

# **TECHNICAL DOCUMENTATION**

## **1. USER REQUIREMENTS SPECIFICATION**

### **1.1. Introduction**

This application is made for gym users to make it easier to keep records. The world is fully digitalized, so it would be time to digitize the running of the gym as well. The aim of this application is to facilitate the monitoring of obligations of gym users, employees and the owner.

### **1.2. Software system description**

The software system that we will implement will be independent and will have a centralized database. The data that will be in the database are basic information about users, employees, trainings and membership fees.

- Information on users and employees will be name, surname, OIB, e-mail address and telephone number and type of training (selected by the user / conducted by the employee).
- Training data will be the type of training, the employee conducting the specific training, the training schedule and occupancy.
- Membership information will be the date of issue of the invoice and (un) paid.

This application requires an internet connection and a computer.

### **1.3. Application users**

Users of this application are:

- Administrator - performs all tasks related to the database (can add, delete, update and change data) or users, employees, training and membership fees.

- Employee (trainer) - the employee of the application must be registered and thus gets the opportunity to view and update their profile. He also has insight into the data of members who have registered for their training, I can send them an e-mail and see the schedule for their trainings.
- The user - the user of the application, as well as the employee (trainer) must be registered and thus gets the opportunity to view and update their profile. In addition, the user can see the types of training that take place in the gym, enroll in one of the trainings, report to the employee (trainer) and follow your training schedule. Also, the user receives an e-mail bill every month to pay the membership fee.

#### 1.4. Functional requirements

No	Feature	Description
F01	Credentials	To begin with, the user must register in order to be able to use the application or log in in the future.
F01.A	Sign Up	User needs to create account for him to be able to use the app
F01.B	Login	Before he is able to use the app, user needs to login
F01.C	Password reset	User can request password reset
F02	Membership fee	User pays gym membership fee every month
F03	Generate + send PDF of invoice	The user receives the PDF via email about the paid invoice
F04	Training Managment	User can choose between training types and add them to the schedule
F04.A	Apply for training	User chooses training type group training or individual (with coach)
F04.B	Pay chosen training plan	After the user has selected his training plan, a bill is sent to him to be payed
F05	Training schedule	All actions regarding training schedule

F05.A	Choose timetable	User chooses appointment of training
F05.B	Notifications	The coach gets e-mail notification when client is trying to hire him
F06	Reminders	User gets reminders for his appointments, reminders are sent via e-mail.
F07	Coach	Coach decides if he accepts or rejects User/Client
F08	Coach Profiles	Coaches can fill in their info (i.e. education) which will be visible to the clients who are planing to hire him
F09	User profiles - Superadmin	Superadmin is able to create multiple roles for the app. First role would be admin, he is a gym owner and he will have regular admin privileges (such as, being able to track statistics, cann create coaches, can approve new users/clients and whatever he need for his gym to function). Second role is coach. Coach can edit his profile, training sessions etc. Final role is regular User/Client and his role is to apply into the app, pay membership fee and, if user/client wants to, take training type
F10	Statistics	Gym owner and admin has review of all relevant statistics for the gym (users, trainigs, presence ect.)
F11	PDF bill generator	App generates PDF bill for gym that uses our app

## 1.5. Non-functional requirements

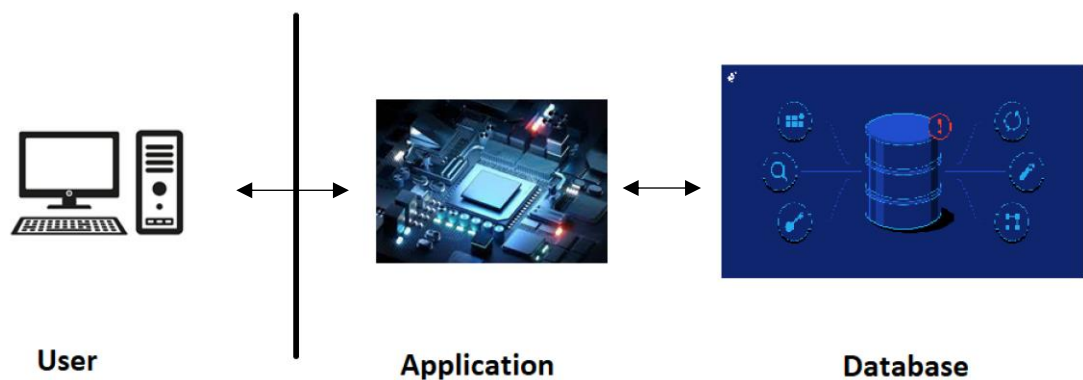
Non-functional requirements are the limitations, standards and quality of the system as a whole. To improve the user experience, the application must have the following:

- High level of security - the application includes user registration, which means that private user data is included. What is very important is that the application and the server have a high level of security to protect user privacy.
- Clear interface design - the interface must be clear and concise so that the user can easily access any functionality offered by the application.
- availability - the application must offer all the possibilities at all times. Because the application is made on Windows, it must be available on all Windows platforms.

- Appropriate performance - the response and speed of user-interface interaction must be fast
- scalability - in the case of a large number of users, the application must continue to work with acceptable performance.

## 2. SOFTWARE SYSTEM DESIGN

When designing a software system, the architecture of the system, which is very simple, is very important. The user, with the help of a graphical interface, interacts with the application, which is actually an intermediary between the user and the database. An overview of the system architecture of this application is shown in the figure.



### 2.1. Application users

The users of this application are the user, trainer and administrator.

#### 2.1.1. Administrator

As in every application, so in this one, the administrator has the most important role and the most possibilities. Here are the following use cases that the administrator has:

- Assigning a role to users - the administrator has the ability to add roles to all registered users. The roles that can be assigned to them are the user, employee or coach or administrator.
- Data access - in this application only the administrator has access to all data from all users of the application.
- Add / Delete / Edit - The administrator can add new users, delete users or delete specific user data as well as edit information for each user.
- Statistics overview - in this case the administrator is the only one who has the ability to view statistics. Statistics include data on the types of training, ie whether the user trains alone, whether the user is part of group or individual training. The statistics also show data on how many gyms there are users and at the same time what the overall presence of users in the gym is. Also, training attendance is another item that belongs to the group of statistics.

