



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Távközlési és Médiainformatikai tanszék

Tanyi Szvetlin

# **Intelligens lakópark, hardvertől a szolgáltatásokig**

## **TUDOMÁNYOS DIÁKKÖRI DOLGOZAT**

KONZULENS

**Dr. Varga Pál**

BUDAPEST, 2018

# Tartalomjegyzék

Kivonat .....	4
Abstract .....	5
<b>1. Bevezetés .....</b>	<b>7</b>
<b>2. Rendszerek rendszere – az intelligens lakópark.....</b>	<b>10</b>
2.1 Célkitűzés .....	10
2.2 Az Arrowhead keretrendszer.....	10
2.3 Green-IoT alapú okos otthonok.....	14
2.4 Lakóparkok közös erőforrásai .....	16
2.5 Lakópark üzemeltető .....	16
2.6 Nyilvánosság felé megosztott adatok .....	17
2.7 Hatóságok.....	17
2.8 Intelligens lakópark lokális felhői és szenzorokkal való kapcsolattartás .....	18
<b>3. Tervezés.....</b>	<b>18</b>
3.1 Green-IoT keretrendszer továbbfejlesztése .....	18
3.2 Mobil alkalmazás .....	20
3.3 Arrowhead grafikus felhasználói felület .....	21
3.4 A csomóponti szenzorillesztő nyomtatott áramköre .....	21
3.5 Költségbecslés .....	23
3.6 Közösségi terek, biztonsági kamerák .....	25
3.7 Szerverek költsége.....	25
3.8 Szoftverek költsége .....	25
<b>4. Implementáció és kivitelezés .....</b>	<b>26</b>
4.1 Green-IoT API implementációja .....	26
4.2 Docker .....	28

4.3	Arrowhead front-end fejlesztése .....	28
4.4	Mobil alkalmazás .....	31
4.5	A Relay csomópont nyomtatott áramkörének elkészítése.....	36
<b>5.</b>	<b>Vizsgálatok és eredmények.....</b>	<b>42</b>
5.1	Unit tesztek.....	42
5.2	Integration tesztek .....	43
5.3	Manuális tesztek Postman segítségével.....	43
5.4	Nyomtatott áramkör tesztje .....	44
5.5	Lakópark rendszereinek összekapcsolt működése .....	44
<b>6.</b>	<b>Kiberbiztonság.....</b>	<b>45</b>
<b>7.</b>	<b>Jövőbeni tervek.....</b>	<b>46</b>
7.1	Mobil alkalmazás továbbfejlesztése.....	46
7.2	A Relay csomópont áramkörének továbbfejlesztése.....	46
7.3	Arrowhead front-end továbbfejlesztése .....	46
7.4	3D nyomtatott egyedi tok .....	46
<b>8.</b>	<b>Összefoglalás .....</b>	<b>47</b>
	<b>Irodalomjegyzék .....</b>	<b>48</b>

## Kivonat

A Gartner tavalyi 2017-es jelentésével ellentétben, az idei 2018-as jelentésében [1] az okos otthonokkal kapcsolatos megoldások sokkal előrehaladottabb állapotban vannak és realiztikusabb célokra fókuszálnak. Viszont az idei jelentésben a görbe felívelő ágán vannak az Okos Munkahelyekkel kapcsolatos megoldások. Ezekhez nagyon közel állnak az Okos Lakóparkok, mivel hasonló problémákkal kell megküzdenie az üzemeltetésnek a mindennapok során, csak az emberek a munkahelyekre főleg nappal, míg haza este járnak.

Különféle innovatív vállalatok kísérleteznek azzal lakótelepeken, hogyan tudnák meglévő okos otthon megoldásaikat összekapcsolni, ezáltal egy könnyebben fenntartható élőkönyezetet teremteni az ott lakók számára. Ez még korántsem egy kiforrott terület, mivel a felhasználók sem biztosak abban, hogyan és mire szeretnék használni, ezáltal a vállalatok sem tudták még tökéletesen megjósolni a fejlesztés jövőbeli irányát. Az általam tervezett Okos Lakópark ilyen szempontból próbál kitűnni: előremutató felhasználási területeket és továbbfejlesztési irányokat mutat be. Mindemellett természetesen az egyes lakások adataihoz csak az adott lakás lakói férnek hozzá, ez biztosítja azt, hogy senki ne férhessen hozzá a beleegyezésükön kívül. A lakások engedélyével az üzemeltetés felé megosztott adatok nagyban hozzájárulnak a lakóközösség életszínvonalának emelésében.

Az általam tervezett Okos Lakópark alapként tekint az általam múlt évben bemutatott Green-IoT okos otthon keretrendszerre. A keretrendszert a társasházak lakásaiba integrálva azok önálló okos otthonokként tudnak működni, viszont összekapcsolva az üzemeltetés felé tudnak szolgálni hasznos jelzésekkel esetleges problémák fennállása esetén, legyen az csőtörés, tűzeset, vagy netán betörés. Így jelentősen csökkenthető a reakcióidő és az esetleges kár is.

Külön figyelmet kapott az Okos Lakópark fel nem használt erőforrásainak külső hasznosítása, legyen szó a napkollektorok által megtermelt áram eladásáról, vagy kihasználatlan parkolók bérbe adásáról. Ebben játszik kulcsfontosságú szerepet az Arrowhead keretrendszer [2], amely egy olyan nagy európai innovációs konzorcium által fejlesztett, szabványosítás alatt álló keretrendszer, amelyet várhatóan mindennapjainkban fogunk használni.

E dolgozatban bemutatásra kerülnek ennek a rendszernek a tervezési, implementációs illetve verifikációs lépései. Fontos szerepet kapott a lakópark rendszerének integrálhatósága az Arrowhead keretrendszerrel. A dolgozat a lakópark tervezésének, implementációjának és verifikációjának bemutatása mellett annak jövőbeli terveit is taglalja.

## **Abstract**

On the contrary to Gartner's last year 2017 report, in this year's outline the solutions related to smart homes are in a more advanced state and they are trying to achieve more realistic goals. Also, there is a rise of curvature regarding to the solutions for smart workplaces. These are closely related to smart residential areas, as these have to overcome very similar problems. The only difference is that people are working during daylight hours, and go home to rest at nighttime.

Several innovative companies are experimenting in these residential areas to create interconnected smart solutions, to provide sustainable living conditions for the residents. This field of research is yet to be highly developed, as the users are still unsure in what ways they would like to utilize such solutions, so the companies cannot predict the future ways of development impeccably. This is the main focus of my design of smart residential areas: it presents forward thinking applications and fields of development. Needless to say, only the residents of the exact apartment can access its data, restricting external intrusion without consent. Although with the assent of the separate apartments to the proprietor, the shared data can largely improve the living conditions of the community.

My design of the smart residential area builds upon the foundation of the Green-IoT framework, which I presented last year. This framework – when integrated separately to an apartment – is able to make it function as an independent smart home. However, when you connect the framework to the proprietor's system, it is able to signal variable distresses, such as pipe ruptures, fires, or even burglary, prompting a quicker reaction, and thus mitigating collateral damage.

The focus to the utilization of excess resources produced by the smart area was improved significantly, whether that means the auction of the energy residue of solar collectors, or renting unused parking spaces. The Arrowhead framework plays a significant role in such circumstances, which is a framework developed by a large European consortium of innovations. Right now it is under the process of being standardized, and it is expected to find use in our everyday lives.

This paper deals with the planning, implementation, and verification steps of this particular framework. The ability to integrate with the Arrowhead framework is of primary importance.

The paper takes into account the designing, implementation, and verification of the residential area, and also discusses its future prospects.

# 1. Bevezetés

A folyamatos innováció és a felhasználók visszajelzése hatására az okos otthonokkal kapcsolatos megoldások egyre kiforrottabbak és felhasználóbarátok lettek. Ez nagyban hozzájárult ahhoz, hogy azok is betekintést nyerjenek az általuk nyújtott kényelembe, akiknek erre eddig nem volt lehetőségük. A különféle innovatív megoldások lehetővé tették a nagyfokú automatizációt otthonainkban – legyen ez a termosztát állítása, mosás elindítása vagy csak az otthoni hőmérséklet ellenőrzése.

Az okos munkahelyek és okos lakóparkok területén kezdeti próbálkozások vannak, de ez egy nagyon korszerű – nem lezárt – terület, kutatási eredmény sem sok található ebben a problémakörben. A feltörekvő technológiák Gartner görbéjének [1] legelején tart, tehát a kutatások ezen a területen csak most kezdődnek.

Jelenleg a piacon nem sok olyan valódi megoldás található, amely teljes lakópark összes lakását okos otthonokká alakítja és egy központi rendszer által össze is köti, amely az üzemeltetés felé képes hasznos adatokat szolgáltatni.

Magyarországon jelenleg egy ilyen megoldást kínálnak, amelyet az Immonova nyújt [4]. Az általuk alkotott rendszer integrálható új vagy régi építésű társas házba. Megoldásukba nem feltétlenül lehet megfelelő biztonsági szintek mellett a lakásokat bevonni – rendszerük tehát nem köti össze az okos otthont az okos lakótérrel. Megoldásukban megelégszenek a közös terek érzékelőkkel történő felszerelésével az általuk kívánt hatás eléréséhez. A rendszer legfőbb erénye az elektromos hálózat központi vezérlése, monitorozása és naplózása, amely elemzésével képet kaphatunk az egyes lakások fogyasztásáról. Ezzel az eszközzel minimalizálható a fölöslegesen használt energia, tovább csökkentve a fenntartási költségeket. Riasztásokkal képes szolgálni az üzemeltetés felé egy esetlegesen elromlott lift, kiégett izzó vagy betört ablak esetén. E hasznos tulajdonság segítségével könnyebb orvosolni az esetlegesen felmerülő problémákat. A közösségi terek felszerelhetőek mozgás és jelenlét érzékelőkkel, amelyek segítségével csak akkor üzemel a világítás, ha arra szükség van. Ezzel tovább csökkennek a költségek.

Az Immonova rendszere – hasonlóan más, jelenleg elérhető megoldásokhoz – kifejezetten a közös terekre összpontosít, a felhasználó szempontjából nem kínál lehetőséget az okos otthon nyújtotta lehetőségek integrációjára, még adat-megjelenítés szintjén sem. Dolgozatom ebben nyújt újdonságot: egyrészt koncepcionálisan bemutatja az egyéni és

közösségi „okos terek” adatkezelésének módját (az Arrowhead keretrendszer segítségével), másrészt kézzel fogható módon kiterjeszti az okos-otthon megoldást (egy saját fejlesztésű multi-szenzoros hardver bemutatásával).

A jelenlegi rendszerek tehát már képesek külső rendszerek integrációjára, ám az okos otthonnal – vagy az okos munkahellyel – való integrációt nem vetítik előre. A lakóparkok közös tereiről a következő rendszerek szolgálnak információval:

- Kaputelefon;
- Kamerarendszer;
- Beléptető rendszerek (kapuk, elektromos zárok );
- Lépcsőház és kert vagy udvar világítási rendszerek;
- Épület fényfestés (díszfények );
- Mozgásérzékelők;
- Felvonók és azok kiszolgáló automatikái (segélyhívó rendszer, stb.);
- Légtechnikai eszközök (szellőztetők, klímák );
- Öntöző rendszerek;
- Közös területek fűtése és vagy hűtése (utak, parkolók, lépcsőház);
- Gondnok vagy őr szolgálat;
- Fűtés és melegvíz ellátás, egyedi mérés.

További szenzorok is illeszthetőek az Immonova rendszeréhez, amelyek segítségével további értéknövelt szolgáltatások lesznek elérhetők:

- Üvegtörés érzékelők (pincék, lépcsőházak, kapuk, kirakatok, vitrinek, stb.);
- Csőtörés, vízfolyás érzékelők;
- Gázszivárgás érzékelők;
- Emeleti, folyosói ajtók – nyitás, nyitásérzékelés, vezérlés;
- Füstérzékelők – veszélyt jelző sziréna jelzéssel is;
- Fagyás érzékelők;
- Rágcsáló riasztás;
- Solar rendszerek – vezérlése.

A 2000 és környékén született fiatalok (az úgynevezett Z-generáció tagjai) mostanában kezdenek el dolgozni. Ők okostelefon nélkül jellemzően el sem indulnak otthonról, mindennapjaik része az általuk nyújtott kényelmi szolgáltatások intuitív használata. Számunka fontosak a leendő munkahely kiválasztásánál a munkakörülmények, az irodaház jellege, a



„hozzáadott értékek”. Ezért is vannak felszerelve az újonnan épült modern irodaházak rendkívül sok technológiai újdonsággal. A régebbi irodaházak pedig alternatív megoldásokkal próbálják behozni a lemaradásukat. Ezek az újdonságok nem csak az fiatalok, hanem az régebbi generáció számára is nagyban hozzájárul a munka kényelmesebbé és könnyebbé tételében.

Az irodaházak és lakóparkok között nem is akkora a különbség, mint gondolnánk. A munkahelyeken lakások helyett irodák találhatóak, azok okos eszközökkel való felszerelése egyrészt nagyban hozzá tud járulni a munkavállalók komfortérzetének növeléséhez, másrészt az üzemeltetés munkáját könnyíti meg. Lakóparkokhoz hasonló gondokkal kell megküzdeniük az irodaházakban is a mindennapokban. Például elromolhatnak a csapok, mosdók, termosztátok. Esetleg az irodaház erőforrásai nincsenek megfelelően kihasználva. A konditerme kihasználatlanul áll a nap jelentős részében, vagy akár sok üres parkolóhely található a mélygarázsban. A dolgozatom ezt a problémakört vizsgálja és próbál megoldást nyújtani.

## 2. Rendszerek rendszere – az intelligens lakópark

### 2.1 Célkitűzés

A cél egy olyan intelligens lakópark megtervezése, melynek lakásai önállóan is képesek okos otthonokként működni, viszont összekapcsolva fontos információkat tudnak szolgáltatni az üzemeltetés felé, az esetlegesen előforduló problémák gyors és hatékony elhárítása érdekében. A lakópark kihasználatlan erőforrásait kijaánlja, ezáltal azok harmadik feleknek is elérhetőek lesznek, legyen szó kihasználatlan parkolóhelyekről vagy a napkollektorok által termelt áram eladásáról.

