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)**

Célok:

● 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

**Teszt Teljesség (Test Completeness)**

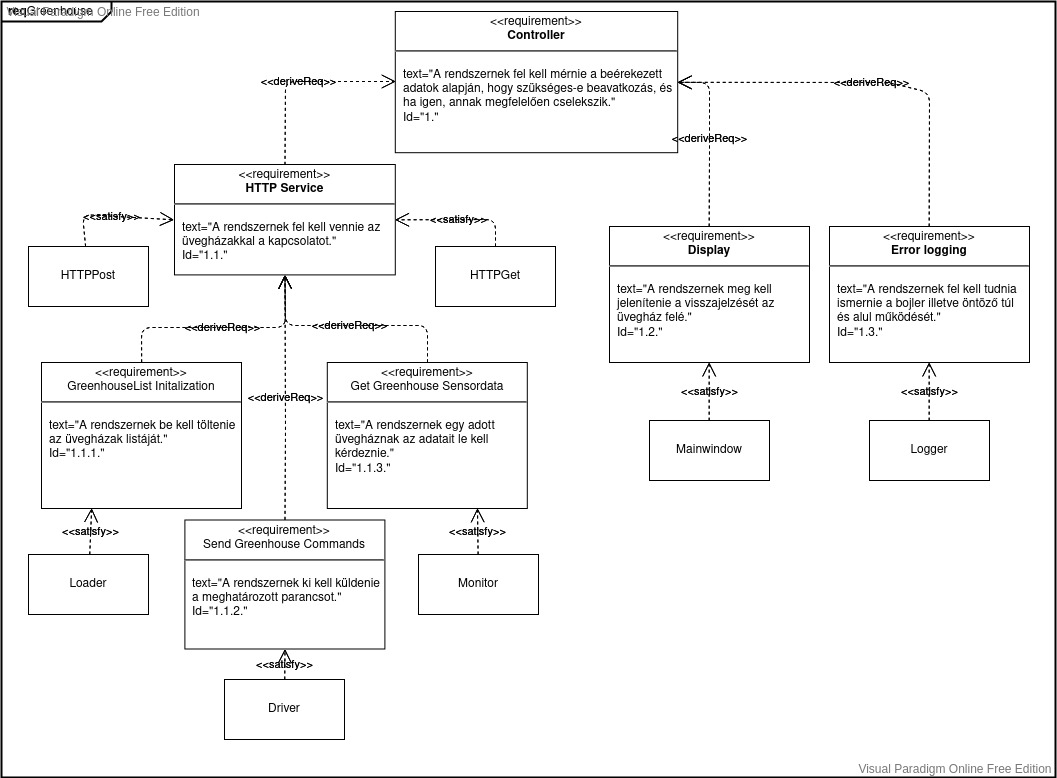
● Tesztek sikeressége 100%

● Minden teszteset végrehajtásra került

● Minden súlyos hiba kijavításra került

1. **Követelmény modell**

**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.

**teszt esetek (átadási teszteléshez)**

**követelmény követhetőségi mátrix**

1. **Funkcionális specifikáció**

**UC diagram**

**UC táblázat (benne tesztelési információ)**

**teszt esetek (rendszer teszteléshez) o követelmény követhetőségi mátrix**

1. **Architektúra tervezés**

**Komponens diagram (vagy csomag, vagy osztály)**

**teszt esetek (integrációs teszteléshez)**

**követelmény követhetőségi mátrix**

1. **Interfész tervezés**

**Szekvencia diagram (vagy osztály)**

**teszt esetek (interfész teszteléshez)**

**követelmény követhetőségi mátrix**

1. **Modul / komponens tervezés**

**Osztály diagram (legalább 1 osztály, több metódussal / funkcióval)**

**teszt esetek (modul teszteléshez) (az osztályok által megvalósított funkciót teszteli)**

**követelmény követhetőségi mátrix**

1. **Implementáció**

**C++ (lehet más is)**

**teszt esetek (unit teszteléshez) (egy osztály metódusaihoz)**

**követelmény követhetőségi mátrix**

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

**komponens (unit / modul)**

**interfész**

**integrációs**

**rendszer**

**átadási**

**minden teszt végrehajtáshoz tesztnapló**

**minden teszt ciklus végén tesztjelentés**

**legvégén tesztelés kiértékelése**

1. **Felhasználói dokumentáció**