### **2.1.2. Coach**

In this application the trainer has the following use cases:

- Editing the profile - the trainer can edit his profile, ie his personal data even after registration in the application.
- Acceptance / rejection of the client - when the user applies for the type of training led by the trainer, the trainer receives a notification that the user has sent a request to enroll in his training. The trainer can accept or reject this client request.
- Training schedule - the trainer can use the application to see the schedule of their trainings, on which day they are held, at what time and how many users are registered for group training, or which user is registered for individual training.
- Editing trainings - in this case the trainer can change the schedule of his trainings and transfer users from one group to another.

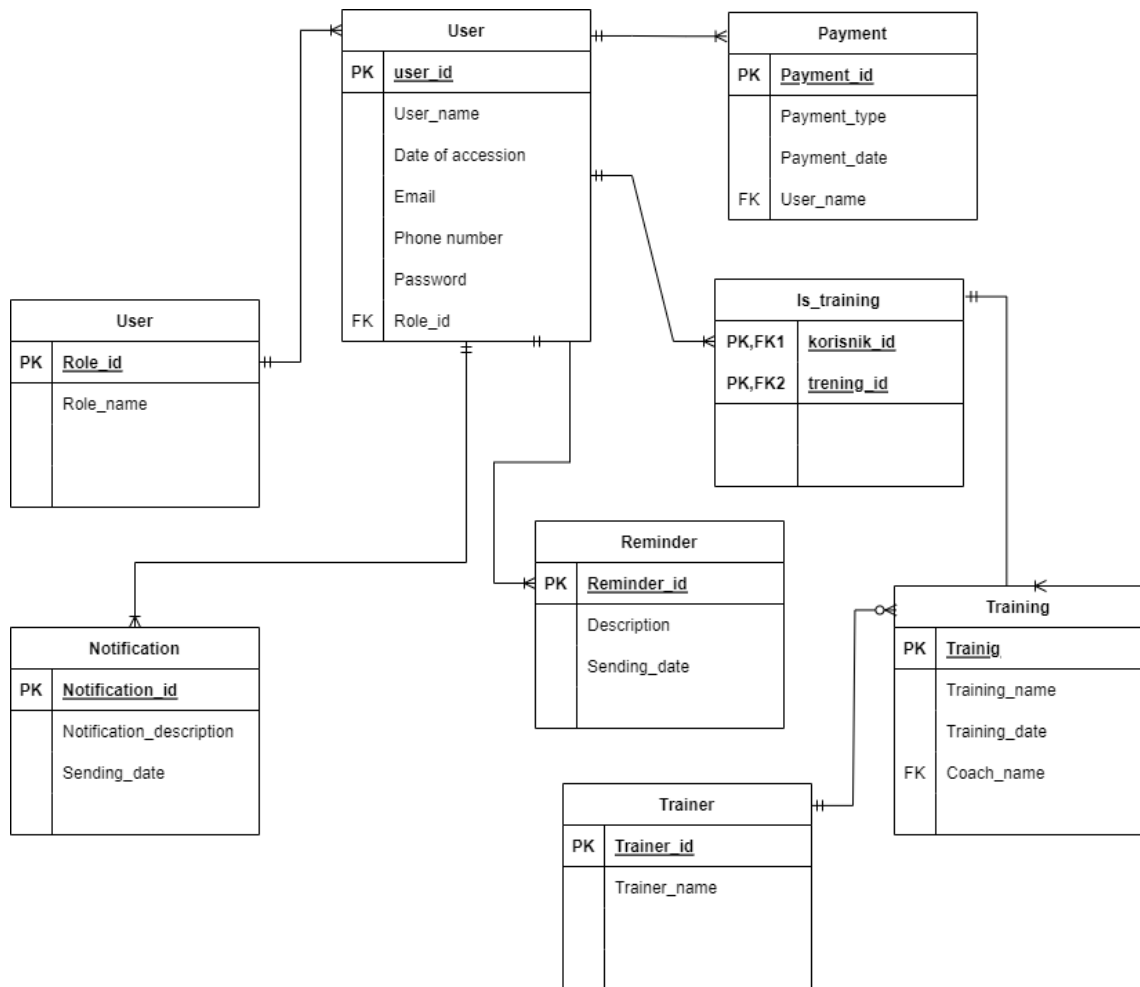
### **2.1.3. User**

The user has the least role in this application. The cases used by the user are as follows:

- Payment of membership fee - when the user registers in the application, that registration is considered as registration in the gym. Immediately after registration, the user receives a PDF invoice for the membership fee, which he can pay via the application.
- Choose the type of training - the user can choose between 3 types of training. The first type is to train alone, the second type is group training, and the third type is individual training. If the user chooses the second or third type of training, another PDF invoice comes to his e-mail, which contains information on the payment of that type of training.
- Choose a training schedule - if the user enrolls in a group or individual training, he can choose the training schedule that suits him.
- Choose a trainer - the user has the option of choosing a trainer for individual or group training. He sends a request to the coach, however, in order for the user to be with the coach he has chosen, that coach must approve the request to the user.

## 2.2. ERA model

The following figure shows the ERA model. The tables shown in the model are sufficient to perform all the functionalities in the application.



We will explain the ERA diagram of our project as follows. The user can make several membership fee payments, while one membership fee payment can be made by only one user. One role can have several users, while one user can be a member of only one group (administrator, trainer, client). One notification sent to a user can only be sent to one user, while one user can receive multiple notifications. The same goes for reminders for upcoming workouts that are emailed. One reminder can only be sent to one user, while one user can potentially receive multiple reminders. It is true for a coach to be able to lead multiple trainings, while one training session can be led by only one coach. On the other hand, when we look at the relationship between users and training, we can say that one training session can be attended by several gym users, while also one gym participant can attend more training sessions.