Követelmény, hogy az otthonok a lakók életét megkönnyítsék és a lehető legegyszerűbb használatot tegyék lehetővé. Az üzemeltetés felé könnyen áttekinthető képet nyújtson, és a riasztások által gyors beavatkozási lehetőséget biztosítson. Fontos kihangsúlyozni az adatbiztonságot: a rendszer tervezésénél kiemelt figyelmet kapott az, hogy az adatok gyűjtése, hozzáférése és megosztása biztonságos legyen, illetéktelen hozzáférés ne fordulhasson elő. Az adatok biztonságos megosztásában kulcsfontosságú szerepet játszik az Arrowhead keretrendszer.

### 2.2 Az Arrowhead keretrendszer

Az Arrowhead [2] egy európai innovációs konzorcium által fejlesztett, szabványosítás alatt álló keretrendszer, amelyet várhatóan a mindennapjainkban fogunk használni.

Az Arrowhead keretrendszer a szolgáltatás orientált architektúrák alapelvei szerint működik, célja az ipari IoT rendszerek együttműködésének elősegítése. Az ipari IoT különféle területein használható, ideértve a termelés (gyártás, feldolgozás, energia), intelligens épületek és infrastruktúrák, elektro-mobilitás és virtuális energiapiac által lefedett szegmenseket. Az Arrowhead célkitűzése az, hogy lehetővé tegye együttműködő automatizált hálózatok létrehozását beágyazott eszközök használatával. A keretrendszer feltételezi, hogy a szolgáltatás-alapú megközelítés ma már a kiberfizikai rendszerek területén is megvalósítható technológia. Segítségével lehetővé válik a *dinamikusan összekacsolódó, együttműködő* rendszerek automatizációja egy nyitott hálózati környezetben, amely számos beágyazott eszközt összekapcsol.

Az Arrowhead sikere nem csak a technológián múlik, hanem azon is, hogy képes-e újításokat létrehozni és azokat megvalósítani a konkrét ipari szegmensekben. Ha sikeres, akkor – ígérete

szerint – 75%-kal lehet csökkenteni a tervezésre és megvalósításra szánt munkát a több milliárd elkészülendő hálózatba kapcsolt eszközök esetén. [5] [6]

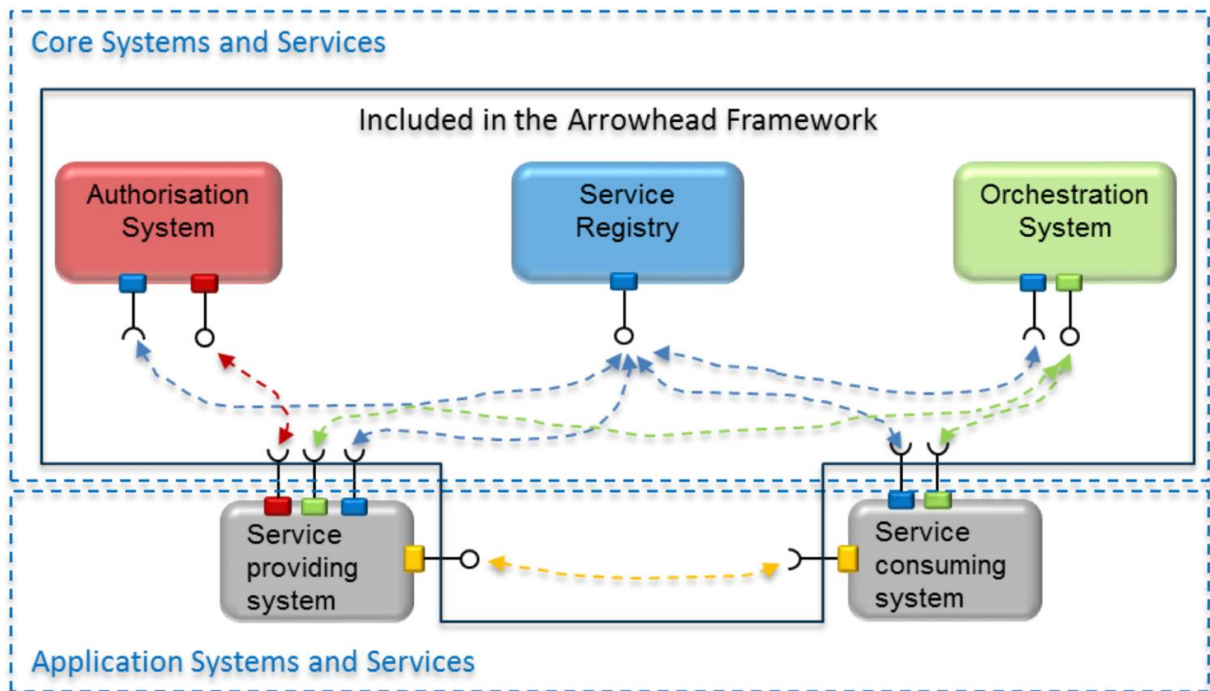
Társadalmunk kihívásokkal néz szembe az erőforrásokért vívott verseny terén. Ezek a kihívások szorosan kapcsolódnak, és új dinamikus interakciókat kívánnak az energiatermelő és energia felhasználó, a szolgáltatás nyújtó és igénylő gép és rendszer, ember és rendszerek között. Az együttműködő rendszerek automatizációja a kulcsa ezeknek a dinamikus kölcsönhatásoknak, ezek teszik lehetővé az Internet of Things megoldások fejlődését – a szolgáltatás alapú architektúrák segítségével.

Az Arrowhead projekt célja, hogy feltárja a technikai és megvalósítási kihívásokat, amelyek az együttműködő rendszerek automatizációjához kapcsolódnak. Célja, hogy

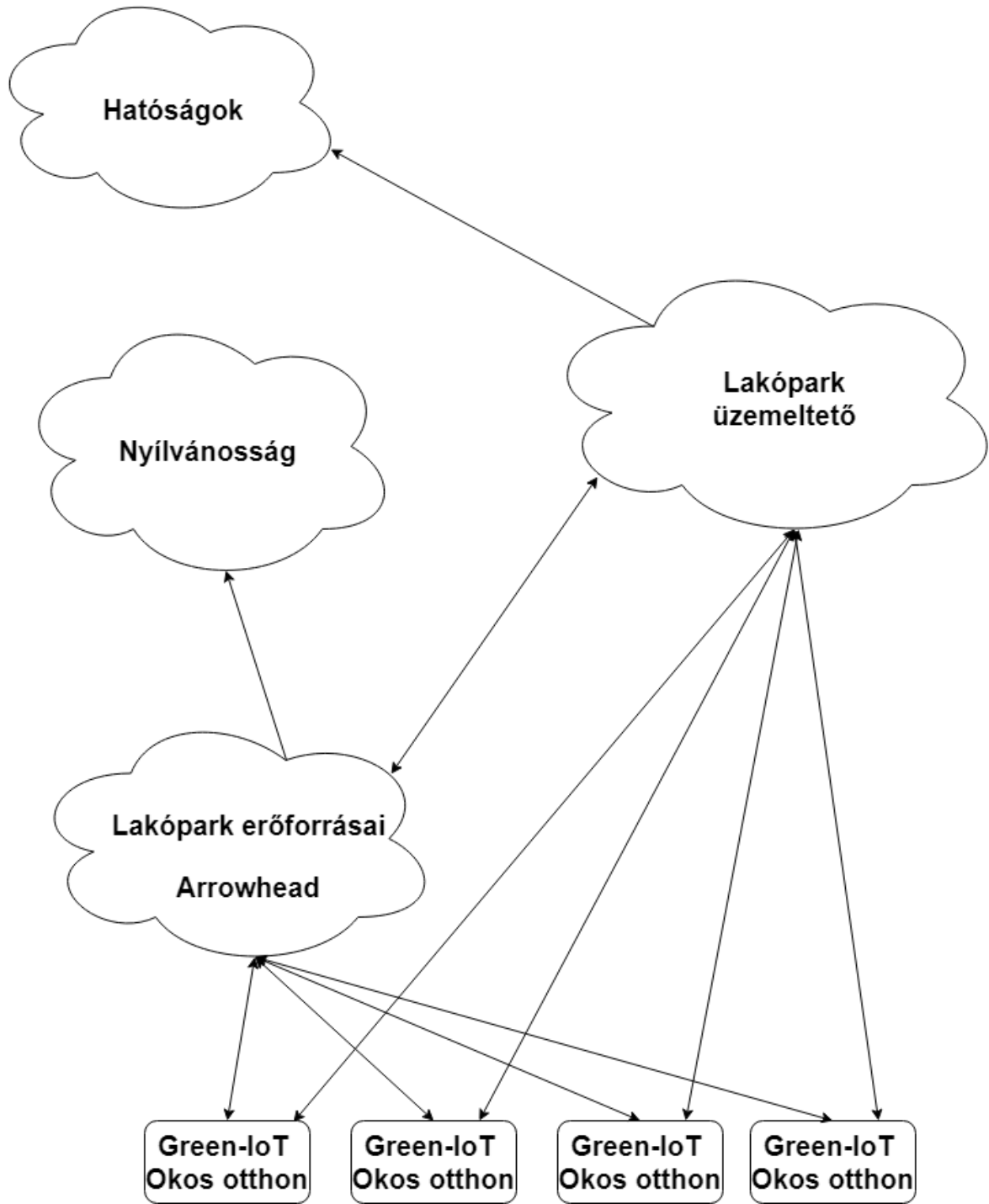
- olyan keretrendszert biztosítson, amely funkciójában és teljesítményében kiemelkedő;
- javasoljon megoldásokat régi rendszerek integrációjára;
- életszerű, valós kísérleteket végezzen és értékeljen az eletro-mobilitás, intelligens épületek, infrastruktúrák, intelligens városok, ipari termelés, energiatermelés és energia-virtuális piac kollaboratív automatizálásának területén.
- új szolgáltatásoknak köszönhetően emelje ki az elérhető újításokat
- vezesse az utat a szabványosítási folyamathoz.

Az 1. ábrán látható módon, az Arrowhead három fő központi rendszerrel rendelkezik és két alkalmazásbeli rendszerrel.

- Authorization System: autentikációs és autorizációs feladatokat lát el. Emellett tanúsítványkezelést is végez.
- Service Registry: Tárolja az összes rendszert és az általuk nyújtott szolgáltatásokat.
- Orchestration System: Felelős a rendszerek orchesztrációjáért. Segítségével tudják a rendszerek, hogy hová csatlakozzanak, melyik szolgáltatást használják.
- Service Providing System: Egy vagy több szolgáltatást nyújt, amelyet más rendszerek igénybe vehetnek.
- Service Consuming System: Egy másik rendszer által nyújtott szolgáltatást képes igénybe venni.



1. ábra - Arrowheadhez kapcsolódó szolgáltatások

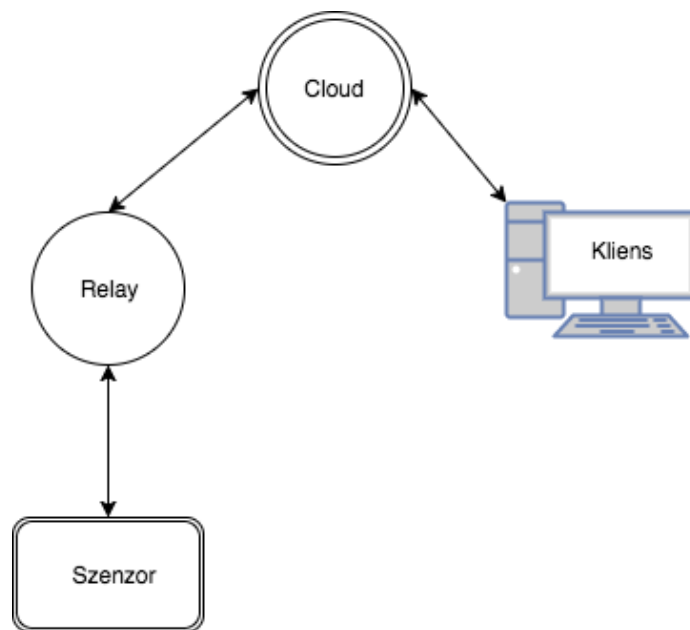


2. ábra Intelligens lakópark architektúrája

Az általam tervezett intelligens lakópark öt fő elemből épül fel, amelyek a 2. ábrán látható módon kapcsolódnak egymáshoz:

- Green IoT alapú Okos Otthonok;
- Lakópark erőforrásai;
- Lakópark üzemeltető;
- Nyilvánosság felé megosztott adatok;
- Hatóságok.

### 2.3 Green-IoT alapú okos otthonok



3. ábra Green-IoT keretrendszer modellje

Az elmúlt évi TDK dolgozatomban [7] részletesen kifejtettem, hogyan működik az általam tervezett és megvalósított Green-IoT keretrendszer –itt a későbbi érthetőség kedvéért röviden összefoglalom a felépítését. Az egyes otthonokban működő rendszer 3 fő összetevőből épül fel, amely az 3. ábrán tekinthető meg.

- Cloud: felhő alapú adatfeldolgozó rendszer, management felület;
- Relay: összeköttetést biztosít a szenzorok és a Cloud között;
- Szenzorok: ESP8266 [9] mikrokontroller alapú eszközök.

A Cloud egy Központi feldolgozó egység, ahová befutnak a mérési adatok és az feldolgozza őket. Ez egy felhő alapú szolgáltatás, amely webes menedzsment felületet biztosít a lakók számára, saját szenzorjaik adatainak hozzáféréséhez.

