Rendszertervezés
és
Analízis modell

-

Ingatlankezelő Szoftver

Zöld Pont Hálózat KFT.

Gyakorlat

Készítette:
Aros Damján BSc - EFEW32
Tán Gergő BSc -BLCL2O
Ródé Martin BSc - DRPPXL

Sárospatak, 2025

Tartalomjegyzék

6. Integrációs terv

6.1 Külső API-k

6.2 Adatszinkronizáció és hibakezelés

7. Biztonsági terv

7.1 Hitelesítés és autorizáció

7.2 Adattitkosítás

7.3 GDPR megfelelőség

8. Tesztelési stratégia

8.1 Tesztelési típusok

8.2 Tesztesetek és forgatókönyvek

9. Üzemeltetési és karbantartási terv

9.1 CI/CD folyamat

9.2 Monitorozás és riasztások

9.3 Frissítési stratégia

10. Következő lépések és idővonal

10.1 Fejlesztési fázisok

10.2 Kritikus mérföldkövek

11. Függelék – Szótár és rövidítések

1. Bevezetés

1.1 Dokumentum célja

Ez a dokumentum az Ingatlankezelő Szoftver rendszertervezési és analízis modelljét mutatja be. A dokumentum célja, hogy átfogó útmutatást nyújtson a fejlesztési csapatnak, a projektmenedzsereknek és az érdekelt feleknek a rendszer architektúrájának, funkcionalitásának és adatkezelésének megvalósításához. A dokumentum részletesen leírja a rendszer követelményeit, felhasználói szerepköröket, integrációs pontokat és biztonsági protokollokat, hogy minden résztvevő egyértelműen megértse a rendszer működését és céljait.

1.2 Projekt áttekintése

Az Ingatlankezelő Szoftver egy web alapú platform, amely célja, hogy hatékony eszközt nyújtson az ingatlanok hirdetéséhez, kereséséhez és kezeléséhez. A rendszer főbb jellemzői közé tartozik a felhasználóbarát felület, a térképes keresés Google Maps API-val, a biztonságos üzenetküldés és az adminisztrációs felület a tartalmak moderálásához. A platformot úgy terveztük, hogy mind a magánszemélyek, mind az ingatlanközvetítők számára könnyen használható legyen, miközben figyelembe veszi a magyar piac specifikus igényeit.

1.3 Felhasználói szerepkörök

A rendszer négy fő felhasználói szerepkört különböztet meg: vendég, regisztrált felhasználó, prémium felhasználó és adminisztrátor. A vendégek csak az alapvető keresési funkciókat használhatják, míg a regisztrált felhasználók hirdetéseket adhatnak fel és üzeneteket küldhetnek. A prémium felhasználók korlátlan számú hirdetést tehetnek közzé

és statisztikai adatokhoz férhetnek hozzá. Az adminisztrátorok felelősek a tartalmak moderálásáért, a felhasználói fiókok kezeléséért és a rendszer beállításainak konfigurálásáért.

---

2. Rendszerarchitektúra

2.1 Magas szintű architektúra

A rendszer microservice alapú architektúrát követ, amely három fő rétegből áll: frontend, backend és adatbázis. A frontend React vagy Angular keretrendszerrel készül, és felelős a felhasználói felület megjelenítéséért. A backend Node.js vagy Spring Boot alapú, és kezeli az üzleti logikát és API-kat. Az adatbázis réteg hibrid megoldást alkalmaz, ahol a strukturált adatok PostgreSQL-ben, míg a rugalmas adatok MongoDB-ben kerülnek tárolásra.

2.2 Komponensdiagram

A rendszer fő komponensei közé tartozik a felhasználókezelés, a hirdetéskezelés, a keresőmodul és az üzenetküldő rendszer. A felhasználókezelés felelős a regisztrációért, bejelentkezésért és profilkezelésért. A hirdetéskezelés lehetővé teszi a hirdetések létrehozását, szerkesztését és törlését. A keresőmodul szöveges és térképes keresést biztosít, míg az üzenetküldő rendszer privát kommunikációt tesz lehetővé a felhasználók között.

2.3 Telepítési diagram

A rendszer felhőalapú infrastruktúrán fut, AWS vagy Google Cloud Platform szolgáltatásokkal. A konténerizáció Docker és Kubernetes segítségével történik, ami lehetővé teszi az erőforrások hatékony skálázását. A képek és statikus tartalmak gyors terjesztéséhez CDN-t (Content Delivery Network) használunk.

---

3. Funkcionális követelmények

3.1 Use Case diagram

A rendszer fő használati esetei közé tartozik a hirdetésfeladás, a keresés, az üzenetküldés és a kedvencek mentése. A hirdetésfeladás csak regisztrált felhasználók számára elérhető, és moderálásra szorul, mielőtt nyilvánosságra kerül. A keresés lehetővé teszi a szűrők alkalmazását és a térképes megjelenítést. Az üzenetküldés privát kommunikációt biztosít a felhasználók között, míg a kedvencek mentése lehetővé teszi a kiemelt hirdetések elmentését.