## 2.3. Description of functionality

### 2.3.1. Credentials

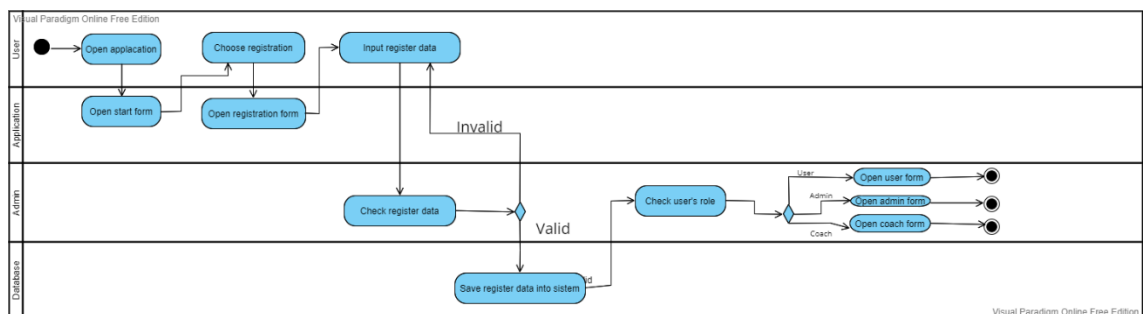
#### Description

Registration and login functionalities can be used by all users of the application. When logging into the application by checking the data, the current user will be shown further options depending on the role he holds.

#### The script

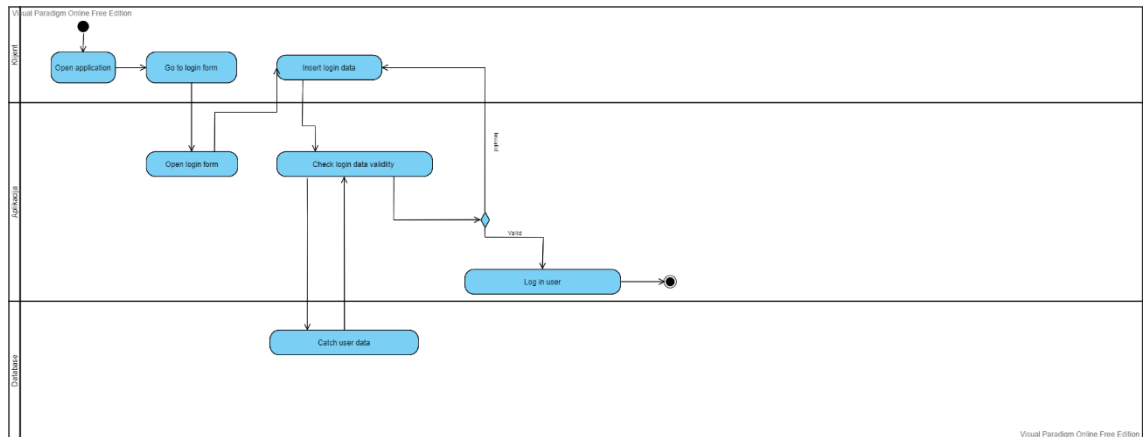
1. The user accesses the main menu
2. When using the application for the first time, select the registration option
3. The system returns the registration form to the user
4. The user registers and his data is checked
5. After successful registration, the system returns the login form to the user
6. The user enters his data via the application form
7. After checking the data, the user is shown the home screen of the application depending on the role he holds

