REHIT

Enginyeria de Software per Aplicacions Web

Joel Lamata i Martin

Elias Asskali Assakali

Jordi Tortosa Perez

Adam Dbouk Alberich

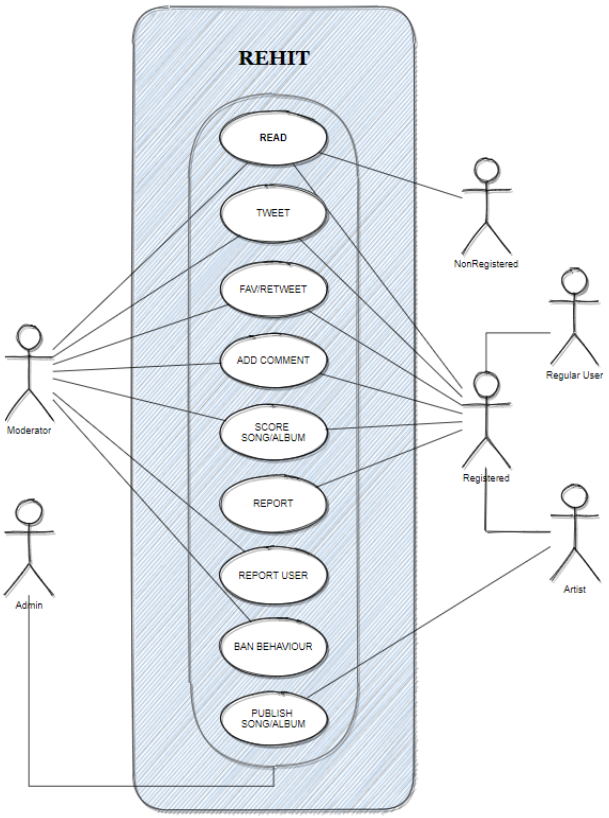
Índex

INTRODUCCIÓ

En primer lloc, al conèixer la idea del projecte van sorgir mil i una idees diferents, des de Twitter de videojocs a un Twitter tecnològic per compartir idees i projectes. Finalment, la idea de realitzar un twitter de música on poder compartir la música preferida de cadascú i alhora donar l'oportunitat a petits i grans Artistes de compartir la seva música i fer-la a arribar a molta més gent. Abans d'entrar en les característiques i les funcionalitats de l'aplicació web explicarem com funciona i que es pot fer en ella.[1] En primera instancia, mencionar que hi haurà tres tipus d'usuaris diferents: anònims, estàndard i artistes. Diferenciats segons les funcions que se'ls permet realitzar.

En primer lloc, els usuaris anònims només podran visualitzar l'aplicació, restringint tota mena de funcionalitat que involucri modificacions. En segon lloc, els usuaris estàndards i els artistes gaudeixen de totes les funcionalitats de la web. On podem trobar publicacions de tweets i seguir a nous usuaris.

A més, els artistes gaudeixen de la possibilitat d'ensenyar les seves cançons i de publicar-ne de noves. Per acabar, trobem que hi ha usuaris que són administradors encarregats de gestionar els perfils i tweet, un administrador té accés a tot i pot editar i borrar tota mena d'informació o tweet relacionat amb un usuari que no sigui administrador.

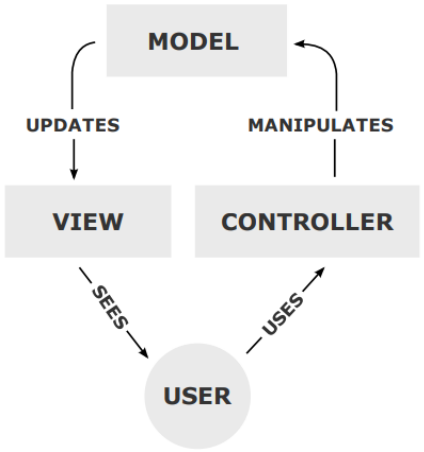


[1] Web application Use case diagram

MODEL VIEW CONTROLLER

Model View Controller (MVC) és un patró de disseny que permet organitzar millor el codi. MVC es divideix en tres parts: els models, els controllers i les views. Aquests tres elements formen un cicle que es retroalimenta a mesura que l'usuari interactua amb la web i executa funcionalitats. Identifiquem l'inici del cicle en la view principal de l'aplicació on haurà la primera interacció de l'usuari amb la web.