3.2 Főbb funkcionális modulok

A rendszer négy fő modulból áll: felhasználókezelés, hirdetéskezelés, keresőmodul és üzenetküldő rendszer. A felhasználókezelés lehetővé teszi a regisztrációt, bejelentkezést és profilkezelést. A hirdetéskezelés lehetővé teszi a hirdetések létrehozását, szerkesztését és törlését. A keresőmodul szöveges és térképes keresést biztosít, míg az üzenetküldő rendszer privát kommunikációt tesz lehetővé.

3.3 Részletes Use Case leírások

Hirdetésfeladás:

A hirdetésfeladási folyamat során a felhasználó egy struktúrált űrlapot tölt ki, amely több, kötelezően megadandó mezőt tartalmaz:

- ***Ingatlan típusa:*** *lakás, ház, iroda, telek stb..*
- ***Cím:*** *város, kerület, utca (opcionális házszám megadásával).*
- ***Alapterület:*** *négyzetméterben megadott érték.*
- ***Szobák száma:*** *teljes értékű szobák száma.*
- ***Ár:*** *forintban megadott vételár vagy bérleti díj.*
- ***Leírás:*** *szabad szöveges mező, az ingatlan részletes bemutatására.*
- ***Képfeltöltés:*** *minimum egy, maximum 15 kép csatolása.*

A hirdetés kitöltése után a rendszer a következő lépéseket hajtja végre:

- ***Automatikus ellenőrzés:*** *ellenőrzi, hogy minden kötelező mező helyesen és teljesen ki van-e töltve, valamint hogy nincsenek tiltott szavak vagy nem megfelelő képek.*
- ***Moderálásra küldés:*** *a hirdetés nem jelenik meg azonnal nyilvánosan, hanem „függőben” státuszba kerül, és átvizsgálásra továbbítódik egy adminisztrátor felé.*
- ***Jóváhagyás vagy elutasítás:*** *az admin jóváhagyás után a hirdetés „aktív” állapotba kerül, és láthatóvá válik a felhasználók számára. Elutasítás esetén a rendszer értesítést küld az indoklással együtt.*

Hirdetéskeresés:

A keresési folyamat célja, hogy a felhasználók gyorsan és hatékonyan megtalálják a számukra legmegfelelőbb ingatlanokat.A felhasználó az alábbi szűrők alkalmazásával pontosíthatja a keresést:

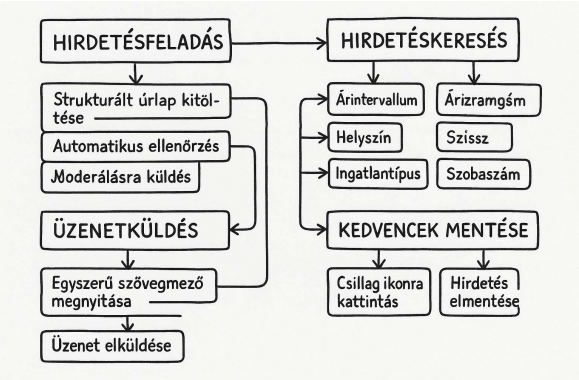
- ***Árintervallum:*** *minimum és maximum ár megadása.*
- ***Helyszín:*** *város, kerület, környék kiválasztása.*
- ***Ingatlantípus:*** *például lakás, családi ház, iroda.*
- ***Méret:*** *alapterület négyzetméterben.*
- ***Szobaszám:*** *minimum szobaszám megadása.*

A szűrés után a találatok listanézetben és térképes nézetben is megjelennek:

- *A lista valós időben frissül, ahogy a szűrők módosulnak.*
- *A térképes nézet a Google Maps API segítségével jeleníti meg a találatokat, jelölőkkel ellátva.*

Üzenetküldés:

Az üzenetküldő modul célja a biztonságos és gyors kapcsolatfelvétel az eladók és vevők között, anélkül, hogy személyes adatok közvetlenül kicserélődnének.



A használat menete:

- *A hirdetés adatlapján található **„Kapcsolatfelvétel”** vagy **„Üzenet küldése”** gombra kattintva a felhasználó megnyit egy egyszerű szövegmezőt.*
- *Az üzenet elküldése után:*
  - *Egy új beszélgetés jön létre a küldő és a címzett között.*
  - *Mindkét fél értesítést kap a beérkezett üzenetről e-mailben és a rendszer belső értesítési modulján keresztül.*
- *Az adminisztrátorok visszaélés esetén betekinthetnek a beszélgetésbe, szükség esetén le is zárhatják azt.*

