

CATXMEIFYOUCAN

MEMÒRIA DEL PROJECTE



23-05-2014
Sara Esteban
Guillem Muñoz

BREU DESCRIPCIO DEL TREBALL

El projecte és un joc en temps-real de dos jugadors, implementat amb Yii, que posseeix vídeo xat i xat. Això està realitzat amb WebRTC juntament amb la llibreria Peerjs.

ÍNDEX

1. INTRODUCCIÓ DEL TREBALL.....	4
2. MATERIAL PRESENTAT.....	5
3. PLA DE TREBALL A LLARG PLAÇ.....	6
4. PLA DE TREBALL DIARI.....	7
5. AUTOVALUACIÓ.....	8
- Objectius i grau d'assoliment	
- Incidències i resolució	
6. COS DEL TREBALL.....	10
1.- WEBRTC.....	10
1.1. Descripció general	
1.2. Teoria del funcionament	
1.3. Especificacions	
2.- PEERJS.....	11
2.1. Descripció general	
2.2. Teoria del funcionament	
3.- YII.....	12
3.1. Descripció general	
3.2. Teoria del funcionament	
4.-	
AWS.....	13
4.1. Descripció general	
4.2. Teoria del funcionament	
5.- CRUGE.....	14
5.1. Descripció general	
5.2. Teoria del funcionament	
6.- WEB SERVICE.....	15
6.1. Descripció general	
6.2. Teoria del funcionament	
7.- BASE DE DADES.....	16
7.1. Descripció general	
7.2. Teoria del funcionament	
8.- ESQUEMES.....	18
7. CONCLUSIONS.....	20
8. REFERÈNCIES DOCUMENTALS.....	21

1.- INTRODUCCIÓ DEL TREBALL

El projecte consisteix en una web on el tema principal o la novetat és un joc en temps real, aquest realitza una connexió client a client. Això és possible gràcies a l'API WebRTC i una llibreria de javascript (peer.js) que implementa aquesta API.

A més a més, en el joc real-time podràs gaudir d'un xat i una vídeo trucada mentre estàs realitzant la partida per si vols comentar com va el joc o vols veure el teu contrincant.

La part de vídeo trucada i xat és possible també en mòbil. El joc només és operatiu des d'un ordinador.

Quan vols jugar Run2live el primer que et trobes és una sala amb la llista dels usuaris que estan a la web i quins estan disponibles per jugar i quins no ho estan.

La web també conté jocs en local, ja que el tema de la web són els jocs i aquesta web està feta amb Yii.

Una altra característica és que posseeix un apartat on administra els usuaris i aquí només pot accedir l'usuari admin i conté un apartat amb el rànquing de les partides del joc en temps real (Run2live).

2.- MATERIAL PRESENTAT

- Manual tècnic (en format digital PDF)
- Manual d'usuari (en format digital PDF)
- Memòria del treball (en format digital PDF i **imprès**)
- Annexos (en format digital PDF)
- Guió escrit de l'exposició basat en el memòria
- Presentació multimèdia de l'exposició

3.- PLA DE TREBALL A LLARG PLAÇ

TASQUES	TEMPS EMPLEAT
Planificació dels objectius	Els dos primers dies
Anàlisi dels objectius	Els dos primers dies
Disseny de com fer el desenvolupament	Els dos primers dies
Desenvolupament del joc	Les tres setmanes
Montar servidor web	La primera setmana
Implementació WebRTC + PeerJS	Les tres setmanes
Xat	La primera setmana
Vídeo trucades	La primera setmana
Disseny de la web	L'última setmana
Registres d'usuaris	La segona setmana
Implementació Framework (yii)	La segona setmana
Documentació del projecte	L'última setmana
PowerPoint de presentació	Última setmana




ACTIVITATS	MAIG								
	SETMANES								
	1		2		3		4		
Anàlisi dels objectius									SARA+GUILLEM
Disseny de com fer el desenvolupament									SARA+GUILLEM
Desenvolupament del joc									SARA
Montar servidor									GUILLEM
Implementar joc a real-time (peerjs)									SARA
Xat									SARA
VideoCall									SARA
Disseny de la web									SARA + GUILLEM
Menus de l web									GUILLEM+SARA
Registres d'usuaris									GUILLEM
Ranking									GUILLEM
Implementar joc a mb el Framework									SARA
WebService									SARA + GUILLEM
Yii Framework									GUILLEM
Documentació Projecte									SARA+GUILLEM
PowerPoint de presentació									SARA+GUILLEM

