

# Moments

- EN MIKROBLOGG

DOKUMENTSTRUKTURER, TNM065

Linköping University

---

Linnéa MELLBLOM

linme882

16 december 2015

# Innehåll

<b>1</b>	<b>Bakgrund</b>	<b>2</b>
<b>2</b>	<b>Implementering</b>	<b>2</b>
2.1	Utvidgningar . . . . .	3
<b>3</b>	<b>Resultat och köranvisning</b>	<b>3</b>
3.1	Startsida . . . . .	3
3.2	Redigera ett inlägg . . . . .	5
3.3	Sökresultat . . . . .	5
3.4	Logga in . . . . .	6
3.5	Profil . . . . .	6
3.6	Admin . . . . .	7
3.7	Sida för ett enskilt inlägg . . . . .	7
3.8	Mobilversion . . . . .	8
<b>4</b>	<b>Reflektion</b>	<b>9</b>

# 1 Bakgrund

Bakgrunden för detta projekt är att skapa en XML-baserad webbtjänst för flerkanalspubliserings. Tjänsten ska kunna utnyttjas oberoende av skärm, det vill säga webbläsare eller genom mobiltelefon med en mindre skärm. Huvudsakliga områden som fokuserats på är XML med XSLT, sökning i XML-strukturer, DTD och dynamisk generering av XML med hjälp av PHP och även AJAX och MySQL databas.

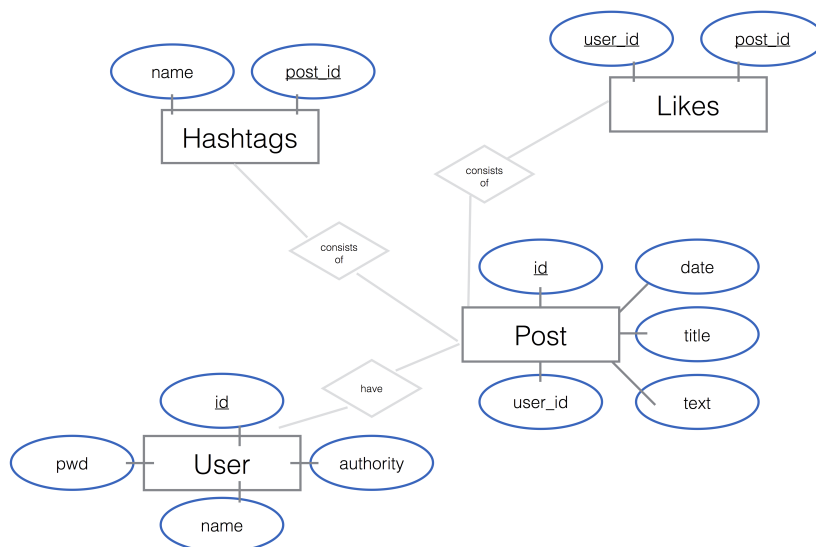
## 2 Implementering

Med tekniker som bakgrund beskrev skapades projektet *Moments*. *Moments* är en form av mikroblogg liknande exempelvis *Twitter* där huvudsakliga tjänsten är att publicera inlägg. Det finns ingen gräns på hur långa inläggen får vara, men formuläret är designat för kortare texter.

Ett inlägg består av en rubrik, text och sen eventuellt hashtags som beskriver vad inlägget handlar om. I tjänsten kan du som inloggad även interagera med andra inlägg genom att *gilla* andras inlägg. Om en användare har skapat ett inlogg kan även denna ändra sin profilbild mellan fyra förvalda bilder.

Tekniker som använts i detta projekt är som sagt PHP, XML, XSLT, AJAX och MySQL som databas.