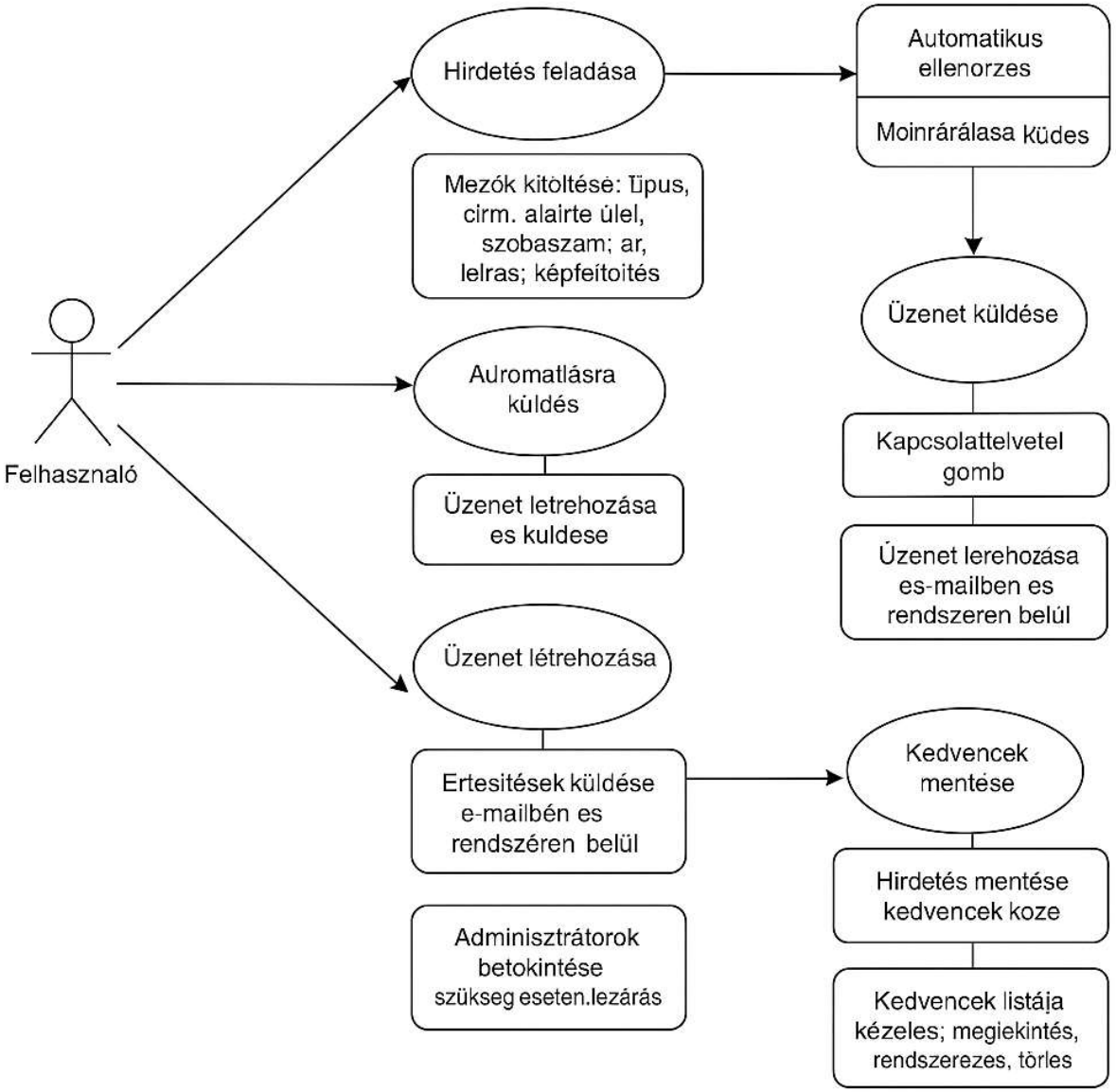
Kedvencek mentése:

A kedvencek funkció célja, hogy a felhasználók könnyedén elmentsék azokat a hirdetéseket, amelyek felkeltették az érdeklődésüket.

A működés:

- *Minden hirdetésnél megjelenik egy csillag ikon.*
- *A csillagra kattintva a hirdetés elmentésre kerül a felhasználó „Kedvencek” listájába.*
- *A kedvencek menüpont alatt a felhasználó áttekintheti, rendszerezheti és eltávolíthatja a mentett hirdetéseket.*
- *A funkció csak regisztrált és prémium felhasználók számára érhető el.*

Ez a funkció segít a felhasználóknak abban, hogy könnyen visszataláljanak az őket érdeklő ingatlanokhoz, még akkor is, ha több keresésen és böngészésen vannak túl.


Felhasznaló
Hirdetés feladása
Automatikus ellenorzes
Moinrárálasa küdes
Mezók kitöltése: Tipus, cirm. alairte úlel, szobaszam; ar, lelras; képfeítoités
Üzenet küldése
Auromatlásra küldés
Kapcsolattelvetel gomb
Üzenet letrehozása es kuldese
Üzenet lerehozása es-mailben es rendszeren belül
Üzenet létrehozása
Kedvencek mentése
Ertesitések küldése e-mailbén es rendszéren belül
Hirdetés mentése kedvencek koze
Adminisztrátorok betokintése szükseg eseten.lezárás
Kedvencek listája kézeles; megiekintés, rendszerezes, törles

4. Adatmodell és adatbázis tervezés

4.1 Entitás-kapcsolat diagram (ERD)

Az adatmodell fő entitásai a következők:

- ***Felhasználók (Users):***

  A rendszer regisztrált felhasználói, akik hirdetéseket adhatnak fel, üzeneteket küldhetnek és fogadhatnak.

