

Hamal

A jövő reménye

Kiss Benedek, Huszka Adrián, Nagy Roland

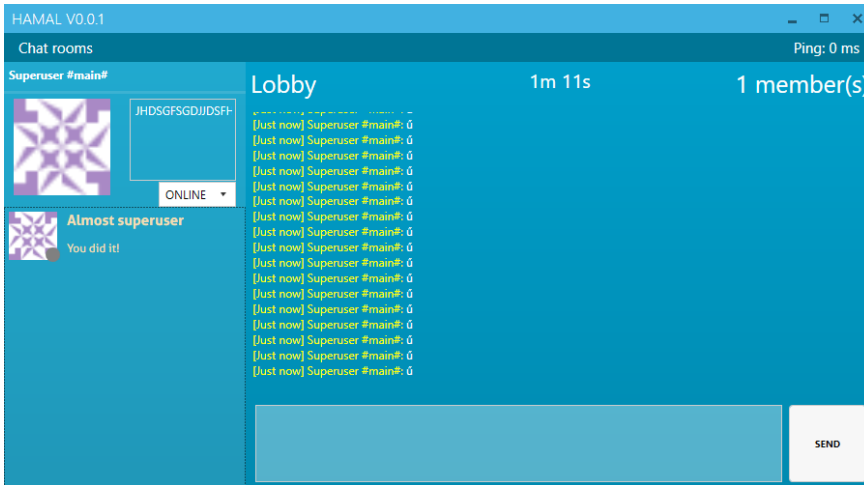
Felkészítő tanár: Vári Ágnes

*Hódmezővásárhelyi SZC Pollák Antal Szakképző Iskolája, 6600 Szentés
Apponyi tér 1.*

1. Bevezetés

A Hamal egy valós időben történő kommunikációs alkalmazás, amelyben képesek vagyunk barátainkkal, ismerőseinkkel beszélgetni, sőt csoportot is létre tudunk hozni, ahol képesek vagyunk többen kommunikálni. A Hamal beszerzése után nincs más dolgunk, mint regisztrálni az erre fejlesztett oldalon, ahol egy felhasználónevet, egy email címet, egy valódi nevet és egy jelszót kell megadnunk. Miután ezt megtettük, nincs más dolgunk, mint belépni és élvezni a Hamal által nyújtott lehetőségeket. A program Microsoft Visual Studio 2013-ban íródott, C# nyelven. A forráskód teljes mértékben objektumorientált. A megadott adatok egy tömörített bináris ini fájlban mentődnek el (melynek kiterjesztése hdf = **H**amal**D**ata**F**ile), szerveroldalon. A bejelentkezés során, képesek vagyunk menteni az adatainkat, hogy később ne keljen azzal foglalkozni, hogy beírni a felhasználónevet és jelszót. Be tudjuk állítani, hogy automatikusan léptessen be minket a program. Miután beléptünk, azonnal a chat ablak kerül a szemünk elé.

