FELADATKIÍRÁS

Az elektronikusan beadott változatban ez az oldal törlendő. A nyomtatott változatban ennek az oldalnak a helyére a diplomaterv portálról letöltött, jóváhagyott feladatkiírást kell befűzni.



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Fazekas Bence

Android alapú testedzést támogató alkalmazás fejlesztése

Konzulens

Dr. Ekler Péter

BUDAPEST, 2018

Tartalomjegyzék

[Összefoglaló 6](#_Toc531202686)

[Abstract 7](#_Toc531202687)

[1 Bevezetés 8](#_Toc531202688)

[1.1 Témaválasztás 8](#_Toc531202689)

[1.2 Hasonló megoldások 9](#_Toc531202690)

[1.3 A szakdolgozat felépítése 10](#_Toc531202691)

[2 Alkalmazás ismertetése 11](#_Toc531202692)

[2.1 Feladatspecifikáció 11](#_Toc531202693)

[2.2 Felhasználók 11](#_Toc531202694)

[2.2.1 Felhasználó 12](#_Toc531202695)

[2.2.2 Nem regisztrált felhasználó 12](#_Toc531202696)

[2.3 Funkciók 13](#_Toc531202697)

[2.3.1 Regisztráció 13](#_Toc531202698)

[2.3.2 Bejelentkezés 15](#_Toc531202699)

[2.3.3 Gyakorlatok megjelenítése 17](#_Toc531202700)

[2.4 Kalóriaigény számítás 18](#_Toc531202701)

[3 Felhasznált technológiák 19](#_Toc531202702)

[3.1 Android platform 19](#_Toc531202703)

[3.2 Node.js 20](#_Toc531202704)

[3.3 Express 20](#_Toc531202705)

[3.4 MySQL 20](#_Toc531202706)

[3.4.1 Lekérdezések 21](#_Toc531202707)

[3.4.2 Adatok bevitele 21](#_Toc531202708)

[3.4.3 Adatok módosítása 21](#_Toc531202709)

[3.4.4 Adatok törlése 22](#_Toc531202710)

[3.5 JSON Web Token 22](#_Toc531202711)

[3.6 REST 23](#_Toc531202712)

[4 Architektúra bemutatása 24](#_Toc531202713)

[4.1 Háromrétegű architektúra 24](#_Toc531202714)

[5 Adatbázis elkészítésének a folyamata 26](#_Toc531202715)

[5.1 Kapcsolatok a táblák között 26](#_Toc531202716)

[5.2 Users tábla 26](#_Toc531202717)

[5.3 Trainings tábla 27](#_Toc531202718)

[5.4 UsersTrainings tábla 28](#_Toc531202719)

[6 Szerveralkalmazás elkészítésének a folyamata 29](#_Toc531202720)

[7 Mobilalkalmazás elkészítésének a folyamata 30](#_Toc531202721)

[8 Mobilalkalmazás bemutatása 31](#_Toc531202722)

[9 Tesztelés 32](#_Toc531202723)

[10 Összefoglalás 33](#_Toc531202724)

[11 Irodalomjegyzék 34](#_Toc531202725)

Hallgatói nyilatkozat

Alulírott **Fazekas Bence**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző, cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy hitelesített felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Kelt: Budapest, 2018. 12. 07

...…………………………………………….

Fazekas Bence

Összefoglaló

A mai világban egyre fontosabb az emberek számára, hogy egészségesen táplálkozzanak és rendszeresen sportoljanak. Észrevehetően a fiatalok többsége a sportolást edzőtermekben valósítja meg, amivel semmi baj nem lenne, de sokan nincsenek tisztában vele, hogy hogyan is kell különböző gyakorlatokat szabályosan elvégezni. Ez számos veszélyt magában foglal. Fizikai sérüléseket, esetleg sérvek, valamint ízületi sérülések kialakulását, amik nem feltétlen azonnal, hanem az idő múlásával jelentkeznek. A regenerálódási idejük hosszú lefolyású, legtöbbször orvosi beavatkozás, műtétek szükségesek a teljes felépüléshez.

Ezért szakdolgozatom célja egy olyan Android alkalmazás fejlesztése, amely segítségével a felhasználók meg tudják tekinteni különböző izomcsoportokhoz tartozó gyakorlatokat, hogyan is kell szabályosan elvégezni. Továbbá saját edzéstervet tudnak maguknak létrehozni, valamint tisztában lesznek vele, hogy napi szinten, mennyi az ajánlott kalóriabevitel számukra.

A szakdolgozat ennek az Android alkalmazásnak a fejlesztését írja le. Az elkészült program lehetővé teszi felhasználók regisztrálását, bejelentkezését. Általam elkészített GIF-ek alapján lehet megtekinteni, hogyan kell szabályosan elvégezni a gyakorlatokat és ezeket a gyakorlatokat lehet hozzáadni saját edzéstervhez. A Node.js-ben írt szerver program a felhasználói adatokat és a gyakorlatokat egy adatbázisba küldi fel és ott kerülnek elmentésre.

Abstract

Nowadays, it is getting more and more important for people, to eat healthy and do some sports regularly. Noticeably the most of the youth choose the gym for doing exercises, which would not be a problem, but a lot of people do not know how to make various exercises properly. This contains several danger. Physical injuries, maybe development of joint diseases and hernia, which may not appear immediately only after a long time.The regeneration takes a considerable period of time, in the most cases medical interventions, operations are needed to the complete recovery.