För att skapa en sida så används PHP som beroende på vilken sida man är på plus om användaren har interagerat på något sätt (lagt till ett inlägg, ny användare osv), skickar en förfrågan till databasen. Genom svaret från databasen så genereras dynamiskt ett XML-dokument. Detta XML-dokument transformeras i sin tur med hjälp av XSLT till HTML. I XSLT så väljs vad från XML:en som ska visas och hur det ska visas. På så sätt så separerar man informationen från databasen (XML) med utseendet. På vissa sidor finns även AJAX där det använts till att dynamiskt ändra innehåll på sidan utan att uppdatera hela sidan. Exempelvis när man ska redigera ett inlägg så hämtas formuläret ifyllt via AJAX som ersätter innehållet där posten visades. Med dessa tekniker så separeras innehåll från utseende och gör påbyggnad och modifiering av källkod till en enklare process, dvs MVC-modellen (Model, View, Controller).



Figur 1: En enkel skiss över databasens struktur.

I databasen är det fyra stycken olika tabeller som data finns lagrat i. Dessa är *post*, *hashtags*, *likes* och *user*. Med olika id så kopplas dessa samman och kan få ut data om alla olika inlägg och användare. En *post* kan bestå av inga eller flera *hashtags* och samma sak gäller för *likes*. För att identifiera vilken *hashtag* som tillhör vilken *post* får dessa postens id. Samma sak gäller här för *likes* där *user-id* visar vilken *user*

som gillat *posten post-id*. En *post* har även en *user* för att veta vem som skrev inlägget. En enkel skiss på databasen kan ses i figur 1.

## 2.1 Utvidgningar

Utöver basuppgiften där implementering av formulär, ändring av inlagd data, lagra i databas, export till XML och olika XSL-stylesheets för två ut-enheter så har en rad utvidgningar gjorts i projektet. Nedan listas det som implementerats utöver basuppgiften.

**Behörighetskontroll** På sidan kan du logga in som användare eller registrera dig. För att exempelvis kunna gilla inlägg och självklart för att skriva inlägg måste användaren vara inloggad. Detta har implementerats genom att använda *PHP SESSION* där användarens id lagras när denne loggar in. Det finns även olika behörigheter på användare och det är administratör och standard. Som admin har du rättigheten att ändra och ta bort alla inlägg.

**Administrera användare** Här kan de användare som är administratör se alla användare som finns i systemet. Administratörerna kan även ändra behörighet på användare och om den vill ändra lösenord eller ändra användarnamn.

**Sökfunktion** På startsidan går det att söka på olika hashtags och därmed enbart visa upp de inlägg som är kopplade till det sökts på. Om exempelvis sökningen enbart var *t* så visas alla inlägg med hashtags som innehåller *t*.

**AJAX** För att uppdatera delar av sidan utan att behöva uppdatera hela sidan har AJAX tillämpats. Detta används när en användare vill redigera ett inlägg. Då hämtas ett formulär som redan är ifyllt med informationen som fanns och ersätter där inlägget visades med formuläret istället.

AJAX används även när en användare vill ändra profilbild. Då skickas en request till databasen som säger vilken bild som den ska uppdatera till, och om allt gick rätt till så kommer scriptet att uppdatera alla användarens bilder på sidan dynamiskt till den nya bilden,

**RSS-feed** En RSS-feed har skapats för gjorda inlägg. Alla inlägg som är gjorda listas här och uppdateras när användare lägger till inlägg.

**Konvertera XML till JSON via XSLT** En konverterare som tar XML till JSON. Detta görs enbart för XML som handlar om *posterna*. Visar alla poster, vilka som gillat posten och hashtags kopplade till posten.

## 3 Resultat och köranvisning

Sidan har fyra olika vyer som nedan kommer visas och även en beskrivning av vad som kan göras på respektive sida. De olika vyerna som finns är startsidan profil, adminsida och en sida för ett specifikt inlägg. Andra sidor som även finns är för att logga in och startsidan med sökresultat. Även beskrivning om hur man redigerar ett inlägg kommer visas.

### 3.1 Startside

Startsidan är det som visas när man kommer in på tjänsten. Det första man ser är alla inlägg som gjorts på sidan. I högerspalten kan man, beroende på om inloggad eller ej, antingen lägga till ett inlägg eller inte.