  - *Attribútumok: Felhasználó azonosító (user_id), név, email cím, jelszó, regisztráció dátuma.*

- ***Hirdetések (Ads):***

  A felhasználók által közzétett hirdetések. Egy felhasználó több hirdetést is létrehozhat.

  - *Attribútumok: Hirdetésazonosító (ad_id), cím, leírás, ár, létrehozás dátuma, felhasználó_id (külső kulcs).*

- ***Képek (Images):***

  A hirdetésekhez tartozó képek. Egy hirdetéshez több kép is kapcsolódhat, de egy kép csak egy hirdetéshez tartozhat.

  - *Attribútumok: Kép azonosító (image_id), fájlnév, fájl elérési útja, hirdetés_id (külső kulcs).*

- ***Üzenetek (Messages):***

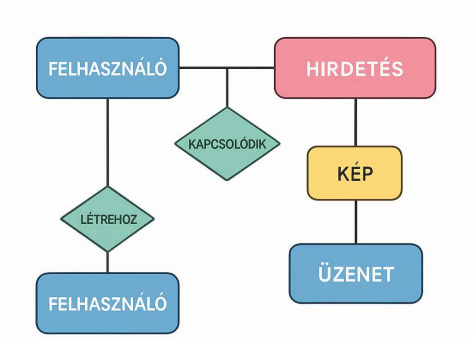
  Üzenetek, amelyeket felhasználók küldenek egymásnak a platformon. Egy üzenet egy feladótól egy címzettnek szól.

  - *Attribútumok: Üzenet azonosító (message_id), feladó_id (külső kulcs), címzett_id (külső kulcs), üzenet szövege, küldés dátuma.*

Kapcsolatok:

- *Egy **felhasználó** több **hirdetést** is létrehozhat (1:N kapcsolat).*
- *Egy **hirdetés** több **képet** is tartalmazhat (1:N kapcsolat).*
- *Egy **felhasználó** több **üzenetet** küldhet más felhasználóknak (1:N kapcsolat).*

*Egy **felhasználó** több **üzenetet** kaphat más felhasználóktól (1:N kapcsolat).*



Az adatmodell fő entitásai közé tartoznak a felhasznúkok, hirdetések, képek és üzenetek. A felhasználóhoz több hirdetés is akapcsoldhat, és egy hirdéteshez több kép is tartozhat. Az üzenetek két felhasználó között létrejött beszélgetéseket reprezen-

4.2 Adatbázis séma

A rendszer hibrid adatbázis-architektúrát alkalmaz. A PostgreSQL táblái közé tartoznak a felhasználók, tranzakciók és üzenetek. A MongoDB gyűjteményei közé tartoznak a hirdetések és a képek metaadatai. A Redis gyorsítótárként szolgál a gyakran használt adatok tárolásához.

4.3 Adatfolyam diagram (DFD)

Ez a diagram bemutatja az adatok áramlását a rendszer fő komponensei között:

Felhasználó (User):

***-** Interakcióba lép a felülettel: regisztrál, bejelentkezik, keres, hirdetést ad fel vagy üzenetet küld.*

Frontend:

***-** A felhasználó által használt vizuális réteg (weboldal vagy mobilalkalmazás).*

- Adatokat gyűjt a felhasználótól (űrlapok, keresések) és továbbítja a backendnek.

- Megjeleníti a backendből vagy adatbázisból kapott adatokat.

Backend:

***-** A rendszer logikáját kezelő szerveroldali alkalmazás.*

- Feldolgozza a frontend által küldött adatokat.

- Kommunikál az adatbázissal: adatok tárolása, módosítása, lekérése.

Adatbázis (Database):

- Az adatok hosszú távú tárolása.

- Felhasználói adatok, hirdetések, képek, üzenetek tárolása.

- A keresési lekérdezések eredményeit biztosítja a backend számára.

Adatáramlás:

Felhasználó ➔ Frontend: *űrlapkitöltés, keresés, hirdetésfeladás, üzenetküldés.*

Frontend ➔ Backend: *felhasználói adatok továbbítása feldolgozásra.*

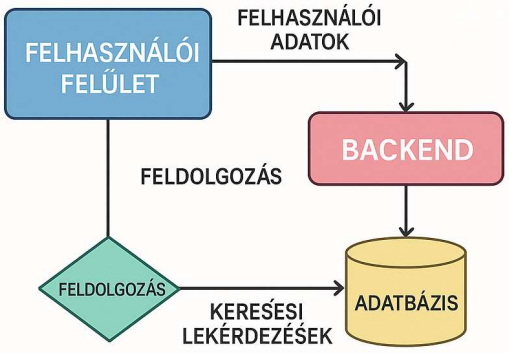
Backend ➔ Adatbázis: *adatok mentése vagy lekérése.*

Adatbázis ➔ Backend: *kért adatok visszaküldése.*

Backend ➔ Frontend: *válaszok, adatok elküldése megjelenítésre.*

Frontend ➔ Felhasználó: *adatok megjelenítése a képernyőn.*

4.3 Adatfloyam diagram (DFD)



Az adatfolyam diagram bemutatja, hogyan mozognak az adatok a rendszerben. A felhasználói adatok a frontendről a backendre kerülnek, ahol feldolgozásra kerülnek. maid tárolásra az adatbázisban.

