Pasul 1. Intelegerea domeniului

Descrierea aplicatiei: O aplicatie Web ce reprezinta o retea sociala care sa permita utilizatorilor sa indice ce jocuri video s-au jucat, sa primeasca recomandari de alte jocuri in functie de preferintele lor actuale, sa ofere recenzii/ratinguri jocurilor, sa interactioneze cu recenziile celorlalti utilizatori, sa comunice intre ei(asemanator ca la email)

Actorii aplicatiei: utilizator(anonim sau logat)

-actiuni: sa-si poate modifica profilul (date personale, jocuri video preferate)

trimiterea de mesaje altor utilizatori

urmarirea unui utilizator

sa ceara recomandari de alte jocuri generate automat in functie de preferintele actuale

sa ofere recenzii/ratinguri jocurilor existente

(si altele)

(toate actiunile de mai sus sunt valabile doar pentru utilizatorii logati)

-actiuni utilizator anonim: el poate doar sa vizualizeze resursele existente pe site, dar nu si sa le modifice

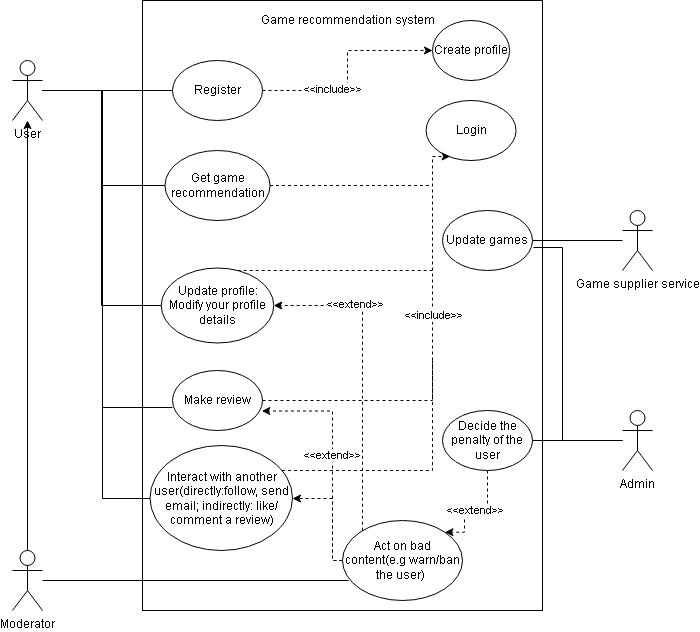
moderatori - aprobarea recenziilor, stergerea acestora, avertizarea/banarea

unui utilizator, in plus fata de ceea ce poate si un utilizator obisnuit (un moderator este un utilizator)

administrator - adaugarea/scoaterea de jocuri din baza de date, stergerea utilizatorilor(cu un motiv intemeiat oferit de moderatori, practic de actiunile sale negative pe site)

serviciu extern – se ocupa de actualizarea jocurilor de pe site (preluarea lor de pe internet)

Scenarii de utilizare(diagrama)



In aceasta diagrama se pot observa urmatoarele: practic toate actiunile unui utilizator obisnuit sunt conditionate de faptul ca s-a logat sau nu. De asemenea, actiunile sale care ar putea produce continut deranjator poate fi verificat de catre moderator si in functie de severitatea situatiei, adminul

ar putea fi anuntat pentru a stabili decizia finala in ceea ce priveste eventuala pedeapsa pe care o va primi userul. (pentru cazuri relativ grave)

Un utilizator poate sa faca recenzii pentru un anumit joc, sa interactioneze cu recenziile altor utilizator(sa dea like, comment), sa dea follow unui utilizator, sa-i trimita un mesaj.

Jocurile de pe site vor fi actualizate in mod automat de catre un serviciu extern, sau alternativ

manual de catre administrator.

Pasul 2. Popularea bazei de date

Baza de date ar putea fi populata cu utilizatori, jocuri.  
Recenziile sunt create de catre utilizatori. Fiecare recenzie este identificata in mod unic(o recenzie apartine unui anumit utilizator).  
De asemenea, mesajele sunt trimise de la un utilizator la altul(fiecare utilizator isi va putea vedea mesajele primite si trimise pe o anumita perioada de timp)

Un joc poate fi jucat de catre mai multi utilizatori. De asemenea, un utilizator poate juca mai multe jocuri.

