Dokumentation för Informationsapp.

Projektmedlemmar: Johan, Maria, Tommi, Jonna, Aaron  
WU15 - Kurs: ME1580

**Produkten**:

En webb- och mobilapplikation som ska ge och visa information som är kopplad till skateprojektet. Detta för att underlätta för utställningens besökare och ge underhållning genom foto, film och ljud  
  
**Produktens syfte**:

Att med hjälp av en digital plattform förbättra en utställning som är en analog plattform.

**Vår frågeställning**:

~~Hur skapar vi en stabil och säker DB och ett tillhörande API?~~ (Torsdag 22/9, ny frågeställning! Se nedan).

Hur skapar vi en säker webbapplikation med tillhörande inloggningssystem och hur hanterar vi kommunikation mellan olika API:er på webbapplikationen?

**Varför**:  
För att skapa förståelse för arkitekturen och kommunikationen mellan de olika delarna

**Projektmetod**: Scrumban med inspiration från XP (eXtreme programming).

**Scrummaster**: Johan

**Språk vi kommer använda**:

→ Python

→ flask → API

*Databas*:

→ Mysql

→ Json

→ AJAX

*Stack*:

→ LAMP // Alla ska sätta sig in lite mer i detta (Linux, Apache, MySQL, PHP)

**Analogt**:

→ UML

→ Loggbok/dokumentation

→ ER-modell

**Ev. hårdvara**:

→ RPi

**Features:**

Foto:  
→ Gallerifunktion  
→ Koppling till instagram och hashtag-flöde  
→ Webbappen kommer visa de senaste 1-5 fotona som laddats upp, istället för alla på en gång.  
→ Mobilappen kommer att ha ett eget galleri där bilder kan laddas upp direkt utan att gå genom instagram.  
→ Lagra bilder i DB tillsammans med namn på fotografer.

Film:

→ Koppling till YouTube / koppling med Bucketlist-projektet / koppling med webbapplikation?

→ Namn på filmskapare och medverkande i DB.

→ Länk till filmer/intervjuer i DB

→ Film uppdelat i två kategorier: film/intervju och film/event.

Hitta:

→ Google Maps-API

→ Karta över området där utställningen kommer vara, med utmarkerade platser samt vägbeskrivning (gps/orienteringsfunktion).

Schema:

→ “Spelschema” där varje användare kan favoritmarkera en speciell händelse.

→ Tid, datum, plats, vad.

→ Egen, självständig tabell i DB (tid, datum, plats, vad).

→ Information om vilken stad utställningen besöker under en specifik vecka.

~~SkateWiki:  
→ Ordlista med olika slang och uttryck som är specifika för skateboardkulturen.~~

~~→ Koppling till Bucketlist-projketet?~~ (Nybörjarapp-projektet tar detta istället, måndag 19/9)

Nyheter:

→ Senast inlagda foto, film, intervju (dessa tre kopplas ihop med Foto/Film), samt text och koppling till Twitter.

Vi hade först en idé om att det skulle gå att länka till blogginlägg, men vi ändrade oss och tog istället med en koppling till Twitter eftersom det känns mer relevant för oss att använda till detta projektet. Hashtaggen som används på instagram kan även användas på Twitter, vilket underlättar.

Sökfunktion:

→ Söka nyckelord på sidan

**Struktur för DB - vad som ska lagras.**

Nyheter:

- ID (pirmary key)

- Text

- M\_ID (foreign key) ?

- St\_ID (foreign key) ?

Schema:

- ID (primary key)

- Dagar

- Time/date

- Plats (var på utställningen)

- Text som beskriver vad som ska hända

- St\_ID (foreign key)

Städer:  
- ID (primary key)

- Stad (Namn på staden)

- Veckor

- Plats (var i staden)

Sociala media:

-

~~Media:~~

*~~Foto:~~*

~~- ID (primary key)~~

~~- URL~~

~~- Time/date~~

~~- Text (null)~~

~~- P\_ID (foreign key)~~

~~- St\_ID (foreign key)~~

~~- M\_ID (Samma som ID)(PK?)~~

*~~Film:~~*

~~- ID (primary key)~~

~~- URL~~

~~- Time/date~~

~~- Typ (Film eller foto)~~

~~Personer:~~

~~- ID (primary key)~~

~~- Förnamn~~

~~- Efternamn~~

~~- Nickname~~

~~Denna tabellen kopplas ihop med foto och film.~~

**Funktioner för API / Application Programming Interface.**

Nyheter:

