An IoT Solution: A Fitness Trainer

Yaroslav Hladkyi[a], Myroslava Gladka[b], Mykola Kostikov[b,c] and Rostyslav Lisnevskyi[b]

[a] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Prosp. Peremohy, 37, Kyiv, 01601, Ukraine*
[b] *Taras Shevchenko National University of Kyiv, Volodymyrska Street, 60, Kyiv, 01601, Ukraine*
[c] *National University of Food Technologies, Volodymyrska Street, 68, Kyiv, 01601, Ukraine*

Abstract

An IoT system of motion detection and control for fitness exercises can be urgent for people who do sports at home. Using such a system as a mobile application allows the user to get the assessment of technique when doing exercises demonstrating them in front of the smartphone camera, as well as recommendations which may help improve doing exercises. As a result, the efficiency of exercises will be increased. These functions are implemented via a neural network able to recognize images. Besides, users will see their progress and information on exercises made with flaws, which will help to fix the execution technique.

The IoT system configures business rules and scenarios, necessary artefacts for users, the exercises they perform, errors and deviations that are committed while doing the exercises, as well as analyzing video files. The IoT solution is built using a neural network that is capable of recognizing the user's body posture during the exercise using a video file captured on the camera of their own smartphone. With the help of mathematical calculations, artificial intelligence will return the result of the video file analysis, where the user will be able to see their flaws while doing a certain task, and get recommendations. A special factor in pattern recognition is the individual initial anthropometric data of each user, which must be taken into account in the analysis.

The use of IoT system of monitoring and control over the performance of fitness exercises will positively affect the trend of a healthy lifestyle in today's world without the involvement of personal fitness trainers.

Keywords [1]

IoT, system, neural network, image recognition, fitness, training, technique, mobile application

1. Introduction

Nowadays, when most professions involve sedentary work, stressful situations, and performing tasks remotely using a PC, the question of recovery and support of full functioning of an organism. emerges. Fitness has become widespread due to its positive effect on the human body: the activation of anabolism, i.e. the accumulation of plastic substances that form body tissues, and energy substances, to ensure vital functions. Full implementation of this process through fitness leads to health improvement when a human body functions in a way that provides complete physical and mental well-being. Therefore, doing fitness exercises, including cardio trainers and various other procedures, has a positive effect on human health state and feelings [1].

The supervision of a coach while doing fitness exercises can eliminate the need for deep knowledge in the field of biomechanics, motion physiology, methods of the training process organization, theoretical points of physical activities and sports [2]. Psychological, methodical, nutrition counselling and pedagogical support of the coach will help to avoid the personal experience of trial and error. No less important is also the psychological aspect. However, the constant

Information Technology and Implementation (IT&I-2021), December 01–03, 2021, Kyiv, Ukraine
EMAIL: hladkky@gmail.com (A. 1); gladkam@fit.knu.ua (A. 2); kostikov@knu.ua (A. 3); lisnevskyir@fit.knu.ua (A. 4)
ORCID: 0000-0003-2914-8685 (A. 1); 0000-0001-5233-2021 (A. 2); 0000-0002-1569-8179 (A. 3); 0000-0002-9006-6366 (A. 4)

supervision of the training process by a fitness trainer limits the time and number of workouts. Therefore, it is important to develop an image recognition system where the position of a fitness trainer will be replaced in part or in full by a highly developed artificial intelligence [3].

Among the mobile application in Ukraine, some offer their users an adaptive schedule for certain exercises. So, the people who don't have the funds to buy a gym membership, or don't have enough free time, have the opportunity to use such applications and maintain the good physical state of the body. The users of such applications follow the instructions of a virtual trainer who develops exercise routines depending on the user's characteristics. But users don't have an opportunity to know how correct is their technique of doing a certain exercise, while it is an integral factor in their efficiency.

However, there are no analogues in the modern market of mobile fitness applications that could offer a real time support for doing exercises by an artificial intelligence. Therefore, such an IoT solution can potentially be an achievement in the field of sports mobile applications.

The aim of the IoT solution under development consists in the right placement of the relations between the most important entities, optimization of performing queries and ensuring the implementation of business processes of various kinds.

2. Methods for Solving the Problem

Artificial neural networks (ANN) is a mathematical functioning model of neural networks that are traditional for living organisms and represent networks of nerve cells [4, 5]. Both biological and artificial neural networks have neurons as their main element. The neurons are interconnected and form layers. The number of layers may be different depending on the complexity of the network and its aim (tasks solved) [6]. Probably the most popular task of the neural networks is visual images recognition [7]. Nowadays the networks are created in which machines are capable of successful recognition of symbols on the paper and bank cards, signatures on official documents, objects detection, etc. These functions allow to facilitate human work significantly, as well as increase reliability and accuracy of various work processes due to avoiding mistakes caused by the human factor [8, 9]. A neural network (NN) is a mathematical model in a form of software and hardware which is based on the functioning principles of biological neural networks [10]. Thus it is appropriate to use this network for image recognition during the training process. A convolutional NN (CNN) has a special architecture that allows it to recognize images most effectively [11, 12] The very idea of CNN is based on the alternation of convolutional and subsampling layers, and the structure is unidirectional. CNN got its name from the convolution operation, which implies that each image fragment will be multiplied by the convolution core element by element, and the result must be added up and written in a similar position of the original image. This architecture provides invariance of recognition regarding the object shift, gradually increasing the "window", which "faces" the convolution, revealing larger and larger structures and patterns in the image [13].