That is why the aim of my examination paper is to develop an Android application which helps the users to view the exercises to the different muscle groups to be aware of how to make them properly. Furthermore, they can create their own training plan, moreover they will know how much is the perfect amount of calorie intake on a daily basis.

The examation paper describes the development of this Android app. The complete program allows the registration and login of the users. It can be viewed by gifs made by me how to do the exercises regularly and these exercises can be added to the own traning plan. The program on the server, written in the Node.js sends the user data and excersises to a database stored there.

# Bevezetés

A bevezetésben bemutatásra kerül a témaválasztásom indoka, hasonló alkalmazások, továbbá a szakdolgozat felépítése.

## Témaválasztás

Témaválasztásom során fontos szempont volt, hogy egy olyan témát találjak, ami hozzám közel áll és szívesen foglalkozok vele. Lassan második éve, hogy majdnem napi szinten konditerembe járok, ezért jött ez az ötlet, hogy egy ehhez kapcsolódó feladatot válasszak. Amikor elkezdtem edzőterembe járni én se voltam még tisztában nagyon sok dologgal, de vettem a fáradtságot és sok mindennek utána olvastam. Sokan nem szeretnek, vagy nem mernek kérdezni a tapasztaltabbaktól, mert kínosnak érzik, de ezzel kockára teszik az egészségüket, valamint fejlődni se fognak rendesen.

Nagy könnyítés lett volna számomra, ha lett volna egy ilyen alkalmazás telefonra. Napjainkban mobilkészülékkel szinte minden ember rendelkezik már, az életben szinte mindenhol segítségünkre lehet a telefonunk. Az utóbbi egy évben döntöttem el, hogy telefonra való fejlesztéssel szeretnék foglalkozni a jövőben. Egyre több program jelenik meg ezekre az eszközökre, valamint egyszerű kezelhetőség és könnyű hordozhatóság miatt ez a szám rohamosan nő, és nőni is fog.



1.1. ábra: Android és iOS logója

Mobileszközök gyors terjedésének köszönhetően, szoftverekre való igény is egyre inkább növekszik. Jelenleg két nagy mobil operációs rendszer van a piacon, ezek az iOS, valamint az Android.

## Hasonló megoldások

Mielőtt neki álltam volna fejleszteni a BuildBody nevű alkalmazásomat, utána jártam, hogy milyen hasonló mobilalkalmazások érhetőek el a felhasználók számára. A Play Áruházban már találhatóak hasonló alkalmazások, de szinte kivétel nélkül hiányérzetem volt, amikor kipróbáltam ezeket az alkalmazásokat.

A legnagyobb probléma az volt, hogy nem volt egy egységes alkalmazás, ami minden testrészhez tartalmazta volna a gyakorlatokat. Minden alkalmazás külön – külön egy testrészre volt szabva, vagyis ha valaki minden testrészhez szeretne segítséget igénybe venni, akkor legalább 8 alkalmazást kell feltennie a telefonjára alsó hangon.

|  |  |
| --- | --- |
|  |  |

**1.2 ábra: Hasonló alkalmazások a Play Áruházból**

Másik probléma amivel találkoztam az az, hogy ezek kész edzésterveket tartalmaznak, és nem lehet saját magunknak edzéstervet összeállítani a meglévő gyakorlatokból. Az ötletet, hogy vannak külön kezdőknek, haladóknak, és tapasztaltaknak is gyakorlat kifejezetten jónak tartom, de amikor megnéztem a gyakorlatokat igencsak csalódott voltam. Nem igazán tükrözte a különbséget a kategóriák között. Legtöbb helyen csak az ismétlésszámban tértek el egymástól, valamint nem tartalmaztak súlyzókkal való edzéshez tanácsokat, csak szabad súllyal történő gyakorlatokat mutattak be.

## A szakdolgozat felépítése

A következő fejezetekben ismertetni fogom az elkészült alkalmazást, valamint a funkciók részletes leírását, néhány helyen folyamatdiagramokkal segítve a megértést. Továbbá kitérek a felhasznált technológiákra, a MySQL adatbázisra, a Szerveralkalmazásra, valamint az Android alkalmazás elkészítésének a folyamatára, ez magában foglalja az architektúra bemutatását is. A végén pedig a kész alkalmazás használata kerül bemutatásra, és az alkalmazás tesztelésre is kitérek.

# Alkalmazás ismertetése

Ebben a fejezetben, a szakdolgozat alatt elkészített feladatnak, a részletes specifikációját fogom ismertetni. Bemutatásra kerülnek továbbá, a felhasználók és az alkalmazástól elvárt funkciók is, valamint a napi kalóriaszükségletnek a számítási módját is ismertetni fogom.

## Feladatspecifikáció

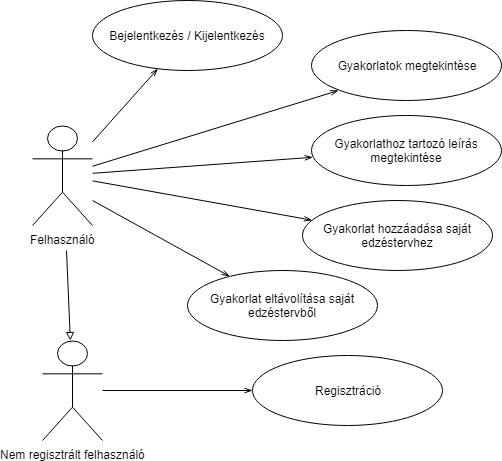
A feladat egy Android alkalmazás elkészítése, amelyben a felhasználók meg tudják tekinteni, hogy hogyan is tudják szabályosan elvégezni, különböző testrészekhez tartozó gyakorlatokat. Ezeket a gyakorlatokat a felhasználók hozzá tudják adni saját edzéstervükhöz, ahol már csak az általuk kiválasztott gyakorlatok jelennek meg a különböző testrészekhez.

