**1. Beadandó feladat dokumentáció**

**Készítette:**

**Molnár Dániel Bence (PWO08K)**

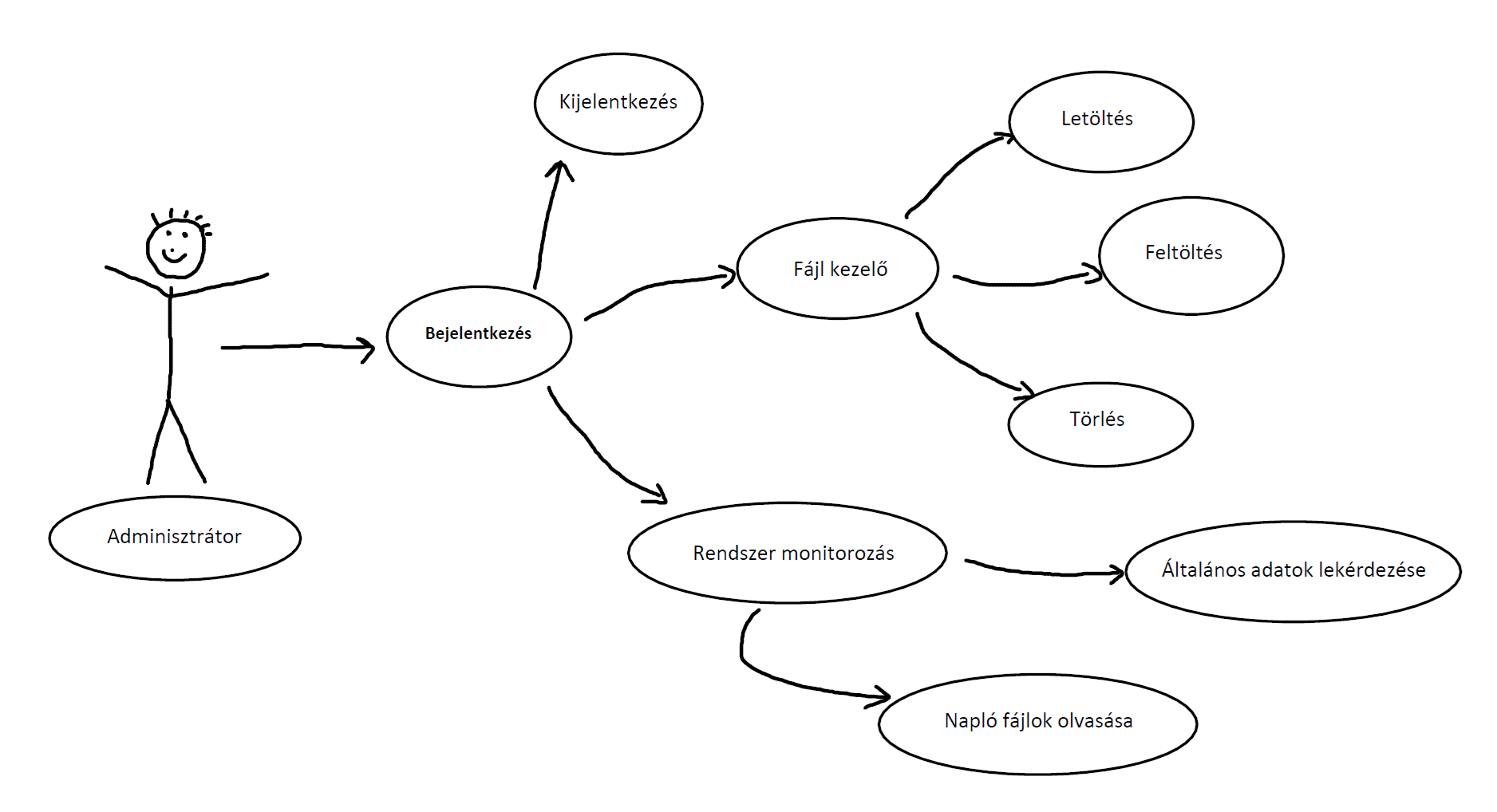
E-mail: **pwo08k@inf.elte.hu**

**Feladat:**

A hétköznapok során sokat foglalkozok a linux szerverem karbantartásával és fejlesztésével. Ezzel kapcsolatban jött az ötlet, hogy készítek egy olyan GUI felületet, amivel távolról tudom kezelni a szerveremet. Az általam készített alkalmazással könnyedén lehet feltölteni/letölteni/törölni fájlt a szerverről, illetve monitorozási lehetőség is van benne. A program segítségével megkönnyítem a mindennapi apróbb feladatok elvégzését.

**Elemzés:**

* Bejelentkezési lehetőség
* Könnyed fájl kezelési műveletek elvégzése, mint például fájl feltöltése, törlése és letöltés
* Monitorozási lehetőség
* Szerver napló olvasása



. ábra

**Tervezés:**

* Programszerkezet:
  + A programot háromrétegű architektúrában valósítom meg. A megjelenítés a gyökér mappában található ’Form.design’ megvalósításban, a modell a ’model’ névtérben, míg a perzisztencia a ’persistence’ névtérben helyezkedik el. A program csomagszerkezete a 2. ábrán látható.

