

1. **Felhasználói specifikáció**

**2. Tesztterv**

**2.1 Áttekintés (Overview)**

**2.2 Bugok “kezelése” (Bug Triage)**

**2.3 Zárási kritérium és folytatási követelmények (Suspension Criteria and Resumption Requirements)**

**3. Követelmény modell**

**3.1 követelmények**

* 1. **tesztesetek (átadási teszteléshez)**

**3.3 követelmény követhetőségi mátrix**

**4. Funkcionális specifikáció**

**4.1 UC diagram**

**4.2 UC táblázat**

**4.3 teszt esetek (rendszer teszteléshez)**

**5. Architektúra tervezés**

**5.1 Komponens diagram**

**5.2 tesztesetek (integrációs teszteléshez)**

**6. Interfész tervezés**

**6.1 Szekvencia diagram**

**6.2 tesztesetek (interfész teszteléshez)**

**7. Modul / komponens tervezés**

**7.1 Osztály diagram**

**7.2 tesztesetek**

**8. Implementáció**

**8.1 C++**

**9. Tesztelés / javítás / újra tesztelés**

**9.1 Tesztnapló**

**10. Felhasználói dokumentáció**

1. **Felhasználói specifikáció**

A projekt során a cél egy olyan felhő alapú vezérlőrendszer fejlesztése, amely képes automatikusan szabályozni az okos üvegházak hőmérsékletét és páratartalmát a fűtés és a locsoló berendezések vezérlésével.

A rendszer képes egyszerre több üvegház kezelésére, amelyeknél a termesztett növény igényeinek függvényében előre definiálva van az elvárt hőmérséklet és minimális páratartalom.

A rendszer a telepített szenzorok által mért adatokat a felhőn keresztül kéri le, amelyek alapján képes önállóan döntést hozni, hogy szükséges-e valamilyen beavatkozás.

1. **Tesztterv**

**2.1 Áttekintés (Overview)**

The test methodology selected for the project:

● Iterative

Ennek az oka, hogy kisebb iterációkban jobban tudunk haladni a tesztekkel, illetve, mivel nem konzultálunk a “megrendelővel” ezt a modellt szeretnénk alkalmazni.

**Unit Teszt:** A szoftver elemeit, osztályait (modulok) külön-külön, minden metódusra tesztelünk, különböző teszt esetekre.

