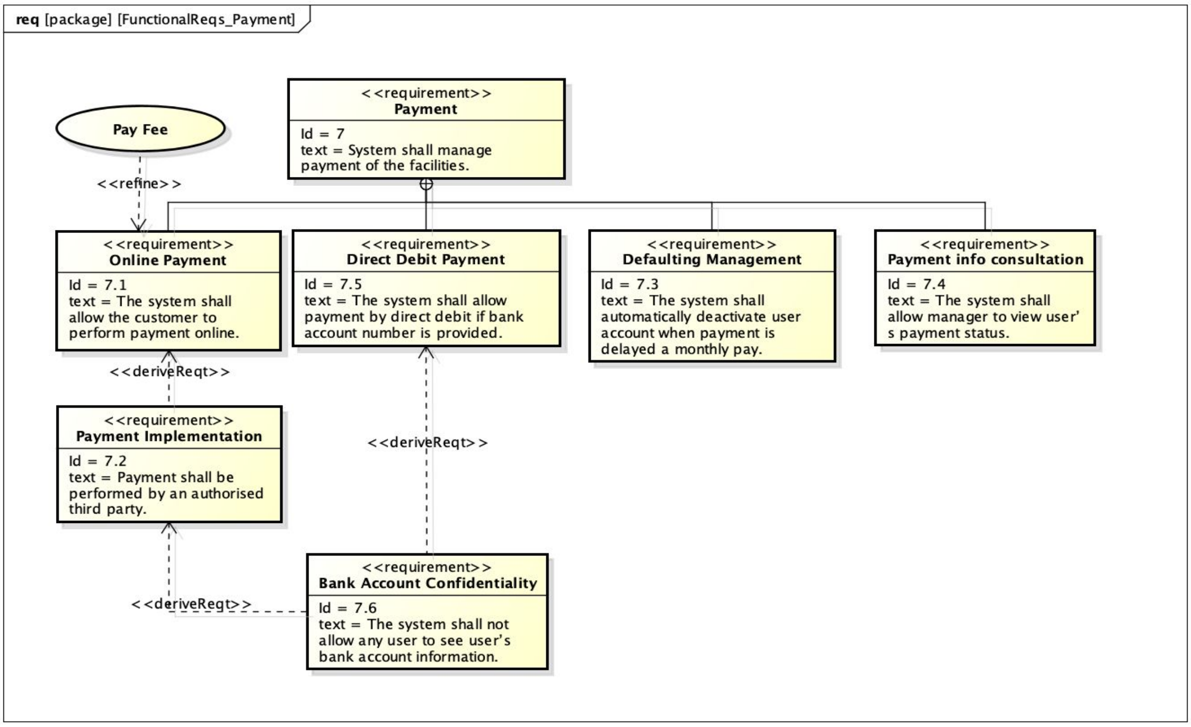
<<requirement>>
Employee Account Creation
Id = 3.3
text = The system shall allow manager to create employee accounts.

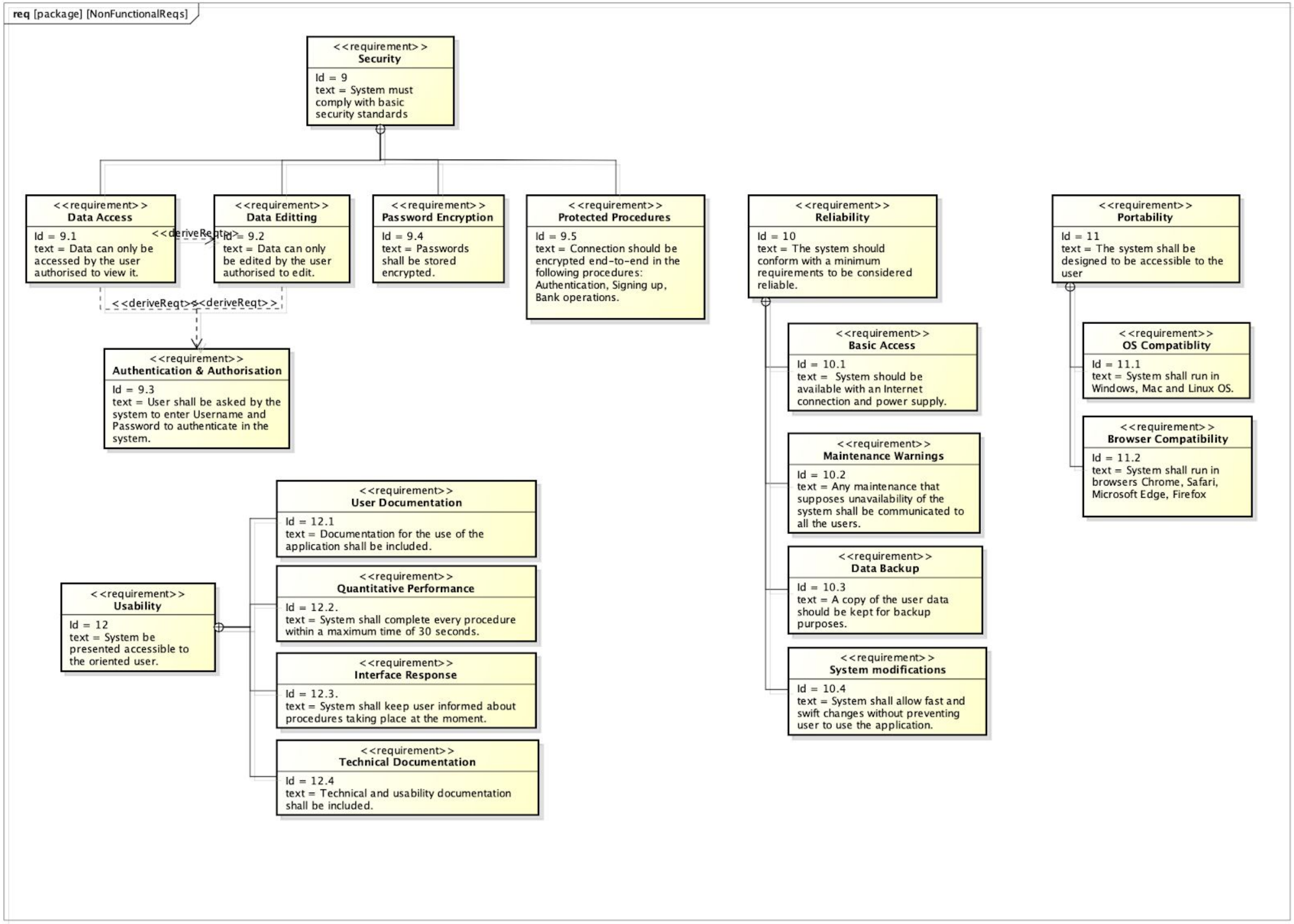
<<requirement>>
Customer Account Deactivation
Id = 3.3
text = The system shall allow manager users to deactivate any account.

<<requirement>>
Employee Account Deactivation
Id = 3.4
text = The system shall allow manager users to activate any account.

<<requirement>>
Account Erasing
Id = 3.5
text = The system shall allow manager users to delete any account.


req [package] [FunctionalReqs_Courses]
<<requirement>>
Activity Scheduler
Id = 4
text = A visual system for arranging fitness club acivities shall be provided
<<requirement>>
Activity Addition
Id = 4.1
text = The system shall allow the manager to add new activities into the schedule.
<<requirement>>
Single Session Erasing
Id = 4.2
text = The system shall allow the manager to delete punctual activities.
<<requirement>>
Weekly Scheduling
Id = 4.3
text = The system shall allow the manager to schedule activities on a weekly repeated basis.
<<requirement>>
Employee Visualization of Schedule
Id = 4.4
text = The system shall allow employee to view activity details in the schedule.
<<requirement>>
Customer Visualization of Schedule
Id = 4.5
text = The system shall allow customers to view activity details in the schedule.
<<refine>>
<<refine>>
<<refine>>
<<refine>>
<<refine>>
Add Course
<<include>>
Manage Course
Browse Courses
<<deriveReqt>>
<<requirement>>
Employee Attendance Information
Id = 5.1
text = The system shall allow employee to view number of customers enrolled in the activity.
<<refine>>
View Enrollment List
<<deriveReqt>>
<<requirement>>
Set Attendance Limit
Id = 5.3
text = The system shall allow manager set which activities have a maximum of enrolled customers.
<<requirement>>
Activity Enrollment
Id = 5
text = System shall manage enrollments in activities and capacity limits.
Enroll Course
<<extend>>
Unenroll Course
<<refine>>
<<refine>>
<<deriveReqt>>
<<requirement>>
Enrollment / Unenrollment
Id = 5.2
text = The system shall allow customers to enroll and unenrol in activities.
<<deriveReqt>>
<<requirement>>
Maximun Attendance
Id = 5.4
text = The system shall not allow customers to enrol in fully occupied activities.


req [package] [FunctionalReqs_Payment]
Pay Fee
<<refine>>
<<requirement>>
Payment
Id = 7
text = System shall manage payment of the facilities.
<<requirement>>
Online Payment
Id = 7.1
text = The system shall allow the customer to perform payment online.
<<requirement>>
Direct Debit Payment
Id = 7.5
text = The system shall allow payment by direct debit if bank account number is provided.
<<requirement>>
Defaulting Management
Id = 7.3
text = The system shall automatically deactivate user account when payment is delayed a monthly pay.
<<requirement>>
Payment info consultation
Id = 7.4
text = The system shall allow manager to view user's payment status.
<<deriveReqt>>
<<requirement>>
Payment Implementation
Id = 7.2
text = Payment shall be performed by an authorised third party.
<<deriveReqt>>
<<requirement>>
Bank Account Confidentiality
Id = 7.6
text = The system shall not allow any user to see user's bank account information.
<<deriveReqt>>



<<requirement>>
Security
Id = 9
text = System must comply with basic security standards

<<requirement>>
Data Access
Id = 9.1
text = Data can only be accessed by the user authorised to view it.

<<requirement>>
Data Editting
Id = 9.2
text = Data can only be edited by the user authorised to edit.

<<requirement>>
Authentication & Authorisation
Id = 9.3
text = User shall be asked by the system to enter Username and Password to authenticate in the system.

<<requirement>>
Password Encryption
Id = 9.4
text = Passwords shall be stored encrypted.

<<requirement>>
Protected Procedures
Id = 9.5
text = Connection should be encrypted end-to-end in the following procedures: Authentication, Signing up, Bank operations.

<<requirement>>
Reliability
Id = 10
text = The system should conform with a minimum requirements to be considered reliable.

<<requirement>>
Basic Access
Id = 10.1
text = System should be available with an Internet connection and power supply.

<<requirement>>
Maintenance Warnings
Id = 10.2
text = Any maintenance that supposes unavailability of the system shall be communicated to all the users.

<<requirement>>
Data Backup
Id = 10.3
text = A copy of the user data should be kept for backup purposes.

<<requirement>>
System modifications
Id = 10.4
text = System shall allow fast and swift changes without preventing user to use the application.

<<requirement>>
Portability
Id = 11
text = The system shall be designed to be accessible to the user

<<requirement>>
OS Compatiblity
Id = 11.1
text = System shall run in Windows, Mac and Linux OS.

<<requirement>>
Browser Compatibility
Id = 11.2
text = System shall run in browsers Chrome, Safari, Microsoft Edge, Firefox

<<requirement>>
Usability
Id = 12
text = System be presented accessible to the oriented user.

<<requirement>>
User Documentation
Id = 12.1
text = Documentation for the use of the application shall be included.