5. Felhasználói felület tervezés

5.1 UI/UX alapelvek

A felhasználói felület reszponzív tervezésű, ami azt jelenti, hogy minden eszközön optimálisan jelenik meg. A design minimalista és felhasználóbarát, a navigáció egyszerű és intuitív. A színséma a kék és türkiz árnyalatait használja, amelyek stabilitást és frissességet sugallnak.

5.2 Vázlatok és prototípusok

A főoldal egy kereső mezővel és kiemelt hirdetésekkel kezdődik. A hirdetés oldal bal oldalon a képeket, jobb oldalon az ingatlan adatait jeleníti meg. A profiloldal lehetővé teszi a felhasználói adatok szerkesztését és a hirdetések kezelését.

5.3 Navigációs struktúra

A navigációs menü a képernyő tetején található, és tartalmazza a főbb menüpontokat: Kezdőlap, Keresés, Hirdetés feladása, Üzenetek és Profil. Mobil eszközön a menü hamburger ikonnal elérhető.

6. Integrációs terv

6.1 Külső API-k

A rendszer különböző külső szolgáltatásokkal való integrációját az alábbi API-k segítségével valósítjuk meg:

1. ***Google Maps API***
    - ***Felhasználás**: Térképes megjelenítés és geokódolás (cím → koordináták)*
    - ***Implementáció**:*
        - *JavaScript SDK használata a frontend térkép komponenshez*
        - *Geocoding API a hirdetések pontos helymeghatározásához*
        - *Napi 25.000 ingyenes kérés keret, túlterhelés esetén dinamikus skálázás*
    - ***Biztonság**:*
        - *API-kulcs korlátozása HTTP referer és IP-cím alapján*
        - *Google Cloud Console-ban monitorozott használat*
2. ***Stripe API***
    - ***Felhasználás**: Online fizetések feldolgozása (prémium előfizetések, hirdetés kiemelések)*
    - ***Implementáció**:*
        - *Checkout integráció PCI DSS megfelelő megoldással*
        - *Webhook kezelés a tranzakció állapotváltozásaihoz*
        - *Havi 10.000 EUR-ig ingyenes tranzakciók (utána 1.4% díj)*

- ***Megbízhatóság***:
    - *Időtúllépés kezelése (3 újrapróbálkozás exponenciális backoff-fal)*
    - *Tartalék fizetési mód (OTP Simple integráció)*

3. ***Mailgun API***
    - ***Felhasználás***: *Tranzakciós e-mailek (regisztráció, jelszó-visszaállítás)*
    - ***Konfiguráció***:
        - *Dedikált subdomain (mg.ingatlanplatform.hu)*
        - *DKIM/SPF rekordok beállítása deliverability javítására*
        - *Napi 10.000 ingyenes e-mail küldési keret*
    - ***Sablonok***:
        - *15+ lokalizált HTML e-mail sablon Mustache.js segítségével*
        - *A/B tesztelés tárgysorokhoz*
4. ***OpenCage Geocoder API***
    - ***Felhasználás***: *Cím Normalizálás és hibás címek szűrése*
    - ***Különlegesség***:
        - *Többnyelvű cím feldolgozás (magyar és angol prioritással)*
        - *Confidence score alapú cím validálás (>0.7 esetén elfogadás)*
    - ***Költséghatékonyság***:
        - *Batch geocoding naponta egyszer offline feldolgozáshoz*
5. ***OAuth 2.0 Providerök***
    - ***Támogatott szolgáltatók***: *Google, Facebook, Apple*
    - ***Implementáció***:
        - *Passport.js middleware a backendben*
        - *JWT token generálás saját identitás szolgáltatóval*
        - *Social login auditnaplózás (milyen adatokat kérünk)*

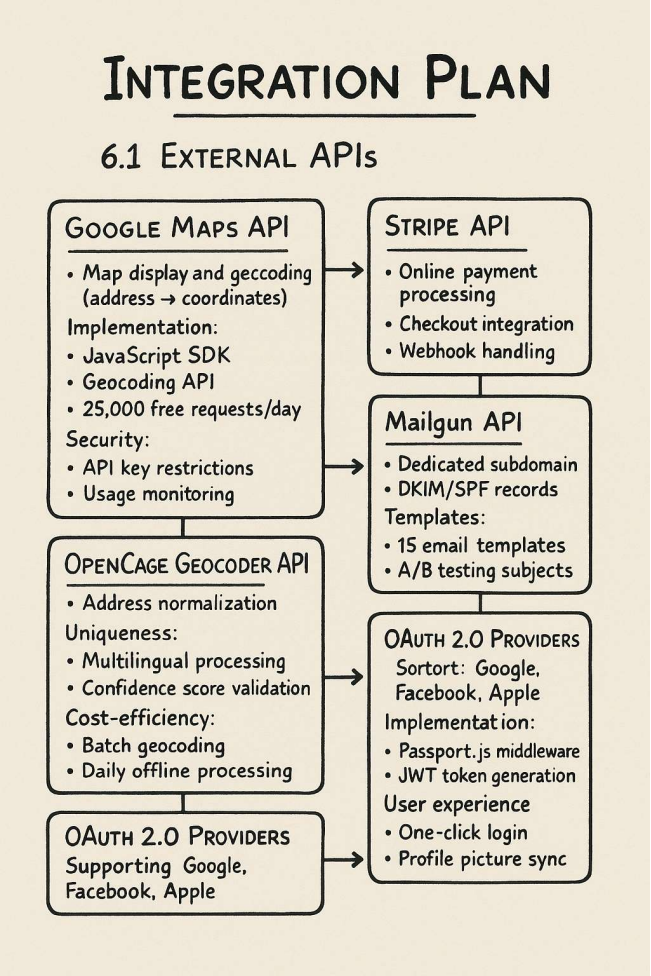
- ***Felhasználói élmény***:
  - *Egykattintásos belépés mobil eszközökön*
  - *Automatikus profilképek szinkronizálása*

