

Önálló laboratórium 2,
Diplomatervezés 1

Belső foglalási rendszer fejlesztése

Bajmóczy Eszter

Tartalom

Téma leírása	3
Specifikáció:.....	3
Funkcionalitás áttekintés	4
Személyek értékelése.....	5
Virtuális kreditrendszer.....	6
Felépítés	6
Menük	6
Home.....	6
Hirdetések kezelése	6
Foglalások.....	7
Üzenetek	7
Saját profil	7
Képek tárolása	7
Megvalósítás.....	7
Authentikáció	8
Adattárolás	8
Redis adatbázis	9
Felépítés	9
Tervek, hasznos linkek	9
Frontend.....	10
Architecture.....	10
UI tervek.....	10
Geocoding	12
Reactive forms	12
Validáció	12
Token.....	12
Calendar view	13
Style-ok kezelése:.....	13

Téma leírása

A hallgató feladata egy full-stack webalkalmazás fejlesztése ASP.NET Core alapokon Angular vagy React technológiával. Az alkalmazás felületet biztosít közös tulajdonban lévő ingatlanok, tárgyak használatának menedzselésére, ezáltal lehetőség van foglalások kezelésére, költségek és aktuális témák számontartására, megvitatására.

A feladat részét képezi a hallgató számára megfelelő frontend technológia kiválasztása és a választáshoz szükséges ismeretek felvétele. Az alkalmazásnak esztétikusnak és könnyen kezelhetőnek kell lennie.

A téma kiírás után a témán több változtatás mellett döntöttünk a konzulensemmel, ezért a végén egy sharing platform-ot terveztem, mellyel a különböző tárgyak kölcsönzését és hirdetését, de akár szolgáltatások hirdetését is lehetővé teszi.

Specifikáció:

Manapság a háztartásokban több olyan eszköz is jelen van, mely csupán évente pár alkalommal kerül használatba. Az ilyen eszközökre, gépekre többen ezen okból sem szeretnének beruházni, annak ellenére, hogy szükségük lenne rá. A probléma megoldására egy kölcsönadásokat és -vételeket menedzselő alkalmazás megfelelő lenne, melyen magánszemélyek bonyolíthatják le az egész folyamatot.

Az alkalmazás célja a tárgyak közös használatának megkönnyítése és annak menedzselésének segítése, továbbá lehetővé teszi szolgáltatások kiajánlását és azok igénybevételét. A különböző tárgyakat, szolgáltatásokat a hirdetésnek megfelelően foglalni lehet, illetve lehetőség van zártkörű csoportokban is hirdetni, mely által többek között egy közösen használt ingatlan foglalását is lebonyolíthatja egy szűk körű társaság.

Mindemellett az alkalmazás használatával megbízható, együttműködő közösségek jöhetnek létre, melyben spórolhat is az ember. Az adott felhasználók értékelésével eldönthető, hogy mennyire megbízható valaki, illetve a tárgyak, szolgáltatások is értékelhetőek, ezáltal jelezhető, ha valami nem olyan minőségű, mint ahogyan azt hirdetik.

Vegyünk néhány példát:

Viszonylag nagy kerttel rendelkezem, és mivel anyagilag megengedhetem magamnak beruháztam egy fűnyíró traktorra, a gépet viszont csak 2-3 hetente használok 1-2 órára, a többi időben pedig a garázsban áll. Számomra megérné pénzügyileg néha kölcsönadni a gépet, míg a másik személynek esetleg

anyagilag jobban megérné az enyémet kölcsönözni, mint beruházni egy újra, illetve neki addig sem kell tárolnia azt.

Egy család a nagypapa révén birtokol egy nyaralót, melyet a család különböző részei különböző időpontokban szeretnének igénybe venni. Az alkalmazás segítségével egy naptár nézetben követhető a nyaraló foglaltsága, illetve könnyen beosztható, hogy ki mikor veszi azt igénybe.

Egy fiatal a nyári szünetben alkalmanként fűnyírással szeretne kis bevételhez jutni, ehhez hasonló hirdetéseket is közzé lehet tenni a weboldalon, melyre a többi felhasználó lecsaphat, és az elérhető időpontok közül könnyen kiválaszthatja a neki megfelelőt.