Használat előtt a felhasználóknak regisztrálniuk kell a mobilalkalmazásban, ezt követően pedig be kell jelentkezniük. A felhasználók által megadott adatokat, a gyakorlatokat, valamint a felhasználók által kiválasztott gyakorlatok MySQL adatbázisban kerültek eltárolásra. A gyakorlatok bemutatása GIF-ek formájában valósul meg, valamint rövid leírást is olvashatnak, hogy hogyan is kell szabályosan kivitelezni a gyakorlatot. A szükséges GIF fájlok tárolása a szerver egy dedikált könyvtárában valósul meg.

## Felhasználók

Az elkészült alkalmazásban minden felhasználó ugyanolyan hatáskörrel rendelkezik, mint mások. Nem készült külön felhasználói csoport olyan felhasználóknak, akik esetleg már tapasztaltabbak az edzések terén, és esetleg szeretnének ők is, hasznos tanácsokat nyújtani az újoncok számára. Az alkalmazásban ennek ellenére, mégis két felhasználói csoport különíthető el:

* Nem regisztrált felhasználó
* Felhasználó



2.1. ábra: Use case diagram

### Felhasználó

A felhasználónak tekintjük azt a személyt, aki már regisztrált az alkalmazásba. Számára lehetőség van az alkalmazás minden funkcióját kihasználni.

* Bejelentkezés / Kijelentkezés
* Gyakorlatok megtekintése
* Gyakorlatokhoz tartozó leírás megtekintése
* Gyakorlat hozzáadása saját edzéstervhez
* Gyakorlat eltávolítása saját edzéstervből

### Nem regisztrált felhasználó

A nem regisztrált felhasználó kizárólag a regisztráció folyamatát tudja elindítani, máshoz nem fér hozzá. Az adatok kitöltése, és a regisztráció gomb megnyomása után az alábbi lehetőségek következhetnek be:

* Sikeres regisztráció, visszakerülünk a bejelentkező felületre
* Sikertelen regisztráció
  + A felhasználó már létezik
  + Valamelyik adat, esetleg adatok nem megfelelően lettek kitöltve

A felhasználó létezésének tényleges jelentése az, hogy ez az email cím már szerepel az adatbázisunkban, vagyis már regisztráltak vele.

## Funkciók

A legfontosabb funkciók, amelyek a felhasználók és a nem regisztrált felhasználók tudnak használni:

* Regisztráció
* Bejelentkezés / Kijelentkezés
* Gyakorlatok megtekintése
* Gyakorlathoz tartozó leírás megtekintése
* Gyakorlatok hozzáadása saját edzéstervhez
* Gyakorlat eltávolítása saját edzéstervből

Néhány bonyolultabb funkcióhoz készítettem folyamatdiagramot is a könnyebb érthetőség kedvéért. A különböző diagramokat egy online tool [1] segítségével készítettem el.

### Regisztráció

A 2.3-as ábrán a regisztráció folyamatát tekinthetjük meg. A felhasználó letöltötte és sikeresen telepítette a mobilalkalmazást. Az alkalmazás indítása során a felhasználónk még a nem regisztrált felhasználók közé tartozik, vagyis csak a regisztráció funkciót érheti el. Regisztráláskor számos adatot kell megadnunk, amelyek a következők:

* Név
* Email
* Jelszó
* Nem
* Aktivitási szint
* Életkor
* Súly
* Magasság

A felhasználó által megadott adatokat, amiket képes, a mobil alkalmazás fogja verifikálni. Lehetséges hibaesetek lehetnek a következők:

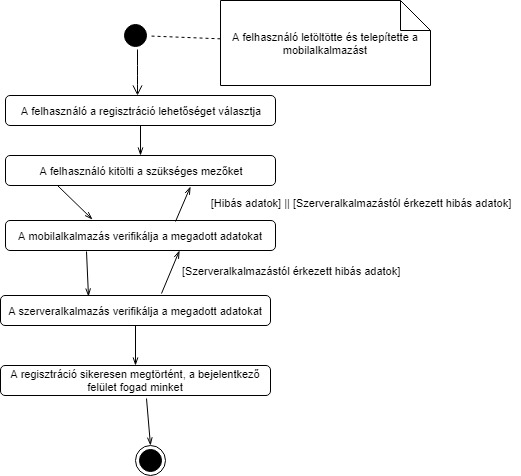
* Nem töltötte ki az adott mezőt
* Nem megfelelő email formátum
* Nem egész számban adta meg az adott mező értékét

A nem egész számban adta meg az adott mező értékét hibaüzenet akkor jelentkezik, amikor az életkor, súly, vagy magasságot töltjük ki, ugyanis itt egész számot vár el az alkalmazás. Ez azért van, mert a MySQL adatbázisban létrehozott táblában, ezeket a mezőket Integerként definiáltam, azaz egész számként várja el az adatokat. Az Aktivitási szint mező egy kis magyarázatra szorul, hogy mit is jelent valójában. Három lehetőség közül választhatunk:

* Könnyű (Napi 2 óra mozgásnál kevesebbet végzünk)
* Közepes (Napi 2-4 óra mozgást végzünk)
* Nehéz (Napi 4 óránál több mozgást végzünk)

A Nem, Aktivitási szint, Életkor, Súly, Magasság megadására azért van szükség, mert ezekkel a paraméterekkel számítódik ki, a napi ajánlott kalóriamennyiség bevitele. A Szerveralkalmazás a magadott adatokon olyan verifikálásokat végez, amelyek mobilalkalmazáson keresztül nem lehetségesek. Ide tartozik például:

* A megadott email cím már foglalt



2.2. ábra: A regisztráció folyamat Activity diagramja

### Bejelentkezés

A 2.4-es ábrán a bejelentkezés folyamatát tekinthetjük meg. A felhasználó sikeres regisztrációt követően megpróbál bejelentkezni az alkalmazásba. A mobilalkalmazás és a Szerveralkalmazás közösen eldöntik az elején, hogy rendelkezik-e a felhasználó érvényes tokennel. Két eset lehetséges:

* A felhasználó tokenje lejárt
* A felhasználó érvényes tokennel rendelkezik

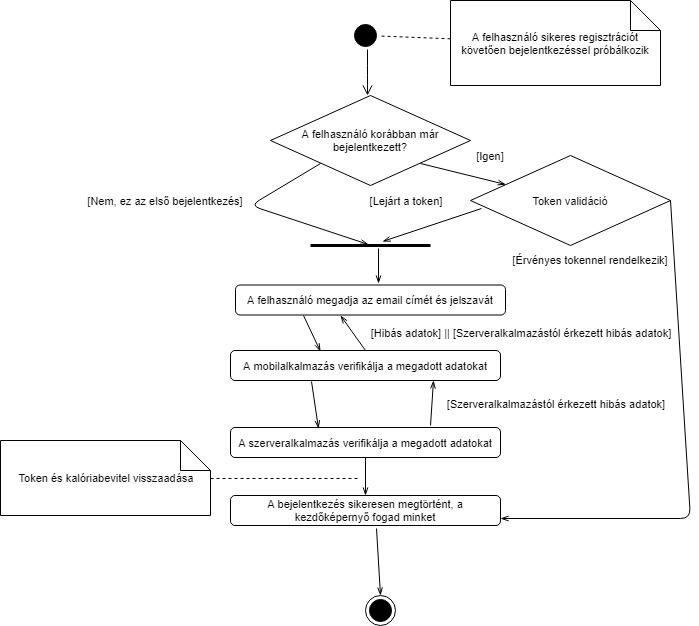
Amennyiben lejárt a felhasználó tokenje, akkor olyan, mint ha most először szeretne bejelentkezni, vagyis a bejelentkező felület fogja őt fogadni. Itt a felhasználó email cím és hozzá tartozó jelszó párossal fogja azonosítani magát. A megadott adatokat, amiket képes, a mobil alkalmazás fogja verifikálni. Lehetséges hibaesetek lehetnek a következők:

* Nem töltötte ki az email cím mezőt
* Nem töltötte ki a jelszó mezőt
* Nem megfelelő email formátum

A Szerveralkalmazás a magadott adatokon olyan verifikálásokat végez, amelyek mobilalkalmazáson keresztül nem lehetségesek. Ide tartoznak például:

* Nincs ilyen email cím
* Nem jó az email – jelszó páros

Ha a felhasználó érvényes tokennel rendelkezik, valamint ha sikeresen megadta az email cím – jelszó párost, akkor a kezdőképernyő fogad minket. Ebben az esetben a szervertől válaszként a token és a napi szinten szükséges kalóriabevitelt kapjuk vissza.



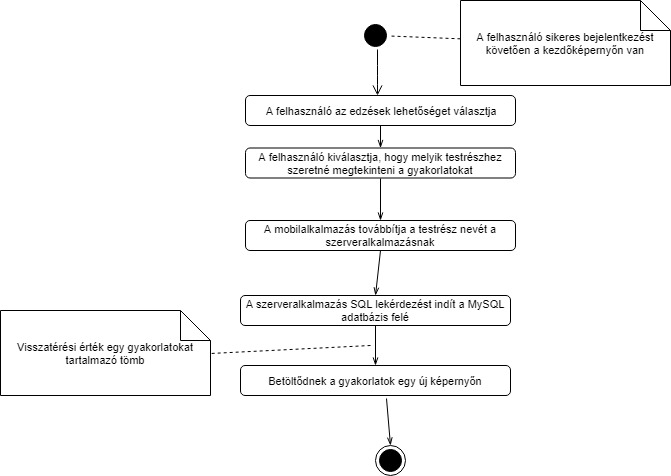
2.3. ábra: A bejelentkezés folyamat Activity diagramja

### Gyakorlatok megjelenítése

A 2.5-ös ábrán a gyakorlatok megjelenítését tekinthetjük meg. A felhasználót sikeres bejelentkezést követően a kezdőképernyő fogadja. Itt két lehetőség közül választhat, de jelen részben csak az edzések lehetőséget fejtem ki. Miután kiválasztásra került az edzések opció láthatóak, hogy milyen testrészekhez vannak gyakorlatok az alkalmazásban. Ezek a testrészek a következők:

* Mell
* Hát
* Comb
* Vádli
* Bicepsz
* Tricepsz
* Váll
* Has

A felhasználó választása után, a testrész nevét küldi el a mobilalkalmazás a Szerveralkalmazás felé. A Szerveralkalmazás egy SQL lekérdezést indít a MySQL adatbázis felé, ahol feltételnek a testrész nevét adja meg. Visszatérési érték a Szerveralkalmazástól egy gyakorlatokat tartalmazó tömb lesz. Ezeket a gyakorlatokat fogja megjeleníteni egy új képernyőn a mobilalkalmazás.



