



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**



SZAKDOLGOZAT

**OE-NIK
2022**

Hallgató neve:
Hallgató törzskönyvi száma:

**Torma László Márk
T/005844/FI12904/N**

SZAKDOLGOZAT FELADATLAP

Hallgató neve:	Torma László Márk
Törzskönyvi száma:	T/005844/FI12904/N
Neptun kódja:	

A dolgozat címe:

Okosóráról adatok lekérdezése és feldolgozása okostelefon segítségével
Data acquisition from smartwatch with smartphone

Intézményi konzulens:	Lovas István
Külső konzulens:	

Beadási határidő:	2021. december 15.
-------------------	--------------------

A záróvizsga tárgyai:	Számítógép architektúrák IoT, beágyazott rendszerek és robotika specializáció
-----------------------	---

A feladat

A feladat, egy olyan android telefonos alkalmazás készítése, amely képes egy okosóráról a különböző szenzorok mérési adatait lekérni és feldolgozni. A program a begyűjtött adatokat legyen képes egy online adatbázis egységbe feltölteni. Lehesse az alkalmazáson belül több felhasználói profilt kezelni és ez alapján megoldani az adatokhoz való hozzáférést. A tervezés során vizsgálja meg a már meglévő rendszereket, biztonsági szempontokat és hasonlítsa össze őket. Készítse el az alkalmazást, és tesztelje a működést.

A dolgozatnak tartalmaznia kell:

- a megoldandó feladat pontos leírását és részletes elemzését,
- a meglévő rendszerek vizsgálatának elemzését,
- a hitelesítési és biztonsági szempontok elemzését,
- az alkalmazni kívánt technológiák elemzését,
- a rendszertervet, döntések indoklásait,
- a megvalósítás leírását,
- a tesztelési tervet és a tesztek eredményeit.



Fléner Rita

Dr. Fleiner Rita
intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

[Signature]
.....
intézményi konzulens



HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 20 22.05.13.

hallgató aláírása



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Torma László Márk Nappali
Telefon: Levelezési cím (pl: lakcím):

.....

Szakdolgozat / Diplomamunka címe magyarul:

Okosóráról adatok lekérdezése és feldolgozása okostelefon segítségével

Szakdolgozat / Diplomamunka címe angolul:

Data acquisition from smartwatch with smartphone

Intézményi konzulens: Külső konzulens:
Lovas István.....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

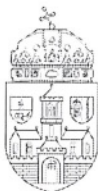
Alk.	Dátum	Tartalom	Aláírás
1.	2021.02.10	Témaválasztási konzultáció	
2.	2021.03.13	Szakirodalom áttekintése	
3.	2021.04.13	Lehetséges megvalósítás áttekintése	
4.	2021.05.04	Tartalmi, formai ellenőrzés	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4 tantárgy követelményét teljesítette, beszámolóra / védésre bocsátható.

Budapest, 2021. 05. 12.


.....
intézményi konzulens



KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:
Torma László Márk
Telefon: Levelezési cím (pl: lakcím):
.....

Szakdolgozat / Diplomamunka címe magyarul:

Okosóráról adatok lekérdezése és feldolgozása okostelefon segítségével

Szakdolgozat / Diplomamunka címe angolul:

Data acquisition from smartwatch with smartphone

Intézményi konzulens: Külső konzulens:
Lovas István.....

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.02.16	Specifikáció áttekintése	
2.	2022.03.09	Rendszerterv áttekintése	
3.	2022.04.06	Fejlesztés, tesztelés ellenőrzés	
4.	2022.05.04	Tartalmi, formai ellenőrzés	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplomamunka 2 / Diplomamunka 3 / Diplomamunka 4 tantárgy követelményét teljesítette, beszámolóra / védésre bocsátható.

Budapest, 2022. 05. 11.

.....
Intézményi konzulens

TARTALOMJEGYZÉK

1.	BEVEZETÉS.....	4
2.	AZ IRODALOM ALAPJÁN A LEHETSÉGES MEGKÖZELÍTÉSI MÓDOK ÉS MEGOLDÁSOK ÁTTEKINTÉSE ÉS ELEMZÉSE	5
2.1.	Fitness alkalmazások összehasonlítása	5
2.1.1.	Garmin Connect Mobile	6
2.1.2.	Google Fit	7
2.1.3.	Strava.....	9
2.1.4.	Apple Health	11
2.1.5.	Samsung Health.....	14
2.1.6.	Fitbit App	16
2.2.	Felhőben való tárolási megoldások összehasonlítása.....	18
2.3.	Autentikációs megoldások összehasonlítása	20
2.3.1.	Azonosítási módszerek osztályozása.....	20
2.3.2.	Azonosítási módszerek csoportosítása	20
2.3.3.	Hitelesítési módszerek alapján való csoportosítás:.....	21
2.4.	Szolgáltató és fogyasztó alkalmazás.....	22
2.4.1.	Alkalmazás szerepkörök összehasonlítása	22
2.5.	Android fejlesztés	23
3.	A MEGOLDÁSI MÓDSZER KIVÁLASZTÁSA, A VÁLASZ INDOKLÁSA	24
4.	SPECIFIKÁCIÓ LEÍRÁSA.....	26
4.1.	Oldal tervek	26
4.1.1.	Regisztráció.....	26
4.1.2.	Login	26
4.1.3.	Főmenü.....	27
4.1.4.	Napló	27
4.1.5.	Profil	28
4.2.	Folyamatábra.....	29
4.3.	Projektek terve	30
4.4.	Fejlesztés és a közben felmerülő tapasztalatok	34
4.5.	Tesztelés	38
4.6.	Hasonló rendszerekkel való összehasonlítás.....	42
5.	TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK	43

6.	ÖSSZEFOGLALÁS.....	43
7.	CONCLUSION.....	44
8.	IRODALOMJEGYZÉK.....	46
9.	ÁBRA ÉS TÁBLÁZATJEGYZÉK	49
	7.1 Táblázatjegyzék	49
	7.2 Ábrajegyzék.....	49

1. BEVEZETÉS

Mindennapi életünk során egyre elterjedtebbé vált a fitness applikációk használata. Az oka ennek a jelenségnek az, hogy ezen alkalmazások használata sok előnnyel járhat. A mobil telefonjaink és a velük párosítható készülékeink temérdek szenzorral és funkcióval vannak felszerelve, amik nagyon egyszerűvé és értelmezhetővé teszik egészségügyi, vagy tevékenységi adataink. A mai okosórák ezen adatokat könnyedén meg tudják mérni. Melléjük egy mobil applikációt rendelve különböző kategóriákban sok részletes lekérdezést készíthetünk és jeleníthetünk meg a felhasználónak. További hasznuk még, hogy az információ megosztás vagy mentés készüléken kívülre gyorsabb és hatékonyabb legyen.

A piacon már rengeteg ilyen alkalmazás található, viszont elmondható, hogy általánosságban véve egy felhasználó csak egy ilyen alkalmazást használ. Mindenki egy olyan alkalmazást keres, amiben megtalálható minden olyan funkció amire valaha szüksége lehet. Ilyen igény lehet, hogy az ember szeretné minél több egészségügyi adatot nagyon pontosan szemmel követni, vagy mondjuk, hogy szeretné a tevékenységeit számon tartani és figyelemmel követni saját fejlődését. Emiatt az alkalmazások törekednek valamilyen fókusz vagy cél kialakítására, amiben különböznek/jobbak, mint vetélytársaik.

A szakdolgozat célja, hogy összehasonlítsak pár már létező alkalmazást, és ezek után elkészítsek egy olyat, ami segíti a használók fizikai és egészségügyi fejlődését, az adatokat felhőben tárolja a telefon memóriája mellett, és egy autentikációs réteggel is rendelkezik.

2. AZ IRODALOM ALAPJÁN A LEHETSÉGES MEGKÖZELÍTÉSI MÓDOK ÉS MEGOLDÁSOK ÁTTEKINTÉSE ÉS ELEMZÉSE

2.1. Fitness alkalmazások összehasonlítása

Hordozható eszközeink népszerűvé válásával egyre egyszerűbbé, hozzáférhetővé válik az, hogy figyelemmel kísérjük a tevékenységeinket, és egészségünket. Divat lett számon tartani, hogy naponta hány lépést tettünk meg, mennyi kalóriát égettünk el, továbbá mennyit edzettünk. Erre kiváló módszer, hogyha okostelefonjainkon valamilyen fitness vagy Health applikációt használunk. Temérdek ilyen található a piacon, sokuk ingyenesen is elérhető.

[2] A fitness applikációknak 2 fő fajtája van. Az elsők, amik főként a tevékenységekre fókuszálnak, a többi pedig az egészségünk megfigyelésére.

- Tevékenységre fókuszáló applikációk

Ezek az applikációk, ahogy fent olvasható főként a különböző testmozgással kapcsolatos tevékenységekre koncentrálnak. Ilyen tevékenységek lehetnek az úszás, futás vagy épp a biciklizés. iOS eszközökre ilyen programok például a Cyclemeter, ami a pontos és bő adatok szolgáltatására fókuszál biciklizés vagy futás közben vagy a Strava, ami hasonló célt szolgál, viszont közben figyelemmel kíséri a megtett távolságot, sebességet, szívrítmust és még sok egyebet. Minél több tevékenység követésére képes egy alkalmazás, annál több lehetősége lesz a felhasználónak, és annál valószínűbb, hogy azt az alkalmazást választja. Némelyik applikáció akár teljes edzéstervezeteket tud ajánlani attól függően, hogy mennyire fitt az illető, avagy mennyi ideje van.

- Egészségünk megfigyelésére fókuszáló applikációk

Ezen alkalmazások nem feltétlenül veszik figyelembe a felhasználó által végzett mozgást, mert az általuk számon tartott dolog nem feltétlenül köthető egy-egy tevékenységhez. Természetesen több információnak van köze hozzájuk, mint például a kalória égetés vagy a megtett távolság, de úgy közelítik meg a problémát, hogy például ahhoz, hogy lefogyjunk maximum mennyi kalóriát kell vihetünk be, és mellette mennyi időt kell valamilyen mozgással töltenünk. Ilyen program például a „MyFitnessPal calorie counter” applikáció, ami az Alkalmazáserőforrás center jelentése szerint a legjobbnak ítélt táplálkozás segítő alkalmazás az Amerikai Egyesült Államokban [2]. További példa a FitBit alkalmazás, ami ugyancsak figyeli a súlyát, kalóriáját (továbbá egyéb adatait, mint a vérnyomás, és glükóz szintjét) a felhasználónak.

Egyéb fontos opcionálisan implementálható funkció közé tartozik alvás vizsgálata, és a nők számára a menstruáció számon tartása.

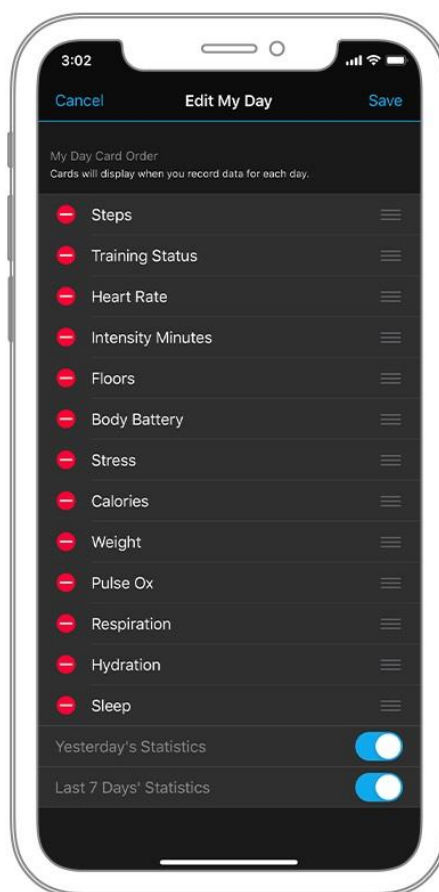
A következőkben szeretnék bemutatni népszerű alkalmazásokat és összehasonlítani őket.

2.1.1. Garmin Connect Mobile

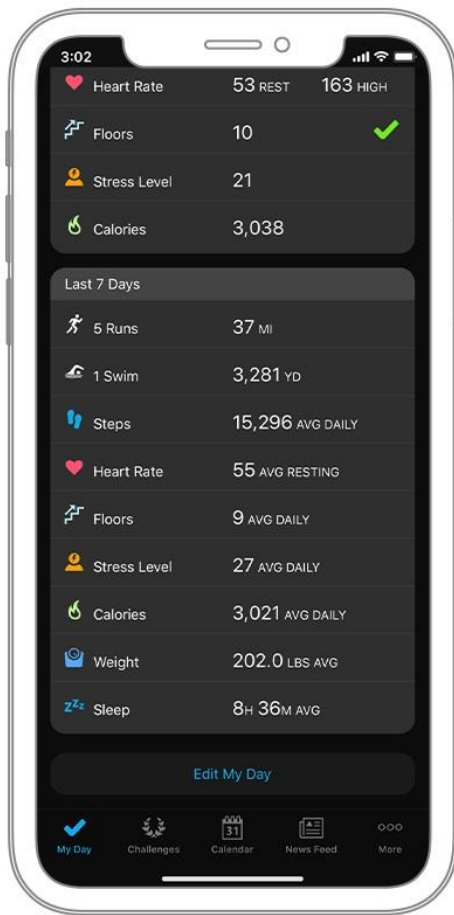
A Garmin Connect Mobile alkalmazás ingyenesen letölthető Android, illetve iOS készülékekre egyaránt. Legtöbb funkcióját akkor érhetjük el, ha párosítjuk egy Garmin márkájú okosórával. Belépéskor színes mezők jelzik számunkra az egyes statisztikákat. Különböző színek, különböző tevékenységet jeleznek így könnyedén átláthatóvá sikerült tenniük a rendszert. Ezen kívül személyesebbé tehető az alkalmazás felülete olyan módon, hogy a felhasználó mondhatja meg, hogy milyen adatokat jelenítsen meg az „My Day” irányítópulton. A „Last 7 Days” nézetben egy gyors áttekintés érhető el az elmúlt heti tevékenységeinkről és egészségügyi adatainkról, ezzel motiválva a jobb és jobb teljesítményt. Az alkalmazáson belül ezeken kívül lehetőség van figyelemmel követni az egyes teljesítmény adatokat (hogy könnyebben elérhesse a felhasználó a magának kitűzött céljait) és az egészségügyi statisztikákat (ezekről jól átlátható gráfokat is készít a program). Ezeken kívül még elérhetőek edzőprogramok, tevékenységkövetés és a nők számára menstruációs ciklusok számon tartása, vagy terhesség figyelemmel követése. A Garmin Connect Mobile egy kitűnő választás, hogyha egy tevékenységet és egészséget is megfigyelő alkalmazást keresünk. [3]



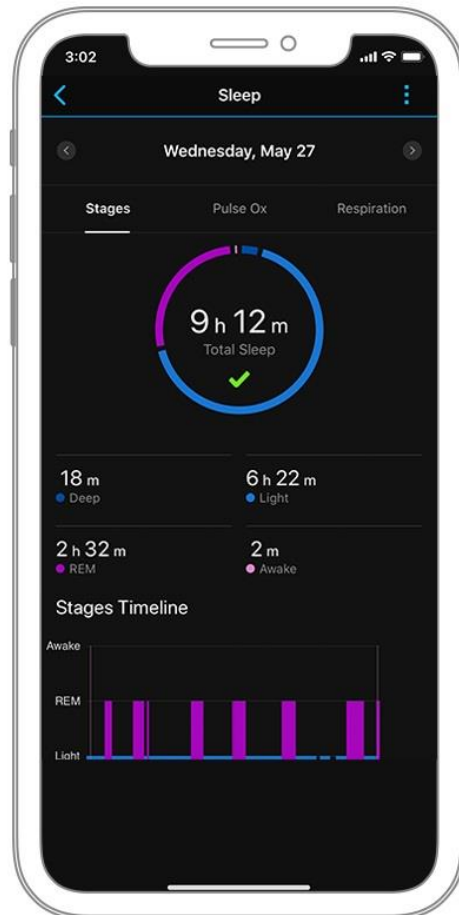
2.1. Garmin Mobile Connect: Menü [4]



2.2. Garmin Mobile Connect: Irányítópult [4]



2.3. Garmin Mobile Connect: Heti statisztikák [4]



2.4. Garmin Mobile: Egészségügyi statisztikák [4]

2.1.2. Google Fit

A Google Fit a Garmin Connect Mobile-lal ellentétben az egyszerűsége törekedett és így kevesebb funkciót is valósít meg. Ez azt jelenti, hogy többek közt nincsenek benne szociális dolgok (mint például a kihívások vagy pedig a barátokkal való versengés), nem veszi figyelembe (például futás közben) a magasság különbségeket, a felhasználó testhőmérsékletét, és nem használja az „auto pause” funkciót, ami röviden azt jelenti, hogy automatikusan leállítja és újra elindítja a különböző mérőeszközöket amikor kell. Négy fő szekciója van az alkalmazásnak: a Főmenü, avagy Home, a Napló, avagy Journal, a Böngészés, ahol bővebb egészségügyi információkhoz juthatunk hozzá, és a Profil.