4.- PLA DE TREBALL DIARI

1 ^a Setmana	<ul style="list-style-type: none"> - La primera setmana va estar destinada a la planificació del projecte i la repartició de les tasques. - Realització del xat, vídeo trucades i posada en marxa del servidor amb els serveis adients. - Començar a desenvolupar el joc en temps real amb les tecnologies web rtc i la llibreria peerjs.
2 ^a Setmana	Continuar amb el desenvolupament del joc i creació del yii amb l'extensió cruge. Implementació del joc amb Yii i adaptar algunes coses a la manera que ho fa el framework i realització de la connexió automàtica a través de la sala (llista d'usuaris).
3 ^a Setmana	<ul style="list-style-type: none"> - Documentar, fer jugable el joc i realització del rànquing de partida. - Modificació del disseny de la web, modificant coses de la plantilla utilitzada i adaptar-la al nostre gust.

5.- AUTOVALUACIÓ

Objectius principals i grau d'assoliment:

Objectius	Grau d'assoliment
Implementar framework	Implementació Framework YII ✓
Administració d'usuaris	Implementació extensió Cruge ✓
Vídeo trucades	Implementació Vídeo trucades (WebRTC) ✓
Xat	Xat implementat amb la tecnologia WebRTC ✓
Joc temps real, diferents nivells, animacions, etc.	<p>Hem implementat el joc en temps real però hem tingut alguns problemes durant el seu desenvolupament que no hem pogut integrar totes les coses que volíem per fer-lo més atractiu i més addictiu.</p> <p>El joc ha quedat simple a comparació de totes les coses que volíem implementar.</p> 
Ranking de les partides	<p>El rànkung de partides podria ser millorable ja que ha quedat simple al tenir el joc simple no hem implementat totes les dades que volíem afegir a la taula rànkung. Tot i que és funcional.</p> 
Connexió automàtica del joc	<p>Hem pogut implementar la connexió automàtica entre els jugadors, creant una sala on surten tots els usuaris amb ajuda del webservice tot i que té alguns defectes petits, està implementat de bona manera i és funcional.</p> 
WebService	Implementació correcte del webservice per passar el token i dades del joc per guardar-les al rànkung. ✓

Posada en producció a través de AWS	La posada en producció ha estat satisfactòria per tant el nostre web està totalment en producció i accessible per a tothom. Ja que hem registrat la màquina AWS en un domini que ens proporciona Dot TK.
-------------------------------------	--



Incidències i resolució:

Les incidències que ens hem anat trobant són les següents:

- Prendre consciència del funcionament de la tecnologia WebRTC amb la llibreria PeerJS i començar el desenvolupament del joc, xat i videocalls.
- Alhora d'implementar tota la part de connexions a temps real amb el yii framework, o sigui integrar el que és el joc, xat i videocalls vam tenir alguns problemes, ja que havíem d'adaptar certes coses del php i javascript, jquery a la manera de fer del framework en qüestió, tal com les consultes a la base de dades, agafar els usuaris registrats en la web, etc.
- Implementar tot el que és la web al AWS també vam tenir algun problema que es va poder resoldre ràpidament.
- En el joc hem tingut problemes alhora d'implementar coses en temps-real, ja que dóna problemes en el fet d'enviar dades d'un client a un altre i que tot estigui alhora i no hi hagi un descontrol.

El fet de fer servir javascript, canvas, etc. Al ser un llenguatge interpretat el JavaScript ens ha faltat acabar-lo de polir al màxim possible, ja que poder hi ha alguna part del codi que podria ser millorable i fer-la d'una manera molt més efectiva i més ràpida. Per tant podríem dir que el que ha donat més problemes és el joc, com que no teníem consciència absoluta del funcionament de l'API hem hagut d'agafar-ho amb més paciència per entendre el que estàvem fent.

- Un altre imprevist és alhora del rànding el fet d'agafar variables de javascript per implementar-les a la nostra base de dades i ensenyar-les a través del yii, hem hagut de fer servir un webservice i vam tenir algun problema.

6.- COS DEL TREBALL

1. WebRTC

1.1. Descripció general

WebRTC és un **Framework** obert destinat a les **comunicacions en temps real**. Aquest inclou els blocs de construcció fonamentals per les comunicacions d'alta qualitat a la web, tals com la xarxa, els components d'àudio i vídeo.

Per accedir aquest component és fa a través d'un api de JavaScript, que permet als desenvolupadors implementar fàcilment la seva pròpia aplicació.

WebRTC està sent **normalitzat en l'àmbit d'API** en el W3C i en el protocol de la *IETF* (*Internet Engineering Task Force*, <http://www.ietf.org>).

Pel nostre projecte fem servir WebRTC per desenvolupar el joc, el xat i les vídeo trucades, sense necessitat de cap mena de plugin ni fer cap tipus d'instal·lació.