*GET:*

St\_ID

M\_ID

text

*POST:*

text

Schema:

*GET:*

time/date

plats

text

*POST:*

time/date  
 plats

text

St\_ID

*PUT:*

time/date

plats

text

Städer:

*GET:*

Stad

Vecka

Plats

*POST:*

Stad

Vecka

Plats

*PUT*:

Vecka

Plats

Sociala media:

*GET:*

*POST:*

*PUT:*

~~Personer:~~

*~~POST:~~*

~~Förnamn~~

~~Efternamn~~

~~Nickname~~

~~Media:~~

*~~GET:~~*

~~URL~~

~~P\_ID~~

~~St\_ID~~

~~time/date~~

~~typ~~

~~text~~

*~~POST:~~*

~~URL~~

~~P\_ID~~

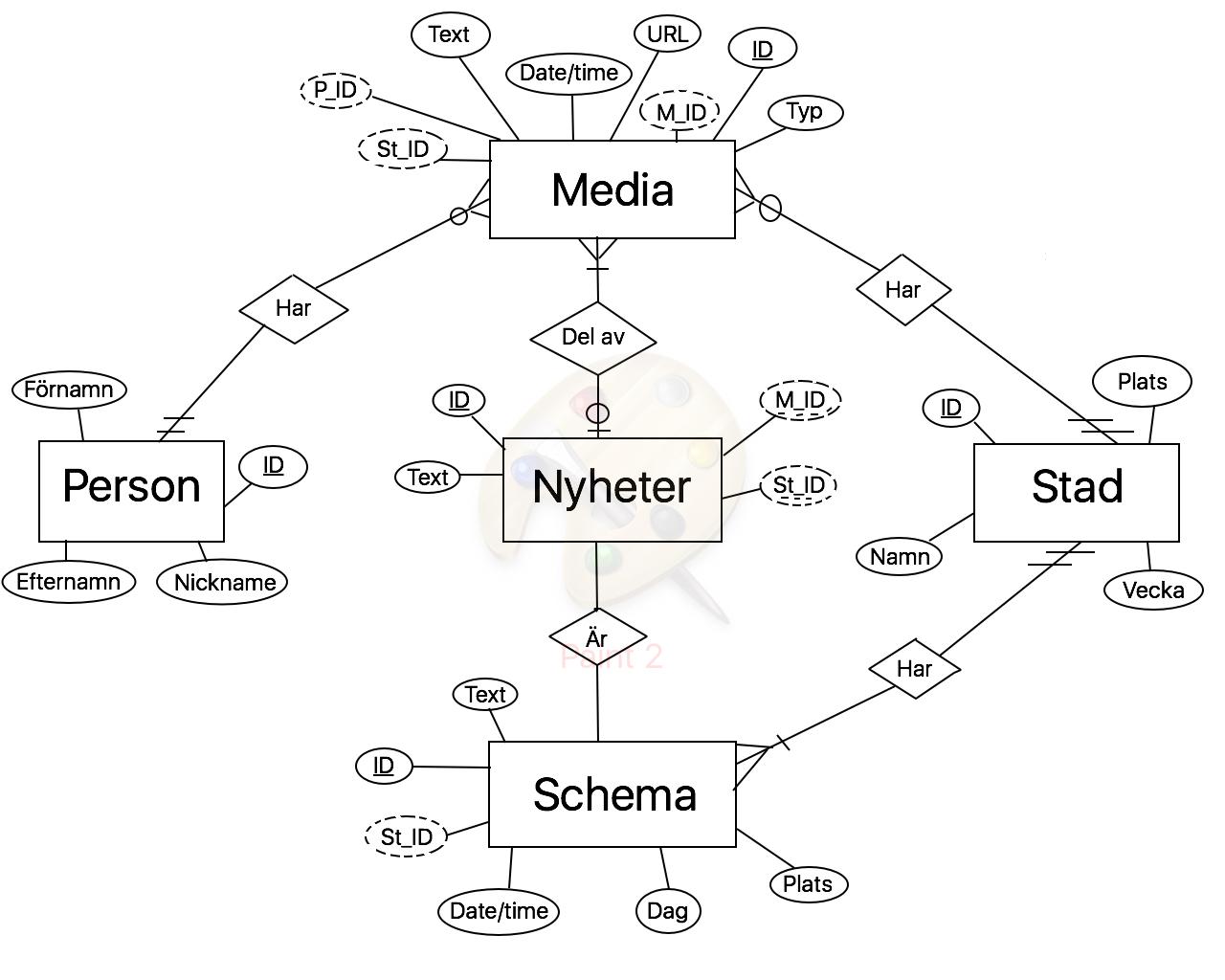
~~St\_ID~~

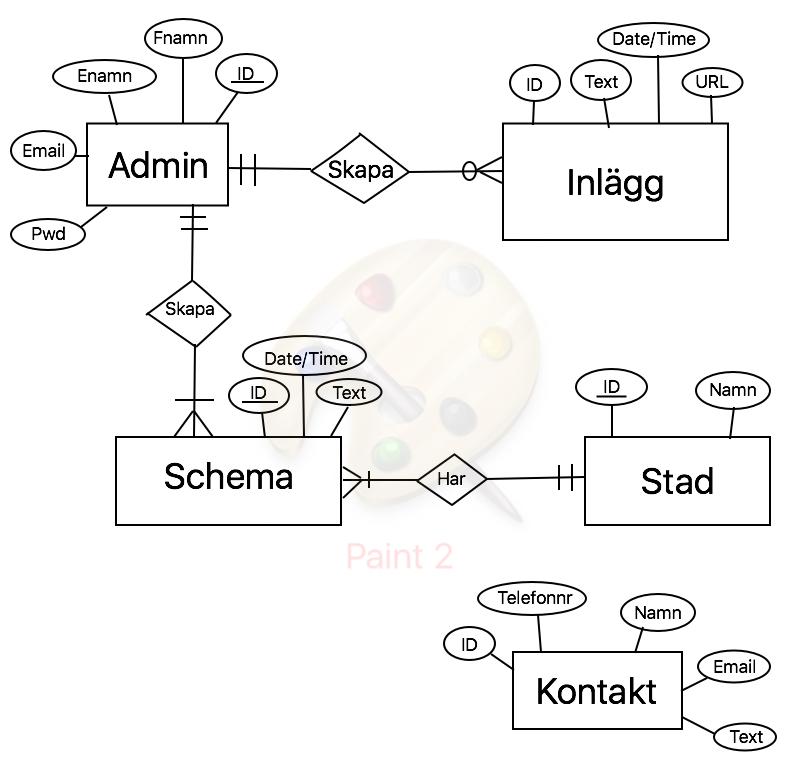
~~typ~~

~~text~~

**ER-modell**

*ER-modell, figur 1*

*ER-modell, figur 2*



Veckoplanering

**Vecka 37 / kursvecka 3 av 9:**  
VAD?

* ~~Vad ska undersökas - frågeställning?~~ - Fredag 16/9
* Planera sprintar – dela upp i mindre stories - (Pågående)

**Vecka 38 / kursvecka 4 av 9 / Sprint 0:**

VAD?

Prio 1:

* ~~Bestämma språk~~ - Måndag 19/9
* ~~Utveckla våra features~~ - Måndag 19/9