Funkcionalitás áttekintés

- Bejelentkezés / Kijelentkezés
- Saját profil nézet / felhasználó nézet
 - o email értesítések bekapcsolása
 - o adatok módosítása
 - o értékelések megtekintése / készítése
 - o hirdetések megtekintése
- Felhasználók értékelésének lehetősége, ezek megtekintése
- Az adott property értékelése
- Property /service létrehozása / frissítése / megtekintés / törlése
 - o cím
 - o leírás
 - o helyszín (nem lesz nyilvános a hirdetésen)
 - o használati utasítás
 - o ár / kredit
 - o elérhetőség (naptár)
 - o láthatóság (public, closed)
 - public: minden felhasználó számára látható
 - closed: csak bizonyos felhasználók számára látható
 - o emberek meghívása / értesítése email alapján
 - o jelentkezési lehetőség
 - o üzenet a tulajnak (ha van, korábbi beszélgetés megnyitása)

- Property/service-ek listázása
 - o keresés
 - o helyszín alapján (x km-es körzet)
- Üzenetek nézet
 - o link a hirdetésre
- Foglalások kezelése nézet (saját hirdetés foglalásainak kezelése és mások hirdetésének foglalásai)
 - o lemondás, módosítás, üzenet
 - o értékelés:
 - hirdető
 - property/service
 - kölcsönző

Személyek értékelése

Mind a kölcsönzőket, mind a kölcsönadókat lehet értékelni, mely később segítséget nyújthat a többi felhasználónak. Az értékelések különböző szempontok 5-ös skálán való értékelését, majd ezekből annak a kiszámítását, mely által az adott felhasználó besorolható a következő csoportokba: pozitív, semleges, negatív.

Az értékelések a felhasználó alatt jelennek meg egyszerűsített formában (pozitív, semleges, negatív) és százalékos értékkel, az értékelések számával együtt, illetve az adott profil megnyitásával a részletes skálák is megtekinthetők.

Az értékelések ketté vannak szedve: hirdető és kölcsönző értékelésekre.

5-ös skálán a következő értékek:

- rugalmasság
- megbízhatóság
- a készüléket épen adta oda / vissza

Az értékelések eredményének kiszámítása súlyozott átlagszámítás:

- a skáláknak a súlya:
 - o rugalmasság: 1
 - o megbízhatóság: 3
 - o a készüléket épen adta át: 3

A jelenleg megvalósított súlyozott átlag számítással készülő értékelést a jövőben bővíteni tervezem, Bayesi frissítéssel, elévülő értékelésekkel. A strategy pattern használatával könnyen bővíthető a jelenlegi megvalósítás.

Csak egy-egy kölcsön lebonyolítása után lehessen értékelni a másik felet, ezzel megakadályozva, hogy az értékelési rendszerrel valaki visszaéljen és tapasztalat nélkül értékeljen valakit.

Virtuális kreditrendszer

A virtuális kreditet nem feltétlenül mindenki tudná megfelelően kihasználni, ezért a rendszer nem korlátozza a kölcsönzés során ezek használatát, lehetőség van pénz, illetve kredit fizetési eszközt is megadni egy-egy hirdetésben.

Egy felhasználó úgy juthat kredithez, hogy kölcsön ad valamilyen eszközt az alkalmazáson keresztül, ezáltal annak értékének X%-ának megfelelő kreditet kap.

Egy kölcsön lebonyolítása során juthat egy felhasználó új kreditekhez, mégpedig annak árának 10%-nak megegyező mennyiségűhöz (amennyiben elfogadható értékelést kapott rá). Továbbá a felhasználók értékelését is támogatjuk kreditekkel, ezzel ösztönözve a felhasználókat a visszajelzések adására.

