

Elcykelpool Utvecklingsprojekt

Individuell rapport

av

Göran Trivic

Högskolan i Halmstad

Innehåll

1. Inledning	4
1.1 Syfte & mål	4
1.2 Avgränsning	4
1.3 Problemställning	4
2. Bakgrund	5
2.1 Delsystem1	5
2.2 Delsystem2	5
2.3 Delsystem3	6
3. Teori	6
3.1 Enkortsdator	7
3.2 LAMP	8
3.3 WordPress	9
3.4 RS485	10
3.5 Kommunikation protokoll	11
4. Metod	12
4.1 Dokumentation och kommunikation	12
4.2 Delsystem	13
4.3 Val av komponenter	13
4.4 Utveckling av databas	14
4.5 Utveckling av Hemsida	15
4.6 Script	15
4.7 Gränssnitt	16
5. Resultat	16
6. Diskussion	18
Referenser	19
7. Appendix	21
7.1.	21

1. Inledning

Detta projekt ingår i kursen Utvecklingsprojekt[1] där vi blev delade i fem olika grupper och varje grupp blev tilldelat ett projekt. Vi blev tilldelade projektet Elcykel där uppgifterna är att driftsätta en elcykelpool för studenterna på Halmstad högskola.

Elcykelpoolen var från början en del av projektet Elmob, projektet leddes av Göteborgs stad och syftet med Elmob var att underlätta hållbart resande i Göteborg[2].

Eftersom Off Course var en av de företag som hade en ledande roll i att utveckla elcykelpoolen fick projektet fortsätta dock utan en elcykelpool när Off Course gick i konkurs[3]. Till sist hamnade projektet och de färdiga delarna i händerna på högskolan i Halmstad, där vi studenter ska analysera och avgöra vad i projektet som kan användas, förbättras eller förändras.

1.1 Syfte & mål

Vårt mål med detta projekt är att få i gång en elcykelpool för studenterna på högskolan i Halmstad. En student ska kunna logga in på vår hemsida och boka en cykel för att sedan hämta ut cykeln från det valda garaget. Projektet består av tre delsystem server, garage, cykel. Där av garage och cykel är de färdiga delarna från Elmob projektet.

1.2 Avgränsning

Projektgruppen består av fyra medlemmar, var av två elektroingenjörer, en mekatronikingenjör och en dataingenjör.

Eftersom projektet innehåller tre delsystem delade vi upp arbetet i två grupper,

- **Data & Mekatronik : Delsystem1**
- **Elektro : Delsystem2**

Denna rapport kommer fokusera på de delar och delsystem som dataingenjören har arbetat på och kommer inte innehålla en fullständig redovisning av delsystem 2 och hårdvaran som elektro gruppen utvecklar.

1.3 Problemställning

Server

- Vad finns det för metoder för att skapa en server?
- Vad finns det för metoder för att skapa en hemsida?
- Vad finns det för metoder för att lagra data från hemsidan och garaget i databaser på servern?

Kommunikation mellan server och garage

- Hur kan servern hämta data och skicka styrsignaler till de olika facken i garaget?

2. Bakgrund

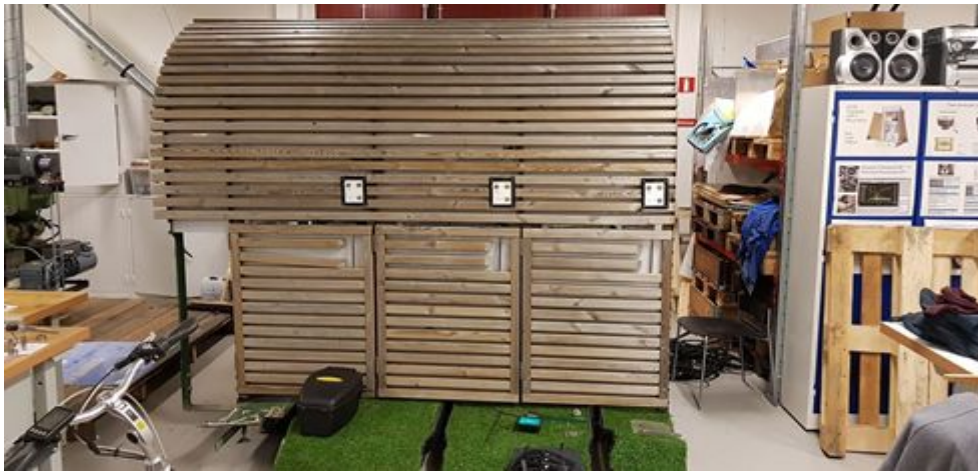
2.1 Delsystem1

Delsystem 1 består av hemsida, databas och server. I Delsystem1 finns det inget som är färdigt från de tidigare projektet, utan detta ska utvecklas från grunden av Data & elektro-gruppen. Detta ger oss en större frihet men även utmaning, då vi alltid haft riktlinjer i de tidigare kurserna. Nu får vi själva välja hur vi ska bygga upp systemet samt vilka språk och verktyg vi ska använda.

2.2 Delsystem2

Delsystem två innehåller garaget och alla komponenter som tillhör garaget.

Garaget som lämnades över till oss är uppbyggt av så kallade moduler där varje modul består av 3 fack. I varje fack finns de en hiss, lås till dörren, laddning till cyklarna, knappsats, IR-länk, sensorer och ett kretskort som kallas "com-module" som styr all intelligens i facken. I varje fack får det endast plats med en cykel. Garaget kan som mest bestå av 4 moduler, vilket innebär att man som högst kan bevara 12 cyklar åt gången.



Figur1: En bild på huvudmodulen.

Den modul som visas i Figur1 kallas huvudmodulen och är den modul som vi kommer fokusera arbetet på.

Det som gör huvudmodulen unik är att det både finns ett elskåp och en enkortsdator i modulen. Enkortsdatorn som sitter i huvudmodulen är en BeagleBone-Black, denna sköter kommunikationen mellan servern och Com-modulerna som sitter i facken via gränssnittet RS485. Medans elskåpet levererar ström till alla moduler som är anslutna.

Grundtanken för garaget var att man skulle spara utrymme genom att hissa upp cyklarna, men som man kan se på Figuren1 blev inte resultatet som tänkt. Man har knappt tjänat något på att hissa upp cyklarna då garaget är tillräckligt stort för att nästan kunna köra in cyklarna helt och hållet utan att hissa upp dem.

Istället har de bidragit till nackdelar som t.ex. en klumpig hiss där kunden själva får ansvara över att styra hissen för att få ut/lämna in cykel.