Aquesta API ofereix un ventall de possibilitats pels desenvolupadors d'aplicacions web, ja que poden realitzar aplicacions en temps real, **sense** necessitat de **plugins**, **descàrregues** i **instal·lacions**.

El **propòsit** d'aquest projecte és ajudar a construir una plataforma RTC (Real Time Connection) que **funcioni** a través de **múltiples navegadors i diverses plataformes**.

És un projecte de **codi obert**, que està sostingut per Google, Mozilla i Opera. A més de la pàgina oficial de WebRTC és fa càrrec l'equip de **Google** (<http://www.webrtc.org>).

Aquest també porta un suport per a servidors proxy, està basat en la força del navegador. Els **paquets** que integra WebRTC són: **vídeo**, **àudio** i **xarxa**.

1.2. Teoria del funcionament

WebRTC s'encarrega de realitzar la connexió entre dos clients a través d'una llibreria JavaScript per fer possible aquesta connexió. La llibreria que hem fet servir és PeerJS.



1.3. Especificacions

Suporta:

- Google Chrome
- Mozilla Firefox
- Opera

2. PeerJS

2.2. Descripció general

PeerJS és una **llibreria** de JavaScript que **integra WebRTC** al navegador per proporcionar una API de connexió completa , configurable i fàcil d'utilitzar d'igual a igual.

Aquest es connecta a un **servidor PeerServer**, tot i que cal tenir en compte que no hi ha dades peer-to-peer a través del servidor, ja que aquest només actua com **agent de connexió**.

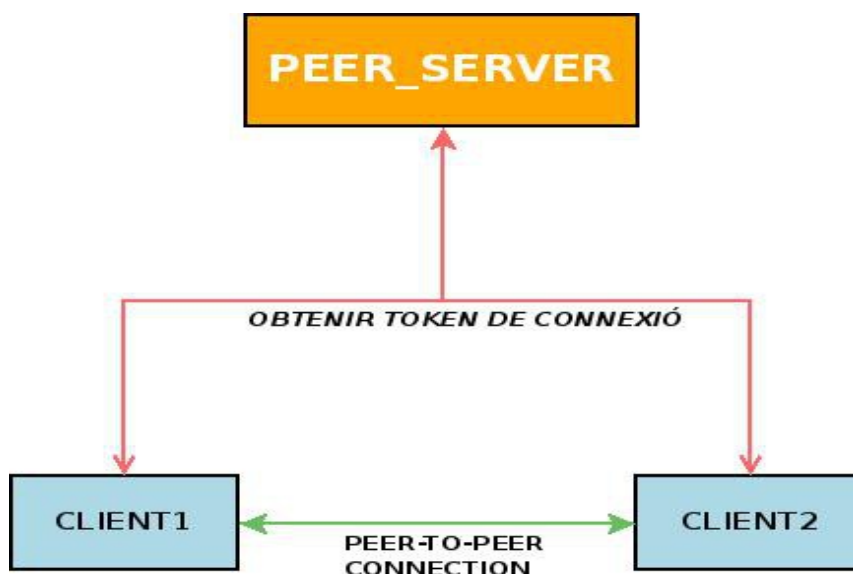
Aquesta llibreria ens ofereix l'opció de fer servir un **PeerServerNube**, és un servidor **gratuït** allotjat al núvol de PeerServer, i des de aquí pots aconseguir la **APIkey** dels servidor Nube.

Si el que volem és muntar el propi servidor, ens donen la possibilitat de fer-ho, PeerServer és de codi obert i està escrit en Node.js, així que és pot executar fàcilment.

Un avantatge d'aquesta llibreria és que ens ha facilitat molt les connexions peer-to-peer de dades, vídeo i trucades d'àudio.

2.3. Teoria del funcionament

El funcionament de PeerJS consisteix a qu els clients es connectin al servidor i aquest obtenen un token que els identifica i llavors els clients s'identifiquen amb aquest token i un cop s'identifiquen l'un amb l'altre estableixen la connexió peer-to-peer.



3. Yii Framework

3.1. Descripció general

Yii és un **Framework** orientat a objectes, de **software lliure**, d'alt rendiment basat en components PHP i framework d'aplicacions web.

Les seves característiques són les següents:

- Disseny : model-vista-controlador
- Database Access Object (DAO), Active Record i migració de base de dades
- Integració amb jQuery
- Entrades de formulari i validació
- Widgets d'Ajax, com l'autocompletat del camp
- Suport d'autenticació
- Personalització d'aspectes i temes
- Especificació i administració de peticions Web Service
- Suporta traduccions, format de data i hora, format de números i localització de la

vista

- Esquema de caching per capes.
- Control d'errors i logging
- Generació automàtica de codi (CRUD)
- És possible utilitzar codi PHP o Zend Framework en una aplicació Yii.