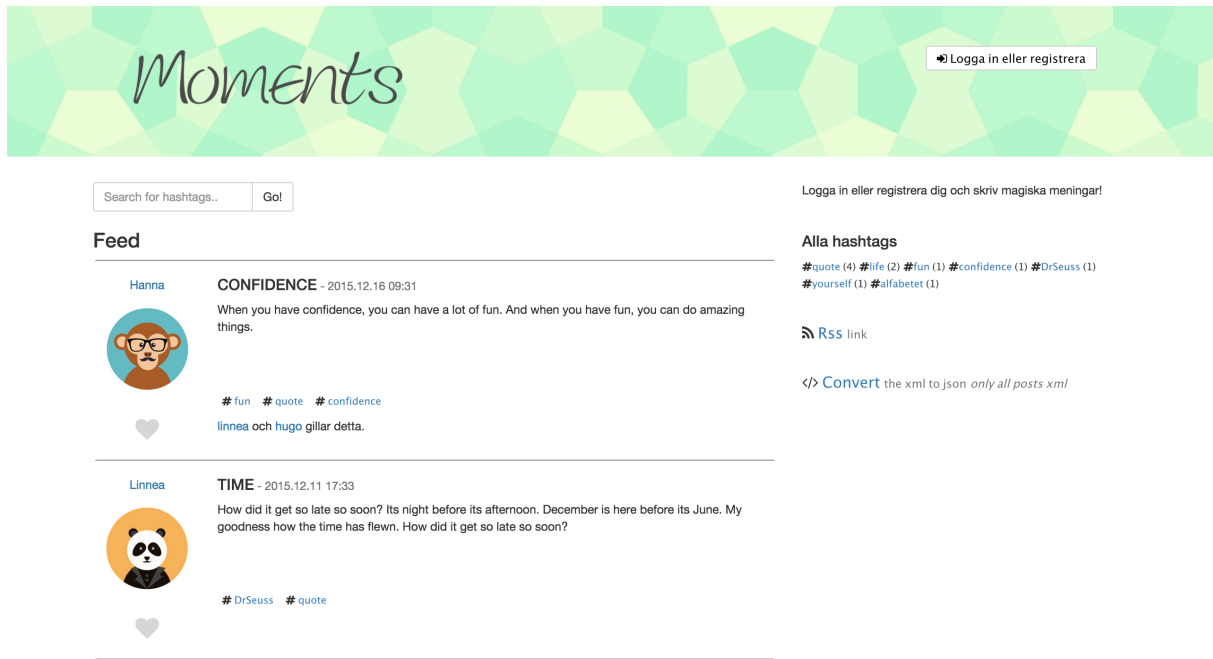
Det går även att söka på hashtags. Se förklaring i en rubrik nedan om *sökresultet*.

Alla hashtags som finns kan även ses och även hur många gånger hashtagsen har förekommit. Under varje inlägg finns det även en länk till hashtagsen. Den länken går till är samma som sökningen görs via.