Detta bidrar till att service måste ske väldigt ofta och underhållningen av garaget blir onödigt dyrt. Samt att de ställer onödiga krav på kunden då kunden måste hissa upp/ner cykeln rätt för att påbörja/avsluta en order.

Då vi bara har ett antal månader på oss kommer vi att fokusera arbetet på att de inre systemet i garaget.

Det finns C kod för BeagleBonen och com-modulerna, koden ska i stort sätt fungera och kunna hantera kommunikationen mellan komponenterna, dock är koden knappt dokumenterad/kommenterad vilket bidrar till att majoriteten av koden är oförståelig.

Vi måste överväga och diskutera om vi vill spendera tid för att förstå koden eller att skapa ett eget system för kommunikationen.

2.3 Delsystem3

Delsystem3 består av elcyklarna.

Elcyklarna som vi kommer att använda i projektet är byggda av OffCourse.

I cyklarna finns det ett kretskort som innehåller information om batteriliv samt den totala sträcka cykeln har färdats. När cykeln befinner sig i facket skickas denna information till com-modulerna via en IR-länk. Kretskorten innehåller även ett unikt serienummer vilket talar om för com-modulen vilken cykel som är parkerad i facket.

Eftersom cyklarna redan är färdiga kommer vi inte ändra på dessa, vi kommer fortfarande använda oss av cyklarna för att testa resterande delsystem.

3. Teori

3.1 Enkortsdator

Enkortsdatorer är kompletta datorer byggda på ett kretskort där de finns en processor, minne samt övriga funktioner som krävs för att skapa en funktionell dator. Då komponenterna är svagare än i en vanlig dator är enkortsdatorerna mer passande för mindre projekt som inte kräver lika mycket processorkraft[4].

Beroende på vad det är för projekt man ska jobba med bör man välja en enkortsdator som passar bäst till projektet.



Figur2



Figur3

BeagleBone-black: BeagleBone som man ser i figur2, är en enkortsdator som är utvecklad av Texas Instruments[5]. BeagleBone har ett 4gb flashminne inbyggt i kretskortet[6], BeagleBone använder operativsystemet Debian[8]. Detta möjliggör för användaren att fritt välja programmeringsspråk och IDE.

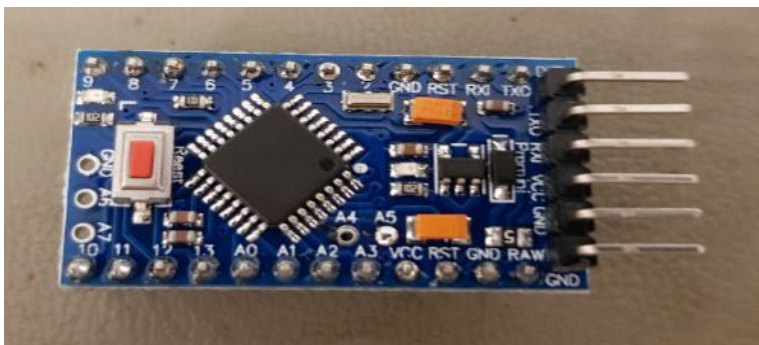
Fördelar med att använda en BeagleBone är att den är open-source, har 92 pins som kan användas fritt av användaren samt är kompatibel med ett stort antal mjukvara system[7].

Nackdelarna med BeagleBonen är att det inte finns lika mycket lättillgänglig information och guider för att lösa problem som uppstår, samt att BeagleBone har en mindre Community jämfört med sin konkurrent Raspberry Pi.

Raspberry pi 3 B: Raspberry Pi som man ser i figur3, är en enkortsdator som är utvecklad av the Raspberry Pi Foundation. Raspberry kräver ett externt SD-kort som minne för att fungera[10], operativsystem består av Raspian som är ett Linuxbaserat operativsystem[9]. Precis som för BeagleBonen är det möjligt för användaren att fritt välja programmeringsspråk och IDE.

Fördelarna med Raspberry Pi är att den är väldigt nybörjarvänlig. Det går snabbt och enkelt att ta sig igenom en installation och påbörja ett projekt. Detta tack för den stora community de har och all information som lätt kan hittas på nätet[12]. Inbyggd bluetooth och Wi-Fi är också ett stort plus[11].

Nackdelen är att man måste köpa till ett SD-kort för att använda produkten. Samt att Raspberry Pi inte har en lika stor variation av mjukvara som den är kompatibel med om man jämför med BeagleBonen[11].



Figur4

Arduino pro mini: Arduino som man kan se i figur4, är ett mikrokontrollerkort. Den är inte lika kraftfull som de övriga enkortsdatorerna och man får mindre funktioner för pengarna men den har andra egenskaper som utmanar de andra två enkortsdatorerna ovan[13].

Arduino har ett eget webbutvecklingsverktyg som kallas Arduino Create som är de rekommenderade verktyget för att programmera Arduinos[15].

I Arduino Create använder man arduinos egna språk som använder ett C/C++ bibliotek som kallas Wiring[13]. Arduino gör det lätt att kompilera och ladda upp koden direkt på Arduinon man använder.

Wiring är bara ett av många bibliotek man kan hämta och använda, Arduino har flertal bibliotek som förenklar sammankoppling med andra komponenter. Även Arduino har en stor Community och är open-source[14].

3.2 LAMP

Lamp är en modell som består av fyra subsystem och har fått sitt namn av Linux, Apache, MySql och Php, som tillsammans bildar en fullständig webbplattform som baseras på fri programvara och är open-source[16].

Linux är ett operativt system, ett operativsystem(os) underlättar användningen av en dator genom att fungera som en länk mellan maskinvaran och applikationerna som användarna använder[17].

I en Raspberry Pi används en Linux distribution som kallas Raspian. Det innebär att Raspian är baserat på Linux programvaran[9].

Apache http Server är utvecklad av Apache Software Foundation. Apache är världen mest använda webbserver, anledningen för den stora populariteten är att apache är gratis att använda samt att den finns tillgänglig för många operativsystem b.la Windows och Linux. Användarna har möjligheten att använda webbservern fritt eftersom källkoden är öppen kan användarna modifiera och anpassa webbservern utan större hinder[18].