**Integrációs Teszt:** A szoftver elemeit egy csoportba rendezve teszteljük (a teljes szoftver tesztje)

**2.2 Bugok “kezelése” (Bug Triage)**

● Az egyes hibák felkutatása, és típus meghatározása

● A hibák fontossági sorrendjébe rendezése

● A hibák kijavítása

**2.3 Zárási kritérium és folytatási követelmények (Suspension Criteria and Resumption Requirements)**

● Minden teszteset végrehajtásra került

● Utasítás lefedés legalább 95%

● Végrehajtási út lefedés legalább 100%

● Elágazás lefedés legalább 100%

● Minden hiba rögzítésre került a rendszerben

● Minden súlyos hiba kijavításra került

● Tesztjelentés a célszemélyeknek átadásra került

**2.4 Teszt Teljesség (Test Completeness)**

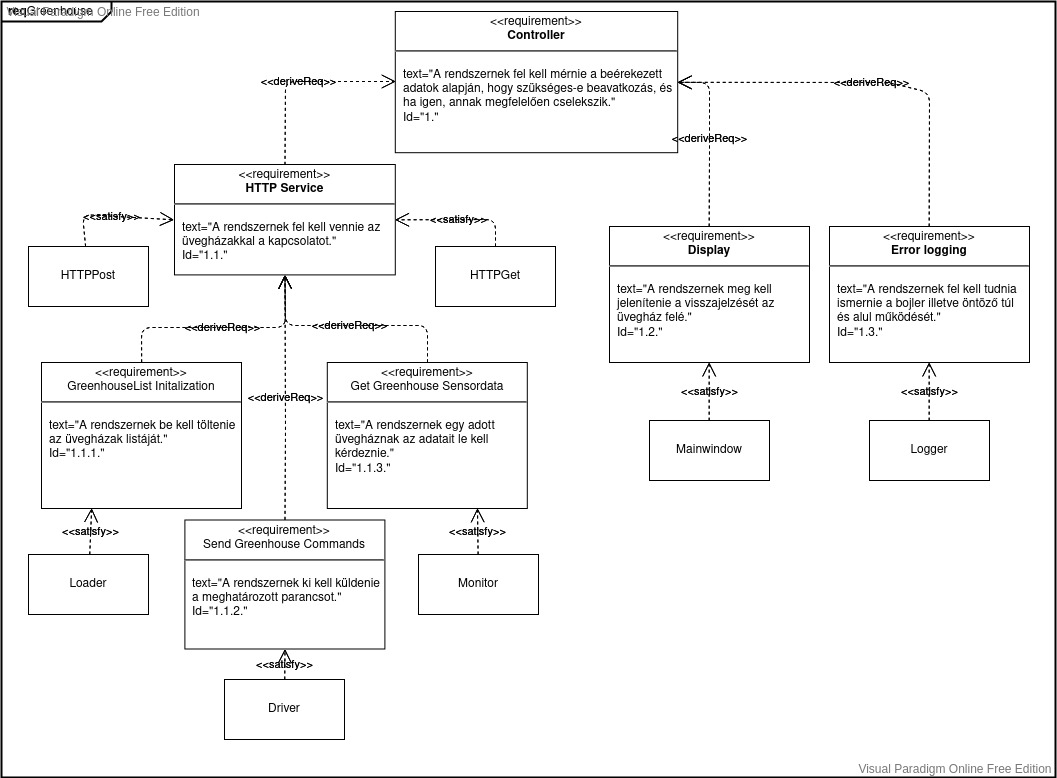
● Tesztek sikeressége 100%

● Minden teszteset végrehajtásra került

● Minden súlyos hiba kijavításra került

1. **Követelmény modell**

**3.1 követelmények**

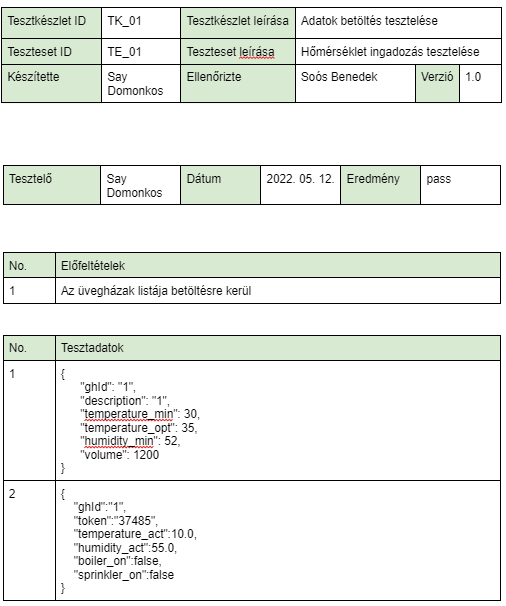
****

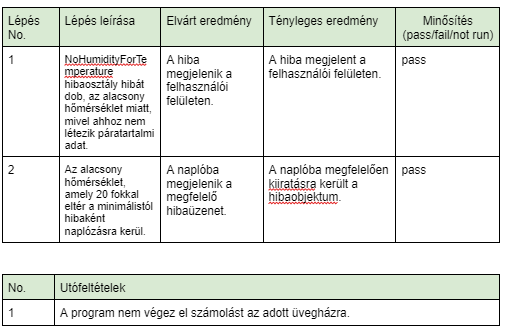
1. A vezérlőegység a működése során automatikusan betölti az üvegházak azonosítóit, az aktuális adatokat a hőmérsékletet és a páratartalmat. A következő lépésben a felhőn keresztül (MonitorService) egy azonosító alapján lekérdezi az elvárt szobahőmérsékletet és páratartalmat, majd döntést hoz, hogy milyen beavatkozásra van szükség.

2. A vezérlőegység a beérkezett adatok alapján döntést hoz, hogy szükséges-e a beavatkozás. Ha a hőmérséklet magasabb vagy alacsonyabb az elvártnál, valamint a páratartalom, akkor a felhőn keresztül (ControllerService) küld üzenetet a megfelelő eszköznek.

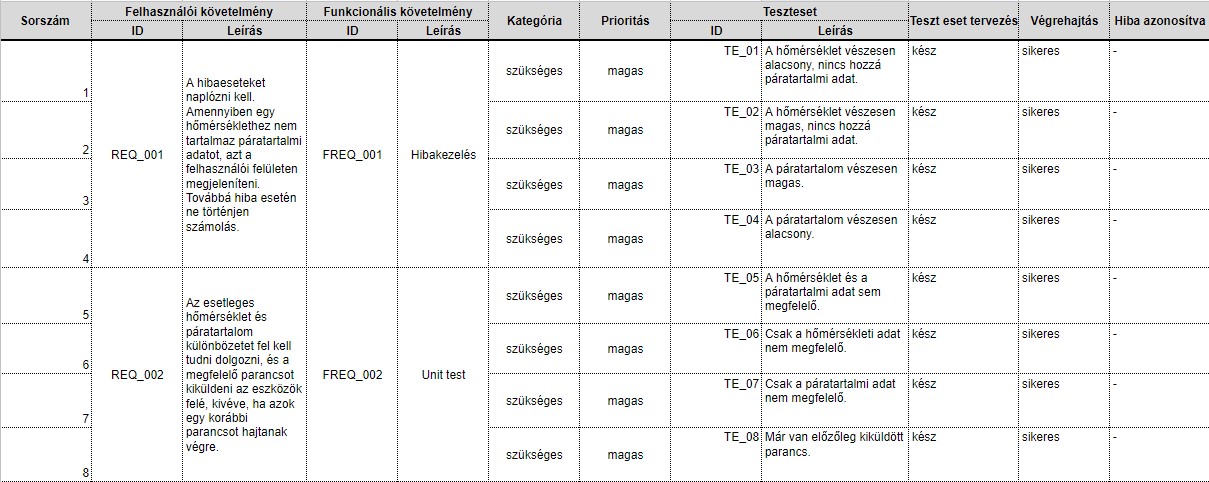
3. Ha a vezérlőegység hibát észlel, vagyis az aktuális hőmérséklet nagymértékben eltér az elvárt hőmérséklettől, illetve a páratartalom is, ekkor a rendszer egy bejegyzést készít egy naplófájlba.

