Követelményspecifikáció

## **1.1 Feladatkiírás**

A hallgatók feladata egy olyan térkép alapú online rendszer kifejlesztése, amelyik lehetővé teszi felhasználóinak, hogy megosszák az általuk ismert ingyenes/előnyös parkolási lehetőségeket a többi felhasználóval. A rendszer legyen elérhető mobil készülékekről. Az alkalmazás tegye lehetővé fényképek feltöltését és a cím alapú keresést.

## **1.2 A fejlesztői csapat**

|  |  |  |
| --- | --- | --- |
| **Csapattag neve** | **Neptun-kód** | **E-mail cím** |
| Gyönki Bendegúz | IZZT5E | gyonkibendeguz@gmail.com |
| Fodor Árpád | S4AZIE | arpadfodor01@gmail.com |

A csapatban a következőképpen osztottuk fel a feladatokat:

* **Bendegúz** a backend (szerveralkalmazás) fejlesztésével foglalkozik
* **Árpád** a frontend (Android kliens) fejlesztését végzi

## **1.3 Részletes feladatleírás**

Célunk a projekt keretében egy olyan rendszer készítése, amellyel a felhasználók bárhol képesek parkolási lehetőségek bejelentésére és megtekintésére. Ezáltal adott területek monitorozására nyílik lehetőség, ami megkönnyítheti a városban autóval járók mindennapjait.

Az alábbiakban ismertetjük az Android operációs rendszerre készített kliens és a szerveroldali alkalmazás tervezett funkcióit.

**1.3.1 Android alkalmazás**

Egy bejelentéshez a GPS koordináták és a szükséges adatok megadásán felül (pl. fizetős/ingyenes parkolás) lehetőség van képek feltöltésére is, amelyek révén a többi felhasználó könnyebben találhatja meg az adott helyet. A képek a gyors bejelentés céljából az alkalmazásban készülhetnek, de lehetőség van eszközön tárolt fotók feltöltésére is.

A felhasználók többféle módon böngészhetnek a bejelentések között. Egyrészt egy interaktív térképen vizuálisan láthatják a bejelentéseket, illetve lehetőség van cím alapján történő keresésre is. A kiválasztott parkolóhely részletes adatai (pozíció, típus, bejelentés ideje, kép) megtekinthetők, illetve navigáció indítása is lehetséges az adott helyre.

Egy felhasználó az alábbi műveleteket hajthatja végre egy potenciális parkolóhely esetén:

* Megadja a tulajdonságait (pl. fizetős/ingyenes)
* Képet készít hozzá
* Bejelenti

Egy már bejelentett parkolóhelyen a felhasználók az alábbi műveleteket végezhetik:

* Lefoglalás (egy felhasználó jelzi, hogy az adott helyre tart, oda fog parkolni)
* Foglalás levétele
* Módosítás (pl. egy felhasználó jelzi, hogy az adott parkolóhelyen már áll valaki; vagy képet fűz hozzá)

