

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM  
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR

# Felhős adattárolás Raspberry Pi támogatással

Önálló laboratórium dolgozat

**Készítette:**

Kecskés Krisztián  
Seres Márk Dániel

**Konzulens:**

Sipos Márton  
doktorandusz

## 1. fejezet

# Bevezető

Ide valami bevezető szöveg jön :)

## 2. fejezet

# Előzetes specifikáció

A tervezés első lépéseként a programtól elvárt funkcionalitást kell megfogalmaznunk valamilyen formában. Erre a legmegfelelőbb módszer a természetes nyelven megfogalmazott use-case-ek használata. Ennek megfelelően a program használatát bemutató, néhány mondatos helyzeteket írtunk. Ezek alapján a program egyes funkciói, és azok alapján annak belső szerkezete már jobban körvonalazhatóvá válik.

### 2.1. Use-case-ek

A következő use-case-ek a program felhasználásnak összes lehetséges esetét fedik le.

**Feltöltés, otthoni hálózat.** A felhasználó fájlokat akar feltölteni az otthoni hálózatra csatlakozva. A kliens ezeket átküldi a Raspberry Pi-on futó szerverre. A Raspberry Pi ezután ezeket a fájlokat kódolja, feldarabolja és feltölti a különböző felhő alapú tárhelyekre, figyelembe véve azok sebességét is.

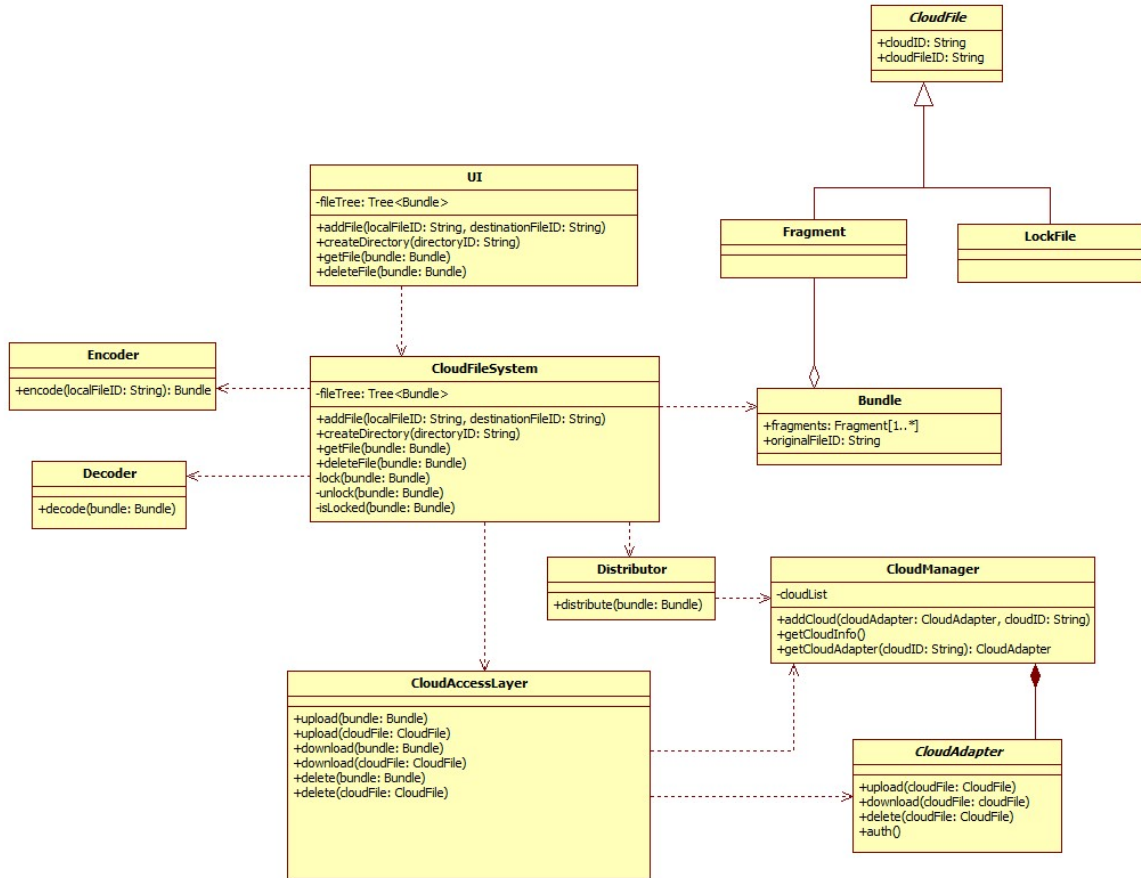
**Feltöltés, nem otthoni hálózat.** A felhasználó a fájljait ezúttal nem az otthoni hálózaton kívánja feltölteni. Ekkor lehetősége van arra, hogy akárhol végrehajtsa a feltöltést, és ilyenkor a kódolást, felhőbe való feltöltést a kliens végzi.

**Letöltés, otthoni hálózat.** A felhasználó az otthoni hálózatra csatlakozva el akar érni egy fájlt a felhő alapú klienssel. A kliens először ellenőrzi, hogy a helyi (kisebb) gyorsítótárában szerepel-e ez a fájl, ha nem, akkor szól a Raspberry Pi-nak, ami eztán ellenőrzi, hogy az ő (nagyobb) gyorsítótárában szerepel-e az, ha pedig itt sem található, akkor a felhőből összeszedi a szükséges fájl darabjait, dekódolja azt, és a fájlt átküldi a felhasználó által használt eszköz kliensének. A fájlok a gyorsítótárakban addig maradnak, amíg a gyorsítótár számára fenntartott hely meg nem telik, eztán a legkevésbé használt fájlok törlődnek az újabbaknak átadva a helyet.

**Letöltés, nem otthoni hálózat.** A felhasználó nem az otthoni hálózatról akar elérni egy fájlt. Ekkor a kliens ellenőrzi, hogy a helyi gyorsítótárában szerepel-e ez a fájl, ha pedig nem találja, akkor a felhőből összeszedi a szükséges fájl darabjait és dekódolja azt.

**Fotó feltöltés, otthoni hálózat, mobil kliens.** Mobil kliens esetén lehetőség van arra, hogy a készülék az otthoni hálózatra csatlakozva a fotók automatikus feltöltését elvégezze, és azokat törölje a készülékről, ezzel helyet szabadítva fel.

**Fotó feltöltés, nem otthoni hálózat, mobil kliens.** A fotók előző módon leírt feltöltésére egyéb hálózatokon is van lehetőség, de ez csak manuálisan indul el, hogy ne terhelje feleslegesen a felhasználó mobil készülékét, és ne termeljen felesleges adatforgalmat mobil internethálózaton.



2.1. ábra. Kezdeti UML osztálydiagram

## 2.2. Osztálydiagram

Az előző pontban leírt use-case-ek alapján már fel tudunk vázolni egy kezdetleges UML osztálydiagramot. Ez látható a 2.1. ábrán.