A Főmenüön kaphatunk egy áttekintést:

- a jelenlegi nap tevékenységeiről,
- egészségügyi adatairól, és ebbe beletartozik az, hogy:
 - Mozgás Percek,
 - Szív Pontok
 - a lépések száma,
 - az elégetett kalória mennyisége,
 - a megtett távolság,
 - szívrítmus adatok,
 - súly,
 - nemrég teljesített gyakorlatok,
 - alvás adatok,
 - és ajánlatok tanító jellegű Youtube videókra.

A Napló egy egyszerű lista, amiben végig nézhetjük az eddig teljesített tevékenységeinket függetlenül a mozgás típusától. Ezeket az applikáció automatikusan hozza létre bármiféle felhasználói beavatkozás nélkül.

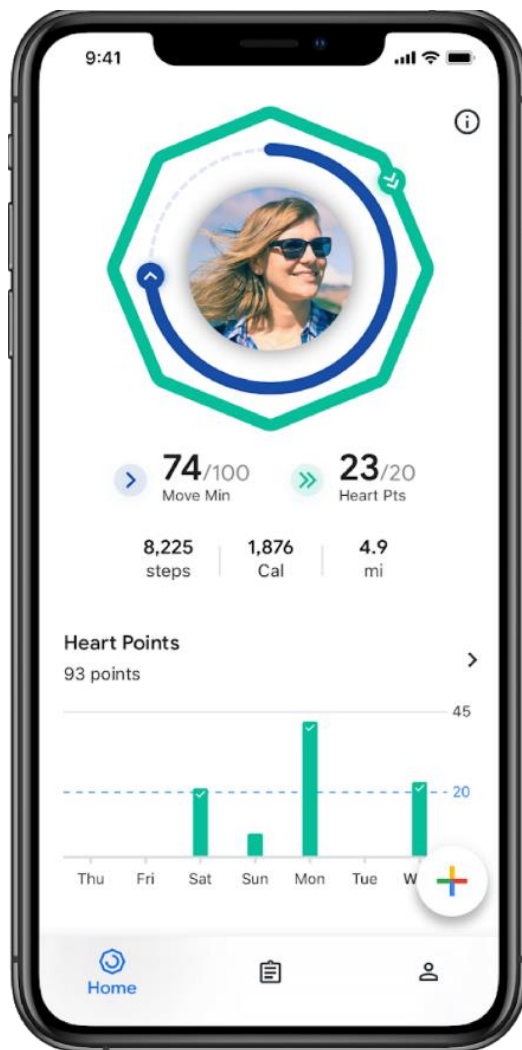
A Böngészés fülben lényegében összegezve és kibővítve láthatjuk azokat az egészségügyi és fitness adatokat, amiket a másik három képernyő is megmutat. Említésre méltó, hogy figyelemmel követhetők az olyan egészségügyi adataink is mint a pulzusszám, vérnyomás, vércukorszint, testhőmérséklet, avagy nők számára a cikluskövetés is.

A Profilon lehetőség van beállítani a személyes adatokat, és a tevékenységcélokat. Ez két adat formájában jelenik meg, amit a Google az Amerikai Szív Egyesület tanácsai alapján dolgozott ki:

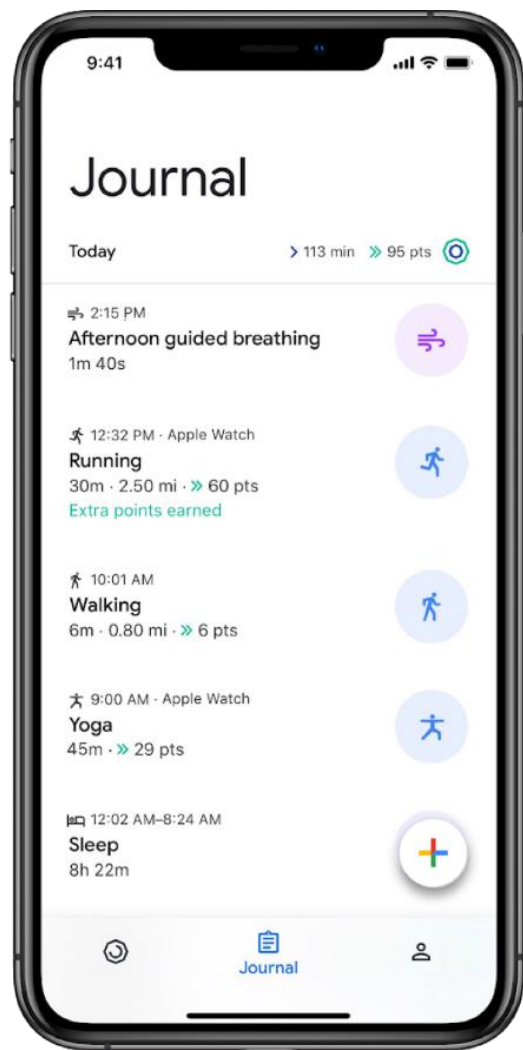
- Mozgás percek: Egyenletes és folyamatos mozgással lehet gyűjteni.
- Szív pontok: Intenzívebb mozgással gyűjthető. Egy perc ilyen mozgás, egy Szív pontot ér.

A két adat arra szolgál, hogy egyszerűbb legyen megérteni az edzés eredményét.

Az alkalmazás erőssége az egyszerűsége. Lényegében csak egy tevékenység követő alkalmazás, ami csak az alap egészségügyi adatokat és a mozgási adatokat követi figyelemmel és nem sok mást. A legfőbb funkció, ami hiányzik belőle az a közösségi platform. Más alkalmazásokkal ellentétben, itt nincs lehetőség az ismerősökkel való versengésre vagy egymás motiválására. Ezen kívül nem tartalmaz edző kurzusokat (igaz, ez nem fő aspektus egy ilyen applikációnál, de megéri felhozni), nincs benne evés vagy ivás naplózásra lehetőség, és nincs implementálva fejlődésmérés az egyes tevékenységekre. A külön megemlített Mozgás percekre és Szív pontok naplózására van lehetőség, tehát lehet látni, hogyha egy hónap után többet ért el a felhasználó, vagy kevesebbet. [5]



2.5. Google Fit: Főmenü [6]



2.6. Google Fit: Napló [6]

2.1.3. Strava

A Strava alkalmazással egy kifinomult és pontos futást, biciklizést és úszást segítő programot kapunk. Fő funkciója ezen három tevékenység naplózása GPS rendszer segítségével és az ezalatt mérhető adatok követése, mint például:

- az magasság különbségek mérése,
- milyen ütemben haladt a felhasználó, és ezzel kapcsolatban statisztikák,
- szívritmus

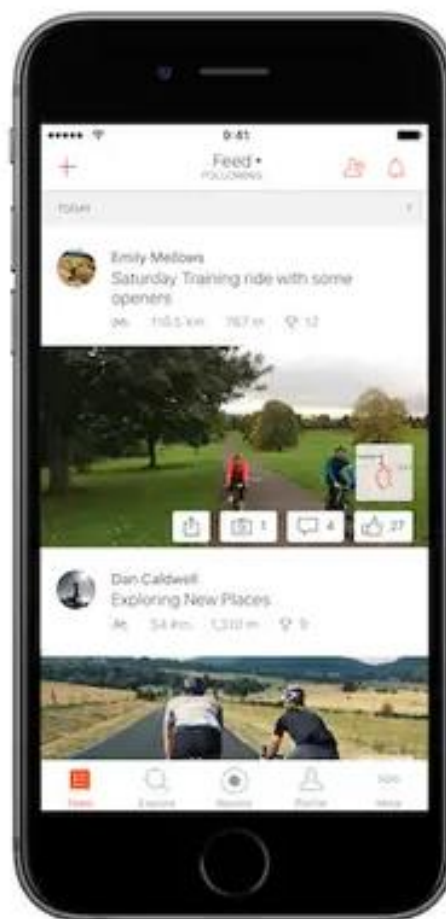
Megoszthatjuk jelenlegi pozíciónkat edzés közben a programon belül és ebből ismerőseink tudhatják, hogy épp merre vagyunk, mit csinálunk.

Az alkalmazás a profil elkészítése után, a saját Csatornámba dob be, ahol megtekinthetjük, csak a saját vagy akár az ismerőseink edzéseit, sporttevékenységeit. Emellett lehetőség van a Felfedezés menüpontban kihívásokra jelentkezni, ismerősöket találni, klubokhoz, csoportokhoz csatlakozni és úgynevezett szegmenseken kipróbálni magunkat, versenyezve másokkal. Ezek a szegmensek lényegében népszerű útszakaszok, amiket Strava felhasználók jártak be és ezekhez ranglistákat adtak hozzá. A középső menüpont a Felvétel, amivel egy utazást indíthatunk el. Ezután még két darab menüpont maradt, a Profil és az Edzés. A Profilunkon belül megváltoztathatjuk az adatainkat, és az egyes sporttevékenységekhez láthatunk külön-külön statisztikákat. Az Edzés fül alatt edzésterveket találhatunk és próbálhatunk ki. Ez nem érhető el az ingyenes verzióban.

A fentiek alapján jól látható, hogy a Strava erőssége a szociális rétege, emiatt az egészségügyi adatokat (mint amiket a Google Fit-nél is említettem) nem gyűjti be. Összegezve a Strava egy kiváló tevékenységre fókuszáló alkalmazás a sport szerelmeseinek. [7]



2.7. Strava: Profil [8]



2.8. Strava: Csatorna [8]



2.9. Strava: Csatorna 2 [8]



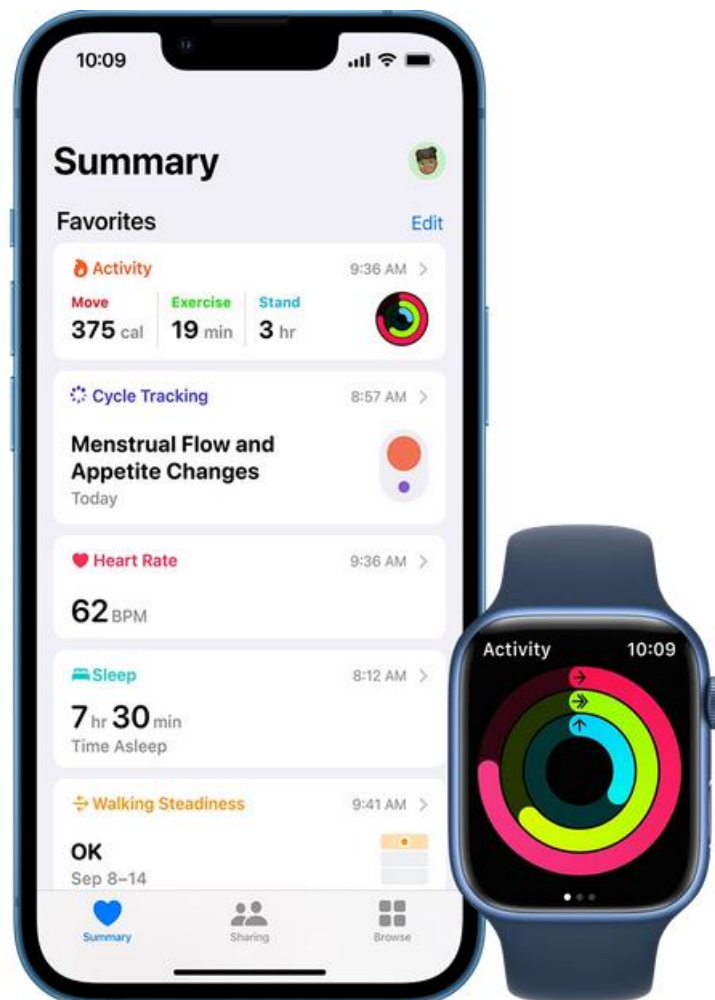
2.10. Strava: Út közben mérés [8]

2.1.4. Apple Health

Az Apple termékek előnye, hogy a legtöbb Apple által készített termék könnyedén összekapcsolható. Ebből eredően, ha van az embernek egy iPhone-ja és egy Apple Watch okosórája, akkor az óra által mért adatokat az Apple Health segítségével tudja kezelhetőbb felületen megtekinteni. Funkcionalitását figyelembe véve nem sokban különbözik a korábban bemutatott alkalmazásoktól.

Az alkalmazás az iPhone vagy pedig, ha rendelkezésre áll, akkor az Apple Watch szenzorjait felhasználva méri és tárolja el a fitness és egészségügyi adataink. Az órán futó alkalmazás valamint az Apple Health úgynevezett tevékenységi gyűrűket használ, amikkel vizualizálni tudja számunkra, hogy mennyit mozgunk aznap. Három színes kört használ, amiben a piros az elégetett kalóriát, a zöld az aznapi edzést vagy sétát, a kék

pedig azt mutatja be, hogy hányszor állt fel a felhasználó és mozgott legalább egy percet. A cél ezzel az, hogy mindhárom kör bezáródjon, ezzel teljesítve a napi ajánlott mozgásmennyiséget.



