**A garden with a barn and windmills

Description automatically generated**

**FrissKert**

**Készitette:**

Gyöngyösi Róbert

Jakab Eduárd

Jakab Evelin

Pap Roland Levente

**Szak:** Infokommunikációs hálózatok és rendszerek (Távközlés) III.év

Tantárgy: Szoftverrendszerek tervezése

Vezető tanár: Ferencz Katalin, Dr. Szántó Zoltán

Sapientia Erdélyi Magyar Tudományegyetem

Marosvásárhelyi kar, 2023-2024

**Tartalomjegyzék**

[1.Bevezető 3](#_Toc155481531)

[2.A projekt célja 3](#_Toc155481532)

[3.Követelmény specifikáció 3](#_Toc155481533)

[3.1. Felhasználói követelmények -> Use Case Diagram 4](#_Toc155481534)

[3.2. Rendszerkövetelmények 5](#_Toc155481535)

[3.2.1. Funkcionális követelmények 5](#_Toc155481536)

[3.2.2. Nem funkcionális követelmények 5](#_Toc155481537)

[3.3. Fontosabb műveletek magyarázata 5](#_Toc155481538)

[3.3.1. Felhasználó regisztrációja 5](#_Toc155481539)

[3.3.2. Felhasználó bejelentkezése 6](#_Toc155481540)

[4. Tervezés 7](#_Toc155481541)

[4.1. Wireframe 7](#_Toc155481542)

[4.2.Architektúra 7](#_Toc155481543)

[4.3. Modulok leírása 9](#_Toc155481544)

[4.3.1. Adatbázis 9](#_Toc155481545)

[4.3.2. Aktivitás diagramok 10](#_Toc155481546)

[a. Aktivitás diagram a bejelentkezéshez 11](#_Toc155481547)

[b. Aktivitás diagram a vásárlás – és eladáshoz 11](#_Toc155481548)

[4.4.Menedzselés 13](#_Toc155481549)

[4.5.Verziókövetés – GitHub - Kanban 13](#_Toc155481550)

[5. Alkalmazás működése 13](#_Toc155481551)

[5.1. UI – konkrét megvalósítás 13](#_Toc155481552)

[6. Összegzés 13](#_Toc155481553)

[6.1. További fejlesztési lehetőségek 13](#_Toc155481554)

[7. Bibliográfia/használati tool-ok 13](#_Toc155481555)

# 1.Bevezető

Minden ember tapasztalhatta már élete során, hogy nincs ideje elmenni az üzletbe, piacra, stb... Megtapasztalhatta már azt is, hogy az üzletekben a zöldség, illetve a gyümölcsök íze és minősége nem olyan volt mint a hazai termelőké. Ezek mellett, a kis termelők gondjai, hogy nem tudják, hogy hogyan adhatnák el könnyebben illetve gyorsabban a terményeiket.

Igy mi ki talátuk, hogy hogyan könnyithetnénk meg a vásárlók vásárlásait és a kistermelők eladásait. Létre hoztunk egy felületet, a FrissKert-et, amely működik weboldalon. A termelők könnyedén feltölthetik a FrissKert oldalára a terményeiket, amelyeket más felhasználók is láthatnak az oldalon a regisztrálás után. Így könnyedén eladhatnak és vásárolhatnak a felület felhasználói.

Fontos kiemelni, hogy egy weboldalon mindig lehet valamit fejleszteni, javitani. Sok kiegészíteni való dolog van még a weboldalon, de az alábbi fázisban a következő funkciók kerültek megvalósításra:

* UI felület
* Bejelentkezés/ regisztráció
* Vásárolható termékek megtekintése
* Új termék feltöltése
* Kapcsolatfelvétel az eladóval e-mailen

# 2.A projekt célja

Projektünk célja, hogy hozzá segítsük az embereket a friss zöldség és gyümölcs vásárlásához direkt a kis termelők kertjéből, ugyanakkor ezzel segítsük a termelők terményeinek eladásait is a modern eszközök használatával. A termelő megoszthatja a terményeit az applikációban/ weboldalon, amelyeket a többi felhasználó megnézheti és láthatja, hogy hol helyezkedik el az a termék, és felveheti a kapcsolatot a termék termelőjével email illetve a megadott telefonszámon keresztül.