A Relay csomópont egy helyi feldolgozó egység, ami a Clouddal és a szenzorokkal fog kommunikálni és összeköti őket. Több is elhelyezhető lakásonként, így a nagyobb, több szintes lakásokban sem lesz gond a lefedettséggel. A Relay a szenzorokkal vezeték nélküli technológiákkal képes kommunikálni (WiFi, Bluetooth), összegyűjti és elő-feldolgozza az adataikat, majd továbbítja a Cloud felé. A szenzorok felé vezérlő-információt is továbbíthat. A Raspberry Pi [8] alapú Relayekhez egyedi hardvert készítettem, amely őket is mérőállomássá tudja alakítani.

A szenzorok jelen architektúrában ESP8266 mikrovezérlő alapú eszközök, amelyekhez különféle szenzorok vannak kapcsolva, ezek továbbítják az általuk mért adatokat a csomópontok felé

Az otthonok számos szenzorral rendelkeznek, ezek a szenzorok által mért adatok hozzáférhetőek a lakók számára a webes felületen, illetve ezek az adatok megoszthatók az üzemeltetés vagy akár a nyilvánosság felé.

Az 1. táblázat mutatja be, milyen szenzorokat lenne célszerű az otthonokba telepíteni.

1. táblázat – Okos otthon szenzorok és hozzáférésük az okos lakóparkban

Szenzorok	Kötelező az üzemeltetés felé megosztani	Általam megvalósított
Hőmérséklet	✓	✓
CO	✓	✓
CO <sub>2</sub>	✓	✓
Föld nedvesség	✗	✓
Ajtó, ablak állapot	✓	✗
Csap szivárgás	✓	✗
Csőtörés	✓	✗
Beázás	✓	✗
Jelenlét és mozgás érzékelő	✗	✗

## 2.4 Lakóparkok közös erőforrásai

Minden lakóparknál problémát jelent, hogy az erőforrások nincsenek megfelelően kihasználva, nagyon sok erőforrás kárba vész, ezáltal növelve az fenntartási költséget, amelyet a lakókra ró. Lakóparkonként számos közös erőforrás található, legyen szó lakásokról, parkolóhelyekről, napkollektorok által termelt áram hasznosításáról, vagy akár közösségi sportlétesítményekről.

Üres lakások esetén célszerű lehet azokat kiadni albérletbe, vagy akár az Airbnb [11] szolgáltatáshoz hasomlóan rövidebb időre is.

Napközben a lakók jelentős része dolgozni jár, ezért az üres parkoló helyeket fizetős parkolási szolgáltatásként lehet abban az időtartamban hasznosítani. Az az eset is előfordulhat, hogy a lakópark több szabad parkolóhellyel rendelkezik, mint amire a lakóknak szüksége lehet, ilyenkor a hosszabb távú bérbeadás is szóba jöhet.

Intelligens lakóparkként elvárás, hogy környezettudatos és valamilyen szinten önellátó legyen, ezért például napkollektorokkal legyen felszerelve. Ezáltal jelentős mennyiségű elektromos áram termelésére ad lehetőséget. Viszont az áram tárolása nehézkes és költséges dolog [3]. Egy magyar háztartás átlagosan havi 175kWh áramot fogyaszt [10]. Egy kilogramm tömegű lítium-ion akkumulátor csak 0.2kWh-nyi áramot képes tárolni. Ez azt jelenti, hogy egy 875 kilogrammos akkumulátora lenne szükség, a havi áram mennyiség tárolásához. Ezért sokkal célszerűbb dolog lehet a fölöslegesen termelt áramot értékesíteni, ezzel is tovább csökkentve a lakópark költségeit.

Számos modern lakópark rendelkezik saját uszodával, sportpályával, szaunával. Ezek kihasználtságát is lehet növelni, ha külső felek is használhatják.

Ezért készítettem egy mobil alkalmazást, amelyen keresztül a lakók bármikor megtekinthetik az otthonuk jelenlegi állapotát, továbbá foglalási lehetőséget biztosítva a lakópark erőforrásaira. Az alkalmazás egy „proof of concept” implementáció, fizetési megoldás egyelőre nem került implementálásra.

## 2.5 Lakópark üzemeltető

A lakópark üzemeltetői számára az egyik legfontosabb dolog, hogy mindig aktuális képet lássanak a lakópark állapotáról és minél hamarabb értesüljenek az esetleges problémákról, hibákról, hogy azokat el tudják hárítani. Abban, hogy az üzemeltetés a különféle



rendszerektől adatot kapjon, az Arrowhead keretrendszer tud segíteni, biztosítva a hatékony adat gyűjtést és feldolgozást.

Az üzemeltetés feladatközébe tartozik:

- Lakások és közös terek állapotának megfigyelése;
- Biztonsági szolgálat biztosítása;
- Karbantartás;
- Hibaelhárítás.

A legtöbb jelenlegi régi panelházban bevett szokás, hogy a lámpák egész éjjel égnek a lépcsőházban, főlegesen használva az áramot és csökkentve a lámpák élettartamát, ezáltal a költségeket is növelve. Sokkal egyszerűbb és elegánsabb megoldás a közös terek mozgás- és jelenlét érzékelő szenzorokkal való ellátása, így csak akkor kapcsolnak fel a lámpák, amikor valaki arra jár.

Térfigyelő kamerák és beléptetőrendszer segítségével könnyedén nyomon követhető az egyes lakók napi rutinja. Ezáltal további biztonsági réteget lehet kialakítani. Rosszindulatú esetben – tegyük fel, hogy elrabolják egy lakó iratait és lakáskulcsát, az iratokból kiderül a lakcím, így napközben – amikor dolgozni megy el a lakó, könnyedén betörhetnek a lakásába. Ilyen esetekre is hasznos a napi rutin felismerése. Így ha attól eltérő viselkedés, rendellenesség tapasztalható, akkor riasztja a felhasználót a telefonos alkalmazáson keresztül. Ha nem hagyja jóvá a belépést, akkor riasztja a biztonsági szolgálatot, amely ellenőrzi, hogy minden rendben van-e.

## **2.6 Nyilvánosság felé megosztott adatok**

A nyilvánosság felé megoszthatók bizonyos szolgáltatások, vagy erőforrások elérhetősége, amennyiben a lakópark úgy dönt. Ezek lehetnek a szabadon használható parkolóhelyek foglaltsági adatai (amelyekért a külsősök alkalmanként fizetnek), vagy a wellness, vagy szabadidős szolgáltatások (pl. szauna, tenispálya, stb.) kihasználatlan – de foglalható – időszakai.

## **2.7 Hatóságok**

Az üzemeltetés össze van kötve a hatóságokkal. Legyen szó tüzesetről, betörésről, egészséget érintő tényezőkről, a lakópark szenzorai segítségével könnyedén értesíthetőek a megfelelő hatóságok, akár automatikusan is.

## 2.8 Intelligens lakópark lokális felhői és szenzorokkal való kapcsolattartás

A lakóparkban számos lokális felhő található. A korábban bemutatott Green-IoT keretrendszer által a lakások saját lokális felhőkkel rendelkeznek, ahová a szenzorok által mért adatok befutnak. A lakó engedélyezheti, hogy mely mérési eredmények kerüljenek megosztásra az üzemeltetés felé. Ezek a mérési eredmények az Arrowhead keretrendszeren keresztül kerülnek beküldésre, az üzemeltetés által karban tartott lokális Arrowhead felhőbe. (Ebben a példában ez a TMIT tanszéken üzemeltetett Arrowhead felhő.) A közös tereken elhelyezett mozgásérzékelő, fényerősség, és egyéb szenzorok által végzett mérési eredmények is az Arrowhead felhőbe lesznek beküldve, ezáltal a beavatkozók értesülnek ezekről és fel tudják kapcsolni a lámpát a folyosón. Ezáltal az üzemeltetés is tud értesülni az egyes lakások és a közös terek állapotáról. A szenzorok és beavatkozók Wi-Fi-n keresztül kommunikálnak.

## 3. Tervezés

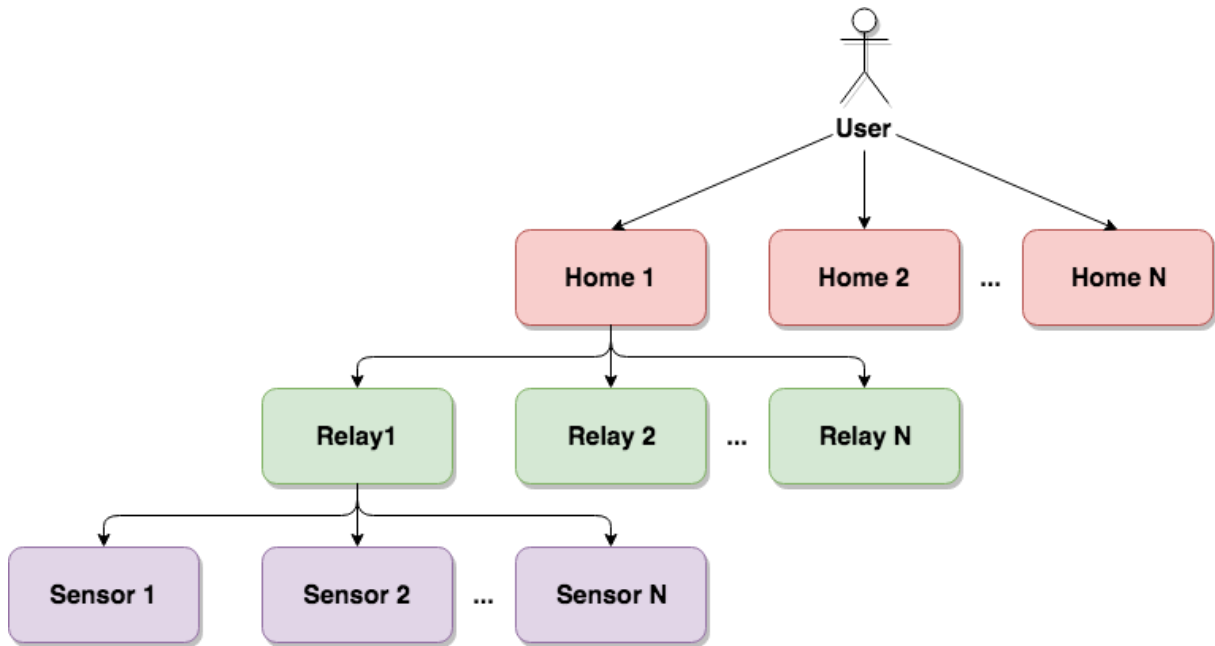
### 3.1 Green-IoT keretrendszer továbbfejlesztése

Az okos otthon keretrendszerem továbbfejlesztését négy lépésben valósítottam meg.

- Backend továbbfejlesztése;
- Mobil alkalmazás fejlesztése;
- Arrowhead keretrendszerhez készült felhasználói felület fejlesztése;
- Nyomtatott áramkör tervezése;

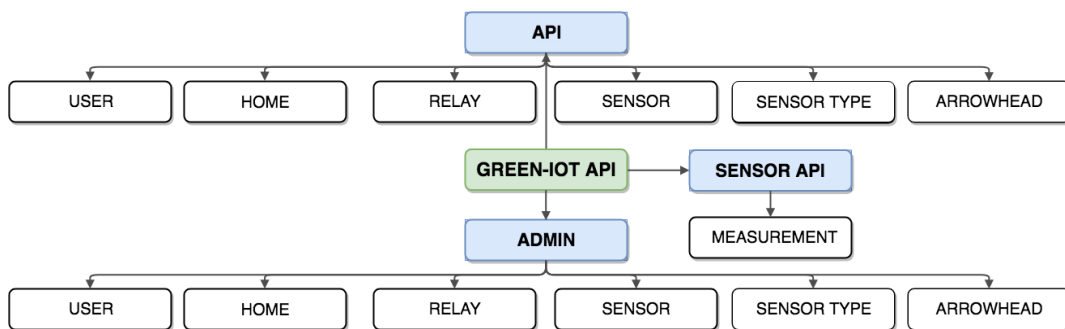
A backend esetén úgy döntöttem, hogy amióta megírtam az első verzióját, azóta sokat fejlődött a tudásom NodeJS-ben; [15] bizonyos régebbi döntéseimet nem találtam optimálisnak. Ezért a teljes újraírás mellett döntöttem. A funkciók nem voltak megfelelően elválasztva, így külső rendszereket nem lehetett hozzá csatolni. Egy új alkalmazás-programozási interfészt (API) terveztem, amelyhez könnyedén lehet csatolni külső rendszereket, így az Arrowhead-et is.

Adatbázis szinten korlátokba ütköztem a MongoDB [17] használata során, ezért úgy döntöttem, egyszerűbb lesz MySQL [18] relációs adatbázist használni.



4. ábra Több otthon hozzáadását lehetővé tevő struktúra

Az előző TDK dolgozattal kapcsolatban kérdést kaptam arra vonatkozóan, hogy egy lakásba több Relay-t is el lehet-e helyezni – ez akkor is lehetséges volt már, de valóban nem volt egyértelmű az architektúra. Ezért most a felhasználók és a Relay-ek közé bekerül egy új réteg, az *otthon*. A felhasználó otthonába tud felvenni Relay-eket, és azokhoz tartoznak a szenzorok, ezáltal egy közös menedzsment felületet is biztosítva a felhasználó számára, amely a 4. ábrán látható.



5. ábra Green-IoT API struktúra

Az 5. ábrán látható a keretrendszer API-ja. Külön API van az Adminisztrátori funkciók elérésére, a hagyományos felhasználók számára illetve a szenzorok adatainak feldolgozására. Az API-k különféle szolgáltatásokat nyújtanak, az Adminisztrátor API és az felhasználókat kiszolgáló API jogosultsági szinten különbözik. A felhasználók csak a saját otthonaikat és eszközeiket látják, míg egy adminisztrátor a felhasználó összes eszközét megtekintheti.