2.11. Apple Health főmenü és Apple Watch [10]

Az alkalmazás „end-to-end” titkosítással rendelkezik. Ez azt jelenti, hogy ha le van zárva a iPhone-unk vagy pedig, amikor az adatok elmentődnek iCloud-ra, csak azok az alkalmazások férhetnek hozzá, amiknek külön engedélyt adtunk. Ezt a titkosítást az adatvédelmi irányelveik alapján implementálták.

IOS 14-től az alkalmazás rendelkezik alvás követő funkcióval, amivel követni tudjuk az alvással kapcsolatos adataink. Beállíthatjuk, hogy mennyi alvásra van szükségünk, és hogy mikor akarunk lefeküdni, avagy felkelni.

Az étkezési szokásainkat is számon tudjuk tartani az alkalmazással. Beüthetjük, hogy mit és mennyit ettünk, amiből az alkalmazás kiszámolja, hogy mennyi tápanyagot vittünk így

be. Érdekesség, hogy iOS 15-től ez még egyszerűbbé vált, mert a készülék kameráját felhasználva is lehetőség van megadni a programnak ezeket az adatokat.

Az amerikai felhasználók részére opcionálisan elérhető Egészségügyi ID kártyát létrehozni. Ez lehetővé teszi, hogy abban az esetben, ha történik valami a tulajdonossal, akkor például a mentősök könnyebben tudjanak hozzáférni az ellátáshoz fontos információkhoz.

Az iPhone használók számára véleményem szerint ez egy elég hasznos alkalmazás, mert temérdek egészségügyi adatot tudnak követni főleg, ha párosítják egy Apple Watch okosórával. [9]



2.12. Apple Health alvás figyelő [11]



2.13. Apple Health Egészségügyi ID kártya [12]

2.1.5. Samsung Health

A Samsung Health nem feltétlenül a legbonyolultabb vagy legtöbb információval szolgáló fitness alkalmazás, ami elérhető a piacon, de az átlag felhasználó számára tökéletes lehet. Kompatibilis iOS és android készülékekkel is, és az Apple Health-hez hasonlóan akkor lehet a legtöbb funkcióját kihasználni, ha párosítja például egy Samsung okosórával. Rengeteg olyan szolgáltatást egyesít, amelyek segítenek kezelni például az alvási, az étkezési, és fitness adatokat egy intuitív felületen. Ezeken kívül még fitness edzői útmutatókat, mozgással kapcsolatos gyakorlatokat és hasonló gondolkodású felhasználók közösségét ajánlja.

Sok alap egészségügyi és fitness adatot követhetünk a Samsung Health programmal. Ilyen adatok például:

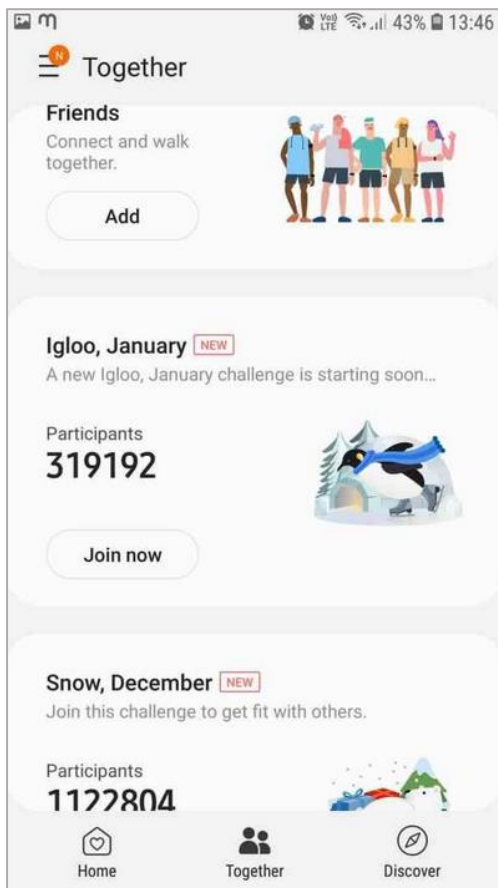
- Különböző edzéseken nyújtott teljesítmény
- Aktívan töltött idő
- Szívritmus
- Alvás
- Kalória égetés
- Ciklusok követése

Az alkalmazásnak négy fő oldala van. A Főmenü (Home), a Közösségi oldal (Together), Fitness, és a Saját adatok (My Page). Kisebb nagyobb átfedésekkel a következőket érhetjük el az egyes oldalakon:

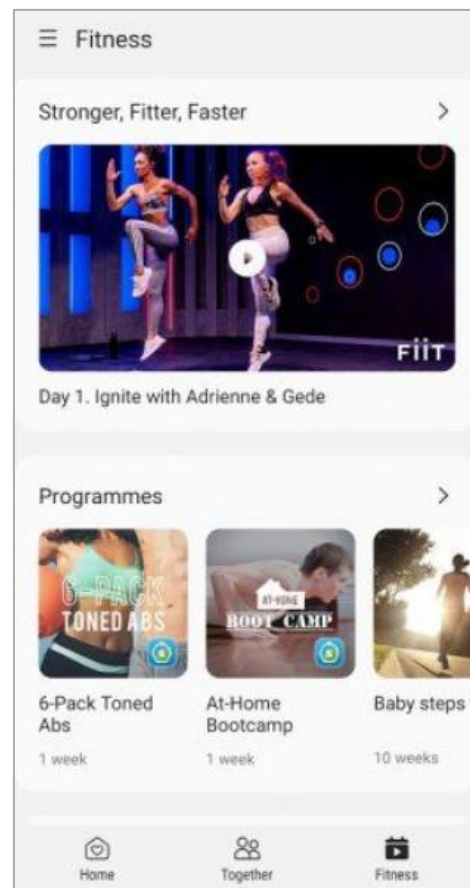
- A Főmenü a kezdőoldal. Itt tudjuk követni a napi céljainkat és a jelenlegi egészségügyi adatainkat. Kicsit lejjebb görgetve hozzá tudunk adni manuálisan is adnatokat, ezzel követve például a súlyunk, étel és ital bevitelünk, avagy menstruációs ciklusunk. Testreszabható, hogy milyen adatokra van szükség és miket nem kívánunk követni.
- A Közösségi oldalon kihívásokat teljesíthetünk, és összehasonlíthatjuk az adataink másokéval. Ezeken kívül lehetőség van barátokat, ismerősöket felvenni, és egymást motiválni az utazásaink során.
- A Fitness oldal egy bő könyvtára az edzéseknek és wellness anyagoknak, mint például edzés programok, videók, Samsung Health TV partner tartalmak, női egészség, és egyebek.
- A Saját adatok oldalon tudjuk szerkeszteni a profilunk adatait. Hozzá tudjuk adni a nemünk, magasságunk, súlyunk, és születési dátumunk, hogy az alkalmazás kalkulálni tudja az elégetett kalóriát és a pulzus tartományunk. Azt is beállíthatjuk, hogy mennyi legyen az a napi tevékenység mennyiség, amit el szeretnénk végezni egy egytől négyig terjedő skálán. Ezeken a dolgokon kívül egy nagyobb képet kaphatunk az egészségügyi adatainkról a heti összeggzéseink, személyes rekordjaink és elért teljesítményeink alapján.

A többi fitnessz applikációval ellentétben (mint például a Google Fit, vagy az Apple Health) nem kínál sok más harmadik féltől származó alkalmazás számára integrációs lehetőséget. Ez alól kivétel például a népszerű futók és biciklizők számára készült Strava alkalmazás.

Összegezve, a Samsung Health egy kiváló opció, mint fitnessz app Samsung hordozható eszközök mellé. Amik különlegessé teszik az a felettébb nagy mennyiségű edzés és közösség építő kihívás, viszont mivel kevés más programmal tud szinkronizálni, így sokan gondolhatják úgy, hogy más programmal próbálkoznak. [13]



2.14. Samsung Health közösségi oldal [14]



2.15. Samsung Health Fitness oldal [14]

2.1.6. Fitbit App

A Fitbit App a világ egyik vezető egészségügyi és fitness alkalmazása. Kompatibilis egyaránt android és iOS operációs rendszerrel rendelkező eszközökkel is. Felhasználhatósága szempontjából nincs benne semmi különleges, hasonló a korábban említett applikációkhoz, természetesen pár különbséggel. A fő ok, amiért valaki használná, az a Fitbit márkájú okosórákkal való párosítási lehetőség.

Az alkalmazás ugyanúgy figyelemmel követi:

- Alvási szokásaink
- Edzéseink adatait
- Súlyunk változását
- Víz fogyasztásunk
- Étkezési szokásaink

Hozzáférhetünk ingyen edzéssel kapcsolatos videókhoz vagy hangfelvételekhez, amiket otthon a nappalinkban is elvégezhetünk, amikor kedvünk tartja. A tárház tartalmaz kardiót, jógát, és még sok másfajta edzéstípust.

Rendelkezik még a Fitbit egy közösségi modullal is, ahol megoszthatjuk és összehasonlíthatjuk edzési vagy kihívási eredményeinket barátainkkal, ezzel is ösztönözve őket további mozgásra.

A Fitbit rendelkezik prémium verzióval, amivel sokkal több edzéssel kapcsolatos videóhoz férhetünk hozzá ismert edzésprogramokkal foglalkozó csoportoktól, vagy az egyes rendszereket sokkal jobban testre szabhatjuk ezzel is egyszerűsítve a napi edzésrutin kialakítását.

Az alkalmazásban nincs semmi olyan, ami nagyon megkülönböztetné a piacon lévő többi terméktől viszont, ha Fitbit okosórával rendelkezünk, akkor az egyik legjobb választás.
[15]



2.16. Fitbit mai nap mérései [15]



2.17. Fitbit heti lépésszámok [15]

Összehasonlítva a hat alkalmazást megállapíthatjuk, hogy mindegyik kitűnően megvalósítja a kitűzött céljait. A Garmin Mobile Connect igyekezett minél több funkciót belerakni, aminek nagyrészt több mint valószínű, hogy az átlag felhasználó nem fogja felhasználni. Viszont amikor valami újat szeretne mérni vagy tudni, akkor a funkciók ott vannak használatra készen. A Google Fit már inkább azt a célt tűzte ki, hogy kevesebb funkciót kerüljön implementálásra, de azok kifogástalanul működjenek. A Strava csak az edzéssel kapcsolatos adatokat veszi figyelembe, de a fő attrakciója az alkalmazásnak inkább a közösségi felületén van. Az Apple Health, a Samsung Health és a Fitbit ugyanúgy kitűnő alkalmazások a piacon. A fő különbség köztük, eltekintve pár különböző funkciótól az az, hogy milyen okosórákkal kompatibilisek. Leszűrhető tehát

ezekből a megállapításokból, hogy a fitness alkalmazások nagy része sokban hasonló, mert a funkciók nagy része közös, és lényegében csak annyi különbség van köztük, hogy mi a cél (kompatibilitás, edzés, fitness, egészség szemmel tartása), és milyen megvalósításokat használtak a fejlesztők.

2.2. Felhőben való tárolási megoldások összehasonlítása

Az előző részben felsorolt alkalmazások mindegyikénél az első lépés egy profil létrehozása volt. Ez abban segít, hogy egy felhasználó csak a saját adataihoz férjen hozzá, ezt az autentikációs megoldások összehasonlítása részben szeretném részletesebben átvenni. Viszont ez egy olyan lehetőséget is lehetővé tesz, hogy adatainkat más eszközökkel is elérhessük. Ennek előkövetelménye, hogy az adatok el legyenek tárolva egy külső táregységen, amire tökéletes a felhő.

A továbbiakban szeretnék bemutatni pár lehetőséget a felhőben tárolásra.

2.2.1. Google Cloud

A Google Cloud többféle szolgáltatást is nyújt. Első körben a Cloud SQL egy teljesen felügyelt relációs adatbázis-szolgáltatás MySQL, PostgreSQL és SQL Server számára. Az ismert relációs adatbázisokat tudjuk rajta futtatni gazdag bővítménygyűjteménnyel, konfigurációs jelzőkkel, fejlesztői ökoszisztémákkal az önkezeléssel járó gondok nélkül. A Cloud SQL-en kívül felhasználható lenne még a Cloud Storage szolgáltatás, abban az esetben, ha nem SQL adatbázist használnék tárolásra. Ha például egy Excel táblázatot szeretnék használni, akkor a Storage jobb megoldás lenne.

Cloud SQL	Cloud Storage
• Felügyelt • Integrált • Megbízható • Könnyű migrációk	• Könnyedén válthatunk olcsóbb tárolási osztályokba. • Többféle opció redundáns/biztonsági másolatok készítésére. • Könnyű adat küldés a Cloud Storage-ba. • Tárolási osztályok bármilyen munkaterheléshez

2.1. táblázat: Cloud SQL és Cloud Storage összehasonlítása [16] [17]

A Google Cloud hátránya, hogy az előbb említett szolgáltatásokat nem lehet teljesen ingyenesen elérni, csak amíg tart a regisztrációkor kapott 300 dollár vagy 90 napig.

2.2.2. Microsoft Azure

A Microsoft Azure egy olyan felhő platform, amivel különböző alkalmazásokat lehet készíteni, telepíteni és futtatni. A feladatom szemszögéből nézve több szolgáltatást is ajánl. Ezek közül az első, ami teljesen ingyenes az az Azure Cosmos DB. Ez egy teljes körűen felügyelt NoSQL-adatbázis a modern alkalmazásfejlesztéshez. Egyszámjegyű ezredmásodperces válaszidőkkel rendelkezik, továbbá automatikus és azonnali skálázhatósággal bármilyen méretben garantálják a sebességet. Az ingyenes verzió kipróbálása esetén egy olyan fiókot használhatunk, amely 1000 RU/s (Request Unit/second) és 25 GB tárhellyel rendelkezik.

Hasonlóan a Google Cloudhoz itt is van lehetőség SQL adatbázisos tárolásra az Azure SQL szolgáltatással, és másfajta fájlok tárolására az Azure Blob Storage és File Storage felhasználásával. A Blob Storage abban az esetben használható, hogyha rendszertelen adatokat szeretnénk tárolni specifikus adatmodell nélkül. Ezzel szöges ellentétben létezik a File Storage szolgáltatás egy teljesen felügyelt elosztott fájlrendszer. Abban az esetben hasznos, hogyha főként standard bővítményű fájlokat használunk, mint például a .docx, .png. Ingyenesen nem érhetőek el teljesen, csak egy rövid ideig, korlátozott funkciókkal. [18]

2.2.3. Firebase Cloud Services

A Firebase egy ingyenesen használható NoSQL felhő adatbázis. Tökéletes arra, hogyha azt szeretné az ember, hogy az alkalmazása összes felhasználója valós időben folyton elérje az adatait, akkor is, amikor az alkalmazás offline. Az adatokat JSON-ban tárolja. Platformokon átívelő alkalmazások készítése esetén minden ügyfél egy valós idejű adatbázison osztozik és megkapják a legújabb frissítéseket és adatokat. [20]

Fő szolgáltatások:

- valós idejű
- offline is elérhető
- azonnal elérhető a kliens készülékekről
- több adatbázisra is skálázható

2.2.4. Lokális webszerver

Korábbi tanulmányaim alatt lehetőségem volt ASP.NET-et tanulni. Ezen tárgy keretein belül megtanultunk webszervereket készíteni, amely lokálisan futtathatók. Előnye, hogy nincs szükség nagyobb felhő alapú szolgáltatóra, mert elkészíthetem az adatbázist SQL-ben magamnak a saját gépemem, korábbi tudást tudok felhasználni és gyakorolni, ami később még jól jöhet tanulmányaim után.