**3.2 tesztesetek (átadási teszteléshez)**



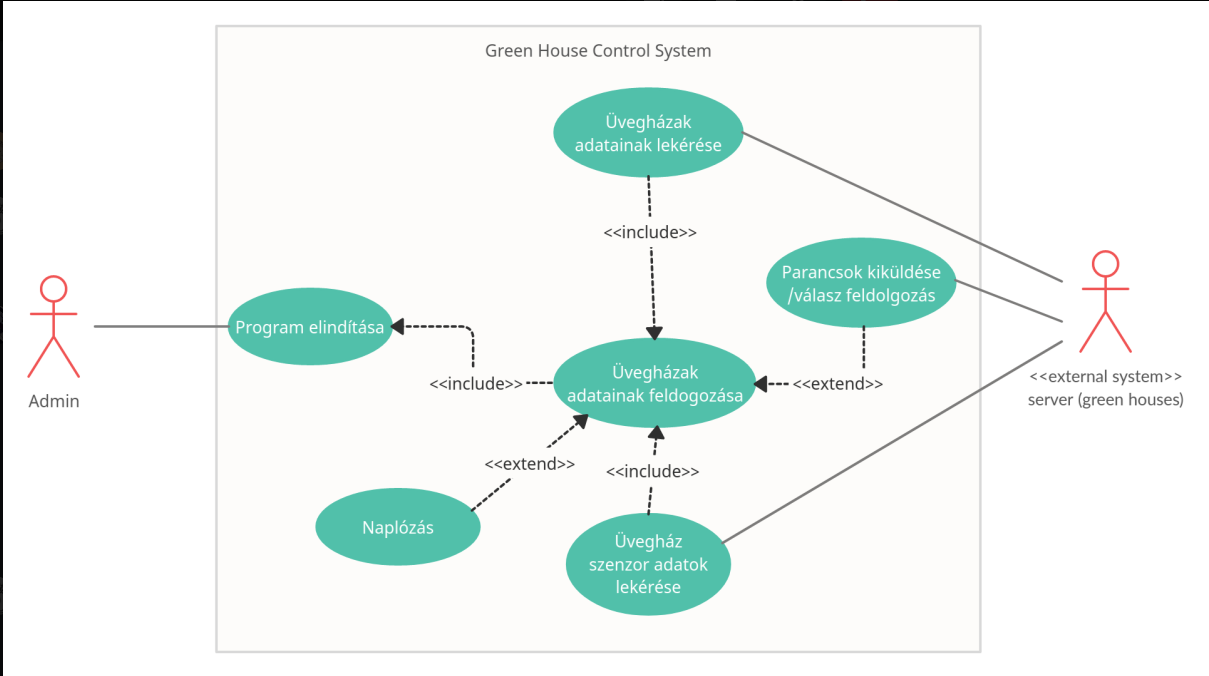


* 1. **követelmény követhetőségi mátrix**

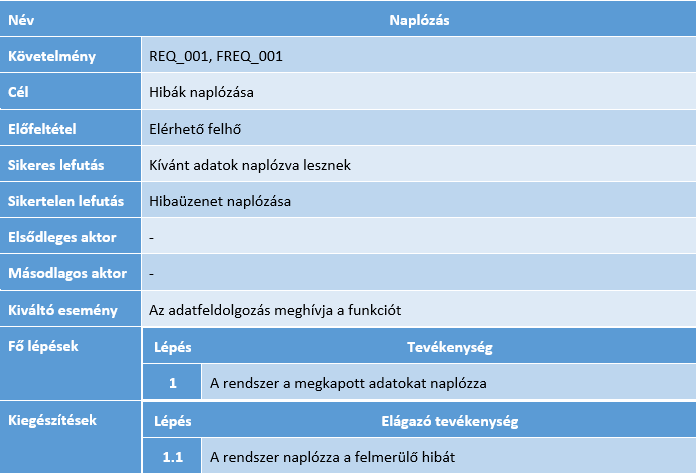
****

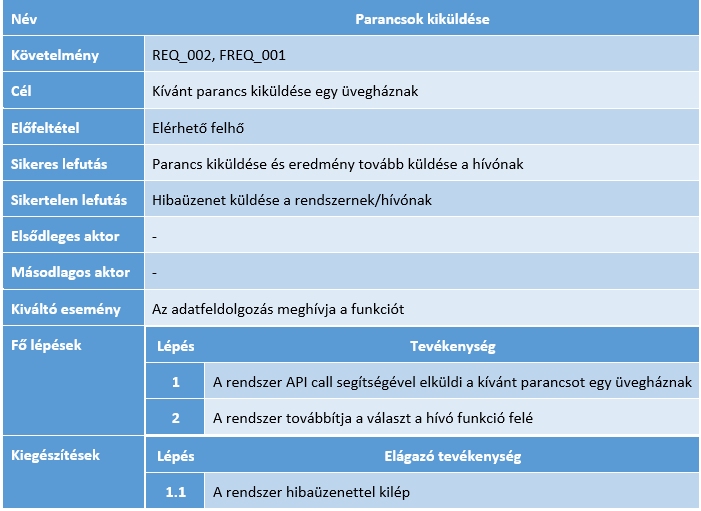
1. **Funkcionális specifikáció**