2.4. ábra: A gyakorlatok megjelenítésének Activity diagramja

## Kalóriaigény számítás

A napi szükséges kalóriabevitel kiszámításához a Mifflin-ST. Jeor képletet [2] használtam fel. A kalóriaszükségletnek a mértékegysége a KJ (kilójoule) pont, mint az energiának. Az emberi szervezet az energiát a bevitt tápanyagokból állítja elő. A kalóriaszámítás nemtől függően a következőképpen alakulhat:

* Férfiak esetén: (10 \* súly) + (6.25 \* magasság) – (5 \* életkor) + 5
* Nők esetén: (10 \* súly) + (6.25 \* magasság) – (5 \* életkor) – 161

Aktivitási szinttől függően a fent kapott értéket meg kell szorozni még a következőképpen:

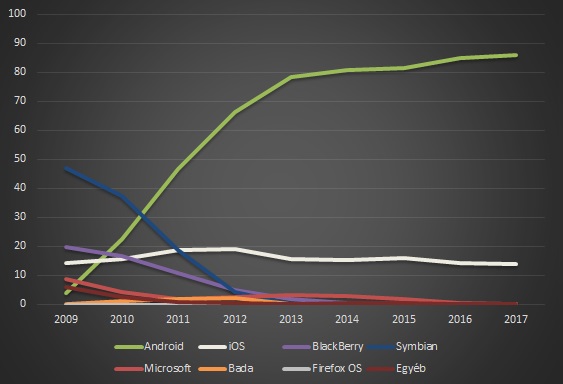
|  |  |
| --- | --- |
| **Aktivitási szint** | **Szorzó** |
| Könnyű | 1,4 |
| Közepes | 1,7 |
| Nehéz | 1,9 |

# Felhasznált technológiák

A következőben azokat a technológiákat mutatom be, amelyeket felhasználtam az alkalmazás elkészítése során.

## Android platform

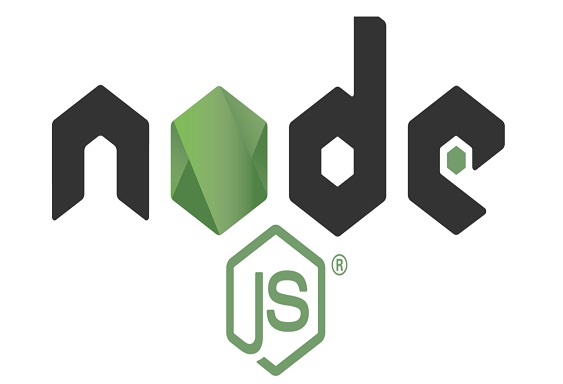
Az Android alapja egy Linux operációs rendszer, amelyet átalakítottak úgy, hogy az eszközök integrált hardvereit megfelelően tudja kezelni. Az Android platform megalkotásának célja az volt, hogy egy nyílt forráskódú, rugalmas, könnyen alakítható rendszer legyen, amelyre könnyű külső alkalmazásokat fejleszteni. 2005-ben felvásárlásra került az Android Incorporated nevű kaliforniai cég, az „IT óriás” a Google által. 2007 elején kezdtek kiszivárogni olyan hírek, hogy a Google belép a mobil piacra és 2007. november 5-én az Open Handset Alliance bejelentette az Android platformot. Az első készülék a T-Mobile által forgalmazott, HTC G1-es készülék volt. Szakdolgozatom elkészítésekor az is az Android platform mellett szólt, hogy továbbra is a piaci részesedést hatalmas mértékben vezeti.



3.1. ábra: Mobilkészülékek piaci részesedésének az eloszlása [3]

## Node.js

Skálázható internetes alkalmazások, méghozzá webszerverek készítésére hozták létre a Node.js [4] szoftverrendszert, amely lehetővé teszi JavaScript futtatását szerveroldalon is. A Node.js a Google-féle V8 JavaScript-motorból és számos beépített könyvtárból tevődik össze. Nem blokkoló I/O hívásoknak köszönhetően nagyon gyors, valamint sok hívás esetén is biztosítja, hogy az általunk használt szál ne legyen túlterhelt.



3.2. ábra: Node.js logója

Számomra ez megfelelő funkciókat látott el, ugyanis a nem blokkoló műveletek közé tartoznak olyanok, mint például:

* Fájlműveletek
* Adatbázis műveletek

Ezekre szükségem volt a szakdolgozatom elkészítése során, ugyanis sok adatott kellett kiszolgálnom.

## Express

Az Express [5] egy nyílt forráskódú webalkalmazás keretrendszer a Node.js-hez. Felhasználásával sokkal könnyebben lehet a webalkalmazásunk kódját karbantartani és strukturálni.