Összehasonlítva, a fent említettek közül a teszteléshez érdemes lenne egy lokális webszerver létrehozása, majd miután a program működik, és van igény terjesztésre, átváltani egy először ingyenes cloud szolgáltatásra, mint a Firebase vagy az Azure Cosmos DB. Nagyobb mennyiségű felhasználó mellett viszont már a fizetős szolgáltatások érnék meg.

2.3. Autentikációs megoldások összehasonlítása

Rengeteg felhasználói azonosítási módszer létezik manapság. Ez kezdődött még az egyszerű jelszó rendszerrel, és mára már eljutott odáig, hogy készülékeink már arcfelismerő rendszerekkel is rendelkezhetnek. Ez még mindig nincs azon a szinten, mint amire az emberi agy képes, de szépen lassan egyre közelebb jutunk hozzá.

2.3.1. Azonosítási módszerek osztályozása

Minden azonosítási módszer osztályozható három kulcs paraméter alapján.

- *Használhatóság*
 - Mennyire természetes és probléma mentes a végfelhasználó számára a használata az azonosítási folyamat?
- *Biztonság*
 - Mennyire nehéz átverni a rendszert?
- *Telepíthetőség*
 - Mennyire egyszerű telepíteni minden platformra vagy eszközre? Itt fontos lehet figyelembe venni, hogy aki számára készítjük a rendszert az egy szimpla fogyasztó, vagy pedig egy cég alkalmazottja.

2.3.2. Azonosítási módszerek csoportosítása

- *Valami, amit tud a felhasználó*
 - Ez lehet egy nem publikus információ, mint mondjuk egy jelszó, vagy egy titkos/személyes kérdés. Általánosságban véve, ezt a legegyszerűbb megvalósítani, de egyben nem is annyira biztonságos, mert könnyen támadható.
- *Valami, amije van a felhasználónak*
 - Valamilyen más eszköz, mint például egy FIDO kompatibilis kulcs. Általában ezek nagyon biztonságosak adathalászati vagy más támadások ellen, mert ez a privát kulcs nem hagyja el az eszközt. Viszont, ha ezt az eszközt ellopják, akkor nincs mit tenni.
- *Valami, ami a felhasználó*
 - Lényegében biometrikus autentikációs módszerek, mint az ujjlenyomatok vagy az arcfelismerő rendszerek. Ezeket tudja a felhasználó a legegyszerűbben használni, viszont nagyon erősen támaszkodnak a

mögöttes platformon vagy eszközön. A módszer a jövőben talán a leggyakoribb technológia lesz, de csak akkor, ha minden platformon a hardware-ek és a software-ek biztonságosak lesznek, és jól működik közöttük a kommunikáció. Szükség van valamilyen pótmechanizmusra még olyan esetben, hogyha mondjuk a felhasználó megvágja az ujját. Így lényegben kialakul egy támadható felület. [21]

2.3.3. Hitelesítési módszerek alapján való csoportosítás:

- *Jelszó alapú*
 - Legelterjedtebb megvalósítása a hitelesítésnek. Szavakból, számokból és speciális karakterekből állhatnak. Ahhoz, hogy megvédje magát a felhasználó a legtöbb helyen erős jelszó kitalálása után engednek csak tovább. Az erős jelszó lényege, hogy minél többfajta karaktert tartalmazzon. A probléma a jelszó alapú hitelesítéssel, hogy a legtöbb ember majdnem minden helyen ugyanazt a jelszót használja, mert egyszerűbb csak egyet megjegyezni. Emiatt ez a módszer könnyedén támadható.
- *Többtényezős*
 - A módszer a hitelesítéshez kettő vagy több egymástól különböző módszert használ. Lényegében rétegezi a védelmet.
- *Tanúsítvány alapú*
 - Digitális tanúsítványokkal éri el az azonosítást. Ilyen tanúsítványok lehetnek például a jogosítvány, vagy pedig az útleve. A felhasználó megadja ezt a szervernek, ami utána megnézi, hogy az hivatalos-e. Ezután rejtjelezéssel a szerver ellenőrzi, hogy a felhasználónak megvan-e a megfelelő privát kulcs, ami hozzá lett kötve a tanúsítványhoz.
- *Biometrikus*
 - A biometrikus hitelesítési módszerek a minden embernél egyedi biológiai karakterisztikákra támaszkodnak. Az előnyei, hogy hozzáadhatóak a többtényezős hitelesítési folyamatokhoz, mint egy lépés, és könnyedén tudjuk összehasonlítani a nyert adatokat az adatbázis elemeivel. Gyakori megvalósításai a következők:
 - Arcfelismerés
 - Ujjlenyomat felismerés
 - Hang felismerés
 - Szem szkennelés
- *Token alapú*
 - A Token alapú a személyes adatok bevitele után ad a felhasználónak egy titkosított szövegfájl véletlenszerű karakterekből. Ezután hozzá lehet férni a program funkcióihoz anélkül, hogy újra beírnánk az adataink

hogya ez a szövegfájl a birtokunkban van. Ez bizonyítja, hogy már van hozzáférésünk a rendszerhez. [22]

2.4. Szolgáltató és fogyasztó alkalmazás

2.4.1. Alkalmazás szerepkörök összehasonlítása

Egy olyan android fitness alkalmazás esetében, ahol az alkalmazás egy fitness eszközről szerez adatokat, el kell gondolkodni, hogy milyen kapcsolat áll fenn a két eszköz között. Általánosságban véve, ilyen esetekben egy szolgáltató/fogyasztó kapcsolatról beszélünk, ahol a fitness eszköz a szolgáltató és az android alkalmazás a fogyasztó. Szeretnék erről a két szerepkörrel pár szót ejteni.

- *Szolgáltató (provider) alkalmazás:*
 - Ahogy a név is utal rá, a támogató alkalmazás adatokat szolgáltat a fogyasztó alkalmazás felé. Az alkalmazások közötti kapcsolat felállítása alatt ez az alkalmazás passzív szerepet játszik. Vár arra, hogy a szolgáltató adatokat kérjen, és ezután eldönti, hogy ezeket a kéréseket elfogadja, vagy megtagadja. Ha elfogadja, akkor megszületik a kapcsolat. A szolgáltató alkalmazás általánosságban véve nem rendelkezik adat kijelzéssel, lényegében csak az eszköz által gyűjtött adatokat szolgáltatja a fogyasztó alkalmazás felé, mikor az kéri.
- *Fogyasztó (consumer) alkalmazás:*
 - A fogyasztó alkalmazás viszont már aktív szerepet játszik a kapcsolat felállításában. Ő küldi a fent említett szolgáltatás kapcsolat kérelmet majd vár, hogy megkapja a kérelem elfogadását vagy elutasítását. Kapcsolódás után viszont már nem küld további adatot a szolgáltató felé, csak fogad onnan. Ez az alkalmazás rendelkezik felhasználói felülettel, és ezek keresztül futnak a felhasználói interakciók a szolgáltató eszköz által gyűjtött adatokkal.

Projektünk esetében kétféle elképzelés lehetséges. Az egyik, amikor az android mobiltelefon a szolgáltató, és az okos óra a fogyasztó. Ebben az esetben minden adat az órán látható, és az okos telefont használjuk adatok lekérésére. A másik eset ennek a fordítottja. Az okos telefont használjuk az adatok kijelzésére és az okos órát, csak az adatok szerzésére használjuk. Felhasználhatóság szempontjából a második opció a jobb, mert az okos telefonoknak sokkal nagyobb a kijelzője, sokkal több minden megjeleníthető rajtuk. [23]

2.5. Android fejlesztés

Egy android alkalmazás fejlesztésénél van egy elengedhetetlen döntés, ami kritikus szerepet játszik az alkalmazás jövőjével kapcsolatban. Ez a döntés az, hogy az egy weboldal, vagy pedig egy önálló (standalone/native) applikáció legyen. Mindkettőnek megvannak az előnyei és hátrányai.

Natív applikáció	Web alapú applikáció
Sebesség A program futásához kellő kód lokálisan tárolódik és nem kell várni, hogy az letöltődjön az internetről.	Egyféle megoldás: A legnagyobb előny, hogy nem kell azon gondolkodni, hogy milyen rendszerre készítjük a kódot. Egy verzió létezik a kódból, de az elérhető minden felhasználó számára több platformon is.
App Store: Egy hozzáférhető platform, ahol reklámozni lehet az applikációt, hogy új felhasználókat szerezzünk és alkalmazáson belüli különleges funkciók felhasználását teszi lehetővé.	Valós idejű fejlesztés: Nem kell várni, hogy az App Store-ban elfogadják a fejlesztéseket. Valós időben lehet fejleszteni.
Helyi tárolás: Dinamikus információk tárolására lehetőség van lokálisan és a felhő szinkronizáció megtörténhet később is.	Szabadság: Nem kell tartani magunkat a felhasználási feltételekhez és szabályokhoz, amiket a különböző Store-ok előírnak.
Funkciók: Hozzáférhetünk platform (amire telepítettük az alkalmazásunk), hardware funkcióihoz, mint például a GPS, vagy a fénykép készítés.	Fejlődés: Technikai szempontból egyre fejlettebbek a web alapú megoldások és egy képzett web fejlesztő lassan áthidalhatja minden limitációt, amit a web alapú fejlesztés tartalmaz.
Hátrányok: Amikor elkészítünk alkalmazást okos telefonra, akkor az első dolog, amit eldöntünk, hogy milyen operációs rendszerre kerüljön. Natív esetén a hátrány az, hogyha váltani akarnánk ezek között, mert lehet teljesen át kell írni a kódot.	Hátrányok: Limitált hozzáférés a hardware-ek funkcióihoz, mint például a GPS, kamera. Egyes funkciókat nem lehet olyan könnyen megoldani, mert figyelembe kell venni az egyes platformok különböző tulajdonságait, és felszereltségét.

2.2. táblázat: Natív és Web alkalmazások összehasonlítása

Összesítve, mindkét megoldásnak vannak előnyei és hátrányai. Abban a döntésben, hogy melyiket használjuk az segít, hogy milyen funkciókat szeretnénk implementálni és milyen szinten szeretnénk elérhetővé tenni a termékünket. Ha hosszú távú céljaink vannak a programmal, akkor lehet jobban megéri a web alapú, viszont azt is figyelembe kell venni, hogy milyen a felhasználói élmény ebben a gyorsan fejlődő mobil iparban. [24]

3. A MEGOLDÁSI MÓDSZER KIVÁLASZTÁSA, A VÁLASZ INDOKLÁSA

A szakdolgozat feladatának egy olyan mobil applikáció elkészítését választottam, ami képes adatokat lekérni egy okosóráról és ezen adatokat összesíteni és megjeleníteni a kijelzőn a megfelelő menüpontok alatt.

Tárolható adattípusok:

- *Tevékenységs adatok*
 - A felhasználó fizikai tevékenységével kapcsolatos adatok. A tevékenységs adatok különböző fizikai tevékenységekkel kapcsolatos adatokra utal, mint például a lépésszám, az elégetett kalória, valamint az egyéb tevékenységek sebessége, beszélhetünk itt a futásról, gyaloglásról vagy épp kerékpározásról.
- *Hely adatok*
 - A hely adatok jelentik a felhasználó aktuális és korábbi tartózkodási helyére vonatkozó adatokat. Ezek lehetővé teszik az alkalmazás számára, hogy statisztikákat jelenítsen meg.
- *Táplálkozás és hidratáció*
 - Táplálkozási és hidratációs adatokba beletartoznak a kalóriabevitel, a mikro- és makró tápanyagokkal, valamint a vízbevitellel kapcsolatos adatok is.
- *Testérzékelőkkel kapcsolatos adatok*
 - Olyan adatok, amik adatokat gyűjtenek a felhasználó fizikai állapotáról, mint például a pulzusszám. [25]

A programomat a Google Fit céljai alapján szeretném felépíteni, értem itt azt, hogy legyen egyszerű, könnyen átlátható és stabil. A tárolást SQL adatbázissal szeretném megoldani, és a felhő rendszer, amit választottam az egy általam írt ASP.NET webszerver, amit lokálisan futtatok.

Az android környezet elkészítéséhez Android Studioban Kotlin nyelvet használnék fel, mert kutatásaim alatt arra jutottam, hogy egyszerre képes kezelni saját nyelvű és Java nyelvű kódot ezzel jelentősen megkönnyítve a munkafolyamatokat.

A hordozható alkalmazást (az óra programját) Visual Studio-ban .NET környezetben szeretném elkészíteni. Az oka ennek az, hogy választhattam milyen nyelven/környezetben készítem azt el. Három lehetőségem volt.

1. **Natív alap C, C++:** A C nyelvben kevés tapasztalatomat figyelembe véve úgy, hogy az android fejlesztés is relatív új volt számomra, úgy ítéltam, hogy feleslegesen nehéz lenne pláne úgy, hogyha van olyan lehetőség, amiben jobban

elgazodnék. Továbbá a Natív és a Web alapú fejlesztés egy harmadik fejlesztőkörnyezetet igényelt volna, a Tizen Studio-t.

2. **Web alapú HTML, CSS3, Javascript:** Hasonlóan a Natívhoz, a Web alapú fejlesztés nyelveinél is limitált tapasztalatom van. Ezen kívül a 2.5-ös pontban írt előnyök és hátrányok miatt Natívan szerettem volna elkészíteni a projektet így ez a lehetőség kizáródott.
3. **.NET alapú C#:** Tekintettel az egyetemi tanulmányaimra és a foglalkozásomra, mint C# fejlesztő ez a megoldás tűnt a leghatékonyabbnak. Egy általam ismert környezetben (Visual Studio) lenne lehetőség programozásra, és egy ismert nyelven. Az óra funkcióinak lekéréséhez ebben az esetben, csak a Samsung Accessory SDK-ra és a „Visual Studio Tools for Tizen” bővítményre volt szükség.

Választásaimat azon módon szeretném indokolni, hogy tulajdonomban van egy Samsung Galaxy Watch Active 2 típusú okosóra, és a saját mobiltelefonom pedig Android rendszerrel rendelkezik. Ennek tekintetében használnám az Android Studio-t és a Samsung Accessory SDK-t, a többi projektet pedig a Visual Studio keretein belül C# nyelven készíteném el.