A felhasználói API az alábbi szolgáltatásokat nyújtja:

- User: A felhasználó autentikációjáért és adatainak változtatásáért felelős;
- Home: A felhasználó számára otthonainak kezelését teszi lehetővé;
- Relay: A felhasználó otthonaiban található Relay csomópontok kezelését teszi lehetővé;
- Sensor: A felhasználó szenzorainak kezelését teszi lehetővé;
- Sensor Type: A felhasználó számára lehetővé teszi, hogy új típusú szenzorokat vegyen fel;
- Arrowhead: A felhasználó megtekintheti a mérési adatait az Arrowhead keretrendszerben.

A szenzor API az alábbi szolgáltatást nyújtja:

- Measurement: A szenzorok által mért értékek rögzítéséért felelős.

Az Adminisztrátori API az alábbi szolgáltatásokat nyújtja:

- User: A felhasználók adatait lehet megtekinteni és módosítani;
- Home: A felhasználók otthonait lehet megtekinteni és módosítani;
- Relay: A felhasználók Relay csomópontjait lehet megtekinteni és módosítani;
- Sensor: A felhasználók szenzorait lehet megtekinteni és módosítani;
- Sensor Type: A felhasználók által létrehozott szenzor típusokat lehet megtekinteni és módosítani.
- Arrowhead: Mérési adatok megtekintésére alkalmas az Arrowheaden keresztül.

Az elmúlt évi megoldáshoz képest fontos módosítás, hogy külön rendering enginet nem tartalmaz a projekt. Ennek hatására csak JSON válaszokat küld a szerver, szemben a korábbi megközelítéssel, a szerver oldalon generált HTML kóddal. Ennek az az óriási előnye, hogy az API-hoz tetszőleges alkalmazás illeszthető. Ezáltal a klienseket és a backendet egymástól függetlenül, más-más technológiával is lehet fejleszteni. Az API-hoz készült mobil illetve webes alkalmazások, később lesznek bemutatva.

## 3.2 Mobil alkalmazás

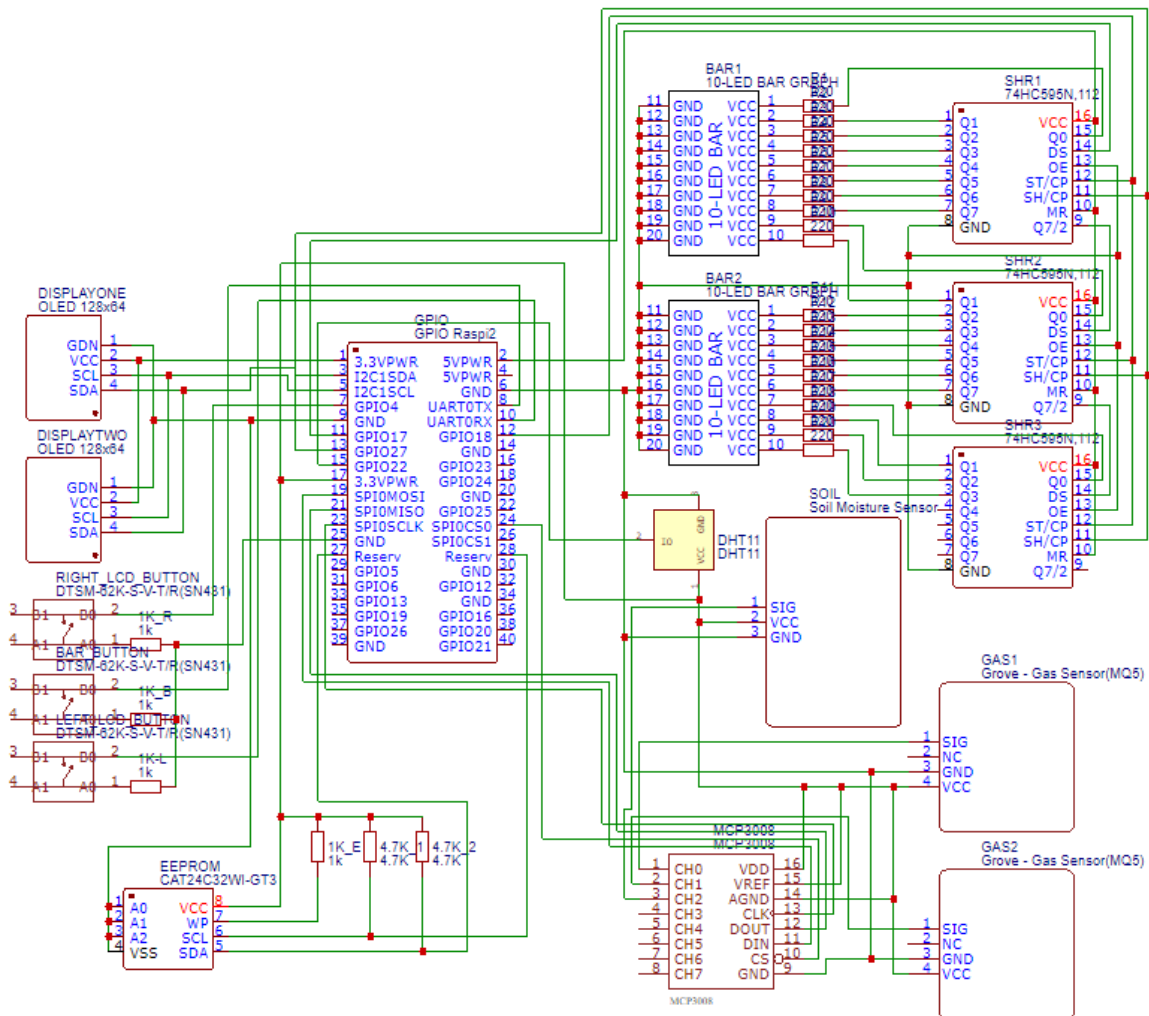
Szinte egy lépést sem teszünk már meg okos telefonjaink nélkül: ügyeket intézünk, banki tranzakciókat kezdeményezünk különféle alkalmazásokon keresztül. Ezért nagyban megkönnyítené a felhasználók életét, ha egy karnyújtásnyira bármikor elérhetnék a lakópark szolgáltatásait, akár menet közben is.

### 3.3 Arrowhead grafikus felhasználói felület

Az Arrowhead keretrendszer egy nagy és összetett keretrendszer, amelynek átlátása és kezelése nem triviális. Ezért készítettem bizonyos kulcs-elemeihez grafikus felhasználói felületet. A megvalósításhoz a React [16] JavaScript könyvtárt használtam, amely nagyban megkönnyíti webes frontend alkalmazások készítését. A felület a 4.3-as fejezetben lesz részletesebben bemutatva.

### 3.4 A csomóponti szenzorillesztő nyomtatott áramkör

Múlt évi dolgozatomban továbbfejlesztési irányként jelöltem meg egy nyomtatott áramkört, melynek segítségével maga a csomópont is szenzorokkal ruházható fel könnyedén. Megterveztem, milyen funkcionalitással lenne praktikus felruházni, amely felhasználóval történő interakciókat is támogat. Úgy döntöttem, hogy szenzorok mellett kijelzők és gombok hasznos részei lesznek az elkészülendő áramkörnek.



6. ábra Az áramkör kapcsolási rajza

Az áramkör tervezése során figyelembe vettem a Raspberry Pi Foundation által specifikált méretbeli és hardveres követelményeket. Az ilyen áramköröket „Hardware Attached on Top”-nak hívják, röviden HAT-nek [12]. Ezek olyan célhardverek, amelyek további funkcionalitással ruházzák fel az eszközt. Az általam készített kapcsolási rajz a 6. ábrán tekinthető meg.

Az áramkör méretbeli követelményeit nem tudtam betartani, mivel sok komponensből áll az áramköröm. A specifikáció része egy EEPROM memória modul is. A Raspberry Pi indulásakor az EEPROM-ból kiolvassa az áramkör nevét és az általános célú be és kimeneti túske (GPIO) állapotát. Ez alapján állítja be őket be- és kimenetre, illetve logikai magas és alacsony állapotba. Ez egy nagyon fontos lépés, mivel ennek hiányában a GPIO-k határozatlan állapotban lennének, amely akár el is ronthatná az áramkört.

Az áramkörön elhelyeztem két levegő-minőség mérő szenzort. A két szenzor által mért értékek megjelenítésére elhelyezem két darab, egyenként 10 LED-ből álló oszlop diagramot. A szenzor által mért értéktől függően egyre több LED gyullad ki letről felfelé. A LED-oszlopról egy szemléltetés a 7. ábrán látható.



7. ábra 10 LED-ből álló oszlop

Egy páratartalom-mérő, egy hőmérő, illetve egy talajnedvesség szenzor is helyet kapott az áramkörön. A szenzorok által mért értékek a 2 darab OLED LCD kijelzőn numerikusan is leolvashatók. Helyet kapott 3 nyomógomb is, amellyel mérést lehet indítani és aztán a kijelzőről leolvasni az értéket. A LED-oszlopok vezérléshez szükséges 3 darab Shift Regiszter, amelyek segítségével könnyedén lehet az értéküket beállítani. Mivel a Raspberry Pi nem rendelkezik analóg bemenettel, ezért egy Analóg-Digitális konverter jelenléte is szükséges. Ez az alkatrész állít elő a szenzorok által mért analóg jelből a Raspberry Pi által feldolgozható digitális jelet.

### **3.5 Költségbecslés**

Az intelligens lakópark luxus befektetést sugall, ám igazából az eszköz és szoftver igénye igencsak visszafogott

Egy adott lakásban működő megoldás megvalósításához az alábbi eszközök merültek fel – aktuális bekerülési költséggel:

2. táblázat – Nyomtatott áramkör költsége

<b>Egy kész nyomtatott áramkör költsége</b>	
<b>Nyomtatott áramkör</b>	100Ft
<b>40 PIN header</b>	50Ft
<b>25db Ellenállás</b>	50Ft
<b>IC foglalatok</b>	50Ft
<b>MCP3008 ADC</b>	1500Ft
<b>3db 74HC595 SHR</b>	50Ft
<b>2db OLED LCD</b>	3000Ft
<b>3db gomb</b>	10Ft
<b>Nem korrodáló földmérő szenzor</b>	3000Ft
<b>2db Levegőmérő szenzor</b>	660Ft
<b>2db 10LED oszlop</b>	660Ft
<b>DHT11 Hőmérő</b>	330Ft
<b>EEPROM</b>	50Ft
<b>Összesen</b>	9510Ft

3. táblázat - Relay csomópont költsége

<b>Relay Csomópont alkatrészei</b>	<b>Ár</b>
<b>Raspberry Pi 3B</b>	12.000 Ft
<b>SD kártya 16GB</b>	3.000 Ft
<b>Nyomtatott áramkör</b>	9510Ft
<b>Hálózati adapter</b>	2500Ft
<b>Összesen</b>	27010Ft



4. táblázat - Mikrovezérlős szenzorok költsége

Mikrovezérlős szenzorok	Ár
<b>EP8266</b>	1000Ft
<b>Egy szenzor</b>	330Ft
<b>Hálózati adapter</b>	1500Ft
<b>Összesen</b>	2500+(1..5)*330Ft

Egy lakásba egy, maximum két Relay csomópont elég és néhány, maximum 5 db EP8266-os szenzor. Tehát lakásonként kb 63.500 Forintba kerül az eszköz –, valamint a havi költség, ami az eszközök áramfelvételéhez tartozik. Ezen eszközök költségeit a **2.**, **3.** illetve **4.** táblázatban mutattam be.

### 3.6 Közösségi terek, biztonsági kamerák

Közösségi tereken, folyosókon elhelyezett mozgásérzékelős világítás költsége körülbelül 5000Ft eszközönként. Lakópark kamerás megfigyelése történhet Intellio [14] kamerákkal. Nagyon fejlett arcfelismeréssel és mozgás-követéssel rendelkeznek.

### 3.7 Szerverek költsége

Green IoT keretrendszer havi költsége 1500 Ft. Az üzemeltetés számára célszerű egy szervert vásárolni és lokálisan tárolni az adatokat. Ennek a szervernek az egyszeri költsége nagyságrendileg 300.000Ft és havi áram-költség, amit a szerver fogyaszt.

### 3.8 Szoftverek költsége

A hardvereken futó szoftvereknek is jelentős költsége van. A szoftvert el kell készíteni, karban kell tartani, javítani és tovább fejleszteni. Ehhez fejlesztőkre van szükség, akik elvégzik a munkát; ezt a költséget itt nem próbálom megbecsülni – de az bizonyos, hogy a fejlesztés költségét sokszáz, vagy sok ezer felhasználó között kell „megosztani” a megfelelő árazással.

## 4. Implementáció és kivitelezés

### 4.1 Green-IoT API implementációja

Három fő részre lett szétválasztva az API, amely három külön interfészt is jelent, tehát egymástól függetlenül lehet őket cserélni, tovább fejleszteni, a rendszer így is teljesen működőképes lesz.

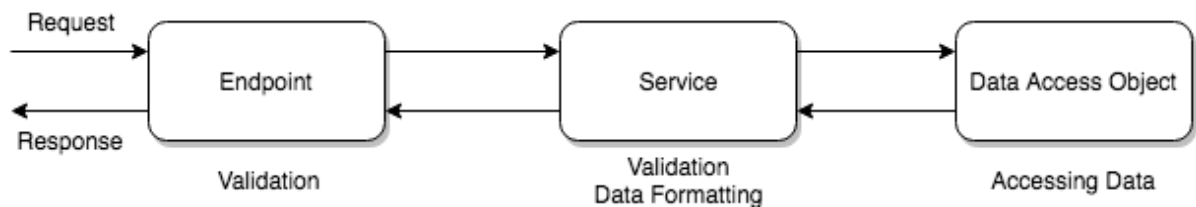
- Alkalmazás API: ehhez csatlakoznak a mobilos, webes alkalmazások. Felhasználó szintű jogosultságokkal rendelkeznek.
- Admin API: ennek segítségével érik ez az adminisztrátorok a rendszert. Teljes körű hozzáféréssel rendelkeznek a rendszeren belül.
- Szenzor API: Új csomópontok és szenzorok és azok által mért eredmények beküldésére alkalmas.

