



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM
Marosvásárhelyi Kar



PROJEKT DOKUMENTÁCIÓ

Készítette:

Kovács Norbert

Makkai Nándor

(Informatika III. év)

Szoftver rendszerek tervezése

Sapientia EMTE

2020

Tartalomjegyzék

1.	Bevezetés.....	3
2.	Célkitűzések.....	3
3.	Felhasználói követelmények	3
4.	Rendszerkövetelmények	4
4.1	Funkcionális követelmények	4
4.2	Nem funkcionális követelmények	4
5.	Tervezés.....	5
5.1	Architektúra.....	5
5.2	Adatbázis	6
5.3	Rendszer működése	6
5.3.1	Regisztráció.....	6
5.3.2	Bejelentkezés.....	7
5.3.3	Statisztika és profil.....	7
5.3.4	Játék.....	8
6.	Kivitelezés.....	8
7.	Szoftver bemutatása	9
8.	Bibliográfia.....	9

1. Bevezetés

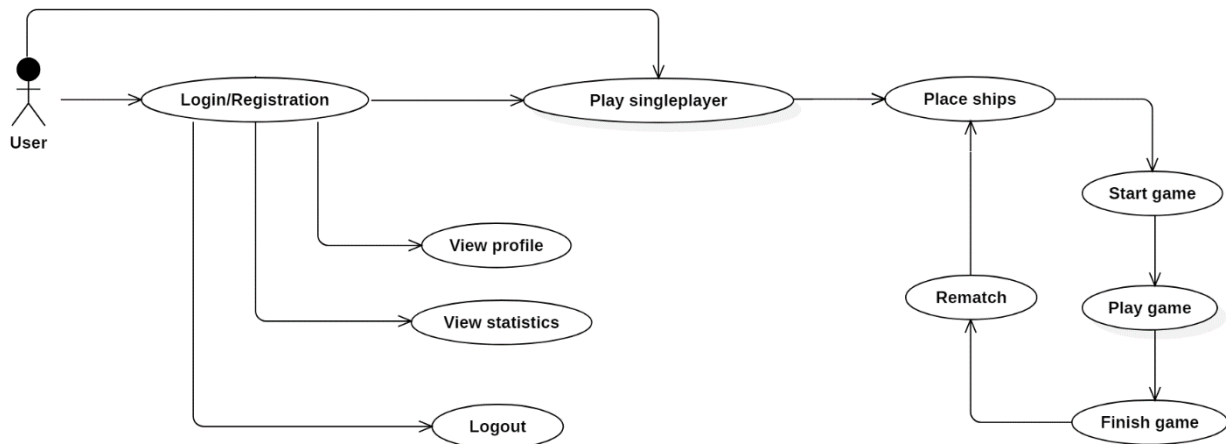
Manapság jobbnál jobb videójátékok készülnek, kiemelkedő grafikával, innovatív megoldásokkal, ezáltal a régi, klasszikus játékok feledésbe merülnek. Biztosan sokan hallottak már a Torpedó nevű játékról. Ez egy klasszikus, stratégiai játék, amely nagyon egyszerű, könnyen megérthető szabályokon alapszik. Adott két 10x10-es rács, amelyen a küzdő felek elhelyezik hajóikat, majd egymás területére lövöldözve próbálják elsüllyeszteni azokat. A játékosok felváltva lőnek a másik játékterére. Az győz, aki hamarabb süllyeszti el az ellenfél hajóit. Ezt a játékot próbáltuk meg kivitelezni egy weboldal keretein belül.

2. Célkitűzések

- Népszerűsíteni a régi, klasszikus Torpedó játékot
- A szoftver funkciói 2-3 kattintáson belül legyenek elérhetők a felhasználó számára
- A képernyő túlszűfolttságának elkerülése érdekében a rendelkezésre álló terület legfeljebb 60 százalékának használata
- Saját tudásunk gyarapítása, általunk eddig ismeretlen keretrendszerek megismerése

3. Felhasználói követelmények

A felhasználók számára egyelőre csak az egyszemélyes játékmód érhető el, ahol a Számítógéppel küzdhetnek meg.