Felépítés

Menük

- Saját profil
- Üzenetek
- Hirdetések kezelése (sajátok kezelése, új hozzáadása) / Mentett elemek (lehessen hirdetést elmenteni)
- Foglalások
- Home

Home

Ebben a menüpontban lehetőség van a hirdetések megtekintésére és azok böngészésére. A keresést a különböző szűrők is segítik, melyekkel beállíthatjuk, hogy kölcsönözhető tárgyat vagy szolgáltatást szeretnénk keresni (esetleg mindkettőt), illetve a fizetőeszköz fajtáját (kredit/készpénz). Továbbá egy település és annak vonzás körzete is megadható az eredmények megfelelő szűrésének az érdekében. A rendezésre a következő lehetőségek vannak: az adott településtől való távolság szerint növekvő, ár szerint növekvő, illetve ár szerint csökkenő. Mindemellett szabadszavas keresésre is van lehetőség.

Egy hirdetésre rákattintva megjelenik annak adatlapja a következő adatokkal: cím, leírás, használati utasítás, képek, település, elérhetőség, kredit és készpénz ár. Illetve lehet foglalást leadni, üzenetet írni a tulajdonosnak, elmenteni a hirdetést.

Hirdetések kezelése

A home oldalhoz hasonlóan jelennek meg felül a felhasználó saját hirdetései, majd lejjebb az általa elmentett hirdetések. A saját hirdetéseket lehet deaktiválni/aktiválni, megtekinteni, módosítani, illetve a foglalásokat megtekinteni és azokat elfogadni vagy elutasítani. Továbbá lehetőség van az egyes foglalásokhoz tartozó felhasználónak üzenetet küldeni.

Szűrni és rendezni a home oldalhoz hasonlóan lehet. Mindemellett itt lehet új hirdetéseket létrehozni egy gomb segítségével. Amennyiben egy felhasználó foglal egy hirdetésen, az automatikusan a mentett hirdetésekhez kerül.

Foglalások

Naptár nézet, külön színnel, amit a felhasználó foglalt és másikkal, amit hozzá foglaltak. Lehetőség van a naptár nézetben szűrésre a két említett típus között. Kattintásra felhossa a foglaló felhasználó profilját, ahol az értékelések is megtekinthetők.

Foglalást lemondhat, elállhat tőle, értékelheti az elmúltakat.

Üzenetek

Üzeneteket lehet megtekinteni, és keresni köztük (név alapján), minden üzenet tartalmaz egy linket a hirdetésre. Az üzenetek nevéként a másik fél neve és a hirdetése címe jelenik meg. Egy egy üzenetre rákattintva megnyílik a chat, ahol további üzeneteket lehet küldeni.

Saját profil

Saját adatok (név, email, értesítések beállítása) megtekintésének, módosításának lehetősége. Továbbá a róla készült értékelések böngészése.

Képek tárolása

Lehetséges képek: profil kép, hirdetés kép.

Fájlrendszerben fogom őket tárolni az alábbi módon:

Főmappa

- userId
 - o profilkép
 - o hirdetés id
 - hirdetéshez tartozó képek

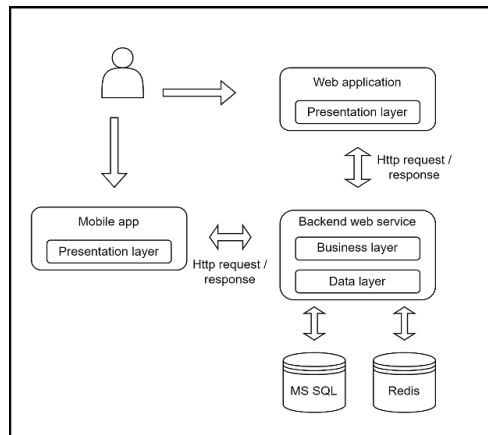
Szerintem korlátozva legyenek a hirdetéshez feltölthető képek száma. 5-nek elegendőnek kellene lennie, max 10.

Megvalósítás

Az alkalmazás backendjét ASP.NET Core-ban készítettem, ugyan a félév során nem teljesen készültem el vele, viszont egy erős, alapjaiban működő rendszert alkottam. Az Api hívások működnek, az értékelés menetén, algoritmusán fogok még változtatni, ennek egy kisebb kutatást szeretnék szentelni, illetve kisebb simítások lesznek még szükségesek.

Frontend-nek az Angular technológiát választottam, melyet majd következő félévben fogok elkezdni megvalósítani.

