## Bevezetés

Manapság nagyon elterjedtek a hangalapú kommunikációs szoftverek. A szakdolgozatom témáját abból az okból választottam, mivel sokat játszom számítógépes játékokkal online és nincsen olyan voice chat alkalmazás, ami megbízható lenne. A szoftvert leginkább a játékos közösségnek és azoknak a családoknak tervezem , akik nem tudnak gyakran beszélni egymással. A szoftverem kialakításánál az egyszerűségre és a user interface design-nál a legjobb átláthatóságra fogok törekedni. A szoftvert C# programozási nyelven fogom megvalósítani, a klienst windows formos alkalmazásban és a szervert pedig console-os formában. A szerver adminok a consoles-os ablakba beírt parancsokkal is tudják majd a szervert használókat koordinálni. A szoftvert csak Windows operációs rendszer alatt lehet futtatni. A szoftverhez használni fogom a NAudio könyvtárat, amivel a hanghoz köthető funkciókat fogom kialakítani. A szerver és a kliens között TCP protokoll fogja kialakítani a kapcsolatot, ezek után az UDP protokoll veszi át a szerepet az aszinkron hang és szöveg küldésénél. A hang és szöveg küldésére, azért fogok UDP protokollt használni, mert az UDP egészen közel van a valós idejű adattovábbításhoz.

## Specifikációk

**Az alkalmazással szemben a következő funkciók várhatók el:**

* csoportos hang chat
* csoportos szöveges chat
* privát chat
* felhasználói profil teljes testreszabása
  1. megjelenítendő név
  2. mikrofon hangereje
  3. kimeneti hang hangereje
  4. mikrofon állapota(némítva, nyomógombos, hang érzékelés)
* admin parancsok(kirúgás, kitíltás, felhasználó némítása)

Szerintem ezek az alapvető funkciók egy voice chat alkalmazáshoz, hogy megfelelően, mindenki számára élvezhetően működjön. A hanggal kapcsolatos funkciókat a NAudio könyvtár használatával fogom megvalósítani.

### Használati eset diagram

A használati eset vagy use case diagramot a felhasználók számára elérhető funkciókat mutatja meg. A használati eset diagrammokat a szoftverfejlesztés azon fázisában használjuk, amikor az ügyfél meghatározza a specifikációkat és ezekből az ügyfél számára is érthető ábrát hozunk létre.



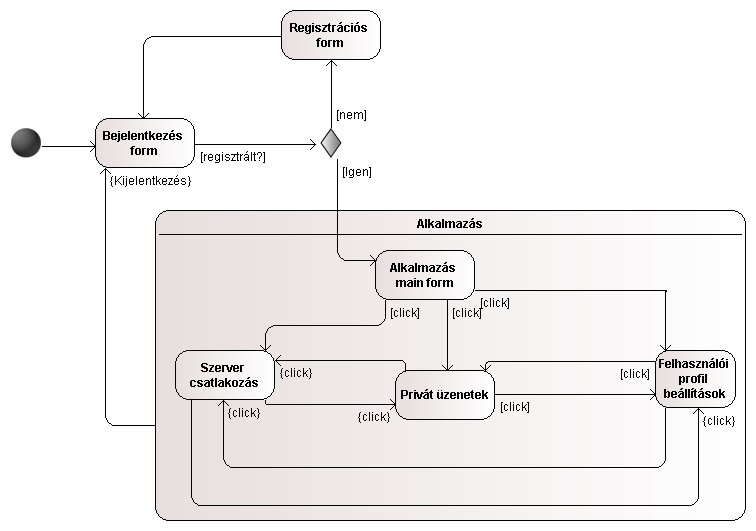
**Aktorok:**

* Átlagos felhasználó
* Admin

Az ábrán a specifikációk cím alatt meghatározott funciók láthatók. Minden funkció eléréséhez bejelentkezés szükséges. Az Admin actor leszármazik az Átlagos felhasználóból, ami azt jelenti hogy az Admin minden funkciót tudni fog amit az Átlagos felhasználó is tud, de viszont az átlagos felhasználó nem éri el az admin által használható funkciókat.

### Állapotgép Diagram

**Az első állapotgép diagrammon a** formok közötti váltásokat mutatja be. Az egész szoftver a bejelentkezés form-nál kezdődik el, azután ha a felhasználó még nem regisztrált, akkor a szoftver a regisztrációs form-ra irányítja át. Ha a felhasználó már regsiztrált akkor bejelentkezés után az összes funkciót eléri a main formból a bal oldalra kihelyezett gombok megnyomásával. A main formon a kijelentkezés gombra kattintva, újra a bejelentkező form lesz az aktív.



### Szekvencia diagram

A szekvencia diagram a kliens, a szerver és az adatbázis közötti üzeneteket fogja szemléltetni.

### Class diagram