

# Utveckling av mobilapplikation för utbytesstuderanden

Valentin Rocha

Examensarbete för ingenjörsexamen (YH)-examen

Utbildningsprogrammet för informationsteknik

Vasa 2019



## EXAMENSARBETE

Författare:	Valentin Rocha
Utbildning och ort:	El- och automationsteknik, Vasa
Inriktningsalternativ:	Informationsteknik
Handledare:	Susanne Österholm

Titel: Utveckling av mobilapplikation för utbytesstuderanden

---

Datum 17.5.2019

Sidantal 28

---

### Abstrakt

Detta examensarbete har gjorts åt internationella enheten vid Yrkeshögskolan Novia i Vasa. Syftet med detta examensarbete var att skapa och planera en engelskspråkig testversion av en mobilapplikation, med målgruppen inkommande utbytesstuderanden till Jakobstad. I mobilapplikationen ska utbytesstuderanden hitta information som är relevant för deras studier i Jakobstad såsom info om campus, boende, försäkringar, visum, hälsa, kultur, tutorskap, händelser, staden och kontaktuppgifter. All information ska gå att uppdateras.

Valet av teknik har varit helt fri och i detta arbete beskrivs hur en WordPress-server har använts som back-end del och hur en Apache Cordova baserad mobilapplikation använts för att skapa en front-end applikation som lätt kan tas ned från nedladdningstjänster.

Resultatet blev en mobilapplikation för Android telefoner som kan laddas ner från Google Play nedladdningstjänsten. Upprätthållaren av mobilapplikationen kan uppdatera allt innehåll i mobilapplikationen.

---

Språk: svenska

Nyckelord: Novia, mobilapplikation, internationell enhet

---

## OPINNÄYTETYÖ

Tekijä:	Valentin Rocha
Koulutusohjelma ja paikkakunta:	Sähkö- ja automaatioteknikka, Vaasa
Suuntautumisvaihtoehto:	Tietotekniikka
Ohjaaja:	Susanne Österholm

Nimike: Mobiilisovelluksen kehittäminen vaihto-opiskelijoille

---

Päivämäärä 17.5.2019

Sivumäärä 28

---

### Tiivistelmä

Tämä opinnäytetyö on tehty Yrkeshögskolan Novian kansainväliselle yksikölle. Opinnäytetyön tarkoituksena oli luoda mobiilisovelluksen englanninkielinen testiversio, kohderyhmänä Pietarsaareen saapuvat vaihto-opiskelijat. Mobiilisovelluksesta vaihto-opiskelijat löytävät tietoja, jotka ovat merkitykselliset opintojen kannalta Pietarsaareessa, kuten infoa kampuksesta, asumisesta, vakuutuksista, viisumeista, terveydestä, kulttuurista, opetusohjelmista, tapahtumista, kaupungista ja yhteystiedoista.

Tekniikan valinta on ollut vapaa ja tässä työssä on kuvattu miten WordPress-palvelin on käytetty back-end-osana ja miten Apache Cordova-pohjainen mobiilisovellus on käytetty luodakseen front-end-sovelluksen, jonka helposti voi ladata latauspalveluista.

Tuloksena oli Android puhelimille asennettava mobiilisovellus, jonka voi ladata Google Play-latauspalvelusta. Mobiilisovelluksen ylläpitäjä voi helposti päivittää mobiilisovelluksen kaikki tiedot, mikä oli suurin vaatimus.

---

Kieli: ruotsi

Avainsanat: Novia, mobiilisovellus, kansainvälinen yksikkö

---

## BACHELOR'S THESIS

Author: Valentin Rocha  
Degree Programme: Electrical and Automation Engineering, Vaasa  
Specialization: Information Technology  
Supervisor: Susanne Österholm

Title: Development of Mobile Application for Exchange Students

---

Date May 17, 2019

Number of pages 28

---

### Abstract

This thesis was done on behalf of the international unit of University of Applied Sciences Novia. The purpose of this thesis was to create and plan an English mobile test information application, with the target group being incoming exchange students to Jakobstad. In this mobile application the exchange students will find information that is relevant to their studies in Jakobstad. Information about the campus, living, insurance, visa, health, culture, tutoring, events, the city and contact information. All information needs to be updatable.

The choice of technology was completely open, and this thesis describes how a WordPress-server has been used to create a back-end part, and how an Apache Cordova based mobile application has been used to create a front-end application that easily can be downloaded from downloading services.

The finished product became a mobile application for Android devices that can be downloaded from the Google Play service. The administrator can easily update all the information of the mobile application.

---

Language: Swedish

Key words: Novia, mobile application, international unit

---

# Innehållsförteckning

1	Inledning.....	1
1.1	Uppdragsgivare.....	1
1.2	Uppdrag.....	1
2	Teori.....	2
2.1	WordPress.....	2
2.1.1	Historia.....	3
2.1.2	WordPress.com.....	3
2.1.3	WordPress.org.....	4
2.1.4	Administratörspanelen.....	4
2.1.5	Anpassningsmenyn.....	5
2.1.6	Inlägg.....	5
2.1.7	Sidor.....	6
2.1.8	Teman.....	7
2.1.9	Plugins.....	7
2.2	Apache Cordova.....	8
2.2.1	Cordova arkitektur.....	9
2.2.2	Cordova WebView.....	9
2.2.3	Cordova Web App.....	10
2.2.4	Cordova plugins.....	10
2.2.5	Plattformsberoende arbetsflöde (CLI).....	10
2.2.6	Plattform-centrerat arbetsflöde.....	10
2.3	Google Play.....	11
3	Utvecklingsmiljö.....	11
3.1	Windows.....	11
3.1.1	Cordova.....	11
3.1.2	Android SDK.....	11
3.1.3	Windows-kommandotolken.....	12
3.1.4	Emulering.....	12
3.1.5	Android-enheter.....	13
3.2	Webbhotell.....	14
3.2.1	Novias nätverk.....	14
4	Utförande.....	14
4.1	Krav.....	14
4.2	Val av teknik.....	15
4.3	Planering.....	15
4.4	Implementering av WordPress-delen.....	17

4.5	Implementering av Cordova-delen.....	21
4.6	Testgrupp .....	23
4.7	Publicering.....	23
4.8	APK-filsignatur .....	24
5	Resultat .....	25
6	Diskussion.....	27
7	Källförteckning.....	28

## Figurförteckning

Figur 1.	WordPress hemsida. [3] .....	2
Figur 2.	Administratörspanelen.....	4
Figur 3.	Anpassningsmenyn. ....	5
Figur 4.	Inlägg.....	6
Figur 5.	Sidor. ....	6
Figur 6.	Teman. ....	7
Figur 7.	Plugins. ....	8
Figur 8.	Cordova applikationsarkitektur. [12] .....	9
Figur 9.	Windows-kommandotolken. ....	12
Figur 10.	Google Pixel emulator med Android 8.1.....	13
Figur 11.	scenario 1.....	15
Figur 12.	Scenario 2. ....	16
Figur 13.	scenario 3.....	17
Figur 14.	Ett exempel på hur Curver ser ut. [16].....	18
Figur 15.	Ett exempel på hur ett inlägg ser ut.....	18
Figur 16.	inläggssortering i administratörspanelen.....	19
Figur 17.	Inläggsmenyn i applikationen. ....	20
Figur 18.	Anpassningsmenyn.....	20
Figur 19.	mappstruktur i Cordova.....	21
Figur 20.	exempel på InAppBrowser-kod. ....	21
Figur 21.	Cordova-events i html-filen. ....	22
Figur 22.	Cordova-events i JavaScript-filen.....	22
Figur 23.	Cordova InAppBrowser-händelse.....	23
Figur 24.	Webbplatsen Google Play Console. [19] .....	24
Figur 25.	Signaturskommando-exempel i Cordova. ....	24
Figur 26.	APK med signatur som har genererats.....	25
Figur 27.	mobilapplikationen på Google Play tjänsten. ....	25
Figur 28.	Upprätthållarens administratörspanel. ....	26
Figur 29.	Slutresultatet.....	26

# 1 Inledning

Detta examensarbete handlar om skapandet av en mobilapplikation för internationella enheten vid Yrkeshögskolan Novia. Utbytesstuderanden som kommer på utbyte till Jakobstad, kan ladda ner applikationen till sin mobiltelefon via nedladdningstjänster.

## 1.1 Uppdragsgivare

Uppdragsgivare till projektet är Yrkeshögskolan Novias internationella enhet som har hand om koordineringen av internationell verksamhet. Verksamheten leds av en chef för internationella ärenden och dessutom har enheten två koordinators. För varje utbildning finns en kontaktlärare som koordinerar och ansvarar för den internationella verksamheten samt har hand om praktiska ärenden. Teamet består av; kontaktlärare, chefen för internationella ärenden, koordinators, Noviums tutorsekreterare och studeranderepresentanter, vilka arbetar tillsammans i frågor som gäller internationella ärenden. [1]

Att studera utomlands betyder att man gör en del av sin examen hos en utländsk högskola. Varje utbildningsprogram har partnerskolor som det samarbetas med. Syftet med utlandsstudier är att ge studeranden ett internationellt perspektiv på det egna ämnesområdet samt att fördjupa sin examen. [2]

Novias internationella koordinators, Camilla Moliis, har fungerat som kontaktperson angående detta examensarbete och vi har träffats under flera tillfällen för att diskutera applikationen.

## 1.2 Uppdrag

Uppdraget har varit att planera och utveckla en engelskspråkig testversion av en mobilapplikation för inkommande utbytesstuderanden till Jakobstad. I applikationen ska ingå information om campus, boende, försäkringar, visum, hälsa, kulturchock, tutorskap, överlevnadsbox, händelser, staden, kurser, kontaktuppgifter osv.

Den studerande ska enkelt kunna ladda ner applikationen innan ankomst till Jakobstad. Applikationen ska innehålla lättläst information som den studerande behöver innan och under utbytet. Applikationen ska ge vägledning under hela utbytet. Upprätthållaren ska enkelt kunna mata in, radera och uppdatera information.

Applikationen byggs för att svara mot det alltmer digitala samhället. Samtidigt är målet att få en användarvänlig och tilltalande informationskälla som enkelt går att uppdatera. Innehållet i applikationen motsvarar den tidigare tryckta Survival Guiden men framställs i modernare tappning och med ett intressant grepp.

En testgrupp bestående av utbytesstuderanden i Vasa har skapats för att göra applikationen passande för målgruppen.

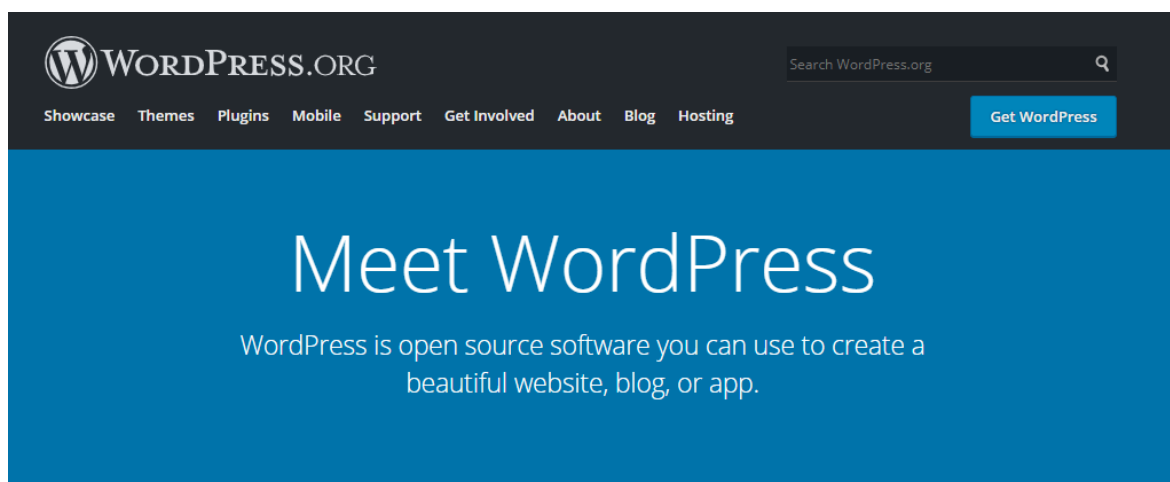
Arbetet med applikationen har skett under våren 2019. Applikationen tas i bruk senast augusti 2019.