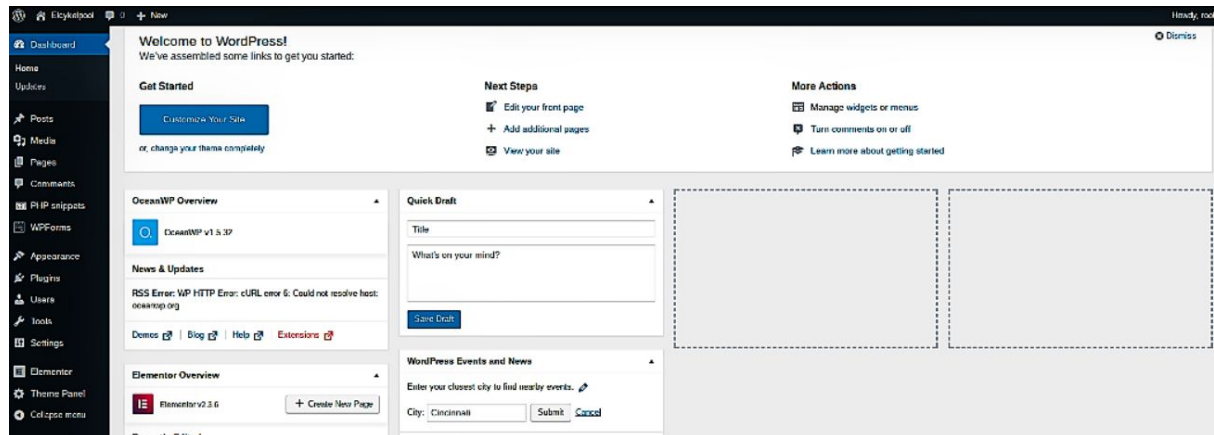
MySql är ett programspråk som används för att designa, skapa, hantera och uppdatera databaser. Används av stora företag som Google, Facebook, Youtube osv[19].

PHP är ett skriptspråk som används på webbservrar för att driva webbapplikationer dynamisk. Vilket innebär att PHP är ett bra verktyg för att sammanlänka en databas med en webbapplikation[20].

3.3 WordPress

WordPress skapades 2003 av Matt Mullenweg och har sedan dess expanderat stort inom marknaden. Idag används WordPress på 60 miljoner[22] webbsidor vilket motsvarar 24%[21] av alla webbsidor idag.

Målet för Matt är att influensera webben med ett verktyg som är tillgängligt för alla individer samt är open-source där alla har rätt till att se på källkoden och bidra till utvecklingen av WordPress[23].



Figur5

I figur5 kan man se sidan som kallas Dashboard, i Dashboard finns alla verktyg och inställningar som behövs för att skapa en hemsida.

När man trycker på Pages får man upp en liten editor. Det är i editorn man skapar sidorna, man kan välja att skriva HTML kod direkt i editorn eller den mer populära metoden där man hämtar Themes.

Themes är färdiga layouter som redan är designade och har färdiga funktioner i sig. Där finns hundratals att välja mellan och de är lätta att implementera. Till exempel vill man skapa en blogg hämtar man ett theme som innehåller strukturen för en blogg, likadant gör man ifall man vill skapa en resesida[24]. Trots att mycket är färdigt har man fortfarande möjligheten att ändra på hela designen, man kan ta bort funktioner och lägga till andra som t.ex. knappar, genom att bara dra och släppa objekt till editorn. Man kan även lägga till PHP kod för att få önskad respons av objekten på sidan.

Plugin är en mjukvara som innehåller en grupp av funktioner som kan underlätta eller bidra med helt nya funktioner för användaren.

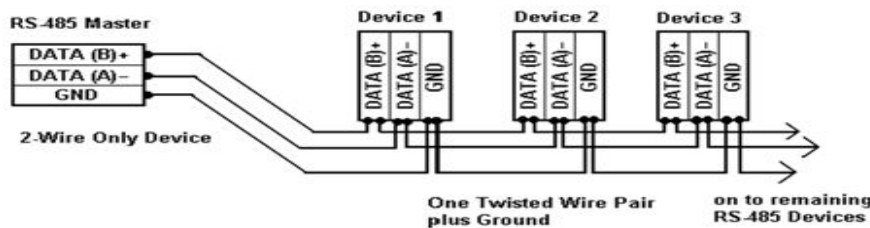
Vill användaren till exempel kunna skriva PHP kod i WordPress editorn behöver användaren bara att installera rätt plugin för problemet[25].

3.4 RS485

RS-485 är en standard som definierar elektriska egenskaper mellan sändare och mottagare i en seriell kommunikation. RS-485 utgör grunden till de vanligaste industriella kommunikationsprotokollen så som Modbus RTU, Optomux, PROFIBUS DP och DH-485[26].

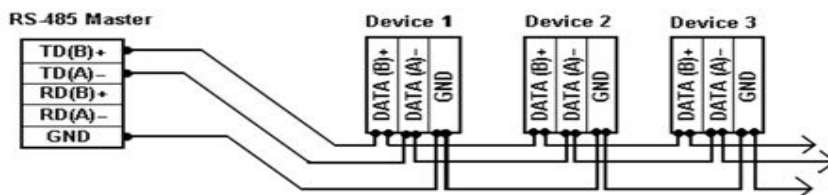
Den äldre versionen av RS-485 kallas RS-232. RS-232 kan bara prata med en enhet i ett avstånd på max 15 meter. RS-485 kan till skillnad från sin äldre versionen täcka ett avstånd

upp till 1200 meter, stödja upp till 32 enheter, samt färdas informationen snabbare i RS-485. RS-485 har även högre tolerans mot buller som kan orsaka problem i kommunikationen. RS-485 kräver att en av enheterna är Master medans de andra enheterna får en unik id. Mastern måste skicka ut ett kommando innan den kan få någon data att läsa av från de olika enheterna[28].



Figur6

RS-485 kan användas som Two Wire eller Four Wire. Two Wire som man kan se ovan i figur6, består av tre kablar B+, A- och GND. B+ och A- skickar samma data dock så skickas A- som en inverterad B+. Detta gör man för att kunna säkerställa att inga störningar har skett när ett meddelande har anlänt genom att jämföra B+ med A-[27].



Figur7

Four Wire har fem kablar se figur7, TD(B+), TD(A-),RD(B+),RD(A-) och GND. Four Wire har en liknande process som Two Wire dock så används TD(B+) och TD(A-) som sändare till de olika enheterna och RD(B+) och RD(A-) används som mottagare för de olika enheterna på så sätt ser alla enheter bara Masterns meddelanden och inte de övriga enheternas respons[27].

3.5 Kommunikation protokoll



Figur8

Modbuss RTU som nämns i 3.2 är ett seriellt kommunikationsprotokoll som används för att överföra information mellan olika elektriska enheter. Modbuss består av två olika enheter

master och slaves, där de vanligtvis bara är en master och ett flertal slaves, en master kan kommunicera med upp till 247 slaves.