**4.1 UC diagram**

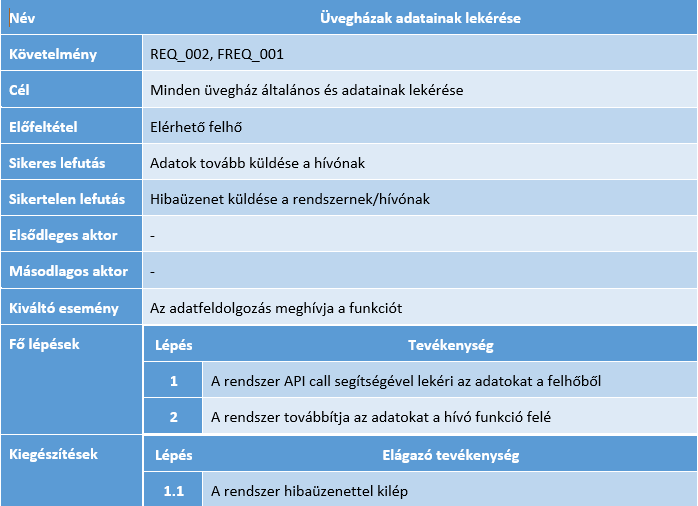


**4.2 UC táblázat**

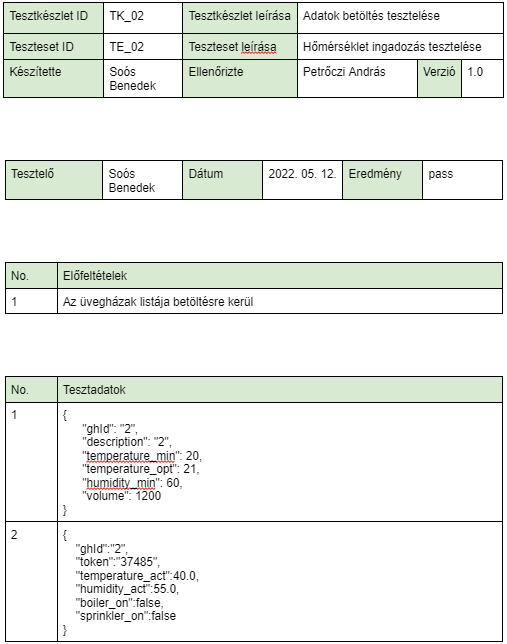


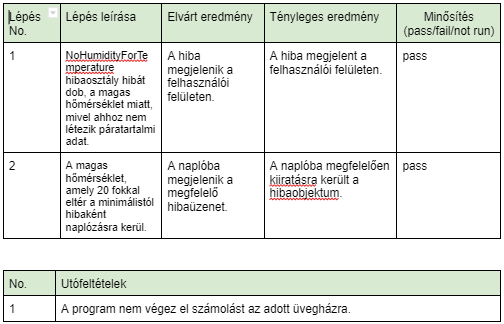






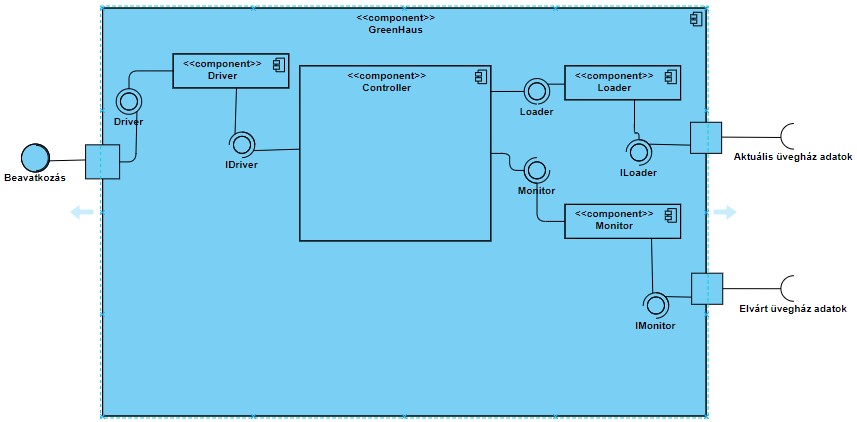
* 1. **teszt esetek (rendszer teszteléshez)**



