

Követelményspecifikáció

Szoftverarchitektúrák tantárgy házi feladat

Közösségi kincskereső a belvárosban - Community treasure hunt in downtown

Feladatkiírás:

A hallgatók feladata egy olyan webes oktató/kaland játék elkészítése, amely a játékosokat a város különböző pontjaira küldi bizonyos küldetések elvégzésére.

- A küldetéseket maguk a játékosok állítják össze egymásnak. A küldetéseket lehet értékelni szövegesen és pontozással.
- A küldetések több részfeladatból állnak és lehetőséget adnak több teljesítési módra.
- A játék tartalmazzon gamification mechanizmusokat, melyek célja a játékosok bátorítása, hogy több küldetést teljesítsenek.
- A játék legyen elérhető mobil készülékekről és ellenőrizze a játékos lokációját GPS-el.
- A moderátor szerkeszthet, felfüggeszthet és törölhet küldetéseket, letilthat felhasználókat.

Fejlesztői csapat:

A csapat tagjai:

Név	Neptun	E-mail cím
Csapó Botond Balázs	XM4SQC	csapo.botond000@gmail.com
Gyulai Gergő László	JJY2VO	gyulaigergolaszlo332@gmail.com
Farkas Réka	IXAGK7	zeefarkas@gmail.com
Simon Zoltán	HRSNUG	zoltan.simon@edu.bme.hu

Részletes feladatléírás:

A megvalósítani kívánt projekt egy kalandjáték tervező és játzó alkalmazás fejlesztése, ami a felhasználói közösség számára egy területet biztosít, ahol kedvük szerint állíthatnak össze érdekes és izgalmas küldetéseket, vagy kereshetnek maguknak játékos és szórakoztató elfoglaltságot. A küldetéseket a felhasználók készítik el saját maguknak, amiket utána egyénileg vagy csoportosan tudnak teljesíteni.

A program kétnyelvű lesz, a felhasználók személyes beállításától függ a nyelv, az alap az angol, és kiegészítésként magyar verziót is csináltunk.

A megvalósított applikáció 2 fő funkcióval rendelkezik: lehetséges benne küldetést tervezni, illetve teljesíteni egy már korábban elkészített küldetést.

- A küldetések privátak vagy publikusak lehetnek.
 - A publikus küldetések egy globális listában láthatók az összes felhasználó számára.
 - A privát küldetéseknél a készítő generál egy kódot, amit külső csatornán keresztül eljuttat a célcsoporthoz, akik szeretnének belőle játékot indítani. A kódot ismerő játékosoknál egy külön listában jelennek meg a vele megosztott privát játékok.
- A felhasználók vizuálisan megjelenítve, drag & drop technikával tudják összeállítani/megtervezni a későbbiekben játszható küldetéseket.
- Küldetés részletei:
 - Minden küldetésnek van neve, készítője, leírása, készítés dátuma és különböző feladatokat tartalmaz.
 - Esetlegesen megadható a küldetés típusa.
 - A feladatok sorrendjét a tervező határozza meg.
 - Egy feladat rendelkezik egy címmel, leírással, egy típussal, megoldással és a teljesítésért járó pontszámmal.
 - A feladatok típusa határozza meg a megoldási módot és az ellenőrzését a játékosok válaszainak. A lehetséges típusok:
 - Automatikus ellenőrzéssel rendelkező feladatok
 - Felsorolt lehetőségekből kell kiválasztani a megoldásokat
 - Számmal kell megadni a választ
 - Dátummal kell megadni a választ
 - Térképen kell eljutni egy területre, egy pont bizonyos sugarú környezetébe
 - Egy megadott listát sorba rendezni
 - Emberi ellenőrzést igénylő feladatok, amiknél az adott játék moderátora ellenőrizhet a beküldött megoldást (csak privát játékba lehetséges betenni ilyet):
 - Szöveges választ kell adni
 - Képet kell feltölteni
 - A feladatok ellenőrzése csak akkor történik meg, ha a játékosok az ellenőrzés gombra kattintanak.
- Egy küldetést lehetséges menteni közzététel nélkül, és később folytatni a tervezését.
- A közzétett küldetést nem lehetséges módosítani utólag.
- A saját küldetését törölheti a tervező.
- Elkészült küldetéssel való játék:
 - Egy felhasználó, ha kiválaszt egy játszani kívánt játékot, akkor azt kétféleképpen tudja elindítani:
 - Elindítva egy session-t ő is részt vesz játékosként a játékban.
 - Moderátorként indítva egy küldetés (akár publikus akár privátot), akkor nem vesz részt a játékban, csak a többi játékos haladást kísérheti figyelemmel, illetve, ha a küldetés tartalmaz moderátori ellenőrzést igénylő feladatot, akkor tudja értékelni a feladatot. Ezt akkor is megteheti, ha nem ő a küldetés tervezője. Privát küldetésnél meg kellett kapnia korábban a küldetés tervezőjétől a küldetés kódját.
 - Mindkét esetben a session indításához egy session kód generálódik. Ezt a kódot egy külső csatornán keresztül eljuttatva és egy “Játékhoz csatlakozás” opcióban megadva további játékosnak lehetséges csatlakozni az elindított session-höz játékosként.

- A session végén az aktuális eredmények olvashatók a játékosok és eseteleg a moderátor számára.
- Publikus játéknál a session eredményét publikussá lehet tenni, amit onnantól globálisan mindenki láthat.
- Egy elindított session félbehagyható.
- A játékosnak az elindított és befejezett session-jei egy listában megtekinthetők.
 - Egy félbehagyott session-re kattintva a játékos folytathatja az abban szereplő feladatokat.
 - Egy befejezett session-re kattintva az ahhoz a játékhoz tartozó statisztikák és értékelések tekinthetők meg.
- Egy küldetést lehetséges szövegesen (kommentként) és skálán (csillag) értékelni egy belőle indított játék végén. A session-ben résztvevő játékosok értékelése megtekinthető a tervező és az adott session-t esetleg moderátori szerepben indító számára.
 - A küldetés tervezője a játékhoz írt kommentet törölhet.
- Publikus küldetések esetében az adminisztrátoroknak is joguk van tervezői és moderátori tevékenységekhez. Tehát törölhetnek publikus küldetést, törölhet publikus küldetésnél szöveges értékelést.
 - További jogköre az adminisztrátornak, hogy megtilthatja más felhasználók számára a szöveges értékelés írását, ha az visszaél ezzel (saját belátása szerint dönthet erről).
 - Egy adminisztrátor másik felhasználónak adminisztrátori szerepet adhat.
- A küldetések statisztikája és értékelése:
 - A játékok során figyelembe vett statisztikák: a szerzett pontszám, teljesítési idő, adott esetben a próbálkozások száma.
 - A játékok statisztikája minden játék végén elérhető az aktuális session-ben résztvevők számára.
 - Egy publikus küldetésnél a statisztikák minden felhasználó számára elérhetők.
 - Publikus küldetésnél a szöveges értékelések és a skála alapúak minden felhasználó számára elérhetők.
 - Privát küldetésnél a szöveges értékelések és a skála alapúak csak a tervezők számára láthatók, illetve azon felhasználók számára, akikkel meg lett osztva a játék kódja.