4. SPECIFIKÁCIÓ LEÍRÁSA

4.1. Oldal tervek

Első lépésként be szeretném mutatni a program különböző oldalainak kinézetére vonatkozó korai elképzeléseimet. Alapvetően a program három oldalból fog állni, de ezeken kívül lenne még egy regisztrációs oldal és egy bejelentkező oldal.

4.1.1. Regisztráció

Ezen az ablakon van lehetőség regisztrálni az adataink megadásával.

TLM FIT

First Name Last Name

User Name

Email Address

Password

Confirm Password

Country Gender

Sign up

4.1. Regisztrációs ablak terv

TLM FIT

User Name

Password

Login

Login with Google+

Create New Account

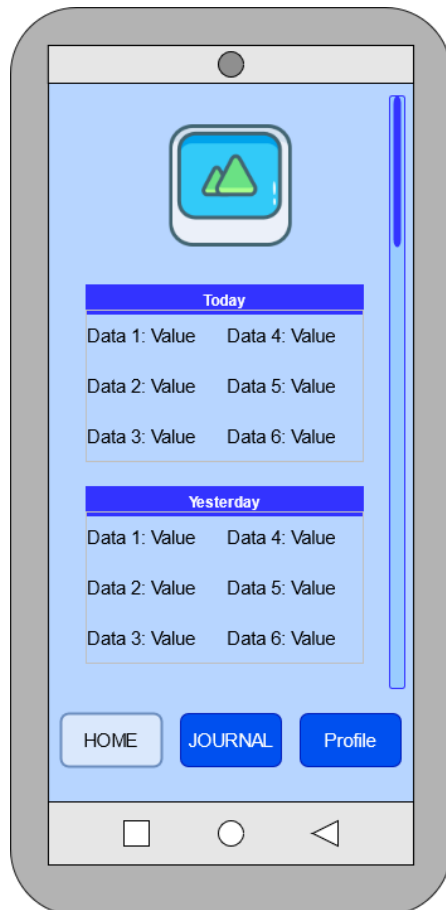
4.2. Bejelentkező ablak terv

4.1.2. Login

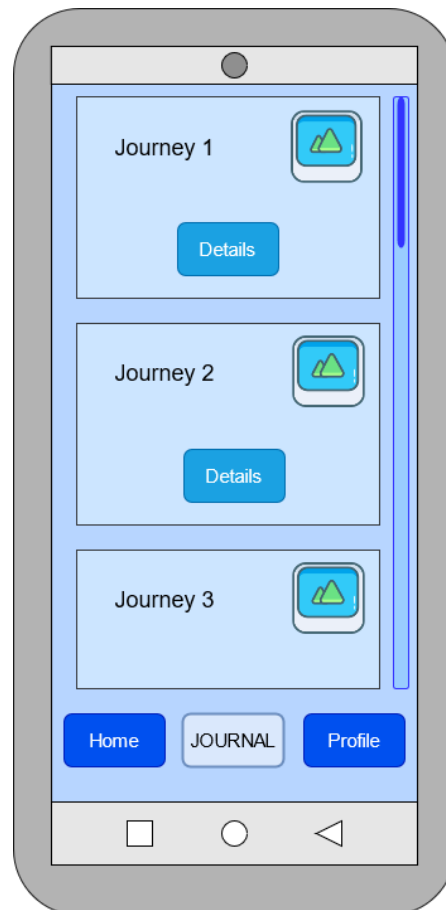
A Login oldalon van lehetőség új felhasználót létrehozni, vagy a Google által tárolt adatok felhasználásával. Ez fogadja a felhasználót az első megnyitás alkalmával.

4.1.3. Főmenü

A Főmenüben (Home) lesz lehetőségünk egészségügyi adatok és egyéb funkciók elérésére. Ez az oldal fogadja a felhasználót, mikor megnyitja az alkalmazást, vagy regisztrálás után.



4.3. Menü ablak terv



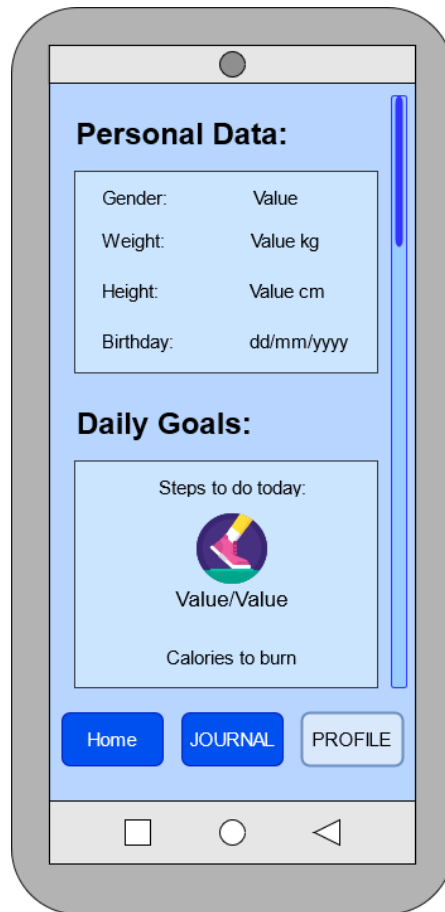
4.4. Napló ablak terv

4.1.4. Napló

A Naplóban (Journal) lesz lehetőség az egyes felvett utak adatait, vagy az elmúlt hónapok adatait visszaneézni.

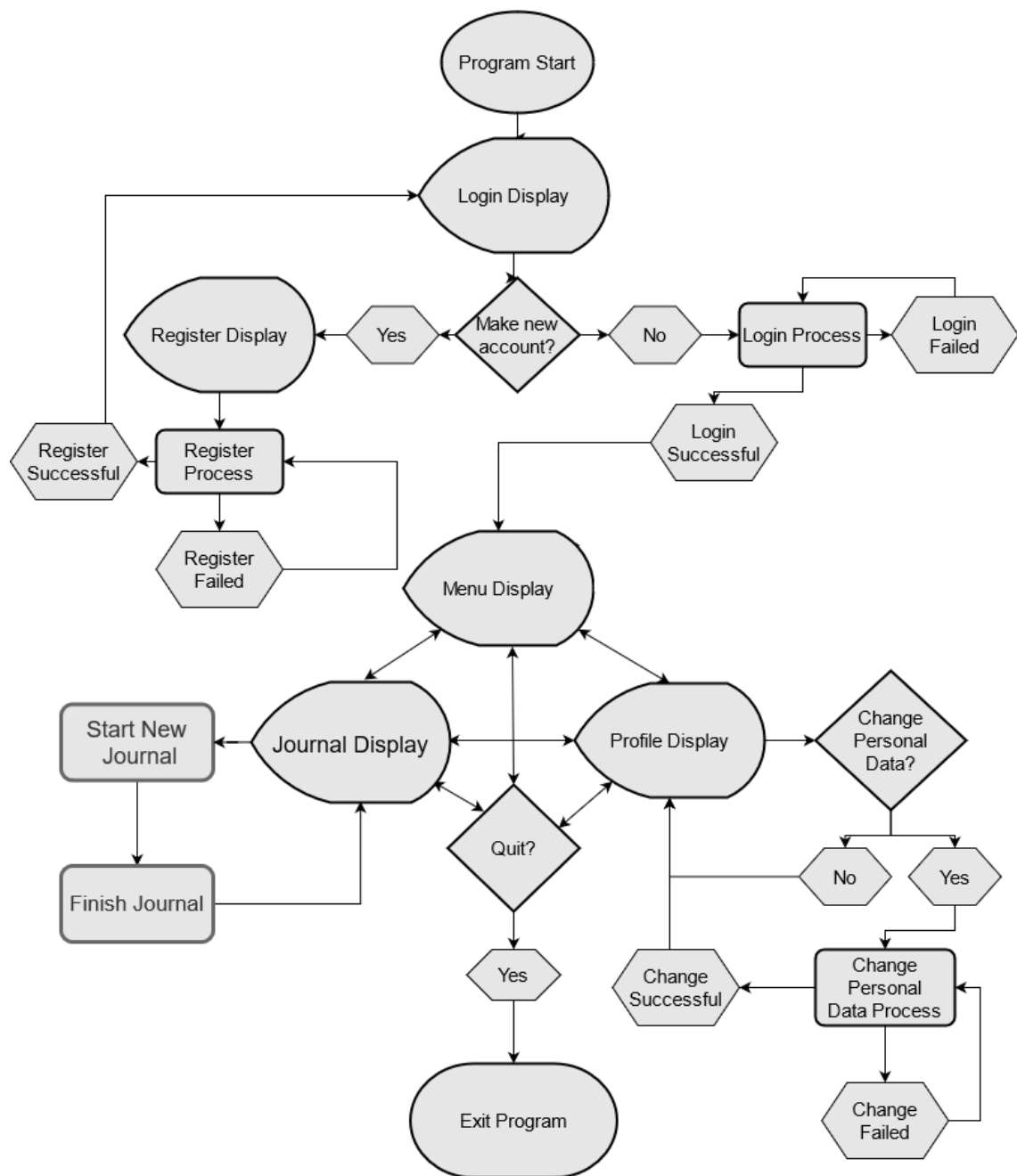
4.1.5. Profil

A Profil (Profile) nézetben van lehetőség megváltoztatni a személyes adatainkat és a céljainkat.



4.5. Profil ablak terv

4.2.Folyamatábra



4.6. Folyamatábra

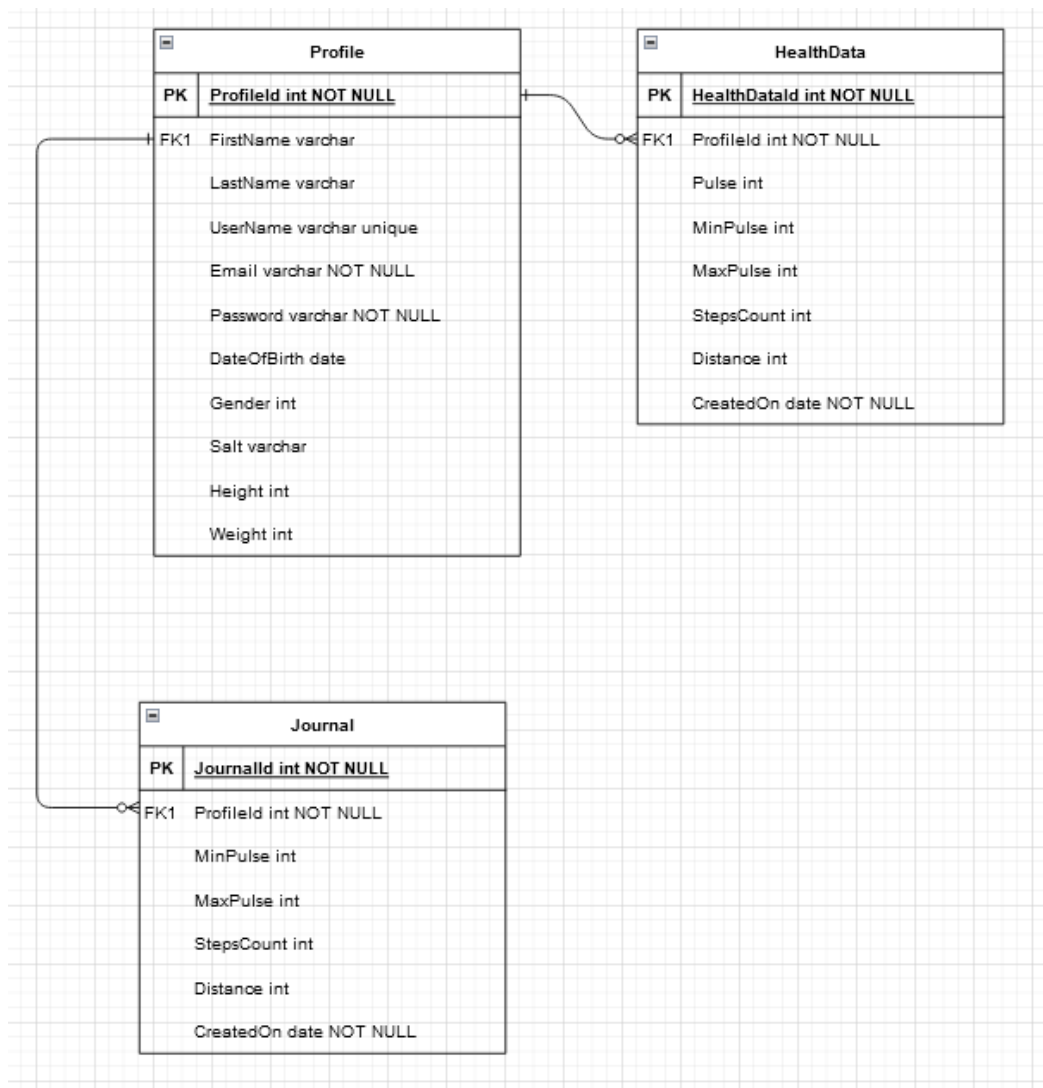
A program indításakor a bejelentkező ablak köszönt minket, ahol lehetőségünk van a már létező profilunkba belépni, vagy pedig egy teljesen újat készíteni a regisztráció gombra kattintva. A regisztrációs ablakon az adataink megadása és a „SIGN UP” gombra való

kattintás után visszakerülünk a bejelentkező ablakra, hogy beléphessünk az alkalmazás fő részébe. Itt a napi mért adataink láthatóak. A fő ablakon lehetőség van még átlépni két másik ablakra is, a „Journal” és a Profil fülekre. A „Journal” fülben lehetőségünk van utazásokat felvenni, aminek a lényege, hogy nem az egész napos adatokat láthatjuk, hanem kedvünk szerint indíthatunk méréseket. A Profil ablakra átlépve személyes adatainkon változtathatunk, mint például a súlyunk vagy a magasságunk. A három ablak között bármikor válthatunk, bármelyiket megtekinthetjük, kivéve akkor, amikor éppen egy utazást indítottunk, mert az egyfajta felugró ablakként fog funkcionálni.

4.3. Projektek terve

A projekt felbontható 3 részre. Az első az android alkalmazás, amivel a felhasználó interakciókat kezdeményezhet. Itt van lehetőség a profil létrehozására és az adatok megtekintésére. Ez egy fogyasztó/szolgáltató kapcsolatban áll a hordozható (wearable) alkalmazással, ami az okosórán futó alkalmazás. Az adatbázis egy SQL adatbázis, amit egy ASP.NET szerverrel használok fel. Itt történnének az kellő számítások, konvertálások, és az adatok tárolása.

Az első lépés az adatbázis megvalósítása lenne. Itt felmerült, hogy az adatbázis szimplán egy Firebase vagy valamilyen más felhő szolgáltatással oldjam-e meg, de végül egy SQL adatbázis mellett döntöttem, amivel korábbi tanulmányaim alatt már használtam ilyet és hozzám legközelebb állt. Három táblából állna az adatbázis, amivel reprezentálni tudom a felhasználói profilokat, az óráról mért adatokat és az egyes edzéseket, amiket a Journal fülben lehet megtekinteni. A táblákat először Microsoft SQL Server Management Studio keretein belül készítettem el, majd „SQL first” megközelítéssel importálnám be a projektbe a modelleket és az adatbázis kontextusát. A rendszert rétegzett projektekké készíteném el, ami áll a Modell, Repository, Logic és Controller rétegekből. Api hívásokkal elérhetőek az egyes modell típusokhoz CRUD és egyéb műveletek.