La View és on es mostra la informació, allà l'usuari fa les accions desitjades, llavors es passa la informació al controller qui s'encarregarà de manegar-la i modificar-la per tal de ja sigui fer canvis a la base de dades o distribuir el fluxe d'activitats a través de les diferents views de la web. En el cas de la modificació en base de dades, entra en joc la model on la informació actualitzada es donada per l'usuari a la view, recollida pel controller, manegada, s'actualitza la model i finalment s'executa el canvi de view. Per altra banda, en cas de que no hi hagi modificació com podrem veure en exemples que comentarem més endavant només es faria l'últim pas, encarregat de la distribució del flux de la web.



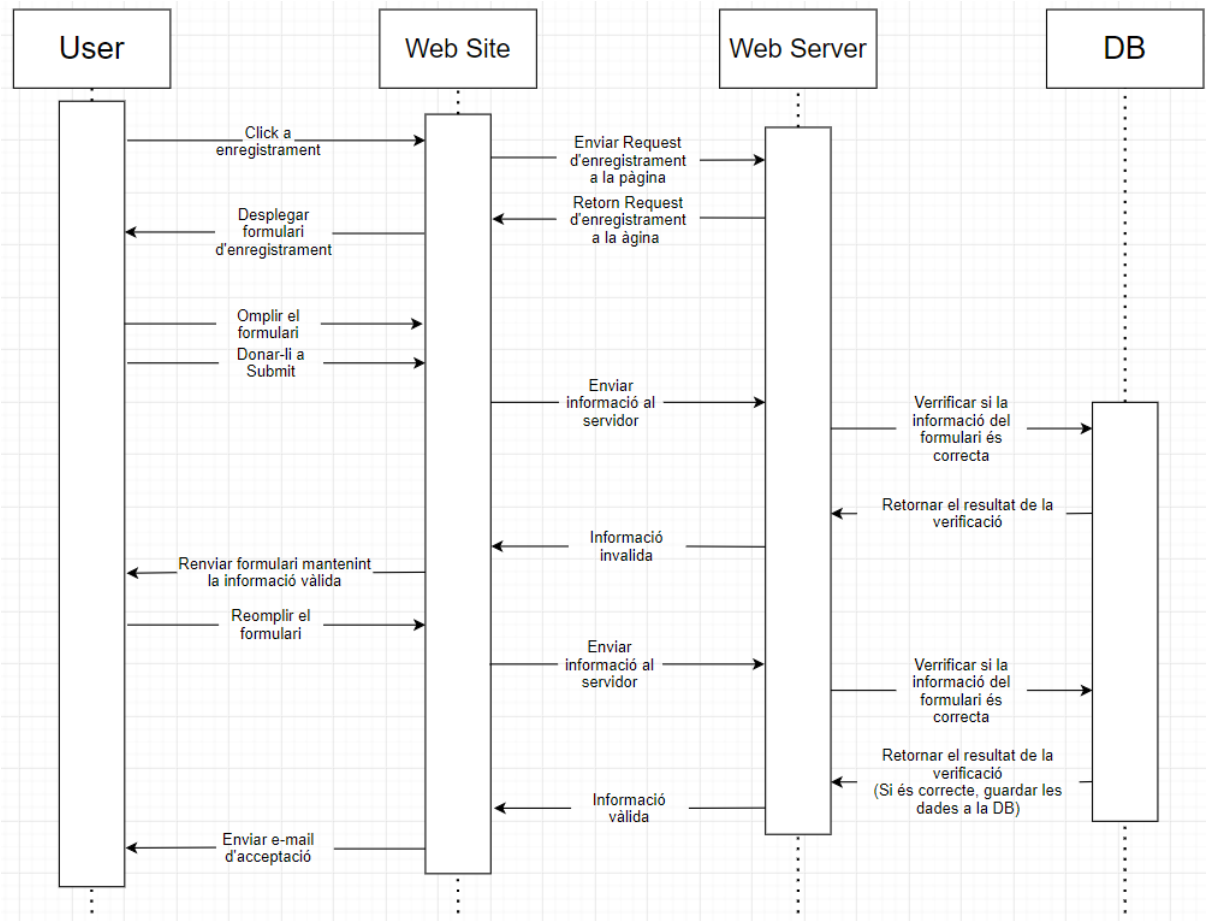
Com es mostra en la imatge següent, el fluxe d'activitats generals de la web es basa en un petit cicle entre la view, el controller, la model i finalment de nou a la view.

FUNCIONALITATS

Farem un explicació detallada de les funcionalitats de la nostra aplicació web en ordre cronològic d'implementació. Així podrem apreciar el desenvolupament de la web pas a pas i com han anat millorant totes les parts fins a ser tot una aplicació web en condicions.

