

Vis'it

Introduktion

Vi har skapat den webbaserade appen *Vis'it* som bygger på att användare kan ta bilder på och lägga upp sevärdheter via sin mobiltelefon. Dessa sevärdheter är positionsbaserade vilket betyder att de läggs upp som markörer på en karta. Användare kan sedan utforska kartan för att hitta intressanta sevärdheter i närheten.

Appbeskrivning, funktioner

Vis'it är en social placeringsbaserad webbapplikation som är till för att man ska visa bilder för sina vänner. Det är en sorts placeringsbaserad variant av den populära tjänsten Instagram. Appen är främst uppbyggd av webbspråken HTML/CSS och JavaScript. Appen använder sig av en rad olika API:er. Dessa är:

Facebook API

Används främst vid inloggningen. Facebooks API gör att vi kan hämta användarens namn samt användarens vänner på Facebook, något som vi använder för att selektera ut vilka sevärdheter som ska dyka upp i nyhetsflödet och på själva kartan.

Google Maps API

Kartan samt de markörer som ligger på kartan skapas genom *Google Maps*.

jQuery

jQuery används lite överallt i appen på olika sätt. Det främsta användningsområdet är vid selektering och manipulering av HTML-element.

jQuery Mobile

jQuery Mobile används för att skapa appens mobilvänliga användargränssnitt.

cloudbase.io

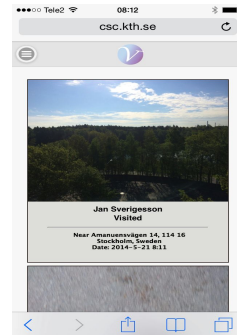
All data som användaren postat lagras i en databas som nås genom API:et *cloudbase.io*. I appen gör vi anrop till denna databas via JavaScript.

Användare identifieras genom inloggning via Facebook. Efter att man loggat in kommer man till appens huvudsida som är nyhetsflödet. Här listas alla sevärdheter som ens vänner lagt upp.

Om man klickar på knappen uppe i appens vänstra hörn (eller tiltar mobilen till höger, ifall mobilen stödjer den funktionen) får man upp en meny där man kan välja bland appens alla funktioner och sidor. Dessa är:

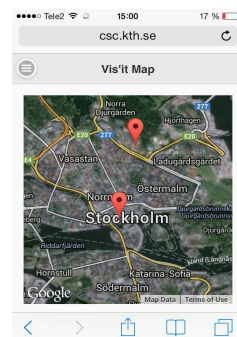
- Feed

På denna sida laddar appen upp alla bildobjekt som finns lagrade i Cloudbase, filtrerade på så sätt att endast bilder som dina vänner från Facebook har lagt upp. Varje bild innehåller en pop-up funktion som aktiveras när man klickar på bilden. Man får där en utökad vy med en bildbeskrivning samt datum och tid då bilden togs. Där kan man även läsa kommentarer och kommentera bilden. Vi använder JQuery Mobiles "pop-up page" för att få detta att funka. Här använder vi oss av Google Maps API för att erhålla koordinater, som sen kan omvandlas till en formaterad adress, dvs en plats. *Se sista sidan för fler och tydligare bilder.*



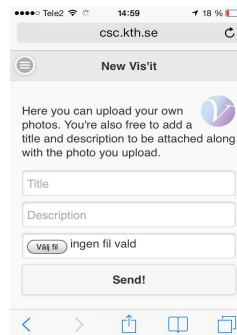
- Map

Den här sidan laddar och visar markern på en google map på de platser där bilderna i din feed är tagna. Du kan även använda shake-funktionen på mobilen för att slumpmässigt visa ett antal markern. Om man klickar på markern får man även där upp en pop-up i form av googles egna info-window, med bilden och all medföljande info som användaren bifogat. Vi använder också geolocation för att hämta användarens nuvarande position.



- New visit

På New visit-sidan används en rad input-fält och en mobilkamera-plugin för att kunna ta en bild, samt lägga till en egen titel och beskrivning för den tagna bilden. Dessutom används geolocation även här för att hämta användarens position. All denna data laddas sedan upp till Cloudbase som ett bildobjekt, och läggs upp i andras nyhetsflöden, samt i ens egna feed. Sidan känner även av så att all data är ifyllt och hindrar användaren från att inte fylla i något.

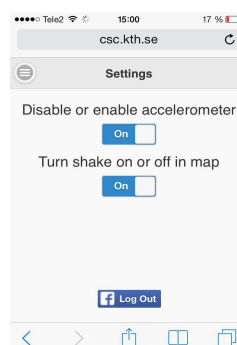


- Your visits

Som feeden, men anpassad så att du endast ser dina egna uppladdade sevärdheter. Här använder vi användarens egna Facebook-ID för att utföra filtreringen.

- Settings

En sida där användaren har möjlighet att ändra vissa inställningar för appen, såsom shake-funktionen och en möjlighet att logga ut från sitt Facebook-konto.