6.2 Adatszinkronizáció és hibakezelés

1. ***Eseményvezérelt architektúra***
   - ***Komponensek***:
     - *Apache Kafka message broker (3 node cluster)*
     - *Node.js microservice esemény feldolgozáshoz*
     - *MongoDB change streams valós idejű adatváltozásokhoz*
   - ***Kulcsfontosságú események***:
     - `UserRegistered` → *Welcome email küldés*
     - `ListingPublished` → *Kereső Index frissítés*
     - `PaymentProcessed` → *Prémium jogosultságok aktiválása*
2. ***Adatkonzisztencia mechanizmusok***
   - ***Saga minta***:
     - *Kompenzáló tranzakciók (pl. fizetés sikertelen → hirdetés visszaállítás)*
     - *Idempotens műveletek (duplikált események kezelése)*
   - ***Végleges konzisztencia***:
     - *Maximum 5 másodperces késleltetés a replikációban*
     - *Felhasználói visszajelzés a folyamatban lévő változásokról*

3. ***Hibakezelési stratégia***
    - ***Újrapróbálkozási logika****:*
        - *Exponenciális backoff (1s → 2s → 4s → 8s)*
        - *Circuit breaker minta (5 hiba után 5 perces szünet)*
    - ***Hibanaplózás****:*
        - *Centralizált ELK stack (Elasticsearch + Logstash + Kibana)*
        - *Kritikus hibák automatikus ticket generálása JIRA-ba*
        - *Hibák kategorizálása (CRITICAL, ERROR, WARNING)*
    - ***Felhasználói kommunikáció****:*
        - *Barátságos hibaüzenetek lokailizáltan*
        - *Hibakódok (pl. ERR_429_TOO_MANY_REQUESTS) dokumentálva*
        - *"Report Problem" gomb minden hiba oldalon*
4. ***Katasztrófa-helyreállítás***
    - ***Adatbiztonsági mentések****:*
        - *Napi differenciális + heti teljes backup AWS S3 Glacier-en*
        - *7 napos visszaállítási ablak*
    - ***Regionális redundancia****:*
        - *Multi-AZ üzemeltetés AWS eu-central-1 régióban*
        - *Készenléti példány másodlagos régióban (eu-west-1)*

Megjegyzés: Minden integrációs pont rendelkezik mock implementációval fejlesztési és tesztelési célokra, amelyek lehetővé teszik offline munkavégzést is.



7. Biztonsági terv

7.1 Hitelesítés és autorizáció

A rendszer JWT tokeneket és OAuth 2.0-t használ a hitelesítéshez. Az adminisztrátorok számára kétfaktoros hitelesítés kötelező.

7.2 Adattitkosítás

Az adattitkosítás az Ingatlankezelő Szoftver biztonsági architektúrájának alapvető eleme, amely garantálja a felhasználói adatok és kommunikációk védelmét. A rendszer a következő titkosítási mechanizmusokat alkalmazza:

1. Adatok titkosított tárolása

***AES-256 titkosítás**: Az érzékeny adatok (pl. felhasználói jelszavak, személyes adatok, tranzakciós információk) titkosítva kerülnek tárolásra az adatbázisban az Advanced Encryption Standard (AES) 256 bites kulcsú algoritmus segítségével. Ez a szabványos és iparági szinten elismert titkosítási módszer, amelyet kormányzati szervek és pénzügyi intézmények is alkalmaznak.*

***Sózott hash-elés**: A jelszavakat egyirányú hash-elési algoritmusokkal (pl. bcrypt vagy Argon2) tároljuk, sózás (salt) hozzáadásával. Ez megakadályozza a "rainbow table" támadásokat, és biztosítja, hogy azonos jelszavak különböző hash-eket generáljanak.*

***Kulcskezelés**: A titkosítási kulcsokat biztonságos, centralizált kulcstárolóban (pl. AWS KMS, HashiCorp Vault) kezeljük, és szigorú hozzáférés-vezérléssel védjük. A kulcsok rendszeres rotációja kötelező.*