Systemet fungerar på så sätt att mastern skickar ett meddelande till en slave som finns i systemet sedan inväntar mastern respons från slaven, varje slave har en unik identifikation som är ett nummer mellan 1-247[29].

I figur8 kan vi se hur ett meddelande från mastern ser ut de första bitarna är de unika slav idn, function code förklarar vilken funktion en slave ska använda på datan (se figur10), data innehåller informationen som en slave ska ta hand om och CRC/Error check säkrar så fel upptäcks och meddelas till mastern[30].

Coil/Register Numbers	Data Addresses	Type	Table Name
1-9999	0000 to 270E	Read-Write	Discrete Output Coils
10001-19999	0000 to 270E	Read-Only	Discrete Input Contacts
30001-39999	0000 to 270E	Read-Only	Analog Input Registers
40001-49999	0000 to 270E	Read-Write	Analog Output Holding Registers

Figur9

Slaven kan lagra information i fyra olika tabeller. Två av tabellerna förvarar diskreta värden, dessa kallas Coils. De andra två förvarar analoga värden och kallas register. Både Coils och Register har en read-only tabell och en read-write tabell. Coilsen har en storlek på 1 bit och register på 16 bitar de både får en data adress mellan 0000-270E där de lagrar datan, Se figur9[29].

Most Common Modbus Function Codes		
Function Code	Action	Table name
01 (01 hex)	Read	Discrete output coils
05 (05 hex)	Write single	Discrete output coil
15 (0F hex)	Write multiple	Discrete output coils
02 (02 hex)	Read	Discrete output contacts
04 (04 hex)	Read	Analog input contacts
03 (03 hex)	Read	Analog output holding registers
06 (06 hex)	Write single	Analog output holding register
16 (10 hex)	Write multiple	Analog output holding registers

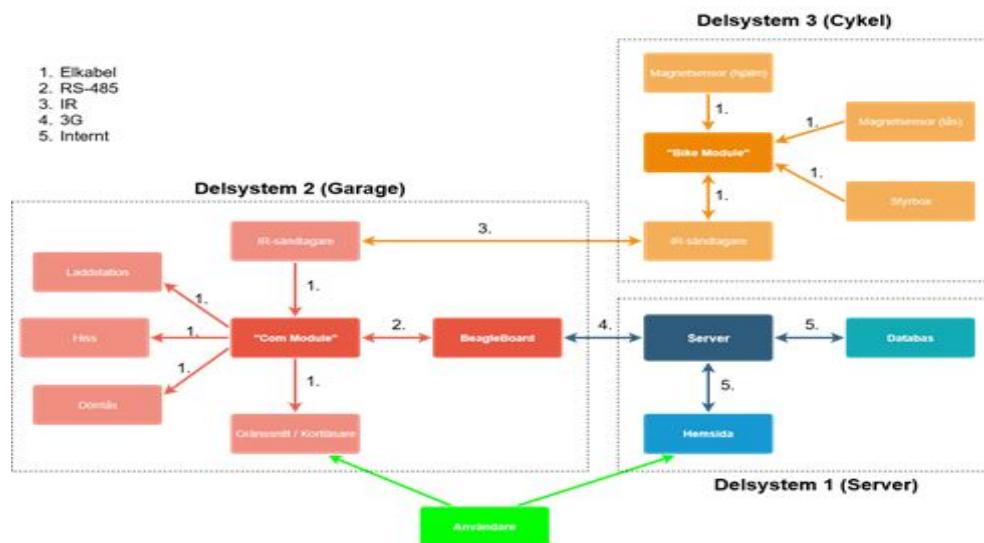
Figur10

4. Metod

4.1 Dokumentation och kommunikation

För dokumentationen användes det Google Docs där alla projektmedlemmar har tillgång till att läsa och redigera samtliga filer som ligger i mappen Elcykel. Som kommunikationsmedel används Facebooks Messenger där vi skapade en grupp för att kunna kommunicera med varandra.

4.2 Delsystem



Figur11

För att förenkla utvecklingen av projektet delades projektet upp i tre delsystem, se figur 11. Fördelarna av detta är att arbetet kan delas upp effektivt vilket bidrar till att de olika grupperna samt individerna kan fokusera på olika aspekter i systemen. I slutet gäller det bara att sammankoppla delsystemen till en slutprodukt.