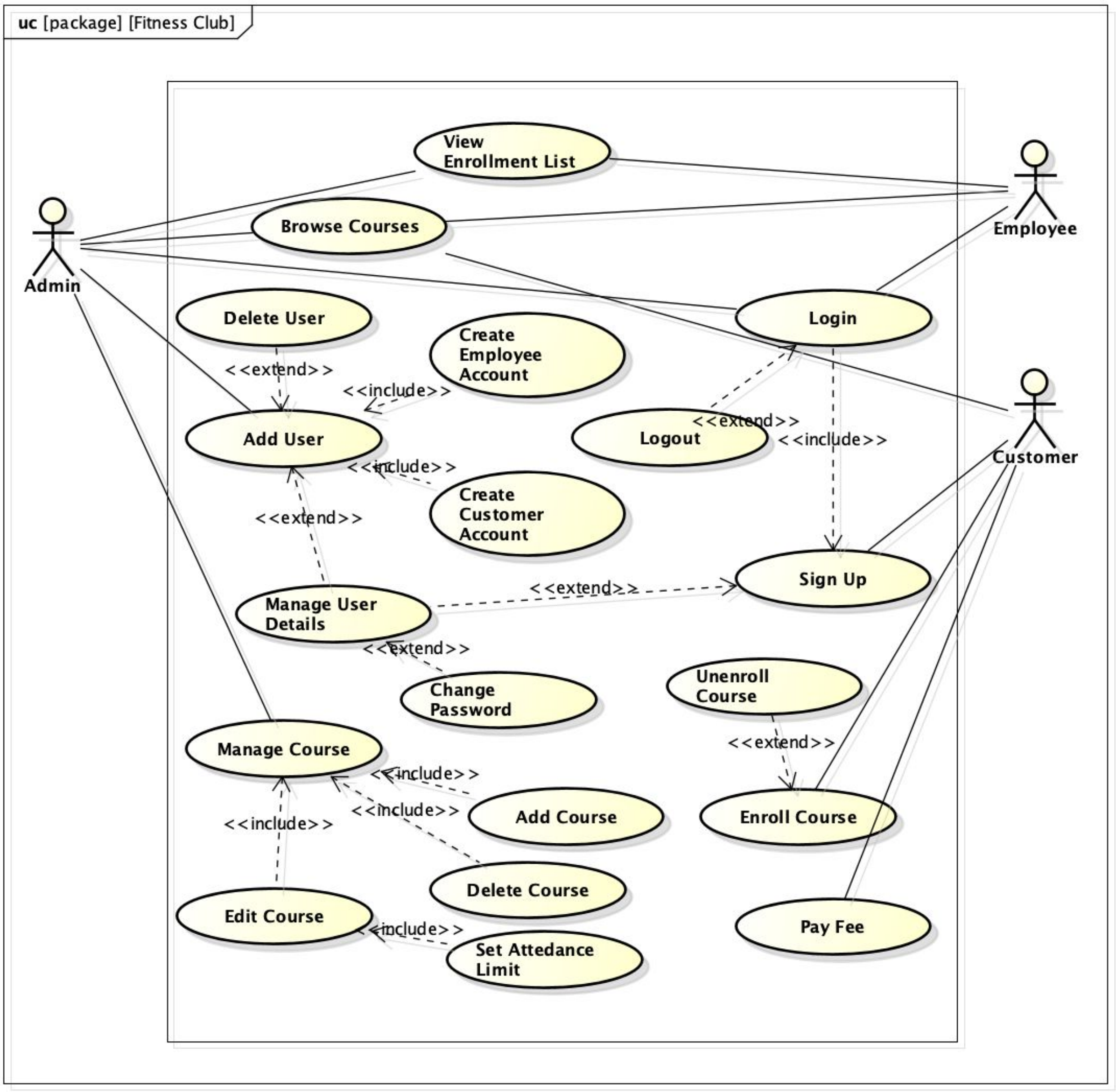
<<requirement>>
Quantitative Performance
Id = 12.2.
text = System shall complete every procedure within a maximum time of 30 seconds.

<<requirement>>
Interface Response
Id = 12.3.
text = System shall keep user informed about procedures taking place at the moment.

<<requirement>>
Technical Documentation
Id = 12.4
text = Technical and usability documentation shall be included.

2.5. Use Case Diagram

The Use Case Diagram depicts how the system interacts with the external actors. Each functionality that the system must provide is represented with its relations with the users and the other functionalities. In this diagram we have included three actors, which were defined in point *1.3 Stakeholder Definition.*



2.6. Dictionary and Business Rules

Business Rules are a set of statements that define behaviours and constraints within the context of a business. For the Fitness Club, we can identify the following patterns concerning different actors:

1. **Manager Rules**

- **MR1_Employee Attendance Checking Rule**: If an employee is not able to attend a scheduled service, a notification is sent to the manager.

- **MR2_Low Inventory Checking Rule**: If the inventory on a certain product is running low, a warning is sent to the manager.

2. **Employee Rules**

- **ER1_Low Inventory Checking Rule**: If a certain inventory is running low, a warning is sent to the employee.
- **ER2_Course Complete Notification Rule**: When a given course has all its places occupied, the system notifies the employee that teaches that course so he can ask another class to be arranged.
- **ER3_Course Cancellation Notification**: If an employee needs to cancel a class, a notification must be sent to the manager(s) in order to have their approval and formalize it.

3. **Customer Rules**

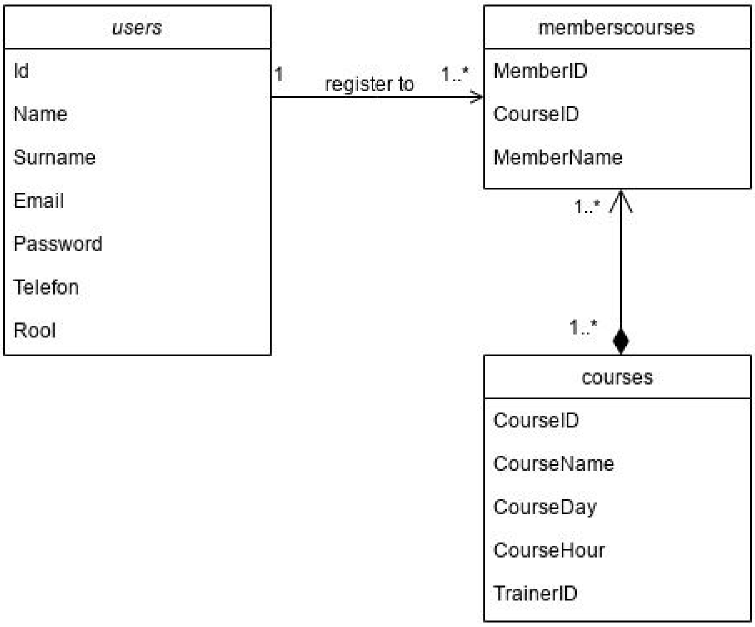
- **CR1_Course Cancellation Notification Rule**: When a given course is cancelled due to unavailability of the employee that teaches it, a notification is sent to the customers that were signed up for it.
- **CR2_Enrollment Payment Problem Notification Rule**: When a payment for a course is not successful, a notification is sent to the customer within the first 30 minutes after the payment.
- **CR3_Enrollment Payment Success Notification Rule**: When a payment is successful, a notification is sent to the customer informing him of the operation within the first 30 minutes after the successful payment.
- **CR4_Special Offer Notification Rule**: When a special offer is created, a notification is sent to all the customers that might be interested in it.
- **CR5_Platform Registration Notification Rule**: When a new user has registered in the platform a notification is sent to him informing him of the fact that the registration has taken place.
- **CR6_Account Authentication**: When the user first access the system, they will be asked to validate their contact details (email or phone number).
- **CR7_Status of Payments Checking Rule**: If the customer is delayed one whole monthly pay in the payment of the fee, their account is set inactive and a notification is sent to the customer. The access to the club will be forbidden until a manager reactivates their account.