#### Activity diagram

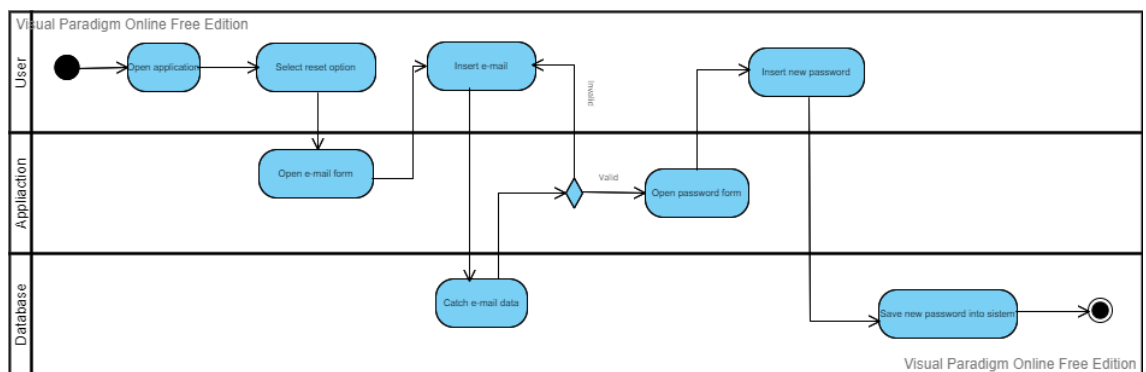


#### Registration



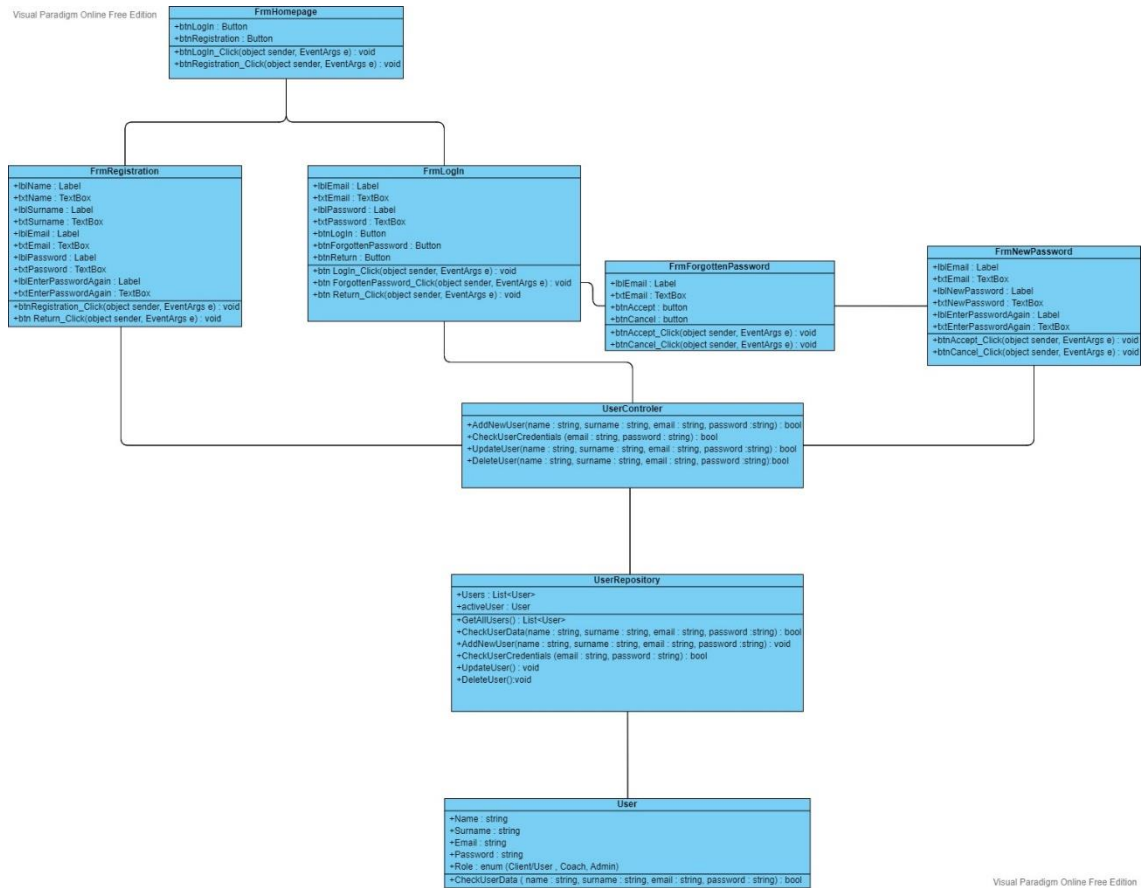


## Log in



## Password reset

## Class diagram



## 2.3.2. Membership fee and Generate + send PDF of invoice

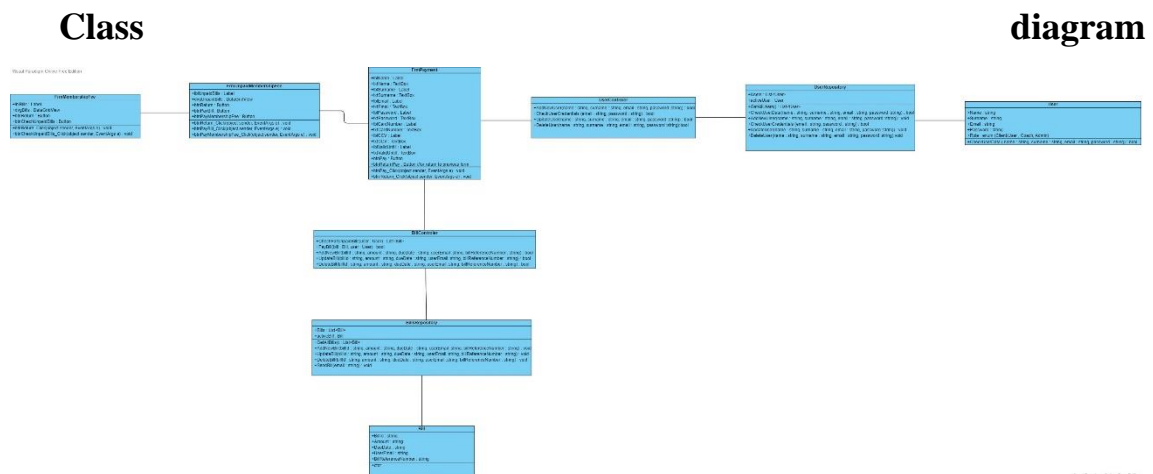
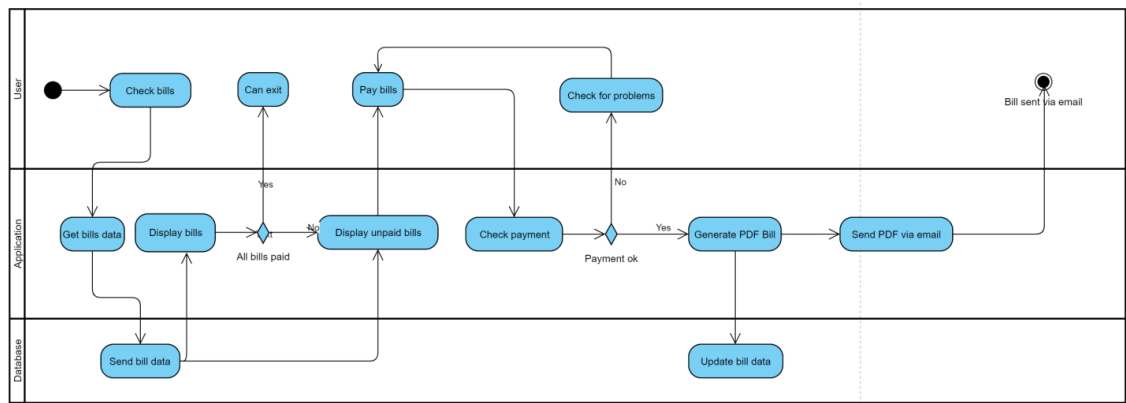
### Description

The functionality of membership fees and sending PDF invoices are linked to each other. These functionalities are accessed by the user.

### The script

1. By registering, the user becomes a member of the gym.
2. The system sends the membership fee invoice to the user's e-mail.
3. The user logs in to the system.