Metoda de populare: fisiere CSV (LOAD CSV, APOC sau neo4j-admin tool)

Pasul 3. Intrebari ce pot fi adresate domeniului.

Ce jocuri se joaca un utilizator?

Care jocuri ale unui anumit utilizator sunt de genul X?

De cine este urmarit un anumit utilizator? Pe cine urmareste un utilizator?

Care este cel mai popular joc? (jocul care este jucat de cei mai multi utilizatori)

Care este jocul cu cel mai bun scor? (scorul este determinat de media tuturor ratingurilor acelui joc) (aceasta intrebare poate fi considerata pe urmatoarele intervale de timp: all-time, last week, las t month, last year etc.)

Care sunt mesajele trimise/primite ale unui utilizator? (pe o anumita perioada de timp, de ex. 3 luni)

Care este cea mai apreciata recenzie pentru un anumit joc(sau cea mai slaba, sau cele mai recente – mai multe tipuri de filtre ar putea fi aplicate la aceasta intrebare)

Care sunt comentariile/like-urile unei recenzii?

Pasul 4. Stabilirea entitatilor (de reflectat in continuare)

User; Game;

Review(nod intermediar) (trebuia cumva exprimat faptul ca un user poate face cel mult un review asupra unui joc – in alte cuvinte, un user nu poate face mai multe review-uri la un anumit joc) – id unic

De asemenea, un user nu poate face un review la un joc pe care nu-l joaca(adica in cazul in care nu exista relatia PLAYS intre user si game)

Message(nod intermediar)

Comment(nod intermediar)

Like(relatie)

Observatii:

Review nu poate fi relatie, deoarece in cazul in care ar fi, nu am mai putea exprima faptul ca un

utilizator a dat like sau a comentat la o recenzie a altui utilizator.

DE FACUT: stabilirea proprietatilor finale ale user-ului si ale jocurilor(momentan vor fi alese doar cateva proprietati pentru a minimaliza implementarea bazei de date; pe parcurs, acest aspect va fi refactorizat)

Stabilirea modului in care se modeleaza media recenziilor unui joc anume(solutie: fiecare joc are asociat

un nod unic numit Average care este la randul lui asociat tutoror recenzilor jocului respective, pentru a fi

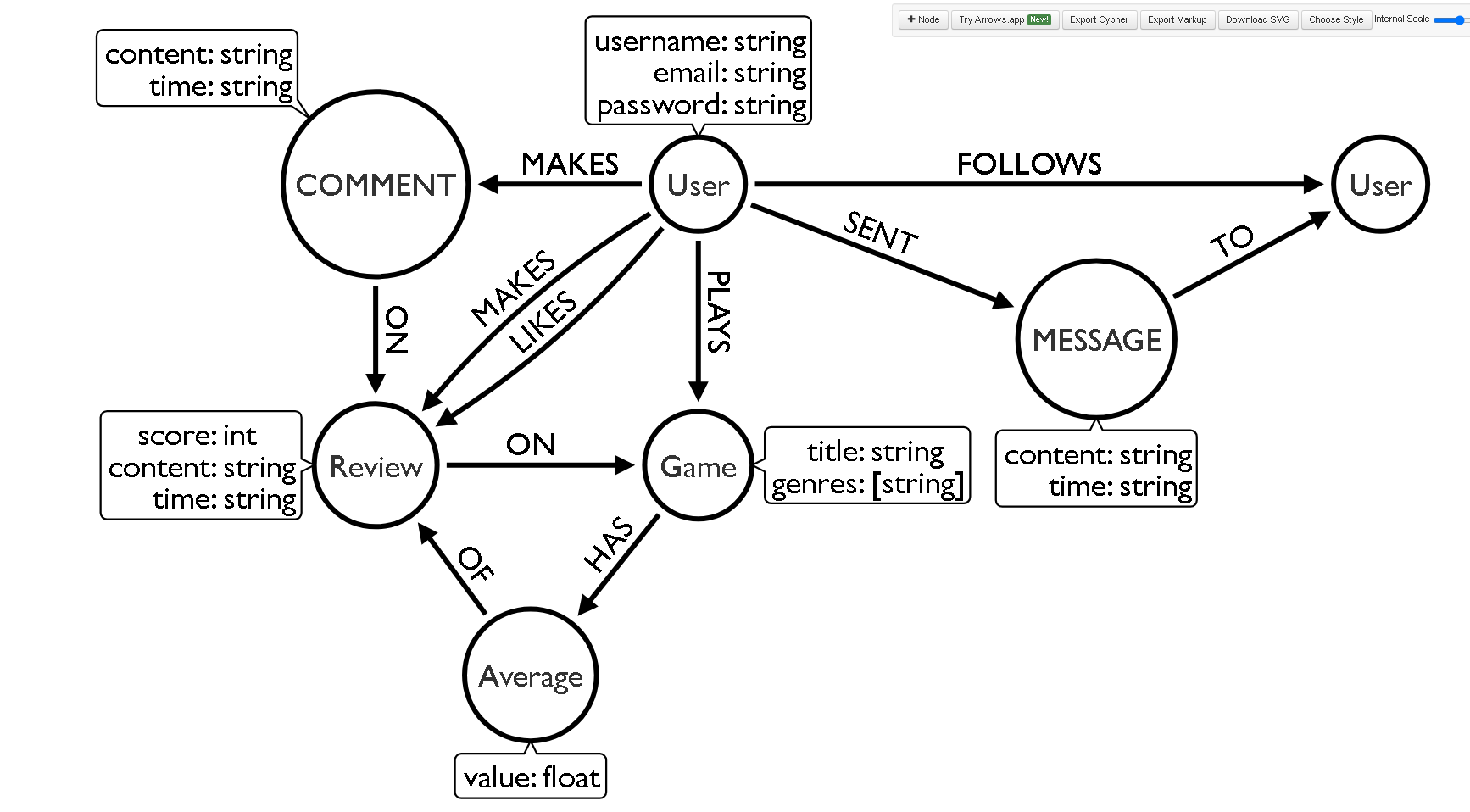
calculata media acestor recenzii; de asemenea, Average poate contine si alte valori medii de pe alte servicii: IGN, Steam etc.)