A tervezett architektúrát a következő ábra mutatja, melyen a mobil kliens csupán egy újabb lehetőség, melyet szeretnék megvalósítani a jövőben amennyiben a webes frontend alkalmazás mellett befér az időmbe.

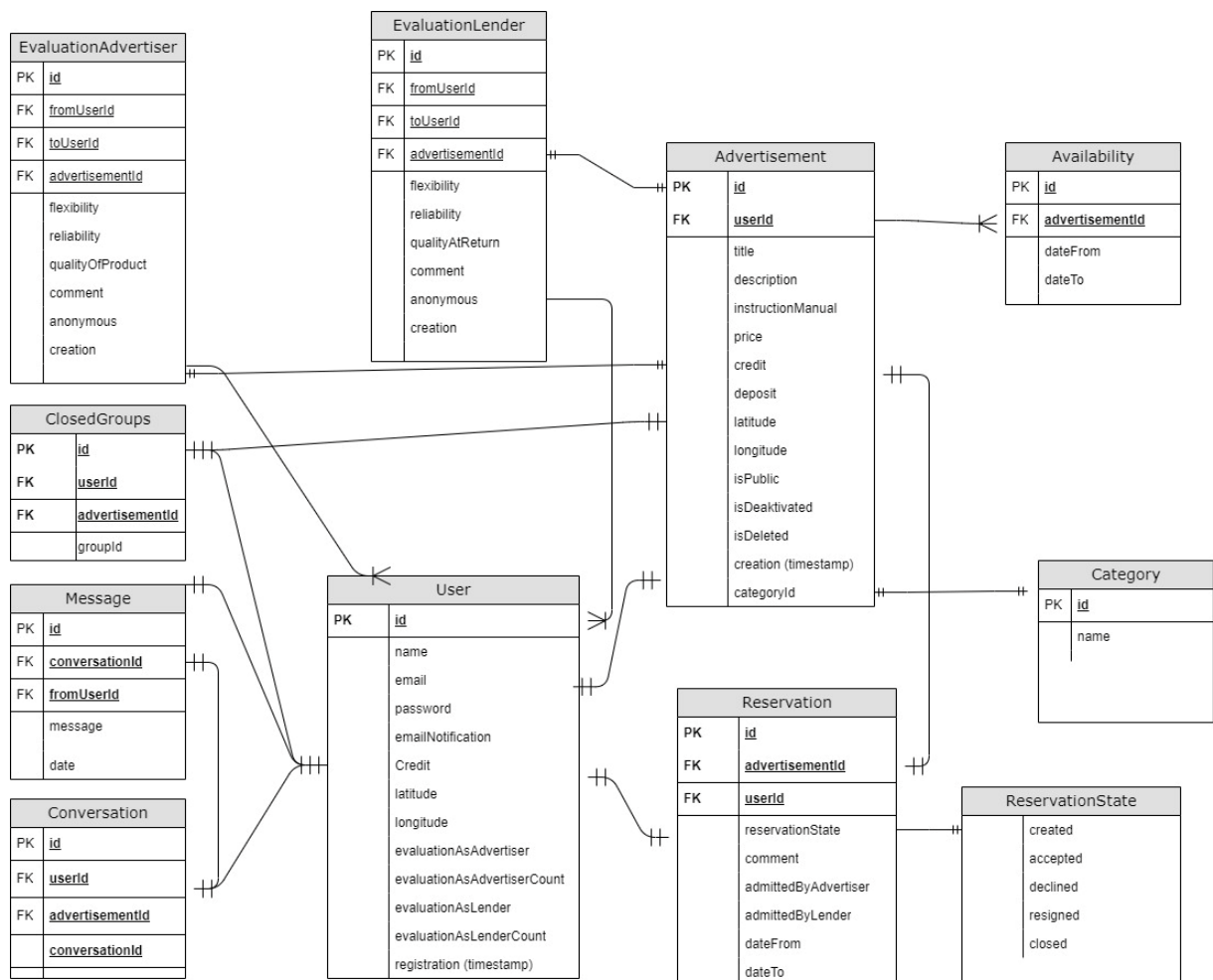


Authentikáció

Az alkalmazásban JWT tokenekkel történik az azonosítás, melyekhez egy refresh token is tartozik.