Jag har helt och hållet fått välja själv hur jag utför uppdraget. Valet av tekniker och sätt som har använts i detta arbete har varit mina egna beslut.

## 2 Teori

I detta kapitel kommer det att tas upp allmän teori om teknikerna som har använts i detta examensarbete. Teorin kommer bland annat att beskriva vad WordPress är och hur det är möjligt att en WordPress-server har använts som en back-end-tjänst samt med vilken teknik mobilapplikationens front-end är uppbyggd.

### 2.1 WordPress



**Figur 1. WordPress hemsida. [3]**

WordPress är ett ramverk för att skapa blogg- och innehållshanteringssystem. WordPress är baserat på skript-språket PHP och använder sig av databashanteringssystemet MySQL.



Ramverket är helt gratis med öppen källkod och fungerar under licensen GPLv2. Verket associeras mest med bloggning men kan användas på olika sätt inom webbutveckling, allting från online-butiker och webbforums till gallerier. WordPress är ett av de mest använda webbverktygen för att skapa webbsidor. Över 70 miljoner webbsidor använder sig av verktyget WordPress och det är det mest använda webbplatsinnehållshanteringssystemet. [4]

WordPress består av två huvuddelar WordPress.org och Wordpress.com, två helt olika plattformar som ofta blandas ihop. I detta slutarbete har WordPress.org använts.

### **2.1.1 Historia**

WordPress grundades år 2003 av Matt Mullenweg och Mike Little. De bestämde sig för att bygga en ny plattform baserad på b2/cafeblog, som var ett sorts bloggverktyg. Den 27 maj 2003 släpptes första versionen av WordPress. 2004 var ett stort år för WordPress. Då introducerades plugin-arkitekturen som gjorde det möjligt för användare att utveckla extensioner för funktionaliteten på sin WordPress webbplats. Samtidigt som plugin-arkitekturen togs i bruk hade en av WordPress största rivaler inom bloggvärden en stor ändring i sina licensvillkor som inte gillades av användarna. Detta ledde till att användarna sökte efter nya sätt att skapa bloggar och detta ledde till att WordPress växte och fick en stor användarkrets. Under åren har WordPress aktivt uppdaterats och sakta men säkert blivit ett av världens mest använda bloggverktyg på internet. WordPress fortsätter att utvecklas till denna dag enligt kraven av de miljontals användarna. [5] [6]

### **2.1.2 WordPress.com**

WordPress.com riktar sig till hobbybloggare och är en tjänst som ägs av WordPress och som man betalar för att använda, detta är inte ett verktyg man laddar ner och skapar en server med. Denna service är gratis i en viss mån, men när man kommer upp till en viss nivå av dataanvändning börjar tjänsten kosta. Denna version av WordPress är en mer begränsad plattform som förbjuder användningen av egen reklam, plugins, egna teman och andra tillbehör. Jämför med att hyra ett hus. [7]

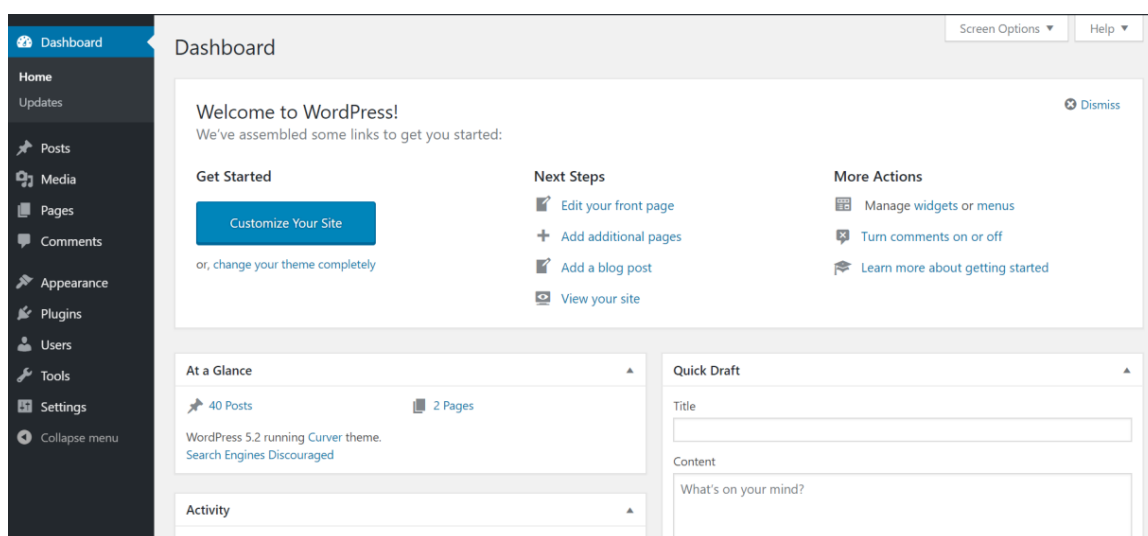
### 2.1.3 WordPress.org

WordPress.org är själva mjukvaran som man laddar ner från WordPress.org och lägger upp på en server för att sedan ha skapat en egen WordPress-baserad webbplats som man har full kontroll över. Man brukar kalla WordPress.org för ”self-hosted WordPress”. Med detta alternativ är man helt fri att göra exakt vad man vill, man är ägaren till webbplatsen. All data är ens egen och man kan göra vad man vill med den. Det är helt upp till upprätthållaren om man vill köra reklam på sidan eller sälja medlemskap, inga avgifter måste betalas till WordPress för att man använder deras tjänst eftersom verktyget är helt gratis och har öppen källkod. [7]

Eftersom webbsidan är helt och hållet ens egen är det upp till upprätthållaren att välja vilket sorts webbhotell som används. Upprätthållaren har allt ansvar och kostnaderna för webbhotellet beror på vilken storlek projektet är. Jämför med att äga ett hus. [7]

### 2.1.4 Administratörspanelen

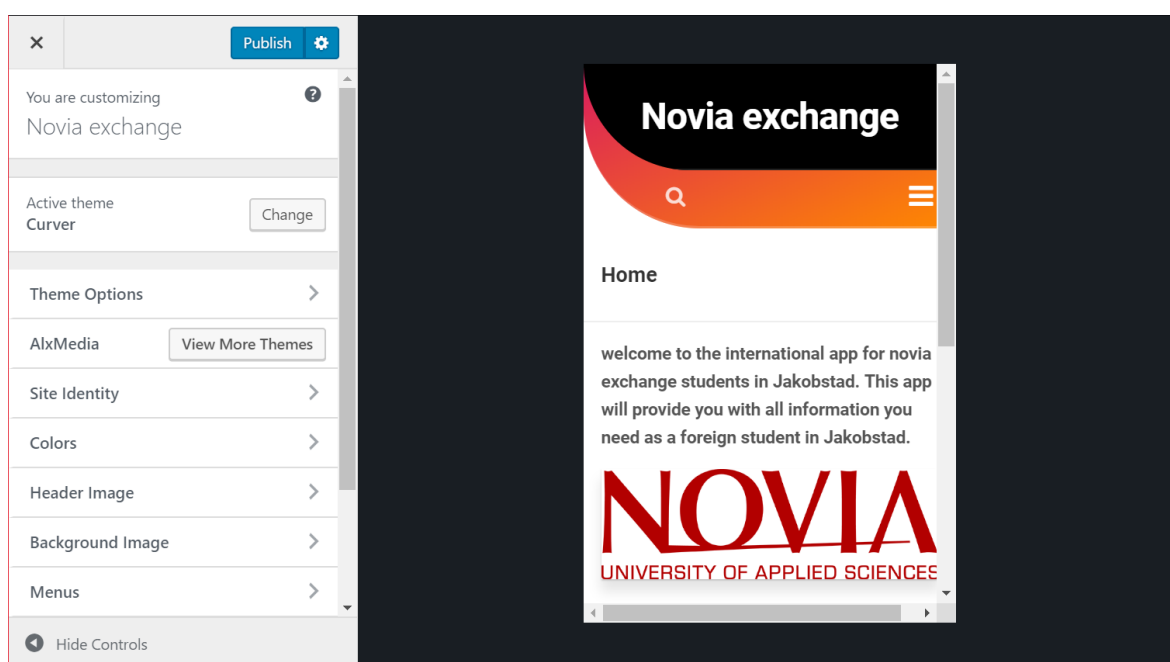
Panelen kallas på engelska ”WordPress admin dashboard” eller ”WP admin panel” och det är här administreringen för webbplatsen sker. Den är en sorts kontrollpanel där man skapar och lägger till funktioner, ändrar teman och hanterar WordPress-webbplatsen. Vid uppbyggnaden av en WordPress.org webbplats, kommer man att spendera tid här. Administratörspanelen ger användaren en simpel överblick av alla detaljer för WordPress-webbplatsen. För att komma åt denna panel måste man efter en lyckad installation logga in med användarnamn och lösenord som användaren skapade under installationen av ramverket. Här kommer man åt alla inlägg, sidor, teman, plugins och mycket mer.



**Figur 2. Administratörspanelen.**

### 2.1.5 Anpassningsmenyn

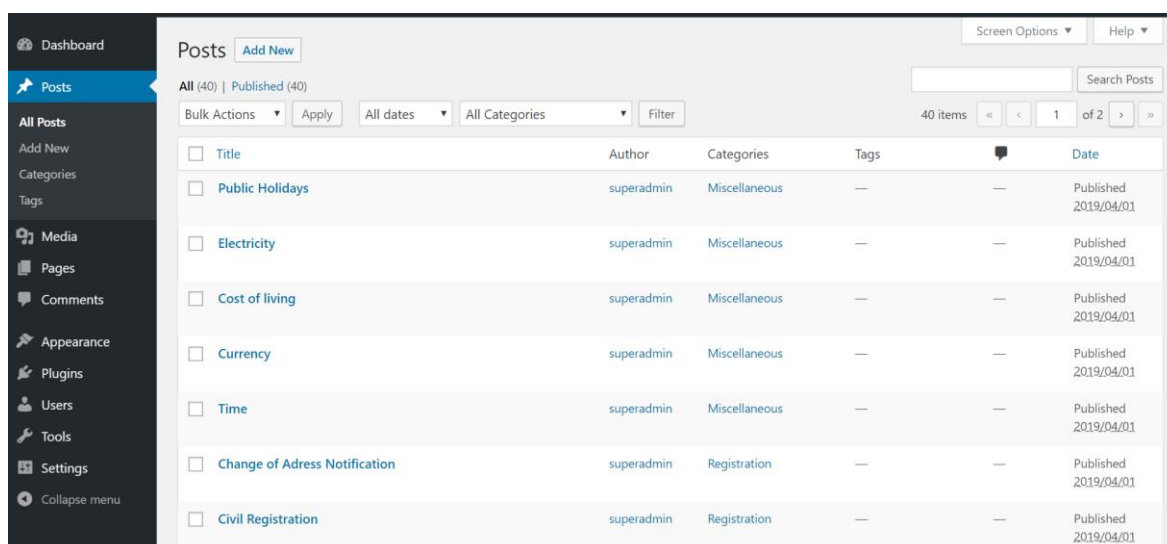
Menyn kallas WordPress customizer och är en snabbmeny inom WordPress-ramverket som man kommer åt efter att man har loggat in och gått till sin webbsida inloggad som administratör. Den är en snabbmeny där man lätt, snabbt och drastiskt kan ändra på utseendet på webbplatsen (se figur 3). Man kan göra ändringar på sidelement såsom sidtitel, bakgrundsbild, menyer, ordningen av inlägg och andra viktiga element som påverkar utseendet. Anpassningsmenyn är beroende av temat som har valts för projektet. Temats utvecklare bestämmer hur mycket man kan anpassa i denna snabbmeny. [8]