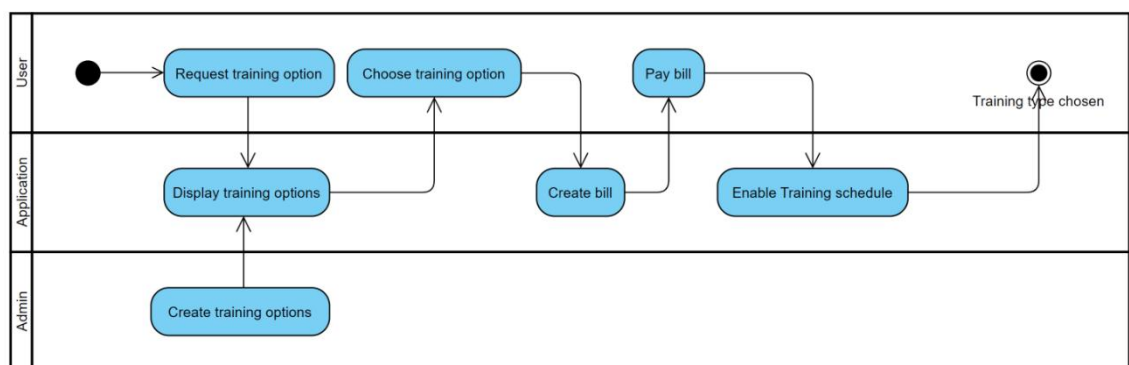
### Activity diagram



1. The user with the help of the application reviews the types of training that are in the system.
2. Enroll for the desired type of training.

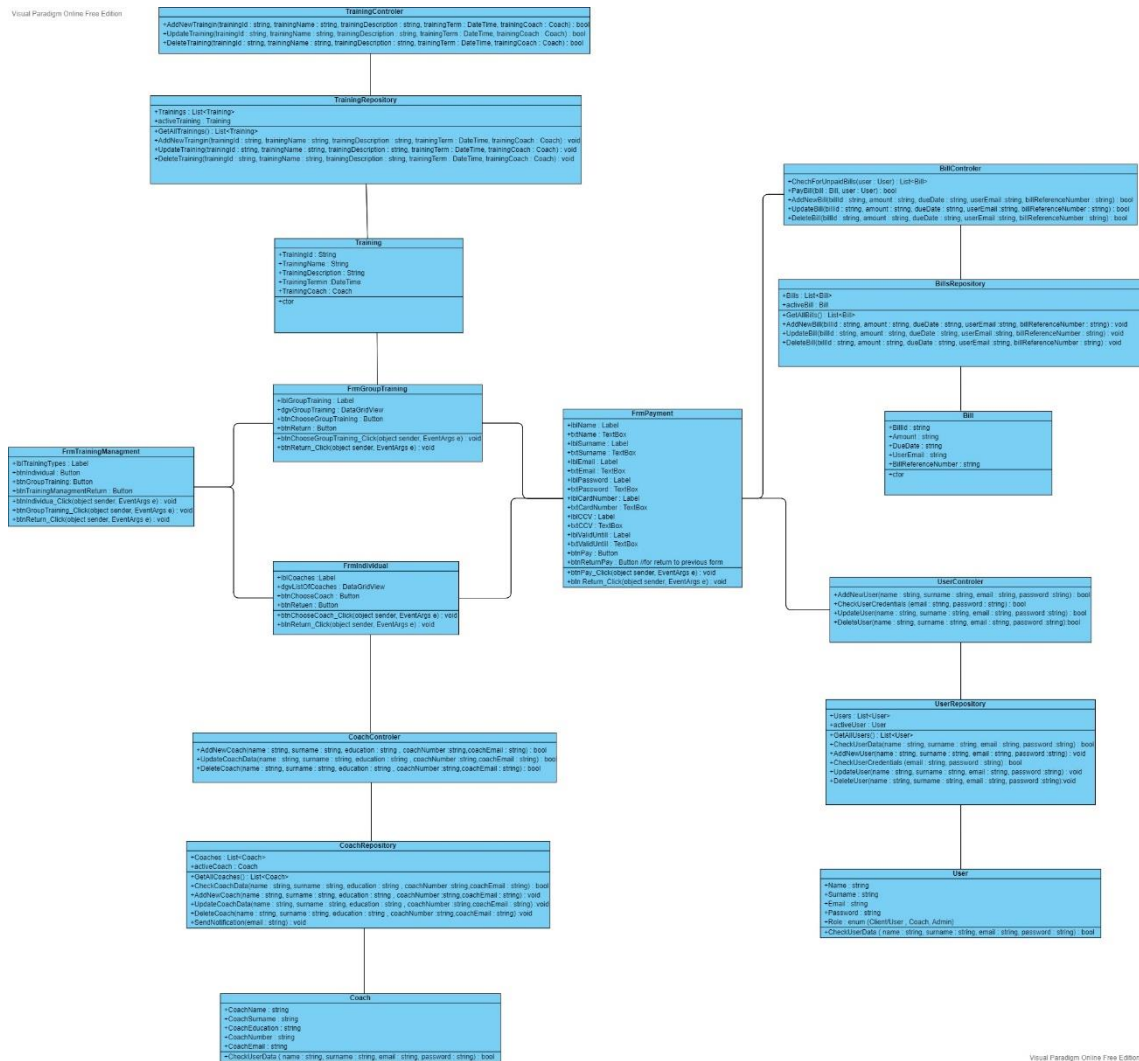
3. The system sends the user a PDF invoice for this type of training to the e-mail address.
4. The user pays the bill in the application.
5. The user is added to the system when he pays the bill.