2.1. Positioning for Image Scanning

Anthropometric parameters of a person performing a fitness exercise are key indicators in determining and recognizing images when performing fitness exercises [14, 15]. Therefore, the basic initial values for the correct calculations of the performance quality should be the following parameters: weight, height, length of different body parts (Fig. 1: 1x, 2x, 3x, 4x, 5x), circumferences of the chest, waist, hips, limbs (Fig. 1: 1v, 2v, 3v, 4v, 5v, 6v, 7v, 8v), as well as physiological indicators of key articular joints locations and distances between them. To do this, it is necessary to measure and use these indicators as a basis for calculations (Fig. 1, e.g., the length of 2r-4r, 6l-7l, etc.) [16]. It should be noted that all points parameters, except for 1 and 3, are represented by double values: separately for the right and left side of the body.

2.2. A Neural Network for User's Motion Recognition

For the correct work of a neural network for motion detection and recognition, the system uses LMST architecture. It is based on anthropometric indicators and representing images of exercise execution. This includes sets of body parts motion and also individual elements recorded while being done by 5 fitness coaches and 24 performers who were not professional athletes. The training process of this neural network is based on patterns depending on indicators of individual body part motion in correlation to the motion range expected when doing a certain exercise element [10, 15]. After this training, we get an encoder that allows predicting an acceptable motion range for each exercise according to the given individual parameters of the user. The function of adding new exercises is implemented by the same principles, with each exercise being represented as a set of individual elements (or parts of elements) of the performer's motion.

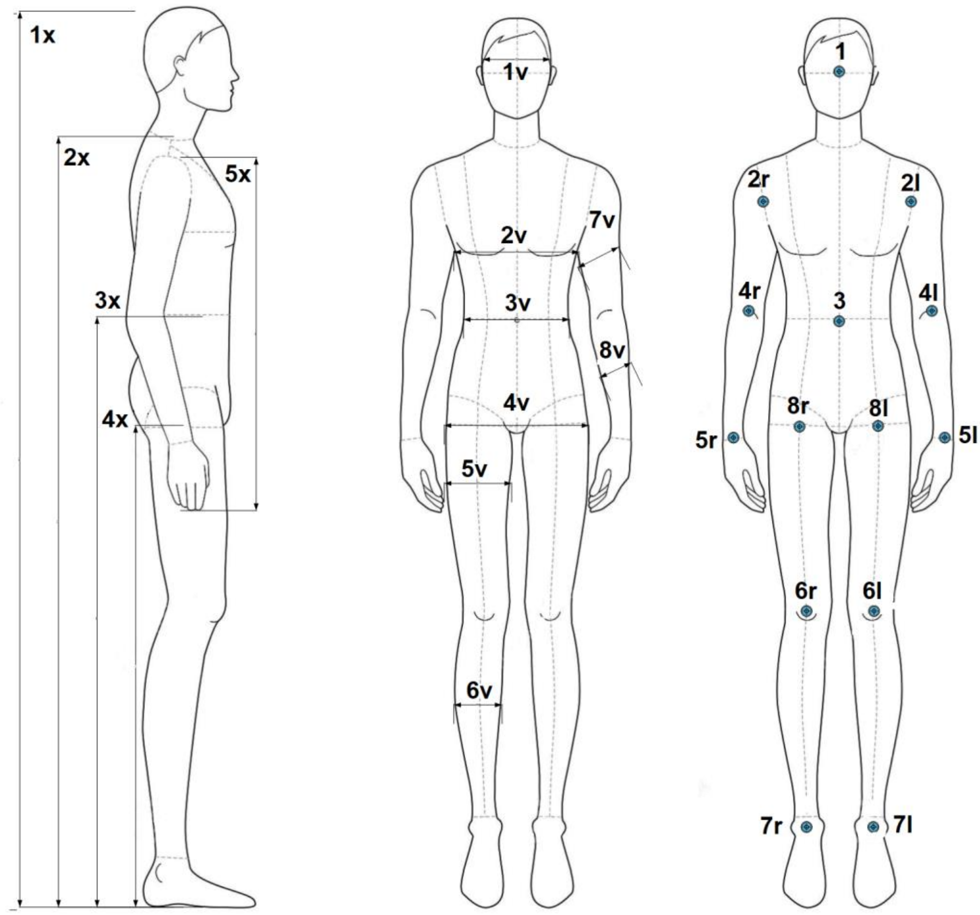

Figure 1: Key points for anthropometric measurements

2.3. Neural Networks in Mobile Applications

Mobile applications development reaches a new level every year. Applying machine learning algorithms in this field becomes a problem which is being solved more and more often [17, 18]. The main aim lies in optimization of certain computing resources. This leads to the possibility of integrating such algorithms into a mobile application. Another solution may consist in transferring all the computing mechanisms to the server. In this case, smartphone users need to have a stable internet connection in order to use the system functions [19, 20]. Nowadays many services that use the possibilities of machine learning are implemented according to the latter scenario [21].