Adattárolás

A fő adattárolás egy MSSQL adatbázisban történik, az ER diagram az alábbi linken található:



A fejlesztés során az Enity Framework Code First workflow-t használtam az adatbázis kialakítására. Először az osztályokat készítettem el a projektben, majd azokat migrációk segítségével vittem át az adatbázisba.

Redis adatbázis

A felhasználó kijelentkezése esetén, a token-ek blacklist-jének tárolására a Redis adatbázist használok. Egyrészt a Redis gyorsasága miatt választottam, illetve be szerettem volna építeni egy újabb, az egyetemen tanult technológiát a rendszerbe, tudástáramba. Használata könnyű, és a tokenek kezelésére szerintem ideális.

Mindenesetre, az adatbázis hiánya, vagy elérési problémák esetén nem áll le a rendszer, a hibákat logolja a rendszer egy külön txt file-ban.

Felépítés

A projektet igyekeztem jól struktúrálni, modelleket és dto modelleket külön szedtem, melyek között az átalakítást converter osztályok végzik. A convertereket pedig az egyes szervíz osztályok hívják, ezáltal a controllerek csak a dto modellekkel érintkeznek. Mind a service osztályok, mind a converterek rendelkeznek interfésszel. Ezzel a felbontással és dependency injection-nel könnyű lesz az osztályokra Unit teszteket írni.

Tervek, hasznos linkek

Wireframe link:

<https://www.figma.com/file/Q4TTcQRk3jrnpK1b1p4PA/Wireframe?node-id=0%3A1>

SwaggerHub:

<https://app.swaggerhub.com/apis/EszterBajmoczy/Lendship.Backend/1.0.0>

ER diagram:

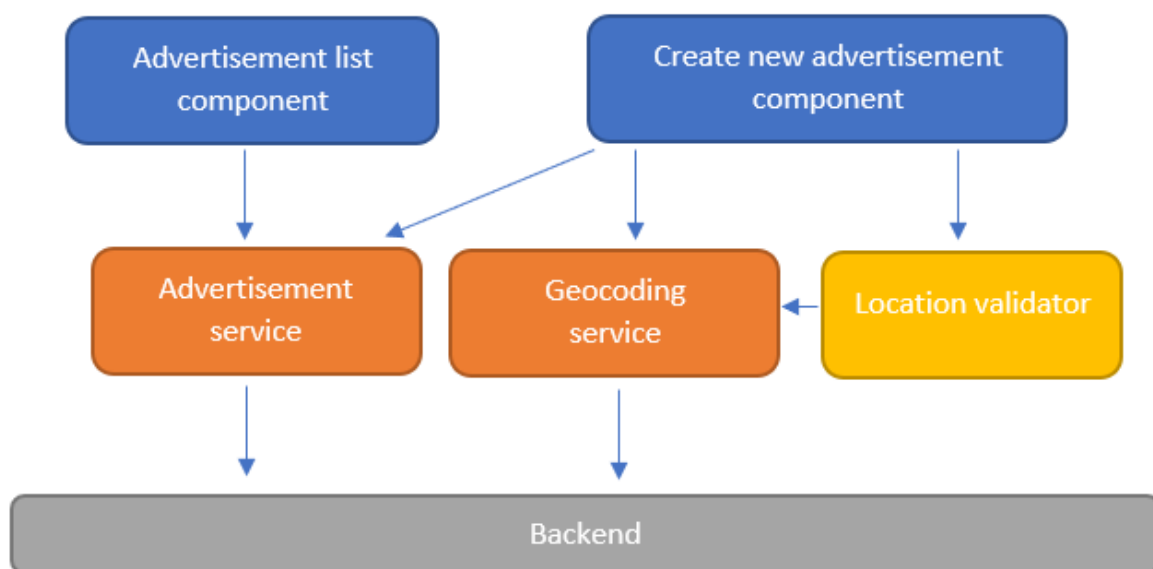
<https://drive.google.com/file/d/1BAisfX0LS9fg8AQi5LGx2N1UEmplwU2M/view?usp=sharing>

Frontend

Az alkalmazás frontend részének egy NodeJS webszervert választottam Angular keretrendszerrel.