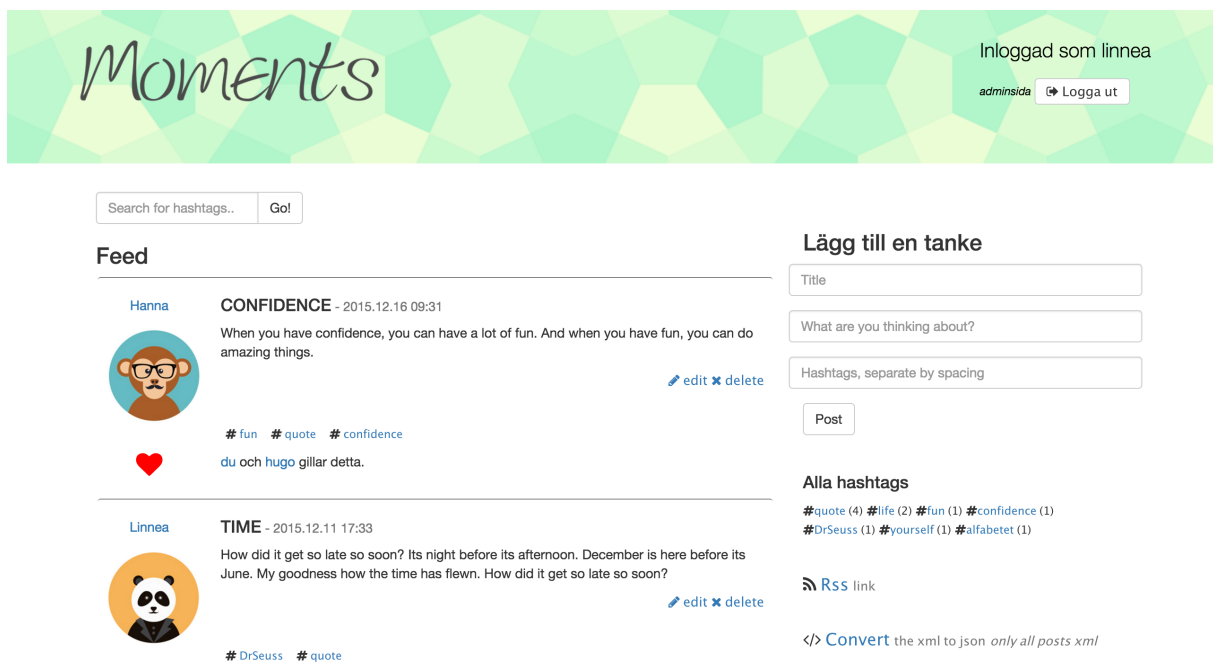
Kan även härifrån ta sig till RSS-flödet. Finns också en länk till konverteringen från XML till JSON. Den XML som används för konverteringen är den XML som genereras till startsidan utan att det har förekommit en sökning.

I ett inlägg finns också ett hjärta. Om du är inloggad så betyder det att du kan gilla ett inlägg. Bredvid hjärtat står det de användare som gillat inlägget om det finns några. Om det är så att du gillat ett inlägg så kommer hjärtat att bli rött och även stå *du* gillat inlägget bredvid hjärtat. Om du redan har gillat ett inlägg och hjärtat lyser rött, kan du avgilla genom att klicka på hjärtat igen och det kommer bli grått.

Längst upp till höger finns en länk till att logga in eller registrera sig om användaren inte är inloggad. Annars är det en länk till din egen profil, länk till adminsida om du är administratör och en logga ut länk.



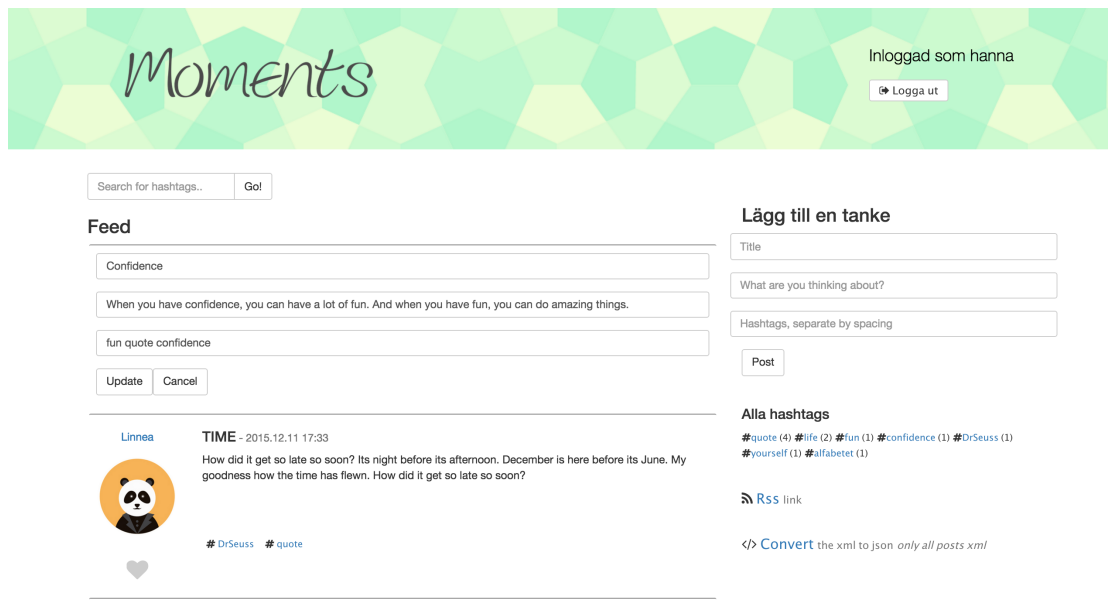
Figur 2: Startsidan med en användare som inte är inloggad. Kan enbart läsa inlägg och navigera mellan användares profiler, söka på hashtags och klicka på olika hashtags.