Technikai részletek:

- Operációs rendszer: A feladatkiírás követelményei alapján úgy láttuk, hogy a legmegfelelőbb egy Android alapú mobilalkalmazás elkészítése lenne.
- Programozási nyelv: Az alkalmazás Kotlin nyelvben íródik, mivel ez az Android fejlesztés elsődleges nyelve.
- Felhasználókezelés: A Firebase Authentication lehetőséget nyújt alkalmazásra való regisztrációra, bejelentkezésre, kijelentkezésre, valamint felhasználói adatok utólagos módosítására.
- Adatok tárolása/kezelése: A Cloud Firestore egy rugalmas, jól méretezhető adatbázis a Firebase és a Google Cloud mobil-, web- és szerverfejlesztéséhez. Felhőalapú és valós idejű, vagyis szinkronban tartja az adatokat az adatbázis használói között a valós idejű listener-eken keresztül.

- **Térkép:** Az Androidhoz készült Google Maps SDK segítségével lehetőség van térképeket hozzáadni Android alapú alkalmazásokhoz. A Google Maps információkat is nyújthat a térkép helyeiről, és támogatja a felhasználói interakciókat, valamint a felhasználó jelölőket és ábrákat is hozzáadhat térképéhez.
- **Felhasználói felület:** Az Android nyújtotta lehetőségeket figyelembe véve a UI XML vagy Jetpack Compose alapú lesz, esetleg a kettő ötvözése.
- **Architektúra:** Az alkalmazás három, vagy háromnál magasabb szintű architektúrával lesz megvalósítva.

Fogalomjegyzék:

Alább listázzuk a specifikációban szerepeltetett kifejezéseket. A kifejezések sorrendje annak figyelembevételével lett kialakítva, hogy a kifejezések definíciója csak korábban definiált kifejezésekre hivatkozzon.

Felhasználó	Bármely természetes személy, aki használja az applikációt.
Küldetés	A küldetésnek van neve, leírása, készítője, készítési dátuma. Egy Küldetés lehet készülő, elkészült. Az elkészült lehet publikus vagy privát.
Feladat	Egy Küldetés több Feladatból áll. Egy Feladatnak van típusa, teljesítési követelménye, megoldása, teljesítésért járó pontszáma.
Játék	Egy futó Küldetés . Ez a tényleges használata egy korábban tervezett Küldetésnek . Játék során általában több Feladatot kell teljesíteni.
Session	Alternatív elnevezése a Játéknak . Kényelmi okokból vezettük be. Gyakran könnyebb így hivatkozni.
Adminisztrátor	Egy felhasználói szerep. Birtokában a felhasználó törölhet Küldetést , szöveges értékelést publikus küldetésben, illetve megvonhatja Felhasználóktól a szöveges értékelés írásának lehetőségét. Adminisztrátori szerepet csak adminisztrátor adhat más felhasználóknak.
Tervező	Egy felhasználói szerep. Tervezői szerepben a felhasználó új Küldetéseket állít össze. Törölhet saját maga által készített küldetést. Privát küldetéshez generáltathat hozzáférési kódot. Az általa készített küldetésekre vonatkozó szöveges értékeléseket törölhet.
Játékos	Egy felhasználói szerep. Játékosként egy adott Küldetésből létrehozott Játékot játszik a felhasználó. A Küldetéseket szövegesen és számszerű skálán értékelheti. Az Adminisztrátor megvonhatja a Játékosoktól a Küldetések szöveges értékelési jogát.
Moderátor	Egy felhasználói szerep. A Moderátor felügyel egy adott privát küldetést. A Küldetésekhöz Játékosok által írt szöveges vagy egyéb, gép számára nehezen értelmezhető megfejtéseket javít ki.
Jelvény	Felhasználók Jelvényeket kapnak jutalmul a program használata során.

Felhasználói szerepek és use-case:

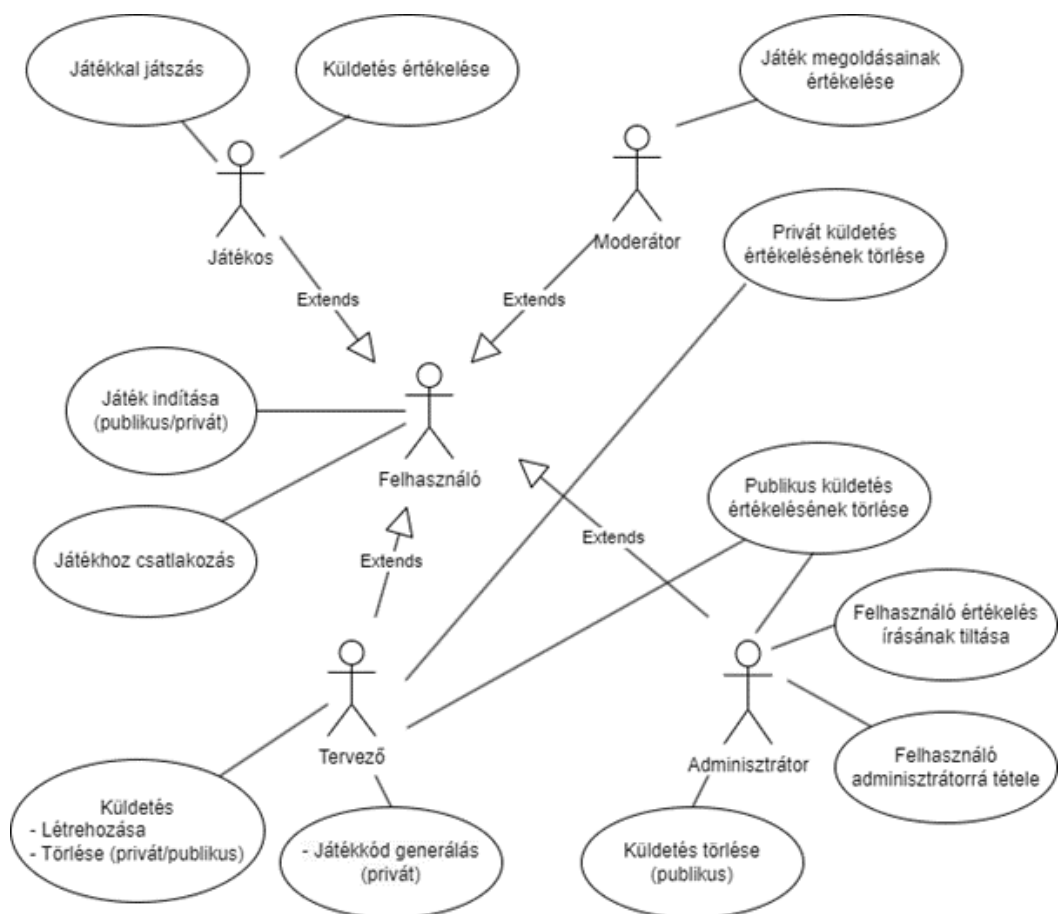
A Felhasználók saját belátásuk szerint használhatják az alkalmazást a definiált szerepkörök szerint. Ezen szerepkörök meghatározzák, hogy milyen akciókat hajthatnak végre az alkalmazásban.