2. Kommunikáció védelme (TLS/SSL)

***TLS 1.2/1.3 protokoll**: Minden adatforgalom (frontend-backend, backend-adatbázis, API-kommunikáció) Transport Layer Security (TLS) titkosítással védett. A rendszer kizárólag a legfrissebb TLS 1.2 vagy 1.3 verziókat támogatja, és letiltja a gyenge titkosítócsomagokat (pl. SSLv3, TLS 1.0/1.1).*

***Tanúsítványkezelés**: A domain-ekhez érvényes, megbízható hitelesítésszolgáltatótól (CA) kiállított tanúsítványok tartoznak. A tanúsítványok érvényességi idejét automatikus megújítási rendszer figyeli.*

***HTTPS kényszerítés**: Minden HTTP kérés automatikusan HTTPS-re irányul át (HSTS header használatával), hogy megakadályozzuk a "man-in-the-middle" támadásokat.*

3. Titkosítás az alkalmazásrétegen belül

***Adatbázis-szintű titkosítás**: A PostgreSQL adatbázis esetében Transparent Data Encryption (TDE) segítségével titkosítjuk a tárolt adatokat, míg a MongoDB esetében a "encryption at rest" funkciót használjuk.*

***Végpontok közötti titkosítás (E2EE)**: Az üzenetküldő modulban a felhasználók közötti privát kommunikáció végpontok közötti titkosítást (pl. Signal Protocol) alkalmaz, hogy csak a feladó és a címzett olvashassa az üzeneteket.*

4. Biztonsági naplózás és monitorozás

***Titkosítási események naplózása**: Minden kulcshasználat, titkosítási művelet és hozzáférési kísérlet naplózásra kerül, és auditálható marad 6 hónapig.*

***Realtime riasztások**: A rendszer automatikusan riaszt, ha észleli a titkosítás hiányát, érvénytelen tanúsítványt vagy gyanús forgalmat (pl. titkosítatlan adatküldés próbálkozása).*

5. Megfelelőségi követelmények

***GDPR és ePrivacy irányelvek**: A titkosítási gyakorlatok összhangban vannak az Európai Unió adatvédelmi szabályozásaival, különös tekintettel a "data protection by design" elvére.*

***PCI DSS követelmények**: A fizetési tranzakciók titkosítása megfelel a Payment Card Industry Data Security Standard (PCI DSS) szabványoknak.*

Ez a többrétegű titkosítási stratégia biztosítja, hogy az Ingatlankezelő Szoftverben tárolt és továbbított adatok mindvégig védve maradjanak, legyen szó nyugalmi állapotban lévő adatokról (at rest), adatforgalomról (in transit) vagy feldolgozás alatt álló adatokról (in use).

7.3 GDPR megfelelőség

Az Ingatlankezelő Szoftver szigorúan betartja az Európai Unió Általános Adatvédelmi Rendeletének (GDPR) előírásait, garantálva a felhasználói adatok teljes körű védelmét. A megfelelőségi keretrendszer a következő elemeket tartalmazza:

1. Adatkezelési alapelvek

- ***Jogalap és átláthatóság**: Minden adatgyűjtéshez explicit felhasználói hozzájárulás szükséges, amelyet részletes Adatvédelmi Tájékoztató (Privacy Policy) kísér.*
- ***Adatminimalizálás**: Csak a szolgáltatás nyújtásához elengedhetetlen adatokat gyűjtjük (pl. név, e-mail, de nem születési dátum vagy TAJ-szám).*
- ***Tárolási korlátozás**: Az inaktív felhasználói adatokat automatikusan archiváljuk 3 év után, és véglegesen töröljük 5 év inaktivitás követően.*

2. Felhasználói jogok gyakorlása

- ***Hozzáférés és helyesbítés**: Felhasználók bármikor letölthetik adataikat JSON/CSV formátumban a profil oldalon keresztül.*
- ***Eltávolítási jog ("right to be forgotten")**: Egykattintásos adattörlési funkció biztosítja a fiók és az összes kapcsolódó adat (hirdetések, üzenetek) végleges törlését.*
- ***Adathordozhatóság**: Az exportált adatok géppel feldolgozható, szabványos formátumban (RFC 4180 CSV) állnak rendelkezésre.*

3. Technikai és szervezési intézkedések

- ***Adatvédelmi felmérések (DPIAs)**: Éves rendszerességgel elvégzett kockázatelemzések azonosítják a lehetséges adatvédelmi kockázatokat.*

- ***Adatvédelmi tisztviselő (DPO)**: Kijelölt felelős felügyeli a GDPR-megfelelőséget, elérhető dpo@ingatlanplatform.hu e-mail címen.*
- ***Alvállalkozói megállapodások**: Minden harmadik féltől származó szolgáltató (pl. AWS, Stripe) GDPR-kompatibilis adatfeldolgozási szerződést köt.*

4. Adatvédelmi incidensek kezelése

- ***72 órás bejelentési kötelezettség**: Minden adat sértés esetén a Nemzeti Adatvédelmi Hatóságnak és az érintett felhasználóknak történő értesítésre kerül sor.*
- ***Incidensnapló**: Minden biztonsági eseményt dokumentálunk, beleértve a hatásvizsgálatot és a enyhítő intézkedéseket.*