Figur 3: Startsidan för en användare som är inloggad. Denna person kan förutom allt som en användare som inte är inloggad kan göra, även skriva ett inlägg och redigera sina inlägg.

## 3.2 Redigera ett inlägg

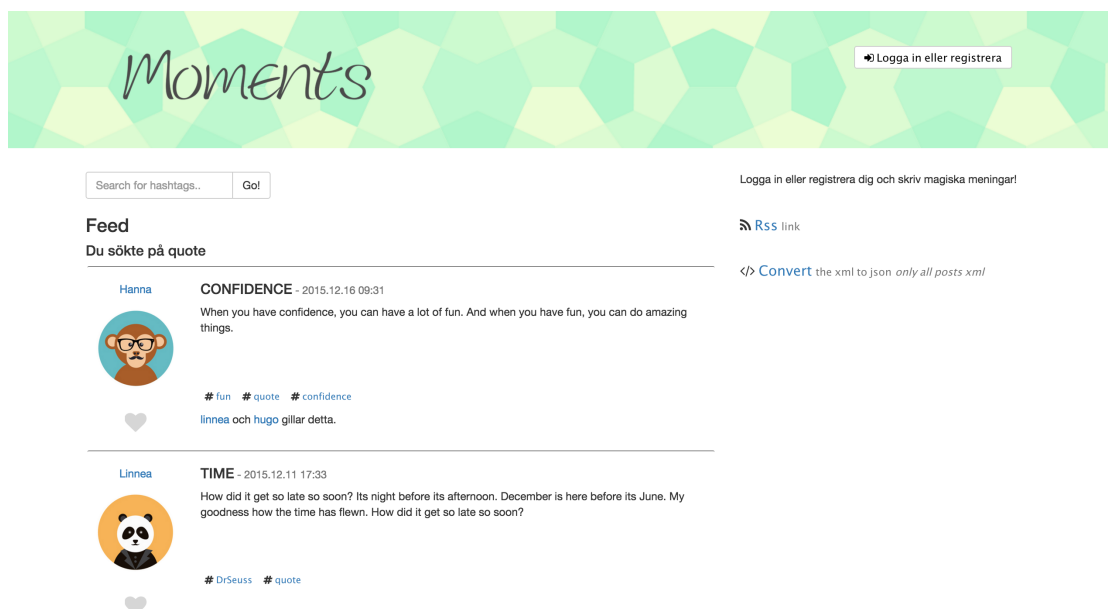
Om du är inloggad användare och ser ett inlägg som är ditt eget kan du välja att redigera detta eller radera. Om du är administratör kan du redigera allas inlägg oavsett. Du kan redigera ett inlägg i olika vyer, vare sig det är på startsidan, profil-sidan eller vyn för ett inlägg. Formuläret för redigera genereras med AJAX som beskrivits om tidigare. I figure 4 ses hur detta formulär ser ut ifyllt.