A programban eddig specifikált szerepek és feladatkörük táblázatos formában:

Akció	Általános felhasználó	Játékos	Tervező	Moderátor	Adminisztrátor
Küldetés készítése			OK		
Küldetés törlése (privát)			OK		
Küldetés törlése (publikus)			OK		OK
Privát küldetéshez játék engedélyezőkód generálása			OK		
Küldetés session indítása (publikusból)	OK				
Küldetés session indítása (privátból)	OK (vele megosztott küldetésből)				
Session-be csatlakozás	OK				
Játék egy küldetéssel egy session-ben		OK			
Küldetés értékelése skálán		OK			
Küldetés értékelése szövegesen		OK			
Ellenőrzi az emberi ellenőrzést igénylő feladatok megoldásait				OK	
Privát küldetésben szöveges értékelés			OK		

törlés					
Publikus küldetésben szöveges értékelés törlés			OK		OK
Másik felhasználó szöveges értékelés írásának tiltása					OK
Másik felhasználó adminisztrátorrá tétele					OK

Use-case diagram a főbb funkciókkal és használati szerepekkel (egy use-case több szerepkörhöz is tartozhat):



Dokumentáció

Szoftverarchitektúrák tantárgy házi feladat

*Közösségi kincskereső a belvárosban - Community treasure hunt in
downtown*

Készítették:

Csapó Botond Balázs

Gyulai Gergő László

Farkas Réka

Simon Zoltán

Feladatkiírás:	1
Fejlesztői csapat:	1
Részletes feladatléírás:	1
Technikai részletek:	3
Fogalomjegyzék:	4
Felhasználói szerepek és use-case:	5
Bevezetés	8
A rendszer célja, funkciója és környezete	9
Feladatkiírás:	9
A rendszer által biztosítandó tipikus funkciók	10
Adatléírás	10
Badge.....	11
Feedback.....	11
Level.....	11
Mission.....	11
Session.....	12
Task.....	12
TaskSolution.....	12
User.....	13
A RainbowCake architektúra	13
Jetpack Compose	15
Az alkalmazás nézetei	15
Az alkalmazás fejlesztése közben felhasznált eszközök	20
Az alkalmazás telepítése	20
Összefoglalás	21
Továbbfejlesztési lehetőségek	22
Hivatkozások	23

Bevezetés

Az különböző közösségi kincskereső játékok nagy népszerűségnek örvendenek fiatal baráti társaságokban. Kincskeresőn itt olyan rejtvényfejtő játékot értünk, amiben a játékosok egyedül vagy csoportosan keresnek elrejtett nyomokat. A megtalált nyomokat felhasználva feladványokat oldanak meg. Több egymás utáni feladatot megoldva egy végső megoldáshoz jutnak. Legyen szó csapatépítő tréningekről vagy együttes kirándulásokról, a közösségsszervezők előszeretettel alakítanak ki ilyen játékokat. Ezek a játékok kiváló lehetőséget biztosítanak a csapatokon belüli kapcsolatok megszilárdítására. Egyes pedagógiai célzatú kincskereső játékok jelentős helytörténeti ismeretekkel gazdagíthatják a résztvevőket.

Példaként elképzelhetünk egy esetet, amikor a játékosok egy történelmi város utcáin bolyongva derítenek ki információt az ott élt történelmi személyekről. A felderített adatokat kombinálva kapják a kincskereső feladatainak megfejtését. Az ilyen és ehhez hasonló játékos tanulást ösztönző küldetések igazán jól illeszkednek a huszonegyedik század fiataljainak információ befogadási szokásaihoz. Egy szokásos kincskereső sikeres levezésnylése nagy

erőbefektetést követel a játék szervezőitől. A feladatok precíz megtervezése elkerülhetetlen. Ez egy kreatív személy munkáját igényli. Egy kész játék levezénylése azonban meglátásunk szerint jól automatizálható. Egy segédprogram segíthet elkerülni a félreértéses helyzeteket, amikor a játékosok rosszul értelmeznék bizonyos feladatokat. Ennek következtében elakadhatnak. Az ilyen helyzetek főlegesen frusztrálják a résztvevőket. Más helyzetekben jó lenne betartatni a feladatok tervezett megoldási sorrendjét. Megeshet ugyanis, hogy a játékosok akár véletlenül is, de nem a megfelelő sorrendben találják meg az elhintett nyomokat. Ez további félreértéseket szül, vagy jogtalan előnyhöz juttat csapatokat. Szoftverünk egy keretrendszert kíván nyújtani közösségi kincskeresők lebonyolításához. Lehetőséget nyújt, hogy a legkülönbözőbb témájú és felépítésű játékokat egy jól szervezett, átlátható formátumú rendszerben lehessen lebonyolítani.

A rendszer célja, funkciója és környezete

Rendszerünk kincskereső játék tervezését és lebonyolítását teszi lehetővé egy Android mobilalkalmazás formájában.

Feladatkiírás:

A hallgatók feladata egy olyan webes oktató/kaland játék elkészítése, amely a játékosokat a város különböző pontjaira küldi bizonyos küldetések elvégzésére.

- A küldetéseket maguk a játékosok állítják össze egymásnak. A küldetéseket lehet értékelni szövegesen és pontozással.
- A küldetések több részfeladatból állnak és lehetőséget adnak több teljesítési módra.
- A játék tartalmazzon gamification mechanizmusokat, melyek célja a játékosok bátorítása, hogy több küldetést teljesítsenek.
- A játék legyen elérhető mobil készülékekről és ellenőrizze a játékos lokációját GPS-el.
- A moderátor szerkeszthet, felfüggeszthet és törölhet küldetéseket, letilthat felhasználókat.

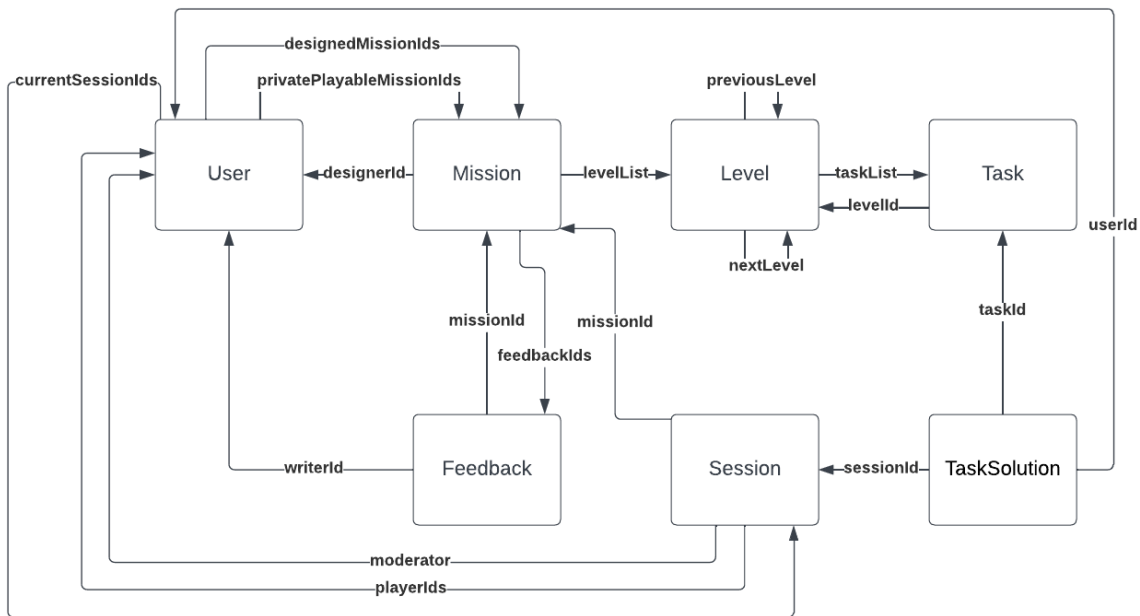
A rendszer által biztosítandó tipikus funkciók

A rendszernek az alábbi funkciókat biztosítania kell:

- Felhasználók regisztrálása és beléptetése az alkalmazásba.
- A Játékosok tudjanak küldetéseket teljesíteni, aktív játékokhoz csatlakozva.
- A felhasználók képesek legyenek küldetéseket létrehozni, bennük különböző típusú feladatokat alkotni. Ezeket tudják megosztani más játékosokkal.
- A játékokat tudják moderálni moderátorok. Nekik képesnek kell lenniük látni az aktív játékosokat, és a feladatokra adott válaszaikat, illetve tudják értékelni azokat.
- A felhasználók tudják értékelni a küldetéseket, illetve meg tudják tekinteni a más felhasználók által írt értékeléseket az adott küldetéshez.