Ábra 1 Use Case Diagram

A felhasználók jogosultak a regisztrációra és a bejelentkezésre. A regisztrációhoz szükséges a *felhasználónév* (minimum 3 karakter, maximum 25 karakter), *e-mail cím* és *jelszó* (minimum 8 karakter), a bejelentkezéshez pedig meg kell adni az *e-mail címet* és a *jelszót*. Két csoportba oszthatjuk a felhasználókat: vendég és bejelentkezett felhasználók.

A vendég felhasználó gond nélkül játszhat regisztráció nélkül, ahogy mások is, azonban ez hátránnyal jár, mert az adatai a játékaikról nem mentődnek el, nem tud a játékosokról rangsort és a játékaikról statisztikát megtekinteni.

A bejelentkezett felhasználók játékaik elmentésére kerülnek a vendég felhasználókkal szemben. Az ebből készült statisztikákat, valamint a játékosok rangsorát könnyedén elérheti. Rendelkezik egy profillal, ahol szemügyre veheti a személyes adatait, pontszámait.

4. Rendszerkövetelmények

4.1 Funkcionális követelmények

Úgy a vendég, mint a bejelentkezett felhasználó kezdőoldalán található egy játék elindító gomb, amely megnyomásával elindul a játék első szakasza, amikor a felhasználó elhelyezi a hajóit a játéktéren. Lehetősége van a hajók elforgatására egy gomb segítségével. Miután elhelyezte az összes hajót, megnyomhatja az indító gombot, ezzel elindul a játék második szakasza. A felhasználó számára láthatóvá válik a saját és az ellenfél játéktere, valamint egy játéközenetet megjelenítő doboz. A játékot a felhasználó kezdi, ezután felváltva támadhatják egymást, azonban amikor valakinek találatja van, akkor ismét ő következik. A program folyamatosan értesíti a felhasználót üzenetekkel: ki lőtt, hova lőtt és talált vagy sem. A játék akkor ér véget, amikor az egyik fél kilövi a másik összes hajóját, ezzel megnyeri a játékot. A játék végét egy felugró ablak jelzi, amelyen megjelenik a csata végkimenetele, valamint a bejelentkezett felhasználók számára a frissen szerzett pontszámok. Lehetőség adódik visszavágóra vagy visszatérésre a kezdőoldalra.

Mindkét felhasználó esetében megtalálható egy menüsáv, ahol a bal oldalon egy logó helyezkedik el, amelyre kattintva megjelenik a főoldal. A menüsáv jobb oldalán egyéb funkciók találhatók attól függően, hogy a felhasználó be van jelentkezve vagy sem. A vendég felhasználó esetében van egy bejelentkezés és regisztráció funkció. A bejelentkezett felhasználók számára három funkció áll rendelkezésükre: profil, statisztika megtekintése, valamint a kijelentkezés.