Figur 4: Visar på hur formuläret för att redigera ett inlägg ser ut. Är samma formulär som för när man lägger till ett nytt inlägg men skillnaden är att nu är formuläret ifyllt från början.

## 3.3 Sökresultat

Genom att skriva in i sökfältet kan du söka på olika hashtags som innehåller det ord du sökte på. Du kan även komma till denna sida genom att klicka på en hashtag-länk.



Figur 5: Sökresultat visas på startsidan men visar enbart de inlägg som har hashtags som innehåller de ordet man sökte på.

### 3.4 Logga in

På denna sida kan du enbart logga in eller registrera dig. Valde att ha enskilt sida så det skulle ta bort fokus från startsidan och smidigt då vid mobilen att ha en enskild sida så detta inte tog plats på startsidan.

Figur 6: Login-sida. Här kan du antingen logga in om du redan är registrerad eller registrera dig.

Figur 7: Login-sida med error. Felmeddelande som i detta fall beror på antingen fel lösenord eller fel användarnamn.

### 3.5 Profil

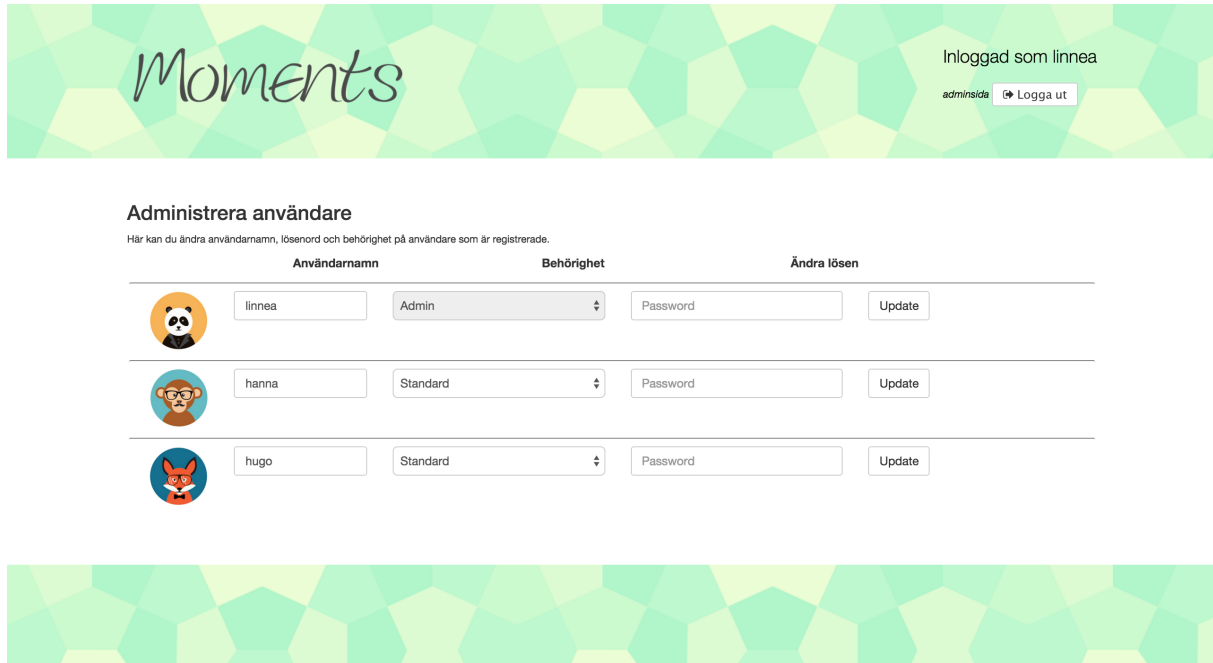
Profilsidan visar en specifik användare och alla inlägg som tillhör. Går även att se vilka hashtags som personen använder. Går även här att interagera med inläggen som i de andra vyerna, nämligen gilla, redigera och ta bort beroende på om användaren är inloggad, administratör osv.