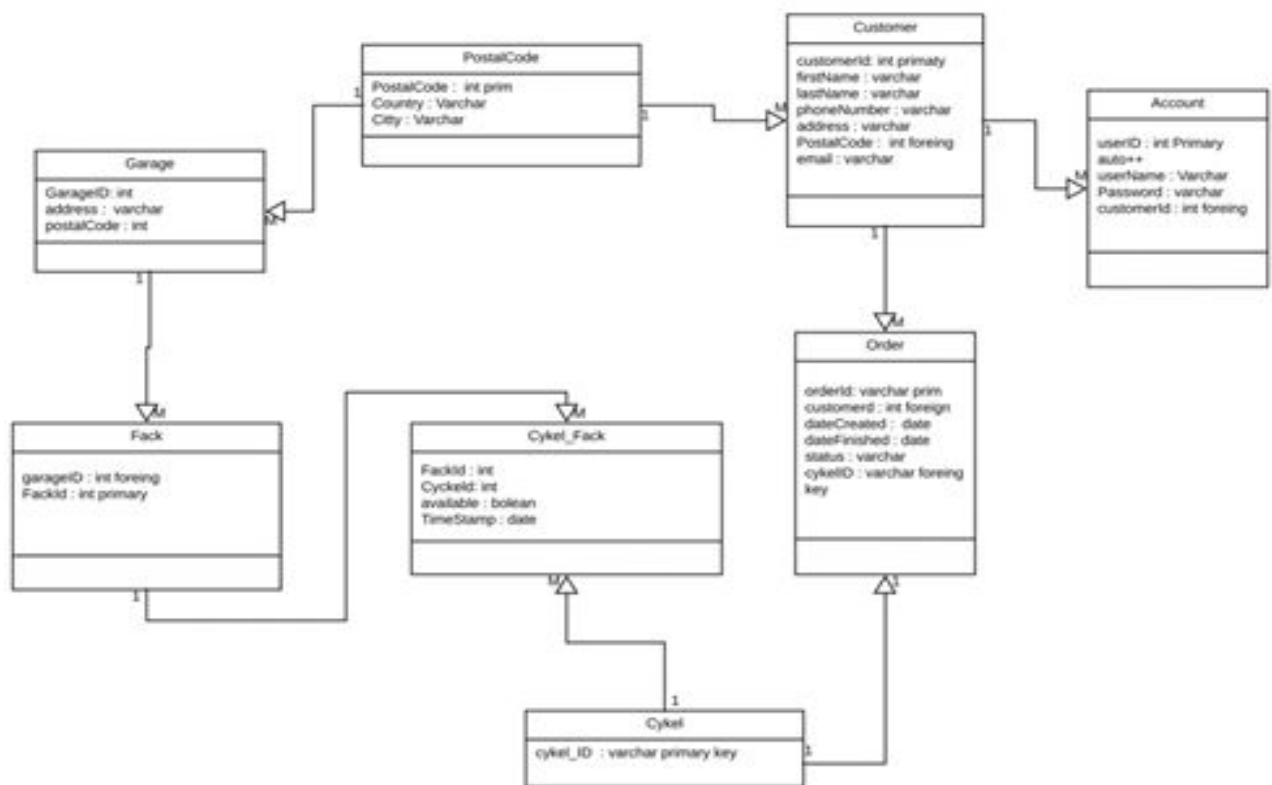
4.3 Val av komponenter

Befintlig enkortsdator var en BeagleBone black. Vi valde att kolla på en konkurrent till BeagleBone, vilket är Raspberry Pi. Beskrivning samt Fördelar och nackdelar för både komponenter hittas i kapitel 3.1. Raspberry Pi produkten vi valde som konkurrent var en Raspberry Pi 3b. Tillslut blev de Raspberry Pi 3b som vann då vi kände att fördelarna av denna produkt passade vårt behov för att lösa projektet.

De stora fördelarna med Raspberry Pi 3b som fick oss att välja produkten är att den är väldigt nybörjarvänlig och har massvis med information på nätet, allt från om hur man kommer igång med produkten till lösningar på specifika problem. Flera sidor är till och med dedikerade för att hjälpa till med problem som man kan stöta på t.ex. Raspberry Pi forum[31], RaspBerry Pi officiella hemsida[12], PiMyLifeUp[32].

Detta bidrog till att de var lätt att komma igång med produkten och ställa in samt sammankoppla den med andra verktyg.

4.4 Utveckling av databas



Figur12

Delsystem1 fick delas in i mindre subsystem som hemsida, databas och server, se figur12.

Databasen var de första subsystem som utvecklades. Diagram gjordes först och sedan kodades databasen så att den skulle likna diagrammet.

Språket vi valde för att koda databasen var MySql som beskriv i kapitel 3.2. Databasen kodades i Command Prompten på Raspberry Pi men för att enkelt kunna följa förändringarna som sker i databasen visuellt installerades PhpMyAdmin(hänvisas till kapitel 3.2 för beskrivning samt figur16 i resultat för bild) som sedan kopplades till databasen.