5 táblázat - Green-IoT API jogosultsági mátrix

	Alkalmazás API	Admin API	Szenzor API
Otthon hozzáadás	✓	✓	✗
Otthon módosítás	✓	✓	✗
Otthon törlés	✓	✓	✗
Relay hozzáadás	✓	✓	✓
Relay törlés	✓	✓	✗
Relay módosítás	✓	✓	✗
Szenzor hozzáadás	✓	✓	✗
Szenzor módosítás	✓	✓	✗
Szenzor törlés	✓	✓	✗
Szenzor Típus hozzáadása	✓	✓	✗
Szenzor Típus módosítása	✓	✓	✗

Szenzor Típus törlése	✓	✓	✗
Arrowhead rendszer hozzáadása	✓	✓	✗
Arrowhead rendszer módosítása	✓	✓	✗
Arrowhead rendszer törlése	✗	✓	✗
Arrowhead szolgáltatás felvétele	✓	✓	✗
Arrowhead szolgáltatás módosítása	✓	✓	✗
Arrowhead szolgáltatás törlése	✗	✓	✗
Arrowhead mérés hozzáadása	✗	✗	✓
Mérés hozzáadás	✗	✗	✓
Mérés módosítás	✗	✓	✗
Mérés törlés	✗	✓	✗

Az 5. táblázatban részletesen bemutatom, mihez férnek hozzá a felhasználói, adminisztrátor és szenzor API-k.



8. ábra Endpoint - Service DAO szeparáció

A 8. ábrán a kérések feldolgozásának folyamatát látjuk. A kérések az Endpointokhoz (végpontok) érkeznek be, itt ellenőrizzük a hozzáféréshez való jogosultságot, paramétereket, aztán meghívódik a Service, ahol újra validáljuk a paramétereket, műveleteket hajtunk rajta végre. Aztán a Data Access Objecthez fordulunk, amely kikéri az adatokat az Adatbázisból.

Visszatérve a Service-be aztán transzformáljuk az adatokat megfelelő formátumra, majd visszaadjuk őket.

A Green-IoT keretrendszerben Green-IoT Service-ként kerültek implementációra az Arrowhead keretrendszer központi elemei is. Kézzel beregisztráltam a Green-IoT keretrendszert mint szolgáltatást az Arrowhead Service Registry-be, mivel mindig elérhető a webszerver és adatokat tud visszaadni. Ha úgy döntünk, hogy megosztjuk a szenzorunkat, akkor autorizálja [14] az Arrowhead felé és egy szolgáltatás-igénylőn (service consumer-en) át le tudjuk kérni az adatokat.

## **4.2 Docker**

A Docker [19] egy virtualizációs platform, amelyben az alkalmazások külön kontéerekben futnak, ezáltal milliszekundumok alatt képesek elindulni és skálázódni terhelés alatt.

A keretrendszer back-endjét dockerizáltam, egy Dockerfile segítségével, amiből aztán docker image-et hoztam létre. Ennek a képfájlnak a felhasználásával hatékonyan lehet új konténereket létrehozni szükség esetén.

## **4.3 Arrowhead front-end fejlesztése**

Elkezdődött az Arrowhead keretrendszer grafikus felhasználói felületének a fejlesztése is; ennek a döntő része a saját implementációm. Ennek a felületnek a célja keretrendszer központi elemeinek menedzselése és monitorozása, felhasználóbarát formában. Jelenlegi állapotában a felület segítségével lehet a Service Registry-be regisztrált rendszereket listázni, az Orchesztrációs státuszt megtekinteni, valamint a Gatekeeper és Relay állapotát megtekinteni.

Service Registry

**AUTHORIZATION**

ID: 51  
 Address: 152.66.246.237  
 Port: 8344  
 Authentication Info:

ID	Service Definition	Interfaces	PORT	Service URI	UDP
	InsecureAuthorizationControl	JSON			false
	InsecureTokenGeneration	JSON			false

Rows per page: 5 1-2 of 2

**GATEWAY**

**EVENTHANDLER**

9. ábra Service Registry

A 9. ábrán látható módon lehet megtekinteni a Service Registrybe regisztrált szolgáltatások listáját és tulajdonságait. Egy listát kapunk, amelynek elemére kattintva az lenyílik, és megtekinthetjük a tulajdonságait, táblázatos formában.

Orchestration Status

Filter by service

Provider  Consumer

List without Core Systems

Order type

**System 1**  
Provides Services on 2018 Sep 21., 18:09

ServiceId	ServiceDef	SysId	SysName	Local Cloud
1	Billing	20	SmartGridSystem2	Dummy7
1	Billing2	2	SmartGridSystem28	Dummy3
2	IndoorHumidity5	10	ChargingReserv	Dummy6
2	IndoorHumidity2	100	ChargingReserv7	Dummy2
2	IndoorHumidity7	15	ChargingReserv	Dummy5

**System 2**  
Provides Services on 2018 Sep 21., 18:09

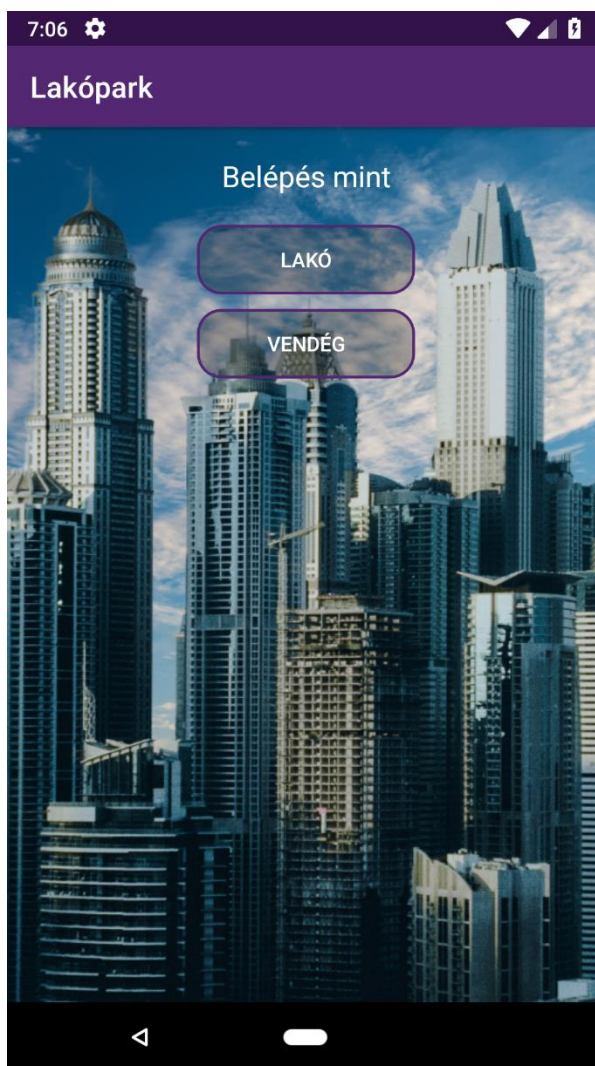
ServiceId	ServiceDef	SysId	SysName	Local Cloud
1	Dummy	20	SmartGridSystem2	Dummy7
1	Dummy5	2	SmartGridSystem28	Dummy3
2	Dummy2	10	ChargingReserv	Dummy6
2	Dummy6	100	ChargingReserv7	Dummy2
2	Dummy2	15	ChargingReserv	Dummy5

10. ábra Orchestration Status

A 10. ábrán látható az orkesztráció jelenlegi státusza, még csak teszt jellegű adatokkal feltöltve. Itt meg tudjuk tekinteni az egyes rendszerekbe beregisztrált szolgáltatásokat és azok főbb jellemzőit táblázatos formában.

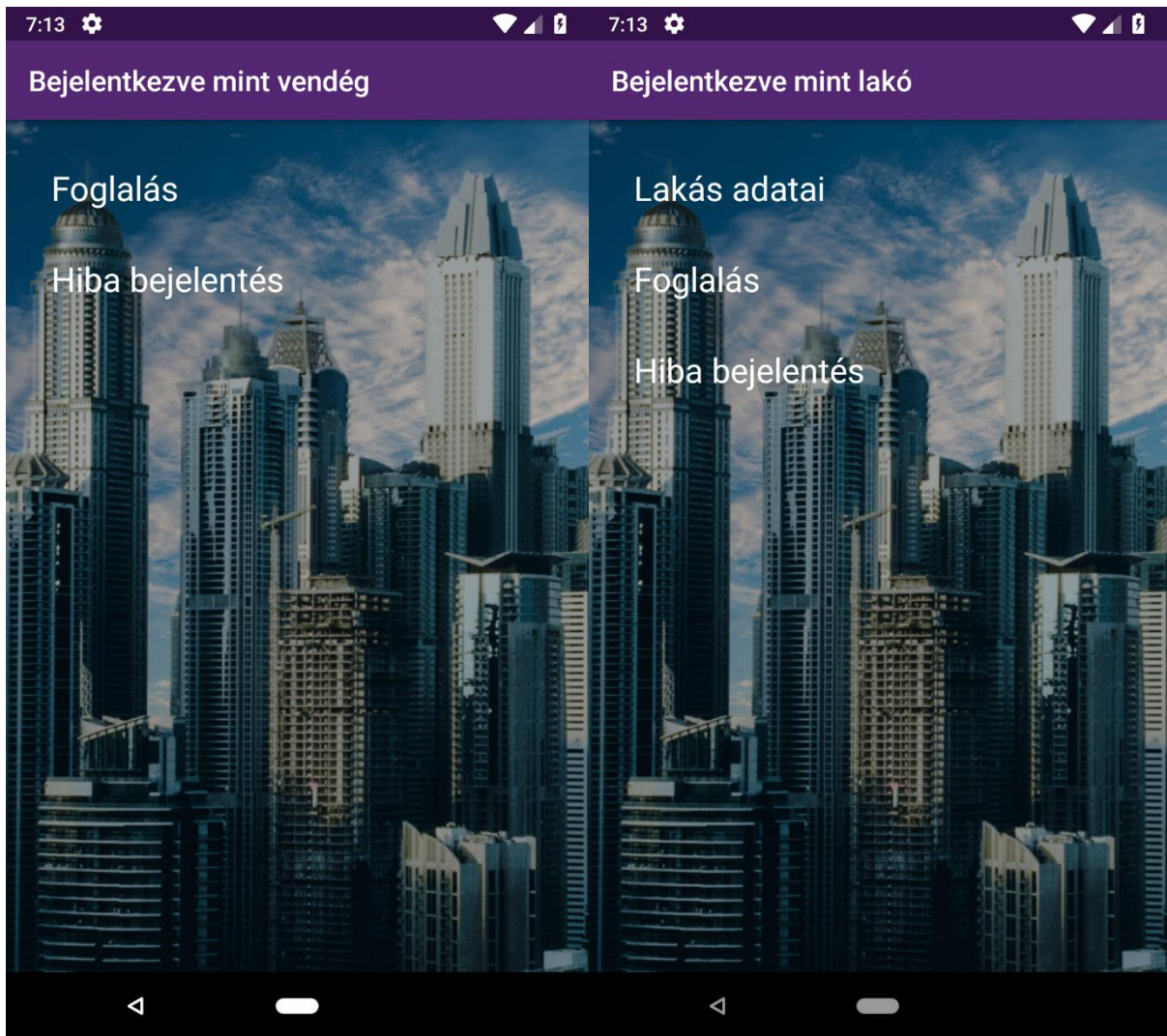
## 4.4 Mobil alkalmazás

A felhasználói oldal szemléltetése érdekében elkészült egy mobil applikáció, amely bemutatja a rendszer működését. A mobil alkalmazás jelenleg csak Android platformon érhető el.



11. ábra Alkalmazás kezdő képernyője

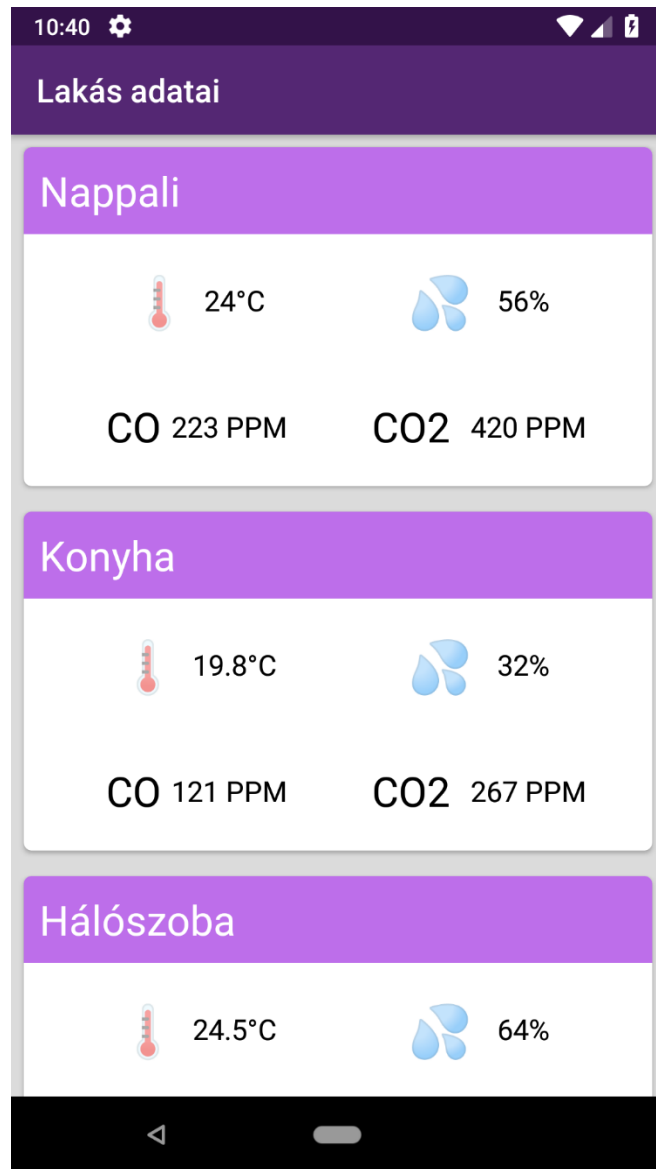
A 11. ábra az alkalmazás kezdő képernyőjét jeleníti meg. Itt a felhasználónak be kell jelentkeznie: mint lakó, vagy vendég. Ez egy fontos lépés a jogosultságkezelés miatt, mivel különböző felhasználói csoportokba tartoznak, más dolgokat tudnak elérni az alkalmazáson belül.