Adatleírás

Ebben a fejezetben bemutatjuk a szoftver által használt adatstruktúrát. Rendszerünk adatmodelljének felépítését grafikus formában az alábbi diagram mutatja be. Ezt követően részletesen leírjuk a tárolt adatokat.



Feedback

A játékosok által hagyott visszajelzés, amelyet egy-egy küldetés teljesítése után hagynak. Ezzel segítik a többi játékos könnyebb tájékozódását a különböző minőségű missziók közti válogatás során.

- Id: Egyedi UUID azonosító.
- MissionId: A misszió azonosítója, amelyhez az értékelés tartozik.
- WriterId: A felhasználó azonosítója, aki a visszajelzést írta.
- Stars: Számszerűsített értékelés, amit az értékelő csillagozással ad meg.
- Comment: A szöveges értékelés.
- CommentState: A visszajelzés aktuális állapota. Lehetséges értékei SHOW vagy HIDE. Ennek megfelelően jelenik meg az értékelés a rendszerben.

Level

A missziók egy adott szintje, amely valamilyen megoldandó feladatot tartalmaz.

- Id: Egyedi UUID azonosító.
- LevelNumber: A szint sorszáma a misszió belül.
- LevelType: A szint típusa. Lehetséges értékei Minimum 1, Exactly 1, All. Ez befolyásolja, hogy a továbblépéshez hány feladatot kell elvégezni a szinten.
- ShowNextLevel: Meghatározza, hogy a következő szint látszódjon-e.
- TaskList: A szinthez tartozó feladatok listája. Egy szinthez több feladat tartozhat. Azt, hogy ezek közül hányat kell elvégezni a LevelType befolyásolja.

Mission

Egy misszió az, amit a tervező összeállít, és amiből létrehozott küldetést a játékosok játszanak. Ez a programunk központi részét képezi.

- Id: Egyedi UUID azonosító.
- Name: A missziót azonosító tetszőleges fantázianév. Ezt a misszió tervezője szabja meg.

- **Description:** A misszióról szóló rövid szöveges bemutató. A tervező írja.
- **CreationDate:** A misszió létrehozásának dátuma.
- **ModificationDate:** Az utolsó módosítás dátuma.
- **HoursToSolve:** A tervező által megszabott időkorlát óra része, ami a feladatok megoldására rendelkezésre áll.
- **DaysToSolve:** A tervező által megszabott időkorlát nap része, ami a feladatok megoldására rendelkezésre áll.
- **AccessCode:** Egy generált kód, amit megosztva válik elérhetővé a misszió más játékosok számára.
- **IsPlayableWithoutModerator:** Egy logikai változó, amely meghatározza, hogy a missziót moderátor közreműködése nélkül lehet-e játszani. Ezt elsősorban a feladatok típusa határozza meg.
- **Visibility:** A misszió láthatósága a felhasználók számára. Lehetséges értékei PRIVATE és PUBLIC.
- **State:** Leírja, hogy az adott misszió még tervezés alatt áll, vagy már elkészült. Lehetséges értékei DESIGNING és FINISHED.
- **DesignerId:** A missziót tervező felhasználó egyedi azonosítója.
- **LevelList:** A missziót alkotó szintek listája. Egy misszió több szintből áll. Ezek mindegyike egy vagy több feladatot tartalmaz.
- **MissionType:** A misszió típusa. Lehetséges értékei egy előre definiált halmazból kerülnek ki: Outdoor, Indoor, Nocturnal, Hard, Easy, Education, Adventure, Photo, Urban.
- **FeedbackIds:** A misszióhoz kapcsolódó felhasználói értékelések egyedi azonosítóinak listája.

Session

Amikor a játékosok játszanak egy missziót, akkor az egy küldetésre példányosodik. Egy ilyen küldetést reprezentál a Session.

- **Id:** Egyedi UUID azonosító.
- **Name:** A misszió neve.
- **MissionId:** A küldetéshez tartozó misszió azonosítója.
- **AccessCode:** A küldetéshez való csatlakozást lehetővé tevő azonosító.
- **StartDate:** A küldetés indításának dátuma.
- **Moderator:** A küldetés moderátor felhasználójának azonosítója, amennyiben szükséges moderátor.
- **PlayerIds:** A játékban résztvevő felhasználók azonosítóinak listája.

Task

A missziókat alkotó szintek több feladatból állhatnak. A Task egy feladat adatait tárolja.

- **Id:** Egyedi UUID azonosító.
- **LevelId:** A feladatot tartalmazó szint azonosítója.
- **Title:** A feladat címe. Ezt a tervező állítja be.
- **Description:** A feladat szöveges leírása.
- **Score:** A feladat elvégzéséért kapható pontszám.
- **Answers:** A megadható válaszlehetőségek.
- **TaskType:** A feladat típusa, amely lehet ListedAnswer, NumberAnswer, DateAnswer, MapAnswer, OrderAnswer, TextAnswer, ImageAnswer.

TaskSolution

A játék működéséhez szükséges típus, amely egy feladatra egy játékos által adott megoldást reprezentál.

- Id: Egyedi UUID azonosító.
- SessionId: A küldetés azonosítója, aminek keretein belül született a feladat megoldás.
- TaskId: A feladat azonosítója.
- UserId: A megoldást beadó felhasználó azonosítója.
- UserAnswer: A felhasználó által adott konkrét válasz.
- Checked: Logikai változó, ami a megoldás leellenőrzöttségét mutatja.
- Correct: A megoldás helyességét mutató logikai változó.

User

A szoftver felhasználóinak adatait tárolja.

- Id: Egyedi UUID azonosító.
- Name: A felhasználó neve.
- Score: A játékokban gyűjtött pontszám. Hasonlóan működik egy egyszerű XP rendszerhez.
- ActualLanguage: A felhasználó által beszélt nyelv meghatározása. Az alapértelmezett az angol nyelv.
- IsAdmin: Logikai változó, amely meghatározza, hogy a felhasználó rendelkezik-e adminisztrátori jogosultsággal.
- AbleToEvaluate: Meghatározza, hogy az adott felhasználónak szabad-e értékelést adni missziókra. Ezt a jogot egy adminisztrátor megvonhatja a felhasználótól, hogyha a felhasználónak korábban sértő megnyilvánulásai voltak.
- PrivatePlayableMissionIds: Egy felhasználó számára elérhető, privát missziók azonosítóinak listája, amelyek eléréséhez a tervezőnek külön meg kellett adni a hozzáférési kódot.
- DesignedMissionIds: A felhasználó által tervezett missziók azonosítóinak listája.
- CurrentSessionIds: A jelenleg futó küldetés azonosítója, amelyben részt vesz a felhasználó.