Tillvägagångssätt

Vi började med att skapa en Facebook-grupp där vi kunde kommunicera och vi hade även ett inledande möte via skype för att brainstorma och komma fram till en gemensam idé. Det som intresserade oss som mest var social geolocation, även om vi hade många andra förslag inom övriga områden så är det social geolocation som vi har utgått ifrån när vi började forma vår idé. Sedan gjorde vi en tidsplan och skrev ned applikationens eventuella funktioner. Tidsplanen har vi försökt att följa men efter en del problem och förhinder så har vi fått anpassa oss likaså för applikationens tänkta funktioner och innehåll. Till en början delade vi upp arbetet två och två för att få en grund att bygga applikationen utifrån. Efter detta följde gruppmöten varje morgon för att göra en lista över det som behövde prioriteras högst just den dagen. Eftersom att vi alla satt i skolan de dagar då vi jobbade var det lätt att koordinera och samarbeta på ett effektivt sätt. Inledningsvis tänkte vi samarbeta via github men då vi alla kunde sitta i skolan så blev det smidigare att istället börja koda direkt. Vi blev därmed tvungna att jobba i olika filer och då vissa funktioner var beroende av andra så fick vi "merge:a" filerna då och då för att se att applikationens olika delar fungerade tillsammans. Därmed har vi försökt skriva koden så att det har varit simpelt att merge:a koden. Vilket innebär bra variabelnamn, kommentarer, väl uppdelade funktioner mm.

Hur vi har jobbat:

Sön 20/4	Ons 23/4	Mån 5/5	Tis 6/5	Ons 7/5	Tors 8/5	Fre 9/5	Mån 19/5	Tis 20/5
20:00-2 3:00	9:00-14: 00	9:00-17: 00	9:00-17: 00	9:00-17: 00	9:00-18: 00	9:00-17: 00	9:00-17: 00	10:00-1 8:30

Brister

Den mest uppenbara bristen i appen är att den mer eller mindre inte fungerar i iOS webbapp-läge. Detta beror på att Facebooks inloggning bygger på inloggning via ett pop-up-fönster, något som webbappen i stand-alone-läge inte klarar av att hantera. Vi gick t.o.m. så långt att vi kontaktade Facebooks support för att få hjälp, men fick då svaret att Facebooks JavaScript-API ej fungerade i nuläget tillsammans med webbappar på hemskärmen. Detta medför att appen enbart fungerar i webbläsaren.

Vidare har Cloudbase-API:et varit ganska omständigt att jobba med då vi vid ett flertal tillfällen fått lösa olika asynkroniska problem som uppstått, något vi tidigare inte handskats så mycket med. Detta har lett till att många av programmeringslösningarna kanske inte är de mest optimala. Cloudbase har även i allmänhet varit rätt krångligt att jobba med. Delvis för att det

knappt fanns någon info och hjälp att hitta online när man stötte på problem med systemet, samt också att databasen i sig inte kändes helt pålitlig när det gällde att ladda in bildobjekt för användning i programmet.

Reflektion över grupparbetet

Samtliga i gruppen är eniga om att grupparbetet fungerade bäst när vi satt samlade tillsammans på en plats, snarare än att vi satt och jobbade hemifrån, som från början var grundtanken till hur själva arbetet skulle genomföras. Vår kommunikation inom gruppen blev definitivt bättre, vilket ledde till att vi lättare kunde lösa svåra problem och i allmänhet jobba mer effektivt.

Samtidigt har vi även fått erfarenhet av att utföra ett programmeringsprojekt i en större grupp, vilket känns bra, då vi från början var skeptiska till att det skulle fungera bra.

Vi införde strikta regler på att man var tvungen att inhandla fika om man var mer än 5 minuter sen, två gånger. Detta medförde att alla gruppledmedlemmar (för det mesta) kom i tid vid samtliga tillfällen.

För övrigt lyckades vi bra med att dela upp det grundläggande kodningsarbetet, där vi till en början delade upp olika arbetsuppgifter i grupper om två

Länk till källkod: <http://www.csc.kth.se/~johanhuu/DM2518/projekt/>

Projektets blogg: <http://mopubvisit.blogspot.se/>

Projektets deltagare:

Erik Dahlström

Simon Dovhage

Niclas Eriksson

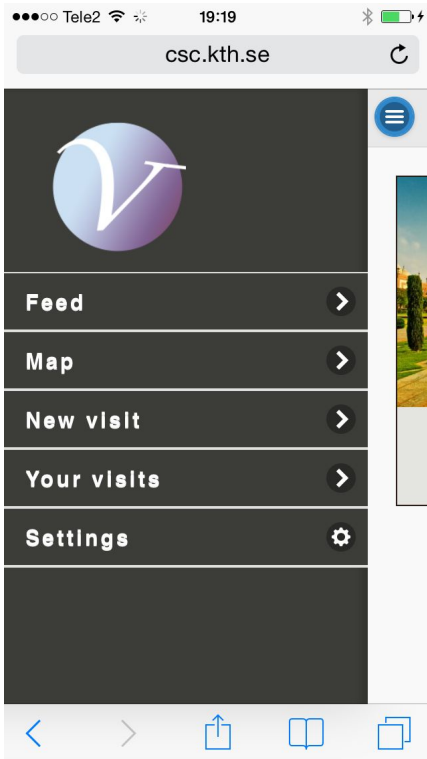
Simon Fransson

Fredrik Hagelin

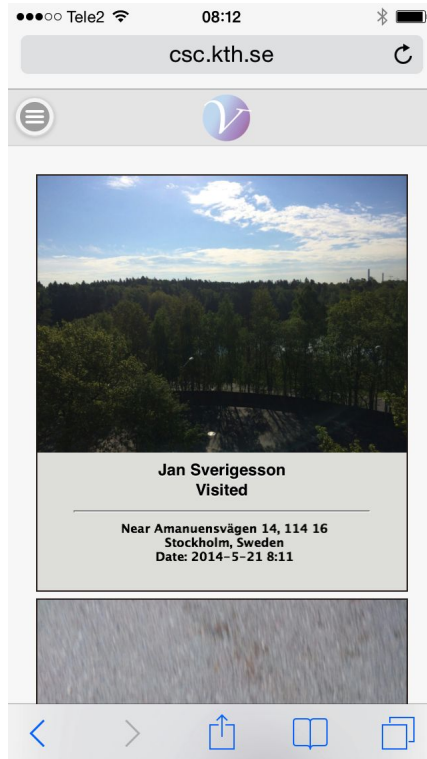
Johan Huusmann

Bilaga: Bilder

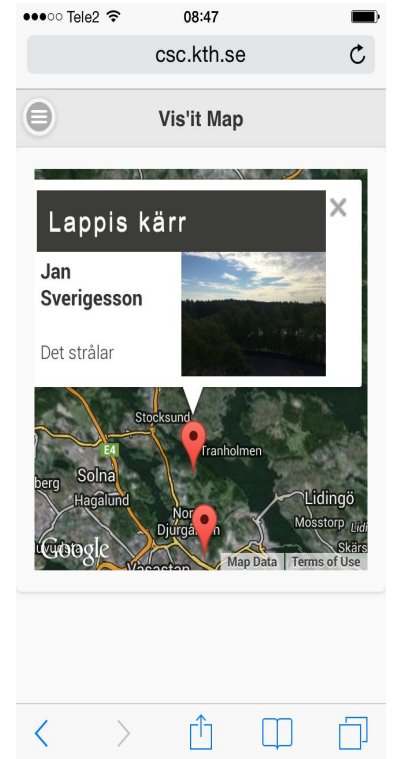
Meny



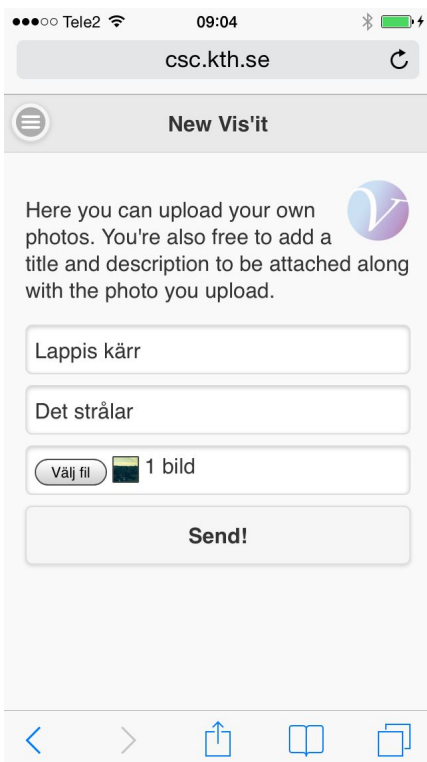
Feed



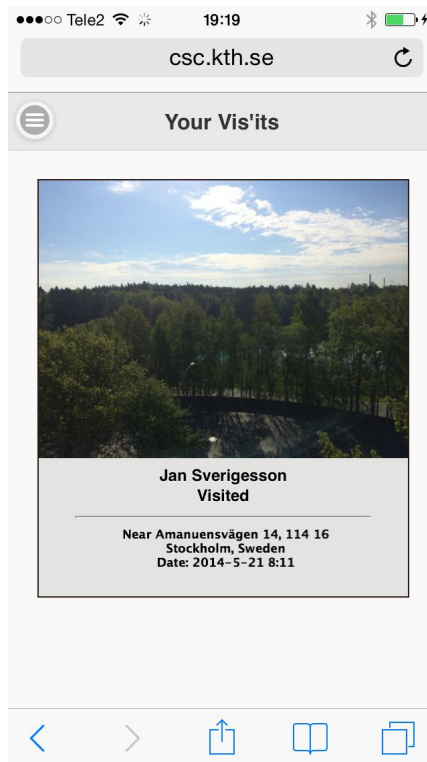
Map



New Vis'it



Your Vis'its



Settings