## MySQL

A MySQL egy relációs adatbáziskezelő rendszer, ahol az adatokat az adatbázis tábláiban tároljuk és ezek az eltárolt adatok innentől kezdve, az adatbázisnak egy-egy rekordjának felelnek meg. SQL utasítások segítségével tudjuk az adatokat kezelni. Ilyen utasítások lehetnek például:

* Lekérdezések
* Adatok bevitele
* Adatok módosítása
* Adatok törlése

### Lekérdezések

SELECT (oszlopok)

FROM (táblák)

[WHERE (feltételek)]

[(csoportosítás)]

[(rendezés)];

A lekérdezés eredménye szintén egy táblát állít elő. Az (oszlopok) határozzák meg az eredménytábla oszlopait. A (táblák) adják a lekérdezésben résztvevő táblákat. A (feltételek) segítségével választhatjuk ki a számunkra szükséges rekordokat. A (csoportosítás) az eredménytábla sorait rendezi. A (rendezés) a megjelenítés sorrendjét befolyásolja.

### Adatok bevitele

INSERT INTO (táblanév) [((oszlopnév) [, (oszlopnév), …])]

VALUES ((kifejezés1) [, (kifejezés2), …]);

Adatok bevitelekor, ha megadunk oszlopneveket, akkor minden megadott mezőnek értéket kell adni, a többi mező null értéket fog felvenni. Ha nem adunk meg oszlopneveket, akkor minden mezőnek értéket kell adni, különben parancsvégrehajtás után hiba üzenetet fogunk kapni.

### Adatok módosítása

UPDATE (táblanév)

SET (oszlopnév) = (kifejezés) [, (oszlopnév) = (kifejezés2) , …]

[WHERE (feltételek)];

Amennyiben a WHERE feltételünk hiányozna, úgy az összes rekordot, ellenkező esetben csak a feltételeket teljesítő rekordok fognak módosulni az adatbázisunkban.

### Adatok törlése

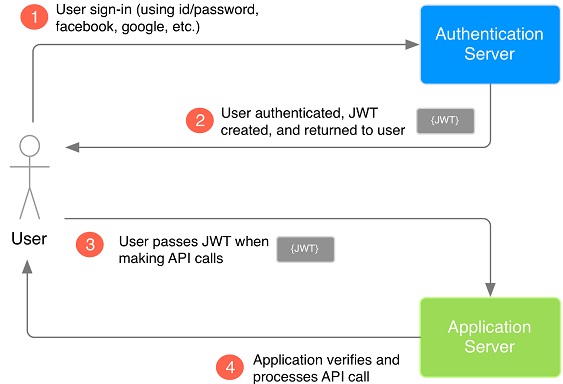
DELETE FROM (táblanév)

[WHERE (feltételek)];

Amennyiben a WHERE feltételünk hiányozna, úgy az összes rekordot, ellenkező esetben csak a feltételeket teljesítő rekordok fognak törlődni az adatbázisunkból.

## JSON Web Token

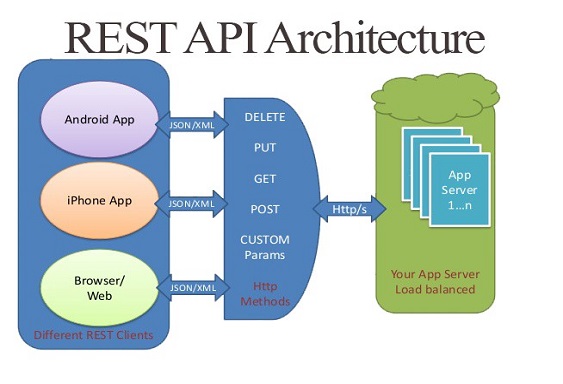
A JSON Web Token (JWT) [6] egy JSON objektum, amelyet az RFC 7519 szabvány definiál. A Token 3 fő részből tevődik össze, mégpedig fejléc, törzs és aláírás. Egy ilyen Token segítségével biztonságosan megvalósítható az információcsere két fél között, valamint bizonyos funkciók engedélyezése. Engedélyezésre egy kiváló példa a felhasználói bejelentkezés. Miután a felhasználó bejelentkezett, onnantól kezdve nem szükséges a számára, hogy Id és Jelszó alapján azonosítsa magát a szerver felé. Ha valami kérést indítana elegendő, ha a Token-t tartalmazza a kérés, ez alapján egyértelműen azonosítani lehet a felhasználókat. A Token rendelkezik még egy lejárati idővel, ami a Token érvényességének a dátumát fejezi ki.



3.3. ábra: JSON Web Token működése

## REST

A REST (Representational State Transfer) [7] egy szoftverarchitektúra típus nagyobb internet alapú rendszerek számára. Az ilyen típusú architektúra szerverekből, valamint kliensekből épül fel. A kliensek különböző kéréseket tudnak küldeni a szerverek felé, amik a kéréseket feldolgozzák, és a választ továbbítják annak a kliensnek, aki a kérést indította.



RESTful rendszernek nevezzük azokat a rendszereket, amelyek eleget tesznek a REST szabályainak. A szabályok a következők:

* Állapotmentesség
* Kliens – Szerver architektúra
* Réteges felépítés
* Gyorsítótárazhatóság
* Egységes interfész
* Igényelt kód (opcionális)

# Architektúra bemutatása

Ebben a fejezetben az elkészült alkalmazás architektúrájának a felépítése kerül bemutatásra.

