Arkitekturbeskrivning

Författare: Max Körlinge

1. Inledning

1.1 Syfte

Arkitekturen ska användas för att först skapa en plan för projektets utförande att utgå ifrån, samt för att sedan behålla en överblick över helheten av projektet under tiden men arbetar med de olika delarna. Därefter används arkitekturbeskrivningen även för att möjliggöra för nya utecklare att arbeta vidare med projektet. Detta dokument ska ge en beskrivning av produktens arkitektur med hjälp av olika vyer för att beskriva olika delar av systemet ur olika synvinklar enligt 4+1-modellen 4+1 [1].

1.2 Omfattning

Dokumentet omfattar arkitekturen för produkten “FlowerPower” [3], som utvecklas av projektgrupp 10 på KTHs kurs “Projekt och projektmetoder” VT2018. Det omfattar ej tester av arkitekturen, som i stället finns i projektets testplan [5]. Det omfattar ej heller koddesign på detaljnivå, det vill säga klassers specifika metoder, attribut, etc. Denna detaljnivå kan i stället hittas i projektets designplan [6].

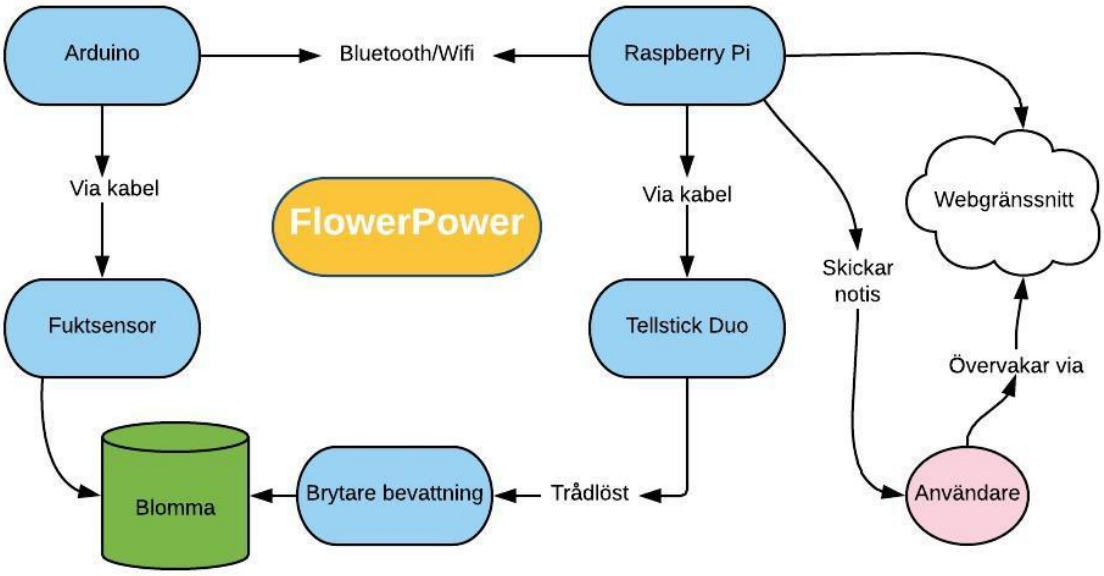
1.3 Referenser

[1] Kruchten, P. B. (1995). The 4+ 1 view model of architecture. Software, IEEE, 12(6), 42-50.
[2] Sommerville, I. (2011). Software Engineering. 9th ed. Pearson.
[3] Projektdefinition FlowerPower. https://github.com/fongie/projectgaming/wiki/6-Projektledning (Hämtad 2018-05-22).
[4] Kravdokument FlowerPower. https://github.com/fongie/projectgaming/wiki/2.-Krav (Hämtad 2018-05-22).
[5] Testplan FlowerPower. https://github.com/fongie/projectgaming/wiki/5-Test (Hämtad 2018-05-22).
[6] Designplan FlowerPower. https://github.com/fongie/projectgaming/wiki/4-Design (Hämtad 2018-05-22).