**Figur 3. Anpassningsmenyn.**

### 2.1.6 Inlägg

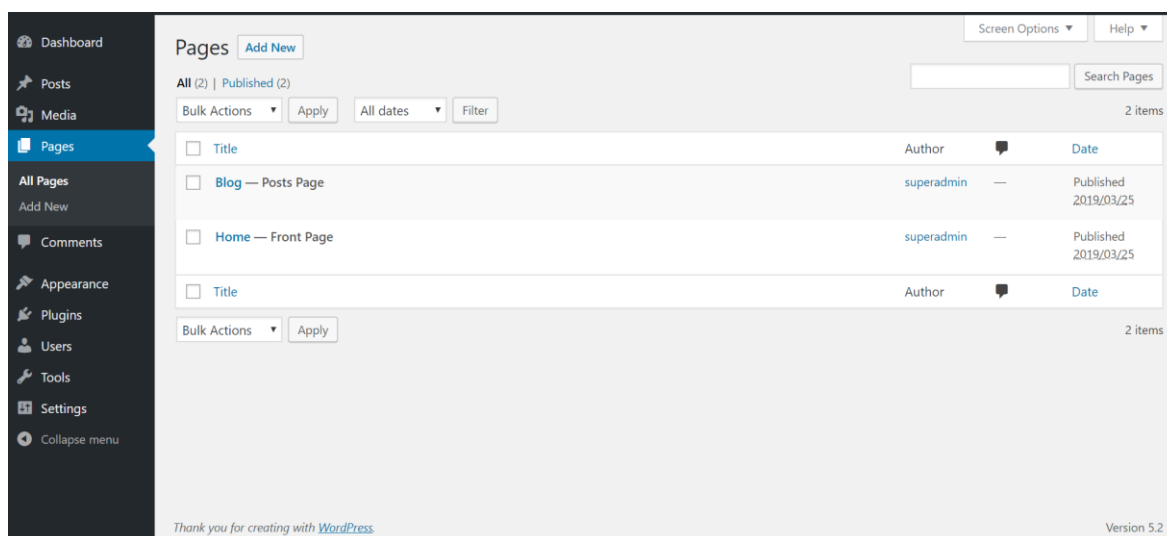
Ett inlägg (post) är sättet man infogar nytt innehåll till en bloggghemsida. När man lägger till ett inlägg väljer man först titel och under skriver man in text. Inläggen är alltid listade i omvänd kronologisk ordning, vilket betyder att de nyaste inläggen kommer högst upp och först på webbsidan. Alla inlägg får också en tid och datum som visar när inlägget publicerades. Naturligtvis lockar blogginlägg till kommentarer och det finns även möjlighet att välja om man tillåter kommentarer i samband med skapandet av ett inlägg. Ett inlägg visar också vanligtvis namnet på skribenten. [9]



**Figur 4. Inlägg.**

## 2.1.7 Sidor

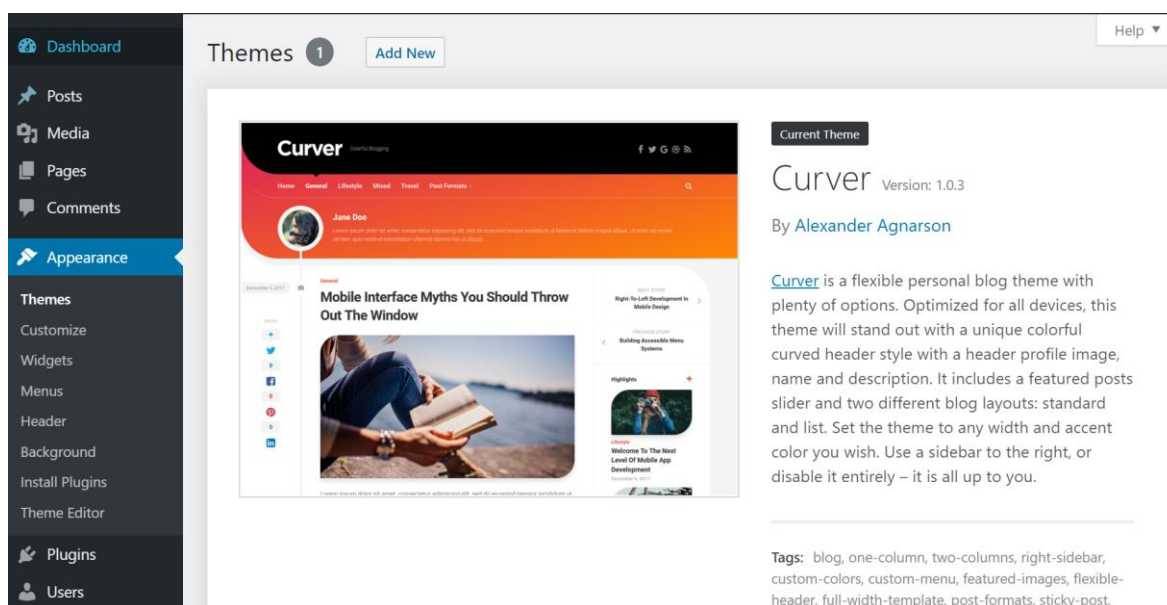
En sida (page) är en sida med ett statiskt innehåll som till exempel en integritetspolicy-sida eller en kontaktsida. Sidorna är inte menade att socialiseras på, vilket innebär att kommentarer inte är ett alternativ här. Sidor har inte heller datum eller skribentnamn. [9]



**Figur 5. Sidor.**

### 2.1.8 Teman

I administratörspanelen kan man hitta en flik som heter Appearance, utseende på svenska. Under den fliken hittar man olika teman. Ett tema är en sorts stilmall som ändrar på utseende, design och egenskaper på webbplatsens sidor. Teman kan editeras. Det finns både gratis och betalda teman. I detta projekt har ett gratis tema använts. Det går också att göra egna modifikationer till ett tema i form av egenskriven kod eller plugins. Ett tema är uppbyggt på tre huvuddelar: filen style.css, WordPress-mallfiler och kodfiler av PHP-format. Alla dessa filer går att editera och man är fri att göra ändringar efter eget behov, men man bör veta vad man gör, eftersom ändringar i temats källkod kan göra temat oanvändbart. De flesta WordPress-teman är responsiva. Responsiv webbdesign betyder att man utvecklar en webbplats på ett sådant sätt att webbplatsen automatiskt ändrar storlek och utseende beroende på bl.a. skärmstorlek så att webbplatsen ser bra ut på alla enheter. [10]

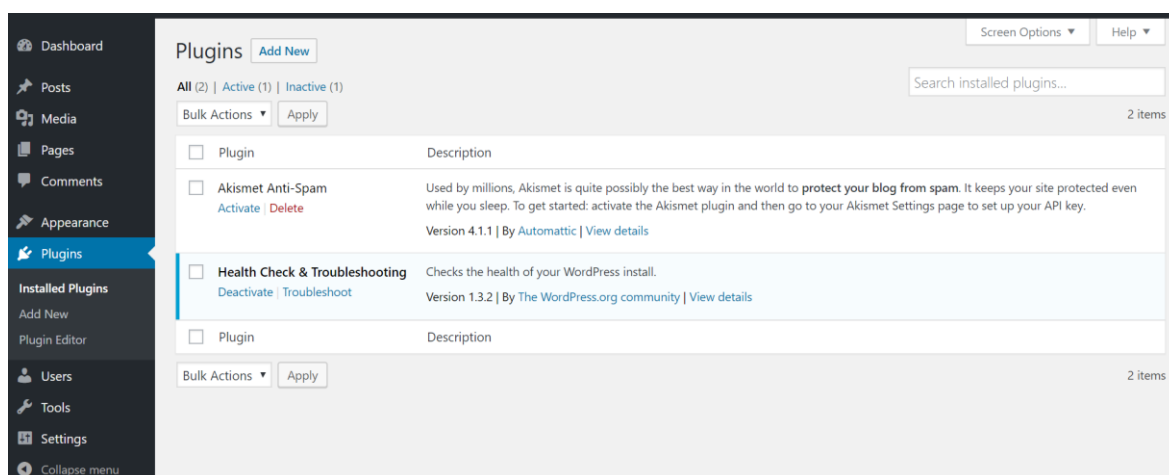


Figur 6. Teman.

### 2.1.9 Plugins

WordPress-plugins är ett sorts plugin som lägger till funktionaliteter till webbsidan. WordPress-plugins skrivs i skriptspråket PHP. Som administratör kan man under fliken ”plugins” hantera vad man vill lägga till, man kan installera och av-installera enligt eget behov. Det finns tusentals plugins som ofta är gratis och som kan skapas av vemsomhelst. Teoretiskt sett finns det inga gränser på vad man kan skapa via ett plugin, vilket leder till att administreringen kan skapa en webbsida efter kundens krav i hög grad. Det finns även risker med att använda ett plugin, eftersom vissa plugins är av lägre kvalitet än andra, vilket kan

leda till att webbplatsen blir betydligt långsammare. Vissa plugins uppdateras inte, vilket kan leda till att pluginet inte kommer att fungera på webbplatsen om WordPress-versionen uppdateras. För att ha hög säkerhet på sin WordPress-webbplats bör man alltid uppdatera WordPress-ramverket till senaste version. Om man inte gör detta är man i riskzon för att webbplatsen blir hittad av webbplatsskanningsverktyg som skannar icke-uppdaterade WordPress-webbplatser. En icke-uppdaterad WordPress-webbplats är extremt utsatt för säkerhetsluckor. Man bör vara försiktig vid valet av plugins och för att ha en pålitlig WordPress-webbplats bör man använda så lite plugins som möjligt. [11]



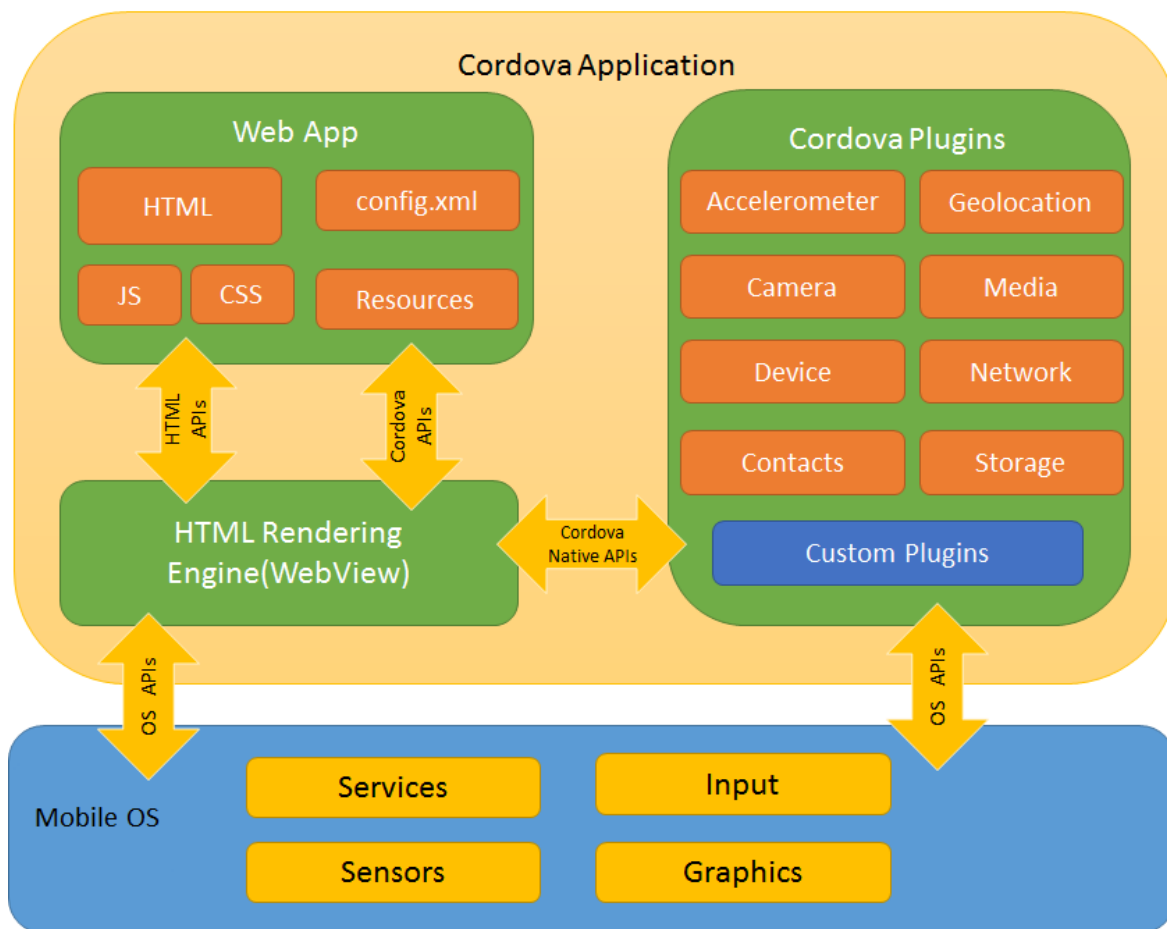
**Figur 7. Plugins.**

## 2.2 Apache Cordova

Apache Cordova är en programvara med öppen källkod för mobilapplikationer som använder sig av några av dom vanligaste webbteknologierna såsom HTML, CSS och JavaScript. Applikationerna exekveras inom omslag (wrappers) som är riktade till varje plattform och som följer Cordova standard för att få tillgång till enhetens funktioner såsom sensorer, data och nätverksstatus. Cordova används av utvecklare som vill skapa en mobilapplikation för flera plattformar utan att vara tvungna att göra om allting för varje plattform samt för att få en slutprodukt som är färdig för distribution i de olika nedladdningstjänsterna för mobilapplikationer. Utvecklare använder även Cordova för att få en äkta mobilapplikation som blandas med ett sorts webbläsarfönster som har tillgång till enheter på systemnivå. [12]

### 2.2.1 Cordova arkitektur

Figur 8 visar en vy över Cordovas applikationsarkitektur.