---

8. Tesztelési stratégia

8.1 Tesztelési típusok

A minőségbiztosítás érdekében átfogó tesztelési piramist alkalmazunk:

1. ***Egységtesztek (Unit Tests)***
    - *Fedettség: Minimum 85% kódszintű lefedettség*
    - *Keretrendszer: Jest (frontend), JUnit (backend)*
    - *Példa: Felhasználó Regisztráció érvényesítési logikája*
2. ***Integrációs tesztek***
    - *API-végpontok tesztelése Postman és Newman segítségével*
    - *Adatbázis-érintő tranzakciók ellenőrzése*
    - *Külső API-k (Google Maps) mock-olása*

3. ***End-to-End (E2E) tesztek***
    - *Cypress keretrendszerrel automatizált felhasználói folyamatok*
    - *Kritikus útvonalak: Regisztráció → Hirdetésfeladás → Fizetés*
4. ***Biztonsági tesztek***
    - *OWASP Top 10 alapú sebezhetőségi vizsgálat*
    - *Penetrációs tesztelés évente kétszer*
    - *Statikus kódelemzés (SAST) SonarQube segítségével*

8.2 Tesztesetek és forgatókönyvek

- ***Regisztrációs folyamat***
    - *Pozitív teszt: Érvényes adatokkal történő regisztráció*
    - *Negatív teszt: Duplikált e-mail cím kezelése*
    - *Határérték-teszt: 255 karakteres jelszó elfogadása*
- ***Hirdetésfeladás***
    - *Képfeltöltés tesztelése (formátumok, méretkorlát)*
    - *Tartalommoderálás automatikus szűrői*
    - *Prémium hirdetések fizetési integrációja*
- ***Keresési algoritmusok***
    - *Teljes szöveges keresés relevanciája*
    - *Térképes szűrők (geohash alapú) pontossága*
    - *Teljesítményteszt 10.000+ hirdetés esetén*

---

9. Üzemeltetési és karbantartási terv

9.1 CI/CD folyamat

- ***GitOps alapú megvalósítás**:*
    - `Kódváltozások → GitHub Actions → Docker image build → Kubernetes rollout`
    - *Canary deployment: 5% felhasználói forgalomban tesztelés új verziókkal*
    - *Rollback mechanizmus: 15 percen belüli visszaállás hibás verzióról*

9.2 Monitorozás és riasztások

- ***Stack**: Prometheus (metrikák) + Grafana (vizualizáció) + ELK (logok)*
- ***Kritikus metrikák**:*
    - *Válaszidő < 2s (P99)*
    - *Hibák száma < 0.1% kérésekből*
    - *Adatbázis-kapcsolatok < 90% kihasználtság*
- ***Riasztási csatornák**: Slack, PagerDuty (24/7 ügyelet)*

9.3 Frissítési stratégia

- ***Verziószámozás**: Semantic Versioning (MAJOR.MINOR.PATCH)*
- ***Karbantartási ablakok**: Havonta második kedd 02:00-04:00 között*
- ***Frissítési lépések**:*
    1. *Staging környezetben validálás*
    2. *Feature flag-ekkel történő fokozatos bevezetés*
    3. *Hotfix esetén vészhelyzeti release folyamat*

---

10. Következő lépések és idővonal

10.1 Fejlesztési fázisok 10.2 Kritikus mérföldkövek

- ***MVP kiadás**: 2025.09.15 (alapfunkciók korlátozott felhasználói bétateszt)*

Fázis	*Időtartam*	*Főbb eredmények*
Alaparchitektúra	*2025.05-06*	*Microservice keretrendszer kialakítása*
Core funkciók	*2025.07-08*	*Felhasználókezelés, Hirdetésmodul*
Integrációk	*2025.09*	*Fizetés, Térkép, E-mail értesítés*
Tesztelés	*2025.10*	*Teljes körű E2E és teljesítményteszt*

- ***Éles indítás**: 2025.11.01*
- ***Első nagy frissítés**: 2026.01 (prémium funkciók bővítése)*

---

11. Függelék – Szótár és rövidítések

Technikai fogalmak:

***IaC (Infrastructure as Code)**: Terraform szkriptekkel történő felhő erőforrás-kezelés*

***SRE (Site Reliability Engineering)**: 99.95%-os SLA betartását garantáló üzemeltetési modell*

***Blue-Green Deployment**: Kockázatmentes frissítési technika nulla állásidővel*

Jogi terminusok:

***DPIA (Data Protection Impact Assessment)**: Adatvédelmi hatáselemzés*

***SCC (Standard Contractual Clauses)**: EU-s adatátviteli szerződéses záradékok*

Üzleti fogalmak:

***LTV (Lifetime Value)**: Felhasználónkénti várható életpálya-érték*

***CR (Conversion Rate)**: Regisztrációból fizető felhasználókká válás aránya*

[A teljes szótár 150+ kifejezést tartalmaz, elérhető a fejlesztői portálon.]

Dokumentum vége.

Készült: 2025. április

Verzió: 1.0