2. Användarfall

2.1 Bakgrund

Användarfallen ska beskriva viktiga sätt som användaren interagerar med produkten på [4]. Figur 1 visar en konceptuell modell över produktens beståndsdelar och hur den interagerar med användaren. Raspberry Pin agerar alltså server och servar användaren, tar in och bearbetar data från fuktsensorn via en Arduino, samt kan skicka signaler för att stänga av och sätta på bevattning via en Tellstick Duo.



Figur 1. En illustration av konceptet för produkten FlowerPower. Tjänar även som illustrering av fysisk vy. Modellen beskriver systemet när det används med endast en blomkruka.

2.2 Användarfall

- Användaren ska kunna övervaka fuktvärdena på användarens blommor via ett webgränssnitt.
- Änvändaren ska få en notis om jorden torkar ut.
- Användaren ska kunna starta bevattning av en blomma via internet.
- Användaren ska kunna ställa in systemet för att automatisk bevattna blomman när den blir torr.

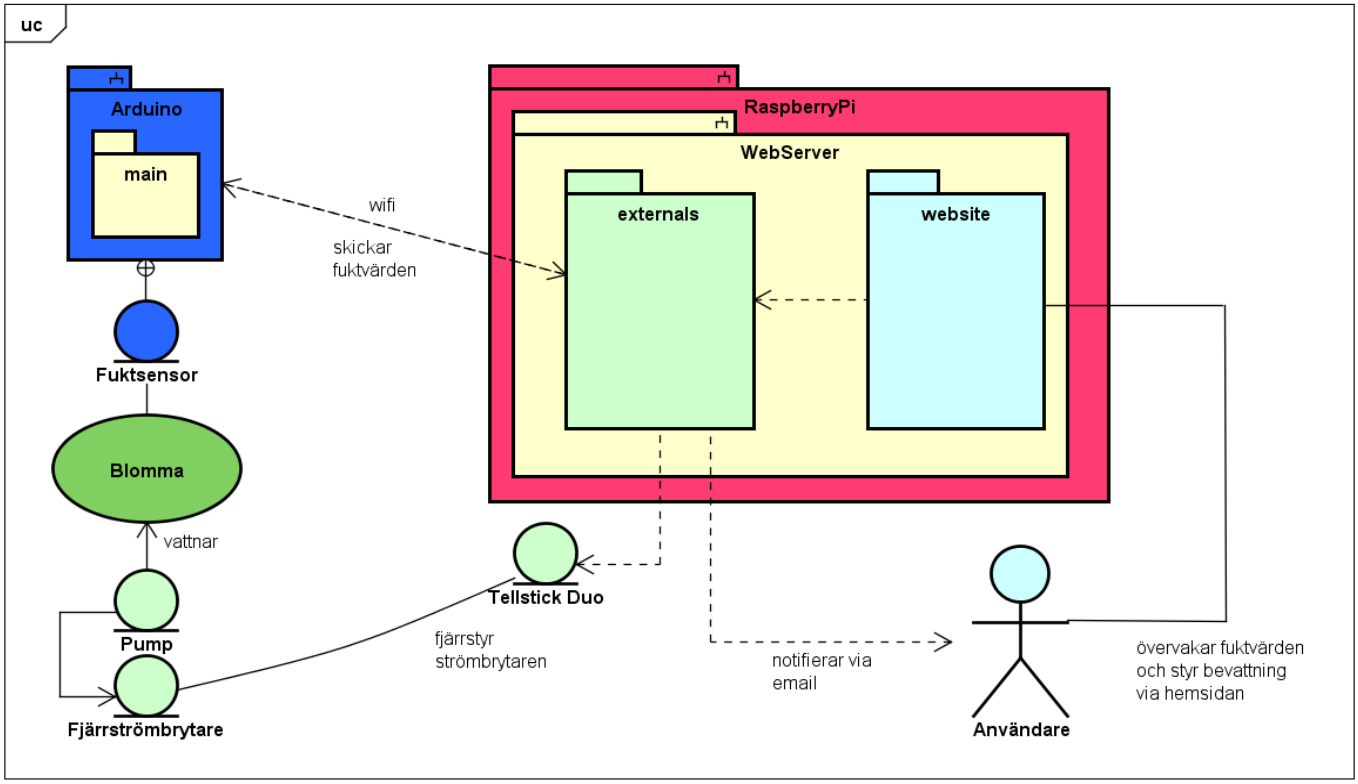
Intressenten som agerar i dessa fall är en ensam användare av produkten, som använder den på en eller flera blomkrukor. Det behövs en Arduino för varje blomkruka, men endast en Raspberry Pi.