A projekt tervezésekor a következő célok kerültek megfogalmazásra:

* megkönnyíti a vásárlók mindennapi vásárlásait
* segít a friss és házi termékek (zöldségek illetve gyümölcsök) egyszerűbb elérését
* gyorsabb és hatékonyabb vásárlás
* egyszerű interakció a felhasználó és az adott használati felület között

# 3.Követelmény specifikáció

## 3.1. Felhasználói követelmények -> Use Case Diagram

A Használati Eset diagram (Use Case diagram) a felhasználó lehetséges interakcióinak ábrázolása a rendszerrel. A Use Case diagram különböző használati eseteket és különböző típusú felhasználókat mutat be, amelyekkel a rendszer rendelkezik.

* Regisztráció: Az alkalmazás használata kizárólag regisztrált felhasználók számára engedélyezett. Az alkalmazás megnyitása után a Regisztrációgombon keresztül érjük el, azt hogy tudjunk regisztrálni. Regisztrációhoz a felhasználónak névre, telefonszámra, lakcímre, email címre és jelszóra van szüksége.
* Bejelentkezés: Ha már regisztrált a felhasználó, utána a bejelentkezésnél már csak email cimre és a jelszóra lesz szüksége a bejelentkezéshez.
* Elemek böngészése: Bejelentkezés után a FrissKert felületén már lehet az eladók és a terményeik között keresgélni.
* Ajánlott elemek megtekintése: A FrissKert felületén bármilyen ajánlott terményt meg lehet tekinteni.
* Új elemek feltöltése: Új zöldségeket illetve gyümölcsöket lehet feltölteni a weboldalra.
* Kapcsolatfelvétel emailen: Email cim által kommunikálhatnak az eladó és a vásárló egymás között.

A projekt esetében a használati eset diagram a következő:

A diagram of a diagram

Description automatically generated

1.ábra: Use Case Diagram

## 3.2. Rendszerkövetelmények

A hatékony használathoz minden számítógépes szoftvernek bizonyos hardverkomponenseknek vagy szoftvererőforrásoknak van szüksége. Ezeket az előfeltételeket rendszerkövetelményeknek nevezzük.

A rendszerkövetelményeken belül megkülönböztetünk funkcionális és nem funkcionális követelményeket.

### 3.2.1. Funkcionális követelmények

* Az alkalmazás használata regisztrációhoz kötött, amely egy érvényes névvel, telefonszámmal, lakcímmel, email címmel és jelszóval valósítható meg.
* Bejelentkezni a már regisztrált felhasználók tudnak a regisztrációkor megadott email címmel és jelszóval.
* A felhasználók Email címmel vannak azonosítva.
* Adatok tárolása és megjelenítése a weboldal leírása szerint.
* Adatok bevitele és ezek mentése.
* Az alkalmazás lehetőséget ad saját termények eladására/ vásárlására.
* Az alkalmazáson belül az eladók oldalán megtekinthetjük az eladók listáját.

### 

### 3.2.2. Nem funkcionális követelmények

* internet kapcsolat szükséges
* megközelíthetőség, azaz könnyű kezelés a felhasználó számára
* reszponzív felhasználói interfész
* kijelző méret szerint adaptív felhasználói interfész
* hardveres követelmények:
* programozási nyelv: HTML, CSS, JavaScript