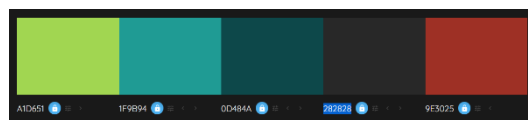
Architecture

Az alkalmazás frontend részében Component architektúrát alkalmaztam, mely által minden nézet saját komponenssel rendelkezik. Az egyes komponensek részei: .html, .js, .scss, .spec.ts file-ok. A komponensek a különböző adatokat service-eken keresztül érik el, mind a backend szolgáltatásról, mind például a geocoding api-tól. Ezeket a service-eket több komponens is használhatja és feladatukról a backend-en használt módon az adott adatcsoport köré épül, mint például a hirdetések készítése, módosítása, lekérdezése. A service-ek mellett több segéd komponenst és osztályt hoztam létre, mint például a különböző felugró ablakoknak, a fájlöltésnek, illetve a form-okhoz szükséges custom validatorokat. Az architektúra egy konkrét kis részlete az alábbi ábrán látszik:

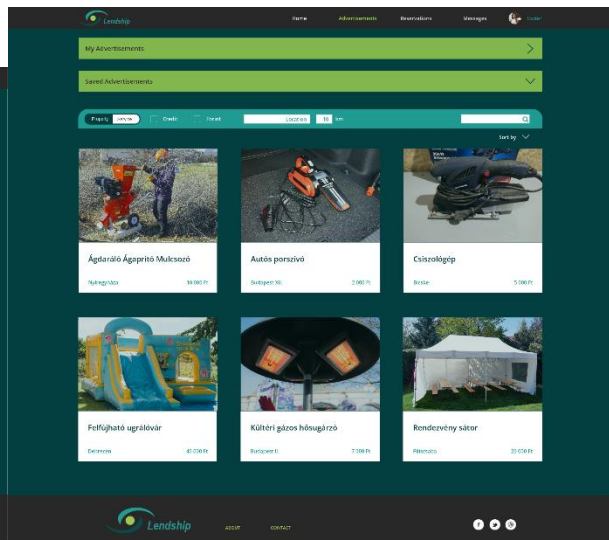
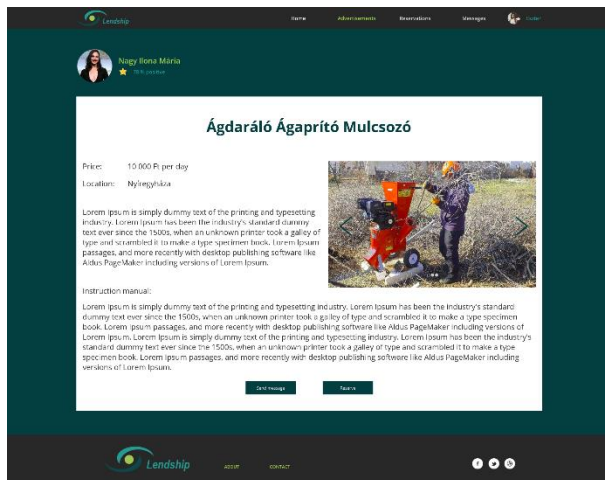
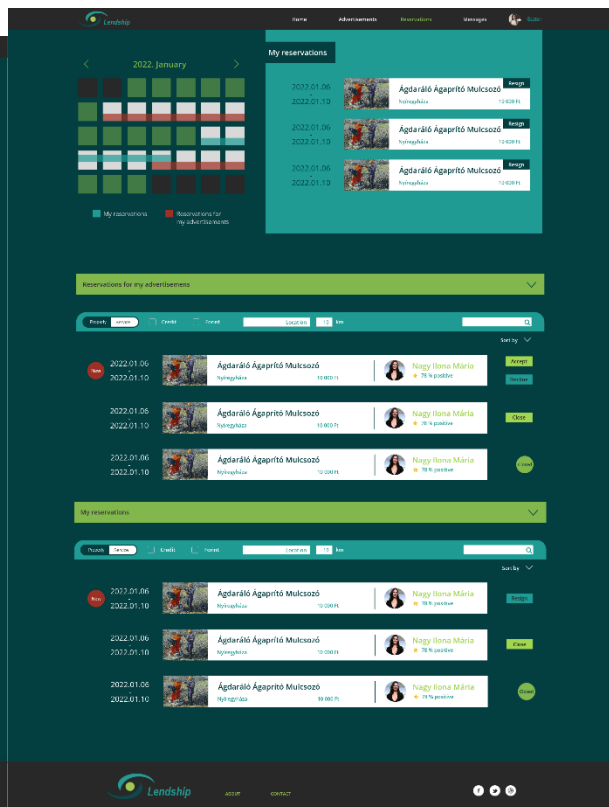
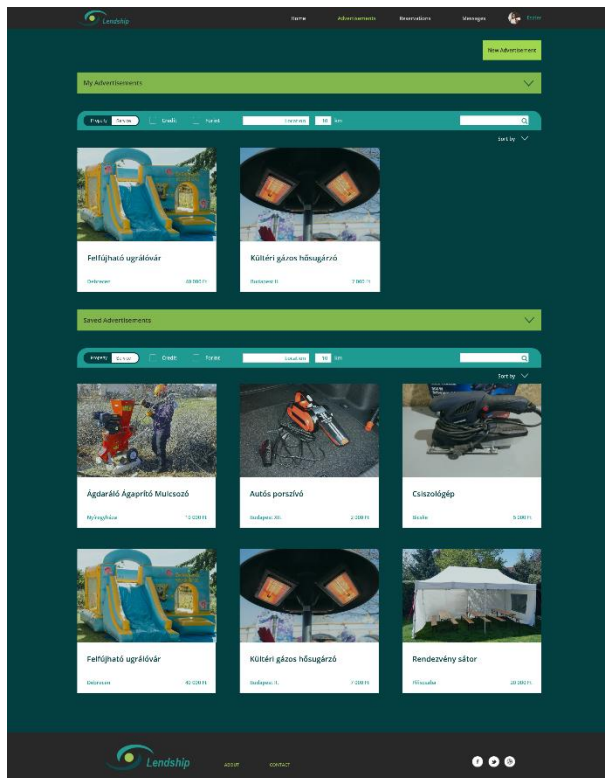