3. Logisk vy

3.1 Bakgrund

Den logiska vyn beskriver hur de viktigaste delarna hänger ihop logiskt. Den ger en överblick av hur systemet passar ihop, och fungerar som en blandning och översikt av de andra vyerna.

3.2 Logisk arkitektur



Figur 2. En logisk vy över fuktkontroll- och bevattningssystemet. Färgkodning för att försöka beskriva vilka komponenter som tillsammans bildar logiska subsystem.

3.2.1 Arduino
Detta subsystem hanterar interaktion med fuktsensorn, samt anslutningen till centralservern (Raspberry Pin) via wifi. Det kommer att sända information om blommans fuktstatus till *externals*.

3.2.2 website
På Raspberry Pin ligger subsystemet website som hanterar gränssnittet som Användaren ser och kan utnjyttja. Därifrån kan användaren övervaka fuktstatus samt starta bevattningar.

3.2.3 externals
På Raspberry Pin ligger också subsystemet externals, som hanterar allt som rör blomman och externa komponenter: mottagande av fuktstatus från Arduinon, startande av bevattning via Tellstick Duon, samt sändandet av notifikationer till användaren vid behov.

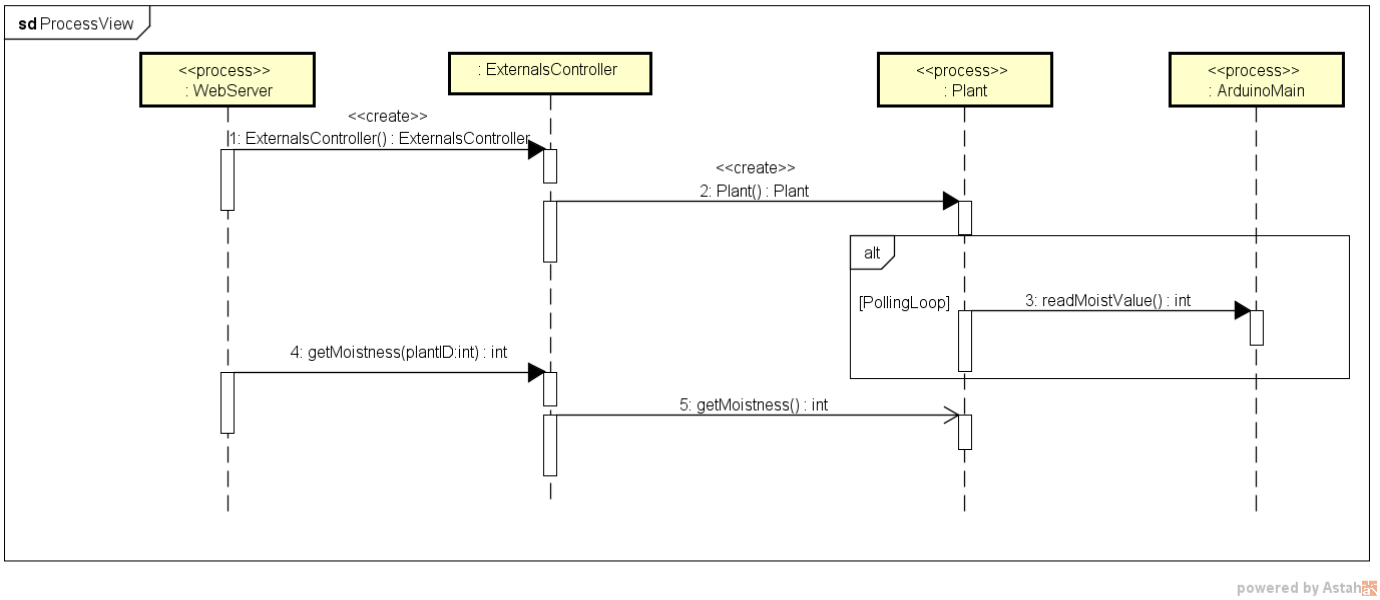
4. Processvy

4.1 Bakgrund

Processvyn beskriver vissa icke-funktionella krav så som optimeringar, parallelism och distribution. Den beskriver processerna som körs på samma eller på olika

fysiska enheter och ger en överblick över hur de interagerar med varandra. För en närmare beskrivning över de fysiska enheterna, se den konceptuella modellen och den fysiska vyn.