## 3.3. Fontosabb műveletek magyarázata

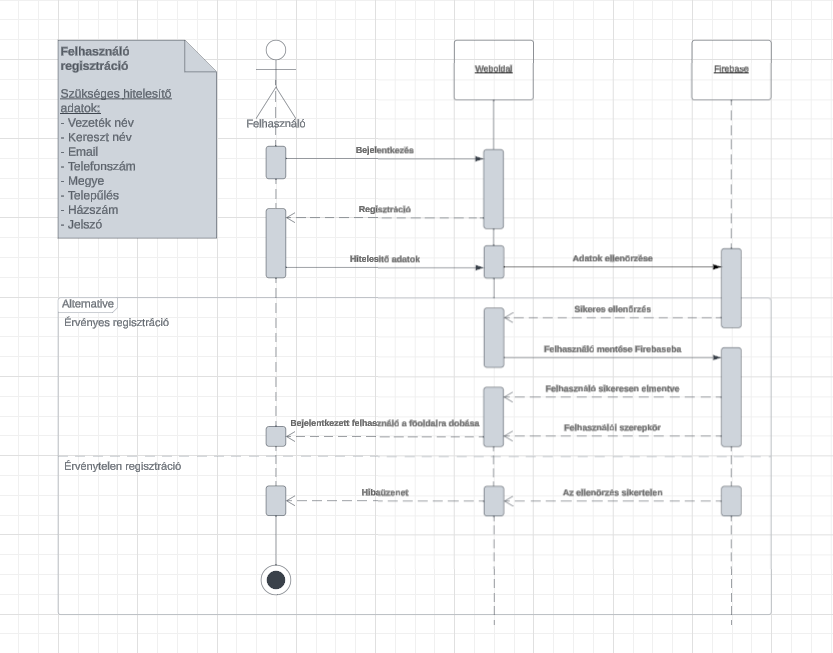
A fontosabb műveletek magyarázatát a szekvencia diagramok segitségével mutatjuk be. A szekvencia diagram feladata objektumok egymás közti üzenetváltásainak ábrázolása egy időtengely mentén elhelyezve.

### 3.3.1. Felhasználó regisztrációja

A alábbi ábrán található a felhasználó regisztrációjának folyamata egy egyszerűsített szekvencia diagram formájában. A FrissKert oldalára be nem jelentkezett felhasználó tud magának késziteni egy profilt a regisztrációhoz szükséges adatok megadásával, amelyek a következők:

* Vezeték név
* Kereszt név
* Email
* Telefonszám
* Lakcím: megye, teletülés, házszám
* Jelszó

Az adatok ellenőrzése után a felhasználó elmentődik az adatbázisunkba,amelyet Firebase-be valósítottunk meg és ez után a felhasználó korlátlanul használhatja a FrissKert weboldalát. Ha nem megfelelő valamelyik mező, akkor a rendszer hibaüzenetet ad vissza és újra kell próbálkozni.



2.ábra: Felhasználó regisztrációja

### 3.3.2. Felhasználó bejelentkezése

Az alábbi ábrán található a felhasználó bejelentkezésének folyamata szintén egy egyszerűsitett szekvencia diagram segítségével bemutatva. Ez hasonlóan működik, mint a fentebb említett felhasználó regisztrációja, annyi hogy a felhasználó már létezik az adatbázisunkban igy nem kell újra regisztrálnia, csak bejelentkeznie.

A bejelentkezés folyamata, hogy a felhasználó megadja a saját email címét és jelszavát. A rendszer a megadott adatokat hitelesíti s majd leellenőrzi azokat. Ez után a rendszer belép a FrissKert főoldalára, ha sikeres az ellenőrzés. Viszont, ha a bejelentkezés sikertelen, akkor a rendszer vissza fog küldeni egy hibaüzenetet.

A diagram of a diagram

Description automatically generated

3.ábra: Felhasználó bejelentkezése

# 4. Tervezés

## 4.1. Wireframe

## 4.2.Architektúra

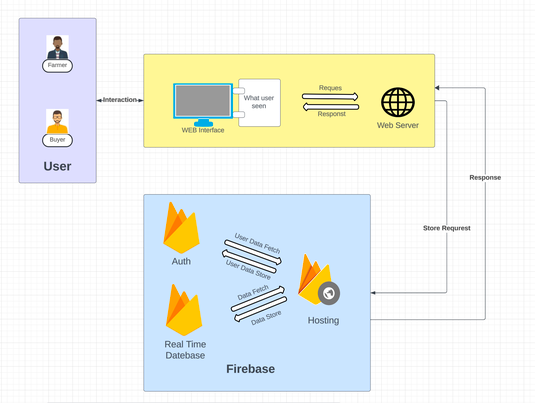
A alábbi ábra a weboldal arhitekturája figyelhető meg. A következő komponensek és ezen komponensek kapcsolataiból tevödik őssze:

* Felhasznalok (Users)
* Böngésző : Web felhasználói felület (Browser user interface)
* Adatbázis

1. Felhasználok  
   A felhasználok egy laptopon vagy asztali gépen található böngésző segítségével kerül interakcioba a webalkalmazással.