3. Design

3.1. Logical Data Model

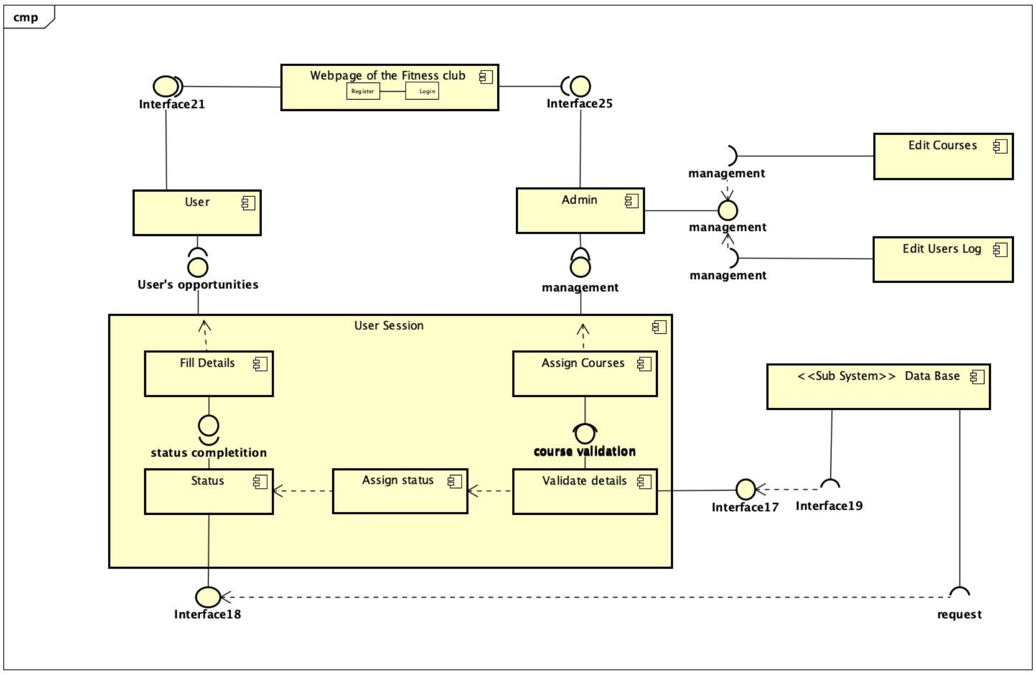
3.1.1. Structural Model

Class Diagram --

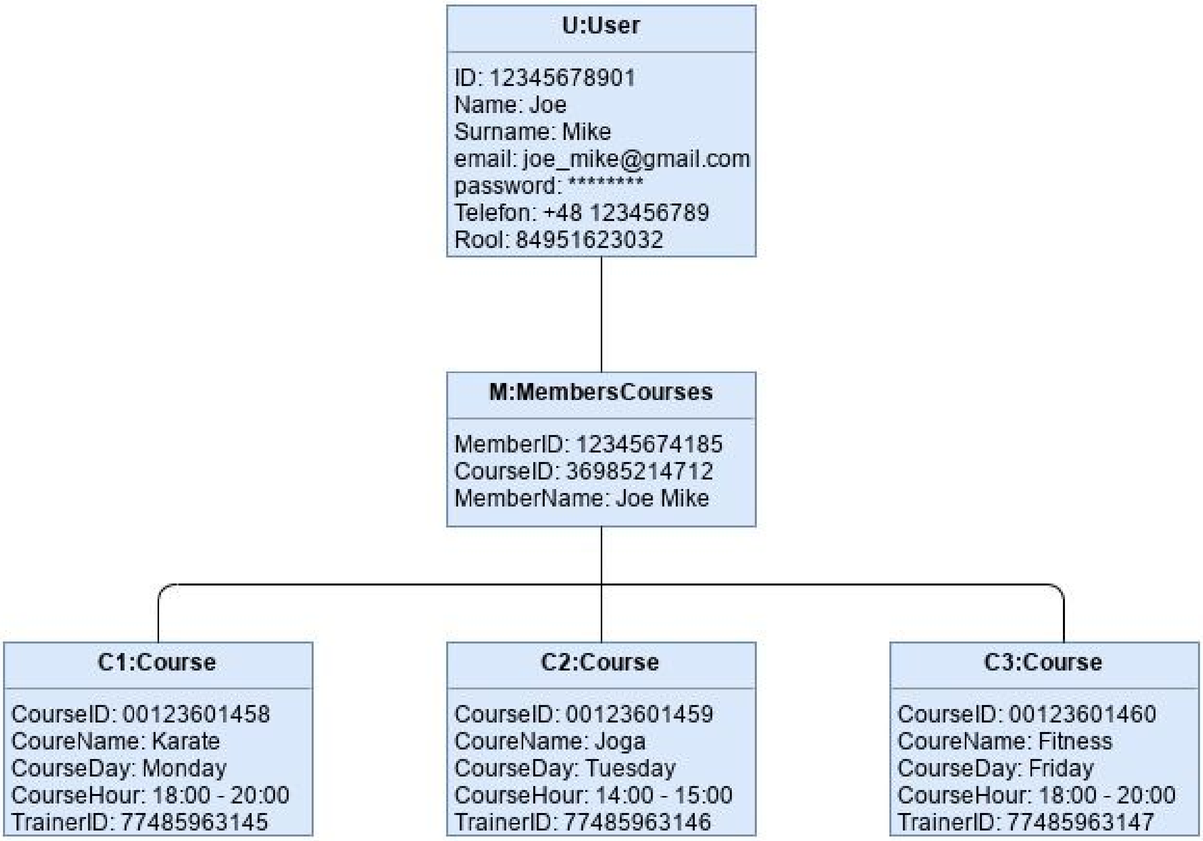


***Class Diagram** for Fitness Club Management System*

Component Diagram --



Component Diagram for Fitness Club Management System.



U:User

ID: 12345678901
Name: Joe
Surname: Mike
email: joe_mike@gmail.com
password: ********
Telefon: +48 123456789
Rool: 84951623032

M:MembersCourses

MemberID: 12345674185
CourseID: 36985214712
MemberName: Joe Mike

C1:Course

CourseID: 00123601458
CoureName: Karate
CourseDay: Monday
CourseHour: 18:00 - 20:00
TrainerID: 77485963145

C2:Course

CourseID: 00123601459
CoureName: Joga
CourseDay: Tuesday
CourseHour: 14:00 - 15:00
TrainerID: 77485963146

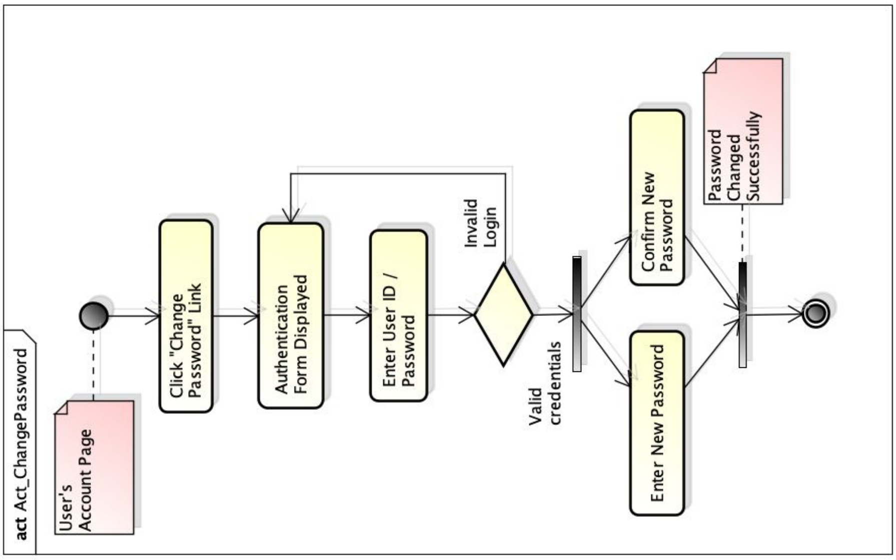
C3:Course

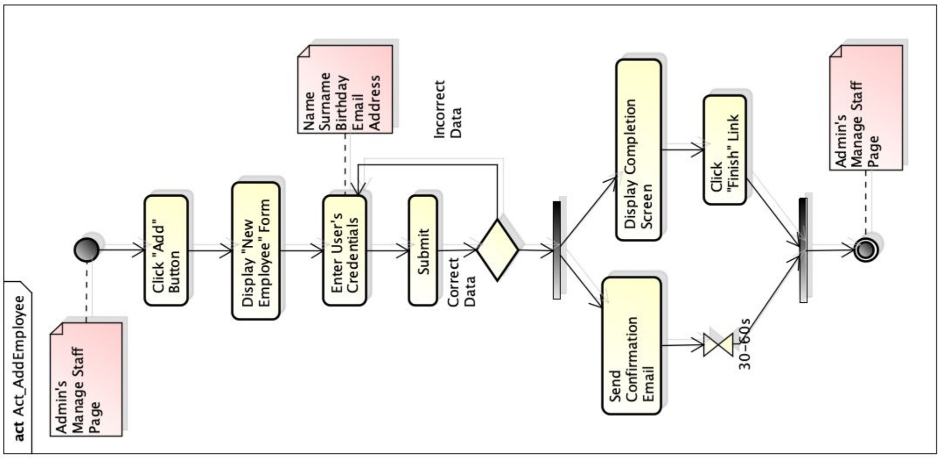
CourseID: 00123601460
CoureName: Fitness
CourseDay: Friday
CourseHour: 18:00 - 20:00
TrainerID: 77485963147

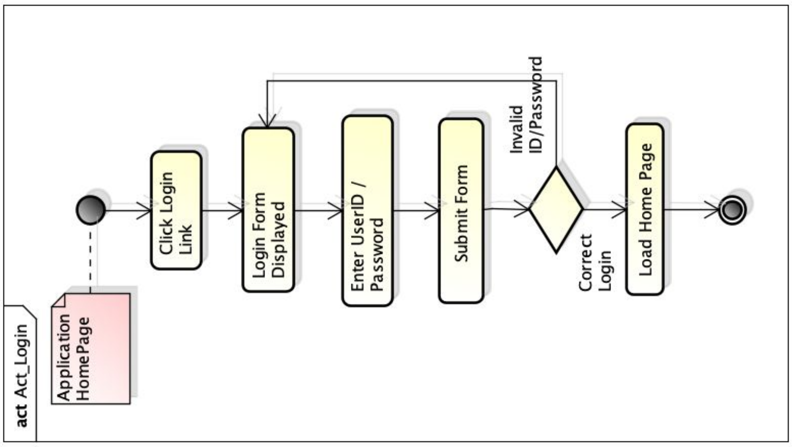
Object Diagram for Fitness Club Management System.

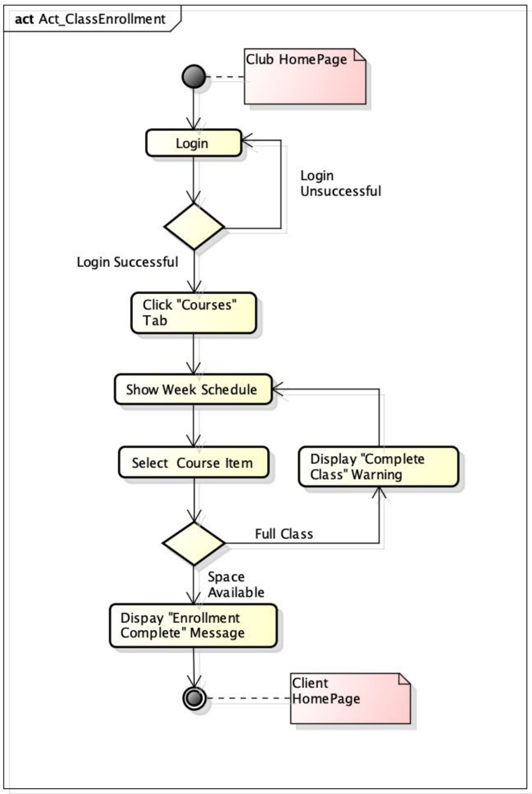
3.1.2. Behavioral Model

Activity Diagrams ---



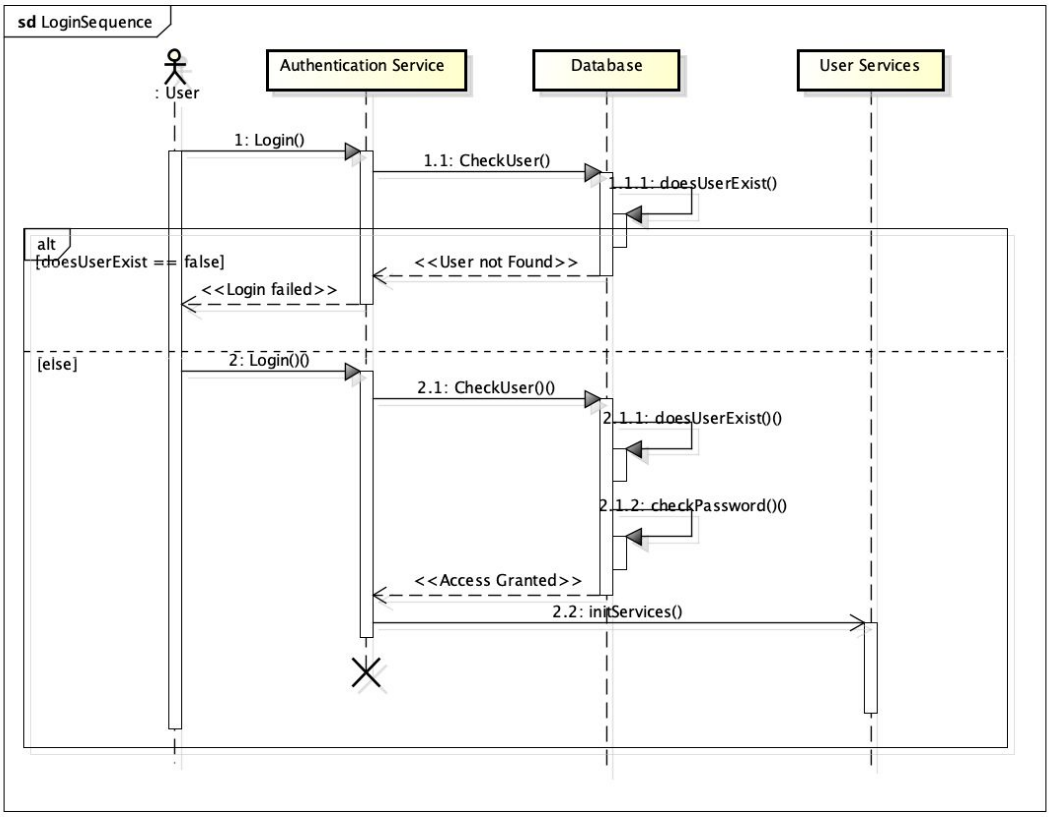

act Act_AddEmployee
Admin's Manage Staff Page
Click "Add" Button
Display "New Employee" Form
Name Surname Birthday Email Address
Enter User's Credentials
Submit
Incorrect Data
Correct Data
Display Completion Screen
Click "Finish" Link
Send Confirmation Email
30-60s
Admin's Manage Staff Page