### Activity diagram



# Class

# diagram



## 2.3.4. Training schedule

## Description

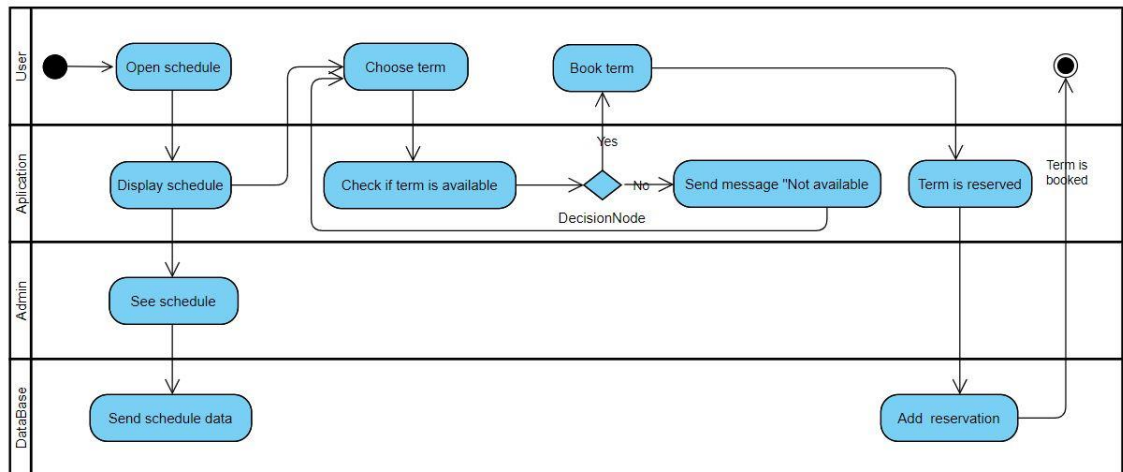
The user can choose the schedule of his trainings so that he can plan in advance when he will go to the training and coordinate the trainings with other obligations.

## The script

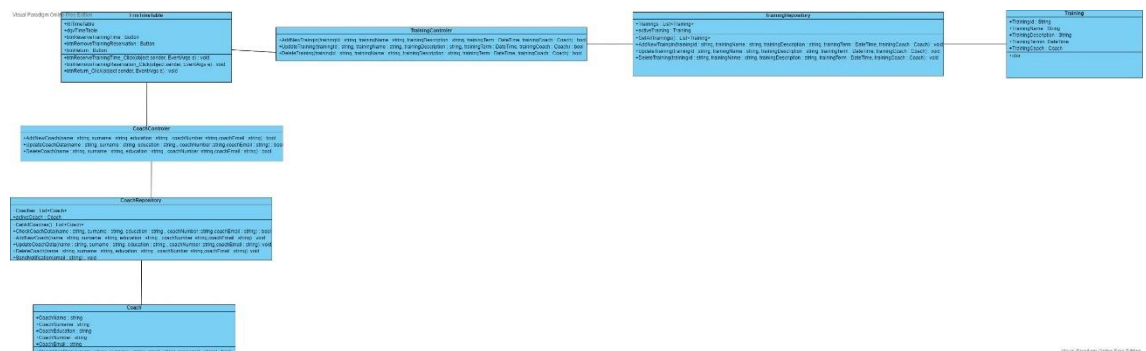
1. The user opens the training schedule from the server via the application.

2. Browse free appointments with the help of the application.
3. It is entered in the desired terms with the help of the application.
4. The application sends information about reserved dates to the server.
5. The server sends a notification to the trainer with updated schedule information.

## Activity diagram



## Class diagram



## 2.3.5. Reminders

### Description

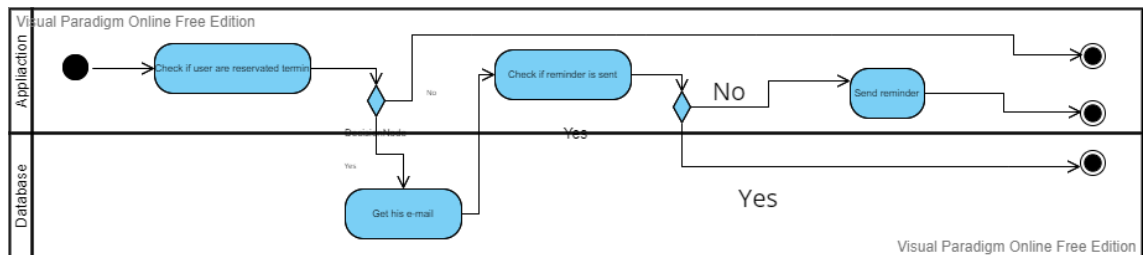
This functionality serves the user to remind him of the training term.