Figur 8. Cordova applikationsarkitektur. [12]

### 2.2.2 Cordova WebView

Applikationer inom Cordova implementeras ofta som en webbläsarbaserad webbvy inom mobilapplikationen. Detta gör Cordova till ett multiplattform-ramverk, eftersom varje operativsystems webbläsarwebbsidor skiljer sig åt. Detta betyder att då applikationen ändras beroende på vilket operativsystem som den kör på, förblir den interna Cordova webb-vyn densamma. Webbvyer är plattformspecifika. För varje plattform som stöder webbvyn måste man enskilt exekvera dem. För tillfället stöds endast Android och IOS. [12]

### **2.2.3 Cordova Web App**

Applikationen i sig själv är en webbsida som baserar sig på en fil som heter Index.html. Filen stöder CSS och JavaScript samt andra medier som kan behövas för att köra. Applikationen körs i en tidigare nämnd webbvy inom applikationsskalet. I denna del ligger all kod för hela applikationen. [12]

### **2.2.4 Cordova plugins**

Cordova plugins är en viktig komponent i ramverket Cordova. Dessa plugins gör det möjligt för Cordova att kommunicera med inbyggda komponenter. Detta gör att man kan anropa inbyggd kod med JavaScript. Cordova har en uppsättning kärnplugins som gör det möjligt för applikationen att få tillgång till telefonens funktioner som batteri, kamera, kontakter och mera. Förutom Cordovas kärnplugins finns det även tredjepartsplugins som ger möjlighet till andra funktioner. Man hittar alla plugins på Cordovas officiella hemsida, men man kan även utveckla egna plugins. När ett Cordova projekt skapas finns inga plugins färdigt installerade. Detta gäller även för kärnplugins. Man bör vara försiktig vid val av plugins eftersom vissa Cordova-plugins endast stöder vissa plattformar och måste konfigureras olika. [12]

### **2.2.5 Plattformsberoende arbetsflöde (CLI)**

Cordova ger utvecklaren möjlighet att skapa en applikation som är oberoende av plattform. Detta är användbart om man vill ha en applikation som kan köras på så många operativsystem som möjligt. Detta arbetsflöde baserar sig på Cordovas CLI (Cross-plattform workflow), ett verktyg som tillåter utveckling för många plattformar samtidigt. Cordova CLI kopierar det man gör i en plattform och automatiskt implementerar det till de andra plattformarna genom att skapa konfigurationer och bibliotek som är nödvändiga. [12]

### **2.2.6 Plattform-centrerat arbetsflöde**

Detta arbetsflöde används om man utvecklar sin mobilapplikation för en enda plattform eller operativsystem, och bygger på en uppsättning skript på lägre nivå som är skraddarsydda för var plattform. Det är generellt svårare att använda detta arbetsflöde, eftersom utvecklaren hamnar att göra separata byggcyklar och tilläggsändringar för varje plattform. Om man bestämmer sig att mitt i utvecklingen byta från plattformsberoende arbetsflöde till plattform-centrerat arbetsflöde kan man inte gå tillbaka, eftersom plattformsberoende arbetsflödes källkod skriver över plattform-centrerat arbetsflödes källkod. [12]



## **2.3 Google Play**

Tidigare känt som ”Android market” är en onlinebutik ägd av Google, där man kan ladda ner mobilapplikationer till Android-telefoner. Google Play-applikationen är alltid förinstallerad på Android-telefoner och här kan man hitta spel, musik, filmer och andra tjänster. Google Play är plattformen som används i detta examensarbete för distribution av slutproduktens front-end dvs. Cordovaapplikationen.

## **3 Utvecklingsmiljö**

Detta kapitel handlar om vilka miljöer och verktyg som har valts för att utföra projektet.

### **3.1 Windows**

All programvara som har krävts för att köra de olika teknologierna har installerats i miljön Windows 10 på en bärbar dator.

#### **3.1.1 Cordova**

Som tidigare har nämnts i teorin har mobilutvecklingsmiljön Apache Cordova använts för mobilapplikationen. För att kunna köra Cordova behöver man installera Node.js som är av öppen källkod och används ofta med webbservrar som kan exekvera JavaScript-kod utanför en webbläsare, ett verktyg på server-sidan. [13]

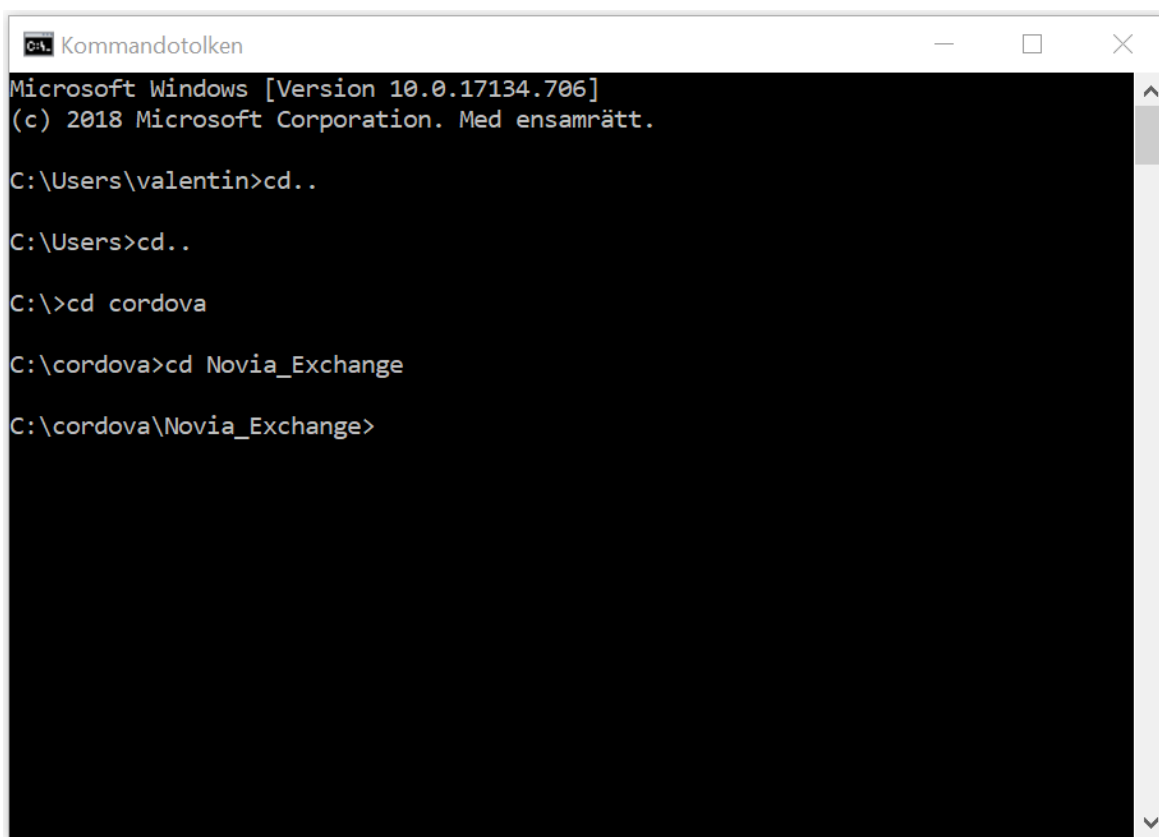
#### **3.1.2 Android SDK**

För att kunna köra och skapa projekt i Cordova i windowsmiljö krävs det att Android SDK (software development kit) är installerat. Android SDK är en form av verktyg som krävs för utveckling av mobilapplikationer, vilket betyder att man måste installera Android Studio. Googles Android Studio är den officiella plattformen för integrerad utveckling, och används för att utveckla applikationer i operativsystemet Android. Ett annat krav för att Android Studio ska fungera korrekt med Cordova är installationen av Java Development Kit (JDK), vilket är ett sorts tillägg till Android SDK-verktyget som man måste installera skilt. [14]

### 3.1.3 Windows-kommandotolken

Windows-kommandotolken är ett verktyg för att tolka kommandon som skrivs in och som är tillgänglig i operativsystemet Windows. Den används för att köra inmatade kommandon eller automatiserade uppgifter. Här körs avancerade administreringsuppgifter, felsökning och allmänna Windows-problem kan fixas här.

Windows-kommandotolken har använts för att navigera i de olika Cordova-projektfilerna samt för att köra alla Cordova-kommandon.



```
Kommandotolken
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. Med ensamrätt.

C:\Users\valentin>cd..

C:\Users>cd..

C:\>cd cordova

C:\cordova>cd Novia_Exchange

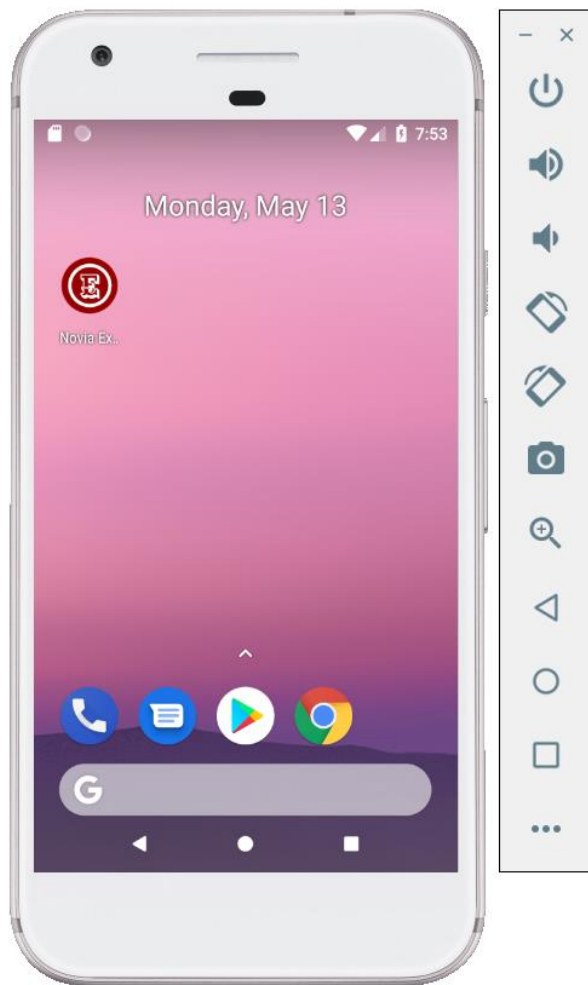
C:\cordova\Novia_Exchange>
```

Figur 9. Windows-kommandotolken.

### 3.1.4 Emulering

För att testa applikationen har Android Studios emuleringsverktyg (AVD) använts. En emulator är ett verktyg för att simulera en mobiltelefon på en dator. En emulator gör det möjligt att testa en applikation utan att behöva äga en riktig enhet. Android Studios emulator erbjuder de flesta funktioner som en riktig Android-telefon har. Med en emulator kan man simulera telefonsamtal, textmeddelanden, telefonens position, olika nätverk och simulera rotation samt andra hårdvarusensorer. En emulator ger stor hjälp vid teststadiet av en mobilapplikation eftersom man kan gratis testa sin applikation på olika telefoner och olika

operativsystemsversioner. En Google Pixel-emulator med operativsystemet Android 8.1 och en Google Nexus S emulator med operativsystemet Android 5.1 har använts. [15]



**Figur 10. Google Pixel emulator med Android 8.1.**

### **3.1.5 Android-enheter**

För att visa applikationen åt uppdragsgivaren installerades applikationen på en Androidtelefon, för att kunna presentera applikationen och framsteg under examensarbetet. Testversioner av applikationen har installerats på några olika Android-enheter.