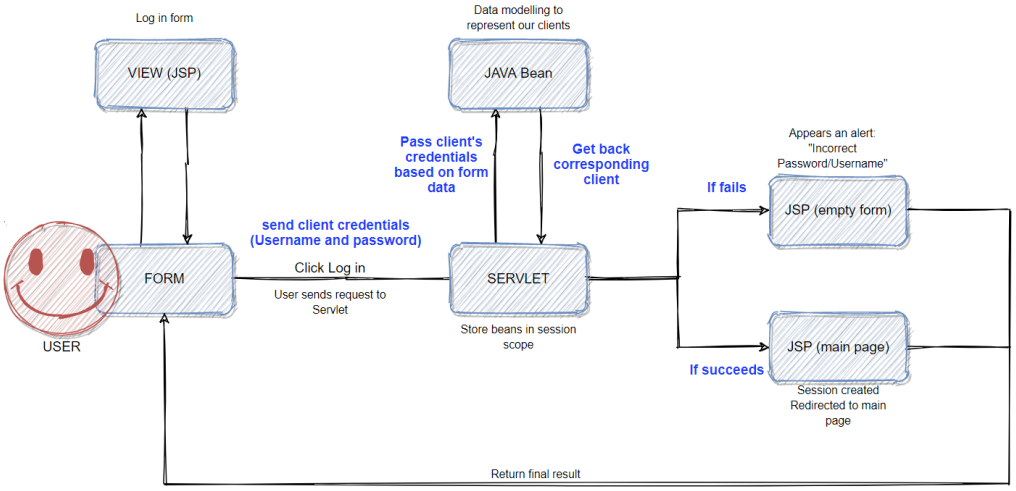
REGISTER FORM

En aquest apartat la part més important s'ha centrat en les validacions en client i en servidor, on en alguns casos hem arrossegat els coneixement de Parsley obtinguts als primers dos labs per poder tenir validacions eficients amb arxius javascript i poder tenir un codi més net i llegible. Com podem veure en la imatge següent es mostra el diagrama de seqüència de com s'executaran les accions tant en servidor com en client. On després de cada input es valida en client i finalment al clicar submit es fa la validació en servidor comprovant que tot sigui correcte i segueix el format establert prèviament. Aquesta validació en servidor comprova tant els patterns de la web com els de la base de dades.



LOGIN FORM

El següent pas en la realització del projecte va ser l'implementació del login form. Aquí apareix el concepte de Model View Controller (MVC) que com hem explicat anteriorment aquest patró de disseny es basa en un cicle de funcions per tal d'executar totes les funcionalitats necessàries de l'aplicació web. El login form implementa al igual que el register form les comprovacions de les dades que s'introdueixen, fent que segueixin els patterns establerts.



A continuació, en el MVC diagram[2] anterior podrem veure l'implementació i funcionament del MVC en el login form. Com veiem, comença tot a la view on l'usuari introduirà les seves credencials, aquesta informació es passa al servlet encarregat d'administrar la request, en aquest cas el Login Controller. Un cop al servlet es comprova la informació amb la model i la base de dades, en aquest punt l'usuari rebrà una response o una altre en funció de si la validació ha sigut exitosa o no. Per tant, en cas afirmatiu, l'inici de sessió es completa i l'usuari accedeix a l'aplicació web, sinó per contra l'usuari tornarà al login form on un error li comunicarà que les credencials no són correctes.

En aquesta part entra en joc no només el MVC sinó també el concepte de AJAX. On durant les diferents accions es busca recarregar la mínima part possible de l'aplicació web, per exemple en el resultat final a la part de login trobem les 4 parts diferenciades, i si per exemple, es clica a Most Retweeted Tweets només es recarrega la part central de la web (#content) mantenint les barres laterals i la barra horitzontal superior.

Aquest exemple és vàlid per la resta de la web, més endavant entrarem en profunditat en el tema.

MVC EN LA NOSTRA APLICACIÓ WEB

En gairebé tot el codi utilitzem el MVC com a patró de disseny ja que ens simplifica l'estructura que tindrà el codi. Com hem definit abans, el MVC funciona tal que la View mostrarà informació a l'usuari, aquesta podrà ser utilitzada per ell, pero no estara utilitzant la view si no el controller, que aquest s'ocuparà de rebre aquesta informació, analitzar la i modificar la model, i aquesta última será quí contingui la información que la view mostrarà.