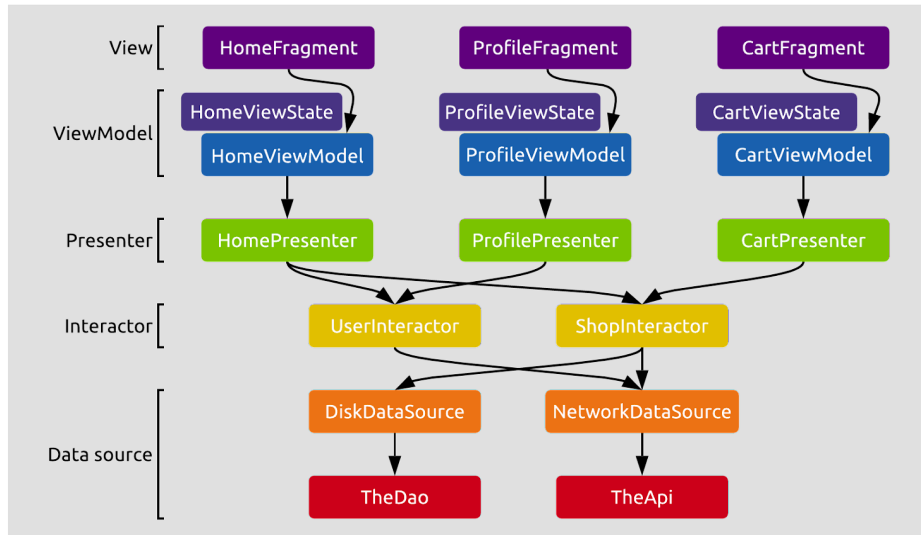
A RainbowCake architektúra

A RainbowCake egy architektúra koncepció Androidos alkalmazások készítéséhez. A RainbowCake erősen épít a Kotlin nyelv tulajdonságaira és a Google Jetpack könyvtárakra. Ez az architektúra nem más, mint pár leszármazás, amelyek osztályokat és egyéb komponenseket tartalmaznak, amelyeket használhatunk a saját alkalmazásunk készítésekor. Az architektúra főbb céljai közé tartozik a felelősségek elválasztása, a nézetek (Views) és a Nézet-Állapotok (ViewState) közötti konzisztens állapot fenntartása, a konfiguráció változások kezelése és a szálkezelés könnyítése.

Az architektúra az alábbi rétegekből áll:

- Views: (Fragments or Activities) Az alkalmazás egy nézetét reprezentálják a képernyőn. A hozzájuk tartozó ViewState-ben található adatokat jelenítik meg, illetve küldenek le oda bemeneti eseményeket.
- ViewModels: Eltárolják a nézet jelenlegi állapotát, kezelik a felhasználói felület logikáját, és frissítik a nézetet, ha változás következik be az állapotukban. Valamint rutinokat kezdenek, amennyiben bemeneti eseményeket kapnak, és ekkor függvényhívásokat továbbítanak a Presenter rétegbe.

- **Presenters:** A ViewModelből kapott hívásokat, feladatokat elvégzik az egyes Interactorokkal. Szálkezelést is végeznek. A feladatok eredményét visszaadják a ViewStateeknek, így módosítva állapotukat.
- **Interactors:** Az alkalmazás logikai rétegét tartalmazzák. Módosítják az adatokat, és számításokat végeznek el rajtuk. Nincsenek képernyőhöz kötve, inkább funkcionalitás szerint vannak csoportosítva.
- **Data sources:** Az alkalmazás adatelérési rétegét tartalmazzák. Biztosítják az adatok biztonságos hozzáférését, valamint elrejtik az adatokhoz való hozzáférés logikáját az alkalmazás más rétegei előtt.



Az elkészült applikáció a RainbowCake architektúrát alkalmazza. Az applikáció adatforrásként egy FirebaseDataSource nevű osztályt használ, amely hívásokat intéz a Firestore adatbázishoz. Ez az applikáció egyetlen, az adatbázissal közvetlenül kommunikáló része. Az Interactorok ezt az osztályt használják az adatbázissal való kommunikációhoz.

Az Interactor osztályok felelősségük szerint vannak csoportosítva. Ezeket az Interactorokat használják az egyes Presenter osztályok. Egy Interactort több Presenter osztály is használhat. Az Interactor funkcionalitást csoportosítják az egyes osztályok:

- **AuthInteractor:** Autentikációhoz, a felhasználók regisztrációjához és be-/kiléptetéséhez szükséges funkciókat implementálja.
- **MissionInteractor:** A Küldetések befolyásoló funkcionalitást tartalmaz. Ide tartozik például a egy új küldetés feltöltése, egy létező küldetés módosítása vagy törlése.
- **SessionInteractor:** A játékok (Sessionök) manipulálását végzi. Ez az osztály tartalmazza a funkcionalitást, amellyel új játékot lehet létrehozni, hozzá lehet csatlakozni már létező játékokhoz, illetve információkat lehet lekérni az épp futó játékokról.
- **TaskSolutionInetractor:** Az egyes feladatok megoldását tölti fel az adatbázisba.
- **UserInteractor:** Felhasználói adatokat lehet lekérni az osztállyal.
- **ImageInteractor:** Képek fel- és letöltéséért felel

Az Interactor osztályokat a Presenter osztályok használják az alkalmazás működése során. Az alkalmazás felépítését innenől az egyes nézetek szerint érdemes bemutatni, mivel minden nézethez

készült egy Presenter, egy ViewState, egy ViewModel és egy Fragment osztály. Ezek az osztályok az architektúra leírásának megfelelő módon viselkednek.

Az alkalmazásban az alábbi nézetek készültek el:

- Bejelentkezés nézet
- Regisztrációs nézet
- Küldetések nézet: Publikus küldetések, Futó küldetések, Saját küldetések
- Küldetés szerkesztés nézet
- Feladat szerkesztés nézet
- Fiók nézet
- Felhasználók nézet
- Játék nézet
- Feladat nézet
- Moderátor nézet: Megoldások nézet, Megoldás értékelése nézet

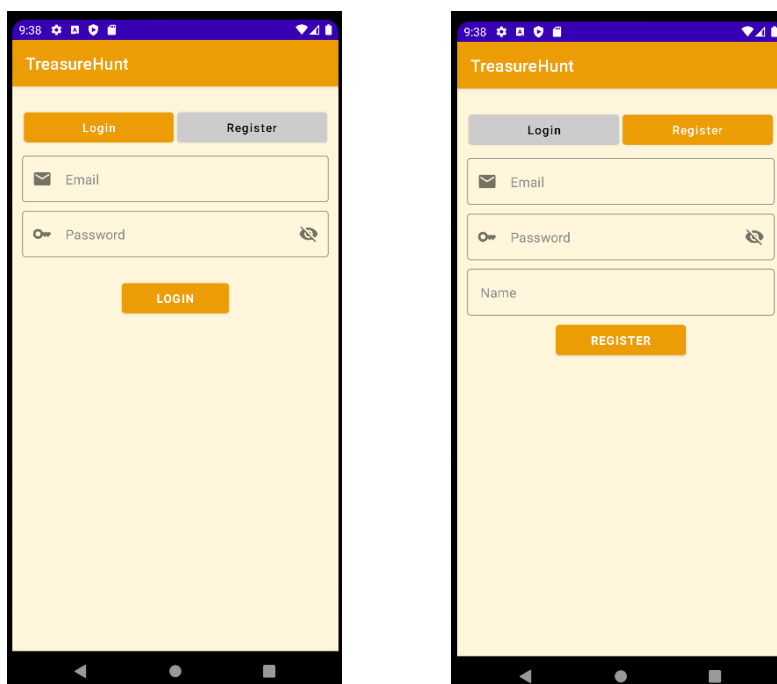
Jetpack Compose

Az alkalmazás nézetei a Jetpack Compose eszközzel készültek el. Ez az eszköz az ajánlott natív android alkalmazások felhasználói felületének elkészítéséhez. Az eszköz gyors és egyszerű eszközöket biztosít a felhasználói felületek elkészítéséhez. Az alkalmazás készítése közben felhasználásra került a Compose és a Preview funkciók alkalmazása.

A Compose funkcióhoz el kell látni a kódrészletet a Composable annotációval, ezután meg lehet kezdeni a felhasználói felület elkészítését. A Preview annotációval ellátott kódrészleteket pedig meg lehet tekinteni a preview ablakban az Android Studio felületén.