## **3.2 Webbhotell**

Webbhotell eller webbhosting på engelska, är en tjänst som väljs av webbplatsägaren för att publicera en webbsida till Internet. Ett webbhotell är en tjänst som drivs av ett företag som är värd för olika teknologier som krävs för att sidan ska kunna hittas på Internet av användare.

### **3.2.1 Novias nätverk**

Eftersom Yrkeshögskolan Novia är ägare till webbplatsen som skapas, var önskemålet att ha webbplatsens webbhotell på Novias eget nätverk. Novias ICT-enhet kontaktades och efter ett möte kom man fram till att en WordPress installation är möjlig. Servern är en basinstallerad CentOS 7 som är ett Linux-distributionssystem, som uppdateras med programmeringsspråket PHP 7.3 och databashanteringssystemet Maria DB 10, och nyaste WordPress-server. Allting är installerat i Linuxmiljö. Eftersom en basinstallation av WordPress går snabbt och enkelt att göra utförde skolans ICT-enhet denna installation.

## **4 Utförande**

Detta kapitel berättar hur arbetet utfördes, dvs. hur en WordPress-webbplats använts för att skapa en informationsapplikation.

### **4.1 Krav**

Uppgiften gick ut på att planera och utveckla en engelskspråkig testversion av en mobilapplikation som ska användas av inkommande utbytesstuderanden i Jakobstad. Applikationen ska utvecklas på ett sådant sätt att den går att implementeras också i Raseborg. Applikationen ska vara en användbar informationskälla. En studerande ska kunna ladda ner applikationen innan ankomst till Jakobstad, vilket innebär att den bör distribueras på ett relevant sätt.

Ett viktigt krav i applikationen var att upprätthållaren ska enkelt kunna mata in, radera och uppdatera information, samt att en testgrupp bestående av utbytesstuderanden ska skapas.

## 4.2 Val av teknik

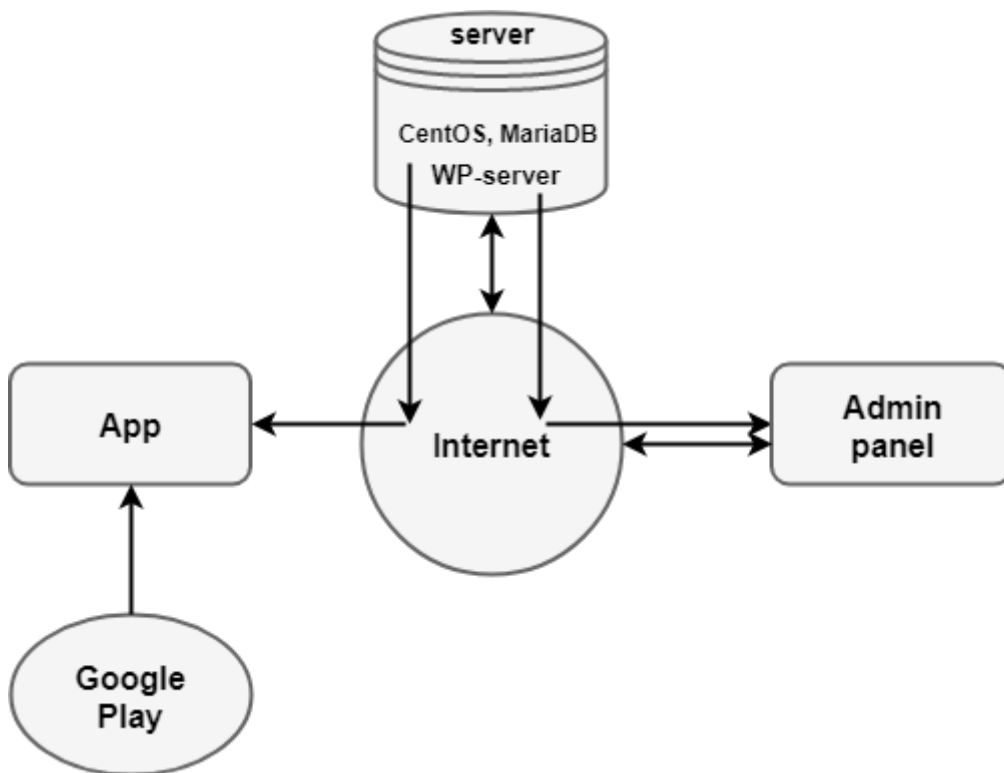
I uppdragsbeskrivningen fanns få specifikationer om hur detta arbete skulle genomföras tekniskt sätt. Det enda som var klart och tydligt från arbetes början var att det skulle vara en mobilapplikation. Utöver det, var valet av teknik helt fri.

## 4.3 Planering

Eftersom valet av teknik var fritt, gick en stor del av arbetet ut på att planera hur det skulle genomföras. Kravet att applikationen ska kunna uppdateras lätt, ledde till att en webb-server av något slag måste installeras (En server är ett datorsystem som betjänar andra system).

Planeringen ledde till att man behövde någon form av applikation som kommunicerar med en webb-server. Med tanke på att applikationen inte har någon form av invecklad struktur, annan än inmatad text som kan uppdateras, kom idén snabbt.

Det lättaste sättet för en upprätthållare att uppdatera informationen, är genom en administratörspanel som kommunicerar via Internet med en WordPress-server. WordPress-servern kommunicerar i sin tur med mobilapplikationen som kan laddas ner från nedladdningstjänsten Google Play (se figur 11).

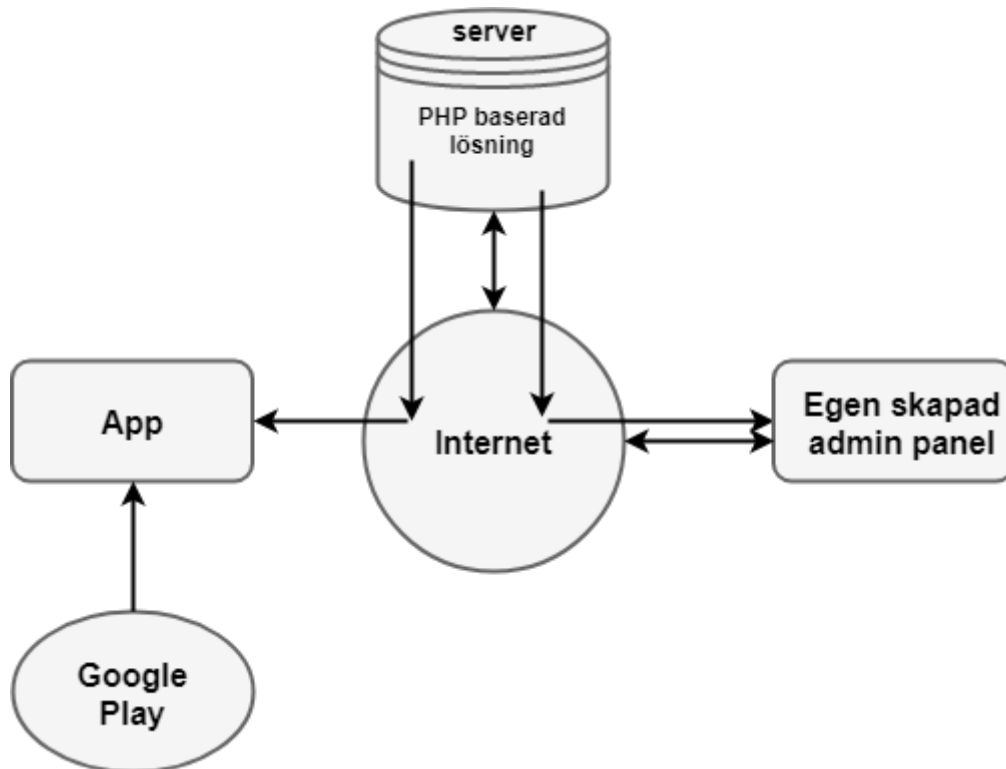


Figur 11. scenario 1.

Sökningen av webbplats-teknik började och WordPress-ramverket hittades snabbt. Det upptäcktes att det finns köptjänster som konverterar WordPress-webbplatser till mobilapplikationer, vilket gav idén att använda en WordPress.org-server och sedan skapa en egen mobilapplikation som kommunicerar med WordPress-webbplatsen. Apache Cordova användes för mobilapplikationen.

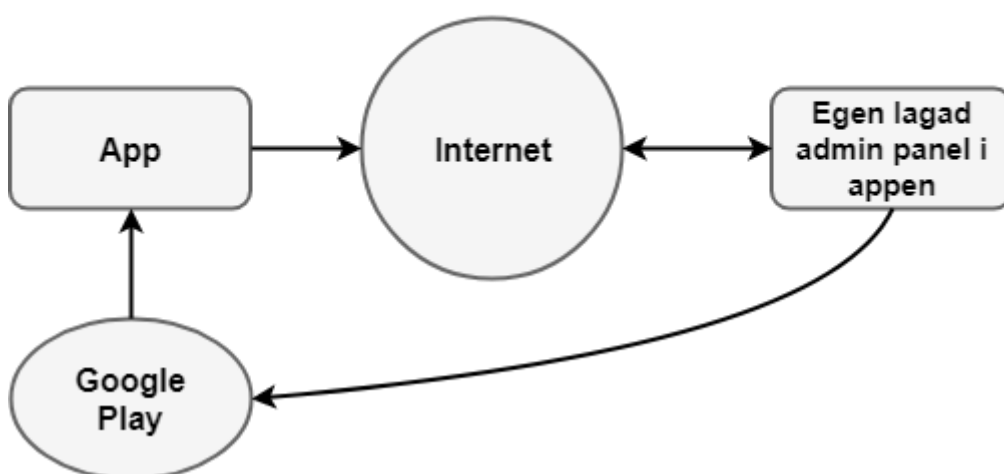
Andra alternativ till hur arbetet skulle genomföras var att inte använda något innehållshanteringssystem och bygga upp allting från början.

Till exempel att skapa en egen administratörspanel som kommunicerar via internet med ett server-program kodat med PHP. På servern skulle det också finnas ett server-program som kommunicerar med mobilapplikationen som är uppladdad till nedladdningstjänsten Google Play (se figur 12). Det skulle ha varit mycket mera krävande att skapa ett vettigt och snyggt uppdateringssystem och slutprodukten skulle aldrig ha blivit så bra som en WordPress-administratörspanel. För att vara säker på att ingen som inte har rättigheter går in och ändrar på innehållet i applikationen, krävs det att någon form av autentiseringssystem sätts i bruk. Att göra detta från grunden skulle ha varit en stor utmaning och en WordPress-server kommer färdigt med denna funktion.



Figur 12. Scenario 2.

En annan idé är att endast skapa en mobilapplikation som inte kommunicerar med en server och när upprätthållaren vill köra en uppdatering, uppdaterar man applikationen endast via Google Play. Detta skulle ha lett till problem eftersom upprätthållaren skulle vara tvungen att känna till mobilutvecklingsmiljön som använts och sedan publicera en uppdatering till tjänsten Google Play (se figur 13).