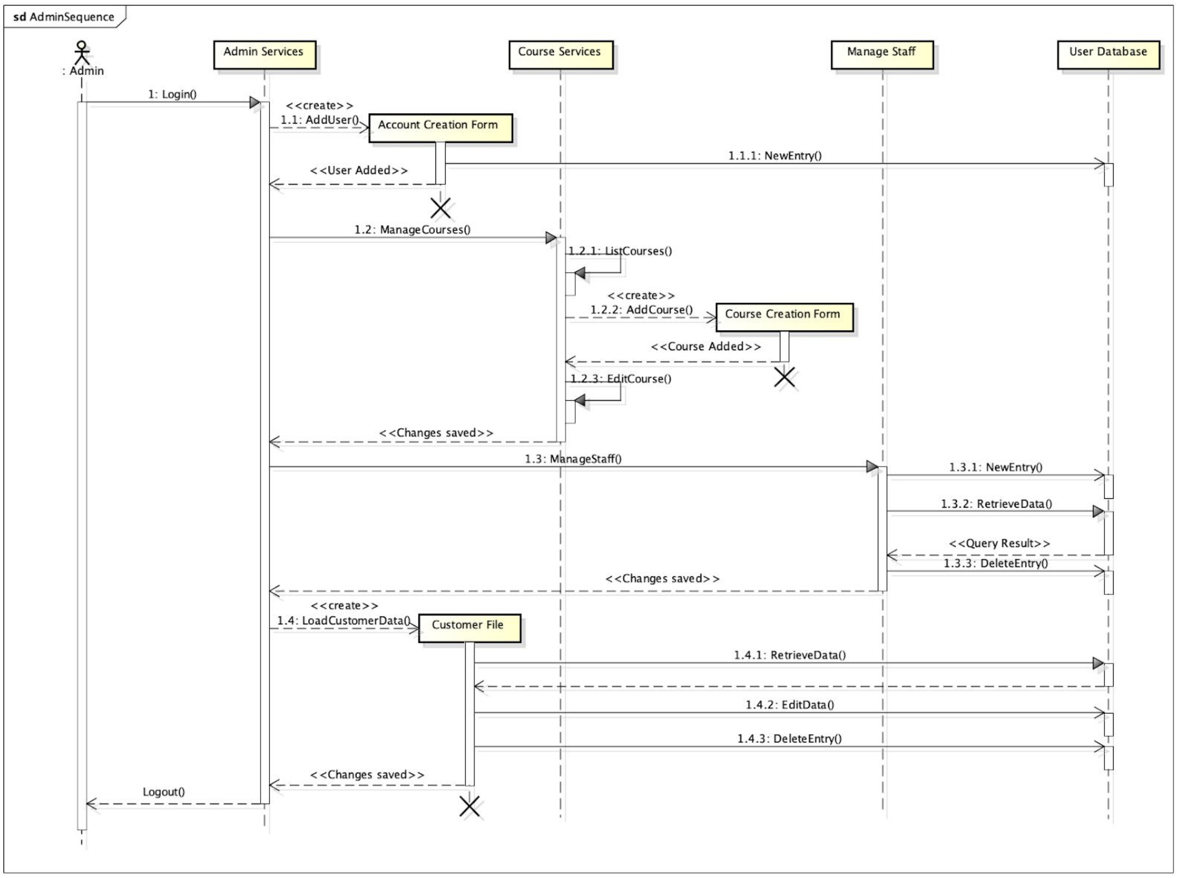

act Act_Login
Application HomePage
Click Login Link
Login Form Displayed
Enter UserID / Password
Submit Form
Invalid ID/Password
Correct Login
Load Home Page

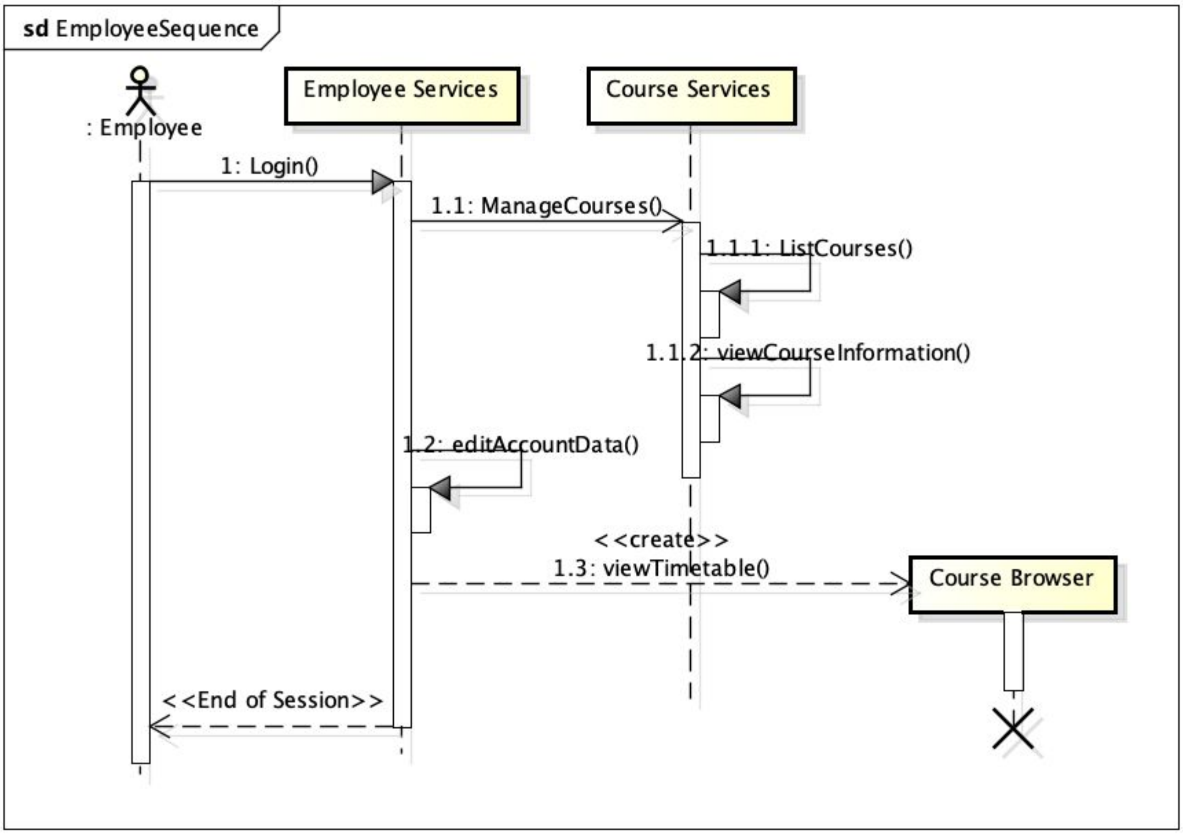


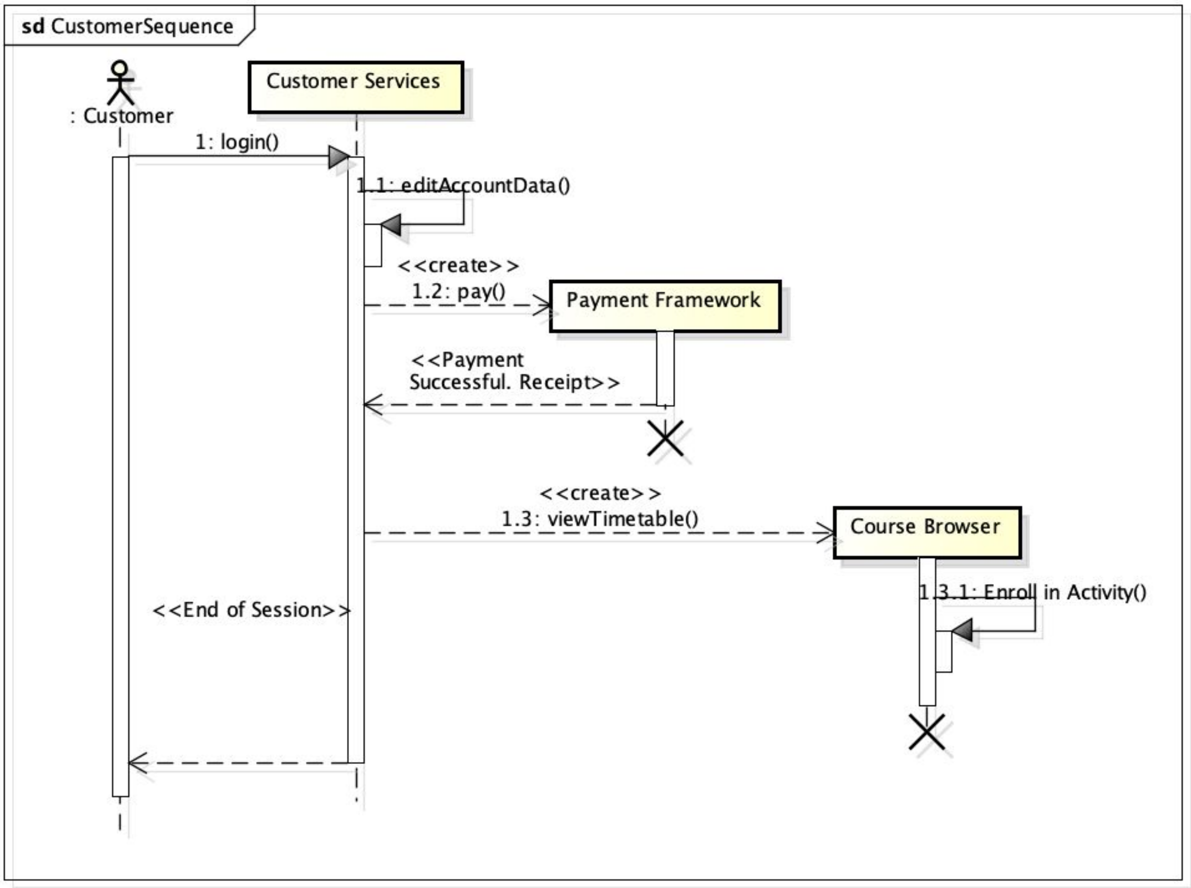
Activity Diagrams for Fitness Club Management System