12. ábra Vendég és lakók által látott menü

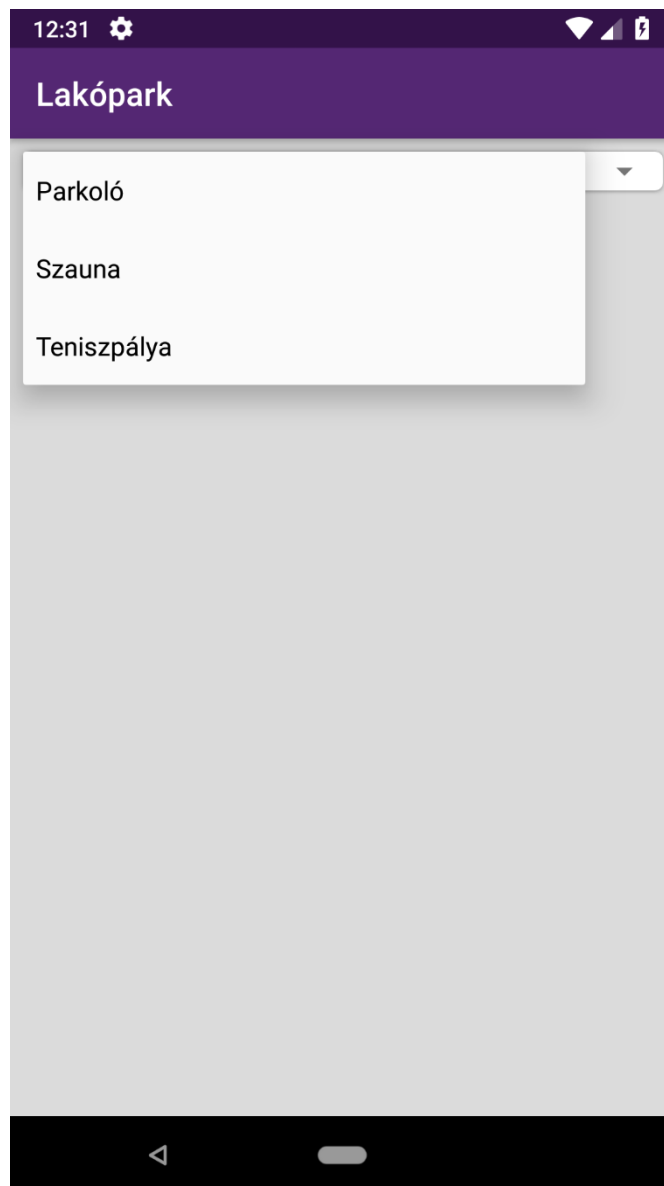
A 12. ábrán látható, hogy az alkalmazás felhasználói felülete megváltozik attól függően, hogy a felhasználó éppen milyen jogosultság körbe tartozik. Jelen példán az látható, hogy a vendég felhasználó nem tudja lekérdezni a lakása adatait, mivel nem rendelkezik olyannal.





13. ábra Felhasználó lakásának adatai

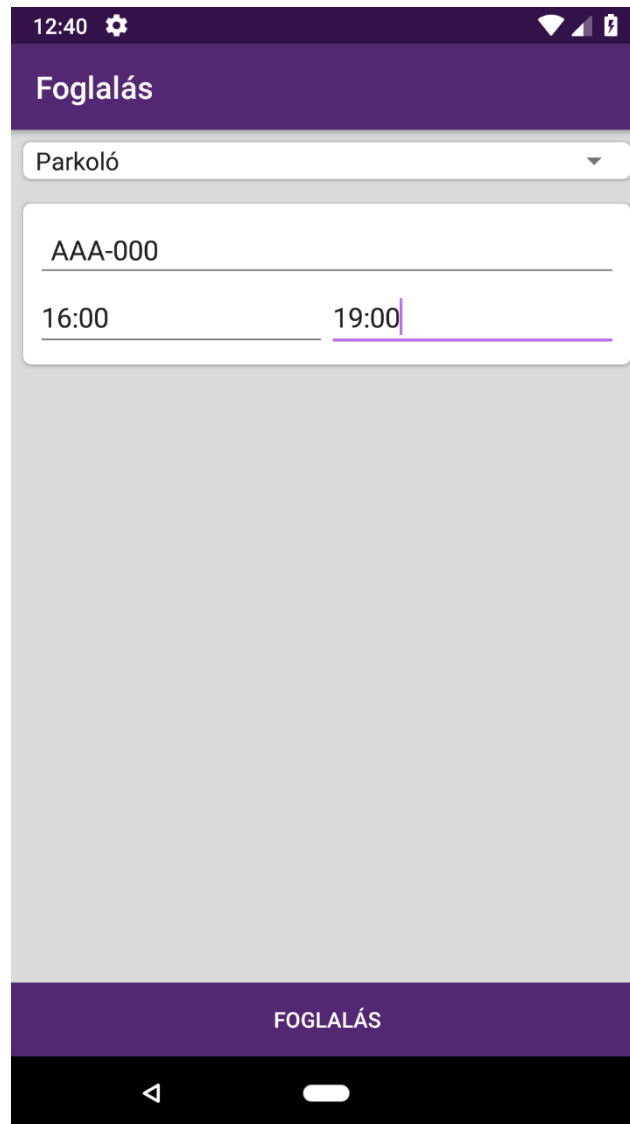
Ha lakóként a főmenüből a lakásunk adatai opcióra kattintunk, akkor **13.** számú ábrán lévő képernyőre jutunk. A lakás különböző szobáiba helyeztem el Raspberry Pi alapú mérőállomásokat, amelyek Green-IoT keretrendszer által nyújtott szolgáltatást használva mérik a szenzorokkal a szoba állapotát óránként, és itt meg tudjuk azt tekinteni.



14. ábra Választható opciók foglaláshoz

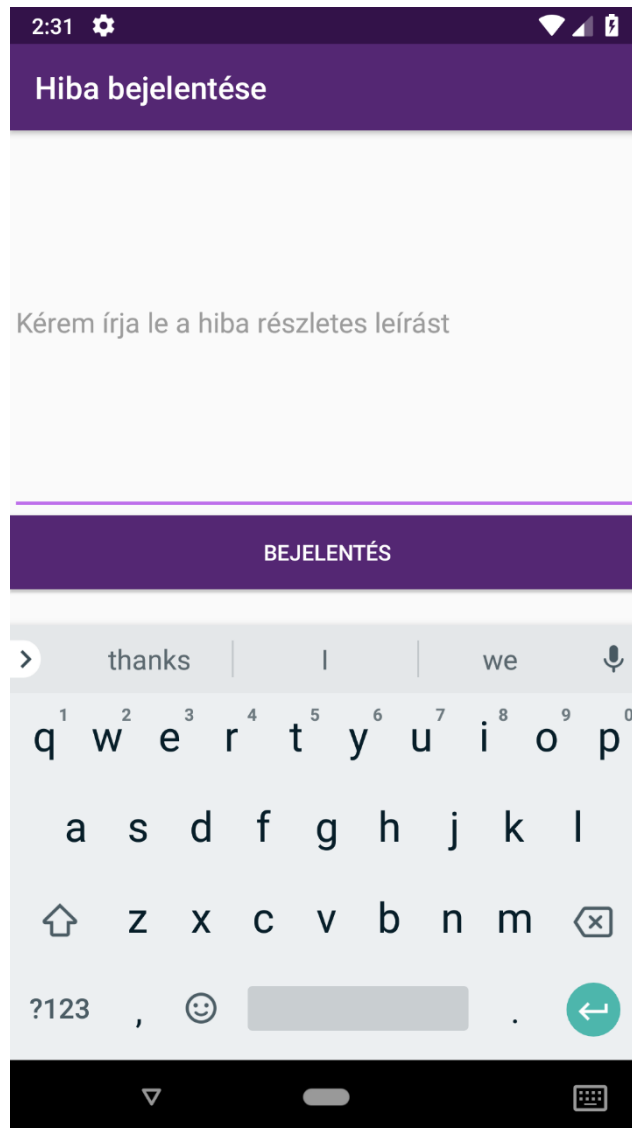
A lakópark éppen szabad erőforrásainak hasznosítására remek megoldás a nyilvánosság felé megnyitni azokat, hogy bárki hozzá tudjon férni, ha szeretne. A felhasználó által választható opciók kerülnek bemutatásra a **14.** ábrán.

Például ha a lakópark sok szabad parkolóhellyel rendelkezik, akkor főleg a vendégek számára hasznos lehet azok foglalása, ezáltal biztosan lesz helyük az ott tartózkodásuk idejére – mivel a lakók valószínűleg fix hellyel rendelkeznek. Emellett a lakók számára is hasznos a sportpályák – ebben a példában a tenispálya foglalása –, aki lefoglalja, annak biztosított a helye a sportolásra.



15. ábra Parkolóhely foglalásának folyamata

A **15.** ábrán látható a foglalás folyamata. A felhasználó kiválasztotta a parkolást, itt megjelenik egy újabb kártya, ahol a felhasználónak meg kell adnia az autójának a rendszámát és az intervallumot, amelyben ott szeretne tartózkodni.

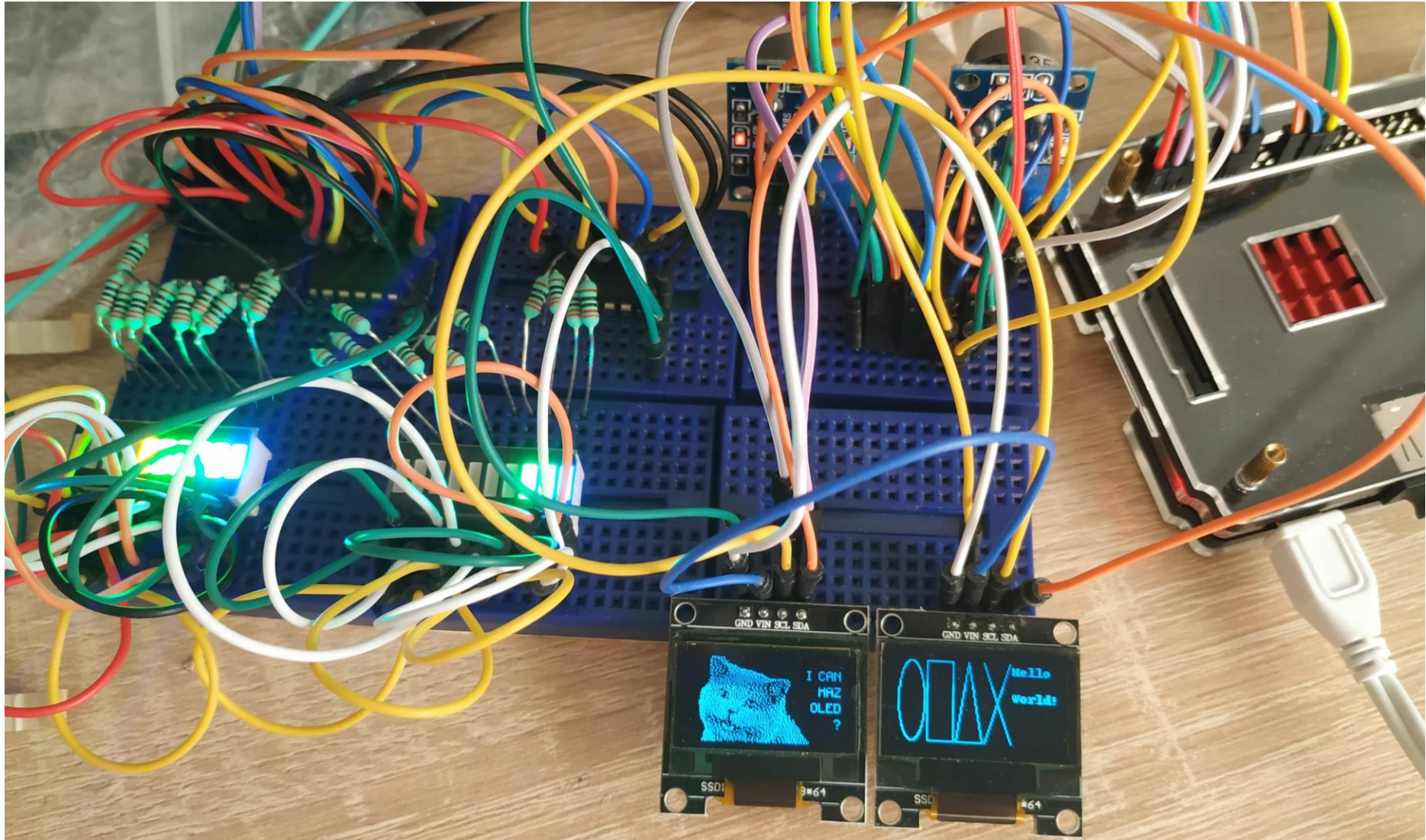


16. ábra Hiba bejelentése

A 16. ábrán szemléltetett képernyőn a felhasználó be tudja jelenteni az általa tapasztalt hibát az üzemeltetés felé, ezáltal az üzemeltetés tudomást szerez róla – ha még ez nem történt volna meg – és mihamarabb elkezdődhet a probléma elhárítása.

## 4.5 A Relay csomópont nyomtatott áramkörének elkészítése

Az kapcsolási rajz alapján elkészítettem az áramkör első próbapaneles verzióját, amely a 17. ábrán látható. Aztán megírtam hozzá a szoftvert is Python [20] programozási nyelven, amely Raspberry Pi-n széles körben támogatva van. Az áramkört vezérlő szoftver első sorban felhasználói interakcióra reagál, tehát gombnyomásra indul egy manuális mérés, majd az eredmény a kijelzőn kerül megjelenítésre.