2.4. Business Processes of a Fitness Trainer

All the functioning of the fitness trainer must be described as a set of rules and algorithms that allow to create a set of exercises, monitor the execution and analyze the results. To develop the algorithms, we need to create business processes that are shown in the Table 1.

The presented business rules are connected with the next views:

- entities: Video, Smartphone Camera, Neural Network, Message, Errors when Doing Exercise, Exercise Execution Result, Execution Assessment, Fitness Exercise, IS User.
- activities: User does the exercise, User fims the exercise with the smartphone camera, Naural network analyzes the execution process from the smartphone camera, Neural network counts the errors when doing the certain exercise, User gets the message from the neural network.

The neural network is an actor behind the scenes that is external to the system under development. Thus, it isn't being modelled. Only the input and output data of this component are described. They are given in the form of anthropometric indicators (fig. 1).

Table 1

Business rules for a mobile fitness trainer functioning

Business Process	User	Aim	Task	Scenario
Adding a new exercise for possible analysis	Main: DB Administrator	To add a new pattern for detecting body parts in a new pose	To save the new parameters for the neural network in the DB	The DB Administrator gets the parameters in the form of digital data from a neural network that was trained to a new pattern which allows to record the deviations in the body posture of the IS User, and stores them in the DB. The IS User can use the new functions after starting a new fitness exercise in front of the camera. The video from the camera will be analyzed by the Neural Network that will use the parameters stored in the DB and will alert about the possible deviations (assess the exercise execution on a certain scale)
	Auxiliary: Neural Network	To learn a new pattern	To create new parameters for recognizing deviations in the body posture according to a new pattern	
	Behind the scenes: IS User	To use the new functions of the system	Having the possibility to learn about the deviation in the posture when doing a new fitness exercise	
Buying a premium account	Main: IS User	To buy a premium account that enhances the available functions	To make a payment in order to get new functions from the IS (e.g., some patterns)	The User wants to get a premium account and makes a payment. The DB Administrator gets a message about the payment and checks it. In case of the successful funds transfer, he gives the user the access to the premium account. Otherwise, does nothing. The IS User gets the message about the successful purchase of the account or cancellation of the purchase operation cancellation in case of an error
	Auxiliary: DB Administrator	To add the IS user to premium users in the DB	To get a message about the user's payment for the premium account and give him the access to premium tools of the IS	
Correction of mistakes	Main: IS User	To redo exercises with the highest number of errors	To get a list of exercises with the highest number of errors during execution that need to be redone	The IS User does the exercises in front of the camera. A neural network analyzes the execution of each exercise and counts the number of errors. Then the data about the exercise execution are obtained by the DB Administrator who organizes exercises prioritizing ones which have to be done better and saves them into the DB. The IS User gets the information about exercises with the maximum number of committed errors and corrects them by doing the problematic exercises in front of the camera.
	Auxiliary: Neural Network	Each exercise is assessed, and the errors number is counted	To transfer the information about the deviations during execution of each exercise to the DB Administrator	
	Auxiliary: DB Administrator	To save the information about errors in the DB	To get the data about the exercises from the Neural Network. To sort the IDs in descending order by the key of committed errors. To save the exercise order into the DB	

Business Process	User	Aim	Task	Scenario
Doing the exercise by the user in front of the camera and getting the information about the execution technique	Main: IS User	To do an exercise and get the assessment of the execution technique	To turn on the camera and to do the exercise in front of it in a strictly defined position and from the best angle	The IS User logs in to the IS with their login and password, turns on the camera and starts filming an arbitrary fitness exercise with the camera. The exercise execution is being analyzed by the Neural Network for a certain time period. After the end of exercise, the User gets the message about the execution technique and counts the committed errors.
	Behind the scenes: Neural Network	To analyze the exercise done by the user	To return the errors in execution technique for the exercise filmed on a camera to the User	

2.5. Representation of the Fitness Trainer Model

The modelled business processes and defined views allow us to form a model of entity-relationship dependencies (fig. 2.). The created diagram includes 6 entities with 14 attributes, and 8 relationships between entities. This is the basis for building a class diagram (fig. 3) that is presented in UML notation [22].

2.6. Data Elements Specification

The defined attributes and their description for the database under development are described in the table 2 [23].