### The script

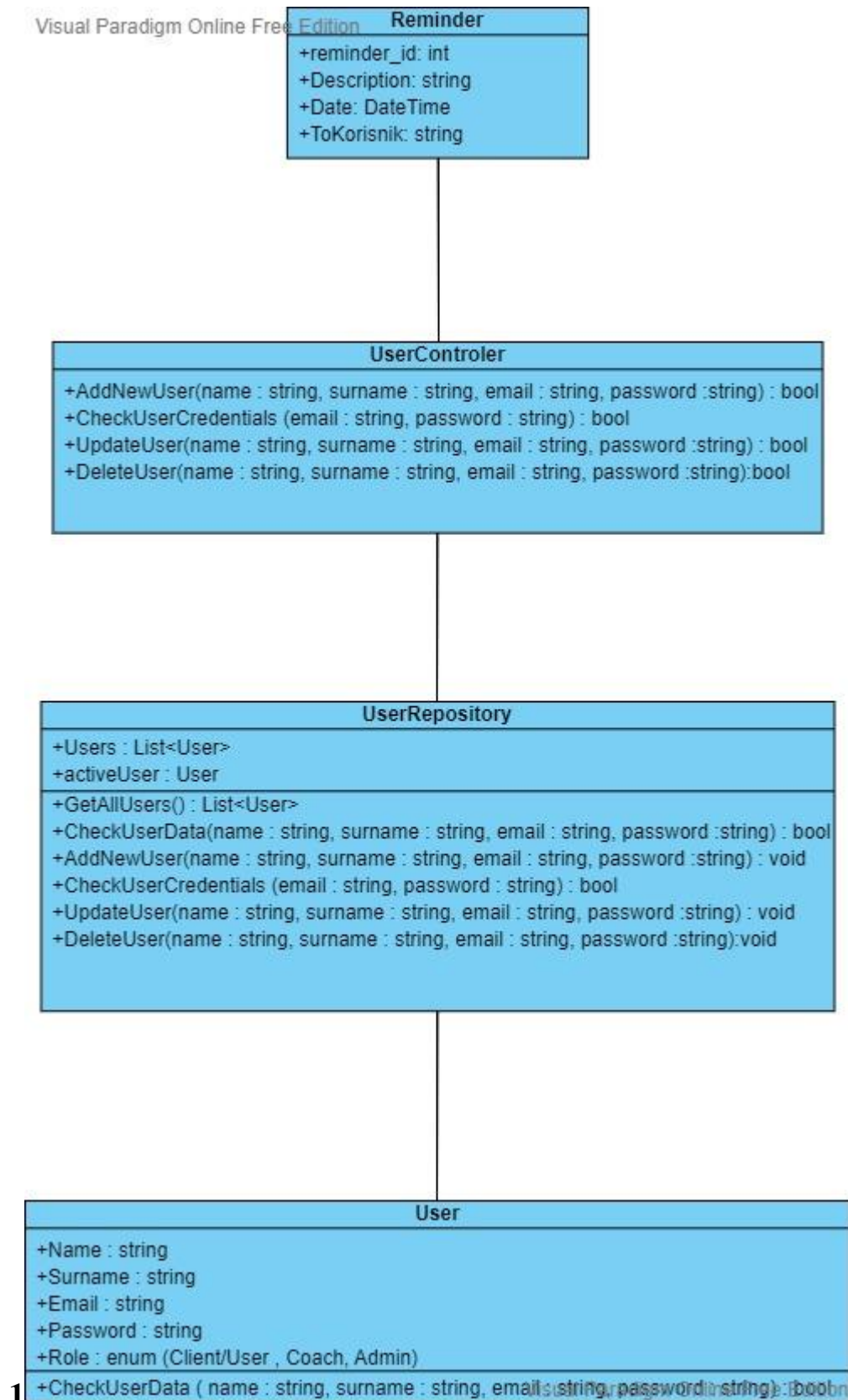
1. The application takes training schedule data from the server.

2. According to that schedule, the server-assisted application finds users who have training.
3. The application takes user data.
4. The application sends the user an email as a training reminder.

## Activity diagram



## Class diagram



### 2.3.6. Coach



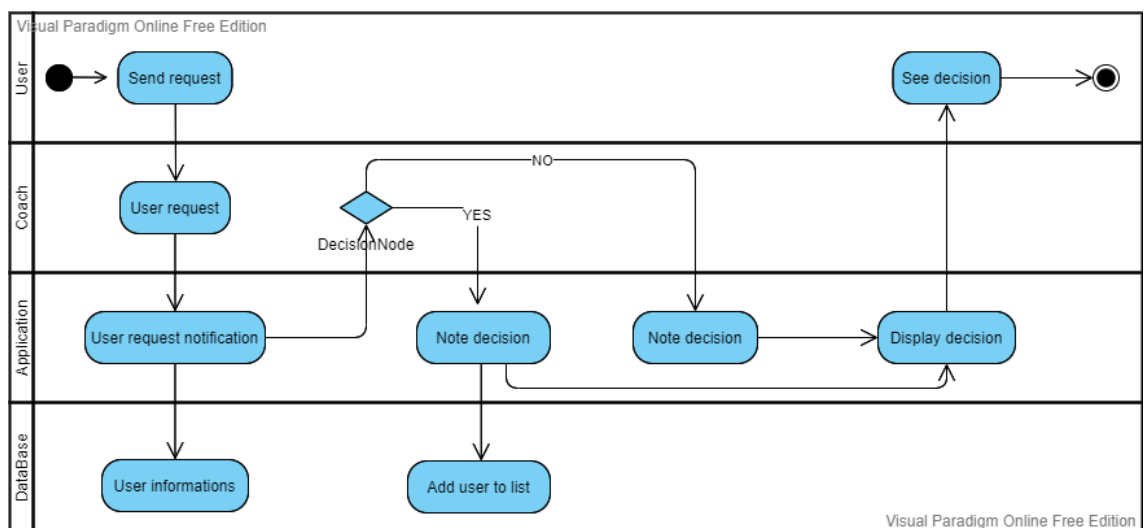
## Description

Ova funkcionalnost napravljena je za trenera da prihvati ili odbije zahtjev korisnika.

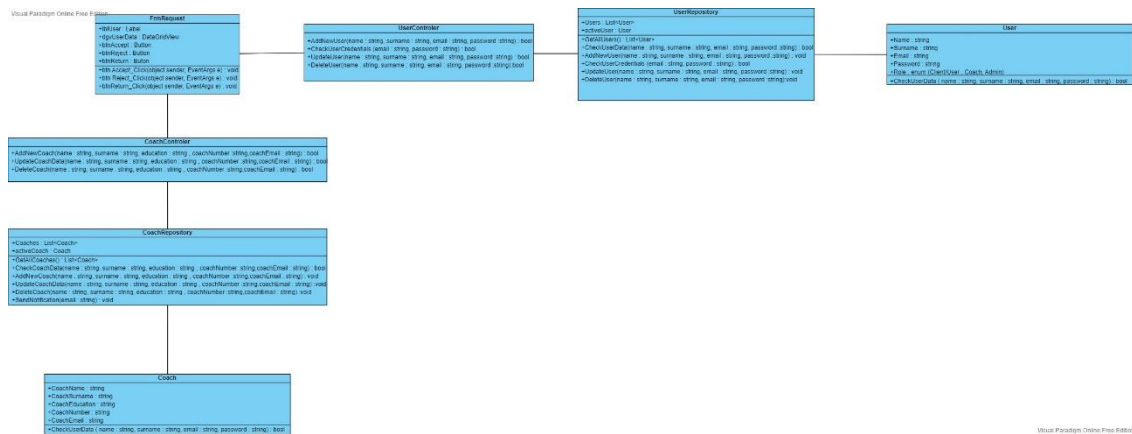
## The script

1. The trainer receives notification via the application about the request of the user who wants to enroll in his training.
2. The application takes user data from the server and displays it to the trainer.
3. The coach can accept, but also refuse the request.
4. If the request is accepted, it is written to the server.
5. The user receives feedback through the application whether the request is accepted or rejected.