Sequence Diagrams ---

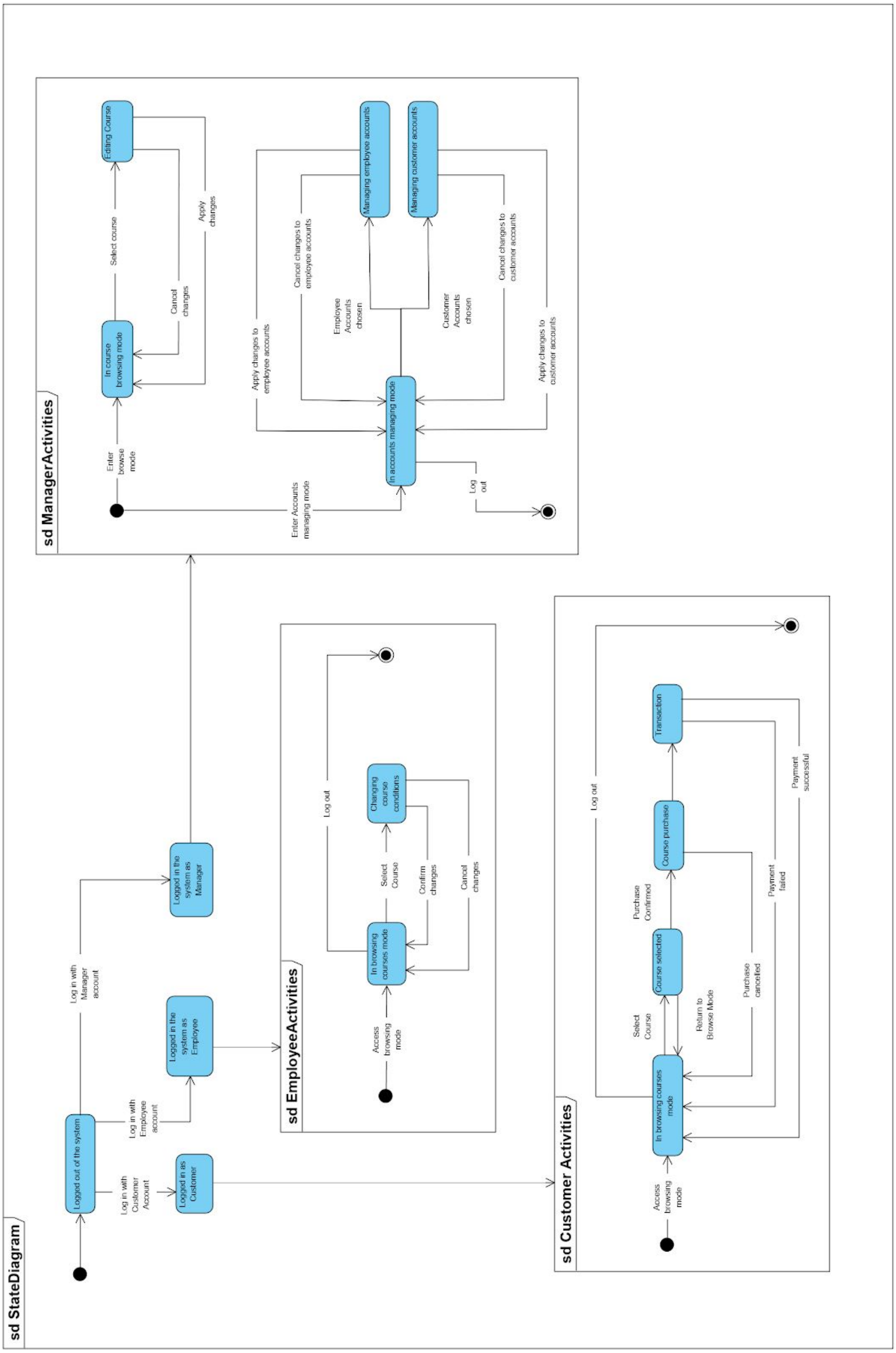



sd AdminSequence
: Admin
Admin Services
Course Services
Manage Staff
User Database
1: Login()
<<create>>
1.1: AddUser()
Account Creation Form
1.1.1: NewEntry()
<<User Added>>
1.2: ManageCourses()
1.2.1: ListCourses()
<<create>>
1.2.2: AddCourse()
Course Creation Form
<<Course Added>>
1.2.3: EditCourse()
<<Changes saved>>
1.3: ManageStaff()
1.3.1: NewEntry()
1.3.2: RetrieveData()
<<Query Result>>
1.3.3: DeleteEntry()
<<Changes saved>>
<<create>>
1.4: LoadCustomerData()
Customer File
1.4.1: RetrieveData()
1.4.2: EditData()
1.4.3: DeleteEntry()
<<Changes saved>>
Logout()


sd EmployeeSequence
: Employee
Employee Services
Course Services
1: Login()
1.1: ManageCourses()
1.1.1: ListCourses()
1.1.2: viewCourseInformation()
1.2: editAccountData()
<<create>>
1.3: viewTimetable()
Course Browser
<<End of Session>>



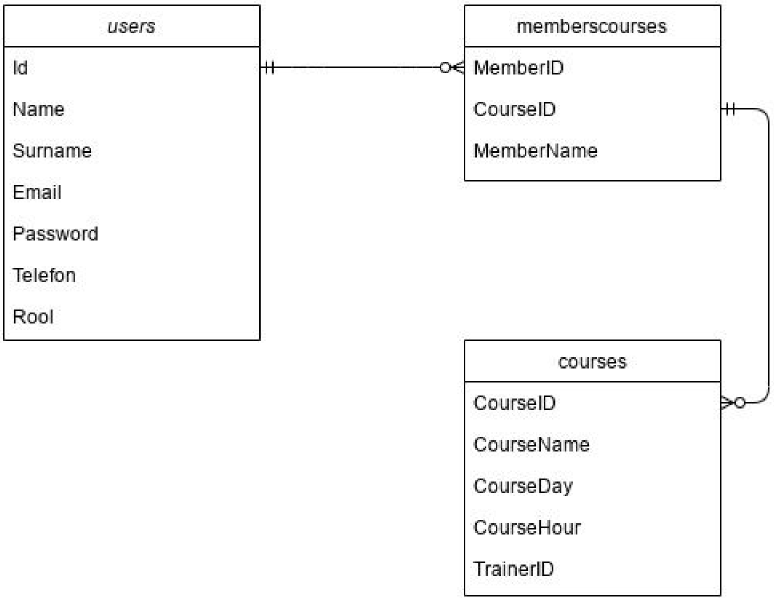
Sequence Diagrams for Fitness Club Management System



State Diagram for Fitness Club Management System

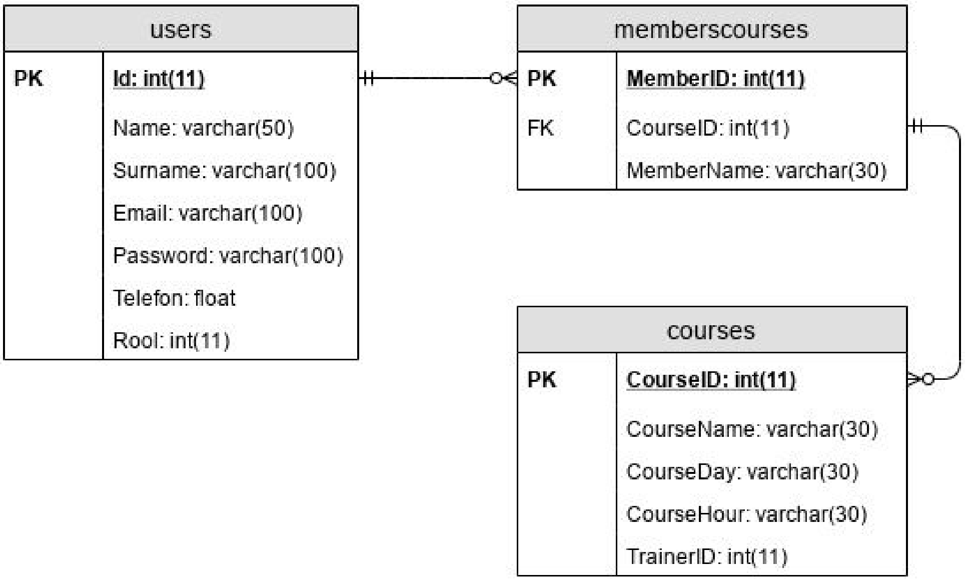
3.2. Database Model

3.2.1. Conceptual Model



Database Model Diagram for Fitness Club Management System

3.2.2. Physical Model

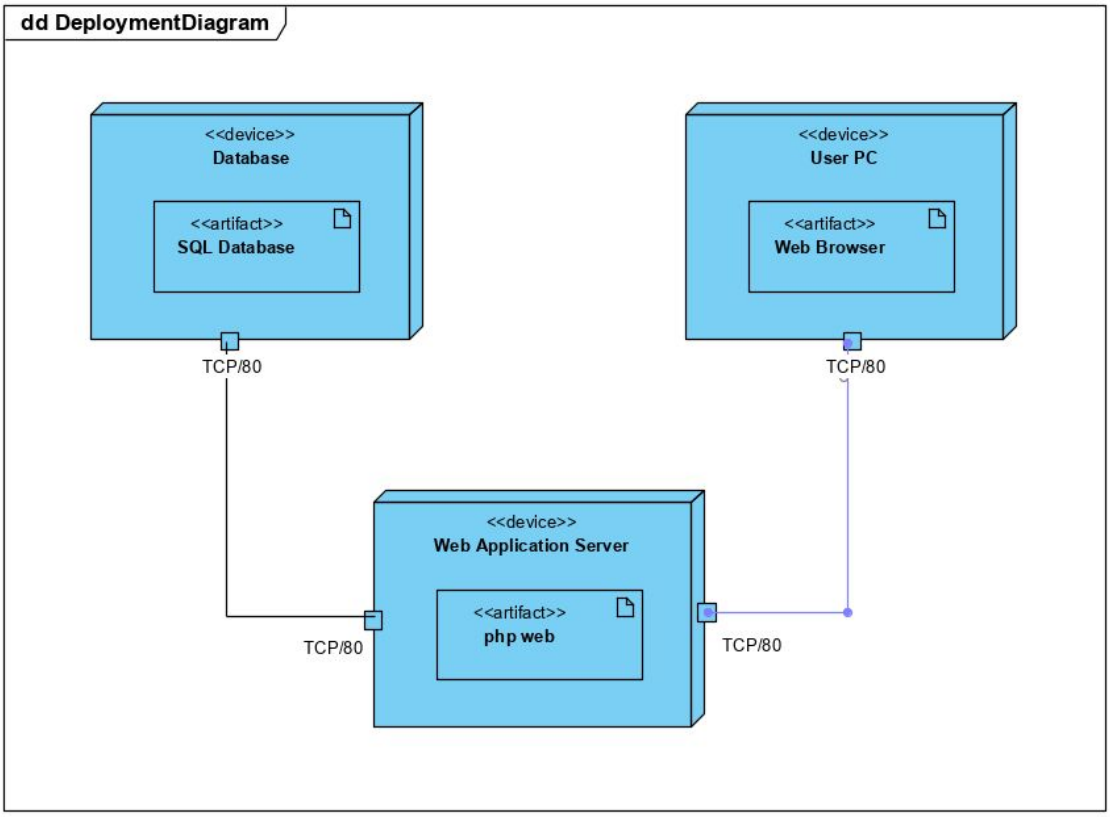


3.3. Software Architecture

Devices:

- Server
- Data Base
- Users Devices

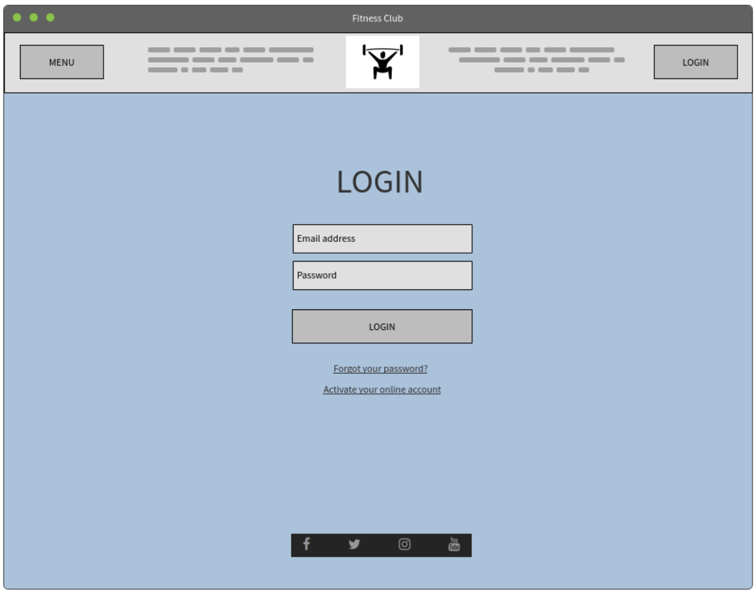
Due to implementation constraints, user device would initially be a computer with a compatible browser. The devices and artifacts are related as it follows in the diagram:



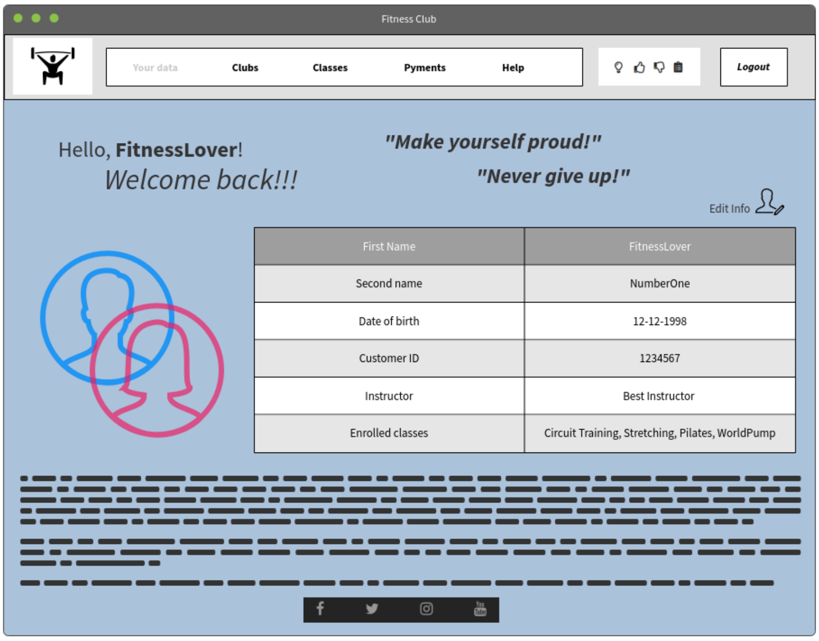
Deployment Diagram for Fitness Club Management System

3.4. User Interface Design

The following images correspond to the first version of the designed UI for the Fitness Club Management System, showing the main disposition of objects, elements and design of the planned product. Attached with the document, an extended version built on AdobeXD (*Fitness Club Management System Prototype.xd)* allows to observe the functionality, behaviour and actions designed for the web page. the PDF document *Fitness Club Management System Protoype.pdf* gathers the views observed in the Adobe XD prototype.



1. ***Login Screen v1.***

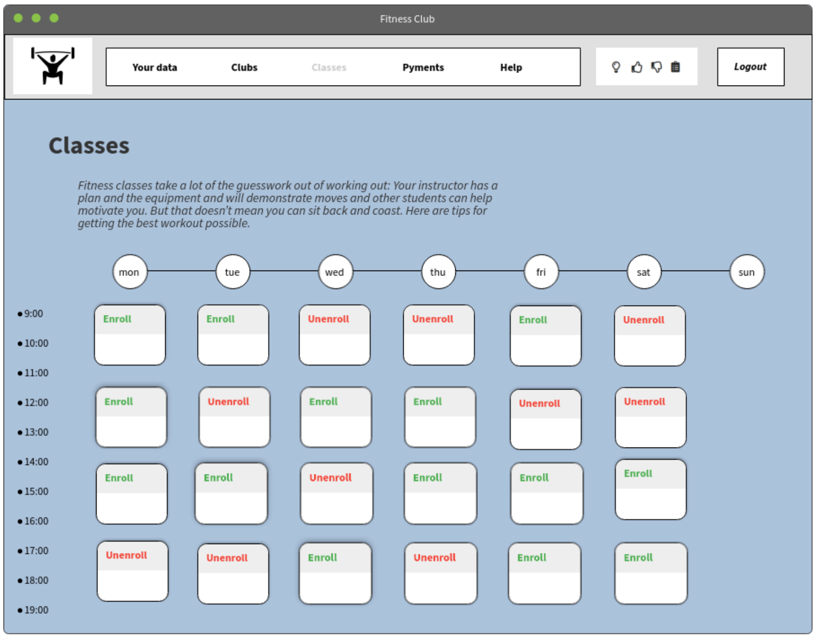


Hello, **FitnessLover**!
Welcome back!!!

"Make yourself proud!"
"Never give up!"

First Name	FitnessLover
Second name	NumberOne
Date of birth	12-12-1998
Customer ID	1234567
Instructor	Best Instructor
Enrolled classes	Circuit Training, Stretching, Pilates, WorldPump

2. ***Home Screen v1.***



Classes

Fitness classes take a lot of the guesswork out of working out: Your instructor has a plan and the equipment and will demonstrate moves and other students can help motivate you. But that doesn't mean you can sit back and coast. Here are tips for getting the best workout possible.

mon	tue	wed	thu	fri	sat	sun
Enroll	Enroll	Unenroll	Unenroll	Enroll	Unenroll	
Enroll	Unenroll	Enroll	Enroll	Unenroll	Unenroll	
Enroll	Enroll	Unenroll	Enroll	Enroll	Enroll	
Unenroll	Unenroll	Enroll	Unenroll	Enroll	Enroll	

- 9:00
- 10:00
- 11:00
- 12:00
- 13:00
- 14:00
- 15:00
- 16:00
- 17:00
- 18:00
- 19:00

3. Course Browsing Screen

4. Construction and tests

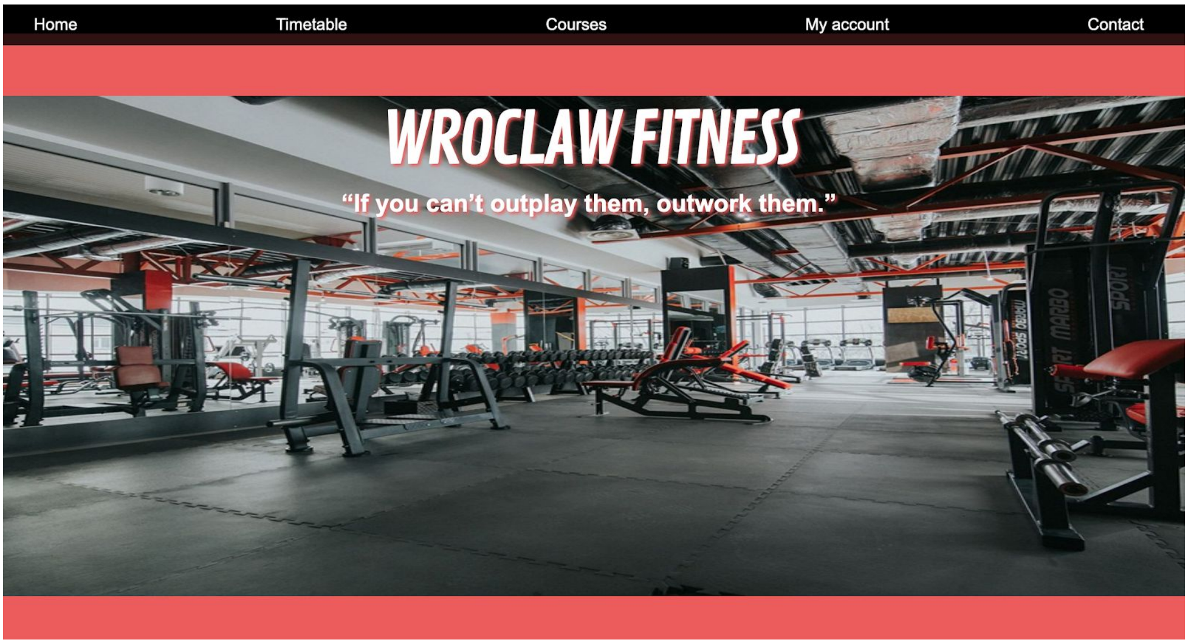
4.1. Implementation

The project was developed using PHP technology applied to an SQL database implementing the View-Controller Model. Additionally, for the web page design and basic behaviour HTML and CSS have been used. For the correct running of the product, a web-server able to hadle SQL databases is essentially needed.

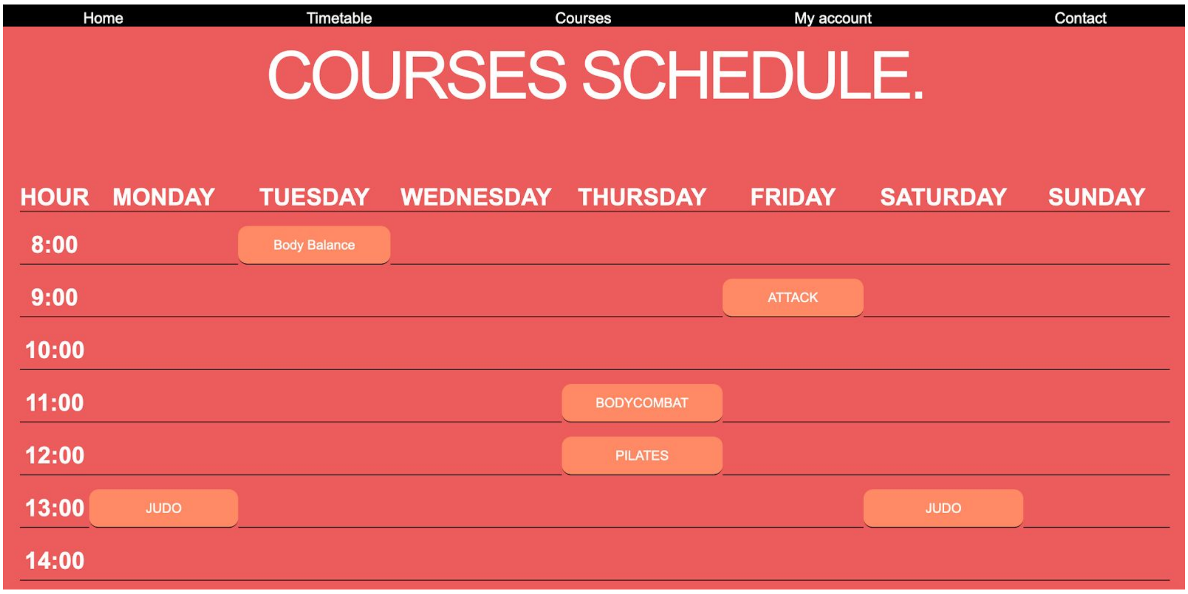
CONTROLLERS: Control Views development and behaviour. Stablish relation with BD.php and thus provides imformation to Views.	AddCourseController.php: Controller for view AddCourse.php
	AddCourseSubmitController.php: Controller for providing database with the new course information.
	addUserController.php: Controller for view AddUser.php
	ControllerContact.php: Controller for view Contact.php.
	CorrectLoginController.php: Controller for view IncorrectValues.php, stablishes new session and its values.
	CoursesController.php: Controller for view Courses.php
	DeleteCourseController.php: Controller for passing information to BD.php passing the entry to be removed from Coursestable.
	deleteUserController.php: Controller for passing information to BD.php passing the entry to be removed from Users table.
	drawTimeTableController.php: Controller for DrawTimetable.php, with the information obtained from database.
	EnrollController.php: Provides BD.php function with arguments for creating a new entry in membersCourses table.
	HomeController.php: Controller for view Home.php
	LoginController.php: Controller for view Login.php
	LogOutController.php: Controller for redirecting after Logging Out.
	manageUsersController.php: Controller for view manageUsers.php
	newUserFormSubmitController.php: Controller for passing information in creating new entry in users table.
	RegisterController.php: Controller for view Register.php
	RegisterSubmitController.php: Asks BD.php to create a new entry in users table.