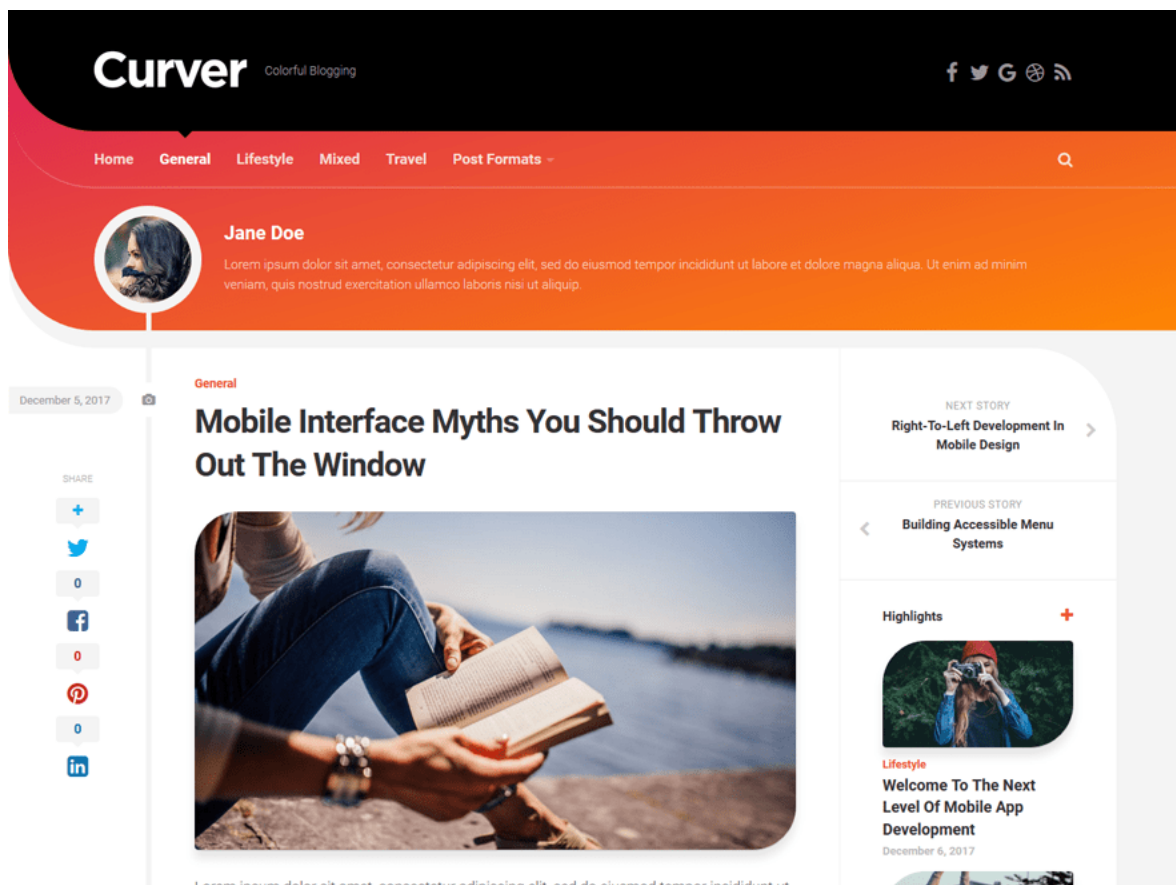
Figur 13. scenario 3.

#### 4.4 Implementering av WordPress-delen

För att skapa en WordPress-webbplats krävs det att en WordPress.org-server är installerad. Novias ICT-enhet skapade en WordPress-installation på Novias eget nätverk som är öppen utåt till Internet.

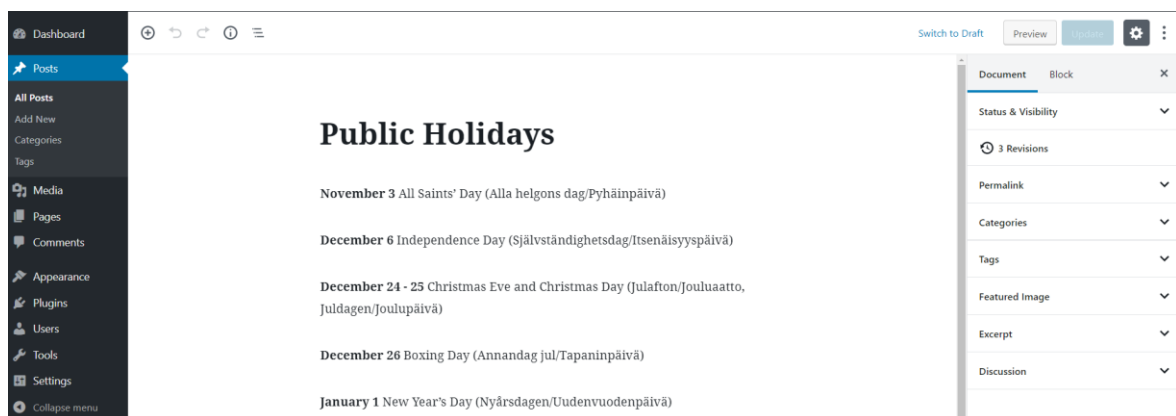
När WordPress-installationen var färdig, skapades ett inloggningsnamn och lösenord för administratören. Ett anti-spamfilter-plugin som blockerar onödig skräppost och ett plugin som kontrollerar hälsan för webbplatsen installerades.

Temat som slutligen valdes är Curver. Curver är ett gratis tema som hittades i WordPress-temakatalog. Curver är ett bloggtema som kan konfigureras, den använder sig av responsiv webbdesign och har en meny som ser bra ut på mobila plattformar. Temat innehåller även en sökmeny. [16]



Figur 14. Ett exempel på hur Curver ser ut. [16]

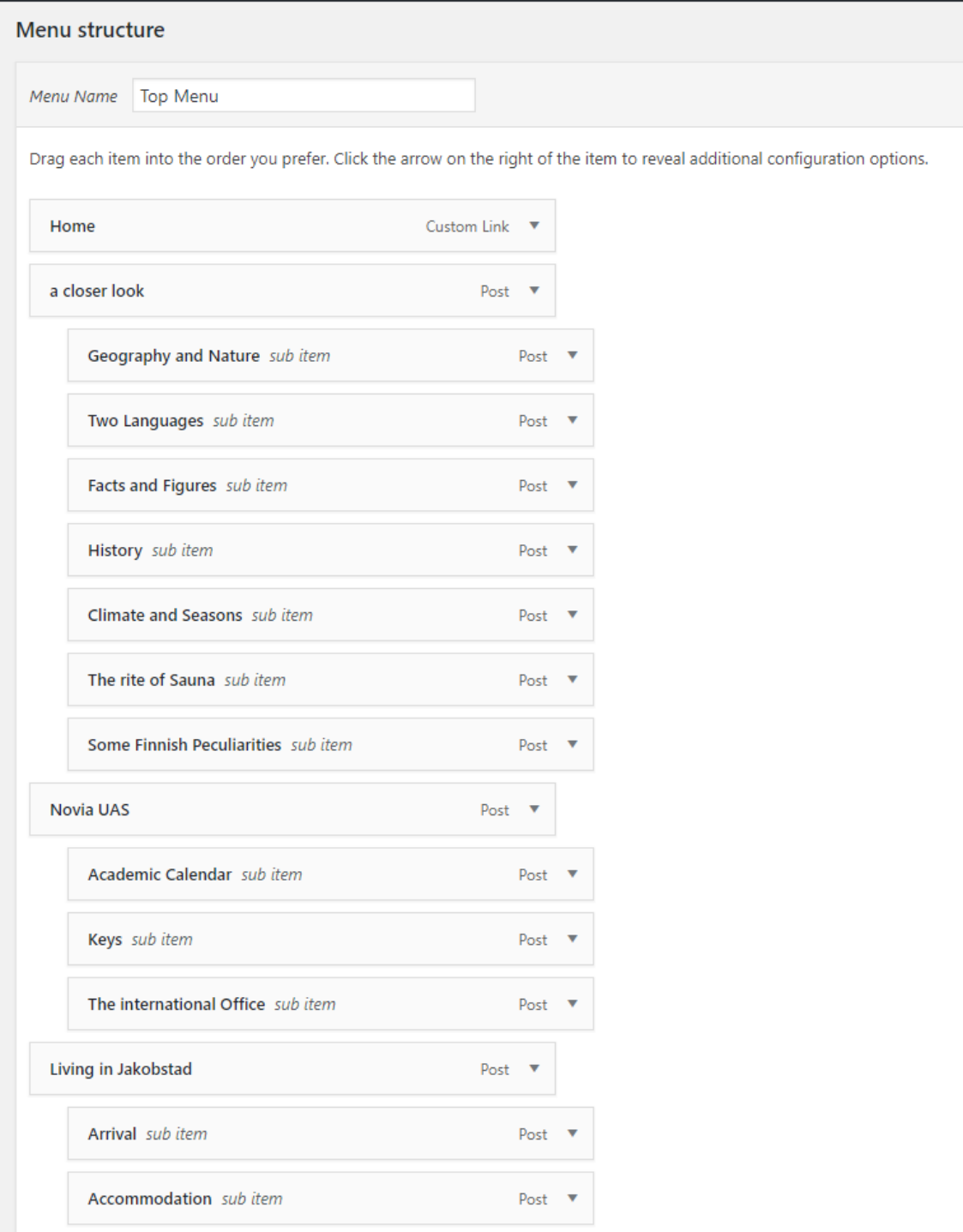
För att lägga till innehållet, textinformationen som applikationen är uppbyggd på användes WordPress-inlägg. På engelska kallas dessa "posts". All textinformation har lagts till i form av inlägg via "Posts"-fliken i administratörspanelen. När tillhörande text lagts in trycker man på knappen "publish" och inlägget publiceras till webbplatsen. All textinformation är uppbyggd på en rubrik som innehåller text, och har underrubriker som innehåller text. Det är också här som upprätthållaren går in om man vill uppdatera informationen.



Figur 15. Ett exempel på hur ett inlägg ser ut.



Följande steg var att konfigurera WordPress-temat. I temat Curver ingår en inläggs-sorterare, där man kan välja i vilken ordning inläggen visas i menyfliken på webbplatsen. Inläggen är sorterade enligt en modell som gavs av uppdragsgivaren. (se figur 16 och 17).



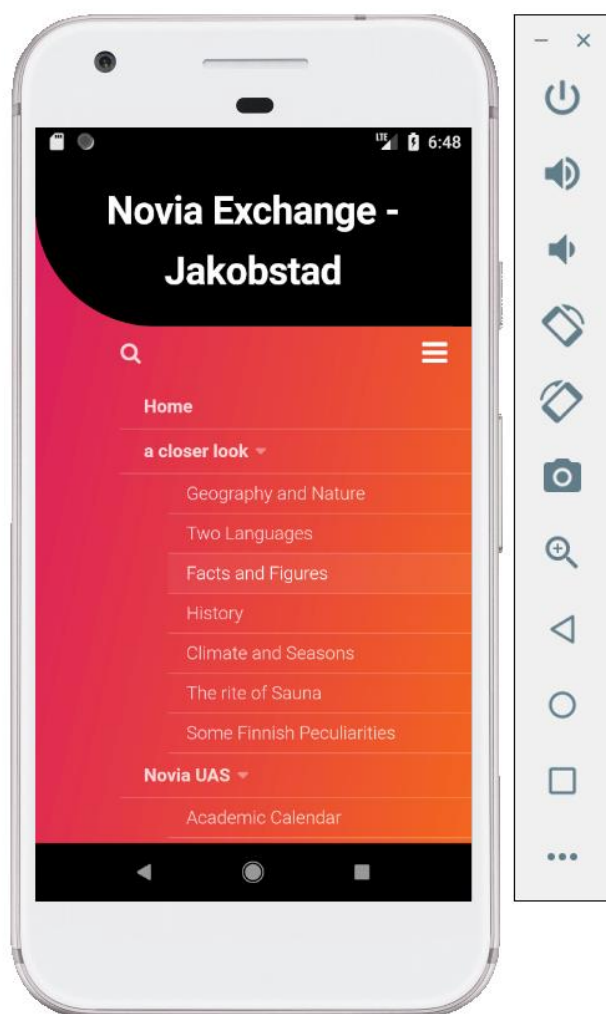
**Menu structure**

Menu Name

Drag each item into the order you prefer. Click the arrow on the right of the item to reveal additional configuration options.