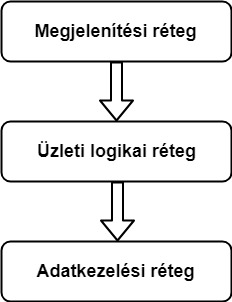
## Háromrétegű architektúra

A többrétegű architektúrák a szoftverfejlesztésben előszeretettel alkalmazottak, ebben az esetben a teljes rendszerünk több különálló részből tevődik össze. A különálló rétegeknek köszönhetően számos előnyt biztosít számunkra más tervezési módszerekkel szemben, ilyenek például:

* Egyszerűbb tovább fejlesztési lehetőség biztosítása
* Rétegek fejlesztése egymástól függetlenül is megvalósítható
* Egy adott réteg akár teljes egészében kicserélhetővé válik
* Lehetővé teszi a kód könnyebb karbantarthatóságát

Ezen előnyöket figyelembe véve döntöttem úgy, hogy többrétegű architektúrát fogok használni a fejlesztésem során. Egyik legnépszerűbb a háromrétegű architektúra, amire pont illik az elkészített rendszerem. A háromrétegű architektúra rétegei:

* Megjelenítési réteg
* Üzleti logikai réteg
* Adatkezelési réteg



4.1. ábra: Háromrétegű architektúra felépítése

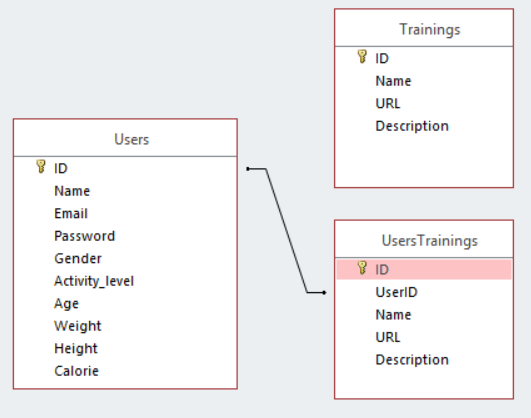
Amint a képen is látható nincs minden réteg összekötve a másikkal. Egy ilyen architektúrával készített rendszerben, a rétegek kizárólag a szomszédos rétegekkel képesek kommunikálni. Megjelenítési rétegnek az elkészült rendszeremben a mobilalkalmazás, Üzleti logikai rétegnek a szerveralkalmazás, Adatkezelési rétegnek pedig, a MySQL adatbázis felel meg.

# Adatbázis

Ebben a fejezetben a MySQL adatbázisban elkészített táblákat fogom bemutatni, valamint a táblák között lévő kapcsolatokat.

## Kapcsolatok a táblák között

Amint az 5.1-es ábrán látható, a Users tábla és a Trainings tábla között nincs semmilyen kapcsolat, azonban a UsersTrainings tábla, ennek a két táblának bizonyos adatainak az együtteséből tevődik össze.



5.1. ábra: Táblák közötti kapcsolat megjelenítése

## Users tábla

Ebben a táblában a felhasználók adatait tárolom el, amiket a mobilalkalmazásban való regisztrációkor adnak meg. Az adatok típusáról, valamint további fontos tulajdonságaikról, a lentebb található táblázat nyújt segítséget.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Oszlop név** | **Típus** | **Kulcs** | **Üresen hagyható** | **Automatikus kitöltés** |
| ID | INT(11) | Igen | Nem | Igen |
| Név | VARCHAR(45) | Nem | Nem | Nem |
| Email | VARCHAR(45) | Nem | Nem | Nem |
| Jelszó | VARCHAR(125) | Nem | Nem | Nem |
| Nem | VARCHAR(45) | Nem | Nem | Nem |
| Aktivitási szint | VARCHAR(45) | Nem | Nem | Nem |
| Életkor | INT(11) | Nem | Nem | Nem |
| Súly | INT(11) | Nem | Nem | Nem |
| Magasság | INT(11) | Nem | Nem | Nem |
| Kalória | DOUBLE | Nem | Igen | Nem |

## Trainings tábla

Ebben a táblában a felhasználók által megtekinthető gyakorlatokat tárolom el. A gyakorlatok GIF-ek formájában tekinthetőek meg a mobilalkalmazásban. Ezeken a GIF-ken én szerepelek, én mutatom be a gyakorlatok szabályos kivitelezését.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Oszlop név** | **Típus** | **Kulcs** | **Üresen hagyható** | **Automatikus kitöltés** |
| ID | INT(11) | Igen | Nem | Igen |
| Név | VARCHAR(45) | Nem | Nem | Nem |
| URL | VARCHAR(250) | Nem | Nem | Nem |
| Leírás | VARCHAR(500) | Nem | Nem | Nem |

## UsersTrainings tábla

Ebben a táblában a felhasználók által kiválasztott gyakorlatokat tárolom el. Ezeket a gyakorlatokat a felhasználó a mobilalkalmazásban, saját edzésterveinél tudja megtekinteni.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Oszlop név** | **Típus** | **Kulcs** | **Üresen hagyható** | **Automatikus kitöltés** |
| ID | INT(11) | Igen | Nem | Igen |
| UserID | INT(11) | Nem | Nem | Nem |
| Név | VARCHAR(45) | Nem | Nem | Nem |
| URL | VARCHAR(250) | Nem | Nem | Nem |
| Leírás | VARCHAR(500) | Nem | Nem | Nem |

A UsersID attribútum egy külső kulcsnak felel meg, ami a felhasználót azonosítja. Ennek az attribútumnak a segítségével tudhatjuk, hogy adott felhasználó, milyen gyakorlatokat választott ki, amiket szeretne a saját edzéstervéhez adni.