3.2. Teoria del funcionament

En el nostre projecte Yii ens ha facilitat molt el treball entema d'administració d'usuaris i disseny.

Hem utilitzat un **tema** que és diu **Hebo**, que porta implementat **Bootstrap**, **html5**, **css3**. És un disseny **adaptat** també per **android** i és totalment modificable.

Això ens ha anat molt bé ja que així tenim un disseny ben implementat i construït.

Pel tema d'administració d'usuaris hem utilitzat una **extensió** anomenada **cruge**, la qual ens ajuda a tenir **controlat** el tema de **SQL injection**, i qualsevol forat de seguretat està protegit amb aquesta extensió.

Així tenim una pàgina web totalment segura i robusta.

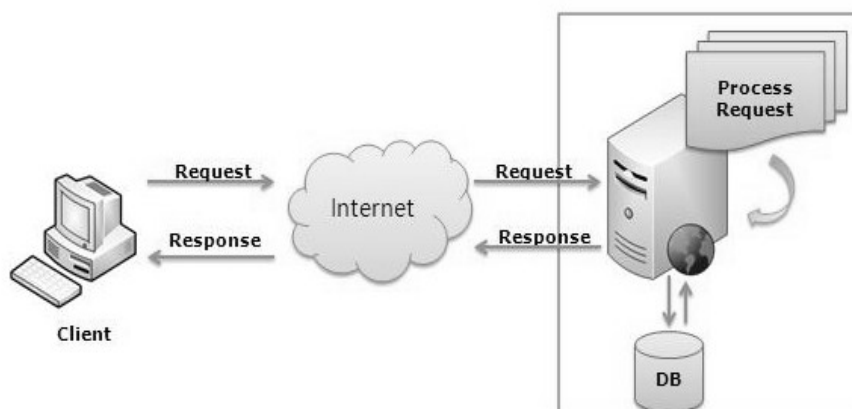
4. AWS

4.1. Descripció general

Amazon Web Service (AWS) és una **colecció de serveis** d'escriptori **remot** també nomenats serveis web, que en conjunt formen una plataforma al núvol, aquests serveis són proporcionats per Amazon.

4.2. Teoria del funcionament

AWS és el nostre servidor on està **allotjada** la nostra **pàgina web** amb el joc real-time, per tant en el nostre cas fa de **servidor web** i des de aquí amb la pàgina Dot TK assignem un **domini** a la nostra màquina per posar-la en **producció** i poder accedir des de qualsevol lloc.



5. Cruge

5.1. Descripció general

Cruge és un mòdul que proporciona una **extensió** pel **mecanisme** estàndard de **Yii Auth**.

Utilitzant Cruge, pots centrar-te en les necessitats de la teva aplicació, deixant a cruge les tasques comuns com : registre, recuperació de contrasenyes, editor de perfils, entre d'altres característiques.

Algunes de les característiques:

- control d'accés
- Registre, recuperació de contrasenya, perfils (camps personalitzables)
- Pantalles de gestió d'usuaris
- Suport d'autenticació remota (Ex. Facebook, Google)
- Gestió de la sessió
- Menú Creator
- API per administrar usuaris
- Compatible amb jQuery, FancyBox,Bootstrap, vistes personalitzables...

5.2. Teoria del funcionament

Cruge és una extensió que ha estat fonamental en el nostre projecte. Ens ha fet tot el **control de sessions** de la nostra pàgina web. Per tant tenim el **registre d'usuaris** totalment controlat i llavors tot el tema de recuperació de contrasenyes i **protecció de SQL injection** queda protegit amb aquest mòdul.

En el nostre projectes es registren usuaris els quals assignem el **rol** de **jugador**per què puguin **accedir** a l'apartat del **joc en temps real**. Els jugadors que siguin visitants en el nostre web i no estiguin registrats no podran veure aquest apartat de la web. A més a més la part d'**administració d'usuaris** està només **disponible** per l'usuari **admin** , per tant els **altres usuaris** queden **exclosos** de veure aquest apartat de la webi no podran accedir-hi.

Gràcies a les diferents funcionalitats de cruge hem pogut fer servir funcions específiques per implementar amb el joc, xat i el videocall.