	ShowListController.php: gets enrollment list and transfers it to view ShowList.php
	showUserResultsController.php: gets members informationn from the search terms entered and communicates the data to showUserLookup.php
VIEWS: Responsible for showing the pertinent information to the user, depending on the information sent by their respective controller. Webpage visual design is implemented within these files.	AddCourse.php: displays add course form.
	addUser.php: displays add user form.
	AlreadyLogIn.php: called when user is already logged-in and press sign-in.
	alreadyRegistered.php: called when user enters data of account already existing.
	Contact.php: displays contact information in contact page.
	header.php: displays header with Fitness Club banner.
	Home.php: manages the elements in the homepage of the site.
	IncorrectValues.php: called when user signs-up with incorrect email-password pair.
	Login.php: displays user log-in form.
	LogOut.php: displays logged-out message.
	manageUsers.php: displays page for checking users list and add them.
	MenuOption.php: displays menu on top of the webpage depending on the session type and settings.
	Register.php: displays registration form.
	Registered.php: showed when the registration of a new user is successful.
	ShowList.php: displays enrollment list.
	showUserLookup.php: displays user search results.
	ValidationDenied.php: displayed when login errors occur.
DB.php - Gathers the methods used for accessing and consulting the database, add and remove entries. Supports controllers providing data.	
index.php - Main directory of the webpage, from where the different controllers and views are launched.	

CSS Styles	estilos.css - supports WebPage design with styles developed specifically for the product.
Pictures	favicon files.
	fondo_head.jpg / fondo_head2.jpg / gym.png.
Gym.sql: database layout representing the database used in the product to be imported. Consists of three table that represent the relationships needed: - users: gathers every user registered in the system, as well as their data and role code. (0 = customer; 1= trainer; 2 = admin) - courses: made out of the added courses available for the clients of the Fitness Club. - membersCourses: establishes a relationship between a course and the users enrolled in it.	



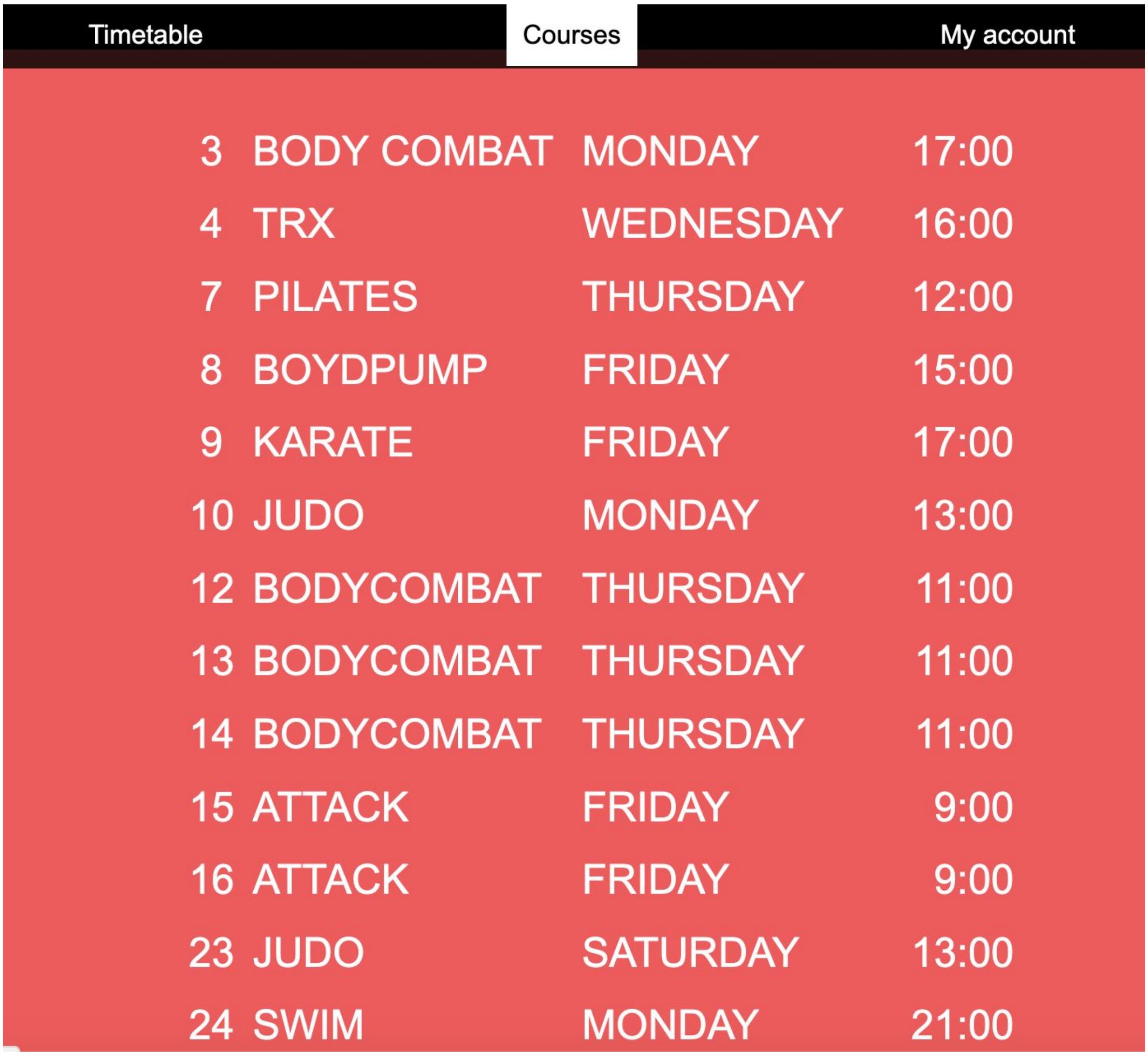
1. Home Screen of Fitness Club Management System.



COURSES SCHEDULE.

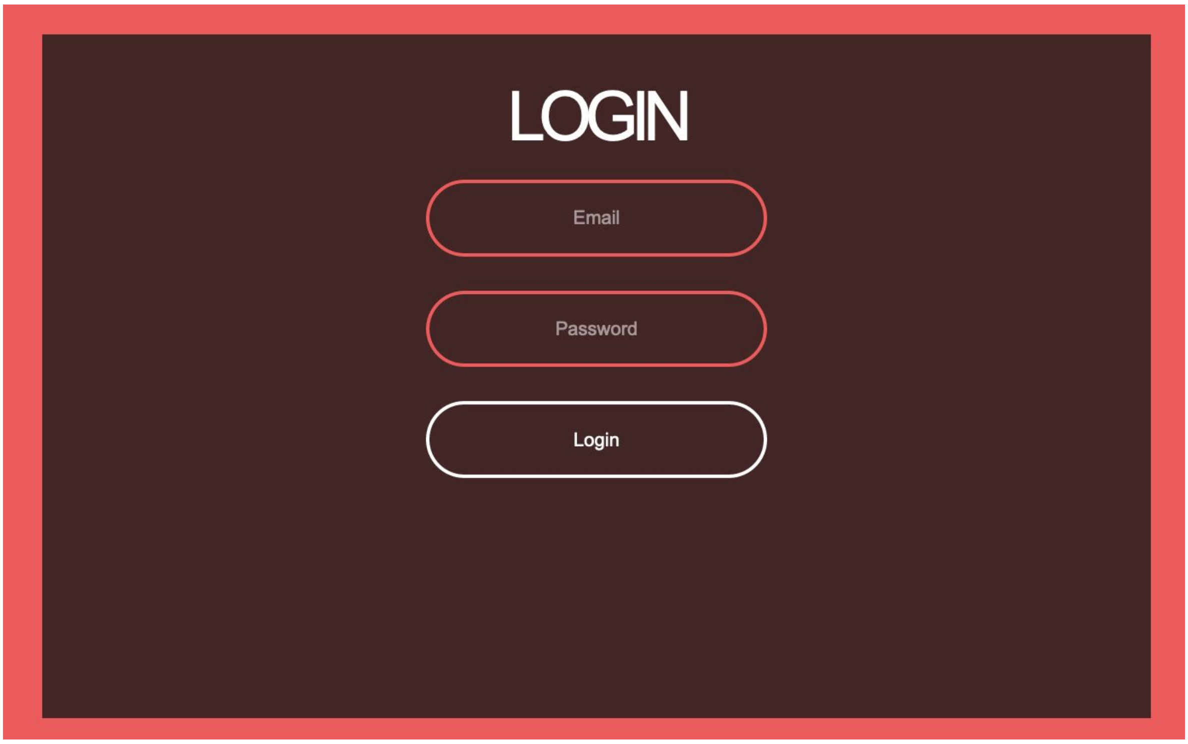
HOUR	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY
8:00		Body Balance					
9:00					ATTACK		
10:00							
11:00				BODYCOMBAT			
12:00				PILATES			
13:00	JUDO					JUDO	
14:00							

2. Courses Schedule Page of Fitness Club Management System.



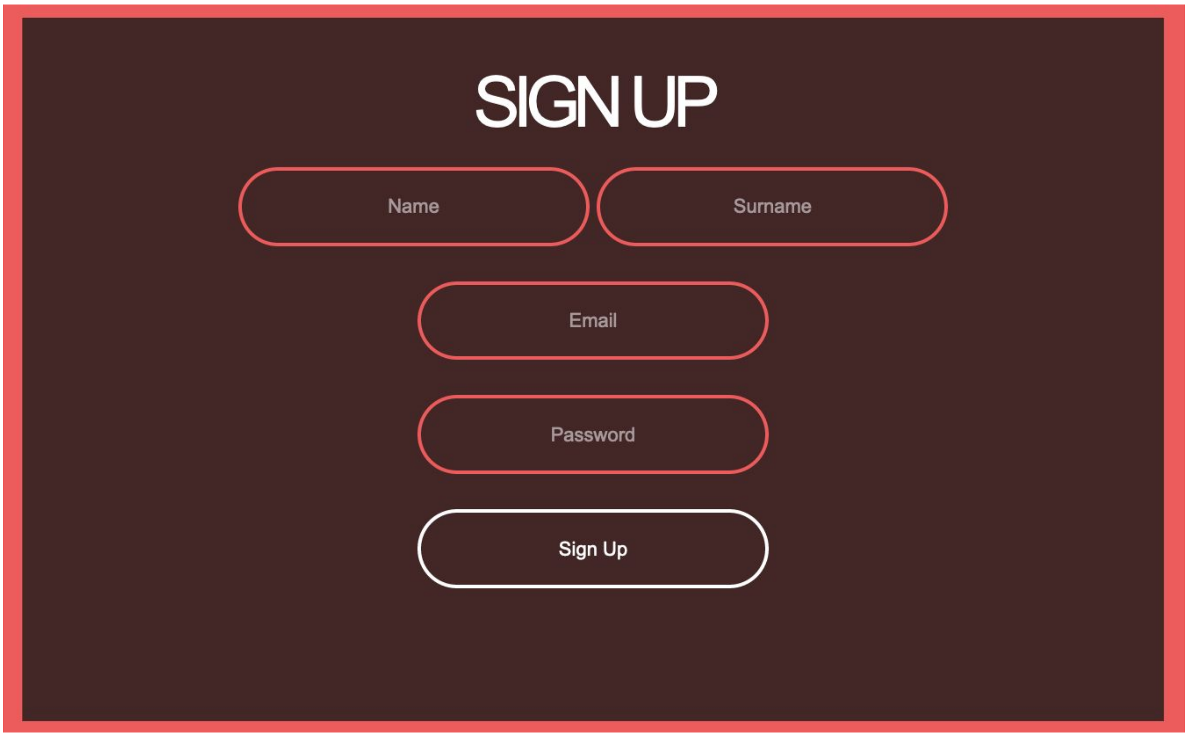
3	BODY COMBAT	MONDAY	17:00
4	TRX	WEDNESDAY	16:00
7	PILATES	THURSDAY	12:00
8	BOYDPUMP	FRIDAY	15:00
9	KARATE	FRIDAY	17:00
10	JUDO	MONDAY	13:00
12	BODYCOMBAT	THURSDAY	11:00
13	BODYCOMBAT	THURSDAY	11:00
14	BODYCOMBAT	THURSDAY	11:00
15	ATTACK	FRIDAY	9:00
16	ATTACK	FRIDAY	9:00
23	JUDO	SATURDAY	13:00
24	SWIM	MONDAY	21:00