A weboldalra két tipusu szerepkörrel rendelkező felhasználó regisztrálhat, valamint jelentkezhet be: vásárlok és eladok. A szerepkörtől, valamint a szándéktól függetlenül a felhasználóknak regisztrálniuk kell és be kell jelentkezniük.

1. Felhasználoi felületek  
   Ez a része a weboldal megjelenitésért felel, más szoval lehetövé teszi a felhasználoknak a weboldal használatát.  
   Ez felelős az adatok megjelenítéséért: megjeleniti az termékeket, a termékek feltöltését.  
   A felhasználói felület a felhasználó által végzett interakciokat küldi az egyes függvényeknek, melyek végrehajtnak egy adott operáciot és visszaadnak egy eredményt a felhasználói felületnek, ahol a felhasználo láthatja az interakciojának az eredményét. Ilyen például a termékek szűrése tipus szerint: gyümölcs vagy zöldség.  
   Ugyanakkor a felhasználói felület magába foglalja az alkalmazás design-ját.
2. Adatbázis   
   Az adatok tárolásáért felel, amelyekre a többi komponenseknek szükségük van, például termékek és felhasználok.  
   Egyiránt lehet adatokat feltölteni, tárolni, valamit tárolt adatokat lekérni.  
   A Firebase Realtime Database az interneten keresztül működik. Amikor a weboldal hasznája a Firebase SDK-t (Software Develoment Kit) a kliensoldali kód kommunikál a Firebase szerverrel az interneten keresztül. Ezáltal az a weboldal valós idöben jeleniti meg az adatbázisban eltárolt adatokat. Ez azt is jelenti hogy folyamatosan frissiti a weboldal állapotát ahogy valami változást észlel az adatbázisban. Iyen változásokat okozhat például új termék feltöltése. Ez különösen fontos esetünkben mivel percenként tölthetnek fel új terméket az eladók.  
   Hasonloképpen a Firebase egy szolgáltatása az Authentication, amely lehetővé tette az ügyfél azonosítás és hitelesítés funkciók használatát.



.ábra: Architektúra diagram

## 4.3. Modulok leírása

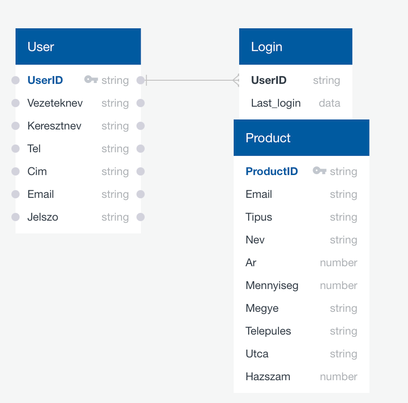
a

### 4.3.1. Adatbázis

Az adatok Firebase-ban vannak eltárolva, a bejelentkezést a Firebase Authentication oldja meg, valamint a többi adat eltárolására Realtime Database-t használtuk.

A Firebase Authentication átveszi a regisztrációs formból az email címet és a jelszót és létre hozza a felhasználó belépési adatait, illetve minden egyes felhasználónak generál egy egyedi UserID-t.

A Realtime Database-be a rendszer eltárolja a regisztrációs formból az adatokat: vezeték név, keresztnév, email, telefonszám, lakcim, viszont kivételt képez a jelszó, mivel azt nem tárolja el.



.ábra: Adatbázis Quick Database Diagram-ban

A fenti ábrán látható User táblázatban tároljuk el az új felhasználóról az adatokat:

* UserID
* Vezeték név,
* Kereszt név,
* Telefonszám,
* Lakcim,
* Email,
* Jelszó,

amelyeket a regisztráció során maga a felhasználó ad meg és amelyek a továbbiakban fontos szerepet játszanak a weboldal működésében.

A Login táblázatban a Last\_login menti el, hogy a felhasználó mikor jelentkezett be utoljára a weboldalra.