Table 2
Entities and attributes of the database under development

№	Entity	Attributes	Entity Description
1	User	login	The main user of the system who directly uses its functions
		password	
2	Exercise	name	An exercise that can be done and filmed by the user
		exercise group	
		description	
3	Error	scale	An error made by the user when doing the exercise
		description	
4	Videofile	size	A video file that contains doing the exercise by the user
		duration	
		format	
		address	
5	Result Message	result date	A message presented to the user after doing the exercise and analyzing the video file
		message text	
6	Result	indicator of the analysis	The result of analyzing the user's video file

3. Practical Implementation

3.1. Views

For views, let's define the key entities that represent the nature of a fitness trainer [24]. The first user data type is the video file format which is described in the Table 3. Another view is the data type that defines the scale of the error made (Table 4). The exercise list that includes certain fitness activities are described in the Table 5.

Queries for creating user data types:

- create type VideoFileFormatType as enum ('mp4', 'avi', 'mov');
- create type ErrorGrossType as enum ('blunder', 'average', 'slight');

- create type ExerciseGroupType as enum ('legs', 'arms', 'chest', 'back', 'abs')

Other views are implemented according to the presented types of corresponding entities (fig. 3).

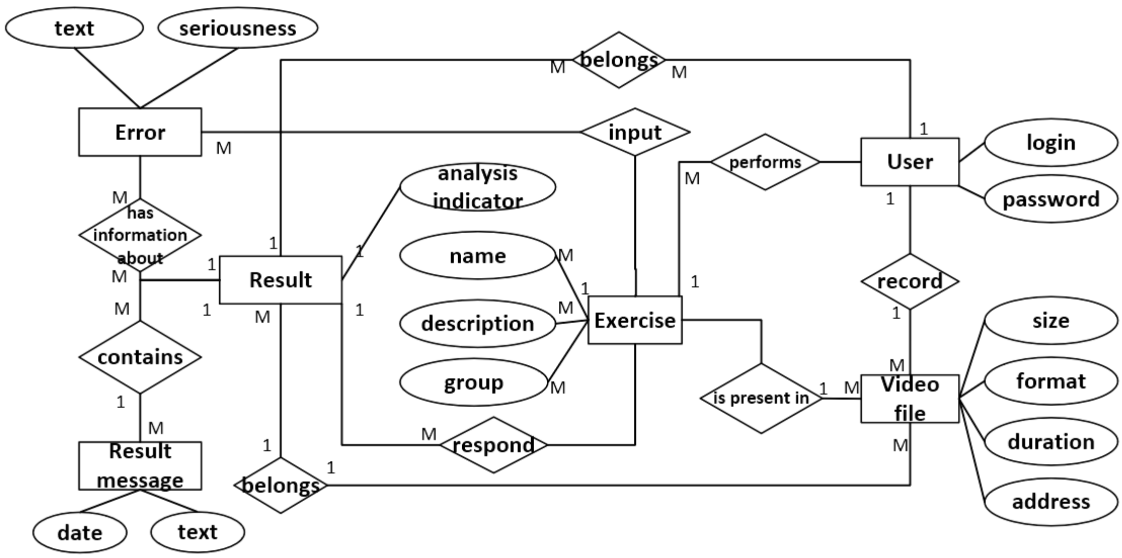

Figure 2: Entity-Relationship Diagram

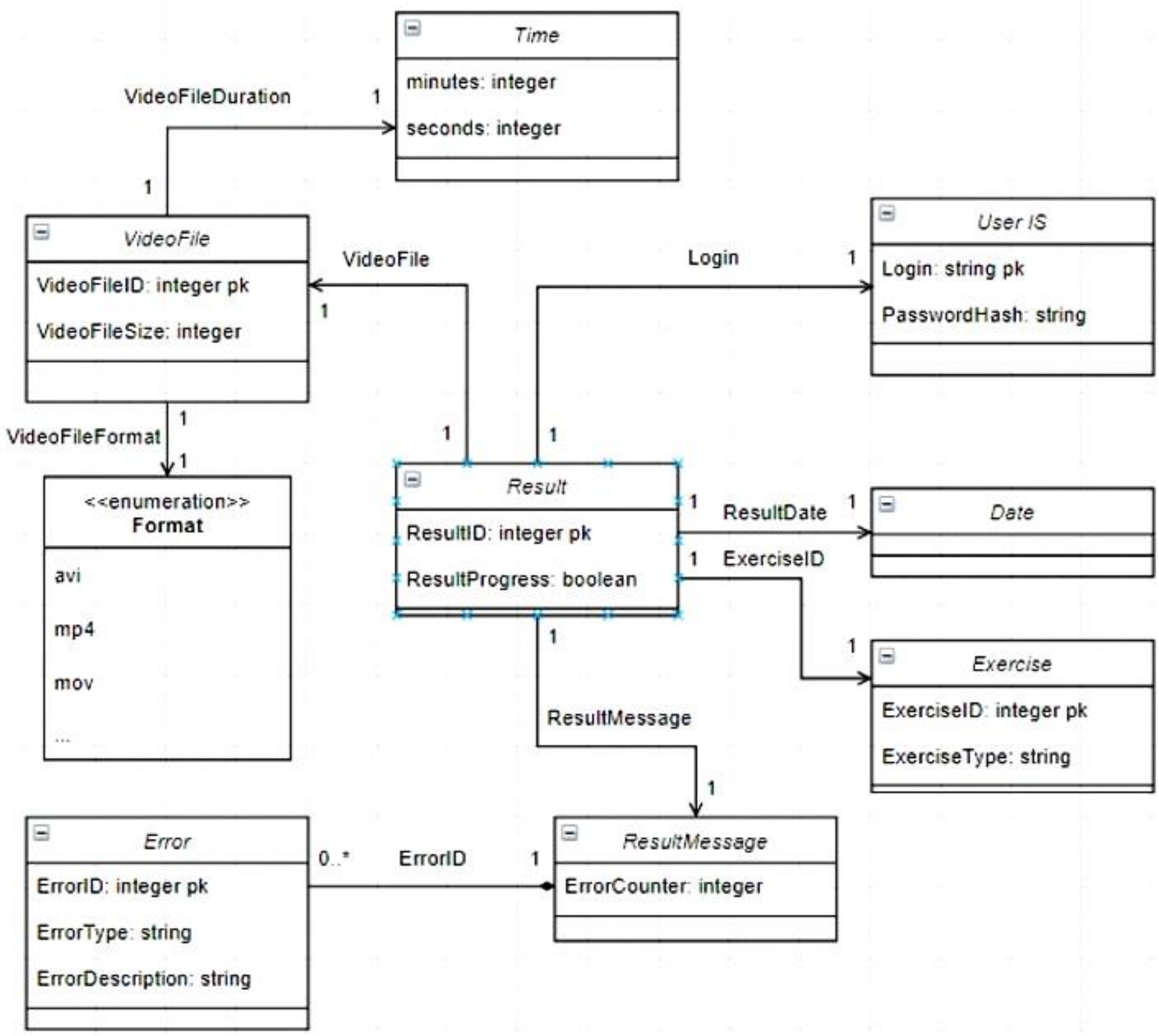

Figure 3: UML Class Diagram

Table 3
Fields of the list "VideoFileFormatType"

№	Field Name	Description
1	mp4	A possible videofile format
2	avi	A possible videofile format
3	mov	A possible videofile format

Table 4
Fields of the list "ErrorGrossType"

№	Field Name	Description
1	blunder	A blunder
2	average	An average arror
3	slight	A slight error

Table 5
Fields of the list "ExerciseGroup"

№	Field Name	Description
1	legs	Legs development exercise group
2	arms	Arms development exercise group
3	body	Chest development exercise group
4	back	Back development exercise group