1. ábra. A bejelentkezési felület



2. ábra. A főoldal

2. Problémák

A programozás során sokszor előfordult az, hogy rossz csomag lett elküldve a szervernek, ami azt eredményezte, hogy összeomlott a program. A másik gyakori hiba a UUID megjelenítésével volt.

```

if (!File.Exists("Users\\0.hdf"))
{
    DataFile t_File = new DataFile("Users\\0.hdf");
    t_File.AddEntry("USER_DATA", "DisplayName", Encoding.UTF8.GetBytes("Superuser #main#"));
    t_File.AddEntry("USER_DATA", "UUID", new UUID().ToBytes());
    t_File.AddEntry("USER_DATA", "Status", new Byte[] { (Byte)UserStatus.OFFLINE });
    t_File.AddEntry("USER_DATA", "StatusMessage", Encoding.UTF8.GetBytes("Just do it!"));
    t_File.AddEntry("USER_DATA", "DatabaseID", BitConverter.GetBytes((Int64)0));
    t_File.AddEntry("USER_DATA", "ProfilePicture", File.ReadAllBytes("D:\\programming\\5fd80d98f96f9adbde7876cbda181715.png"));
    t_File.AddEntry("USER_DATA", "Contacts", Helper.Int64Array(new Int64[] { 1 }));
    t_File.AddEntry("USER_LOGIN_DATA", "Username", Encoding.UTF8.GetBytes("admin"));
    t_File.AddEntry("USER_LOGIN_DATA", "Password", Encoding.UTF8.GetBytes("1234"));
    t_File.Save();
}

if (!File.Exists("Users\\1.hdf"))
{
    DataFile t_File = new DataFile("Users\\1.hdf");
    t_File.AddEntry("USER_DATA", "DisplayName", Encoding.UTF8.GetBytes("Almost superuser"));
    t_File.AddEntry("USER_DATA", "UUID", new UUID().ToBytes());
    t_File.AddEntry("USER_DATA", "Status", new Byte[] { (Byte)UserStatus.OFFLINE });
    t_File.AddEntry("USER_DATA", "StatusMessage", Encoding.UTF8.GetBytes("You did it!"));
    t_File.AddEntry("USER_DATA", "DatabaseID", BitConverter.GetBytes((Int64)1));
    t_File.AddEntry("USER_DATA", "ProfilePicture", File.ReadAllBytes("D:\\programming\\5fd80d98f96f9adbde7876cbda181715.png"));
    t_File.AddEntry("USER_DATA", "Contacts", Helper.Int64Array(new Int64[] { 0 }));
    t_File.AddEntry("USER_LOGIN_DATA", "Username", Encoding.UTF8.GetBytes("admin2"));
    t_File.AddEntry("USER_LOGIN_DATA", "Password", Encoding.UTF8.GetBytes("1234"));
    t_File.Save();
}

```

3. ábra. A teszt admin felhasználók létrehozása

2.1. Szerver rész

A szerver minden egyes művelet megjelenít egy konzolban, amelynek a segítségével hibára tehetünk szert és nyomon követhetjük a felhasználók tevékenységeit. A szerver hozza létre a bináris ini fájlokat, amelyben a felhasználók és a szobák adatait tároljuk.

2.2. Kliens rész

A kliens rész a műveletekért felelős. Ha például küldünk egy üzenet, a kliens küld egy parancsot a JSON Packet-nek. A parancs mellett a szükséges adatok is elküldésre kerülnek (ilyen a felhasználó azonosítója, vagy a szoba azonosítója). Miután elküldtük a parancsot és a szükséges adato(ka)t, a JSON végrehajtja azt a függvényt, amelyre a parancs hivatkozik.

2.3. Helper fájl

A szerver is és a kliens is rendelkezik ezzel az úgynevezett helper fájjal. A helper fájl az alapvető metódusokat tartalmazza, mint például a karakterkonvertálások, a dátum megformázása, alap fénykép betöltése (későbbiekben lesz lehetőség fényképet feltölteni)

```
public static String BeautyDate(DateTime a_DateTime)
{
    DateTime t_Now = DateTime.Now;

    if (t_Now.Day - 1 == a_DateTime.Day)
        return "Yesterday at " + a_DateTime.ToShortTimeString();

    if (t_Now.Day == a_DateTime.Day)
    {
        if (t_Now.Hour == a_DateTime.Hour)
        {
            if (t_Now.Minute == a_DateTime.Minute)
                return "Just now";

            TimeSpan t_Diff = t_Now - a_DateTime;

            return t_Diff.Minutes.ToString() + (t_Diff.Minutes == 1 ? " minute ago" : " minutes ago");
        }

        return "Today at " + a_DateTime.ToShortTimeString();
    }

    if (t_Now.Day + 1 == a_DateTime.Day)
        return "Tomorrow at " + a_DateTime.ToShortTimeString();

    return String.Format("{0}-{1}-{2}. {3}", a_DateTime.Year, a_DateTime.Month, a_DateTime.Day, a_DateTime.ToShortTimeString());
}
```

4. ábra. A dátum formázása

3. Elért eredmények

A program aránylag stabilan működik, stabilabban, mint a Skype. Osztályszinten működik a program és nagy lelkesedéssel használják a tanulók, mint bármi más ami IRC alapon működik, a program JSON sztringekkel kommunikál a klienssel. A fejlesztés közben a Bitbucket nevezetű oldalon tudtuk egymással megosztani a változtatásainkat és ötleteinket. A fejlesztés során többször is kikértük tanáraink véleményét, amely azt eredményezte, hogy hatékonyabb és letisztultabb forráskódot tudtunk írni.