## Activity diagram



## Class diagram



### 2.3.7. Coach profiles

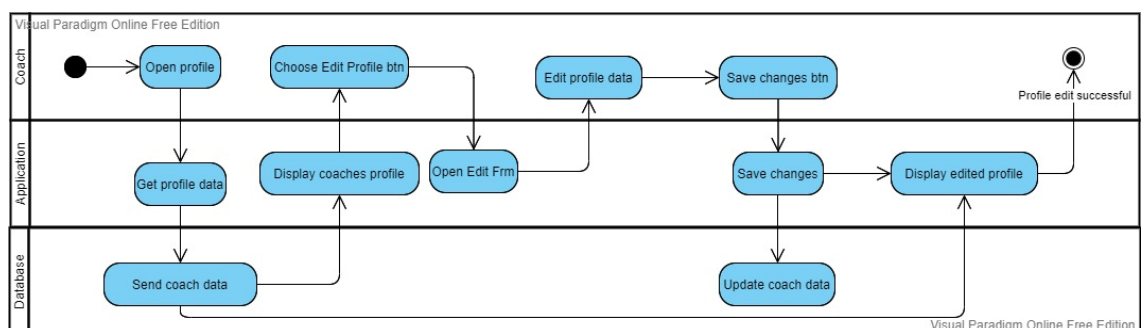
## Description

This functionality is used by the trainer, but it is actually more useful for users to be able to find out information about the trainer they will be working with.

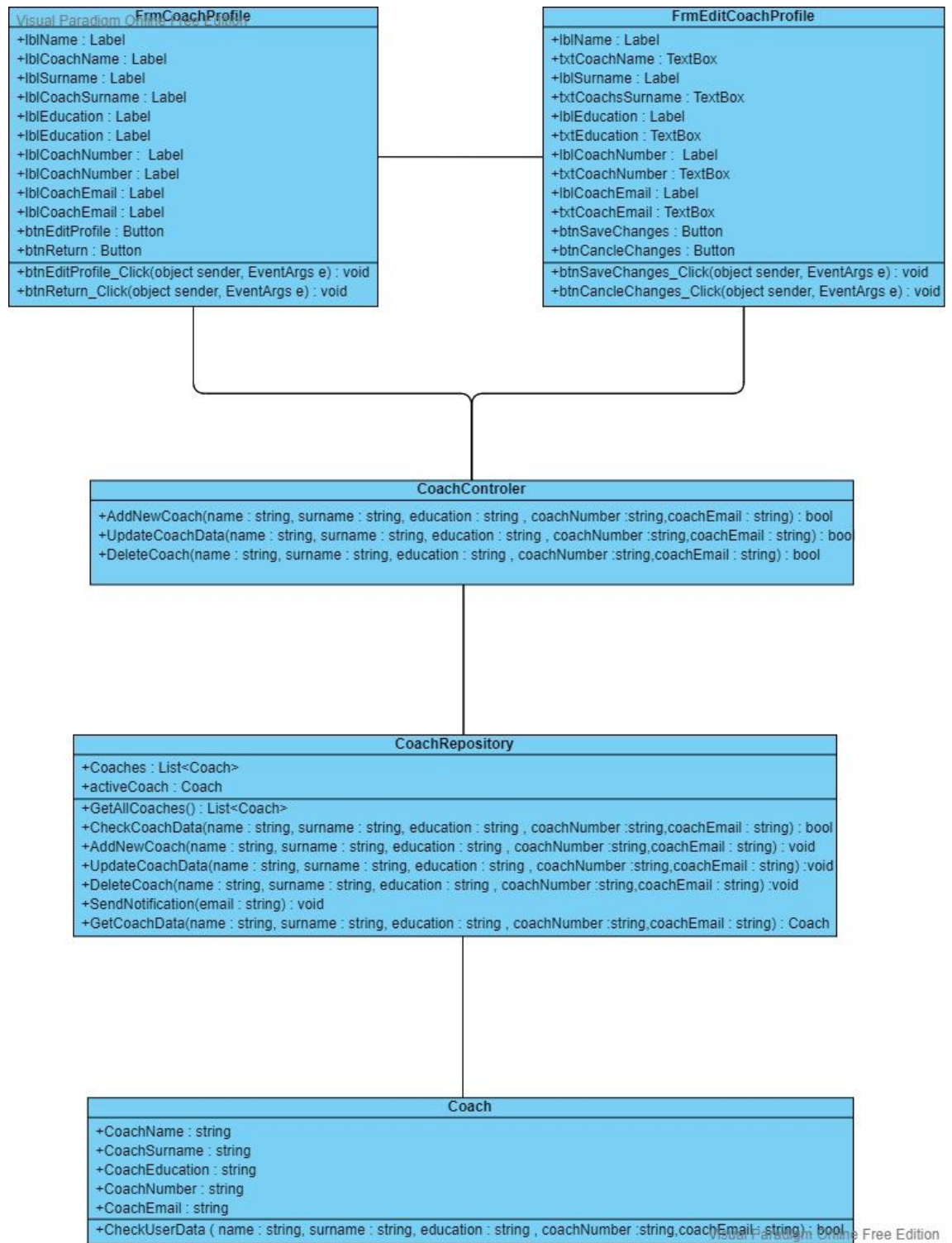
## The script

1. The trainer changes or updates his data through the application.
2. The application sends this data to the server.
3. The server saves new coach information.
4. The application downloads data from the server and displays it to the user and the trainer.

## Activity diagram



## Class diagram



### 2.3.8. User profiles - Superadmin

#### Description

#### The script

**Activity diagram**

**Sequence diagram**

**Class diagram**

### **2.3.9. Statistics**

**Description**

**The script**

**Activity diagram**

**Sequence diagram**

**Class diagram**

### **2.3.10. PDF bill generator**

**Description**

**The script**

**Activity diagram**

**Sequence diagram**

**Class diagram**