4.2 Nem funkcionális követelmények

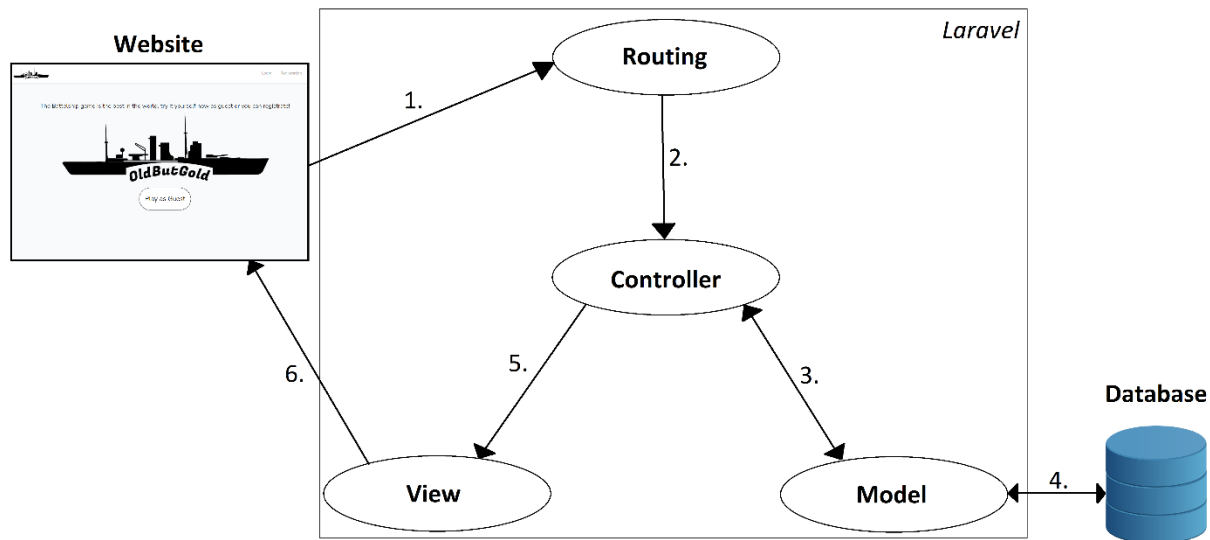
A weboldalt Laravelben készítettük, ami egy PHP keretrendszer, a kliensoldal elkészítéséhez HTML, CSS, JavaScript nyelvet használtunk. Az egyszerű JavaScript mellett használtunk Node.js-t, Vue komponenseket, a statisztikák elkészítéséhez pedig CanvasJS-t.

A programozás során próbáltunk megfelelni a kódolási szabványoknak. A PHP esetében a Laravel saját kódolási szabványát vettük figyelembe. A HTML és CSS nyelveknél kötőjellel elválasztott elnevezéseket használtunk, odafigyeltünk a tördelésekre, azért hogy minél átláthatóbb legyen a kód. A JavaScript használata közben igyekeztünk minden szabályt betartani, például: „camelCase” elnevezési mód, sorvégi pontos vessző, zárójellezés.

5. Tervezés

A projekt tervezését azzal kezdtük, hogy megbeszéltük a szoftver funkcióit, ezt követően készítettünk egy kezdetleges látványtervet és a megfelelő diagramokat. Ezek után megbeszéltük mindenkinek a feladatkörét, hogy ki milyen feladattal fog foglalkozni. Kitűztük a határidőket és elkezdtük a szoftver implementálását.

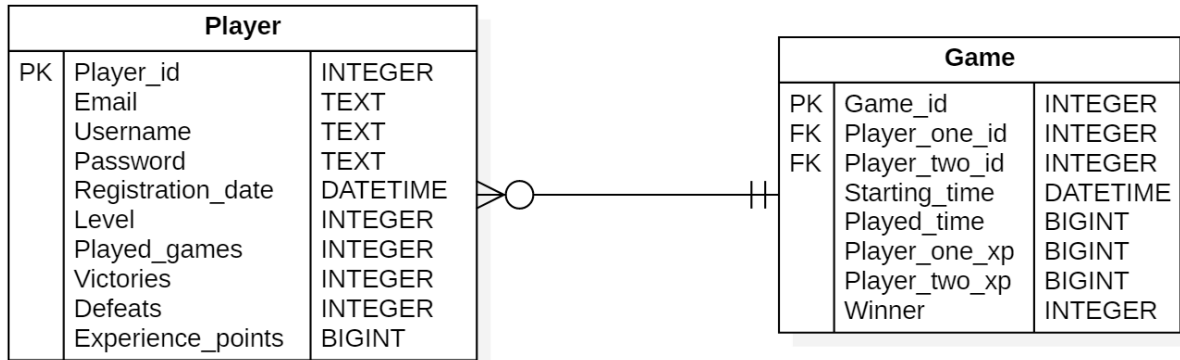
5.1 Architektúra



Ábra 2 Rendszer architektúrája

1. Felhasználói kérés küldése a szervernek
2. A megfelelő controller kiválasztása, amely le fogja kezelni a kérést
3. Együttműködés az adatmodellel
4. Adatbázis kapcsolat
5. A controller meghívja a nézetet a megfelelő adatokkal
6. A nézet megjelenítése a böngészőben