3. List of available courses.



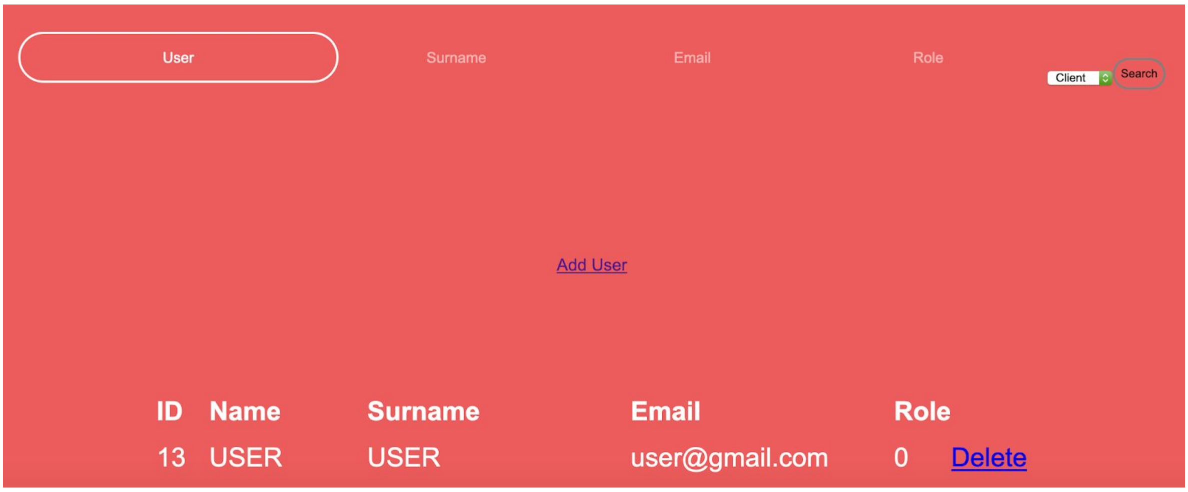
LOGIN

4. Login form of Fitness Club Management System.



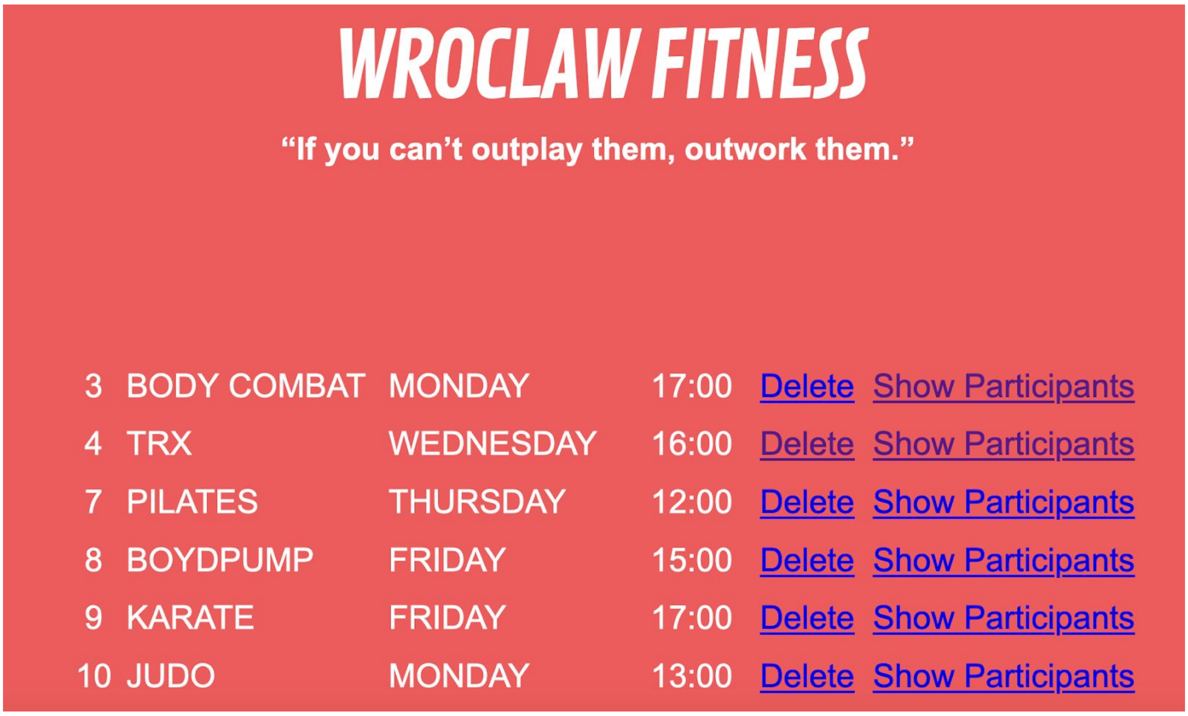
SIGN UP

5. Registration form of Fitness Club Management System.



ID	Name	Surname	Email	Role	
13	USER	USER	user@gmail.com	0	Delete

6. User searcher and results with Deletion option.



3	BODY COMBAT	MONDAY	17:00	Delete	Show Participants
4	TRX	WEDNESDAY	16:00	Delete	Show Participants
7	PILATES	THURSDAY	12:00	Delete	Show Participants
8	BOYDPUMP	FRIDAY	15:00	Delete	Show Participants
9	KARATE	FRIDAY	17:00	Delete	Show Participants
10	JUDO	MONDAY	13:00	Delete	Show Participants

7. Courses management view of Fitness Club Management System

4.2. Tests

4.2.1. Tests' plan

In order to test the php project the PHPUnit testing framework will be used. This program allows us to easily work with Composer, an application-level package manager.

All the functionalities included in the software are related and implies an interaction with the database at certain stages, either in the sense of gathering information under certain parameters, either with the creation of new registries and providing a full usage to the user. Hence, the testing centered on the model, that is, the database object and its functionalities. As it can be read in the framework documentation, the testing checks if the defined functions work properly with the queries to the DataBase. Nonetheless, the configuration of the testing framework for it to work along the database setup was not possible due the impossibility of connecting the database manager with the testing code.

Additionally, with the prospects of not being able to perform an automatic testing, a manually-performed testing will be carried out, in order to test the implemented functionalities.

4.2.2. Report from unit testing for selected implemented functionality

Due to the impossibility of running a standardised test set applied to out code, the following test cases have been designed to check the expected functionalities while detecting the existence of errors and mistaken implementations.

Test Scenario	Test Case	Test Steps	Test Data	Expected Results	Observed Results	Pass/Fail
Check Sign up Functionality	Check if sign up function works	1.Open web-page and click sign up button	User name User email User password	Sign up must be successful	/Successfully signed up/	**Pass**
Check Login Functionality	Check response on entering valid username/email and passwords	1.Open web-page 2.Enter username/mal 3.Enter Password 4.Click Sign in button	User name: *admin* Password: *admin*	Login must be successful	/Login Successful/	**Pass**
Check Browse courses Functionality	Check response if courses will Show up after clicking on courses button	1.Click Courses button on main page if logged in, if not Log in then Click courses		Course selection page must Show up		**Pass**
Check Enroll Functionality	Check if enroll button works	1.Choose a course and click enroll button		Enroll must be successful	/Successfully enrolled/	**Pass**

Check Unenroll Functionality	Check if Unenroll button works	1.Click unenroll button on chosen course		Unenrolling must be successful	/Successfully Unenrolled /	**Pass**
Manage account Functionality For Admin	Check Account management (Delete member-add member-update member details-update employee data-add/remove employee) works	1.Create new membership for customer 2.Update/delete Membership for customer 3.Add/update Employee data		Account management must success	/Successfully created/ /Update Successful/ /New employee added successfully	**Pass**
Logout Functionality	Check logout button works	1.Click my account 2.Click log out		Logo ut must success	/Log out successful/	**Pass**

4.2.3. Report from acceptance testing

After performing a functionality testing and checking the errors that the application could raise, the code has been corrected in order to avoid those. Among the errors and recommendations, we must specify:

- Web page allows for the same email registration, breaking the uniqueness of the client account. A non pre-existence condition has been implemented in order to avoid double users.
- Session variables are not detected in certain environments, due to php server versions and character coding. A pre-setup must be performed in order for the code to work in the server computer. Nonetheless, users accessing the web page should not face any problems whatsoever.
- Payment requisite has not been achieved due to the complexity of the implementation with a third-party application that achieves the security requirements established previously.

Appendix

General rules:

1. Language of the document: English.
2. Final evaluation: printed and electronic in docx or odt and PDF (link send in email).
3. The deadlines for each step according to the schedule.
4. Failure to provide report containing specified in the schedule part within the specified period results in 0% for the evaluation of a given part.
5. Attendance at the classes is obligatory.
6. PM is obliged to report the lack of activity (implementation of tasks) of his team members.
7. It is not possible to issue an evaluation based on a completed, brought-in project.
8. The dates of the presentation are rigorous and it is not possible to present after the deadline.
9. After being assigned to a project group, it is not possible to change it.
10. The project is evaluated in terms of correctness, consistency and completeness. evaluation of the project is a percentage and same for the entire project group with respect to point 7.
11. PM has an additional pool of 30 percentage points. They can be disposed at PM discretion with a maximum limit of 10% per person. These points increase the individual rating of individual group members.
12. PM is assessed separately only by the teacher and this assessment is influenced both by the quality of the project as a whole, as well as by the way project has been managed in all its aspects.

Classes schedule:

Class	Wednesday class	Thursday class	Class Description
1	2.10.2019	3.10.2019	**Presentation of the project and rules for passing. Division into groups. Designation of PM. General division of thematic areas.**
2	9.10.2019	10.10.2019	**Work on Elaboration of application concept (F1)**
3	**16.10.2019**	**17.10.2019**	**Presentation of F1 results. Evaluating of the F1 presentation and collection of documentation.**
4	23.10.2019	24.10.2019	**Work on Requirements specification (F2)**
5	30.10.2019	7.11.2019	
6	**6.11.2019**	**14.11.2019**	**Presentation of F2 results. Evaluating of the F2 presentation and collection of documentation.**
7	20.11.2019	21.11.2019	**Work on Design (F3)**
8	27.11.2019	28.11.2019	
9	4.12.2019	5.12.2019	
10	**11.12.2019**	**12.12.2019**	**Presentation of F3 results. Evaluating of the F3 presentation and collection of documentation.**
11	18.12.2019	19.12.2019	**Work on Construction and tests (F4)**
12	8.01.2020	9.01.2020	
13	15.01.2020	16.01.2020	
14	**22.01.2020**	**23.01.2020**	**Presentation of F4 results. Evaluating of the F4 presentation and collection of documentation.**
15	29.01.2020	30.01.2020	**Final evaluation**