### **1.3.2 Szerveralkalmazás**

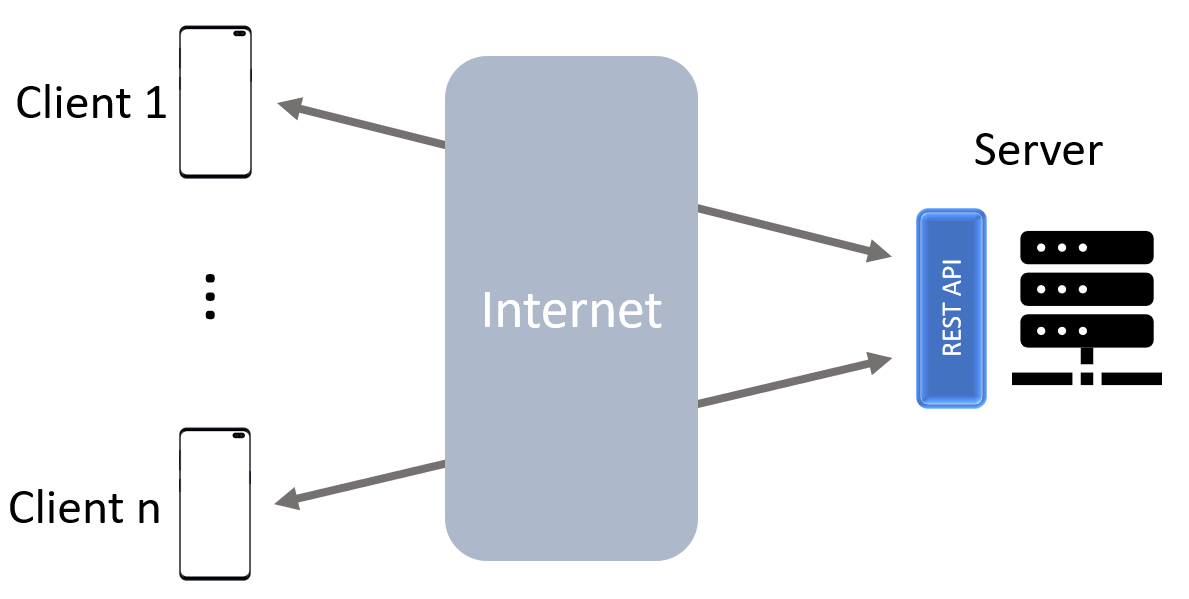
A szerveroldali alkalmazás az alábbi, parkolóhelyekkel kapcsolatos műveleteket támogatja:

* Új parkolóhely hozzáadása az adatbázishoz
* Meglévő parkolóhely adatainak módosítása az adatbázisban
  + Parkolóhely lefoglalása
  + Foglalás levétele
* Meglévő parkolóhely törlése az adatbázisból
* Parkolóhely keresése
  + GPS koordináták alapján és a kliens által megadott sugarú körben

A szerveroldali alkalmazás felel a felhasználók nyilvántartásáért is. A szerver a felhasználók alábbi adatait tárolja: felhasználónév, email cím, jelszó (titkosított formában). A szerver lehetővé teszi új felhasználó regisztrálását, illetve a felhasználói adatok módosítását.

## **1.4 Architektúra**

A rendszert kliens-szerver architektúra alapján tervezzük megvalósítani. A kliensek Android alkalmazások, a szerver pedig egy REST API-t biztosít a kliensek számára. A bejelentett parkolóhelyek, képek, koordináták a szerveren tárolódnak, a kliensek ide tudnak bejelenteni új dolgokat, vagy lekérdezni az aktuális állapotot. A kommunikáció HTTPS protokollon keresztül történik, a felhasználók azonosítása pedig HTTP Basic authentikációval történik.