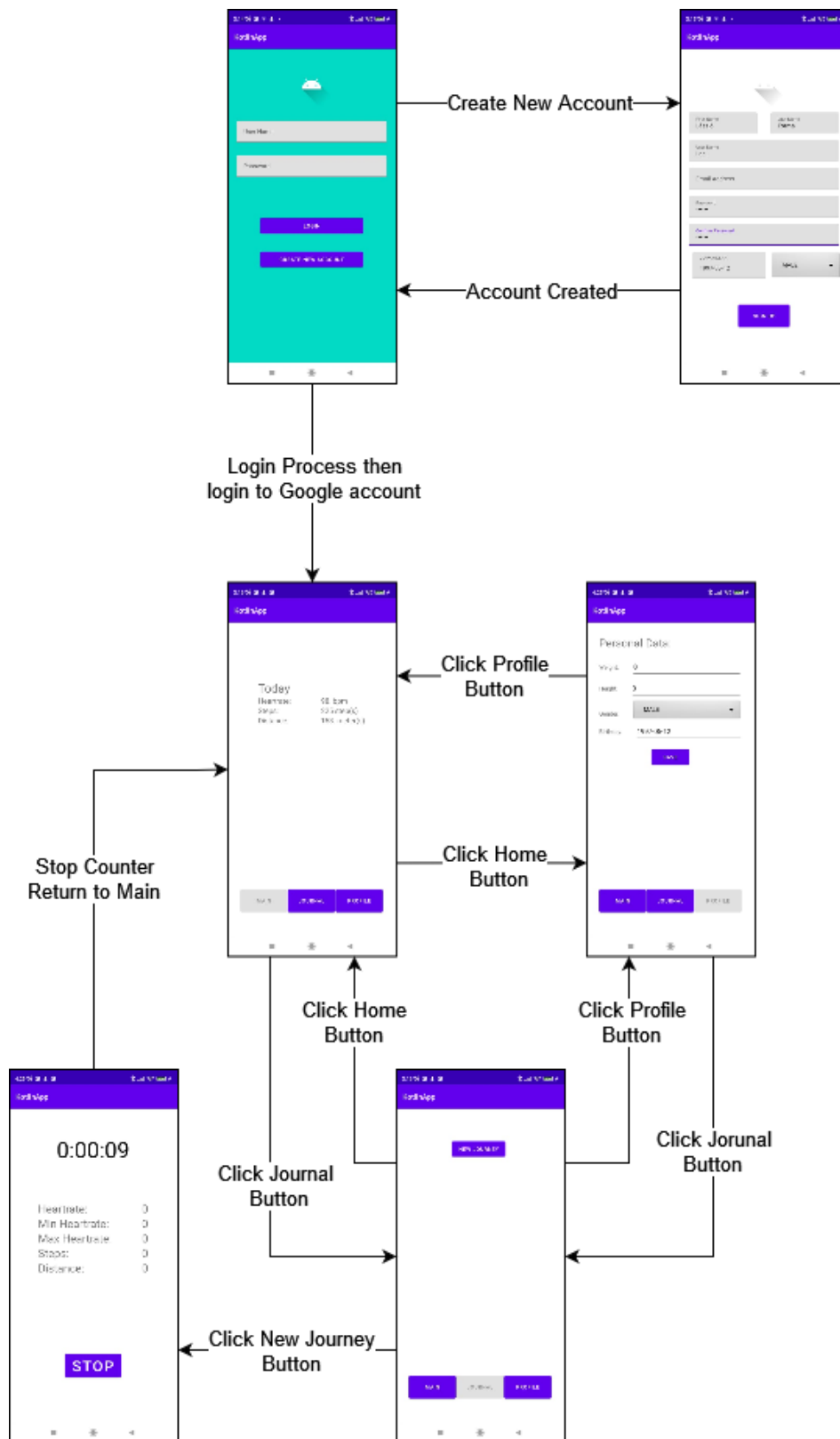
4.7. Adatbázis felépítés osztálydiagram szemléltetéssel

A második lépés ezután az android alkalmazás elkészítése lenne. Felmerült a megvalósításhoz több ötlet is. Elkészíthettem volna a projektet Xamarin nyelven Visual Studioban, vagy Java és/vagy Kotlin nyelven Android Studioban. Munkahelyi tapasztalatok és kutatásaim alapján végül a Kotlin nyelvet választottam, ebből kifolyólag Android Studioban valósítanám meg. A Kotlin nyelv előnye, hogy egyszerre használhatom a projektben a Kotlin nyelven írt osztályokat a Java-ban megírtakkal, jelentősen megkönnyítve a munka haladását. Api hívásokkal tud kapcsolatot létesíteni az applikáció az ASP.NET projekttel. Az Api hívásokat a Retrofit típusbiztos REST klienssel valósítanám meg, ami az OkHttp kliens segítségével kezeli a hálózati kéréseket. A felhasználói interfész négy tevékenységből és három „töredék” nézetből áll majd. A Bejelentkező tevékenységről lehetőségünk lesz a regisztrációs tevékenységbe átlépni és regisztrálni egy új profilt. Ezután visszakerülnénk a bejelentkező földre, ahol a

Szükséges engedélyek az android alkalmazás számára:
Bluetooth és Bluetooth admin engedélyek
Samsung Accessory Framework engedély
Internethozzáférés

4.1. táblázat Tervezett engedély kérések

felhasználónevünket és jelszavunkat beírva hozzáférhetünk a program többi részéhez. Bejelentkezés után bekerülünk a Fő tevékenységbe, ahol minden funkció megjeleníthető a három „Töredék” (Fragment) között váltogatva. A Töredékek lényege, hogy nem hozok létre folyton új tevékenységeket, hanem csak egyen belül változtatom azt, hogy mi jelenjen meg, egy FragmentContainerView belsejében, így dinamikusabbá téve a felhasználói interfészt. A kezdő („Main”) töredéken szeretném megjeleníteni a napi mért adatokat. A „Journal” töredéken lehetőség lenne egy gombra kattintás után átjutni egy új tevékenységre, ahol egy edzést indíthatok, aminek az adatait lementhetem az adatbázisba. Ha be szeretném fejezni az edzést, akkor ezt gombnyomásra megtehetem, és az visszadob a Fő tevékenységre. Az utolsó („Profile”) töredéken pedig a belépett profil adatait szeretném tudni megváltoztatni. Nem minden adatot, csak néhányat úgy, mint a legtöbb fitnessz applikációban lehetséges.



4.8. Az android applikáció drótvázlata

A harmadik lépés az okosóra programja lenne. A Samsung Developers weboldalán több implementációs megoldás is rendelkezésemre állt. Az első, hogy egy a Visual Studio és Android Studio mellett egy harmadik IDE felhasználásával, Natív-an, Tizen Studio-val, C nyelven készítsem el. A második pedig, hogy HTML és Javascript nyelven de Web-es környezetben. Ezekhez az SDK letöltésekor még példaprogramok is jártak, amik bemutatják az üzenetküldési, avagy a fájlküldési folyamatot. A harmadik megoldás a legújabb a három közül. .NET-ben Visual Studio felhasználásával is elkészíthető az alkalmazás a „Visual Studio Tools for Tizen” bővítmény letöltése után Xamarin.Forms típusú projekt formájában. Mivel a .NET nyelvet már ismertem és tapasztaltabb vagyok benne, mint a Javában, emiatt azt a megoldást részesítem előnyben. A hordozható szolgáltató applikáció és az android fogyasztó applikáció közötti kapcsolatért az SAAgent (Accessory Peer Agent) felel. Mindkét projektnek implementálnia kell egy Accessory Service Profile-t globálisan egyforma id-val és „servicechannel” beállításokkal, hogy ez használható legyen. A működésének a lényege az, hogy az Android applikáció Ügynöke (Agent) küld egy szolgáltatás kapcsolat (service connection) kérést egyező külső Ügynök számára, majd visszajelzésre vár, hogy sikeres volt-e a kapcsolatkerés. A hordozható applikációnak le kell reagálnia az egyes bejövő csatlakozási kérelmeket és el kell döntenie, hogy elfogadja-e azokat vagy sem. Ha a kapcsolat létrejött, akkor mindkét Ügynök visszajelzést kap és egy SASocket (szolgáltatás kapcsolat a két Ügynök között) objektum által. [26]

Az alkalmazások verziókezelése Github oldalon került megvalósításra.

4.4. Fejlesztés és a közben felmerülő tapasztalatok

A munkám kezdetekor először az adatbázis oldalról közelítettem meg a projekteket. Létrehoztam a Microsoft SQL Server Management Studio -ban a három táblát megfelelő elsődleges és idegen kulcsokkal. Így a Profile tábla két darab „one-to-many” kapcsolattal rendelkezett. Ezután létrehoztam az ASP:NET szerver projektet, és beimportáltam egy „Scaffold” paranccsal az entitásokat. Ezután implementáltam egy rétegzést. Elkészült a Repository és a Logic réteg plusz mellette a kontrollerek. A munkafolyamat egy későbbi részében készült el még egy Modell réteg, mikor az egyes Api hívások esetén végtelen mély listák kerültek visszaadásra, mert az Entity Framework-ben, ha az entitásokkal dolgozik az ember modellek helyett, akkor az entitások folyamatosan visszareferálnak önmagukra. Ez az esetben úgy nézett ki például, hogy a profilnak volt egy edzés listája és minden edzésnek egy profilja, aminek volt egy edzés listája és így tovább. Ezt a Modell réteggel sikeresen áthidaltam.

A Repository és a Logic rétegek szimpla CRUD metódusokat implementálnak a modellek alapján. Létrehoztam még egy autentikációs logikát, amiben a regisztrációs és a bejelentkező logikát készítettem el. A regisztrációnál a beérkező jelszó egy véletlenszerűen generált szöveg fájlal lett Hash módszerrel titkosítva. Két kontrollert

készítettem még el. Az egyik az autentikációval foglalkozik, tehát az új profil regisztrációjával és az adatbázisba lementésével, valamint a bejelentkezéssel. A bejelentkezés során megérkezik az android applikációról egy felhasználónév és egy titkosítatlan jelszó. A profil létrehozásakor elmentődik még a véletlenszerűen létrehozott szövegfájl, amit Hash-eléshez használt a program, így ezt újra felhasználva meg tudja nézni, hogy ugyanaz a titkosított jelszó készül-e el az érkező jelszóból, mint ami a regisztrációnál generálódott. A másik kontroller a további a további Api hívásokat reagálja le, mint például egy profil adatainak lekérése, vagy pedig egyes edzések lementése.

Az android alkalmazás fejlesztése Android Studioban zajlott főként Kotlin nyelven. Először több tevékenységet (activity) hoztam létre, amik között tudtam gombnyomásra váltogatni. Miután kicsit jobban átláttam a dolgokat és jobban elmerültem az android fejlesztésben, rájöttem, hogy jobb lenne kevesebb tevékenységgel dolgozni, és ahol lehet váltsak töredékekre (Fragment). Így nem kellett minden oldalon gombokat implementálnom az egyes nézetek között való váltáshoz, elég volt egyen megtennem ezt. A tesztelés kezdetekor kiderült az Android Studionak és az emulatornak magasabb hardverigényei vannak, mint amivel a saját gépem rendelkezik, így számítógépet váltottam. Ekkor még úgy gondoltam, hogy az ASP.NET szerver maradhat az első gépemen. Ezután elkezdtem kísérletezni a Retrofit klienssel, hogy hogyan lehet a szerverről adatot szerezni, avagy egyáltalán sikerül-e megteremtenem a kapcsolatot. Nem sikerült több próbálkozásom ellenére sem ez, és egy internetes fórum tanácsa alapján, miszerint a Windows tűzfal lehet egy lehetséges hibaforrás, a szervert is átmozgattam a másik gépre. Folytattam így a tesztelést android emulátorral és próbáltam a lokális host szerverről adatot szerezni. Sok sikertelen próbálkozás után kiderült, hogy az android csak úgy tud csatlakozni a lokális szerverhez, ha a szerver a `http://127.0.0.1/port` URL-en fut, az android pedig a `http://10.0.2.2/port` URL-en. Ez még nem volt elég, különböző csomag problémák kiküszöbölése után kellett készítenem egy „`network_security_config.xml`” fájlt [4.8], amire referenciát kellett tennem az android applikáció manifest fájljában.

```

<network-security-config>
  <base-config cleartextTrafficPermitted="true">
    <trust-anchors>
      <certificates src="system" />
    </trust-anchors>
  </base-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="true">192.168.101.150:5000</domain>
    <domain includeSubdomains="true">10.0.2.2</domain>
    <domain includeSubdomains="true">localhost</domain>
    <domain includeSubdomains="true">127.0.0.1</domain>
  </domain-config>
</network-security-config>

```

4.9. Network Security Config fájl

Ezután sikerült megvalósítani a regisztrációt, a bejelentkezést és a profil adatainak lekérését és megváltoztatását. Itt megemlíteném, hogy a kódban van még egy működő stopper implementálva az új utazás elkészítéséhez, de ez az ablak nem készült el teljesen.

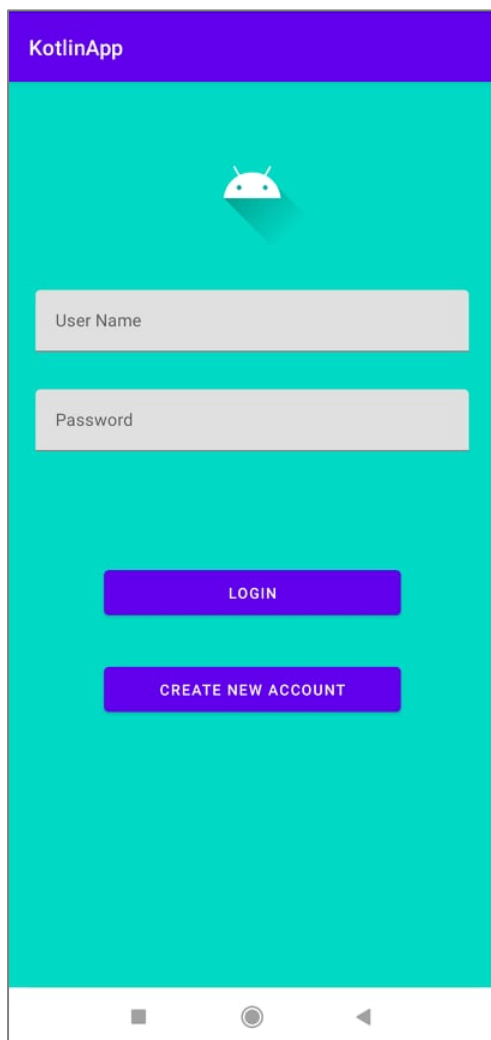
Sikertelenségeim hatására megpróbáltam más irányból megközelíteni a feladatom, és elkezdtem az óra programján dolgozni. A Visual Studioban létrehoztam egy Tizen Xamarin Forms projektet, és C# nyelven elkezdtem az android és az óra közötti kapcsolaton. Ez a projekt az első gépemén futott. A kapcsolat felvételéhez a Samsung által szolgáltatott Samsung Accessory SDK-t használtam, amihez a dokumentáció a SAAgent kliens implementálását javasolta, mint az órán és az android-on. Először ezt is emulátorral kezdtem el tesztelni Tizen 6-os API szinten. Az android applikáció kódját ekkor már kitelepítettem a készülékemre, hogy ne fussak bele ismét az esetleg felmerülő tűzfal problémába. Ennek ellenére nem sikerült kapcsolatot kialakítani. Megpróbáltam ekkor kitelepíteni az órára a kódot, ami nem volt egyszerű különböző okok miatt, de miután mobilnetre váltottam, és levittem az API szintjét az óra kódjának 5.5-re (ez a szint futott a fizikai eszközön), utána már futott a program. A kapcsolat ennek ellenére sem alakult ki. A megoldás után kutatva azt találtam, hogy még alacsonyabb API szinten volt, akinek sikerült. Viszont az az órán már nem futott volna el, így visszaváltottam emulátorra. Levittem tehát az API szintet négyre. Itt viszont a kód egy „Fragmentation fault” hibaüzenettel folyton abbahagyta a működést, és ezt próbálkozásaim ellenére sem sikerült kiküszöbölni. Emiatt az óra kódját nem tudtam elkészíteni.