A Product táblázatban látható adatmezők a következők:

* egyedi ID: ProductID
* Email
* Tipus (zöldség vagy gyümölcs)
* Név (a termék neve)
* Ár (Lej-ben)
* Mennyiség
* Lakcím: Megye, Település, Utca, Házszám: az a hely, ahol a megtalálható az eladandó termék



.ábra: Adatbázi a Realtime DataBase-be

### 4.3.2. Aktivitás diagramok

#### a. Aktivitás diagram a bejelentkezéshez

Az alábbi ábra az Aktivitás diagram a bejelentkezését mutatja be egy aktivitás diagram segitségével. A weboldal elindítása után a felhasználó megpróbál bejelentkezni a FrissKert oldalára. A rendszer le ellenőrzi, hogy hitelesek-e a felhasználó adatai. Ha helyesek az adatok, akkor a felhasználó sikeresen be tud jelentkezni és ezután már a FrissKert főoldalára viszi át. Viszont, ha nem tud sikeresen bejelentkezni, akkor a rendszer Hibát ad ki a felhasználónak, miután ő vagy újra próbálja a bejelentkezést vagy regisztrálja magát. Regisztrálás során is a rendszer ellenőrzi a beírt adatokat és eldönti róluk, hogy helyesek vagy sem. Ha nem sikeres a regisztráció, akkor a felhasználó újra kell próbálja a regisztrálást. Ha a regisztrálás sikeres, akkor a továbbiakban már könnyzedén bejelentkezhet a felhasználó és használhatja a FrissKert oldalát.

A diagram of a flowchart

Description automatically generated

.ábra: Aktivitás diagram a bejelentkezéshez

#### b. Aktivitás diagram a vásárlás – és eladáshoz

Az alábbi ábra az Aktivitás diagram a termék vásárlás – és eladási lehetőségeit mutatja be. Miután a felhasználó sikeresen bejelentkezett, utána eldöntheti, hogy eladó vagy vásárló szeretne lenni.

Ha vásárló szeretne lenni, akkor az egyik lehetősége az, hogy szabadon böngészhet illetve keresgélhet az eladandó termékek között. Egy másik lehetőség a vevő számára, hogy kiválaszthatja azt, hogy gyümölcs vagy zöldséget szeretne nézni illetve vásárolni. Miután a vevő kiválasztotta a terméket amit venni szeretne, beállitja azt, hogy mennyi mennyiséget szeretne vásárolni (kg-ben). Ezt követően a rendszer értesití az eladót és megtörténik a vásárló rendelésének a visszaigazolása. Ezek után a vevő eldöntheti, hogy az eladandó áruk oldalán folytatja a termékek böngészését vagy kijelentkezik a FrissKert oldaláról.

Ha a felhasználó eladó szeretne lenni és ezáltal a termékeit árusítani, nincs más dolga, minthogy megadja a termékeinek az adatait. Miután az eladandó termék adatai beírásra kerülnek a termék feltöltése lépik érvénybe, és ha az eladó mindent jól csinált, akkor a rendszer vissza küld egy „Sikeres feltöltés” üzenetet. A továbbiakban az eladó is eldöntheti, hogy ott marad a FrissKert főoldalán vagy kijelentkezik.

A diagram of a company

Description automatically generated

.ábra: Aktivitás diagram a vásárlás – és eladáshoz

## 4.4.Menedzselés

Csapat munkánkban a fejlesztés során fontos volt az együttműködés illetve ugyan olyan fontos volt a csapat irányítása valamint a kommunikáció is. Ahhoz, hogy mindezt meg tudjuk valósítani a Github volt a segítségünkre, amely egy kiváló eszköz a feladatok haladásának követésére és a munka szervezésre valamint ezek felosztásra.

## 4.5.Verziókövetés – GitHub - Kanban

A projekt megvalósítása során nagyon fontos szerepet játszik a verziókövetés valamint kódok megosztása. Minderre a GitHub-ot használtuk segítségként. Létrehoztunk egy GitHub repository-t, és azon belül minden task-nak egy külön Branch-et annak érdekében, hogy átláthatóbb legyen külön-külön minden funkció fejlesztése.

# 5. Alkalmazás működése

## 5.1. UI – konkrét megvalósítás

a

# 6. Összegzés

a

## 6.1. További fejlesztési lehetőségek

* Aplikációs megvalósítás
* Kedvencek bejelölése
* Több terméket is be lehet tenni a kosárba
* Termékek értékelése

# 7. Bibliográfia/használati tool-ok

a