4.2 Beskrivning



Figur 3. Ett sekvensdiagram över processerna och exempel på hur de kommunicerar med varandra.

4.2.1 WebServer

Webservern som driver hemsidan är också startpunkten för systemet, som initierar de externa systemen via ExternalsController. Denna controller startar och hanterar alla processerna av typen Plant. Webserverprocessen körs på Raspberry Pin.

4.2.3 Plant

En Plant-process per blomma hanterar all kommunikation med extern hårdvara, det vill säga Arduinon med fuktsensorn, och Tellstick Duon som hanterar bevattningen. Plantprocesserna och kommunikationerna från dem hanteras av klassen ExternalsController.

4.2.3 Arduino

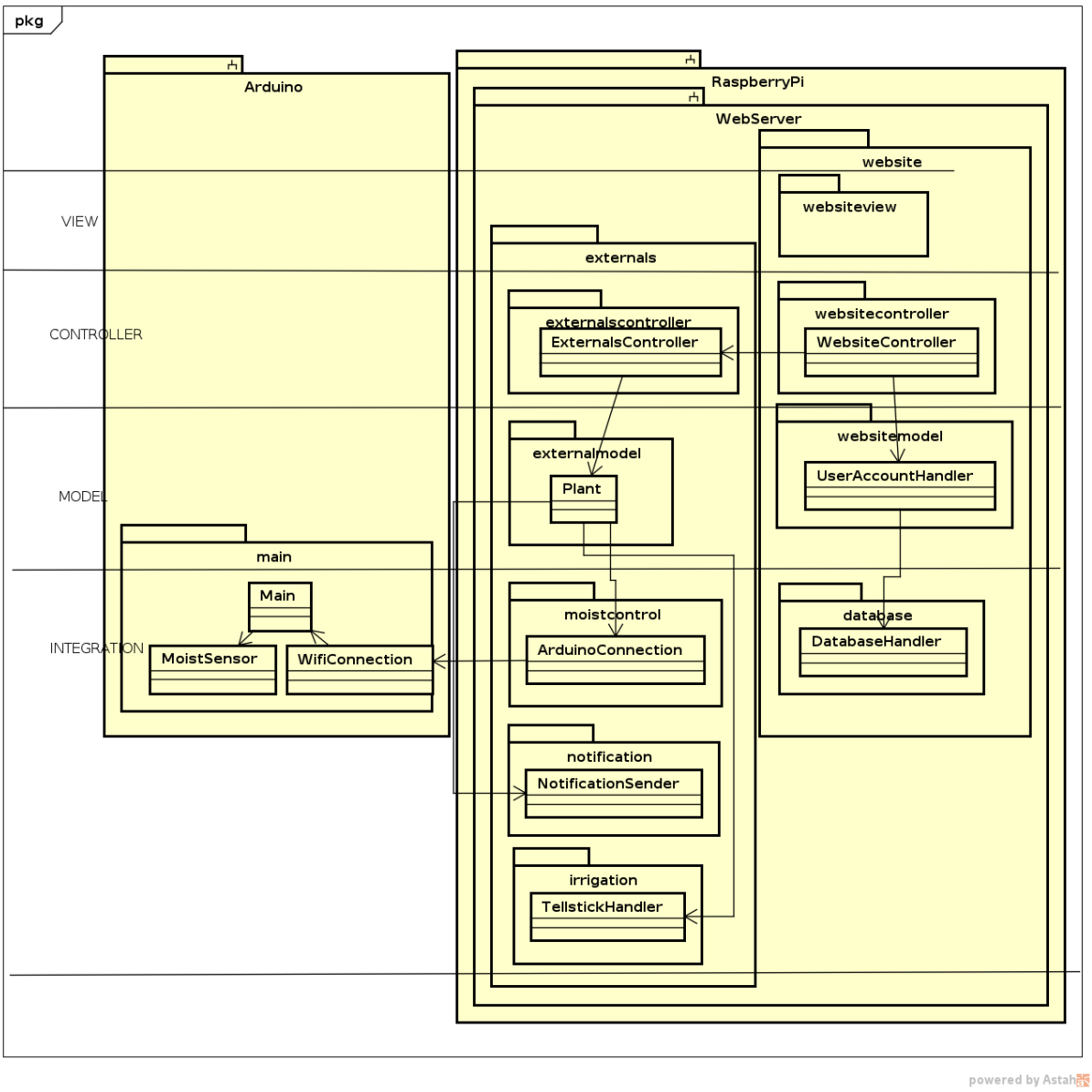
På arduinon körs en process som läser av fuktsensorn och kommunicerar via wifi med Raspberry Pin. En Plant-process pollar denna process för nya uppdaterade fuktvärden vid vissa intervall.