2. ábra


2. ábra

* Perzisztencia:
  + Ebben a névtérben tárolok kettő osztályt ezek rendre: **ApplicationStateManager**, **ApplicationState**. Az utóbbi a kezelőhöz tartozó modell míg az első egy singleton tervezési mintával elkészített belső memória.
* Modell:
  + A modell három részből áll, a felhasználó adatainak eltárolására létrehoztam egy **User.model.cs** állományt, míg a rendszer fájl kezeléshez létrehoztam egy **FileTableDataModel.cs** fájlt. Végezetül pedig létrehoztam egy struktúrát a rendszer adatainak eltárolására, aminek a neve **SystemHealthModel.cs**
* Nézet:
  + A nézetet a **Form1** osztály biztosítja, amely tárolja a modellek példányait, valamint az adatelérés konkrét példányát **ApplicationStateManager.**
  + Nézet felépítése:

**Bejelentkezés nélkül**

A computer and keyboard with a sign in

Description automatically generated

3. ábra

**Bejelentkezést követően (fájl kezelő)**

A screenshot of a computer

Description automatically generated

4. ábra

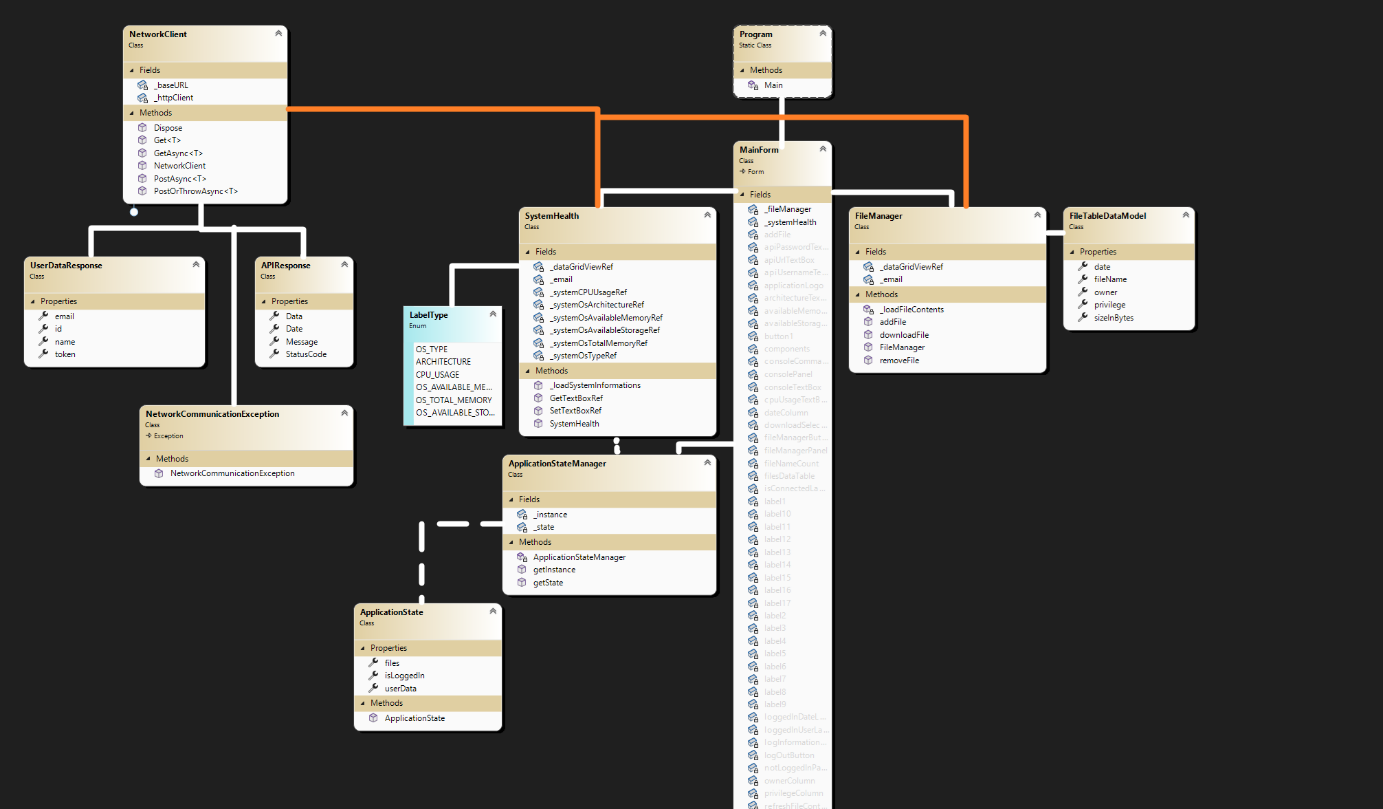
**Rendszer monitorozás**

A screenshot of a computer

Description automatically generated

5. ábra

* Program teljes szerkezete 6. ábrán látható.



6. ábra

**Tesztelés:**

* A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a **UnitTest1** osztályban.
* Az alábbi tesztesetek kerültek megvalósításra:
  + **AuthenticationTest**: Előre megadott belépési adatokkal tesztelem azt, hogy az adminisztátor betud-e lépni
  + **UserDataConvertTest:** A belépést követően visszakapom az adminisztrátor adatait egy speciális JSON formátumban, amit én ‘átalakítok’ az általam értelmezett/használt formátummá és ennek a helyességét tesztelem.
  + **GetFilesFromServer:** Ebben a tesztesetben már belépett felhasználó próbálja meg elérni a szerveren lévő fájlokat.
  + **GetSystemInfos:** Tesztelem a szerver elérhetőségét, az elért szerver adatait és napló fájljait próbálom feldolgozni.
  + **CatchInvalidRequest:** Az utolsó tesztben megpróbálom rekonstruálni azt az esetet amikor a kérés hibára fut az adminisztrátor és a szerver között.