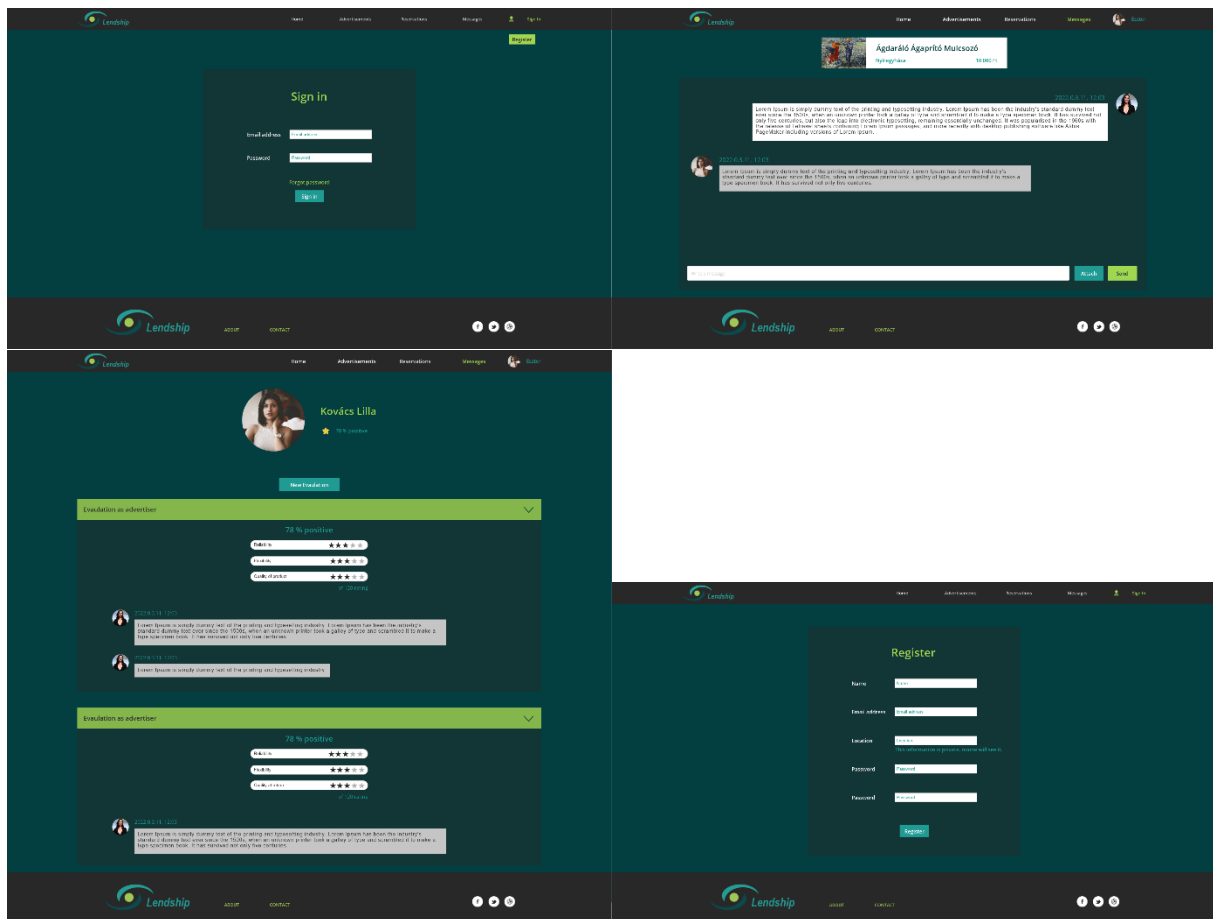
UI tervek

Az alkalmazás kinézetét, felépítését a JustinMind alkalmazással terveztem meg. A tervezés során létrehoztam a logót, és a színpalettát. Egy interaktív, kattintható prototípust is létrehoztam, mely által a nézetek közötti átmenet egyértelműen megfigyelhető.



A főbb nézetek:





Geocoding

Az alkalmazásban több helyen szükség van a címek konvertálására, erre a feladatra a Google Geocoding API-t használom. Viszont fejlesztés során az ingyenes limit betartásához átalakítottam a már meglévő adatstruktúrát, hogy mind a longitude, latitude, mind a város, község nevét eltárolja.

Reactive forms

Az alkalmazásban megjelenő form-oknál az Angular által biztosított reactive forms megközelítést használom. Többek között ennek az előnye a skálázhatóság, robusztusság és újrafelhasználhatóság a template-driven megközelítéssel szemben.

Validáció

A validációk során kettő saját validációt készítettem, egyet a jelszóhoz és egyet a helyszínhez.

A jelszó validációra azért volt szükségem, hogy ellenőrizni tudjam, hogy a jelszó és a jelszó ismétlése mezőben valóban ugyanazt az értéket írta a felhasználó.

A helyszín validációja során a Geocoding API-t használom, a regisztráció, új hirdetés feladásának egyik követelménye, hogy megfelelő helyszínt adjon meg a felhasználó.

Token

Frontenden a localStorage-ban tárolom a felhasználó bearer tokenjeit. A szerverrel való kommunikáció során egy interceptor segítségével adom hozzá minden kéréshez az autentikációs token, illetve szükség esetén a token frissítését is ez a komponens végzi.

Calendar view

A calendar nézetek a ngb-datepicker-rel valósítottam meg. A foglalás és a reservation nézetben is különböző színekkel vannak jelölve a napok az adott jelentésnek megfelelően.

Ezek lehetnek:

- Foglalás esetén:
 - Nem elérhető (a hirdetés azon a napon)
 - Foglalt
- Foglalás nézetben
 - A felhasználó saját foglalásai
 - A felhasználó hirdetésére tett foglalások

Style-ok kezelése:

A főbb stílusleíró osztályokat külön kiszerveztem, mint például a színeket, gombokat, form-okat, kártyákat és a profil különböző megjelenítését. Az egyes componens-ekhez tartozó scss file-okban ezeket beimportálva használom, így könnyen visszakövethetőek a kinézetet befolyásoló értékek.