5	abs	Abdominal muscles development exercise group

3.2. Functional and Ambiguous Dependencies

The user has a login and a password connected to it. They are defined by a unique user ID: userid → login(1), passwordhash (2).

Given the fitness exercise ID, it is possible to determine its name, exercise group, and the description of its execution: exerciseid → exercisename(3), exercisegroup(4), exercisedescription(5).

Given the video file ID, it is possible to determine all of its attributes: videofileid → videofilesize (6), videofileformat (7), videofileduration (8), videofileuri(9).

The result ID contains the information about the success of the analysis: resultid → analysisperformed(10).

In addition, this result determines the user whose video was analyzed, the video file itself, the exercise performed on it, and the message that will be sent to the user: resultid → userid(11), videofileid(12), exerciseid(13), resultmessageid(14).

Besides, the result ID defines errors detected during the analysis: resultid ↠ errorid(15).

The ID of the message about the result gives the information about the message date and its text: resultmessageid → resultmessagedate(16), resultmessagetext(17).

In turn, the error ID can help to determine all of its attributes: errorid → errorgross (18), errordescription (19).

3.3. Normalization

The next step will be the decomposition process for normalizing the relation to the 4th normal form [25]. The initial relation is as follows:

R (userid login passwordhash exerciseid exercisename exercisegroup exercisedescription videofileid videofilesize videofileformat videofileduration videofileuri resultid analysisperformed resultmessageid resultmessagedate resultmessagetext errorid errorgross errordescription)

Key: resulted.

Step 1: FD 6-9 to R:

R1 (videofileid videofilesize videofileformat videofileduration videofileuri).

Key: videofileid.

R2 (userid login passwordhash exerciseid exercisename exercisegroup exercisedescription videofileid resultid analysisperformed resultmessageid resultmessagedate resultmessagetext errorid errorgross errordescription).

Key: resulted.