6. WebService

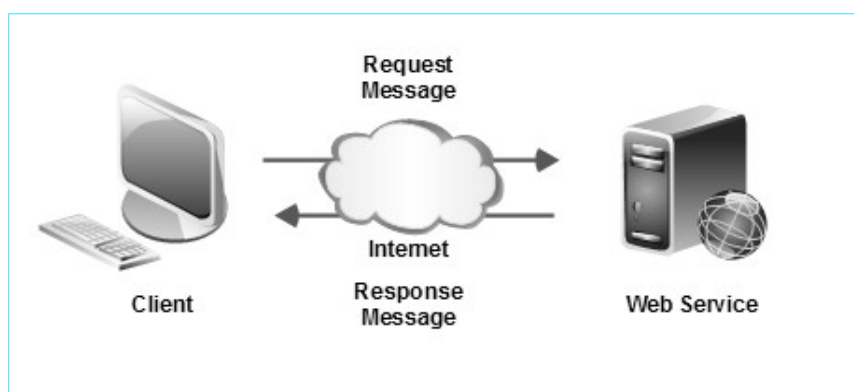
6.1. Descripció general

WebService és una tecnologia que **utilitza** un conjunt de **protocols** i **estàndards** que serveixen per **intercanviar dades** entre aplicacions. Es poden transferir dades entre diferents aplicacions fetes per diferents llenguatges de programació, aquestes dades poden ser intercanviades en xarxes entre ordinadors com Internet.

6.2. Teoria del funcionament

En el nostre projecte fem servir webservice per **transferir** el **token** que proporciona el servidor peerServer **per realitzar la connexió**, aquest token el **desem** a la **base de dades** per després gestionar-lo en el nostre codi de la connexió entre els usuaris. Cada token va canviant cada cop que s'inicia una nova partida o simplement s'actualitza, per tant aquest codi ha d'estar sempre actualitzat a la nostre base de dades i tenir la informació actualitzada per què la connexió sigui satisfactòria.

També fem servir aquest webservice per transferir el resultat del joc i poder realitzar el rànking amb les partides entre els usuaris implicats i els seus resultats.



7. Base de dades (MySQL)

7.1. Descripció general

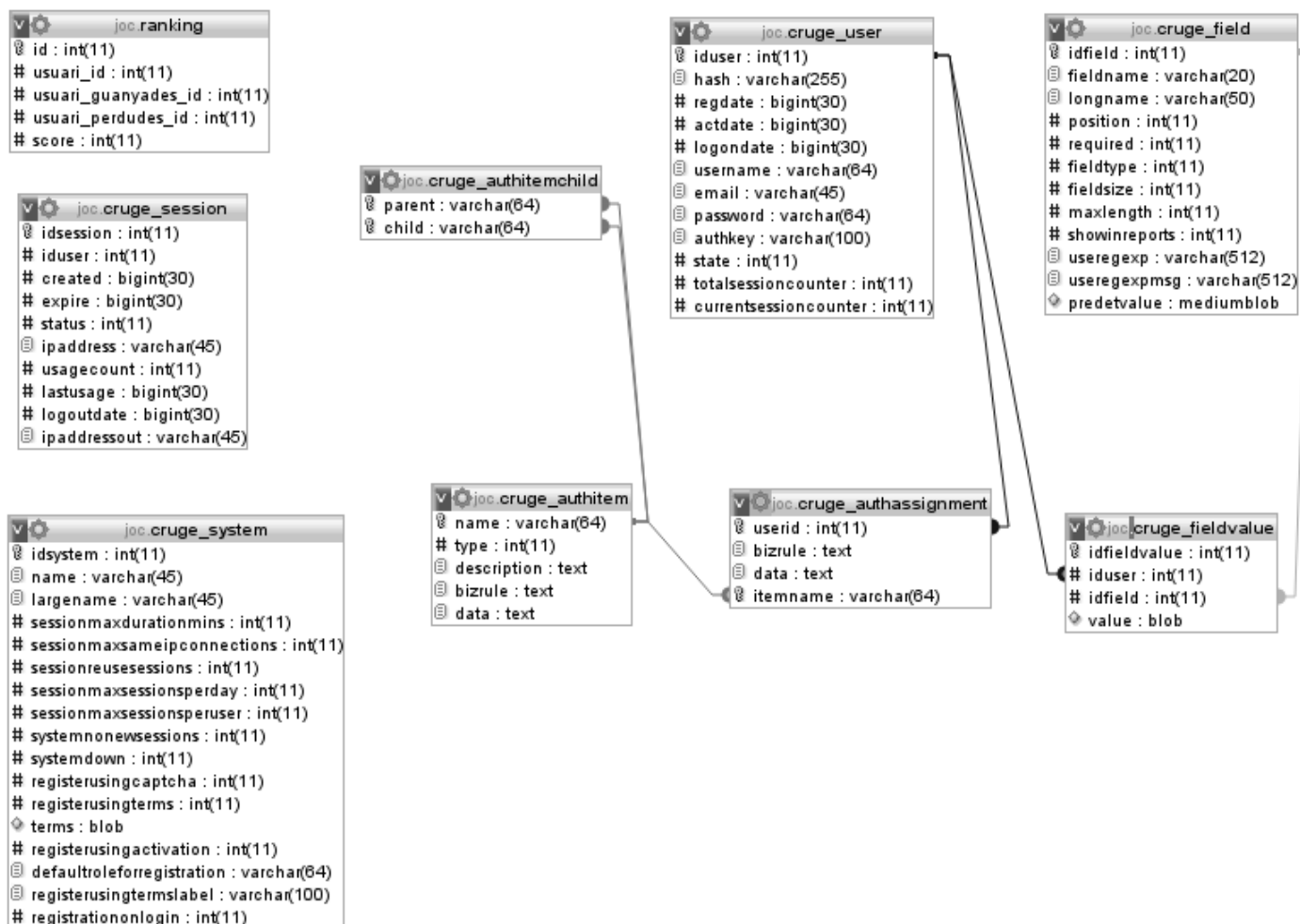
Com hem fet servir l'extensió de cruge, aquest mòdul ens proporciona una base de dades amb el registre d'usuaris i el seu control.