17. ábra Próbapanelen összerakott félkész áramkör tesztelés közben

Építés során néhány kisebb hibát fedeztem fel az áramkörömben, ezeket javítottam. A javított kapcsolási rajzból aztán nyomtatott áramkört terveztem.

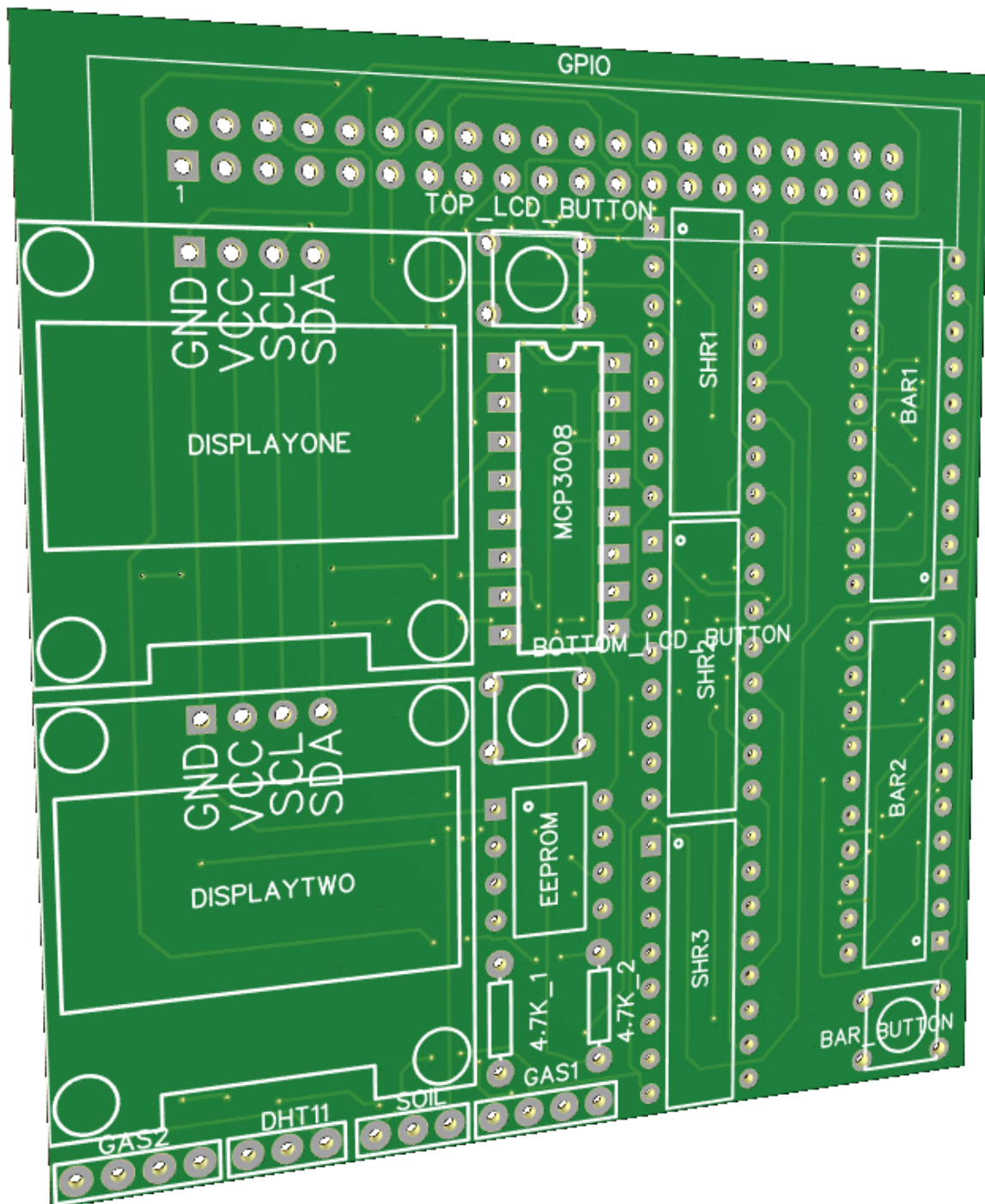
A 18. ábrán személtetett kijelzőből kettőt használtam fel, viszont azok egy I2C buszon osztoznak, alapesetben azonos a címük. Olyan kijelzőt kerestem, amelynek meg lehet változtatni a címét, a hátulján lévő ellenállás átforrasztásával. A felirattal ellentétben nem a 0x78 vagy 0x7A címen érhető el a kijelző, hanem a 0x3C vagy 0x3D címen.



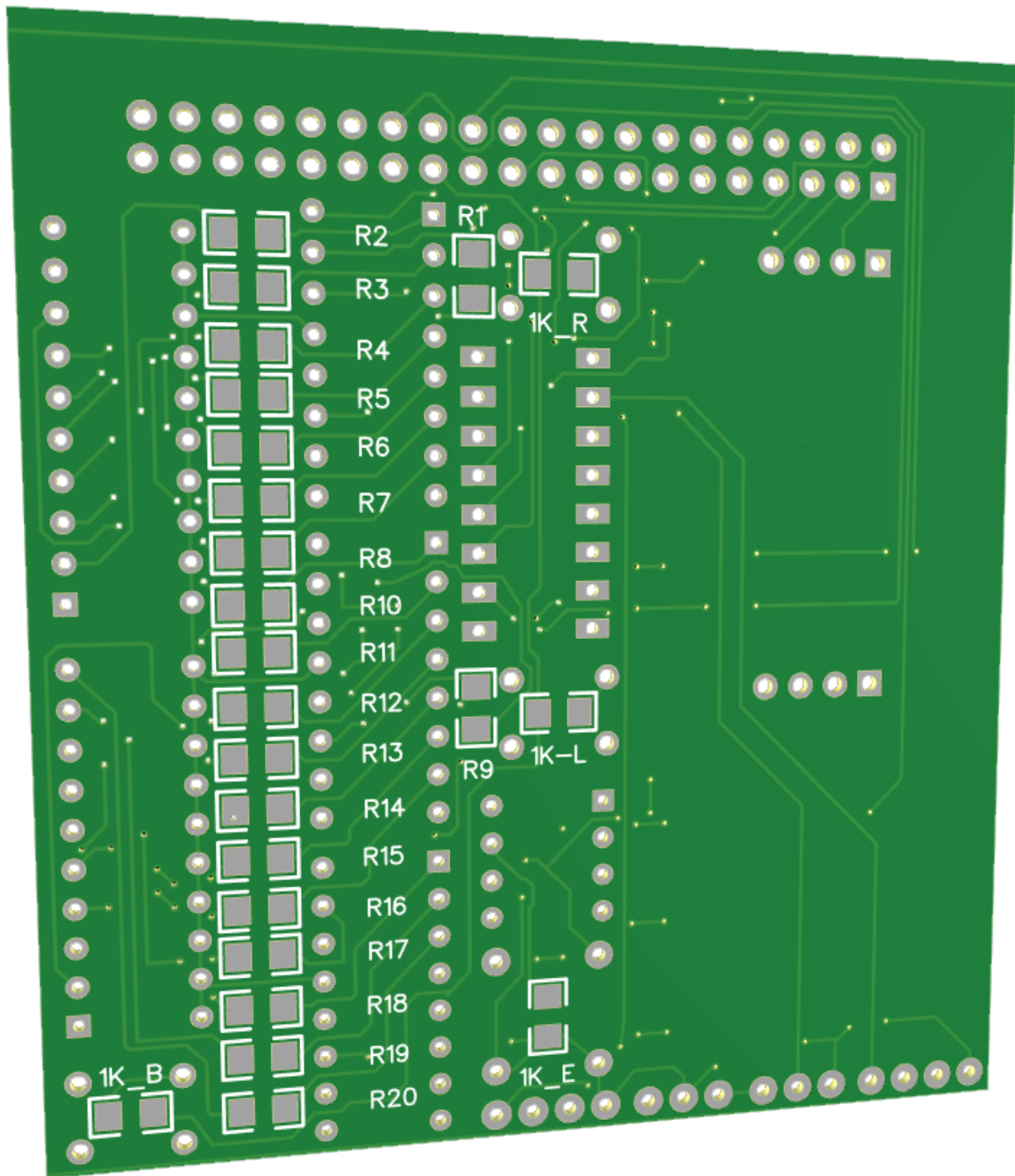
18. ábra 128x64 pixeles OLED LCD

Miután minden elem a helyére került a próbapanelen, elkészítettem hozzá a vezérlő szoftvert [21]. A szoftver esemény-vezérelt módon működik, azaz a felhasználó gombok lenyomása által kiváltott eseményeket kezeli. 3 darab gombot helyeztem el az áramkörben, a két kijelzőnek egyet-egyét és a 2 darab LED oszlopoknak egyet. A bal oldali kijelzőhöz tartozó gomb lenyomása esetén elkezdődik a két levegő-minőség szenzorral egy-egy mérési folyamat és a mért eredmény kerül kirajzolásra. A jobb oldali kijelzőhöz tartozó gombbal lehet hőmérsékletet mérni, a mérési folyamat végeztével a hőmérséklet és a páratartalom aktuális értéke jelenik meg a kijelzőn. A harmadik gombbal szintén a levegő-minőség szenzorokkal

lehet mérést végezni, de a kijelzők helyett az oszlopokon kerülnek megjelenítésre az eredmények.

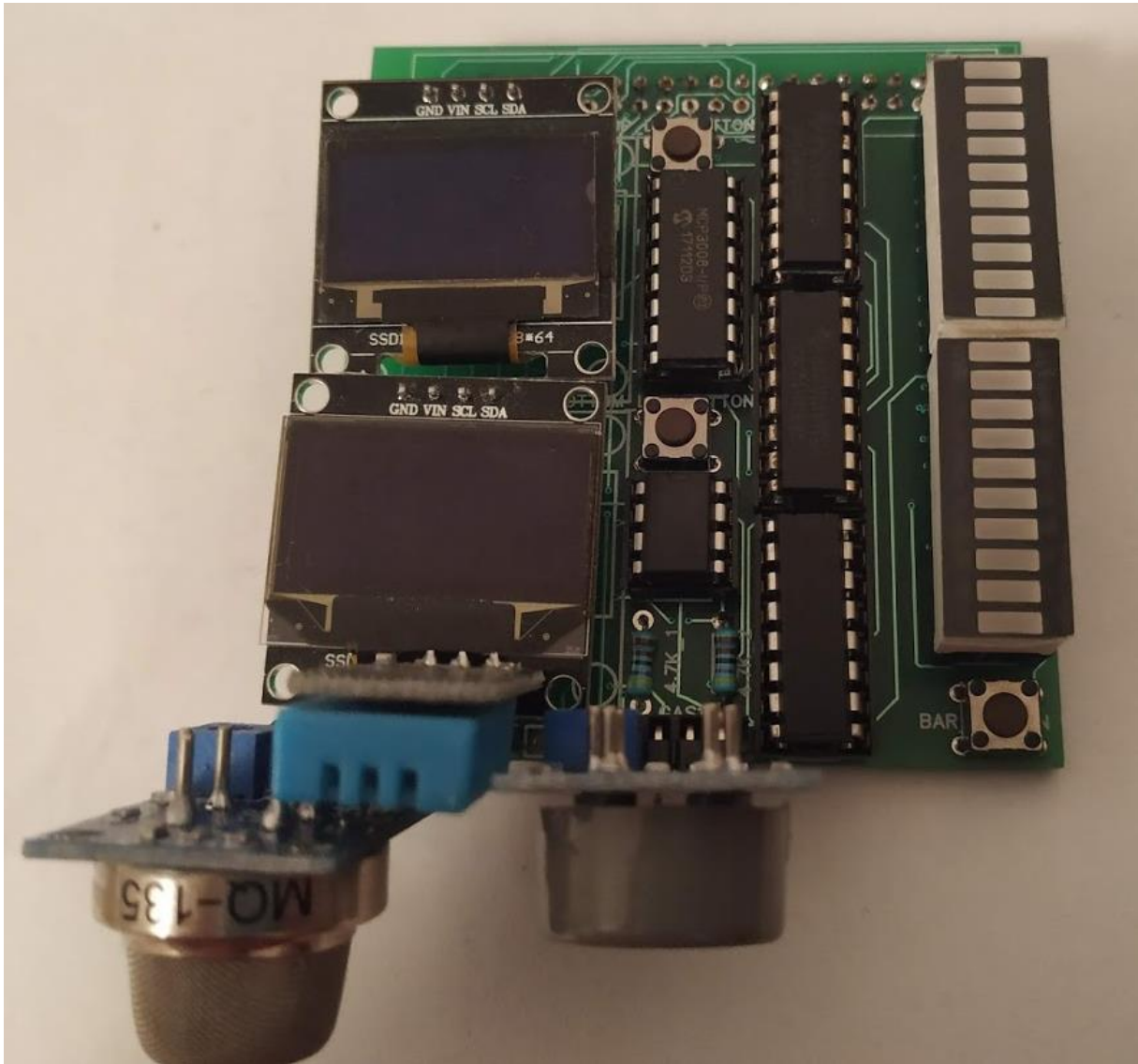


19. ábra Nyomatott áramkör előlapjának 3D modellje



20. ábra Nyomtatott áramkör hátlapjának 3D modellje





21. ábra Elkészült nyomtatott áramkör

Az EasyEDA [22] nevű online elérhető áramkör-tervező szoftverrel terveztem meg mind a kapcsolási rajzot mind a NYÁK-ot is. Könnyen használható és intuitív felületének hála, egész gyorsan beletanultam, tekintve, hogy ez az első áramköröm. Az EasyEDA által generált foto-realistikus kép látható a 19. és 20. ábrán, az elkészült NYÁKról. A nyomtatott áramkört a BME-n található UNIPCB Kft-n [23] keresztül rendeltem meg. A 21. ábrán látható a kész nyomtatott áramkör.