Pasul 5. Stabilirea relatilor(conexiunilor) intre entitati (se vor vedea pe model)

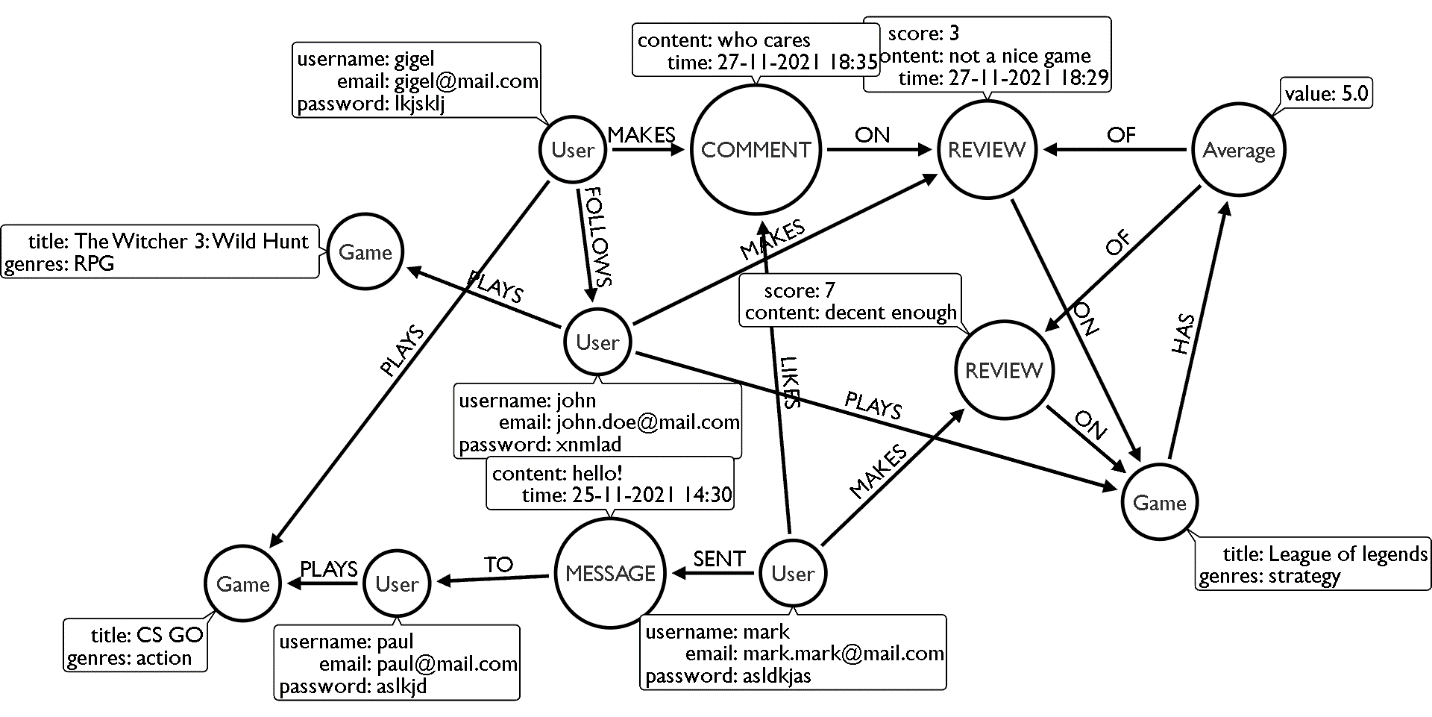
Pasul 6. Testarea intrebarilor pe model. (implementarea efectiva a bazei de date)

