PA1414

Individuellt programvaruprojekt

Dokumentation Bevattningssystem

David Hörnmark - PAGIP16 - daho16@student.bth.se - hornmark94@gmail.com

Systembeskrivning

Vad detta system skulle göra är att kunna hantera sensorer som kan mäta bland annat: fukt, temperatur, sol, gödslingsbehov och magnetkontakt.

Genom att sedan koppla samman detta med olika händelser kommer hanteringen av växthuset automatiseras på ett bättre sätt.
De funktioner som ska finnas är: vattna, skugga, värma, gödsla och se om fönster/dörrar är öppna/stängda.

Genom att sammanföra dessa sensorer och funktioner kommer kunden automatiskt kunna: aktivera bevattning om sensorn meddelar ett värde på fukt som passerar gränsvärdet (t.ex. 30%) i jorden. Detta gäller då även för temperaturen, säg över 35 grader så aktivera systemet skuggning, för temperaturer under 25 grader avaktiverar systemet skuggningen och för temperaturer under säg 3 grader aktiverar system värmen.

I systemet kommer det även vara möjlig att lägga till och ta bort sensorer och manuellt aktivera funktionerna via en webbsida. Dessutom kommer kunden på ett enkelt sätt kunna redigera kategori på sensorerna (t.ex. tomater, gurkor mm.).

Systemet kommer även att aktivt logga olika mätningar och presentation av statistik.

Krav

Som kund vill jag kunna läsa av data från flera sensorer, denna data innehåller mätning av fukt, temperatur, sol, gödslingsbehov samt magnetkontakt.
Som kund vill jag manuellt styra funktionerna: vattna, skugga, värma, gödsla samt se om fönster/dörrar är öppna eller stängda.
Som kund vill jag kunna se statistik för varje sensor.
Som kund vill jag att systemet skall utföra funktionerna vattna, skugga, värma automatiskt genom att datan från sensorerna kontrolleras mot gränsvärden.
Som kund vill jag kunna lägga till samt ta bort sensorer.
Som kund vill jag kunna namnge sensorerna.
Som kund vill jag kunna kom åt systemet via en webbsida.
Som kund vill jag kunna ställa in avläsningsintervall.

Teknik

Tanken är att koppla sensorerna, Xiaomi Mi Flora, till en Raspberry PI 3 via BlueTooth och därigenom extrahera data som sedans lagras i en MySQL-databas (MariaDB). Denna data presenteras på en webbsida som körs lokalt på Raspberry PI:n. Vidare är det tänkt att forwarda en port på routern, vilket gör att kunden kan nå sin hemsida via routerns externa IP-adress plus port.

Servern körs via Express frameworket för NodeJS. Templates som användes för webbsidan skrivs i språket Pug (tidigare Jade). Vidare används Google Charts för att tillverka linjediagram där statistik för varje sensor presentera.

Funktioner

ID	Beskrivning	Prio	Kostnad
F1	Logga in på hemsidan.	Must	10
F2	Extrahera och presentera data om fukt, temperatur, sol, gödslingsbehov samt magnetkontakt från sensor.	Must	20
F3	Manuellt aktivera funktionerna vattna, skugga, värma, gödsla samt se om fönster är öppna.	Must	20
F4	Automatiskt aktivering av funktionerna vattna, skugga, värma, gödsla.	Must	15
F5	Användarvänlig hantering av sensorerna (lägga till, ta bort, namnge).	Must	20
F6	Presentera statistik för varje sensor på den data som har lagrats över tid.	Must	10
F7	Göra GUI:t mer attraktivt för användaren.	Should	20
Testning	Testning av stabilitet samt av samtliga funktioner av systemet.		10

Sprintar

Sprint 1, 30 timmar		
Funktion	**Beskrivning**	**Kostnad**
F1	Logga in på hemsidan.	10
F2	Extrahera och presentera data om fukt, temperatur, sol, gödslingsbehov samt magnetkontakt från sensor.	20

Sprint 2, 35 timmar		
Funktion	**Beskrivning**	**Kostnad**
F3	Manuellt aktivera funktionerna vattna, skugga, värma, gödsla samt se om fönster är öppna.	20
F4	Automatiskt aktivering av funktionerna vattna, skugga, värma, gödsla.	15