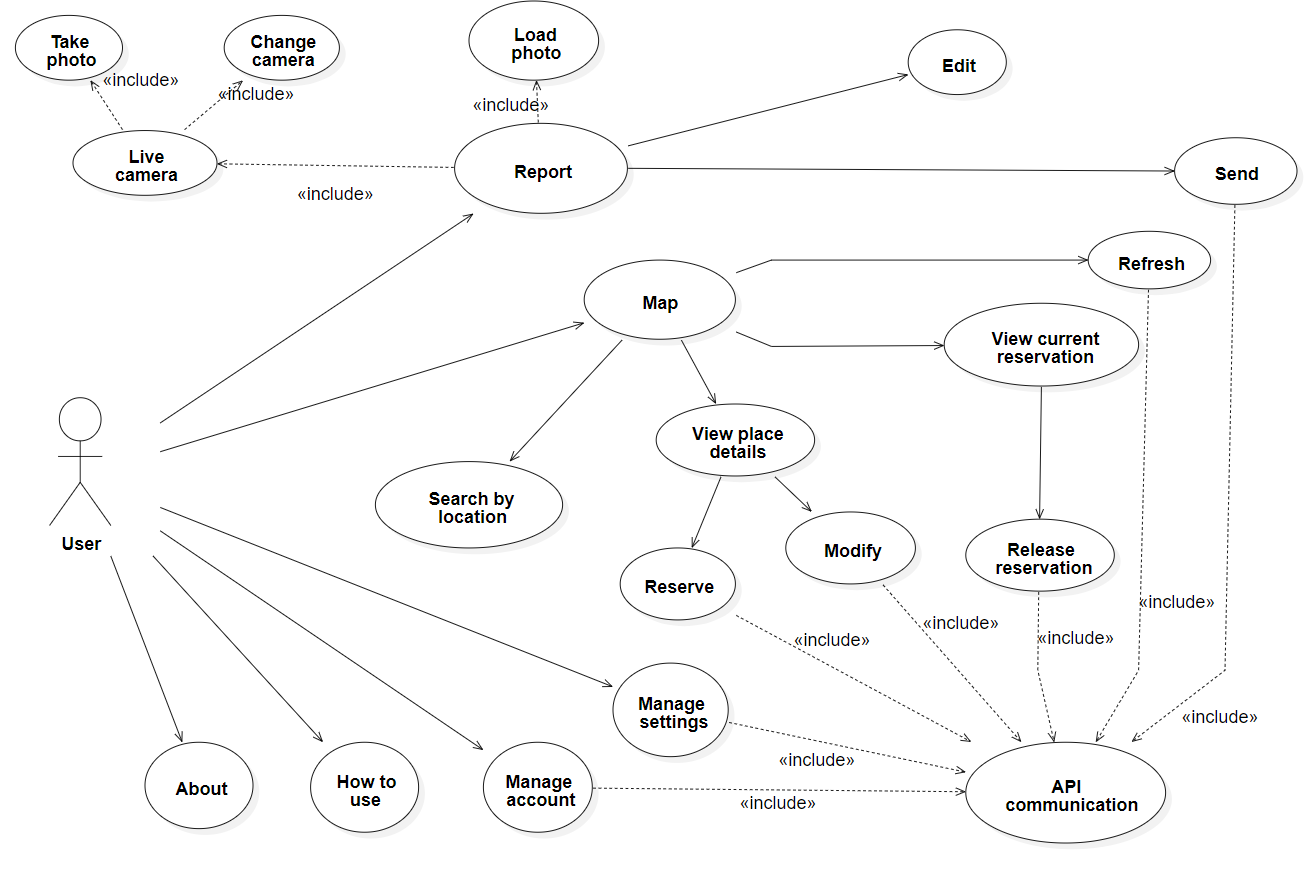
ábra 1: Kliens-szerver architektúra

## **1.5 Technikai paraméterek**

A rendszer kliensoldali része Android platformra készül. Az okostelefonok napjainkra széles körben elterjedtek, a legdominánsabb operációs rendszer pedig jelenleg az Android, ezért ezt a platformot célozva potenciálisan széles lehet a felhasználók köre. A fejlesztés Kotlin nyelven történik. Az alkalmazás specifikus perzisztens adatok SQLite alapú Room adatbázisban kerülnek tárolásra. A kamerakezelés a CameraX API segítségével kerül megvalósításra. A térkép megjelenítéséhez a Google Maps API nyújt segítséget, míg a szerverrel történő hálózati kommunikációhoz a Retrofit kerül felhasználásra.

A szerveralkalmazás fejlesztése is Kotlin nyelven történik, a Ktor keretrendszer felhasználásával. A Ktor egy aszinkron keretrendszer mikroszolgáltatások és webalkalmazások fejlesztéséhez. A szerver az adattároláshoz egy PostgreSQL adatbázist használ, az adatbázisműveletek pedig a Jetbrains által készített Exposed ORM keretrendszer használatával kerülnek megvalósításra.

## **1.6 Use-case diagram**



ábra 2: Android app use-case