Om användaren är inne på sin egna profil går det att ändra sin profilbild. Det går att ändra mellan fyra olika förvalda bilder. Som det beskrevs om i *utvidgningar* så uppdateras bilderna dynamiskt via AJAX och även samtidigt skickas en förfrågan till databasen att ändra profilbild.

Figur 8: Visar en profil-sida. Just denna sida är den aktuella inloggade användaren då denne kan välja att ändra sin profilbild.

### 3.6 Admin

Adminsidan kommer enbart administratörer åt. Här går det att ändra användarnamn, behörighet och lösenord på de användare som finns i systemet. Om du försöker komma åt denna sida utan att vara behörig (det vill säga inte inloggad eller har en vanlig standard behörighet) så kommer ett felmeddelande visas.



	Användarnamn	Behörighet	Ändra lösen
	<input type="text" value="linnea"/>	<span>Admin</span>	<input type="password" value="Password"/> <input type="button" value="Update"/>
	<input type="text" value="hanna"/>	<span>Standard</span>	<input type="password" value="Password"/> <input type="button" value="Update"/>
	<input type="text" value="hugo"/>	<span>Standard</span>	<input type="password" value="Password"/> <input type="button" value="Update"/>

Figur 9: Denna sida kommer enbart administratörer åt. Här kan personen ändra lösenord, användarnamn och redigera behörighet.

### 3.7 Sida för ett enskilt inlägg

En väldigt sparsmakad sida där enbart ett inlägg visas. Detta är främst till för RSS-feeden för att kunna länka till ett inlägg och snabbt kunna ta sig till nyheten”. Om användaren är inloggad så går det att interagera med inlägget som på de andra sidorna (det vill säga gilla eller redigera/ta bort ifall du själv gjort inlägget eller är admin).

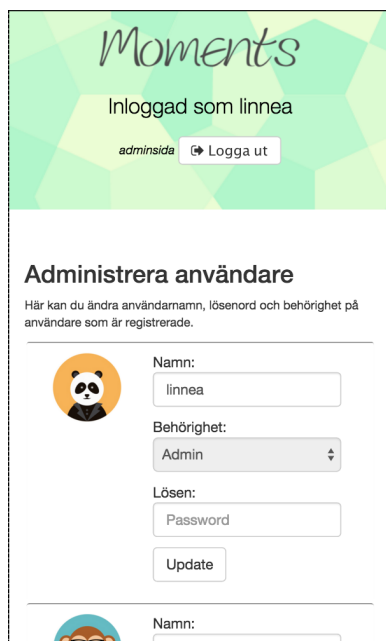


Figur 10: Visar ett enskilt inlägg.