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);
```

Figur13. I exemplet ovan skapas en tabell, där primary key sätts som OrderID. Relationen skapas med Persons tabellen genom att man skapar en foreign key PersonID som refererar till primary keyn i Persons som också heter PersonID.

4.5 Utveckling av Hemsida

Efter att ha installerat alla verktyg som krävdes för att bygga upp en LAMP-modell, se kapitel 3.2. Påbörjades förstudier inom PHP,MySQL samt hur man sammankopplar och använder WordPress med resterande verktyg/system.

Uppbyggnaden av hemsidan påbörjades med att ladda ner plugin som beskrivs i kapitel 3.3. De plugins som användes för att bygga upp hemsidan är följande Elementor, Wp_Forms och PhpSnippet.

Elementor används för att underlätta uppbyggnaden av hemsidan.

Med Elementor behöver man inte skriva html kod i editorn utan man kan enkelt hämta färdiga mallar som kallas Themes som beskrivs i kapitel 3.3.

Efter att man valt ett theme är nästa steg att få funktionerna att fungera som till exempel knappar. Det är här Phpsnippet kommer in i bilden. Med hjälp av Phpsnippet skrevs php kod för alla knappar och funktioner på hemsidan.. Koden lades sedan enkelt till i WordPress editorn.










Wp_Form användes för att underlätta hanteringen av att registrera användare. Eftersom vi ska utveckla en elcykelpool behövde vi kunna registrera användare på hemsidan och uppdatera databasen med de användare som registrerade sig. I Wp_Form fanns de en mall som kallas Wp_Users som innehåller alla funktioner vi behövde registrera, logga in samt spara och hantera informationen om användarna i en databas.

Genom att integrera vår databas med Wp_Users databas, fick vi enkelt i gång en hemsida där användarna kunde snabbt och enkelt registrera sig.

Utan att förlora kontroll över den information vi behövde hantera i vår databas.

4.6 Script

Två separata script fick kodas ett för funktioner som fanns på WordPress så som knappar, login, boka cykel osv. Scriptet som finns i WordPress körs endast då användaren kallar på någon av dessa funktioner där den sedan uppdaterar databasen utifrån vilken användare som är inloggad samt vilken funktion användaren använt. Det andra scriptet körs konstant i bakgrunden så fort Raspberry Pi sätts igång. Detta script söker hela tiden efter signaler från garaget. Signalen består av en etta eller en nolla beroende på om cykeln är tillgänglig eller inte. För fullständig redovisning av kod för båda script se appendix 7.1.

← T →				fackID	cykelID	available	lastupdated
<input type="checkbox"/>	 Redigera	 Kopiera	 Radera	1	1	1	2018-12-21 14:21:18
<input type="checkbox"/>	 Redigera	 Kopiera	 Radera	2	2	0	2018-12-22 16:33:48
<input type="checkbox"/>	 Redigera	 Kopiera	 Radera	3	3	0	2018-12-22 16:28:39

Figur14. Cykel 1 och två har uppdaterats som otillgängliga i databasen. Efter att de har blivit bokade och scriptet i wordpress har körts.

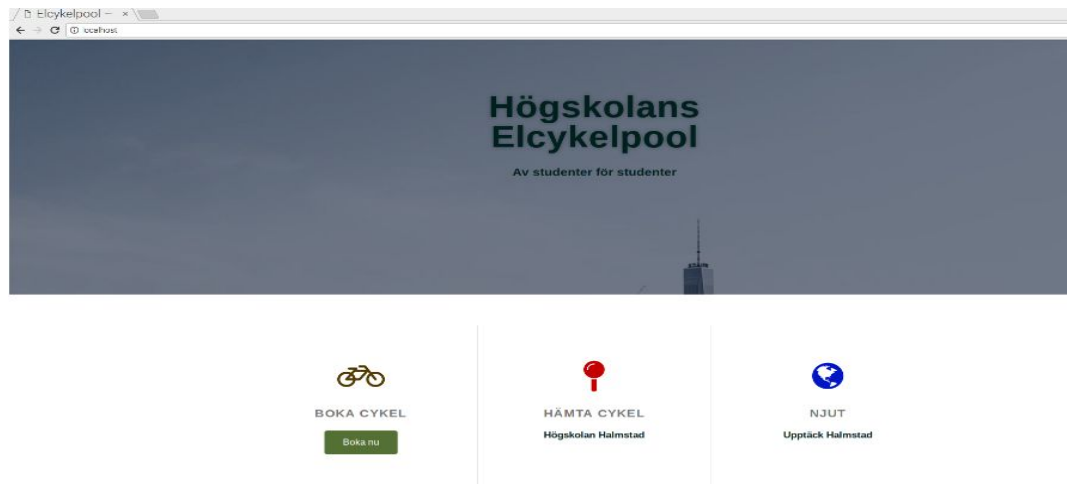
4.7 Gränssnitt

Gränssnitten som används för att kommunicera mellan de olika delsystem kallas Modbus som beskrivs i kapitel 3.5. Hänvisar till Johannes Hellgrens rapport[33] för en djupare beskrivning om vilka metoder som användes för att lösa kommunikationen. .

5. Resultat

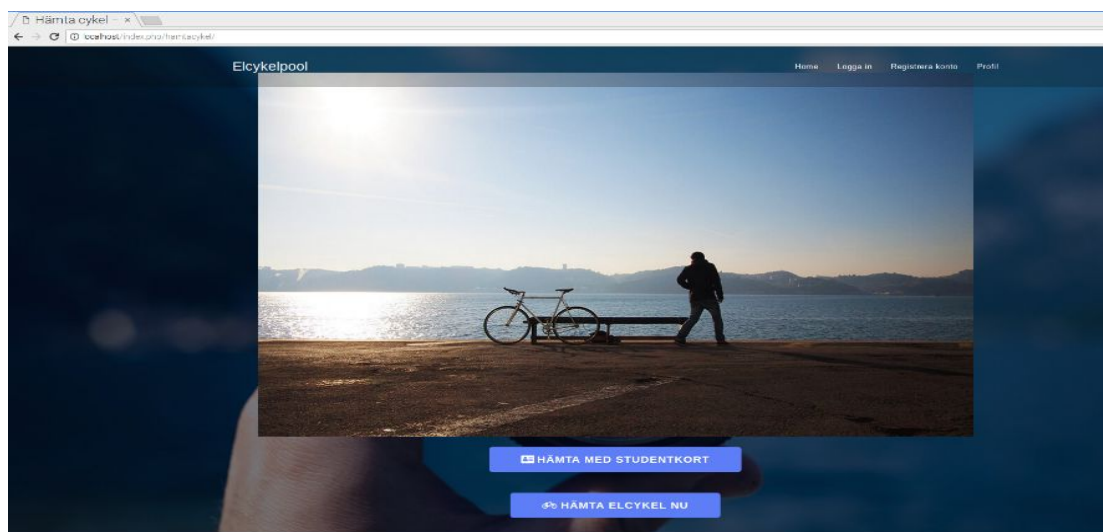
I figur15 ser vi den slutgiltiga hemsidan som gjordes med hjälp av wordpress och php. Användarna kan interagera med knapparna för att registrera sig, logga in, ändra sin profil samt boka cykel. Vilket följer de krav som specificerats i elcykelpoolens kravspecifikation[34].

Elcykelpool Göran Trivic



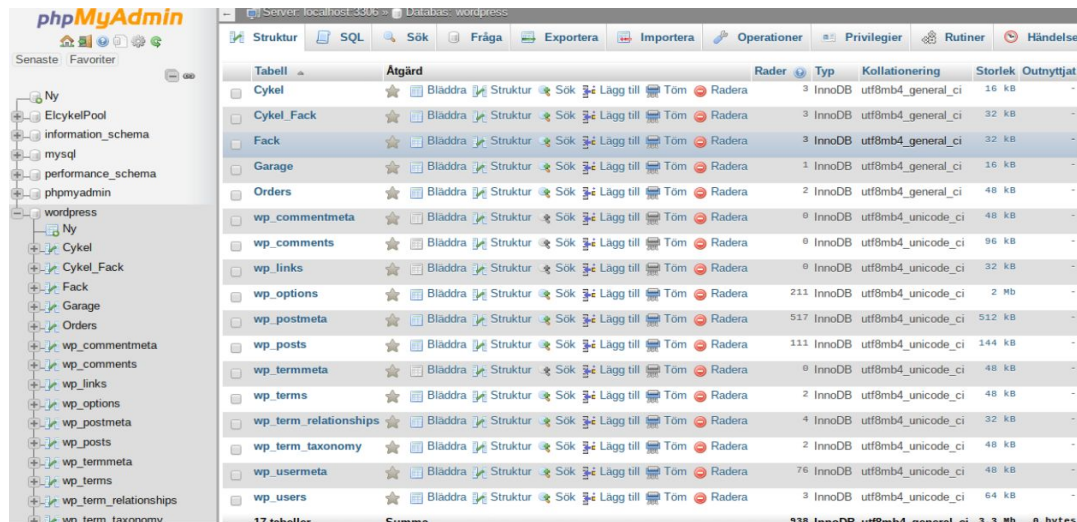
Figur15. Startside för elcykepoolen.

När användaren trycker på boka cykel kommer följande sida upp som syns i figur16. Användaren kan nu välja att hämta ut cykel på plats med mobilen eller att hämta ut den med student id. När användaren gjort sitt val kommer de upp information om vilket fack nummer och cykel nummer användaren blivit tilldelad. Endast hämta elcykel nu knappen är fullständigt implementerad.



Figur16. Val för att hämta ut cykel.

I figur17 ser vi resultaten av sammankopplingen mellan vår databas och Wp_Users databas. Databasen skapades med MySQL.



Tabell	Ätgärd	Rader	Typ	Kollationering	Storlek	Outnyttjat
Cykel	Bläddra Struktur Sök Lägg till Töm Radera	3	InnoDB	utf8mb4_general_ci	16 KB	-
Cykel_Fack	Bläddra Struktur Sök Lägg till Töm Radera	3	InnoDB	utf8mb4_general_ci	32 KB	-
Fack	Bläddra Struktur Sök Lägg till Töm Radera	3	InnoDB	utf8mb4_general_ci	32 KB	-
Garage	Bläddra Struktur Sök Lägg till Töm Radera	1	InnoDB	utf8mb4_general_ci	16 KB	-
Orders	Bläddra Struktur Sök Lägg till Töm Radera	2	InnoDB	utf8mb4_general_ci	48 KB	-
wp_commentmeta	Bläddra Struktur Sök Lägg till Töm Radera	0	InnoDB	utf8mb4_unicode_ci	48 KB	-
wp_comments	Bläddra Struktur Sök Lägg till Töm Radera	0	InnoDB	utf8mb4_unicode_ci	96 KB	-
wp_links	Bläddra Struktur Sök Lägg till Töm Radera	0	InnoDB	utf8mb4_unicode_ci	32 KB	-
wp_options	Bläddra Struktur Sök Lägg till Töm Radera	211	InnoDB	utf8mb4_unicode_ci	2 MB	-
wp_postmeta	Bläddra Struktur Sök Lägg till Töm Radera	517	InnoDB	utf8mb4_unicode_ci	512 KB	-
wp_posts	Bläddra Struktur Sök Lägg till Töm Radera	111	InnoDB	utf8mb4_unicode_ci	144 KB	-
wp_termmeta	Bläddra Struktur Sök Lägg till Töm Radera	0	InnoDB	utf8mb4_unicode_ci	48 KB	-
wp_terms	Bläddra Struktur Sök Lägg till Töm Radera	2	InnoDB	utf8mb4_unicode_ci	48 KB	-
wp_term_relationships	Bläddra Struktur Sök Lägg till Töm Radera	4	InnoDB	utf8mb4_unicode_ci	32 KB	-
wp_term_taxonomy	Bläddra Struktur Sök Lägg till Töm Radera	2	InnoDB	utf8mb4_unicode_ci	48 KB	-
wp_usermeta	Bläddra Struktur Sök Lägg till Töm Radera	76	InnoDB	utf8mb4_unicode_ci	48 KB	-
wp_users	Bläddra Struktur Sök Lägg till Töm Radera	3	InnoDB	utf8mb4_unicode_ci	64 KB	-
17 tabeller	Summa	938	InnoDB	utf8mb4_general_ci	3.3 MB	0 bytes

Figur17

6. Diskussion