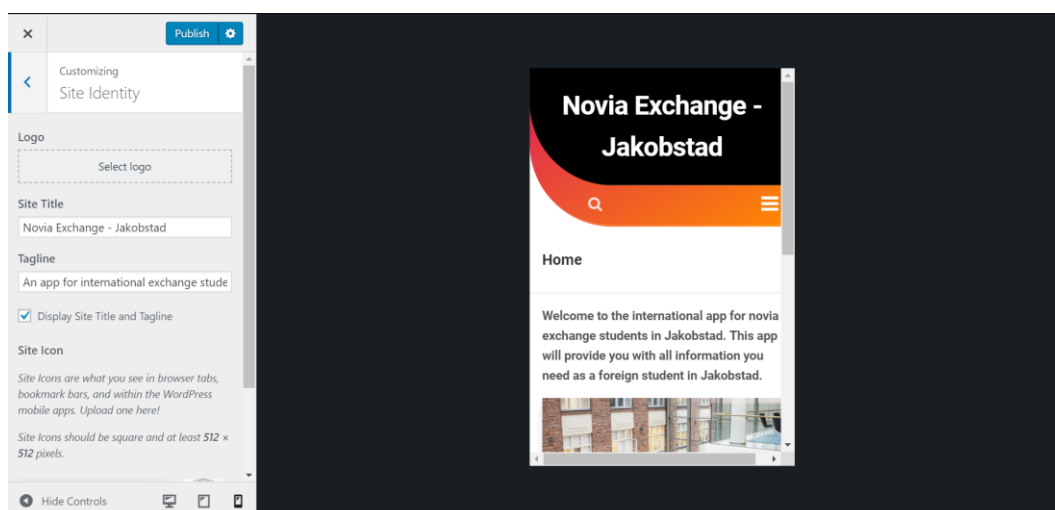
- Home Custom Link ▼
- a closer look Post ▼
  - Geography and Nature *sub item* Post ▼
  - Two Languages *sub item* Post ▼
  - Facts and Figures *sub item* Post ▼
  - History *sub item* Post ▼
  - Climate and Seasons *sub item* Post ▼
  - The rite of Sauna *sub item* Post ▼
  - Some Finnish Peculiarities *sub item* Post ▼
- Novia UAS Post ▼
  - Academic Calendar *sub item* Post ▼
  - Keys *sub item* Post ▼
  - The international Office *sub item* Post ▼
- Living in Jakobstad Post ▼
  - Arrival *sub item* Post ▼
  - Accommodation *sub item* Post ▼

**Figur 16. inläggssortering i administratörspanelen.**



**Figur 17. Inläggsmenyn i applikationen.**

Nästa steg blev att göra ändringar i WordPress-anpassningsmenyn, där man anger saker som: sidtitel, färger och andra inställningar. En del temainställningar gjordes här, till exempel sidofältet. Här togs det bort saker som inte behövdes som kommentarer till varje inlägg och sidofält som var helt onödig i detta examensarbete.

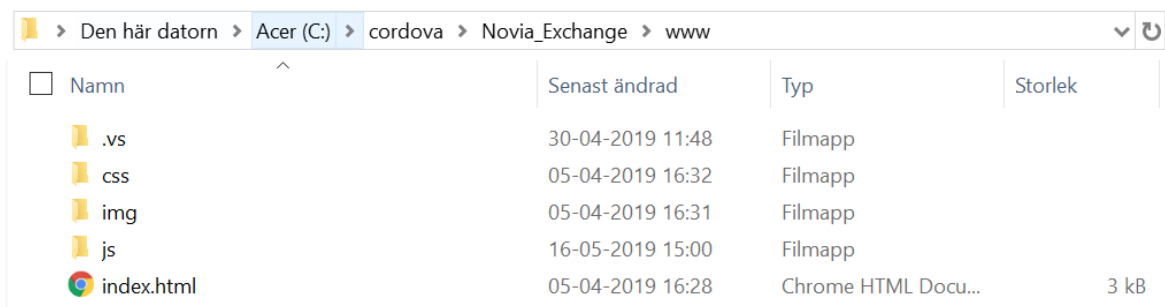


**Figur 18. Anpassningsmenyn.**

## 4.5 Implementering av Cordova-delen

Utförandet i Cordova börjades med att skapa ett Cordova-projekt genom att navigera till Cordova grund-mappen och sedan köra Cordova-kommandon för att lägga till plattform samt att lägga till plugin. När ett Cordova-projekt skapas, skapas alla standard-Cordovafiler som bygger upp hela projektet.

All programmering har skett i html, CSS och JavaScript som är standard för Cordova-projekt. För att komma åt dessa filer öppnar man mappen "www" i Cordovas grund-mapp.



Namn	Senast ändrad	Typ	Storlek
.vs	30-04-2019 11:48	Filmapp	
css	05-04-2019 16:32	Filmapp	
img	05-04-2019 16:31	Filmapp	
js	16-05-2019 15:00	Filmapp	
index.html	05-04-2019 16:28	Chrome HTML Docu...	3 kB

**Figur 19. mappstruktur i Cordova.**

Plattform som har använts i detta examensarbete är Android, med tanke på att endast 10 utbytesstuderanden per år kommer att använda applikationen, var det inte ett krav att utveckla applikationen för IOS-operativsystem, men det är något som lätt kan implementeras eftersom Cordova stöder plattformsberoende arbetsflöde.

Plugin som använts är tillägget InAppBrowser. InAppBrowser kan öppna resurser i form av en webbläsare som öppnas inom Cordova-applikationen. Om man öppnar en länk i en normal mobilapplikation, öppnas länken med en tredjeparts webbläsare beroende på vad som är installerat på användarens telefon. Applikationen är uppbyggd med denna teknik. [17]

```
var ref = cordova.InAppBrowser.open('http://apache.org', '_blank', 'location=yes');
```

**Figur 20. exempel på InAppBrowser-kod.**

En Cordova-applikation byggs upp med hjälp av "Events", händelser på svenska. Applikations-koden kan ha lyssnare (listeners) som lyssnar på en händelse. En händelse kan endast köras när API-biblioteken har laddats in i Cordova-applikationen, dvs. när händelsen OnDeviceReady har körts (se figur 21 och 22).

```

<!DOCTYPE html>
<html>
  <head>
    <title>Device Ready Example</title>

    <script type="text/javascript" charset="utf-8" src="cordova.js"></script>
    <script type="text/javascript" charset="utf-8" src="example.js"></script>
  </head>
  <body onload="onLoad()">
  </body>
</html>

```

**Figur 21. Cordova-events i html-filen.**

```

// example.js file
// Wait for device API libraries to load
//
function onLoad() {
  document.addEventListener("deviceready", onDeviceReady, false);
}

// device APIs are available
//
function onDeviceReady() {
  document.addEventListener("pause", onPause, false);
  document.addEventListener("resume", onResume, false);
  document.addEventListener("menubutton", onMenuKeyDown, false);
  // Add similar listeners for other events
}

function onPause() {
  // Handle the pause event
}

function onResume() {
  // Handle the resume event
}

function onMenuKeyDown() {
  // Handle the menubutton event
}

// Add similar event handlers for other events

```

**Figur 22. Cordova-events i JavaScript-filen.**

Efter att InAppBrowser-tillägget konfigurerats och alla länkar är insatta korrekt, väntar applikationen på att OnDeviceReady har körts och sedan körs InAppBrowser-händelser som sedan utför funktioner (se figur 23).

```

var app = {
  initialize: function() {
    document.addEventListener('deviceready', this.onDeviceReady.bind(this),
false);
  },

  onDeviceReady: function() {
    //code that starts InAppBrowser
  },
};

```

**Figur 23. Cordova InAppBrowser-händelse.**

Med hjälp av InAppBrowser-inlägget i Cordova, görs det möjligt för mobilapplikationsdelen att länkas ihop med WordPress-delen av detta examensarbete.

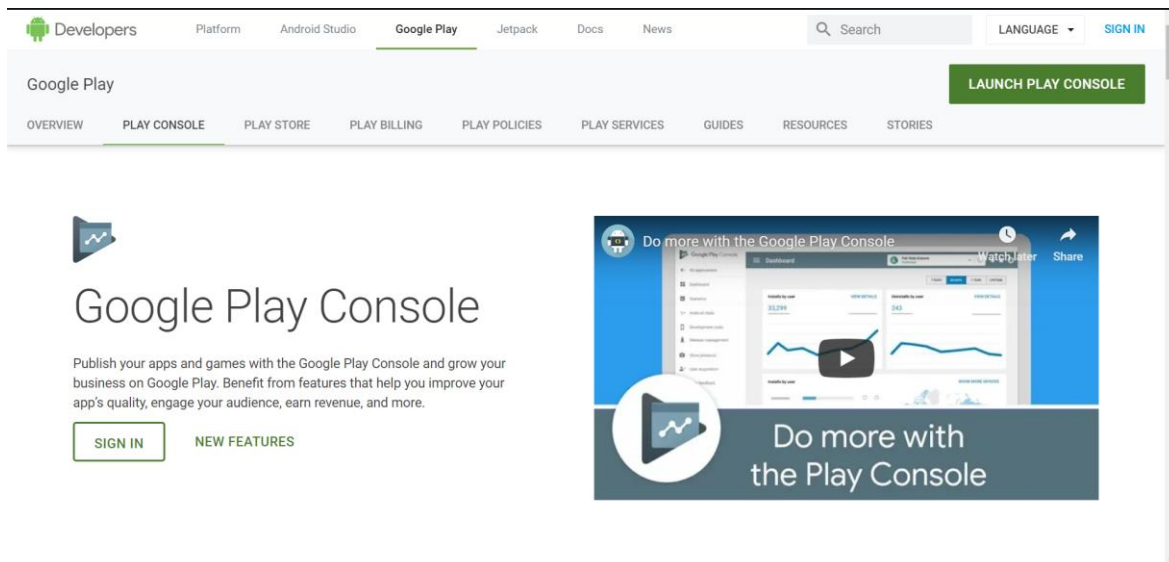
## 4.6 Testgrupp

När mobilapplikationsdelen blev färdig, skapade uppdragsgivaren en testgrupp som bestod av utbytesstuderanden. Man ville ha återkoppling, för att anpassa mobilapplikationen för målgruppen. Testgruppen fick en vecka på sig att ge återkoppling. Återkopplingen var positiv och mobilapplikationen fick endast kommentarer på själva text-innehållet vilket de bads att inte kommentera på, eftersom det kommer att ändra.

## 4.7 Publicering

För att distribuera mobilapplikationen måste man använda sig av Google Play som är nedladdningstjänsten för mobilapplikationer på Android-telefoner. Det går att använda andra tjänster för distribuering men de är inte lika populära. För att ha rättighet till att publicera något på Google Play-butiken, behöver man ha ett Android-utvecklingskonto. För att få ett Android-utvecklingskonto måste man betala en engångsavgift på 25 euro. Efter det är man fri att ladda upp applikationer till nedladdningstjänsten Google Play. Det är lätt att publicera en applikation till Google Play och man behöver inte gå igenom långa ansökningsprocesser. [18]

Google Play Console heter webbplatsen dit man laddar upp själva applikationen när den är redo för distribuering. Här loggar man in med Android-utvecklingskonto och påbörjar uppladdningsprocessen, men innan man laddar upp appen måste man se till att man har lagt en signatur på sin APK-fil (Android application package). APK-filen är filen man laddar upp till Google Play Console, hela applikationen finns i APK-filen. [19]



Figur 24. Webbplatsen Google Play Console. [19]

## 4.8 APK-filsignatur

Efter att man skapat ett projekt och vill publicera det, måste man märka applikationen med ens egen signatur. Man skapar ett nyckel-certifikat som man lägger sin signatur på för att bevisa att APK-filen som man laddar upp tillhör den som har utvecklat applikationen. Signaturen innehåller information som utvecklarens namn, företag, nyckellösenord, ort och land. [20]

För att skapa en signatur på sin APK-fil kör man kommandon i Cordova (se figur 25) som genererar en signatur efter att man angett den information som behövs för att skapa signaturen. [21]

```
C:\cordova\signtest>keytool -genkey -v -keystore appreleasedemo.keystore -keyalg RSA -keysize 2048 -alias appreleasedemoalias -validity 10000
```

Figur 25. Signaturskommando-exempel i Cordova.

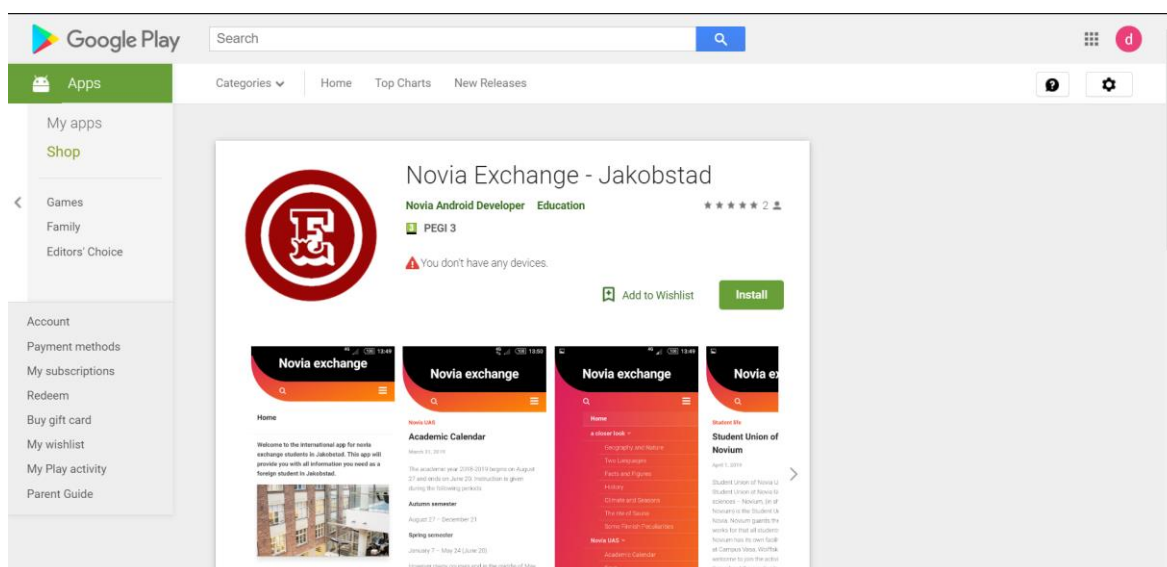
När signaturskommandot har körts utan problem, skapas en APK-fil med signatur (se figur 26). Denna fil uppladdas för publicering.