Un exemple seria la funcionalitat del botó Follow. La view ens mostrarà aquest botó per pantalla, amb la seva forma, posició i color corresponent, quan nosaltres, o l'usuari, interaccioni amb el botó, en veritat no està fent res amb la view, si no amb el controller. Aquest, amb l'ajuda de l'arxiu index.jsp, s'activarà, rebrà la informació de quin botó ha sigut pres, i amb aquesta modificarà la model i a la nostra base de dades afegint un nou follow a la taula corresponent. Finalment, la part de la web on ha ocorregut la interacció, serà refrescada, agafant nova informació que ha sigut modificada en la model, i mostrant la correctament, en el cas del Follow, l'usuari deixa de ser recomanat per seguir, i es mostraran els tweets del nou follow a la feed.

Finalment, degut a la temàtica de la web hem trobat necessari crear una nova model que seran les Cançons del artistes i per tant, també és necessari un manager que ajudi a controlar el seu funcionament i els servlets corresponents per a dur a terme el funcionament.

[2] MVC diagram login example

BASE DE DADES

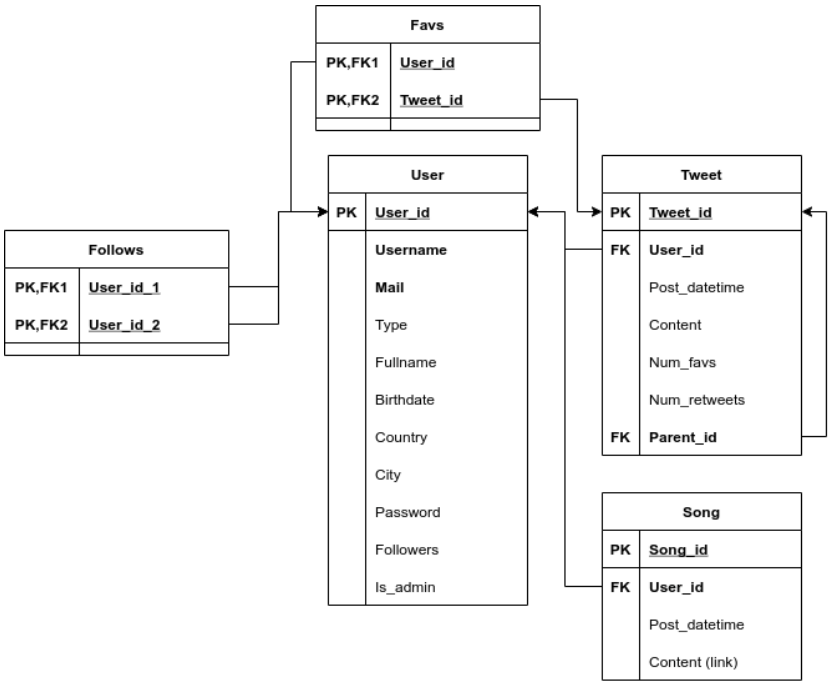
En quant a la base de dades hem creat diferents taules per indicar entitats i relacions entre entitats, les taules més importants són “users” que identifiquem amb un enter únic, en la qual guardarem la informació personal de l’usuari entre altre informació com el tipus d’usuari, el número de seguidors i si és un administrador o no.

Sense tweets la web no tindria contingut així que necessitem una taula tweets que identifiquem amb un id únic per cada tweet. Cada tweet tindrà informació com la data de publicació, el contingut, una referència a l’usuari que l’ha penjat, el numero de favs i retweets i en cas que sigui un retweet tindrà una referència al tweet que esta retuitejant.

Els usuaris artistes podran publicar cançons al seu perfil inserint un enllaç de youtube a l’apartat de cançons. Així doncs, la taula cançons tindrà un identificador únic, la referència a l’usuari que l’ha penjat, la data en la que s’ha penjat i el contingut, que serà un enllaç embed (format necessari per poder reproduir-los) que serà de la plataforma youtube.

Per altre banda també guardarem en una taula els favs, on tindrem una referencia a l’usuari que ha donat el fav i una al tweet al que s’ha donat fav. També farem el mateix pels follows, en una taula guardar una referència a l’usuari que està seguint i una referència a l’usuari seguit.

Finalment, necessitarem triggers que s'activen quan algú fa like, retweet, segueix a algú… Aquest triggers sumaran els contadors de likes, retweets, seguidors d’aquesta forma mantindrem actualitzades aquestes dades. També, hem creat un procedure retweet ja que no podem actualitzar el contador de retweets d’un tweet a la vegada que s'està inserint una columna a la mateixa taula tweets (mysql bloqueja la taula per lo que no es pot fer amb un trigger), aquest procedure afegirà el retweet a la taula tweets i un cop afegit aumentará el contador de retweets del tweet pare.