Az alkalmazás nézetei

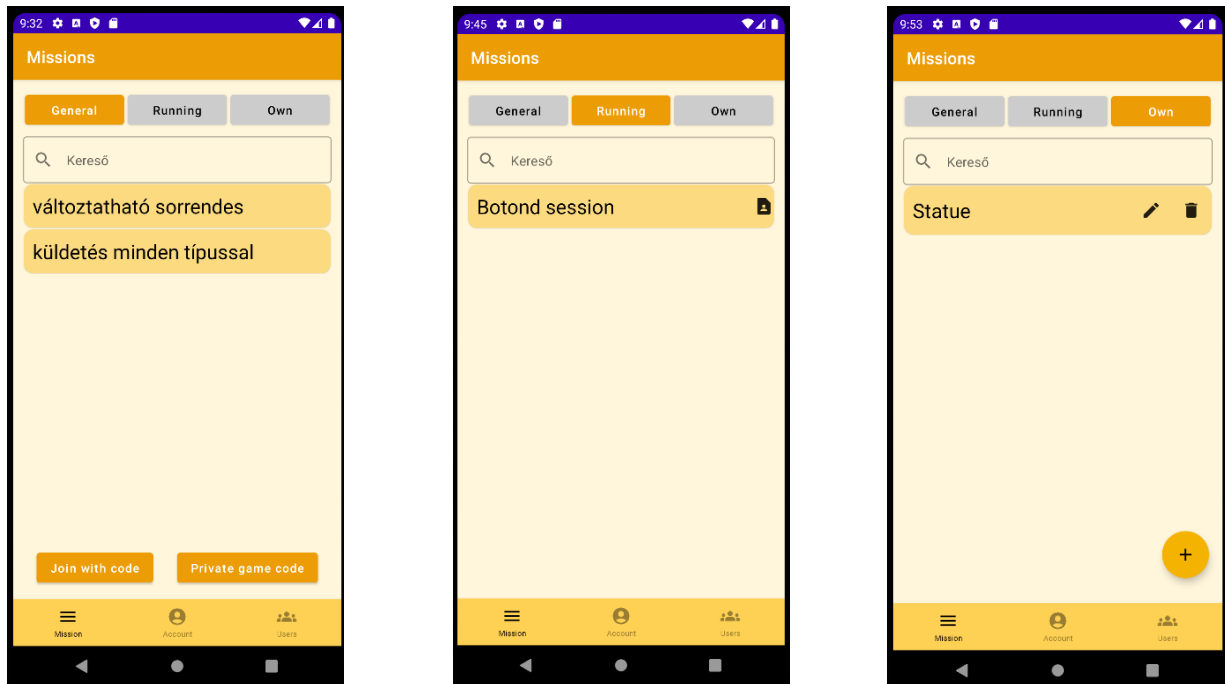
Bejelentkezés és Regisztrációs nézet:



A bejelentkezéshez először saját felhasználói fiókot kell regisztrálnunk. A Regisztrációt követően a megadott email címre megerősítő levelet kap a felhasználó. Az emailben megtalálható linkre kattintva

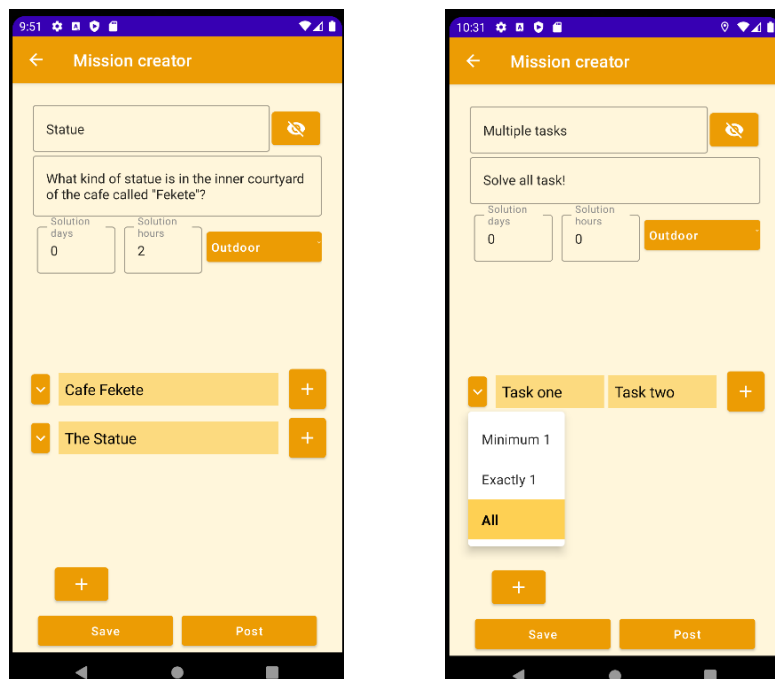
aktiválja a felhasználó a fiókját, és bejelentkezhet az alkalmazásba. Ezután átvált az alkalmazás a Küldetések nézetbe.

Küldetések nézet:



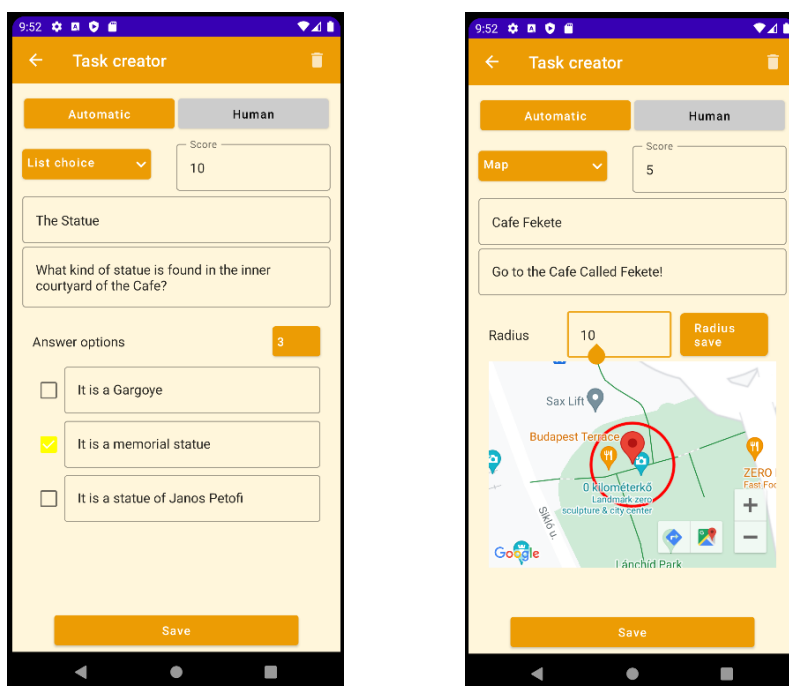
A küldetések nézetben a felhasználó küldetések listáját látja. A felhasználó a felül található gombokkal válthat a publikus küldetések (General), az épp futó küldetések (Running) és a saját (Own) küldetések között. A General nézetben található “Join with code” és “Private game code” gombbal a felhasználó csatlakozhat egy futó küldetéshez. A gomb megnyomása után egy dialógus ablakban adhatja meg a kódot. Ezután a játékos áttugrik a játék nézetbe. A saját küldetések fülön a felhasználó a küldetéseit szerkesztheti, törölheti, illetve létrehozhat új küldetéseket is a lebegő plusz jelre kattintva.