Resultaten är förvånansvärt bra jämfört med den tid vi fick för att utveckla projektet. Under kursens gång har man lärt sig att trots hur bra planering man än gör, så är det alltid något som kan gå fel. Under förstudien var vi duktiga på att ta kontakt med varandra och företag som AES som var inblandade i utvecklingen av ElMob projektet. Vi slutförde kravspecifikationen, riskanalys och projektplan tidigt och var i fas med tidsplaneringen.

Det var under denna period vi skulle påbörja förstudierna dock så hade vi inte tillgång till garaget förens slutet av oktober. Koden som satt i BeagleBonon fick vi först efter 1 november. Efter att vi fick se koden som fanns till för BeagleBonon insåg vi att vi fick ändra vår kravspecifikation och projektplan. Detta skedde så sent som 7 november.

Trots de hann vi leverera en slutprodukt som är bra. Detta med hjälp av många timmars slit och bra samarbete. I slutändan när man tänker efter är de nog bättre att vi ändrade planeringen, eftersom projektet blev mer lärorikt för oss alla. Detta eftersom vi själva fick planera, välja och utföra alla moment i projektet. Det fanns inget som var klart som vi kunde använda, utan allt gjordes av oss projektmedlemmar.

Självklart finns de delar som kan förbättras ifall man hade mer tid. Detta är t.ex. RFID taggen så att man skulle kunna hämta ut cykeln med ett student id. Även slutföra funktionerna i hemsidan så att användaren ska kunna ha valet av att hämta ut cykel med student id.

Referenser

- [1]”Kursplan utvecklingsprojekt Halmstad Högskola,”[Online],Available:
http://www.hh.se/sitevision/proxy/arstudent/sokkursplan.4677.html/svid12_70cf2e49129168da015800074301/752680950/se_proxy/utb_kursplan.asp?kurskod=ET6008&revisionsnr=4&format=pdf. [Används 2018-11-15].
- [2]”Lindholmen.se,”[Online], Available:
<https://www.lindholmen.se/nyheter/unik-elcykelpool-invigs-pa-lindholmen-science-park>.
[Använd 2018-11-15]
- [3]”Goteborg.se,”[Online], Available:
https://goteborg.se/wps/portal/enhetssida/Innovation-och-utveckling-far-framtidens-mobilitet-i-Gateborg/elmob/!ut/p/z1/hY0xC8lwFIR_Tde8FxskuqWDhbZ7axZpJaaFNK-k0YC_3oKToHjb3X3HgYYOtO8fk-3jRL53mz_r_UXsSuSV4LWU_IQqP1SyKAQvGw7tP0BvNf6QQqhAW0fD-0r5IZcWdDA3E0xg97DFY4zLeswww5Qs0TWGXalOcNvk5HWCN0nCvcPRvTqhc96T6Y/dz/d5/L2dBISEvZ0FBIS9nQSEh/. [Använd 2018-11-15].
- [4]”EnkortsDator,”[Online], Available: https://en.wikipedia.org/wiki/Single-board_computer. [Använd 2018-11-17].
- [5]”BeagleBoard,”[Online],Available: <https://en.wikipedia.org/wiki/BeagleBoard>. [Använd 2018-11-17].
- [6]”BeagleBoard.org,”[Online],Available: <https://beagleboard.org/black>. [Använd 2018-11-17].
- [7]”BeagleBoard.org,”[Online], Available: <http://beagleboard.org/Support/bone101>. [Använd 2018-11-17].
- [8]”Wikipedia,” [Online], Available: <https://en.wikipedia.org/wiki/Debian>. [Använd 2018-11-18].
- [9]”Wikipedia”, [Online], Available: <https://en.wikipedia.org/wiki/Raspbian> . [Använd 2018-11-18].
- [10]”Wikipedia”, [Online], Available: https://en.wikipedia.org/wiki/Raspberry_Pi. [Använd 2018-11-18].
- [11]”Raspberrypi.org”, [Online],Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Använd 2018-11-18].
- [12]”Raspberrypi.org”,[Online],Available: <https://www.raspberrypi.org/community/>. [Använd 2018-11-18]
- [13]”Wikipedia,”[Online], Available: <https://sv.wikipedia.org/wiki/Arduino>. [Använd 2018-12-2].
- [14]”Aruino Forum”[Online], Available: <https://forum.arduino.cc/>. [Använd 2018-12-2].
- [15]”Arduino Create,”[Online], Available: <https://www.arduino.cc/en/Main/Create>. [Använd 2018-12-2].
- [16]”Wikipedia,”[Online], Available :[https://en.wikipedia.org/wiki/LAMP_\(software_bundle\)](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) . [Använd 2018-12-13].
- [17]”Wikipedia,”[Online], Available : <https://en.wikipedia.org/wiki/Linux> . [Använd 2018-12-13].
- [18]”Wikipedia,”[Online], Available : https://en.wikipedia.org/wiki/Apache_HTTP_Server
[Använd 2018-12-13].

- [19]"Wikipedia,"[Online], Available : <https://en.wikipedia.org/wiki/MySQL>. [Använd 2018-12-13].
- [20]"Wikipedia,"[Online], Available : <https://en.wikipedia.org/wiki/PHP> . [Använd 2018-12-13].
- [21]"Linkedin", [Online],Available:
<https://www.linkedin.com/pulse/billion-dollar-tech-company-offices-email-glenn-leibowitz> . [Använd 2018-12-12].
- [22]"Wikipedia", [Online], Available:
https://en.wikipedia.org/wiki/WordPress#cite_note-5 . [Använd 2018-12-17].
- [23]"Forbes", [Online],Available:
<https://www.forbes.com/sites/jicolao/2012/09/05/the-internets-mother-tongue/#16e52b7569f6> . [Använd 2018-12-17].
- [24]"Developer",[Online],Available: <https://developer.wordpress.org/themes/getting-started/what-is-a-theme/> . [Använd 2018-12-12].
- [25]"Wpbeginner", [Online],Available: <https://www.wpbeginner.com/glossary/plugin/> . [Använd 2018-12-17].
- [26]"Wikipedia", [Online],Available:
<https://en.wikipedia.org/wiki/RS-485> . [Använd 2018-12-23].
- [27]"B+B SmartWorx", [Online],Available:
<http://www.bb-elec.com/Learning-Center/All-White-Papers/Serial/RS-485-Connections-FAQ.aspx> . [Använd 2018-12-23].
- [28]"Youtube", [Online],Available: <https://www.youtube.com/watch?v=3wgKcUDIHuM> . [Använd 2018-12-23].
- [29]"Schneider-electric", [Online],Available: <https://www.schneider-electric.co.in/en/faqs/FA168406/>. [Använd 2018-12-23].
- [30]"Youtube", [Online],Available: <https://www.youtube.com/watch?v=JBGaInI-TG4>. [Använd 2018-12-25].
- [31]"Raspberrypi.org", [Online],Available: <https://www.raspberrypi.org/forums/>. [Använd 2018-12-25].
- [32]"Pymyifeup.com", [Online],Available: <https://pimylifeup.com/>. [Använd 2018-12-25].
- [33]"Johannes Hellgren rapport,"[Online], [Available]:
https://docs.google.com/document/d/1lms5AmxCYTCgA35DNgzWCUP7yGBivoyZI_NBD6jB9k/edit. [Använd 2018-01-05].
- [34]"Kravspecifikation,"[Online],[Available]:
<https://docs.google.com/document/d/1VsQZzi5Hs4sC4LynfgmDDf5kIVTrvyGbNaqofvSQaC4/edit>. [Använd 2018-01-05].

7. Appendix

7.1.

