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

Automatizálási és Alkalmazott Informatikai Tanszék

Fazekas Bence

Android alapú testedzést támogató alkalmazás fejlesztése

Konzulens

Dr. Ekler Péter

BUDAPEST, 2018

Tartalomjegyzék

[Összefoglaló 5](#_Toc530575359)

[Abstract 6](#_Toc530575360)

[1 Bevezetés 7](#_Toc530575361)

[1.1 Témaválasztás 7](#_Toc530575362)

[1.2 Felhasznált technológiák 7](#_Toc530575363)

[1.2.1 Android platform 7](#_Toc530575364)

[1.2.2 Node.js 8](#_Toc530575365)

[1.2.3 Express 9](#_Toc530575366)

[1.2.4 MySQL 9](#_Toc530575367)

[1.2.5 JSON Web Token 11](#_Toc530575368)

[1.2.6 REST 11](#_Toc530575369)

[1.3 A szakdolgozat felépítése 12](#_Toc530575370)

[2 Feladatspecifikáció 13](#_Toc530575371)

[3 Irodalomjegyzék 14](#_Toc530575372)

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 kialakulását, 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

Ide jön a ½-1 oldalas angol nyelvű összefoglaló, amelynek szövege a Diplomaterv Portálra külön is feltöltésre kerül. Ez a magyar nyelvű összefoglaló angolra fordított változata.

# Bevezetés

A bevezetésben bemutatásra kerül a témaválasztásom indoka, a fejlesztés során megismert, valamint felhasznált technológiák részletezése, 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 mobiltelefonnal szinte minden ember rendelkezik. Mostanra 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.

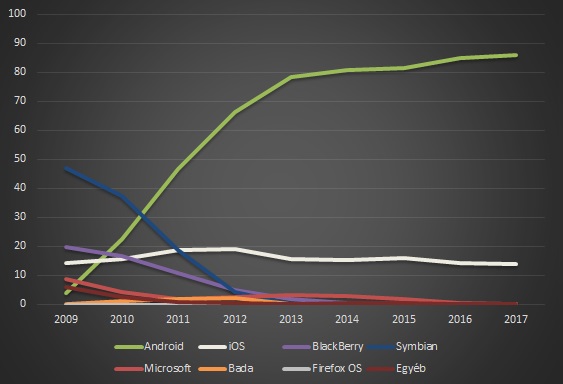
Mobileszközök gyors terjedésének köszönhetően, szoftverekre való igény is egyre inkább növekszik. Jelenleg két nagy mobil operációs rendszer van a piacon, ezek az iOS, valamint az Android.

## Felhasznált technológiák

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

### Android platform

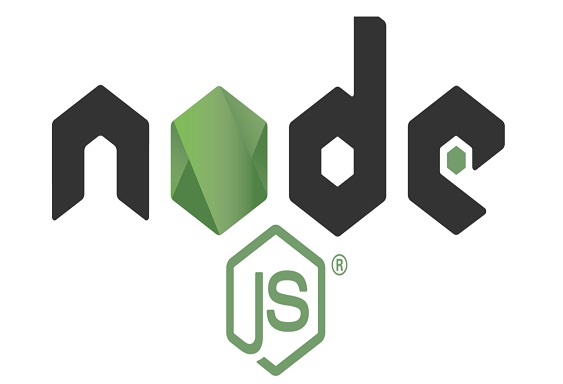
Az Android alapja egy Linux operációs rendszer, amelyet átalakítottak úgy, hogy az eszközök integrált hardvereit megfelelően tudja kezelni. Az Android platform megalkotásának célja az volt, hogy egy nyílt forráskódú, rugalmas, könnyen alakítható rendszer legyen, amelyre könnyű külső alkalmazásokat fejleszteni. 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.



1.1. ábra: Mobilkészülékek piaci részesedésének az eloszlása [1]

### Node.js

Skálázható internetes alkalmazások, méghozzá webszerverek készítésére hozták létre a Node.js [2] 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.



1.2. 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 [3] 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 <logikai kifejezések>]

[<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 <logikai kifejezések> 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 <logikai kifejezés>];

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

#### Adatok törlése

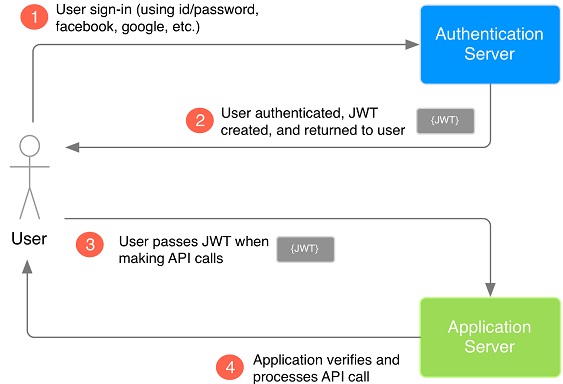
DELETE FROM <táblanév>

[WHERE <logikai kifejezések>];

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

### JSON Web Token

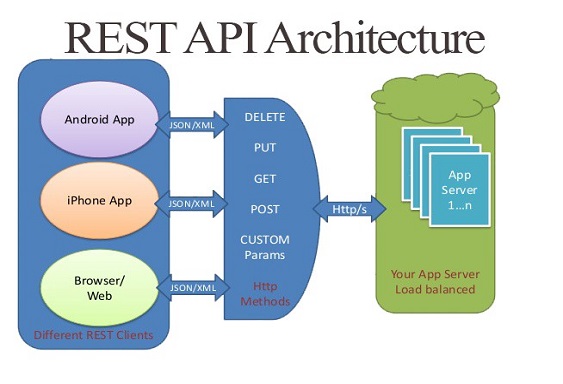
A JSON Web Token (JWT) [4] egy JSON objektum, amelyet az RFC 7519 szabvány definiál. A Token 3 fő részből tevődik össze, mégpedig fejléc, törzs és aláírás. Egy ilyen Token segítségével biztonságosan megvalósítható az információcsere két fél között, valamint bizonyos funkciók engedélyezése. Engedélyezésre 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.



1.3. ábra: JSON Web Token működése

### REST

A REST (Representational State Transfer) [5] 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)

## A szakdolgozat felépítése

A következő fejezeketben az Android alkalmazás, a szerveralkalmazás, valamint a MySQL adatbázis elkészítésének a folyamatát fogom bemutatni. Ez magában foglalja az architektúra bemutatását, valamint a funkciók részletes leírását, néhány helyen folyamatdiagrammal segítve a megértést. A végén pedig bemutatom a kész alkalmazást, annak használatát, továbbá a tesztelésre is kitérek.

# Feladatspecifikáció

# Irodalomjegyzék

|  |  |
| --- | --- |
| [1] | 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]. |
| [2] | „Node.js,” [Online]. Available: https://nodejs.org/en/about/. [Hozzáférés dátuma: 20 november 2018]. |
| [3] | „Express,” [Online]. Available: https://expressjs.com/. [Hozzáférés dátuma: 20 november 2018]. |
| [4] | „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]. |
| [5] | „REST,” [Online]. Available: https://hu.wikipedia.org/wiki/REST. [Hozzáférés dátuma: 20 november 2018]. |