Step 2: FD 18-19 to R2:

R3 (errorid errorgross errordescription).
Key: errored.
R4 (userid login passwordhash exerciseid exercisename exercisegroup exercisedescription videofileid resultid analysisperformed resultmessageid resultmessagedate resultmessagetext errorid).
Key: resulted.
Step 3: FD 16-17 to R4:
R5 (resultmessageid resultmessagedate resultmessagetext).
Key: resultmessageid.
R6 (userid login passwordhash exerciseid exercisename exercisegroup exercisedescription videofileid resultid analysisperformed resultmessageid errorid).
Key: resulted.
The normalization of other views is made similarly. The obtained relations in the 4th normal form:

- R1 (videofileid videofilesize videofileformat videofileduration videofileuri)
- R3 (errorid errorgross errordescription)
- R5 (resultmessageid resultmessagedate resultmessagetext)
- R7 (userid login passwordhash)
- R9 (exerciseid exercisename exercisegroup exercisedescription)
- R11 (resultid errorid)
- R12 (resultid analysisperformed userid exerciseid videofileid resultmessageid)

3.4. Tables and Subject Area Constraints

The table **videofile** represents the video file entity and corresponds to the relation R1. It has a unique identifier for each row. All other attributes cannot be null, and the video file address is unique:

```
create table videofile (
    videofileid serial primary key,
    videofilesize integer not null,
    videofileformat VideoFileFormatType not null,
    videofileduration integer not null,
    videofileuri varchar(128) unique not null
);
```

The table **error** represents the error entity and corresponds to the relation R3. It has a unique identifier for each row. The error description and the error scale attribute cannot be null:

```
create table error (
    errorid serial primary key,
    errorgross ErrorGrossType not null,
    errordescription text not null
);
```

The table **resultmessage** represents the result message entity and corresponds to the relation R5. . It has a unique identifier for each row. All the other attributes cannot be null. Besides, the result date cannot be later than today:

```
create table resultmessage (
    resultmessageid serial primary key,
    resultmessagedate date not null check (resultmessagedate <= current_date),
    resultmessagetext text not null
);
```

The table **user** represents the user entity and corresponds to the relation R7. It has a unique identifier for each individual user, and the login field cannot be null:

```
create table "user" (
    userid serial primary key,
    login varchar(16) unique not null,
    passwordHash text
);
```

Besides, the password must have at least 8 symbols. Otherwise, a hashed password is entered into the table:

```
create or replace function user_password_hashing_trigger()
returns trigger as
$$
begin
    if length(new.passwordhash) < 8 then
          raise exception 'Too short password';
    end if;
    new.passwordHash = crypt(new.passwordHash, gen_salt('md5'));
    return new;
end;
$$ language PLPGSQL;
create trigger user_password_hashing
    before insert on "user"
    for each row
    execute procedure user_password_hashing_trigger();
```

The table **result** represents the result entity and corresponds to the relation R12. It has not null foreign keys for corresponding tables, except for the result message ID, as if the attribute responsible for the analysis (which also cannot be null) takes 'false' value, there will be no message for this result:

```
create table result (
    resultid serial primary key,
    analysisperformed boolean not null,
    userid integer not null references "user"(userid),
    exerciseid integer not null references exercise(exerciseid),
    videofileid integer not null references videofile(videofileid),
    resultmessageid integer references resultmessage(resultmessageid)
);
```

After creating the **result** table, in the table **errorsonresult** it is necessary to perform the restriction of the foreign key to the former table:

```
alter table errorsonresult
    add constraint resultid_fk
    foreign key(resultid) references result(resultid)
    on delete cascade;
```

Similarly, we form views, triggers and algorithms for other entities.

4. Result

When implementing the fitness trainer mobile application, we perform the analysis of the views for business processes, presented in the table 1.

The first execution plan responds to a query that returns all gross errors made by users:
explain analyze
select count(*) from error e
where errorgross = 'blunder';

Execution plans without and with indices are illustrated on fig. 4 and 5 respectively.

QUERY PLAN
Aggregate (cost=42.21..42.22 rows=1 width=8) (actual time=0.643..0.644 rows=1 loops=1)
-> Seq Scan on error e (cost=0.00..40.51 rows=678 width=0) (actual time=0.020..0.558 rows=678 loops=1)
Filter: (errorgross = 'blunder'::errorgrosstype)
Rows Removed by Filter: 1363
Planning Time: 1.073 ms
Execution Time: 0.683 ms

Figure 4: A query execution plan before indexing

An execution plan for a query that returns all the users and the exercise group along with the number of exercises from this group that were done today, sorted by the number of exercises in descending order:

```
select login, exercisegroup, count(*) as total from result
natural join resultmessage r
natural join exercise e2
natural join "user" u
where resultmessagedate = current_date
group by (login, exercisegroup)
order by 3 desc;
```

The execution plans without and with indices are illustrated on figures 6 and 7.

```
QUERY PLAN
Aggregate  (cost=25.84..25.85 rows=1 width=8) (actual time=0.911..0.912 rows=1 loops=1)
  ->  Index Only Scan using errorgrossindex on error e  (cost=0.28..24.14 rows=678 width=0) (actual time=0.740..0.855 rows=678 loops=1)
        Index Cond: (errorgross = 'blunder'::errorgrosstype)
        Heap Fetches: 0
Planning Time: 1.807 ms
Execution Time: 0.956 ms
```