CASCADING STYLE SHEETS

Per començar, vam decidir que la pàgina es digués ReHit, basat en el lema de “Re-listen your favourite hits”. El logotip de ReHit són uns auriculars simbolitzant la pau que un sent al posar-se'ls i escoltar les seves cançons preferides. Per tant, la base de la paleta de colors de ReHit busca transmetre un feeling semblant al que hem mencionat pel logotip, la paleta és una composició de negre i blanc, amb aquests 2 colors es busca expressar serietat i tranquilitat, tot i que deixem marge a colors vius en algunes zones de la web on sigui necessari ressaltar-les i fer saber al usuari que allò és important o volem que llegeixi allò. A més a més, les tipografies que hem escollit són Fantasy per el títol i subtitol i Poppins-Regular,sans-serif per el text normal, la raó per la qual hem escollit aquestes dues tipografies són per augmentar l’esperit estil juvenil i urbà alhora que no surt de l'estàndard per poder captar el major número de gent.

Una altre implementació relacionada amb el disseny és la estructura visual de la pàgina, l'hem distribuït mitjançant 3 columnes igual que el template proporcionat pero amb certes diferències que ara es comenta a la part de layout. D’aquesta part caldria comentar la imatge de fons que roman estàtica i complementa a la perfecció amb la paleta de colors escollida..

Per acabar entrarem més en el concepte de CSS tècnic, on vam voler tenir un tracte personal amb el codi CSS per poder personalitzar i poder modificar tot al nostre gust, sabent que això dificulta en part el desenvolupament gràfic i de disseny de la web, encara que en bona part de l’aplicació l’ús de la llibreria W3C està vigent. El template va ser modificat i adaptat als nostres requeriments i necessitats, per exemple, aplicant la paleta de color seleccionada, l’estructura que fos més adient amb l’estil de la web i els colors vius en els botons per tal de mantenir l’estil juvenil i alegre.

Per altre part, l’objectiu principal del CSS en la nostra aplicació web és no fer pensar a l’usuari, que no hagi d'endevinar com funciona sino que amb una ullada ja sàpiga fer tot el que vulgui. LLavors una vegada varem decidir com treballar el css, vam començar a aprendre quines funcionalitats i característiques té la llibreria W3C per poder fer la interacció molt més amena i satisfactòria per a l'usuari de la web. Fent que l'experiència de l'usuari amb la web sigui la millor possible brindant una interacció intuïtiva i fàcil de fer servir.

Finalment, l’estructura de l’arxiu de CSS serà amb classes i especificacions pròpies per a certes parts, per tant, només queda assignar la classe o especificació corresponent a cada part, paràgraf, div… de les nostres views. L’accés als elements de la web és mitjançant els botons, per tant, purament visual, i al fer-los servir s’executarà la funció obvia que es mostra.

NAVEGACIÓ I AJAX

La nostra web està dividida principalment en quatre parts: el menú (barra col·locada a la part superior des d'on podem accedir als nostres tweets, els usuaris als que seguim o a un buscador entre d'altres), la columna de l'esquerra (on es mostra la informació de l'usuari i publicitat), la part central (on tindrem la informació principal de la pàgina com els tweets o la informació a editar, ja sigui d'un tweet o d'un user), i per acabar la columna de la dreta (on es mostren els usuaris que no seguim i se'ns permet accedir al seu perfil).

En quant al AJAX de la pàgina quan utilitzem les funcionalitats del menú en general només es recarrega la part central, ja que la majoria de funcionalitats tenen a veure amb ordenar i mostrar tweets i usuaris (informació que es mostra a la part central).

Quan interactuem amb els tweets (donar Like, Retweet, etc.) es carrega només la part central per tal de no re-carregar informació que es romandrà igual , durant la implementació vam intentar carregar únicament el tweet en comptes de content però no ens vam sortir.

A l'hora de fer Follow a un Usuari es clica a follow i es recarrega la columna de la dreta, ja que s'actualitzen els usuaris als quals no seguim i també es recarrega la part central, la qual ens redirigirà a la View Tweets i es mostraran tots els tweets dels usuaris que seguim actualitzats amb el nou usuari inclòs.

Al no carregar tots els elements de la pàgina cada cop que interactuem amb aquesta estem optimitzant el funcionament de la web.