5.2 Adatbázis



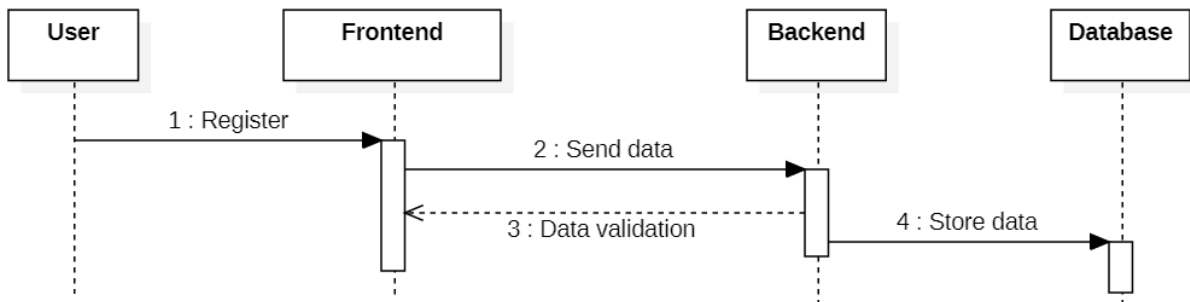
Ábra 3 Adatbázis szerkezete

A szoftver az adatok tárolására SQLite adatbázist használ. Azért választottuk az SQLite adatbázist, mert kis erőforrásigénnyel rendelkezik, képes a megfelelő szolgáltatások biztosítására, egyszerűen hordozható, mert az egész adatbázist egyetlen fájlban tárolja. Az adatokat két táblába szerveztük: Player (Felhasználó) és Game (Játék). A két tábla között egy a sokhoz kapcsolat van: egy felhasználóhoz több játék is tartozhat.

5.3 Rendszer működése

5.3.1 Regisztráció

sd OldButGold usage



Ábra 4 Szekvencia diagram, regisztráció

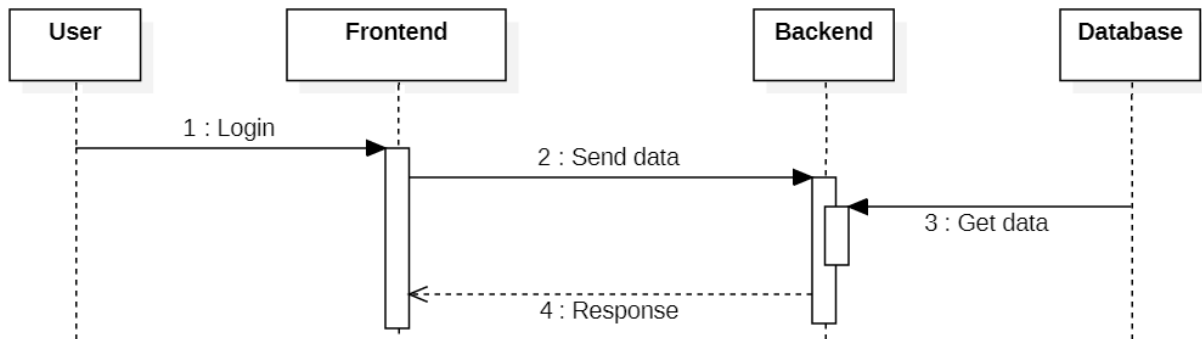
A felhasználó a regisztráció során megadja szükséges adatait, amelyek a szerveroldalon lesznek leellenőrizve:

- Felhasználónév – minimum 3 karakter, maximum 25 karakter, tartalmazhat bármilyen karaktert
- E-mail cím – helyes formátuma kell legyen
- Jelszó – minimum 8 karakter, nincsenek kötelező karakterek
- Jelszó megerősítése

Ha a szerver valami problémát észlel, akkor jelzi a felhasználónak. Ellenkező esetben eltárolja a felhasználó adatait és automatikusan bejelentkezik.