Sprint 3, 20 timmar		
Funktion	**Beskrivning**	**Kostnad**
F5	Användarvänlig hantering av sensorerna (lägga till, ta bort, namnge).	20

Sprint 4, 10 timmar		
Funktion	**Beskrivning**	**Kostnad**
F6	Presentera statistik för varje sensor på den data som har lagrats över tid.	10

Sprint 5, 30 timmar		
Funktion	**Beskrivning**	**Kostnad**
F7	Göra GUI:t mer attraktivt för användaren.	20
Testning	Testning av stabilitet samt av samtliga funktioner av systemet.	10

Att göra - Backlog

Att göra för Sprint 1, F1, 10 timmar	
Aktivitet:	A1 - Starta upp en lokal server
Klar när:	Kontakt har etablerats med den lokala servern via webbläsaren.
Tid:	2
Status:	Klar - 2017/09/11 - 1 timme
Aktivitet:	A2 - Göra login sidan
Klar när:	Startsidan består av element som krävs för att kunna logga in.
Tid:	2
Status:	Klar - 2017/09/11 - 3 timmar
Aktivitet:	A3 - Skapa och koppla en databas med tabell för att hantera användare till servern
Klar när:	Databasen innehåller en tabell och som hanterar användare och är kopplad till servern.
Tid:	2
Status:	Klar - 2017/09/12 - 2 timmar
Aktivitet:	A4 - Göra SQL-procedur för att autentisera inloggningsuppgifterna
Klar när:	Proceduren kan ge svar vid korrekta samt felaktiga uppgifter.
Tid:	2
Status:	Klar - 2017/09/12 - 1 timme
Aktivitet:	A5 - Test av inloggningsfunktionen
Klar när:	Lokalt demo kan köras framgångsrikt.
Tid:	2
Status:	Klar - 2017/09/12 - 1 timme

Att göra för Sprint 1, F2, 20 timmar	
Aktivitet:	A1 - Koppla in sensorerna till Raspberry PI:n
Klar när:	Test kan påvisa att kontakt till sensorn är etablerad.
Tid:	10
Status:	Klart - 2017/09/18 - 8 timmar
Kommentar:	I och med att sensorn saknar API så krävdes det att jag hittade ett tredje-parts bibliotek som kunde lösa det. Finns bibliotek skrivna i JavaScript och Python. Blev tillslut GATTLib som är skrivet i Python.
Aktivitet:	A2 - Skapa python-funktion för extrahera data från sensorn och lagra den i databasen som kan köras automatiskt med visst intervall.
Klar när:	Metoden visar sig ha lyckats extrahera data och kunna lagra datan i tabellen på databasen automatiskt med visst intervall.
Tid:	5
Status:	Klart - 2017/09/25 - 2017/09/28 - 12 timmar
Kommentar:	Var svårt att hitta ett sätt som kunde köra med SUDO-rättigheter och script inuti script. Löstes via användning av CRONTAB.
Aktivitet:	A3 - Skapa SQL-procedur för att extrahera den senaste datan från tabellen.
Klar när:	Proceduren kan leverera den senast lagrade datan för vardera sensor.
Tid:	5
Status:	Klart - 2017/09/25 - 2017/09/28 - 6 timmar
Kommentar:	Det som tog lite tid här var att jag ville omformatera datumen till mer lättlästa så krävdes att jag läste en del referensmaterial.
Aktivitet:	A4 - Skapa en sida som presenterar datan.
Klar när:	Korrekt data visas för korrekt kolumn och sensor.
Tid:	5
Status:	Klart - 2017/09/25 - 2017/09/28 - 3 timmar