Alternatívaként, hogy tudjak adatot szerezni, amit megjeleníthetek az android applikáción bizonyítva, hogy működik, a Google Fit api-jához fordultam. Az api History Client-ének segítségével sikerült hozzáférnem limitált mennyiségű az android eszköz által mért adathoz, mint például a napi lépcsőszám és a napi megtett táv. Hogy hozzáférjek a Google

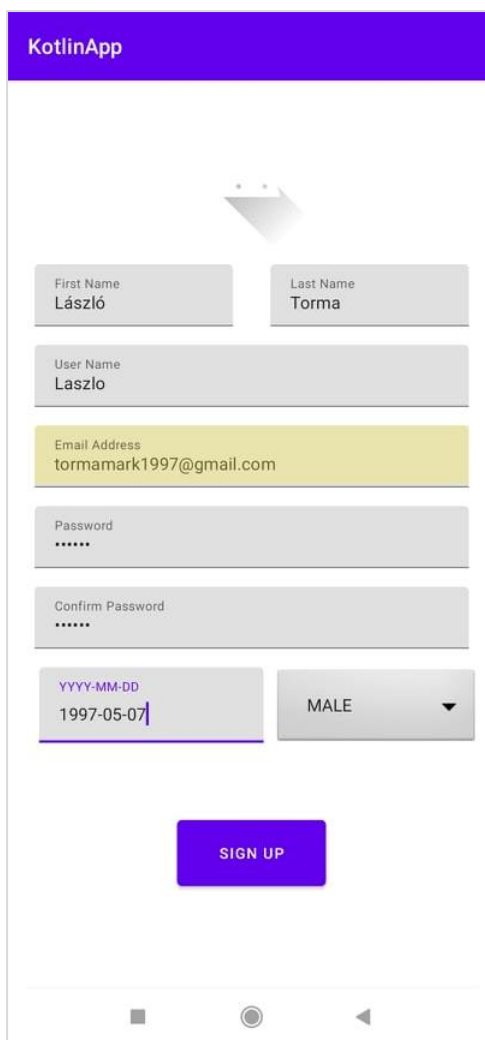
Fit-en mért adatokhoz készítenem kellett egy egyszerű Google bejelentkezést az android applikáción belül, ami a saját bejelentkezési folyamat után történik a Fő tevékenységben. Utánajártam, hogy hogyan lehetne még más adatokat is szerezni, és azt találtam, hogy a Google Fit és a Samsung Health szoftvereket szinkronizálni lehet egy Health Sync nevű alkalmazás használatával. Ezzel ugyancsak limitáltan, de hozzáfértem a Samsung Health-ben tárolt pulzus értékekhez is. Ami miatt szükség volt erre az egész szinkronizálásra az az volt, hogy a Google Fit nem tudott adatot lekérni a Samsung Galaxy Watch Active 2-es okosórámról, viszont a Samsung Health pedig igen. Azért nem a Samsung Health-el kezdtem el dolgozni a Google Fit helyett, mert a Samsung Health api-ja csak partner csoportoknak elérhető, a partnerség viszont jelenleg nem lehetséges, legalábbis egy olyan magánszemélynek, mint én. Végeredményben sikerült adatot szereznem az óráról.

4.5. Tesztelés

A tesztelési folyamat első lépése az új profil létrehozása. A Bejelentkező ablakból át is léphetünk a regisztrációs ablakba „Create New Account” gombra kattintva.



4.10.Bejelentkező ablak indításkor



4.11.Regisztrációs ablak kitöltött adatokkal

A „SIGN UP” gomb lenyomása után leellenőrizhetjük a regisztráció sikerességét az adatbázisban is.

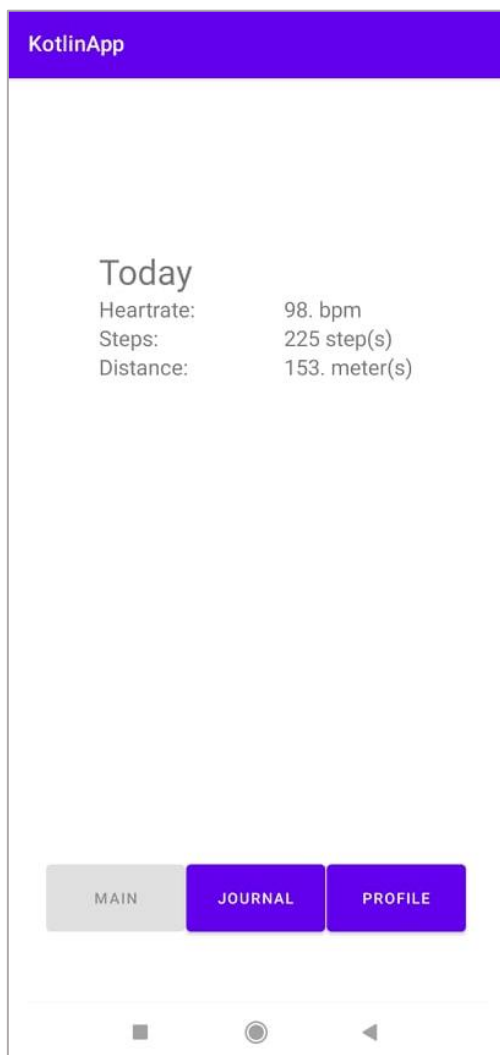
3	Torma	László	TormaLá	taszlortorma@gmail.com	q33vLU0GEXNtWEINtQAKZKKK0YtASWttnQ9dArltW3C3i3U=	1997-05-07	0	jujgzeziw0JALDrltU0pntltu0A==	NULL	NULL
4	Kerekes	Bence	BenceUr	bencekerekes@gmail.com	OCAD+whyBVltUptizS5avm1A85jKUU7wrKhVCan022Q=	1992-11-17	0	zRrGOVMql8jG13yPpFB4eA==	NULL	NULL
5	F	F	F	F	hgDRKBjKRSgZRDkAzhwoSC+mCgCKrJ+7sZQISfwpVWM=	1987-05-04	0	cS6ES42mkd5PgYQLCskqoQ==	NULL	NULL
6	László	Torma	Laci	tormamark1997@gmail.com	hIR48PGY0qFhmIl2zEIEAGcyHtmfSeKX08YbUDmyew=	1997-05-07	0	qEhdb0FsYurzn/AaD0tj/A==	NULL	NULL
7	Torma	László	László	tormamark1997@gmail.com	RhgU76b62tnBSzuU8hnbzCnjlS0SOpIWeq0UDzcZDlk=	1997-05-12	0	dYrgu7TDMoGJU3hVCtuSQ==	NULL	NULL

4.12.Hetes számú Profil regisztrációja után

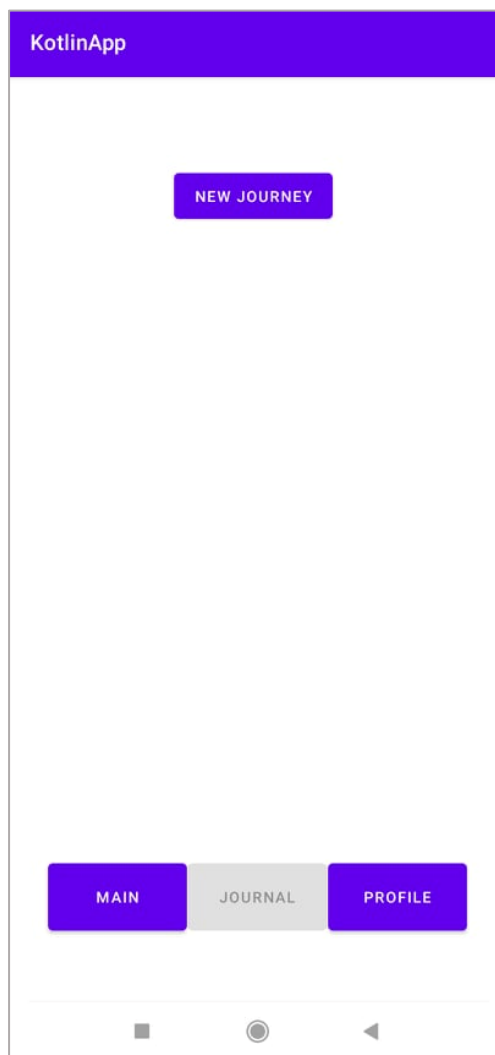
4	Kerekes	Bence	BenceUr	bencekerekes@gmail.com	OCAD+whyBVltUptizS5avm1A85jKUU7wrKhVCan022Q=	1992-11-17	0	zRrGOVMql8jG13yPpFB4eA==	NULL	NULL
5	F	F	F	F	hgDRKBjKRSgZRDkAzhwoSC+mCgCKrJ+7sZQISfwpVWM=	1987-05-04	0	cS6ES42mkd5PgYQLCskqoQ==	NULL	NULL
6	László	Torma	Laci	tormamark1997@gmail.com	hIR48PGY0qFhmIl2zEIEAGcyHtmfSeKX08YbUDmyew=	1997-05-07	0	qEhdb0FsYurzn/AaD0tj/A==	NULL	NULL
7	Torma	László	László	tormamark1997@gmail.com	RhgU76b62tnBSzuU8hnbzCnjlS0SOpIWeq0UDzcZDlk=	1997-07-20	1	dYrgu7TDMoGJU3hVCtuSQ==	80	180

4.13.Hetes számú profil adatváltoztatás után

A 4.9-es ábrán láthatjuk, hogy sikeres volt a regisztráció. Regisztráció után visszakerülünk a bejelentkező ablakba, ahová felhasználónevünket és jelszavunkat beírva megérkezünk a Főmenübe, ahol a napi mért adataink láthatóak. Jelenleg a program bejelentkezés után elindít egy Google bejelentkezési folyamatot azért, hogy megkapja a felhasználotól az engedélyt egy Google fiók használatához, és vele az ahhoz tartozó Google Fit adatok lekéréséhez. Ezután sikeresen megjelennek a mért adatok.

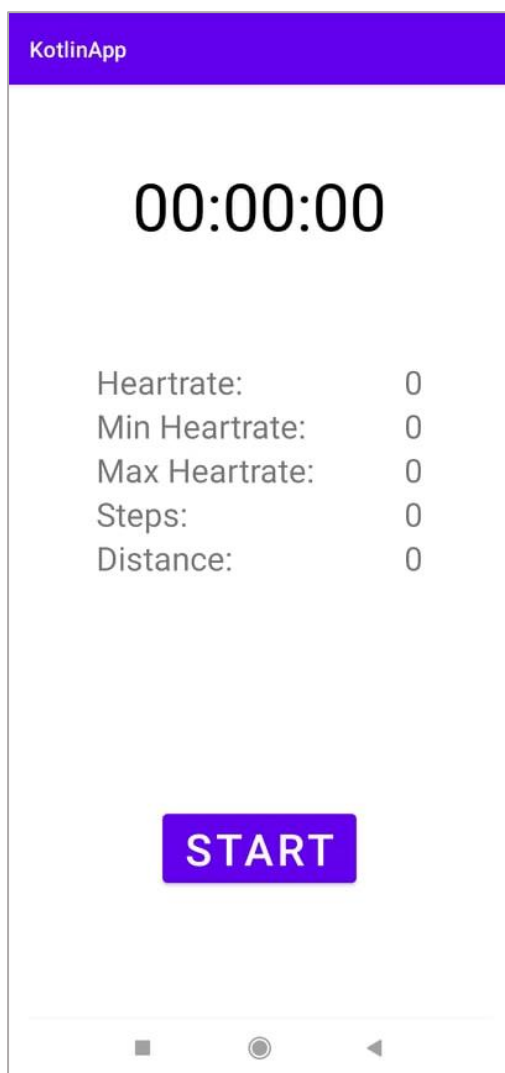


4.14. Bejelentkezés után a Főmenü ablak

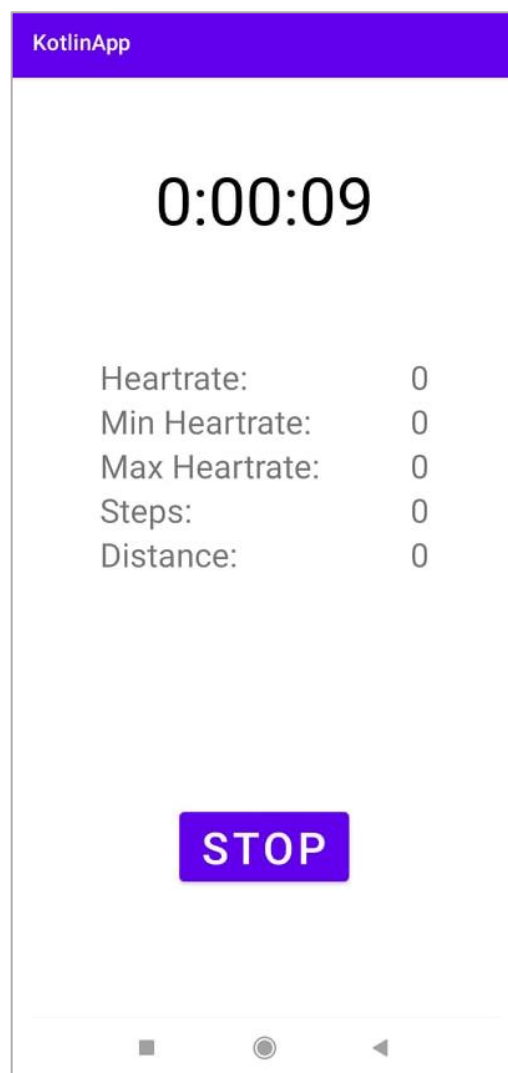


4.15. Journal ablak

A Journal ablakra átkattintva egy gombot láthatunk, amivel átmehetünk az új Utazás létrehozása ablakba.