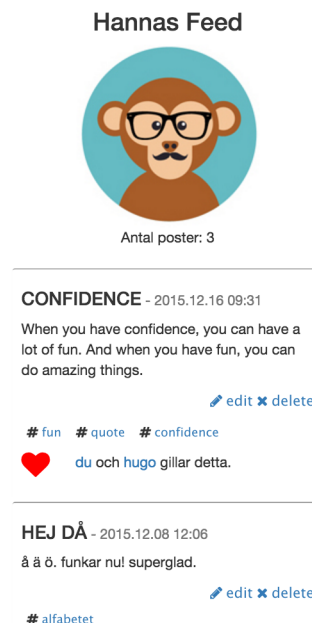


### 3.8 Mobilversion

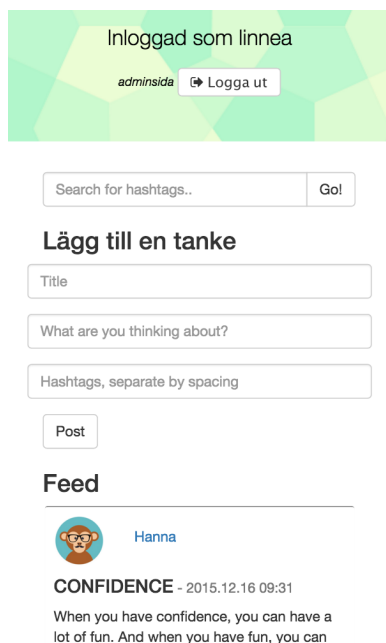
Alla vyer har samma funktionalitet på mobilen som den har i webbläsaren. Det som skiljer åt är hur det visas. Nedan kommer ett par skärmdumpar enbart att visas.



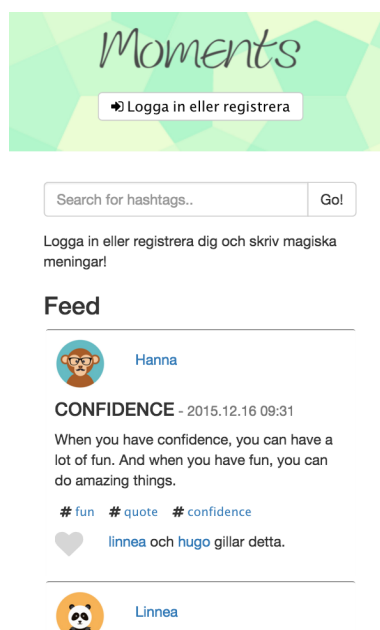
Figur 11: Mobilversion av admin-sidan. Samma funktionalitet som webbläsarens version men utseendemässigt skillnad. .



Figur 12: Mobilversionen av profil-sidan. Här visas användarens bild större först för att sedan inte alls synas i flödet skillnad mot webbversionen.



Figur 13: Mobilversion av startsidan. Samma funktionalitet som webbläsarens version men utseendemässigt skillnad. Här kan man direkt se att man kan göra ett inlägg och sen se flödet.



Figur 14: Mobilversionen av startsidan när ingen användare är inloggad. Eftersom ingen är inloggad visas bara flödet.

## 4 Reflektion

Reflektionerna kommer främst vara baserad från erfarenheter som uppkommit under detta projekt och inte nödvändigtvis stämma överrens med vad som gäller för XML generellt.

En stor fördel var att det är lätt att komma igång med XML och har en bra struktur. Även att det är uppdelat ordentlig med information som hämtas från databasen och vad som visas för användaren. Det är då lätt att jobba med delarna var för sig. Det vill säga att det utgår från MVC-modellen.

En annan fördel med att använda XML är att det är enkelt att välja hur man vill visa data. Exempelvis transformera till JSON-format eller visa upp informationen. Detta kan vara en fördel när man vill visa information på många olika utkanaler alternativt om man har komplex datastruktur.

Nackdelar är att det kan bli många olika filer om man vill ha olika vyer för mobil och webbläsare på datorn, alltså många XSL-filer vilket blir jobbigt och inte alltid lätt att navigera mellan. Samtidigt som det är en nackdel, så är det bra att man har kontroll över vad som exakt visas på mindre skärmar.

Nackdel att det kan bli lite väl krångligt i XSL-filerna för dynamisk data när det gäller if-satser/choose. Med vanlig PHP blir dessa rader ibland mycket smidigare.

En annan nackdel är att det inte alltid är så lätt att debugga och hitta vart fel har begåtts.

En slutsats att dra är att det kan vara effektivt att bygga upp ett publiceringssystem på detta sätt men beror helt och hållet på användningsområde. Hur komplicerad är datan och vilka utkanaler är en stor faktor som ska tas i åtanke innan man bestämmer i vilket språk som ska utvecklas i. Om det är en komplicerad datastruktur och med många utkanaler kan det vara lämpligt med XML/XSLT medan om det är mindre komplex data med få utkanaler kan detta utvecklingssätt kännas lite väl krångligt och tidskrävande.