Att göra för Sprint 2, F3, 20 timmar	
Aktivitet:	A1 - Skapa sida och lägga till element för manuell aktivering av funktionerna: vattna, skugga, värma, gödsla och skuggning.
Klar när:	Nödvändiga element som krävs för att visuellt kunna manuellt aktivera funktionerna finns tillgängliga för användaren.
Tid:	5
Status:	Delvis klar - 2017/10/05 - 1 timme
Kommentar:	Implementerat det som krävs för bevattningen.
Aktivitet:	A2 - Skapa Python-script som kopplar på/stänger av ventiler, switchar, mm.
Klar när:	Funktionerna resulterar i att reläkortet skickar en elektrisk signal.
Tid:	5
Status:	Delvis klar - 2017/10/05 - 2 timmar
Kommentar:	Implementerat det som krävs för reläkortet och bevattningen.
Aktivitet:	A3 - Skapa SQL-procedurer för att uppdatera status på givarna i databasen.
Klar när:	Alla procedurer passerar testet.
Tid:	5
Status:	Delvis klar - 2017/10/06 - 2 timmar
Kommentar:	Implementerat det som krävs för bevattningen.
Aktivitet:	A4 - Koppla elementen på webbsidan till Python-scripten och SQL-procedurerna.
Klar när:	Elementen på hemsidan kan lyckat genom test aktivera ventiler, switchar osv.
Tid:	5
Status:	Delvis klar - 2017/10/06 - 2 timmar
Kommentar:	Implementerat det som krävs för bevattningen.

Att göra för Sprint 2, F4, 15 timmar	
Aktivitet:	A1 - Skapa gränsvärden i databasen för att vattna, skugga, värma, gödsla. Samt skapa SQL-procedurer för att lagra, uppdatera och nå värdena.
Klar när:	Gränsvärden går att lagra, uppdatera och nå via SQL-procedurer.
Tid:	5
Status:	Delvis klar - 2017/10/06 - 2 timme
Kommentar:	Implementerat det som krävs för bevattningen.
Aktivitet:	A2 - Skapa Python-kod som jämför aktuell data mot gränsvärdena för bevattning, skugga, värma och gödsla.
Klar när:	Koden kan hämta gränsvärdena via SQL-procedure och lagra de i Python-variabler som kan användas för att jämföras mot aktuell data.
Tid:	5
Status:	Delvis klar - 2017/10/06 - 1 timme
Kommentar:	Implementerat det som krävs för bevattningen.
Aktivitet:	A3 - Sammanför A1 och A2 till Sprint 1 F2 för att få en automatisk aktivering av bevattning, skuggning, värme och gödsling.
Klar när:	Reläkortet aktiveras genom att ett värde som har lästs från sensor som understiger gränsvärdet för den funktionen.
Tid:	5
Status:	Delvis klar - 2017/10/07 - 1 timme
Kommentar:	Testa det som krävs för bevattningen.

Att göra för Sprint 3, F5, 20 timmar	
Aktivitet:	A1 - Skapa sida med element för användarvänlig hantering av sensorerna (lägga till, ta bort, namnge).
Klar när:	En ny sida med alla nödvändiga element existerar.
Tid:	5
Status:	Klar- 2017/09/30 - 2017/10/01 - 6 timmar
Aktivitet:	A2 - Skapa SQL-procedurer för att skapa nya, ta bort och redigera sensorer.
Klar när:	Procedurerna på ett lyckat sätt kan utföra sina uppgifter
Tid:	10
Status:	Klar - 2017/09/30 - 4 timmar
Aktivitet:	A3- Koppla element på sidan till procedurerna.
Klar när:	Varje element klarar av att aktivera korrekt procedur.
Tid:	5
Status:	Klar - 2017/10/01 - 2 timmar

Att göra för Sprint 4, F6, 10 timmar	
Aktivitet:	A1 - Skriv JavaScript-kod för routerna som ska användas för att visa linjediagrammen,.
Klar när:	Routern går att nå via POST-request.
Tid:	2
Status:	Klar - 2017/10/07 - 1 timme
Aktivitet:	A2 - Skriv Pug-template för sidan med integrerat Google Charts
Klar när:	Linjediagram som ska visa all data finns.
Tid:	8
Status:	Klar - 2017/10/07 - 2017/10/08 - 6 timmar

Att göra för Sprint 5, F7, 20 timmar	
Aktivitet:	A1 - Allmänt göra användargränssnittet mer attraktivt.
Klar när:	Användargränssnittet är tillräckligt attraktivt.
Tid:	20
Status:	Påbörjad - Har hållits på med allt eftersom saker ha implementerats. - 10 timmar

Att göra för Sprint 5, Testning, 10 timmar	
Aktivitet:	Testning av stabilitet samt av samtliga funktioner av systemet.
Klar när:	Systemet påvisar en faktiskt stabilitet och hög grad av funktionalitet.
Tid:	10
Status:	Påbörjad - Testning sker allt eftersom - 2017/10/08 - 5 timmar