* ~~Struktur för DB, vad som ska lagras~~ - Tisdag 20/9
* ~~Funktioner för API~~ - Tisdag 20/9

Prio 2:

* ~~Läsa/undersöka DB och API, hur funkar det?~~ - Tisdag/Onsdag 20-21/9
* ~~Hur optimeras de?~~
* ~~ER-modell, rita upp~~ - Onsdag och torsdag 21-22/9
* ~~UML, rita upp~~ (Flyttat till nästa vecka)
* ~~Planera Sprint 1~~ (Flyttat till nästa vecka)

**Vecka 39 / kursvecka 5 av 9 / Sprint 1:**

VAD?

Prio 1:

* ~~Planera Sprint 1~~ - Tisdag 27/9
* ~~Kolla in API:er~~
* Få igång Twitter-flöde
* ~~Skissa upp flödesschema för appen~~ - Måndag 26/9
* ~~Göra inloggningssida och bloggpost sida~~
* ~~Börja på DB och kryptering av inlogg~~
* ~~Skapa layout för appen - HTML, CSS, JS~~
* ~~ER-modell~~ - Tisdag 27/9
* UML-diagram

**Vecka 40 / kursvecka 6 av 9 / Sprint 2:**

VAD?

Prio 1:

* Göra klart Twitter-API
* ~~Få färdigt DB-koppling~~ - Måndag 3/10
* Göra klart layout för webbapplikation (HTML, CSS, JS)
* Göra klart Inloggnings- och bloggpostsida för webbapplikation
* Göra klart YouTube-API