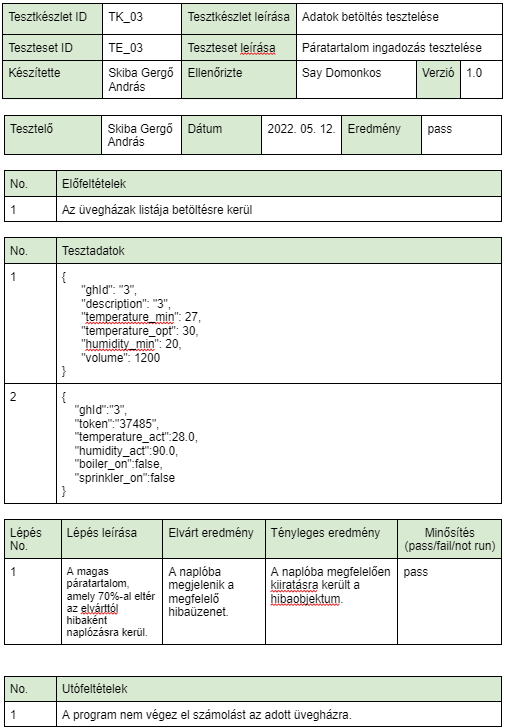


1. **Architektúra tervezés**

**5.1 Komponens diagram**

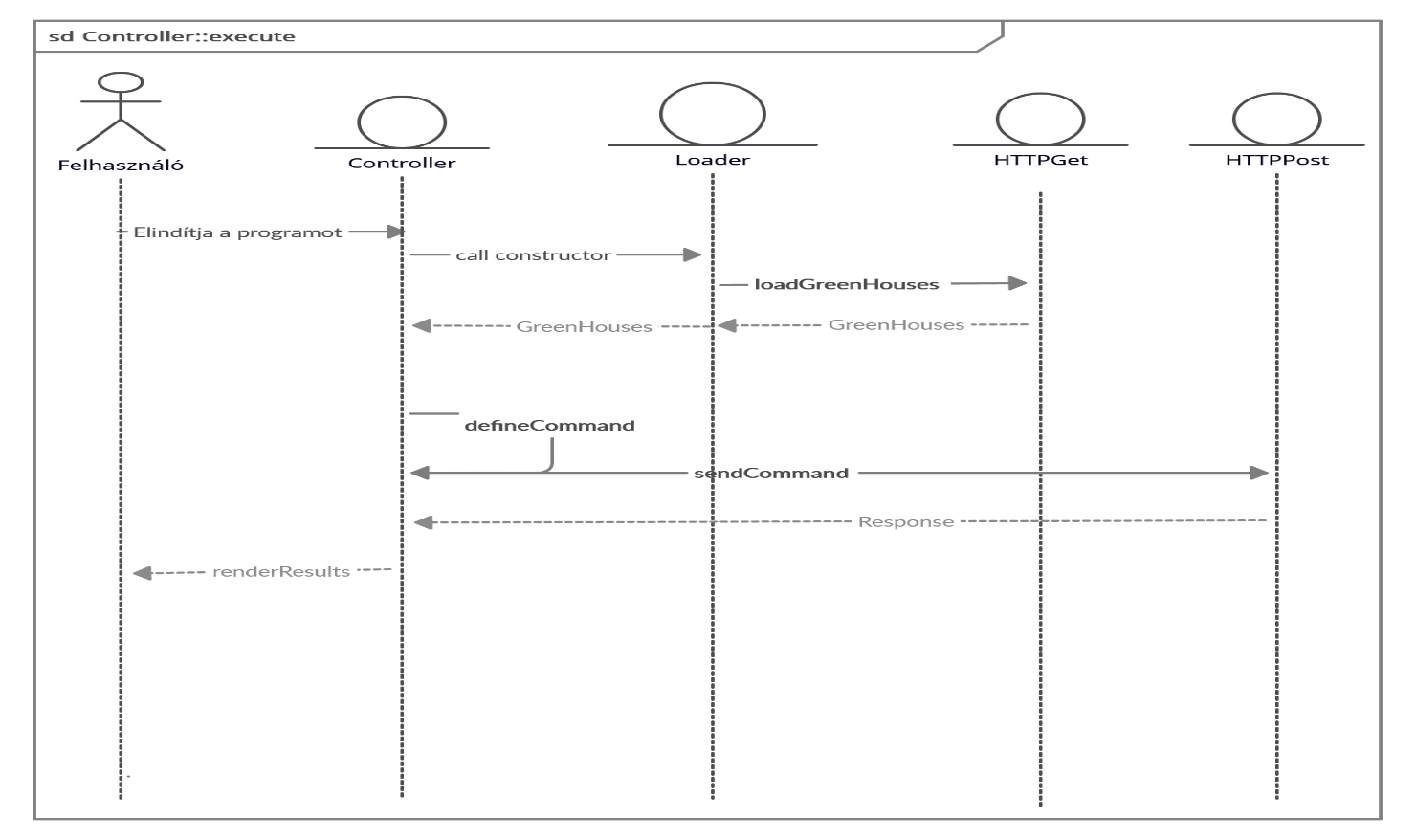
****

**5.2 tesztesetek (integrációs teszteléshez)**

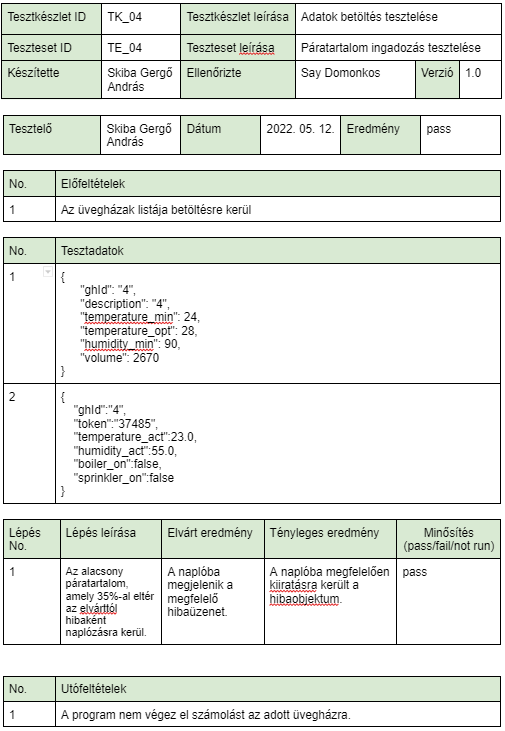


1. **Interfész tervezés**

**6.1 Szekvencia diagram**

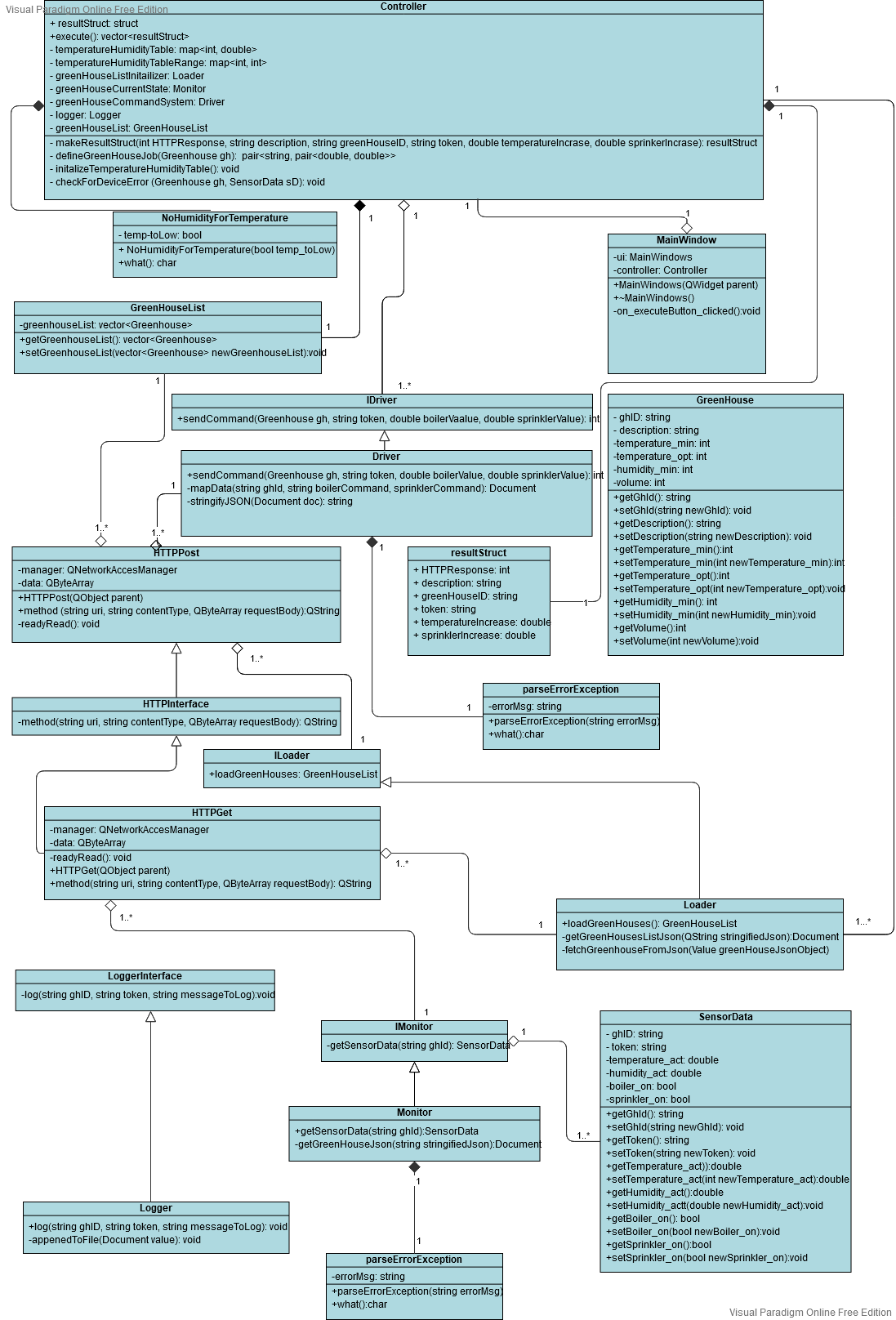
****

**6.2 tesztesetek (interfész teszteléshez)**

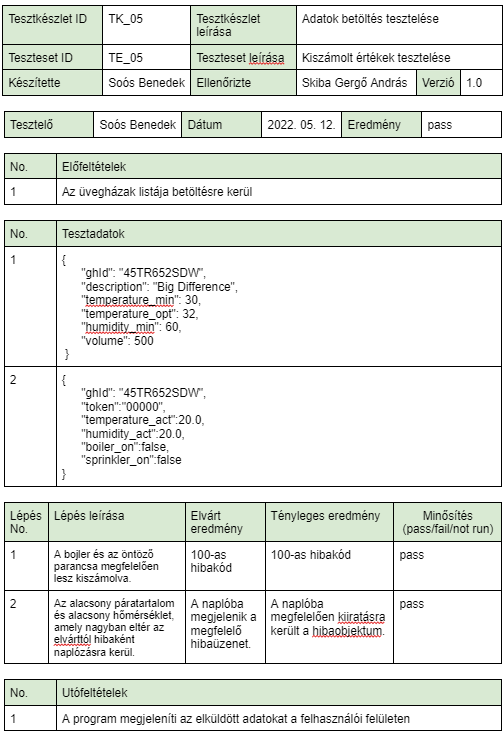