5.3.2 Bejelentkezés

sd OldButGold usage



Ábra 5 Szekvencia diagram, bejelentkezés

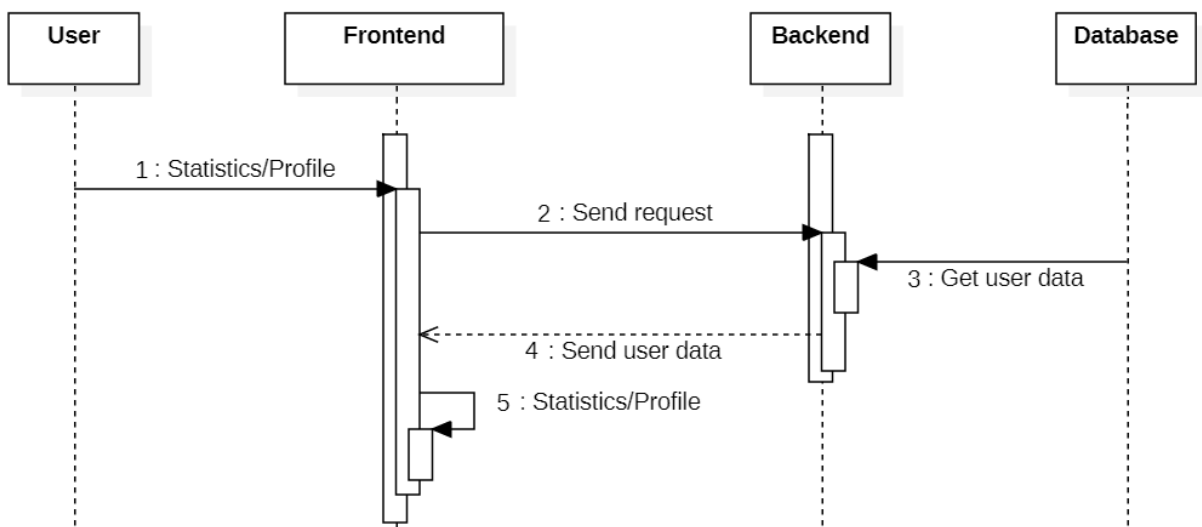
A bejelentkezéshez a felhasználónak meg kell adnia a szükséges adatokat:

- E-mail cím
- Jelszó

A megadott adatok által a szerver leellenőrzi, hogy létezik-e az adatbázisban az adott felhasználó, ha nem, akkor hibaüzenetet küld, más esetben bejelentkezteti a felhasználót.