Küldetés szerkesztés nézet:



Amikor a felhasználó egy saját küldetését szerkeszteni akarja, vagy új küldetést készít, akkor a küldetés szerkesztés nézetre ugrik. A felhasználó módosíthatja a küldetés nevét, leírását, időkorlátot állíthat be hozzá, illetve megadhatja, hogy a küldetést kültéren kell-e megoldani vagy sem. A felhasználó a “Save” gombbal elmenti a változásokat, a “Post” gombbal meg publikussá teszi azt, így elérhetővé válik más játékosok számára, hogy játékokat indítsanak belőle. Az egyes feladatokra, illetve a plusz gombokra nyomva a felhasználó hozzáadhat új feladatokat, illetve módosíthatja a meglévőket. Ekkor az alkalmazás a feladat szerkesztés nézetre ugrik. A feladatokat hozzáadhatja egymás alá, vagy elhelyezhet három feladatot egy sorban. A sorok mellett látszódó legördülő gombbal kiválaszthatja, hogy a sorban lévő feladatok közül elég-e egyet megcsinálni, hogy a következő sort elérje a játékos, vagy mindegyiket meg kell csinálnia, illetve kiválaszthatja, hogy a feladatok közül maximum csak egy feladat kerül majd pontozásra.

Feladat szerkesztés nézet:



A felhasználó a feladat szerkesztés nézetre ugrik, amennyiben módosítani akar egy feladatot, vagy létrehoz egy újat a küldetés szerkesztés nézetben. Az első képen egy küldetés látható két feladattal. A felhasználó megadhatja, hogy a feladat kiértékelése automatikus legyen, vagy moderátor, azaz egy másik ember által legyen ellenőrizve. A feladat kiértékelésének módjától függően más választípusok közül választhat a küldetés készítője. A feladattípusok listája:

Automatikusan kiértékelhetőek:

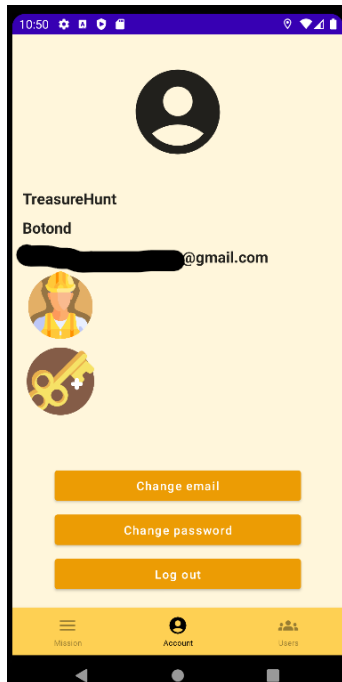
- Lista: A játékosnak több megoldás közül kell választania. A feladat készítőjének meg kell adnia a lehetséges válaszokat, és ki kell jelölni a helyes választ vagy válaszokat.
- Szám: A válasz szám formátumú.
- Dátum: A válasz dátum formátumú.
- Térkép: A válasz térkép formátumú. A feladat készítőjének meg kell adnia egy lokációt és egy sugarat, amin belül ha elér a felhasználó, akkor teljesítette a feladatot.
- Rendezett Lista: A játékosnak helyes sorrendbe kell rendeznie a megadott válaszokat. A feladat készítőjének el kell készítenie a válaszokat a helyes sorrendben.

Ember által kiértékelhetőek:

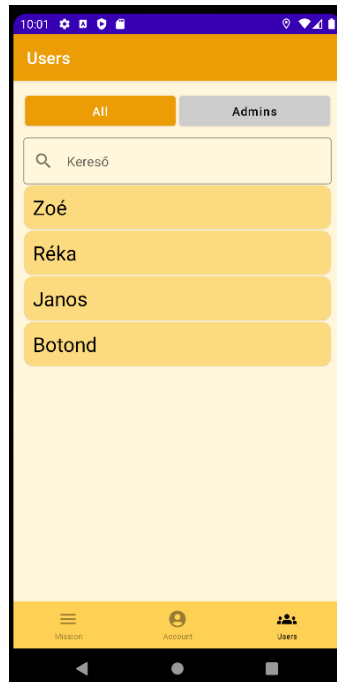
- Szöveg: A játékos bármit írhat válaszként.
- Kép: A játékosnak képet kell feltöltenie megoldásként.

A "Save" gomb hatására a feladat mentésre kerül. A jobb felül látható kuka ikonra kattintva a feladat törlésre kerül.

Fiók nézet:

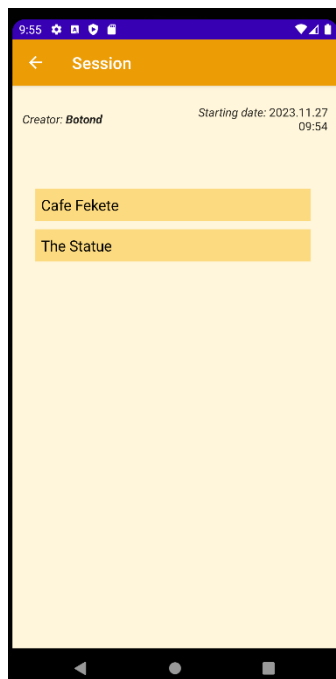


Felhasználók nézet:

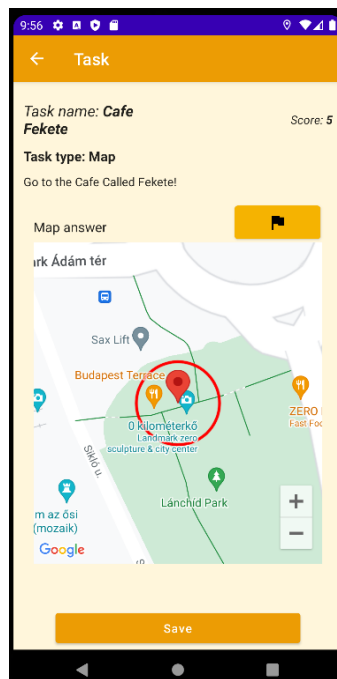


A küldetések nézetből a felhasználó az alsó navigációs sávban át tud váltani a fiók és a felhasználók nézetbe. A fiók nézetben a felhasználó módosíthatja az adatait: Az email címét és a jelszavát, valamint ki tud jelentkezni a fiókjából. A felhasználók nézetben megtekintheti az alkalmazást más felhasználóit, illetve szűrhet az adminisztrátor felhasználókra a felső menüsáv segítségével.

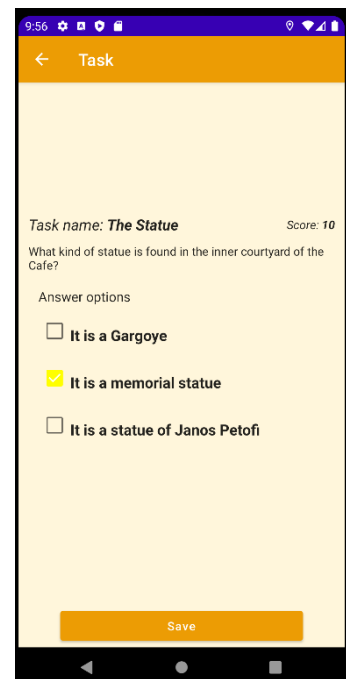
Játék nézet:



Térkép válasz:

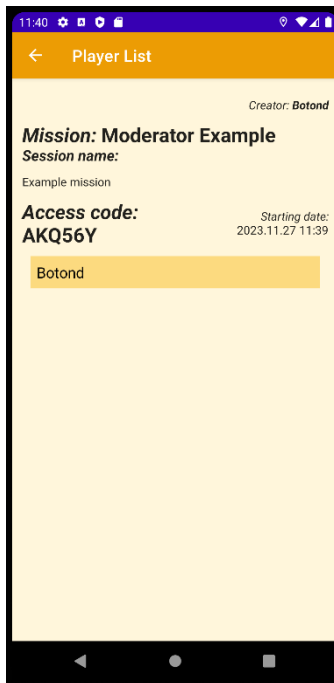


Lista válasz:

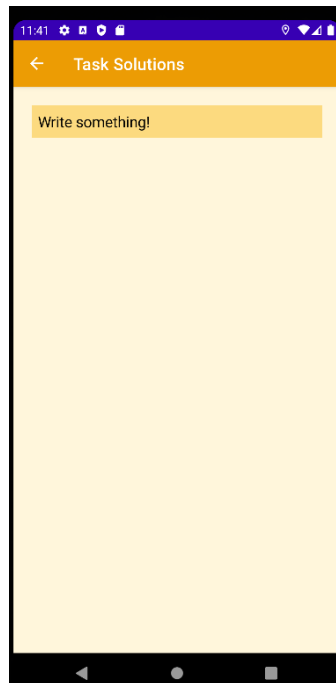


A küldetések nézetben a felhasználó egy kóddal csatlakozott egy futó játékhoz, ekkor lép a játék nézetre. A nézetben a játékos egy küldetést lát, annak a feladatait kell megoldania. A feladatokra kattintva a játékos megadhatja a válaszokat. A fenti képeken egy térkép válasz és egy lista válasz látható. A válaszok a "Save" megnyomásakor az adatbázisba kerülnek, és a játék moderátora ki tudja értékelni azokat, amennyiben szükséges.

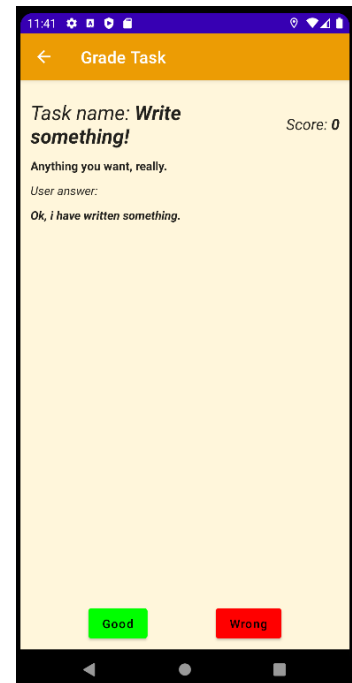
Moderátor nézet:



Megoldások nézet:



Megoldás értékelés nézet:



Amennyiben egy küldetés tartalmaz olyan feladatot, amelynek van nem automatikusan kiértékelhető válasza, akkor a játéknak szüksége van egy moderátorra, hogy kiértékelje azokat. Az aktív játék nézete látható az első képen, itt láthatóak a játékhoz csatlakozott játékosok is. Az egyes játékosokra kattintva a moderátor megtekintheti a játékos válaszait. A válaszokra kattintva pedig értékelheti azt, helyesnek vagy helytelennek.

Az alkalmazás fejlesztése közben felhasznált eszközök

- Android Studio: Fejlesztőkörnyezet (1)
- Firebase: Felhasználók kezelése (bejelentkezés/regisztráció) (2)
- Firestore: Adatbázis, adatok tárolása és manipulálása (3)
- GitHub: Verziókezelés (4)
- Google Docs: Specifikáció, Dokumentáció készítéséhez felhasznált eszköz (5)
- Jetpack Compose: Felhasználói felület elkészítése (6)
- Kotlin: Az alkalmazás elkészítéséhez felhasznált programozási nyelv
- Lucidchart: Rajzeszköz, az adatbázis osztályainak a tervezéséhez használtuk (7)
- Microsoft Teams: Csapattagok közötti kommunikáció (8)
- Notion: Jegyzet alkalmazás (9)
- RainbowCake: Az alkalmazás architektúrája (10)
- Wireframe: UI tervező eszköz (11)

Telepítési útmutató

Rendszerkövetelmények

Mielőtt elkezdené a telepítési folyamatot, győződjön meg róla, hogy az eszköz megfelel az alábbi követelményeknek:

Android operációs rendszer verzió legalább 24-es.

Legalább 32 MB tárhely.

Internetkapcsolat (az alkalmazás letöltéséhez).

Az alkalmazás letöltése

Nyissa meg a Google Play Áruházat:

Érintse meg a Google Play Áruház ikonját az eszköz kezdőképernyőjén vagy az alkalmazásfiókban.

Keresse meg a Kínskereső *TreasureHunt* alkalmazást:

A keresősávba írja be az alkalmazás nevét, majd érintse meg az Enter billentyűt.

Az alkalmazás lokalizálása:

Amint megtalálta az alkalmazást a keresési eredmények között, érintse meg a lapot az alkalmazás oldalának megnyitásához.

Az alkalmazás letöltése:

Érintse meg a "Telepítés" gombot az alkalmazás letöltéséhez és telepítéséhez az eszközödre.

Engedélyek

A telepítés során az alkalmazás bizonyos engedélyeket kérhet. Ezek az engedélyek szükségesek az alkalmazás megfelelő működéséhez. Fogadja el az engedélyeket a zökkenőmentes felhasználói élmény érdekében.

Összefoglalás

A feladat megoldása során létrehoztunk egy mobilalkalmazást, amellyel a felhasználók játékos küldetéseket tudnak létrehozni, azokkal játszani, illetve azokat értékelni. A fejlesztés közben több modern tervezőeszközt is használtunk, amelyek vizuális segédeszközként támogatták a munkánkat. Az implementálás során pedig alkalmaztunk már meglévő technológiákat, mint a Firebase, Jetpack Compose és a RainbowCake, így gyorsítva fel a munkát.

A munka eredménye az elkészült mobilalkalmazás, amelynek a szoftverarchitektúrája megfelel a RainbowCake architektúrájának, és a Firebase adatelérési szolgáltatással kezeli a felhasználókat, és az alkalmazás használatához szükséges adatokat.

Továbbfejlesztési lehetőségek

Az alkalmazást tovább lehet fejleszteni a csoportos játék létrehozásával. Jelenleg mindegyik játékkal egy ember egyedül játszik. Meg lehetne valósítani, hogy egy játék létrehozása után a játékhoz egyszerre több ember is hozzáférjen, és megoldásokat adjon meg az egyes feladatokhoz. Ekkor a játékosoknak együtt kell működniük, hogy ne oldják meg például ugyanazt a feladatot egyszerre, és fel kell osztaniuk a feladatokat.

Az alkalmazást tovább lehet fejleszteni más feladattípusok hozzáadásával is. Valamint az ellenőrzés módját is lehet javítani, például a képeket mesterséges intelligenciával ellenőrizni.

A játékba létre lehetne hozni egy globális ranglétra rendszert, ahol a játékosok egymás ellen tudnak versenyezni. Ekkor a játékok megoldásáért pontok járnának, és a több ponttal rendelkező játékosok értelemszerűen feljebb lennének a ranglétrán. Ekkor a feladatok gyorsabb teljesítéséért több pont járna például.

Hivatkozások

- [1] Android Studio: <https://developer.android.com/>
- [2] Firebase: <https://firebase.google.com/>
- [3] Firestore: <https://firebase.google.com/products/firestore>
- [4] GitHub: <https://github.com/>
- [5] Google Docs: <https://docs.google.com/>
- [6] Jetpack Compose: <https://developer.android.com/jetpack/compose/>
- [7] Lucidchart: <https://lucid.app/lucidchart/>
- [8] Microsoft Teams: <https://www.microsoft.com/hu-hu/microsoft-teams/>
- [9] Notion: <https://www.notion.so/>
- [10] RainbowCake: <https://github.com/rainbowcake/rainbowcake>
- [11] Wireframe: <https://wireframepro.mockflow.com/>