5. Utvecklingsvy

5.1 Bakgrund

Utvecklingsvyn ger en mer detaljerad beskrivning av systemets kodstruktur, indelat i subsystem, lager, och klasser. Koden ska vara indelad så att den i lagom stora delar kan läggas ut på de olika utvecklarna av projektet.

5.2 Beskrivning



Figur 3. En ritning över utvecklingsvyn för systemet.

Utvecklingen av systemet sker i form av flera mindre subsystem, som var för sig är inordnat enligt MVC-mönstret (Model-view-controller). Figur 3 är en översikt över subsystemen och paketen som ingår där i. Mer detaljerad beskrivning av klassernas publika interface och implementation finns i projektets Designplan [6].

6. Fysisk vy

6.1 Bakgrund

Den fysiska vyn beskriver framförallt icke-funktionella krav såsom vilka hårdvarudelar systemet är uppbyggt av och hur de är sammankopplade.

6.2 Beskrivning.
Figur 1 visar den fysiska sammansättningen av systemet. Ett av huvuddragen i designen är att kommunikation sker trådlöst mellan systemdelarna som är nära blomman (Arduinon samt Tellstick Duon) och Raspberry Pin. Detta för att man ska kunna använda systemet till flera blommor över ett längre avstånd. Se nedan för mer detaljerad beskrivning.

6.3 Hårdvara

- Raspberry Pi 3B med Raspbian OS (+ strömkälla)
- Arduino (+ strömkälla)
- Wifi-modul ESP8266 till Arduino
- Fuktsensor kompatibel med Arduino, till exempel Art.nr 41015738 på electrokit.com (hämtad 22-05-2018).
- Tellstick Duo
- Fjärrstyrd strömbrytare kompatibel med Tellstick Duo.
- Eheim Pump Compact ON 300 eller annan vattenpump som styrs med elektricitet

6.3.1 Raspberry Pi 3B
Raspberryn är ansluten till:

- Vägguttag för el, via sladd
- LAN-port, för att ansluta till internet och få en statisk IP
- Arduinon, trådlöst via wi-fi
- Tellstick Duo, via kabel

6.3.2 Arduino
Arduinon är ansluten till:

- Raspberry Pin, via wi-fi
- Vägguttag för el, via sladd
- Fuktsensorn, via sladd

6.3.3 Fuktsensor
Fuktsensorn är ansluten till Arduinon via sladd, och dess placering ska vara i blomjorden som ska mätas.

6.3.4 Tellstick Duo
Tellstick Duons sändare är kopplad till Raspberry Pin via USB. Sändaren kommunicerar trådlöst med fjärrströmbrytaren.

6.3.5 Pump
Pumpen är kopplad till fjärrströmbrytaren, som sitter i ett elnätuttag. Pumpen ska befinna sig i en behållare med vatten, och dess slang ska leda vattnet till blomjorden. När fjärrströmbrytaren sätts på, ska vattenpumpen pumpa vatten.