## 5. Vizsgálatok és eredmények

A Green-IoT kereterendszer új API-ját Unit, Integration és manuális teszteknek vettem alá. A manuális teszteket a Postman [25] nevű program segítségével végeztem. Unit és Integration tesztekhez Chai [26] és Mocha [27] JavaScript könyvtárakat használtam.

### 5.1 Unit tesztek

Unit teszteket a Green-IoT Serviceken végeztem. (lásd 22. és 23. ábra), hiányzó és helytelen bemeneti adatokat teszteltem.

```
service.user
  get user data by id
    ✓ should not get user if no user id is given
    ✓ should not get user if bad user id is given
    ✓ should get user if valid id is given

108 passing (14s)
```

22. ábra Összesen 108 Unit tesztet készítettem

A 23. ábrán található kódrészlet egy unit tesztet mutat be. A teszt által generált felhasználónak létrehozunk két Relay csomópontot, eltérő névvel. Majd lekérjük a felhasználó csomópontjait a filter paraméter alapján. A visszakapott eredményen vizsgálatot végzünk, ha minden feltételnek megfelel, akkor sikeres a teszt. Ellenkező esetben sikertelen.

```
it('should get 1 relay by filter', async () => {
  try {
    userId = user.id
    const relayName1 = 'Development Relay'
    const relayName2 = 'Test Relay'
    filter = relayName1.substring(1, 6)
    await generateRelay(relayName1, testHomeId)
    await generateRelay(relayName2, testHomeId)

    const result = await RelayService.getRelaysByUser(testHomeId, userId, filter)
    expect(result).to.exist
    expect(result.count).to.equal(1)
    expect(result.relays.length).to.equal(1)
    expect(result.relays[0].name).to.equal(relayName1)
  } catch (error) {
    throw Error('should not be reached')
  }
})
```

23. ábra Példa egy unit tesztre

## 5.2 Integration tesztek

Integration teszteket a végpontokon végeztem el, itt is ellenőriztem a bemeneti paramétereket, hozzáférési jogosultság hiányát – ezekre példát a **24.** és **25.** ábrán láthatunk.

```
integration.home
get homes by user
  ✓ should not get homes by user if no user id is given
  ✓ should get homes by user if user id is given
  ✓ should get empty homes by user if filter does not match
  ✓ should get one home if filter matches
create home by user
  ✓ should not create home if no user id is given
  ✓ should not create home if no name is given
  ✓ should create home if all params are given
  ✓ should create home if no lat long is given
create home by email
  ✓ should not create home by email if no email is given
  ✓ should not create home by email if bad email is given
  ✓ should create home by email if valid email is given

77 passing (10s)
```

24. ábra Összesen 77 integration tesztet készítettem

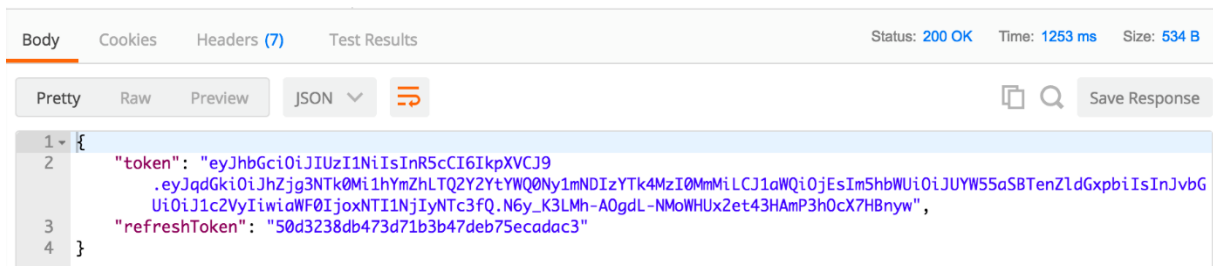
```
it('should not authenticate the user with wrong email', (done) => {
  agent
    .post('/api/v1/user/login')
    .send({
      email: normalUser.email + 'wrong',
      password: normalUser.password
    })
    .end((err, res) => {
      expect(err).to.exist
      expect(res).to.exist
      expect(res.status).to.equal(400)
      done()
    })
  })
})
```

25. ábra Példa egy Integration tesztre

Az integration tesztek végpontokat vizsgálnak. A 25. ábrán látható integration teszt, a bejelentkezésért felelős végpontot teszteli. Jelenlegi teszt esetben helytelen email jelszó párossal próbál bejelentkezni, de a szerver visszautasítja az autentikációt.

## 5.3 Manuális tesztek Postman segítségével

Fejlesztés folyamán végig próbálgattam az API-t, hogy az elvárásoknak megfelelően működik-e, illetve a válasz megfelelő formátumú-e. A **26.** ábrán egy sikeres bejelentkezéshez tartozó válasz látható.



26. ábra Bejelentkezésre kapott válasz

## 5.4 Nyomatott áramkör tesztje

Ha a Raspberry Pi-hez már boot időben van csatlakoztatva egy ID EEPROM-ot tartalmazó áramkör, akkor a memória adatai beolvasásra kerülnek. Ilyenkor létrejön egy új *hat* nevű mappa a */proc/device-tree/* könyvtárban. A 27. ábra szemlélteti ha a mappából kiíratjuk a vendor és product fájlok tartalmát.

```
pi@raspberrypi_plus:/proc/device-tree/hat $ more vendor
Szvetlin Tanyi
pi@raspberrypi_plus:/proc/device-tree/hat $ more product
Szvetlin Smart Home Board V1
pi@raspberrypi_plus:/proc/device-tree/hat $
```

27. ábra HAT mappában található vendor és product fájlok értékei

## 5.5 Lakópark rendszereinek összekapcsolt működése

A Green-IoT kereterendszer segítségével a felhasználók hozzáférnek a saját otthonuk által mért adatokhoz, viszont a Green-IoT Arrowhead Servicén keresztül az otthonok képesek megosztani a mérési eredményeiket az üzemeltetés felé. Az Arrowheadhez készült egy grafikus felhasználói felület, amely az üzemeltetés számára megkönnyíti a lakópark fenntartását. A felületen láthatják az egyes otthonok Relay csomópontjai és szenzorai által mért adatokat.

Az általam készített megoldás még nem teljes, de egy nagyon erős alapot nyújt a jövőbeli fejlesztések számára.

## 6. Kiberbiztonság

Egy ilyen nagy rendszer esetében a tervezés pillanatától kezdve kulcsfontosságú szerepet kap a biztonság. Nagyon fontos, hogy illetéktelen hozzáférések ne történjenek, és szenzitív adatokhoz se férjenek hozzá.

A Green-IoT okos otthon keretrendszer használatakor az otthonokban lévő eszközök esetén megbízunk a WiFi szabvány által támogatott biztonsági megoldásokban. A szenzorok nem kommunikálnak a belső hálózatról kifelé, csak a Relay-en keresztül. A szenzorok csak helyi hálózaton működnek. A Cloud – Relay, Cloud – alkalmazások közötti kommunikáció biztonságát a HTTPS biztosítja, ehhez szükséges az SSL tanúsítvány.

A Green-IoT keretrendszer API struktúrája is ezt az irányelvet követve lett kialakítva és 3 részre szeparálva. Ez magába foglalja a jogosultságok kezelését is, ezáltal felhasználók nem képesek hozzáférni más felhasználók adataihoz.

Az Arrowhead keretrendszer biztonságáért az Authentication Service felel. A rendszerek hozzá fordulnak, hogy identifikálják magukat és jogosultságaikat lekérdezzék. Ez a megközelítés különösen hasznos abból a szempontból, hogy a szolgáltatásokat nyújtó rendszereknek nem kell autentikációval és autorizációval foglalkozniuk, mert ezt elvégzi az Authentication Service helyettük.

Lokális felhők és az Arrowhead keretrendszer közötti kommunikáció biztonságát szintén a HTTPS biztosítja. Az Arrowhead keretrendszer is a lehető legbiztonságosabban lett megtervezve, decentralizált architektúrájából adódóan.

## **7. Jövőbeni tervek**

### **7.1 Mobil alkalmazás továbbfejlesztése**

A mobil alkalmazást több irányba szeretném továbbfejleszteni. Először is cross platform megoldást szeretnék megvalósítani React Native-ben [24], ezáltal nem csak Androidon lenne elérhető. Továbbá funkcionalitásában is szeretném bővíteni. A felhasználó legyen képes a saját adatainak beállítására, illetve akár a szabad parkolóhelyek számát is meg lehessen jeleníteni, foglalás során.

### **7.2 A Relay csomópont áramkörének továbbfejlesztése**

Mivel ez a nyomtatott áramkör első változata amit készítettem, tapasztalatlanságomból adódóan számos, a funkcionalitást nem érintő, de a használatot zavaró problémák léptek fel. A szenzorok elhelyezése nem lett a legszerencsésebb, nagyon közel helyezkednek el összezsúfolva, az alsó kijelzőt félig kitakarva. Gombok is az áramkör közepén helyezkednek el, hozzáférésük nehézkes. Illetve a 2 darab 10 LED-ből álló oszlop is nagyon jó ötletnek tűnt, mint indikátor, de a kijelzőkkel egyszerűbben meg lehet jeleníteni, akár több adatot is és nincsen szükség további 3 shift regiszterre sem. Ezáltal tovább csökkenne az áramkör mérete.

### **7.3 Arrowhead front-end továbbfejlesztése**

Az Arrowhead menedzsment felületét is szeretném tovább fejleszteni, további rendszerek implementációját megvalósítani – például az Authorizáció, az Esemény kezelő, valamint a System és Device Registry rendszerekét.

### **7.4 3D nyomtatott egyedi tok**

Hasznos fejlesztés lenne 3D nyomtatóval készíteni egy tokot hozzá, amelybe a továbbfejlesztett áramkör kapna helyet, ezzel egy teljes csomagot alkotva.

## 8. Összefoglalás

A feladat motivációjaként bemutattam a jelenleg egyetlen magyar intelligens lakópark megoldást, megállapítottam az előnyeit és hátrányait. Ezeket figyelembe véve egy több előnnyel rendelkező irányt jelöltem ki saját rendszeremnek. Ennek jellemzői:

- Önállóan működő okos otthonok;
- Üzemeltetés felé hibajelzés;
- Kihasználatlan erőforrások hasznosítása;
- Megújuló energiaforrások használata;
- Biztonságra vonatkozó irányelvek.

Külön kiemelendő az a biztonsági szempontból jelentős előny, hogy a rendszer által gyűjtött adatokkal a felhasználó rendelkezik; ezeket csak az ő beleegyezésével lehet megosztani az üzemeltetés felé.

Dolgozatomban bemutattam az Intelligens lakópark tervezését, a rendszer felépítését. Itt kitértem a hardveres és szoftveres megvalósításra. Részletesen leírtam ezek tervezési és implementációs lépéseit. Bemutattam, hogyan lehet szenzor adatokat az Arrowhead-en keresztül megosztani.

Az általam tervezett rendszert unit és integration tesztekkel is verifikáltam.

Dolgozatomban dokumentáltam a keretrendszer verifikációjának főbb elemeit, kitértem a kiberbiztonságra is, mivel ez is elengedhetetlen része a keretrendszernek.

A bemutatott, saját tervezésű és fejlesztésű rendszer egy, a követelményeknek megfelelő, alaposan átgondolt rendszer, amely komponenseit a jövőben lehet fejleszteni.

Látszik, hogy ez egy korszerű, nem lezárt terület, a Gartner görbe lelegején tart, kutatási szinten. Egyelőre nincsenek még olyan megoldások, amelyeket minden problémára megoldást nyújtanának.

# Irodalomjegyzék

Minden webes hivatkozás 2018. 10. 26.-án elérhető volt.

- [1] Gartner Inc. – Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017
- [2] J. Delsing (ed.) – IoT Automation: Arrowhead Framework, CRC Press, 2017.
- [3] Bill Gates on Storing electricity <https://www.gatesnotes.com/Energy/It-Is-Surprisingly-Hard-to-Store-Energy>
- [4] Immonova intelligens lakópark megoldás <https://immonova.hu/tarsashazak-lakoparkok/>
- [5] Arrowhead <http://www.arrowhead.eu/>
- [6] Productive 4.0 <https://productive40.eu/>
- [7] Green-IoT keretrendszer okos otthonokhoz <http://tdk.bme.hu/VIK/Halozatok2/GreenIoT-keretrendszer-okos-otthonokhoz>
- [8] Raspberry Pi <https://www.raspberrypi.org/>
- [9] ESP8266 <https://en.wikipedia.org/wiki/ESP8266>
- [10] Átlagos áram évi áram fogyasztás háztartásonként <https://www.nkmaram.hu/pages/aloldal.jsp?id=550565>
- [11] Airbnb <https://www.airbnb.hu/>
- [12] Raspberry Pi HAT <https://www.raspberrypi.org/blog/introducing-raspberry-pi-hats/>
- [13] Intellio <https://intellio.eu/index.php/hu/>
- [14] Cs. Hegedus, D. Kozma, G. Soos, P. Varga, “Enhancements of the Arrowhead Framework to refine intercloud service interactions,” in 42<sup>nd</sup> Annual Conference of IEEE Industrial Electronics Society (IECON), Florence, Italy, 2016
- [15] NodeJS <https://nodejs.org/en/>
- [16] ReactJS <https://reactjs.org/>
- [17] MongoDB <https://www.mongodb.com/>



- [18] MySQL <https://www.mysql.com/>
- [19] Docker <https://www.docker.com/>
- [20] Python <https://www.python.org/>
- [21] Áramkört vezérlő program <https://github.com/tsvetlin/rpihat>
- [22] EasyEDA <https://easyeda.com>
- [23] UNIPCB <http://www.unipcb.hu/>
- [24] React Native <https://facebook.github.io/react-native/>
- [25] Postman <https://www.getpostman.com>
- [26] Chai <http://www.chaijs.com>
- [27] Mocha <https://mochajs.org>