Tot i que hem modificat aquesta base de dades per tal que de la taula on es guarden els usuaris tingui un nou camp que és hash i llavors hem afegit una taula que es diu Ranking.

Aquesta base de dades el que fa és emmagatzemar el registre d'usuaris.

7.2. Teoria del funcionament

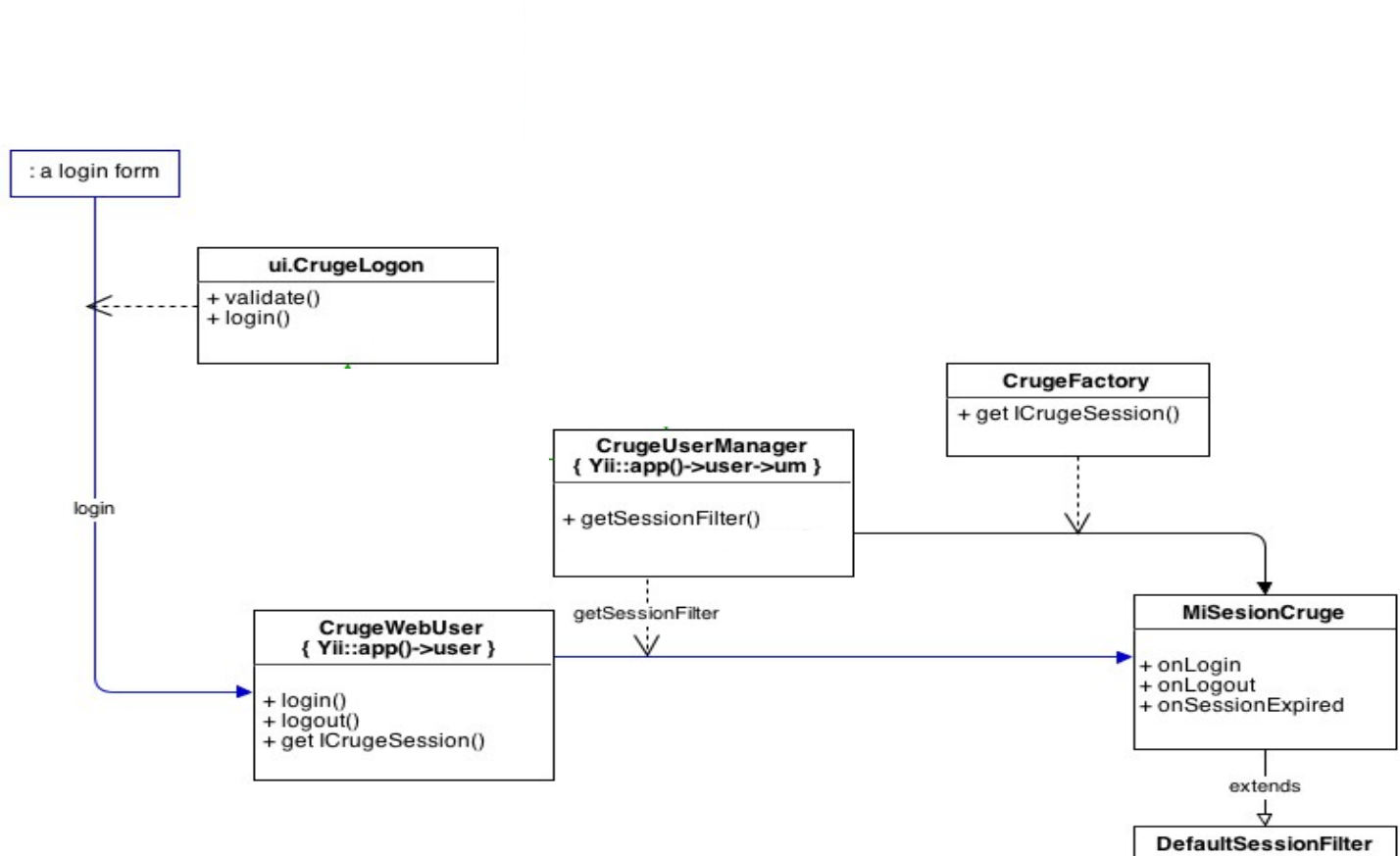
Estructura de la base de dades:



Passos que segueix el cruge:

Seqüència 1: L'usuari fa click a LOGIN al formulari d'inici de sessió:

1. loginformfa instància \$model basada en CrugeLogon, per validar les credencials fa model.validate, i tras l'èxit invoca a model.login()
2. model.login() invoca a Yii::app()->user->login(), solicita CrugeUserManager(getsessionFilter), per tant aquesta li dona una instància de MiSesionCruge
3. De tornada a CrugeWebUser.login, utilitza un filtre per informar el login satisfactori.



8. Esquemes d'arquitectura del sistema

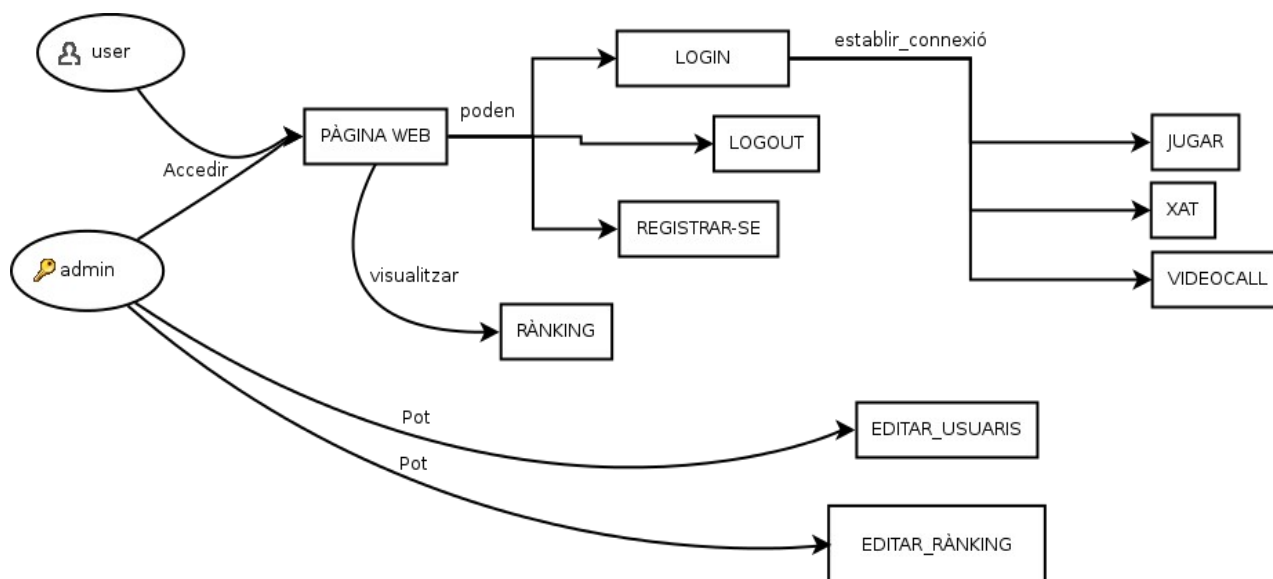
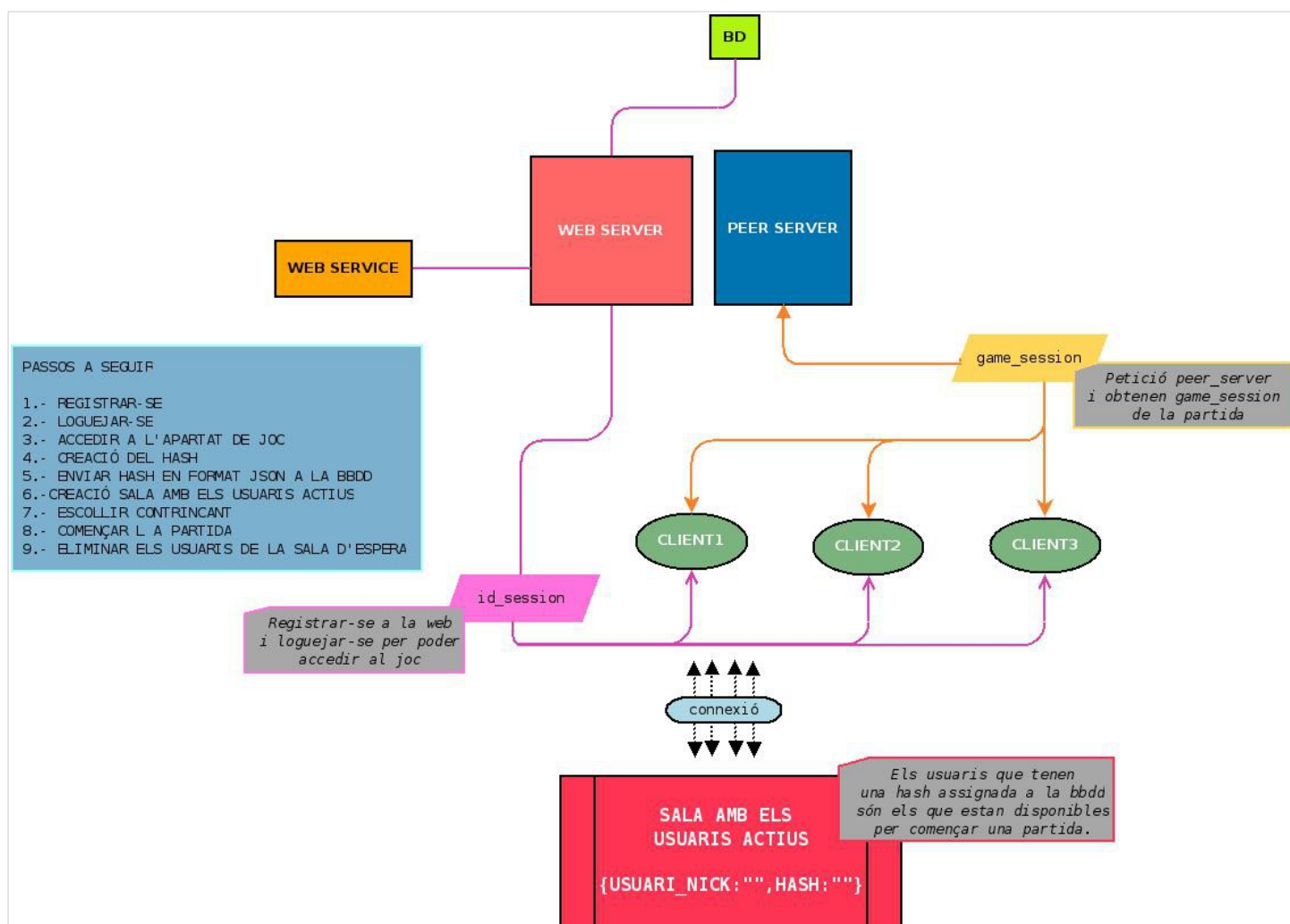