1. **Modul / komponens tervezés**

**7.1 Osztály diagram**

****

**7.2 tesztesetek**

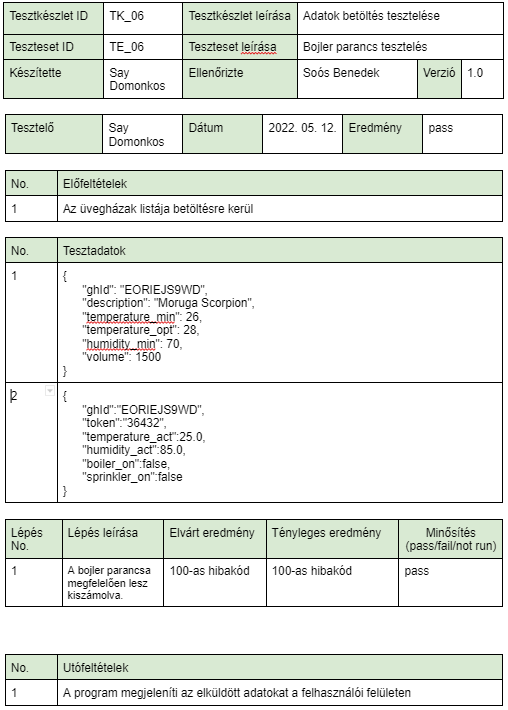


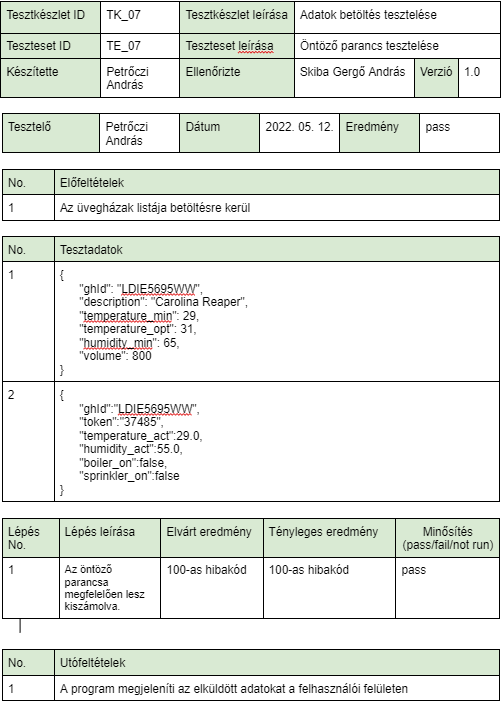
1. **Implementáció**

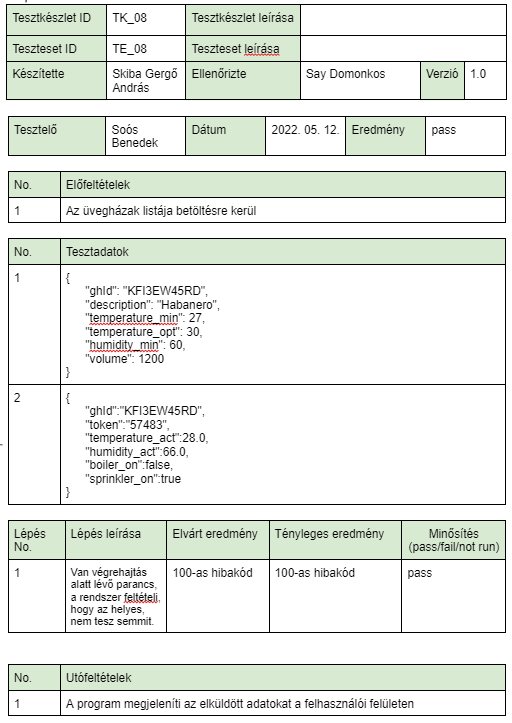
**8.1 C++**

* QNetworkAccessManager
* rapidjson library
* Qt gui

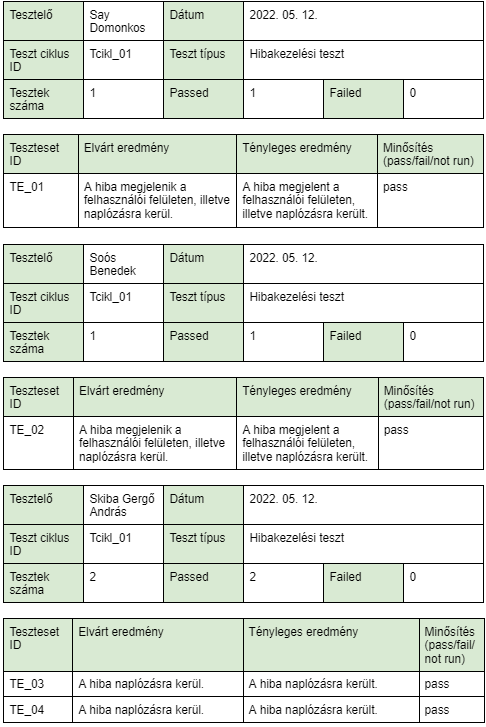
1. **Tesztelés / javítás / újra tesztelés**