Fig.5 A query execution plan after indexing

```
QUERY PLAN
Sort  (cost=119.57..120.41 rows=336 width=19) (actual time=0.932..0.945 rows=336 loops=1)
  Sort Key: (count(*)) DESC
  Sort Method: quicksort  Memory: 51kB
  ->  HashAggregate  (cost=102.11..105.47 rows=336 width=19) (actual time=0.843..0.886 rows=336 loops=1)
        Group Key: u.login, e2.exercisegroup
        ->  Hash Join  (cost=70.59..99.59 rows=336 width=11) (actual time=0.513..0.765 rows=336 loops=1)
              Hash Cond: (result.exerciseid = e2.exerciseid)
              ->  Hash Join  (cost=52.04..80.15 rows=336 width=11) (actual time=0.490..0.687 rows=336 loops=1)
                    Hash Cond: (u.userid = result.userid)
                    ->  Seq Scan on "user" u  (cost=0.00..21.00 rows=1000 width=11) (actual time=0.004..0.061 rows=1000 loops=1)
                    ->  Hash  (cost=47.84..47.84 rows=336 width=8) (actual time=0.481..0.482 rows=336 loops=1)
                          Buckets: 1024  Batches: 1  Memory Usage: 22kB
                          ->  Hash Join  (cost=27.20..47.84 rows=336 width=8) (actual time=0.253..0.445 rows=336 loops=1)
                                Hash Cond: (result.resultmessageid = r.resultmessageid)
                                ->  Seq Scan on result  (cost=0.00..18.00 rows=1000 width=12) (actual time=0.005..0.058 rows=1000 loops=1)
                                ->  Hash  (cost=23.00..23.00 rows=336 width=4) (actual time=0.240..0.240 rows=336 loops=1)
                                      Buckets: 1024  Batches: 1  Memory Usage: 20kB
                                      ->  Seq Scan on resultmessage r  (cost=0.00..23.00 rows=336 width=4) (actual time=0.007..0.205 rows=336
                                            Filter: (resultmessagedate = CURRENT_DATE)
                                            Rows Removed by Filter: 664
              ->  Hash  (cost=13.80..13.80 rows=380 width=8) (actual time=0.019..0.019 rows=10 loops=1)
                    Buckets: 1024  Batches: 1  Memory Usage: 9kB
                    ->  Seq Scan on exercise e2  (cost=0.00..13.80 rows=380 width=8) (actual time=0.013..0.015 rows=10 loops=1)
Planning Time: 0.298 ms
Execution Time: 1.038 ms
```

Figure 6: A query execution plan before indexing

An execution plan for a query that returns all the video files larger than 30 MB, file format MP4 and containing at least 3 errors.

```
select videofileid, count(e.errorid) from videofile
natural join result
natural join errorsonresult e
where videofilesize > 30000000
```

```sql
and videofileformat = 'mp4'
group by videofileid
having count(errorid) > 2
order by 2 desc;
```

Query plans are developed for working with each view of the system.

5. Conclusions

The development of the IoT fitness trainer resulted in designing a mobile application that stores the information about exercise types, parameters and execution features, anthropometric parameters of the user. It allows users to monitor the correctness of doing fitness exercises in real time thanks to using neural networks for image recognition. To develop this solution, DBMS PostgreSQL was used. All the scripts in SQL language were written and executed in a free cross-platform tool named DBeaver and in the terminal of the DBMS itself (PSQL Shell).

The product is based on the described business processes related to the fitness trainer IoT solution. An important concept of the presented solution is the support of the people health under quarantine restrictions, lack of time or opportunities to attend fitness classes with a coach. Using the presented IoT solution will fully ensure the accessibility for everyone, and the introduction of image recognition technology implements the mechanism of correct performance and correlation of user performance, which eliminates the risk of damage from improper fitness technique.

```
QUERY PLAN
Sort  (cost=119.57..120.41 rows=336 width=19) (actual time=2.463..2.497 rows=336 loops=1)
  Sort Key: (count(*)) DESC
  Sort Method: quicksort  Memory: 51kB
  ->  HashAggregate  (cost=102.11..105.47 rows=336 width=19) (actual time=2.223..2.345 rows=336 loops=1)
        Group Key: u.login, e2.exercisegroup
        ->  Hash Join  (cost=70.59..99.59 rows=336 width=11) (actual time=1.348..2.024 rows=336 loops=1)
              Hash Cond: (result.exerciseid = e2.exerciseid)
              ->  Hash Join  (cost=52.04..80.15 rows=336 width=11) (actual time=1.304..1.826 rows=336 loops=1)
                    Hash Cond: (u.userid = result.userid)
                    ->  Seq Scan on "user" u  (cost=0.00..21.00 rows=1000 width=11) (actual time=0.011..0.166 rows=1
                    ->  Hash  (cost=47.84..47.84 rows=336 width=8) (actual time=1.282..1.284 rows=336 loops=1)
                          Buckets: 1024  Batches: 1  Memory Usage: 22kB
                          ->  Hash Join  (cost=27.20..47.84 rows=336 width=8) (actual time=0.654..1.169 rows=336 loops
                                Hash Cond: (result.resultmessageid = r.resultmessageid)
                                ->  Seq Scan on result  (cost=0.00..18.00 rows=1000 width=12) (actual time=0.012..0.153 ro
                                ->  Hash  (cost=23.00..23.00 rows=336 width=4) (actual time=0.628..0.629 rows=336 loops=
                                      Buckets: 1024  Batches: 1  Memory Usage: 20kB
                                      ->  Seq Scan on resultmessage r  (cost=0.00..23.00 rows=336 width=4) (actual time=0.0
                                            Filter: (resultmessagedate = CURRENT_DATE)
                                            Rows Removed by Filter: 664
              ->  Hash  (cost=13.80..13.80 rows=380 width=8) (actual time=0.036..0.036 rows=10 loops=1)
                    Buckets: 1024  Batches: 1  Memory Usage: 9kB
                    ->  Seq Scan on exercise e2  (cost=0.00..13.80 rows=380 width=8) (actual time=0.023..0.027 rows=
Planning Time: 4.294 ms
Execution Time: 2.634 ms
```