```
$servername = "localhost";
$username = "root";
$password = "elcykel";
$wordpress = "wordpress";
$conn = new mysqli($servername,$username,$password,$wordpress);

if($conn->connect_error){
    die("Connection failed: " . $conn->connect_error);
}
$current_user = wp_get_current_user();
$checkbikes = "SELECT cykelID,fackID FROM Cykel_Fack WHERE";
$resultcheckbikes = $conn->query($checkbikes);
if($resultcheckbikes->num_rows>0){
    while ($row = $resultcheckbikes->fetch_assoc()) {
        $cykelid = $row["cykelID"];
        $fackid = $row["fackID"];
    }
}else{
    echo " ..";
}
$sql = "INSERT INTO Orders (userID,active,cykelID) VALUES";
$result = $conn->query($sql);
if($result === TRUE){
    echo "Fack " . $fackid . " har öppnats, du har blivit tilldelad";
}else{
    echo " Ingen cykel är tillgänglig ";
}
$conn->close();
```

Bilaga1. Bild på fullständiga WordPress scriptet.


```
1 <?php
2
3 require_once dirname(__FILE__) . '/../Phpmodbus/ModbusMaster.php';
4 $servername = "localhost";
5 $username = "root";
6 $password = "elcykel";
7 $wordpress = "wordpress";
8 $conn = new mysqli($servername,$username,$password,$wordpress);
9
10 // Create Modbus object
11 $modbus = new ModbusMaster("192.168.1.177", "TCP");
12
13 try {
14     // FC 1
15     $recData = $modbus->readCoils(1, 62, 1);
16     echo"$recData";
17 }
18 catch (Exception $e) {
19     // Print error information if any
20     echo $modbus;
21     echo $e;
22     exit;
23 }
24 if($conn->connect_error){
25     die("Connection failed: " . $conn->connect_error);
26 }
27 while(TRUE){
28     $checkexists = "SELECT * FROM Orders";
29     if($conn->query($checkexists)){
30         $checkorder = "SELECT userID,active,cykelID FROM Orders WHERE active = TRUE";
31         $result = $conn->query($checkorder);
32         if($result->num_rows>0){
33             while ($row = $result->fetch_array()) {
34                 $userid = $row["userID"];
35                 $sac = $row["active"];
36                 $cykelid = $row["cykelID"];
37                 $flag = TRUE;
38                 echo $cykelid;
39             }
40         }else{
41             echo "can't reach ";
42         }
43         $result->close();
```

Bilaga2. Bild på första delen av bakgrund scriptet.

```
    if($sac == TRUE && $flag == TRUE){
        $result = "UPDATE Cykel_Fack SET available = FALSE WHERE cykelID = $cykelid";
        if($conn->query($result)){
            echo " cykelid är inte längre tillgänglig";
        }
    }else{
        echo " cykelid hittas inte";
    }
    $userid->close();
    $sac->close();
    $result->close();
    $flag->close();
}
if($recData == TRUE){
    $updaterafack = "UPDATE Cykel_Fack SET available = TRUE WHERE cykelID = 2";
    if($conn->query($updaterafack)){
        echo"fack uppdaterad";
    }else{
        echo"kunde inte uppdatera fack";
    }
    $updateraorder = "UPDATE Orders SET active = FALSE WHERE cykelID = 2 AND userID = $userid";
    if($conn->query($updateraorder)){
        echo"order är nu arkiverad";
    }else{
        echo"order kunde inte arkiveras";
    }
}
$cykelid->close();
}
// Print read data
echo "</br>Data:</br>";
var_dump($recData);
echo "</br>";
$conn->close();
```

Bilaga3. Bild på andra delen av bakgrund scriptet.