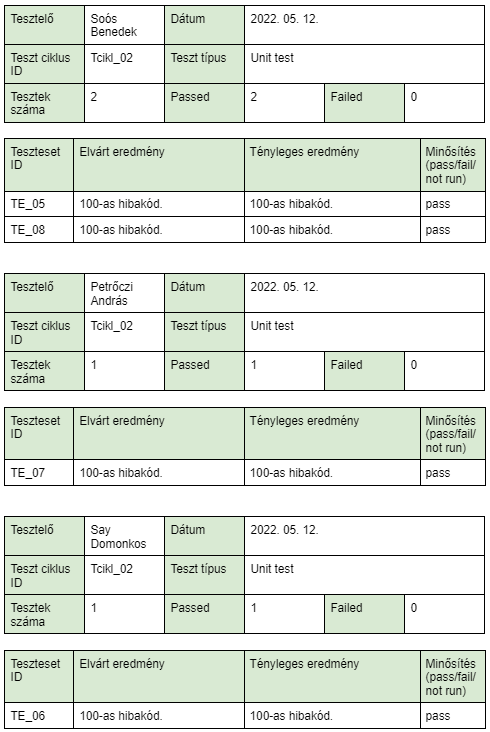






**9.1 Tesztnapló**





1. **Felhasználói dokumentáció**

**Feladat leírása:**

A projekt során a cél egy olyan felhő alapú vezérlőrendszer fejlesztése, amely képes automatikusan

szabályozni az okos üvegházak hőmérsékletét és páratartalmát a fűtés és a locsoló berendezések

vezérlésével. A rendszer képes egyszerre több üvegház kezelésére, amelyeknél a termesztett növény

igényeinek függvényében előre definiálva van az elvárt hőmérséklet és minimális páratartalom. A

rendszer a telepített szenzorok által mért adatokat a felhőn keresztül kéri le, amelyek alapján képes

önállóan döntést hozni, hogy szükséges-e valamilyen beavatkozás.

**Harverkörnyezet:**

Operációs rendszer: Linux/Windows

Grafikus kártya: Nem igényelt

RAM: max 500 MB

Perifériák: Billentyűzet