« cordova » signtest » platforms » android » app » build » outputs » apk » release				Sök i release
↑	□ Namn	Senast ändrad	Typ	Storlek
	app-release.apk	15-05-2019 12:56	BlueStacks Android P...	1 729 kB
	output.json	15-05-2019 12:56	JSON-fil	1 kB

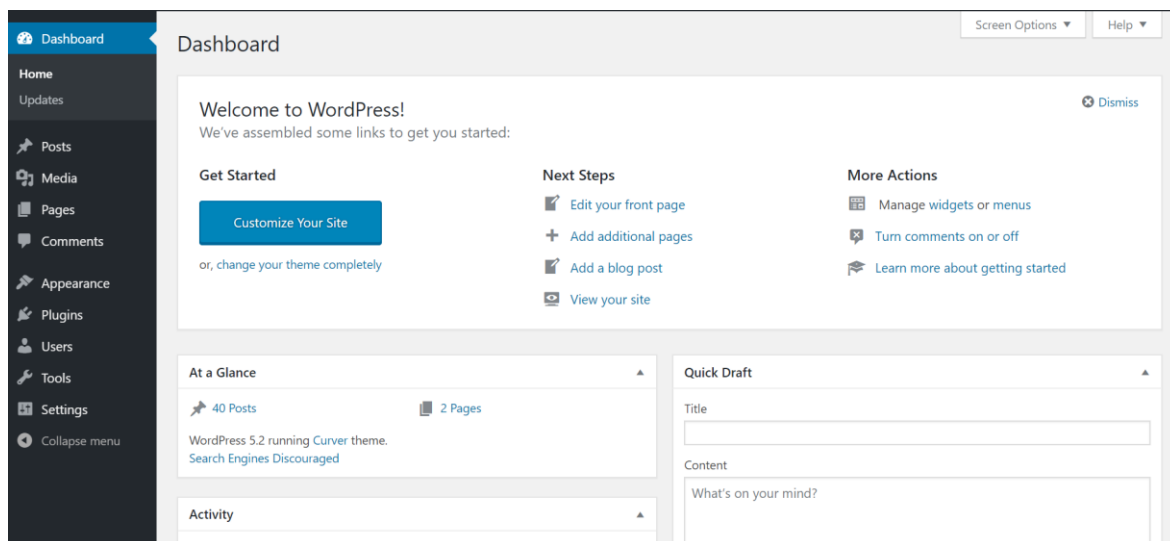
Figur 26. APK med signatur som har genererats.

## 5 Resultat

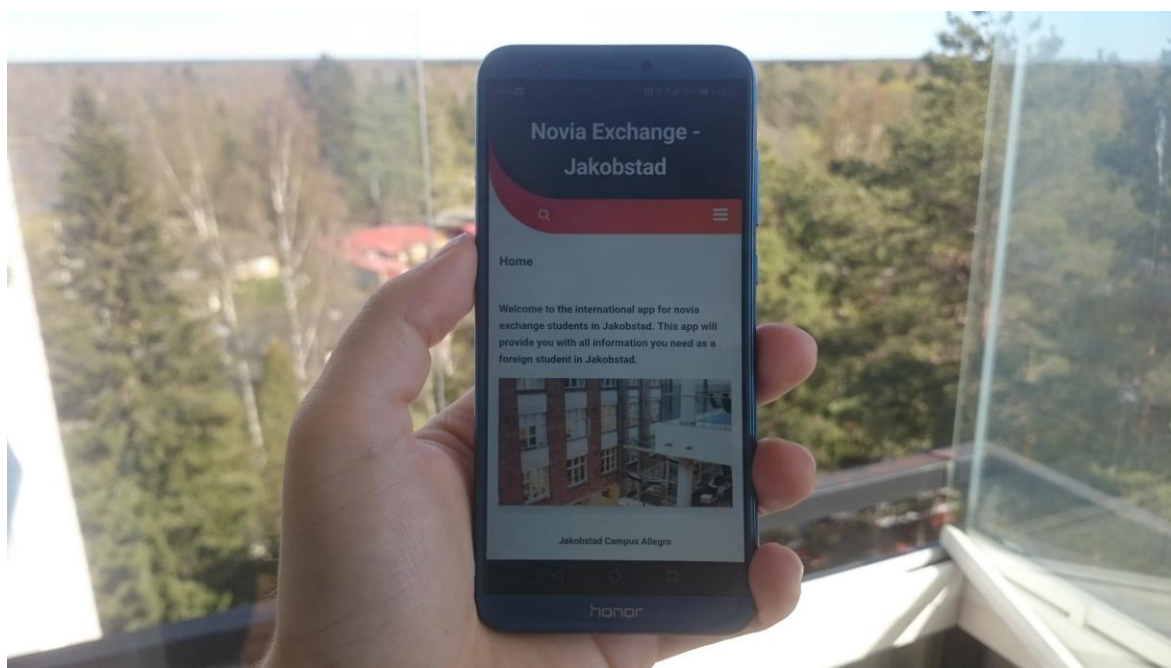
Resultatet blev en fullt funktionerande mobilapplikation som levde upp till uppdragsgivaren krav. Applikationen finns på nedladdningstjänsten Google Play där utbytesstuderanden lätt kan ladda ner mobilapplikationen (se figur 27). Applikationen kan lätt uppdateras via WordPress-administratörspanelen (se figur 28) och det finns möjlighet att replikera applikationen för bruk vid andra orter.



Figur 27. mobilapplikationen på Google Play tjänsten.



**Figur 28. Upprätthållarens administratörspanel.**



**Figur 29. Slutresultatet.**



## 6 Diskussion

Detta examensarbete har varit intressant och lärorikt. Jag har lärt mig mycket eftersom jag aldrig har skapat en mobilapplikation förr. Jag är nöjd med slutresultatet men det finns nog saker som skulle kunna förbättras, t.ex. att publicera mobilapplikationen till Windows- och IOS-operativsystem men det krävdes inte i detta skede.

Det uppstod ett antal utmaningar under utvecklingstiden, om hur WordPress-delen skulle hostas, dvs. valet av webbhotell. Eftersom Novia är uppdragsgivare måste servern installeras på Novias eget nätverk och jag hade ingen aning om vad jag hade för begränsningar innan jag hade ett möte med ICT-avdelningen. Detta möte skedde väldigt sent i utvecklingen eftersom jag inte tänkte så stort. Det blev också funderingar om Novia redan hade ett Android utvecklingskonto, men det visade sig att Novia inte hade det och ett sådant måste anskaffas.

Man kan argumentera att mobilapplikationsdelen är svag med tanke på hur beroende den är av WordPress-webbplatsen, men med tanke på att jag var helt fri inom valet av teknik tänkte jag att det blir nog den bästa lösningen för alla. Mobilapplikationen blev mycket snyggare designvis än vad trodde den skulle bli och jag är nöjd med arbetet.

Applikationen tas i bruk i augusti 2019 och det slutgiltiga namnet blev Novia Exchange – Jakobstad.

## 7 Källförteckning

- [1] C. Pundars-Mitts, "intra.novia.fi," Yrkeshögskolan Novia, 20 8 2018. [Online]. Available: <https://intra.novia.fi/tjanster/internationell-verksamhet/for-studerande/>. [Använd 6 5 2019].
- [2] I. Westerlund, "intra.novia.fi," 22 1 2009. [Online]. Available: <https://intra.novia.fi/assets/Sidor/Tjanster/Internationell-verksamhet/For-studerande-2/Guide-for-utbytesstudier-och-utlandspraktik/Studera-utomlands/givp2.pdf>. [Använd 6 5 2019].
- [3] WordPress, "wordpress.org," WordPress, [Online]. Available: <https://wordpress.org/>. [Använd 20 5 2019].
- [4] B. Barron, "whoishostingthis.com," WhoIsHostingThis?, 12 5 2019. [Online]. Available: <https://www.whoishostingthis.com/compare/wordpress/stats/>. [Använd 14 5 2019].
- [5] Kinsta, "kinsta.com," Kinsta, 15 4 2019. [Online]. Available: <https://kinsta.com/learn/wordpress-history/>. [Använd 8 5 2019].
- [6] E. staff, "wpbeginner.com," WordPress, 1 1 2019. [Online]. Available: <https://www.wpbeginner.com/news/the-history-of-wordpress/>. [Använd 8 5 2019].
- [7] E. Staff, "wpbeginner.com," WordPress, 8 3 2019. [Online]. Available: <https://www.wpbeginner.com/beginners-guide/self-hosted-wordpress-org-vs-free-wordpress-com-infograph/>. [Använd 7 5 2019].
- [8] T. Ewer, "themeisle.com," Themeisle, 3 9 2018. [Online]. Available: <https://themeisle.com/blog/wordpress-customizer-how-to/>. [Använd 8 5 2019].
- [9] E. staff, "wpbeginner.com," WordPress, 9 4 2019. [Online]. Available: <https://www.wpbeginner.com/beginners-guide/what-is-the-difference-between-posts-vs-pages-in-wordpress/>. [Använd 9 5 2019].
- [10] E. staff, "wpbeginner.com," WordPress, 2 11 2013. [Online]. Available: <https://www.wpbeginner.com/glossary/theme/>. [Använd 9 5 2019].
- [11] E. staff, "wpbeginner.com," WordPress, 5 4 2013. [Online]. Available: <https://www.wpbeginner.com/glossary/plugin/>. [Använd 10 5 2019].
- [12] N. Khandelwal, D. Blotsky, M. Myers, S. Rajamanickam, R. Knoll och O. Mefire, "Github.com," Apache Cordova, 17 10 2018. [Online]. Available: <https://github.com/apache/cordova-docs/blob/master/www/docs/en/dev/guide/overview/index.md>. [Använd 10 5 2019].

- [13] Node.js, "nodejs.org," Node.js, [Online]. Available: <https://nodejs.org/en/about/>. [Använd 12 5 2019].
- [14] EVOthings, "evthings.com," EVOthings, 2016. [Online]. Available: <https://evthings.com/doc/build/cordova-install-windows.html>. [Använd 20 5 2019].
- [15] A. s. u. guide, "developer.android.com," Android Studio, [Online]. Available: <https://developer.android.com/studio/run/emulator>. [Använd 13 5 2019].
- [16] A. Agnarson, "wordpress.org," WordPress, 19 3 2019. [Online]. Available: <https://wordpress.org/themes/curver/>. [Använd 15 5 2019].
- [17] Apache Cordova, "github.com," Cordova, 6 3 2019. [Online]. Available: <https://github.com/apache/cordova-plugin-inappbrowser/blob/master/README.md>. [Använd 17 5 2019].
- [18] Google, "play.google.com," Google Play, (u.å). [Online]. Available: <https://play.google.com/store>. [Använd 17 5 2019].
- [19] Google, "developer.android.com," Google, (u.å). [Online]. Available: <https://developer.android.com/distribute/console>. [Använd 17 5 2019].
- [20] A. d. guides, "developer.android.com," Google, [Online]. Available: <https://developer.android.com/training/articles/keystore>. [Använd 17 5 2019].
- [21] Cordova, "github.com," Cordova, 27 3 2019. [Online]. Available: <https://github.com/apache/cordova-docs/blob/master/www/docs/en/dev/guide/platforms/android/index.md>. [Använd 17 5 2019].