Figure 7: A query execution plan after indexing

6. References

[1] Keane, B.: The Fitness Mindset: Eat for energy, Train for tension, Manage your mindset, Reap the results. Rethink Press Limited, Norfolk (2019)

[2] Swettenham, N.: Total Fitness After 40: The 7 Life Changing Foundations You Need for Strength, Health and Motivation in your 40s, 50s, 60s and Beyond. Independently published (2021)
[3] Logothetis, N. K., Sheinberg, D. L.: Visual Object Recognition. Annu. Rev. Neurosci. 19, 577–621 (1996). https://doi.org/10.1146/annurev.ne.19.030196.003045
[4] Gill, N. S.: Artificial Neural Networks Applications and Algorithms (2021). https://www.xenonstack.com/blog/artificial-neural-network-applications
[5] Kriesel, D.: A Brief Introduction to Neural Networks (2007). http://www.dkriesel.com/en/science/neural_networks
[6] Chen, J.: Neural Network (2020). https://www.investopedia.com/terms/n/neuralnetwork.asp
[7] Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Pearson Education, London (2011)
[8] Osowski, S.: Sieci neuronowe do przetwarzania informacji [Neural Networks for Information Processing]. Oficyna Wydawnicza PW, Warsaw (2000)
[9] Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Hoboken, NJ (1998)
[10] James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning (with Applications in R). Springer, New York, NY (2013)
[11] Saha, S.: A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way. Towards Data Science (2018). https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[12] Shankar, S., Robertson, D., Ioannou, Y., Criminisi, A., Cipolla, R.: Refining Architectures of Deep Convolutional Neural Networks. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 2212–2220 (2016).
[13] Shapiro, L. G., Stockman, G. C.: Computer Vision. Pearson, London (2001)
[14] Durnin, J. V., Womersley, J.: Body fat assessed from total body density and its estimation from skinfold thickness: measurements on 481 men and women aged from 16 to 72 Years. Br. J. Nutr. 32(1), 77–97 (1974). https://doi.org/10.1079/bjn19740060
[15] Bland, M.: An Introduction to Medical Statistics. University Press, Oxford (2015)
[16] Fedotov, N. G.: Theory of Signs of Recognition of Patterns on the Basis of Stochastic Geometry and Functional Analysis. Fizmatlit, Moscow (2015)
[17] Horton, J.: Android Programming with Kotlin for Beginners. Packt Publishing, Birmingham (2019)
[18] Benedetto, J. I., Sanabria, P., Neyem, A., Navon, J., Poellabauer, C., Xia, B.: Deep Neural Networks on Mobile Healthcare Applications: Practical Recommendations. In: Proceedings of the 12th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2018), 2(19), 550 (2018). https://doi.org/10.3390/proceedings2190550
[19] Cisco Annual Internet Report (2020). https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/index.html
[20] Angelova, N.: Mobile Applications for Business. Trakia J. Sci., 17(Suppl.1), 853–859 (2019). http://dx.doi.org/10.15547/tjs.2019.s.01.140
[21] Arovina, M.: Prospects of Application of Mobile Apps in the Regional Information Market of Ukraine. Skhid, 3, 5–10 (2016). http://nbuv.gov.ua/UJRN/Skhid_2016_3_2
[22] Lopes, A. G.: Business Process Modeling with UML. In: ICEIS (2), pp. 679-685 (2001)
[23] Liu, L., Özsu, M. T. (Eds.): Encyclopedia of Database Systems. Springer, New York (2009)
[24] Blockeel, H., Calders, T., Fromont, E., Goethals, B.: Mining Views: Database Views for Data Mining. In: Proceedings 24th IEEE International Conference on Data Engineering (ICDE'08, Cancún, Mexico, April 7-12, 2008), 1608–1611 (2008)
[25] Garcia-Molina, H., Ullman, J. D., Widom, J. D. Database Systems: The Complete Book. Pearson, London (2008)