# Szerveralkalmazás

Ebben a fejezetben a Node.js-ben elkészített szerveralkalmazás felépítését, valamint fontosabb részeinek a működését fogom bemutatni.

## Felépítés

A Szerveralkalmazás az alábbi komponensekből tevődik össze:

* Controllers
* Counters
* Database
* Gifs
* Middleware
  + Validators

## Controllers

A MySQL adatbázisban szereplő adatok lekérdezését, módosítását, valamint új adat felvételét teszik lehetővé. A szerveralkalmazás 3 ilyen kontrollerrel rendelkezik:

* UserController
* TrainingController
* UsersTrainingsController

### UserController

A UserController osztálynak a feladata a felhasználók kezelése, ezért a Users táblához fér hozzá. Két fontos funkciót szeretnék itt megemlíteni, mégpedig a regisztrációt, és a bejelentkezést.

### TrainingController

A TrainingController osztálynak a feladata a gyakorlatok kezelése, ezért a Trainings táblához fér hozzá. Legfontosabb funkciója egy lekérdezés, ami egy gyakorlatokat tartalmazó tömböt ad vissza. Ennek alapján a mobilalkalmazásban betöltődnek az adott testrészhez megfelelő gyakorlatok.

### UsersTrainingsController

A UsersTrainingsController osztálynak a feladata a felhasználó által kiválasztott gyakorlatok kezelése, ezért a UsersTrainings táblához fér hozzá. Két fontos funkciót szeretnék itt megemlíteni, mégpedig a gyakorlatok felvételét, és a gyakorlatok eltávolítását.

## Counters

Jelenleg az alkalmazás kizárólag a kalóriaszámításhoz tartalmaz egy számlálót. A 2.4-es résznél már ismertettem ennek a folyamatnak a számítási menetét, itt csak az implementációja valósult meg a képleteknek.

## Database

const mysqlConnection = mysql.createConnection({

host: ’localhost’,

user: ’root’,

password: ’Administrator’

database: ’BuildBody’

multipleStatements: true

});

A MySQL Adatbázishoz való csatlakozást biztosítja, méghozzá adminisztrátori jogosultságokkal.

## Gifs

Ez a mappa tartalmazza a gyakorlatokat GIF fájlok formájában. Ennek a mappának az elérési útvonala, és adott gyakorlathoz tartozó GIF neve tárolódik el a Trainings, és a UsersTrainings táblák URL attribútumában.

## Middleware

Jelenleg a Szerveralkalmazás csak a LoginMiddleware-rel rendelkezik. Ennek az osztálynak a feladata az alkalmazás indulásakor a token validációja. A validáció három fő részre osztható:

* Rendelkezésre áll a token
* Lejárt a token
* Módosult a token

Amennyiben ezek a feltételek teljesülnek, a bejelentkezés sikeresen megtörténik.

## Validators

A validátorok feladata az adatbázisba történő adatok felvételekor ellenőrizni, hogy a következő megkötések teljesülnek-e:

* Ki van töltve minden szükséges mező
* Attribútum által elvárt típusban lettek kitöltve
* Nem szerepel még az adatbázisban

Amennyiben ezek a feltételek teljesülnek, az új adat sikeresen bekerül a MySQL adatbázisba. Jelenleg a Szerveralkalmazás három ilyen validátorral rendelkezik, amelyek a következők:

* UserRegisterValidator
* TrainingRegisterValidator
* UsersTrainingsRegisterValidator

# Mobilalkalmazás

Ebben a fejezetben az Android alkalmazás felépítését, valamint fontosabb részeinek a működését fogom bemutatni.

## Felépítés

Az Android alkalmazás az alábbi komponensekből tevődik össze:

* Activitys
* Adapters
* Application
* Fragments
* UserInformation
* RecyclerViewElements

# Mobilalkalmazás használatának a bemutatása

# Tesztelés

# Összefoglalás

# Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | „Diagramok készítése,” [Online]. Available: https://www.draw.io/. [Hozzáférés dátuma: 26 november 2018]. |
| [2] | „Kalória számítás,” [Online]. Available: Mifflin-St.Jeor képlet. [Hozzáférés dátuma: 26 november 2018]. |
| [3] | H. Ferenc, „Mobilkészülékek piaci részesedésének az eloszlása,” [Online]. Available: https://www.hwsw.hu/hirek/58482/android-ios-okostelefon-platform-gartner.html. [Hozzáférés dátuma: 19 november 2018]. |
| [4] | „Node.js,” [Online]. Available: https://nodejs.org/en/about/. [Hozzáférés dátuma: 20 november 2018]. |
| [5] | „Express,” [Online]. Available: https://expressjs.com/. [Hozzáférés dátuma: 20 november 2018]. |
| [6] | „JSON Web Token működése,” [Online]. Available: https://medium.com/vandium-software/5-easy-steps-to-understanding-json-web-tokens-jwt-1164c0adfcec. [Hozzáférés dátuma: 20 november 2018]. |
| [7] | „REST,” [Online]. Available: https://hu.wikipedia.org/wiki/REST. [Hozzáférés dátuma: 20 november 2018]. |