4.16.Új utazás ablak stopper indítás előtt



4.17.Új utazás ablak stopper indítás után

Az új utazás ablakon lett volna lehetőség alkalmi adatokat felvenni és azokat elmenteni az adatbázisba, viszont a Retrofit, a szerverről érkező Json-okat átkonvertálja objektumokká, így nem tudtam adatot menteni és kinyerni. A kinyerést a Journal ablakon (4.13) szerettem volna megvalósítani. Az adatszerzést az Új utazás képernyőn pedig, mérési adathiány miatt nem sikerült megvalósítani.

Az utolsó rész, ami még a program része a Profil nézet. Egy frissen létrehozott profil csak születési dátummal és nemmel rendelkezik. A „Save” gomb lenyomásával van lehetőség adatot frissíteni az adatbázisban. A frissítés sikerességéről a 4.11-es kép tanúskodik.

KotlinApp

Personal Data:

Weight: 0

Height: 0

Gender: MALE

Birthday: 1997-05-12

SAVE

MAIN JOURNAL PROFILE

4.18. Frissen létrehozott profil Profil nézete

KotlinApp

Personal Data:

Weight: 80

Height: 180

Gender: FEMALE

Birthday: 1997-07-20

SAVE

MAIN JOURNAL PROFILE

4.19. Frissített profil Profil nézete

4.6. Hasonló rendszerekkel való összehasonlítás

Az eredeti tervekhez viszonyulva az volt az ötlet, hogy egy a Google Fit-hez hasonló alkalmazást készítek. Tekintve, hogy az egy elég kényelmes és letisztult program, igyekeztem az alkalmazáshoz hasonló ablakokat és mérési funkciókat készíteni.

A Stravához képest annyi a különbség, hogy a program jelenlegi verziója inkább az egészségügyi adatok lekérésére fókuszál napi vagy bizonyos időn belüli szinten. Nem azon van a fókusz például a Journal fülben, hogy merre tettük meg az utat, hanem hogy

mit történt az alatt az út alatt. Ezen kívül hiányzik a programból az ilyen alkalmazások közösségi oldala, amire a Strava nagy hangsúlyt fektet. A program egyelőre csak egy ember saját adatait tudja megmutatni és nem lehet másokkal alkalmazáson belül versengeni.

5. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Többféleképpen is lehetne fejleszteni a programot a jövőben. Első sorban az adatbázis lecserélése lenne a fő cél, mert teszteléshez tökéletes az ASP.NET webszerver, de ha nem fut egy számítógépen a szerver, akkor a program nem használható, továbbá egy bizonyos felhasználószám után már inkább hátrányosabb a használata. Ezért előnyös lenne lecserélni egy felhő szolgáltatóra. Saját lehetőségeimet figyelembe véve, és anyagi szempontból megközelítve a legkézenfekvőbb megoldás a Firebase szolgáltatásai lennének, mert azok ingyenesek, így bármikor egy webszerver futtatása nélkül is használhatóvá válik a program.

További adatok lekérése, avagy több egyéb funkció implementálása lehet még fejlesztés, mint például edzéstervek, vagy hogy az új Journal létrehozásakor jelenjen meg térkép és lehessen rajta útvonalat követni és az is mentődjön le. Ez lényegében már automatikus funkcióként jelen van a GoogleFit alkalmazásban.

A Stravához hasonlóan lehetne még benne valamilyen közösségi funkció, mint például a Stravánál a szegmensek, amiket az emberek egymással össze tudnak hasonlítani, és úgy kihívásokat állítani egymás elé.

A Garmin Mobile alkalmazásban szélessávú választék lehetséges testreszabhatóságra. Több adat lekérése esetén implementálásra kerülhet egy új ablak, amiben személyre szabható lehetne az, hogy az egyes ablakokban milyen adatok jelenjenek meg

UI szépítése az Android alkalmazáson belül.

Megoldást keresni arra, hogy fusson a hordozható (óra) programja API 5.5-ös verzióval, hogy lehetséges legyen a kitelepítés rá.

Lehetőség a Google Bejelentkezés implementálása, amivel a regisztrációs folyamat tehető sokkal egyszerűbbé, mert onnan lekérhetjük a felhasználó adatait, amivel elkészíthető egy profil az alkalmazáson belül.

6. ÖSSZEFOGLALÁS

A szakdolgozatom témája egy olyan android alapú applikáció fejlesztése volt, amely az egyes fitness adatokat egy érzékelővel felszerelt okosóráról szerzi be. Ehhez először

irodalomkutatás végeztem és megvizsgáltam, milyen hasonló alkalmazások is vannak már a piacon. Összehasonlítottam, több nagynevű applikációt, ezzel ötleteket merítve a feladatom megoldásához. A következő lépés számomra az volt, hogy kiderítsem, hogyan fogom tárolni az összegyűjtött adatokat. A legtöbb ilyen alkalmazás már a felhőben tárolja a felhasználók adatait, viszont a célom az volt, hogy gyorsan és hatékonyan tudjak tesztelni, ezt pedig csak egy olyan rendszerrel tudtam, amit jól ismerek. Ezután megvizsgáltam milyen titkosítási lehetőségek vannak, amikkel a felhasználók adatait védeni lehet. A Hash kódolt jelszó alapú bejelentkezés a leggyakoribb a piacon, így én is emellett döntöttem.

Az android alkalmazás fejlesztése volt a szakdolgozat fő pontja. Egy fitness alkalmazás célja az, hogy a felhasználó könnyen hozzáférjen a saját egészségügyi, avagy sporttal kapcsolatos adataihoz. Ehhez egy felhasználóbarát és könnyen átlátható kezelőfelületre van szükség, ami pont annyi adatot biztosít, amennyire egy embernek szüksége van. Az ilyen alkalmazásokra a piac hatalmas, és nagyon nehéz kiemelkedő programot készíteni.

Az utolsó rész az okosóra programja lett volna. Az okosórák könnyebben hozzáférnek az emberek egészségügyi adataihoz, viszont a képernyőjük limitációi miatt megéri, ha van mellé egy okostelefon rajta egy alkalmazással, ami kényelmesebbé teszi adataink szemmel tartását.

Végezetül örülök, hogy ezen a projekten dolgozhattam, mert ennek hála sok hasznos tapasztalattal bővílhettem, ami javamra fog válni a jövőben.

7. CONCLUSION

The topic of my thesis was to create an android application, which can gather health information from a smartwatch. To do this, first I researched the other fitness apps that are on the market. I compared many different applications to gather ideas about fitness apps and how they work. The next step was for me to figure out which option I am going to use to store health data. Most fitness apps use cloud to do just that, but my aim was to test features effectively so I decided to use a system that I was familiar with. Then I examined various encryption methods to protect the user's data. The apps that use Hash coded encryption are the most common, so I decided to use that as well.

The development of the android application was the main goal of the thesis. The target of a fitness application is for the user to comfortably access his/her health or sport related data. To make it happen we need a user-friendly and an easy to manage user interface, which provides enough data for anyone in the world. These applications have a huge market, and for that it is tough to make a program that can be outstanding.

The last part was the code for the smartwatch. These watches can more easily access health data, but they have limited space to show data, so it is worth it to pair it with a smartphone, which makes checking or health data more comfortable.

At last, I am happy to be able to work on this project. Thanks to it, I managed to gather a lot of useful experience that will be of use for me in the future.

8. IRODALOMJEGYZÉK

- [1] Abeer Younis Saidam, Yasmeen Talal AbuKwiak: Android Mobile Application For Healthy Fitness (AMAHF) Idézet: <http://dspace.up.edu.ps/jspui/handle/123456789/62>
- [2] Fitnesz alkalmazások összehasonlítása,
(<https://versus.com/en/health-and-fitness-app/best>, utoljára látogatva: 2021.04.22.)
- [3] Garmin Connect Mobile Értékelés,
(<https://buy.garmin.com/en-US/US/p/125677#overview>, utoljára látogatva: 2021.04.22.)
- [4] Garmin Connect Mobile Képek
(<https://buy.garmin.com/en-US/US/p/125677>, utoljára látogatva: 2021.04.27)
- [5] Google Fit Értékelés,
(<https://www.androidauthority.com/google-fit-393110/>, utoljára látogatva: 2021.04.22.)
- [6] Google Fit Képek,
(<https://www.theverge.com/2019/4/24/18513918/google-fit-ios-release-date-features-move-minutes-heart-points-rings>, utoljára látogatva: 2021.04.22.)
- [7] Strava jellemzők,
(<https://www.strava.com/features>, legutóbb látogatva: 2021.04.22.)
- [8] Strava Képek,
(<https://www.businessinsider.com/interview-strava-ceo-james-quarles-social-network-2017-12>, utoljára látogatva: 2021.04.23.)
- [9] Apple Health tulajdonságok,
(<https://appleinsider.com/inside/apple-health>, utoljára látogatva: 2022.05.11.)
- [10] Apple Health Főmenü kép,
(<https://developer.apple.com/health-fitness/>, utoljára látogatva: 2022.05.11.)
- [11] Apple Health alvás figyelő oldal kép,
(<https://www.apple.com/ios/health/>, utoljára látogatva: 2022.05.11.)
- [12] Apple Health Egészségügyi ID kártya kép,
(<https://appleinsider.com/inside/apple-health>, utoljára látogatva: 2022.05.11.)
- [13] Samsung Health tulajdonságok,
(<https://www.androidauthority.com/samsung-health-3037491/>, utoljára látogatva: 2022.05.11)
- [14] Samsung Health képek,
(<https://eu.community.samsung.com/t5/mobil-eszk%C3%B6z%C3%B6k/mi-is-az-a-samsung-health/td-p/3171856>, utoljára látogatva: 2022.05.11)

- [15] Fitbit tulajdonságok és képek,
(<https://www.fitbit.com/gb/app>, utoljára látogatva: 2022.05.11)
- [16] Google Cloud storage értékelés.
(<https://cloud.google.com/storage/>, utoljára látogatva: 2021.11.14.)
- [17] Google Cloud SQL értékelés,
(<https://cloud.google.com/sql/>, utoljára látogatva: 2021.11.14.)
- [18] Microsoft Azure értékelés,
(<https://azure.microsoft.com/en-us/free/cloud-services/>, utoljára látogatva: 2021.11.14.)
- [19] Azure értékelés
(<https://www.pcmag.com/reviews/microsoft-onedrive>, utoljára látogatva: 2021.04.23.)
- [20] FireBase értékelés,
(<https://firebase.google.com/docs/database>, legutóbb látogatva: 2021.11.14.)
- [21] A felhasználói hitelesítési módszerek összehasonlítása három paraméteren,
(<https://www.cyberark.com/resources/blog/comparison-of-user-authentication-methods-on-three-parameters?fbclid=IwAR2C4ymOgWQtstJB-ILM9e5GwJyyN6RDe1OcUZnVJKkCxGjYbf4JgLtnZOs>, utoljára látogatva: 2021.04.23.)
- [22] 5 hitelesítési módszer, amelyek megakadályozhatják a következő jogsértést,
(https://www.idrnd.ai/5-authentication-methods-that-can-prevent-the-next-breach/?fbclid=IwAR1vZmgIfUkKL2JG_qfZBklXOjmXi9gxP_ad9ShOBzwG1Ap6SJGrXGuGRHk, utoljára látogatva: 2021.04.23.)
- [23] Consumer és Provider alkalmazás összehasonlítása,
(<https://kennay-kermani.medium.com/create-a-simple-samsung-galaxy-watch-companion-net-app-part-1-1466fd9e2e6c>, utoljára látogatva: 2021.11.14.)
- [24] Natív és Web alapú alkalmazások értékelése,
(<https://uxmag.com/articles/native-or-web-based-selecting-the-right-approach-for-your-mobile-app>, utoljára látogatva: 2021.11.14.)
- [25] Alkalmazások tárolása a Google Fitneszhez,
(<https://support.google.com/accounts/answer/6098255?co=GENIE.Platform%3DAndroid&hl=hu#zippy=%2Ctev%C3%A9kenys%C3%A9gadatok%2Chely adatok%2Ct%C3%A1pl%C3%A1lkoz%C3%A1s-%C3%A9shidrat%C3%A1ci%C3%B3%2Ctest%C3%A9rz%C3%A9kel%C5%91kkel-kapcsolatos-adatok%2Calv%C3%A1ssal-kapcsolatos-adatok>, utoljára látogatva: 2021.04.21.)

[26] Samsung programozói útmutató pdf,
(<https://developer.samsung.com/galaxy-accessory/programming-guide.html>, utoljára látogatva: 2021.11.14.)

[27] Adatszerzés okosóráról,
(<https://github.com/Samsung/Tizen-CSharp-Samples/tree/master/Wearable/Xamarin.Forms>, utoljára látogatva: 2022.04.24.)

9. ÁBRA ÉS TÁBLÁZATJEGYZÉK

7.1 Táblázatjegyzék

- 2.1. táblázat: Cloud SQL és Cloud Storage összehasonlítása
- 2.2. táblázat: Native és Web alkalmazások összehasonlítása
- 4.1. táblázat Tervezett engedély kérések

7.2 Ábrajegyzék

- 2.1. Garmin Mobile Connect: Menü
- 2.2. Garmin Mobile Connect: Irányítópult
- 2.3. Garmin Mobile Connect: Heti statisztikák
- 2.4. Garmin Mobile: Egészségügyi statisztikák
- 2.5. Google Fit: Főmenü
- 2.6. Google Fit: Napló
- 2.7. Strava: Profil
- 2.8. Strava: Csatorna
- 2.9. Strava: Csatorna 2
- 2.10. Strava: Út közben mérés
- 2.11. Apple Health főmenü és Apple Watch
- 2.12. Apple Health alvás figyelő
- 2.13. Apple Health Egészségügyi ID kártya
- 2.14. Samsung Health közösségi oldal
- 2.15. Samsung Health Fitnessz oldal
- 2.16. Fitbit mai nap mérései [15]
- 2.17. Fitbit eheti lépésszámok [15]
- 9.1. Regisztrációs ablak terv
- 4.2. Bejelentkező ablak terv
- 4.3. Menü ablak terv
- 4.4. Napló ablak terv
- 4.5. Profil ablak terv
- 4.6. Folyamatábra
- 4.7. Adatbázis felépítés osztálydiagram szemléltetéssel
- 4.8. Az android applikáció drótvázlata
- 4.9. Network Security Config fájl
- 4.10. Bejelentkező ablak indításkor
- 4.11. Regisztrációs ablak kitöltött adatokkal
- 4.12. Hetes számú Profil regisztrációja után
- 4.13. Hetes számú profil adatváltoztatás után
- 4.14. Bejelentkezés után a Főmenü ablak
- 4.15. Journal ablak
- 4.16. Új utazás ablak stopper indítás előtt
- 4.17. Új utazás ablak stopper indítás után

4.18.Frissen létrehozott profil Profil nézete

4.19.Frissített profil Profil nézete