Diagrama casos d'ús – Navegació pel lloc web.

En aquest diagrama podem observar quines són les accions que pot realitzar l'usuari, tant si ets visitant, o l'administració del web.

Et mostra totes les possibilitats d'accions que pots fer.



- El treball consta d'una base de dades on fem tot el registre d'usuaris.
- El servidor web on tenim també el web service i la web allotjada.
- Fem servir el PeerJS que és un servidor gratuït per poder realitzar la connexió real-time
- Tenim els clients que es connecten al web.



7.- CONCLUSIONS

Hem après a treballar en equip per poder dur a terme totes les tasques que teníem en aquest projecte.

Hem assolit uns coneixements sobre la comunicació a temps real que ens cridava l'atenció i teníem curiositat per saber com era el seu funcionament, hem augmentat els coneixements sobre el framework yii i veure el ventall de possibilitats que tenim amb aquest framework i com ens pot estalviar el temps per a realitzar algunes tasques bàsiques com pot ser el registre d'usuaris, etc.

Crec que els coneixements que hem adquirit són interessants i ens poden servir en un futur. També hem après a ser autònoms, ja que ens hem hagut d'informar per entendre tot el funcionament de les comunicacions en temps real per poder començar a realitzar el projecte amb certa lògica per no anar desorientats.

8.- REFERÈNCIES DOCUMENTALS

- <http://yiiframeworkenespanol.org/wiki/index.php?title=Cruga> (Cruga -yii)
- <http://stackoverflow.com/>
- <http://peerjs.com/docs/#api> (Peerjs)
- <http://peerjs.com/docs/#start> (Peerjs)
- Recursos propis