Pasul 7. Testarea scalabilitatii. (implementarea efectiva a bazei de date)

Modelare (in general)



Modelare(exemplu – ca si cum ar fi populat)



Constrangeri

Fiecare user are un id unic. (bifat)

Fiecare joc are un id unic. (bifat)

Username-ul unui user este unic. (bifat)

Un user trebuie sa ‘joace’ un joc inainte sa-i poata face review. (de facut cu trigger/procedure conditionale APOC)

Un user poate face cel mult un review la un joc, iar acel review este assignat in mod unic unui User(un review nu poate fi asociat la mai multi utilizatori) (verificarea aceasta se face prin unicitatea id-ului review-ului

Fiecare review are de asemenea un id unic(cel mai probabil compus din id-ul user-ului si id-ul jocului ) (id-ul acestuia a fost setat ca Node key in baza de date, insa trebuie vazut cum se face compunerea acestui id din id-ul userului cu cel al jocului)

In ceea ce priveste comment-urile, un user poate face mai multe comment-uri la acelasi review, insa bineinteles ca un comment nu poate fi asociat la mai multi utilizatori. Asadar, fiecare comment va avea un id(compus din id-ul userului si un id-unic asociat comment-ului) (asemanator ca la review)

In schimb, un user poate trimite unui alt user mai multe mesaje. Acestea sunt diferentiate de continut si de momentul in care au fost trimise. Pentru siguranta, fiecare mesaj va avea o proprietate in plus, un id unic, insa acesta nu va avea legatura cu continutul si momentul cand a fost trimis mesajul. (va fi compus din id-ul expeditorului, destinatarului si id-ul mesajului propriu zis) (de stabilit cand/cum se face compunerea acestui id)

Fiecare joc va avea cel mult o conexiune cu nodul Average(prin relatia HAS). Aceasta conexiune depinde de existenta a cel putin a unui review a jocului respectiv. Asadar, daca jocul respectiv are cel putin un review, atunci acesta va avea o conexiune cu nodul Average, iar calcularea efectiva a proprietatii efective a acestui nod se face prin calcularea mediei tuturor conexiunilor intre acel nod Average cu acele review-uri(prin relatia OF) asupra jocului caruia se calculeaza media.(!!!Atentie: cel mai probabil, acest aspect va fi refactorizat ulterior) . In caz contrar, daca nu exista niciun review asupra unui joc, atunci nu ii va fi calculata media. (de facut cu trigger)

Relatia de follows nu este reciproca.(de ex: daca A follows B, nu neaparat se intampla B follows A)

Fiecare user trebuie sa aiba un id, username, email si parola(proprietati obligatorii) (bifat)

Fiecare review trebuie sa aiba un scor si timpul in care a fost trimis(proprietati obligatorii), dar nu si continutul efectiv al acestuia(proprietate optionala) (bifat)

Fiecare comment trebuie sa aiba proprietatile content si time. (bifat)

Fiecare message trebuie sa aiba content si time. (bifat)

Comment-urile pot fi vazute de toata lumea(utilizatori logati/nelogati), insa mesajele pot fi vazute doar de catre expeditor si destinatar. (de vazut ulterior)

Fiecare joc are proprietatile title si genres(!!!posibil sa fie si altele ulterior), + proprietati optionale(de stabilit). (bifat – proprietatile obligatorii)