5.3.3 Statisztika és profil

sd OldButGold usage

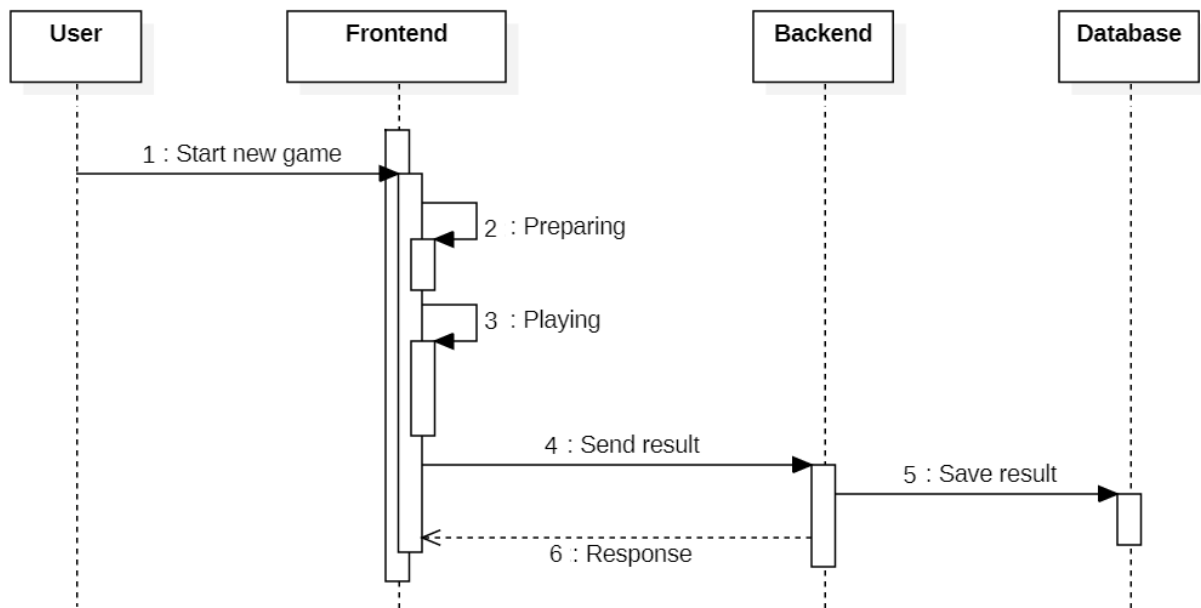


Ábra 6 Szekvencia diagram, statisztika és profil

A statisztika és profil megtekintéséhez a szerver lekéri a felhasználó adatait az adatbázisból, ezeket visszaküldi a kliensoldalra, amely elvégzi az adatok megjelenítését a megfelelő módon.

5.3.4 Játék

sd OldButGold usage

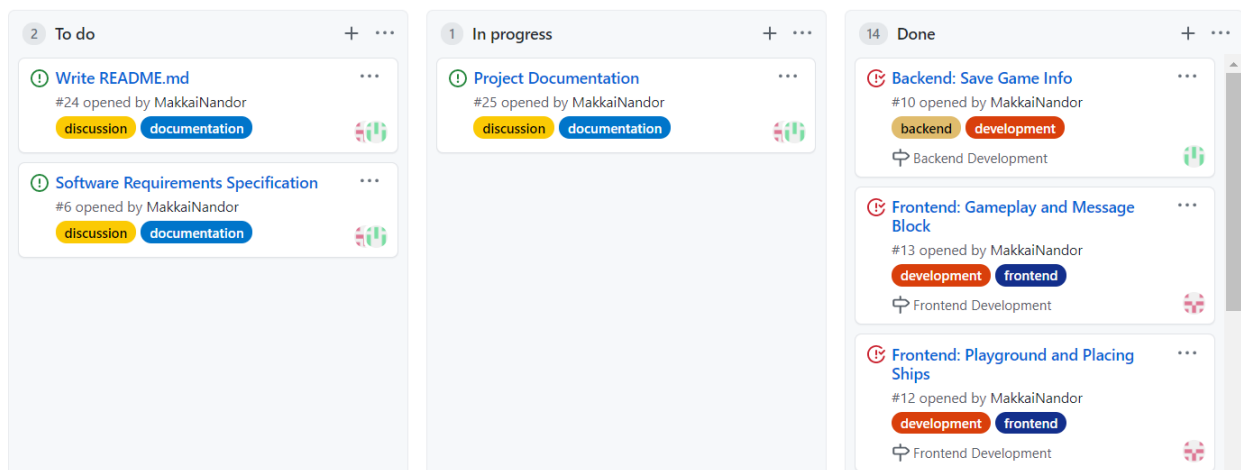


Ábra 7 Szekvencia diagram, játék

A játék kezdete után kliensoldalon elindul játékelőkészítés szakasz, ezt követi maga a játék menete. A játék végén elküldésre kerülnek az adatok, amelyeket a szerveroldal elment az adatbázisban.

6. Kivitelezés

Mielőtt belekezdünk volna a weboldal létrehozásába, létrehoztunk egy GitHub repository-t, amihez hozzáadtuk a projektben résztvevő személyeket. Ezen belül készítettünk egy projektet, ahol nyomon tudtuk követni az elvégzendő, a folyamatban lévő, illetve a befejezett feladatainkat.



Ábra 8 GitHub Kanban board

7. Szoftver bemutatása

8. Bibliográfia

<https://www.youtube.com/watch?v=ImtZ5yENzgE>

<https://www.w3schools.com/>

<https://stackoverflow.com/>

<https://laravel.com/docs/8.x>