

SweatIT

RAPORT DE PROIECTARE

GameVision

Martie 2016

1. Scopul aplicației

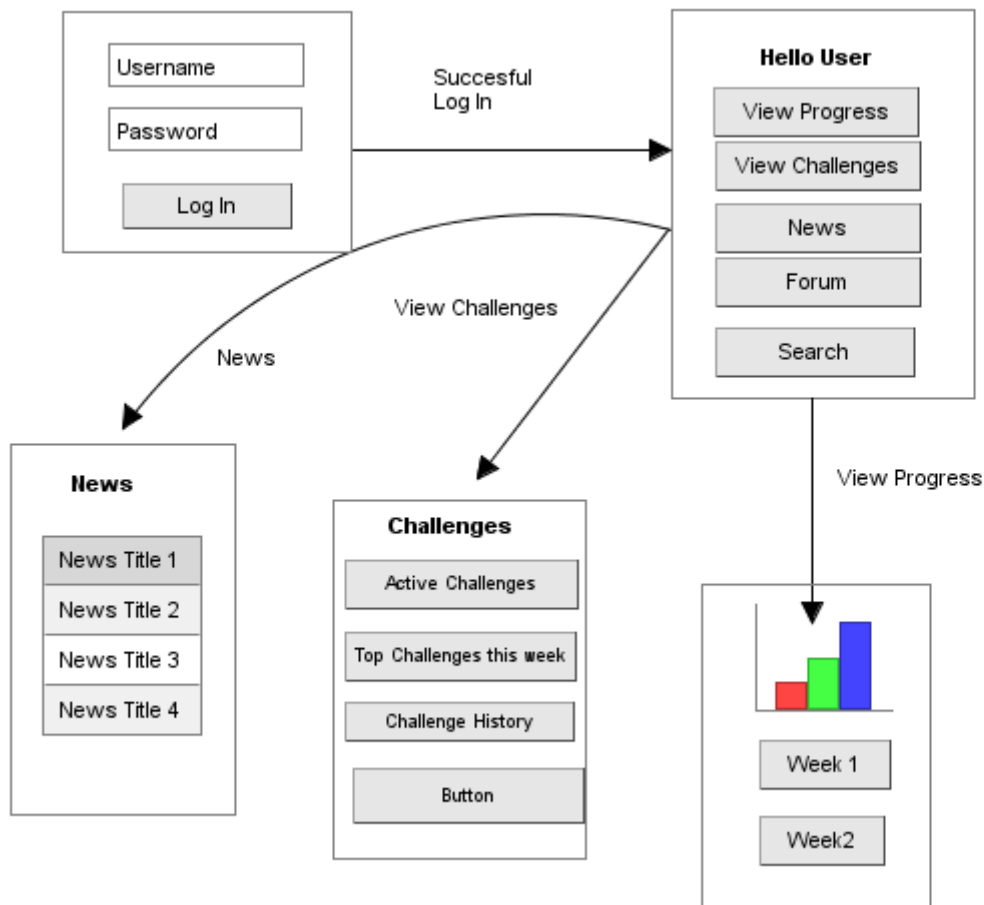
Proiectul nostru constă într-o aplicație mobilă pe platforma Android. Aplicația are ca scop încurajarea unui stil de viață sănătos prin mișcare fizică frecventă și eliminarea sedentarismului.

2. Aria de acoperire a aplicației

Aplicația va funcționa pe platforma mobilă Android pentru a fi la îndemâna utilizatorilor. Aplicația este destinată persoanelor de toate vârstele ce își doresc să își îmbunătățească modul de trai prin activitate fizică. Aceasta îi ajută în special pe cei ce își pierd motivația datorită stilului de viață cotidian.. Deoarece aplicația folosește tehnica *gamification*, o să implementăm un sistem de progres si challenge-uri pentru a angrena cât mai mult utilizatorul și a-l face să își îndeplinească scopul, acela de a face mișcare. Mai mulți useri o să poată concura între ei pentru puncte și pentru un ranking mai bun. Această competiție va fi una constructivă, fără premiu material, cu unicul scop de a încuraja mișcarea.

3. Perspectiva funcțională

Perspectiva funcțională a aplicației este reprezentată prin wireframe-ul de mai jos. Aici este descrisă interfața cu utilizatorul și structura generală a aplicației.



4. Perspectiva structurală

Prima componentă majoră a aplicației este motorul de log-in. Utilizatorul trebuie să își introducă credențialele în interfață după care se face un request către server care verifică datele introduse cu înregistrările din baza de date pentru a identifica utilizatorul. Altă componentă cheie este sistemul de înregistrare a progresului. Când utilizatorul face un request din aplicație la server pentru progres, serverul interoghează baza de date pentru a vedea realizările acestuia după care realizează câteva rapoarte statice ce îi sunt afișate grafic utilizatorului pentru a îi fi mai ușor să urmărească.

O altă componentă este sistemul de challenge-uri. Acesta definește provocări în care utilizatorii pot să se angreneze, să definească ei noi provocări și să urmărească un istoric al acestora împreună cu un clasament general al utilizatorilor ce s-au descurcat cel mai bine.

O altă componentă este motorul de știri acesta folosește un API a unui site de știri din domeniul sportului, nutriției etc pentru a genera content dinamic utilizatorului.

Componenta ce se ocupă de forum ține în baza de date topicurile de discuții împreună cu comentariile aferente. Utilizatorii pot să propună noi teme de discuții, să comenteze la unele deja existente. Sistemul face diferențe între anumite tipuri de utilizatori, existând utilizatori cu privilegii mai mari ce au rolul de moderator și pot șterge comentariile ce le consideră nelalocul lor.

Motorul de căutare realizează o căutare în baza de date pentru anumiți utilizatori.

5. Perspectiva comportamentală

Utilizatorul își introduce credențialele, dacă este înregistrat în baza de date atunci va fi logat dacă nu, va fi întrebat dacă vrea să își facă cont. Utilizatorul logat va primi un meniu unde trebuie să aleagă acțiunea pe care vrea să o întreprindă. Are de ales între 5 opțiuni : Progresul, Challenge-urile, Știrile, Forumul și Căutarea.

Dacă alege progresul atunci este făcut un call către server pentru a prelua informațiile referitoare la activitățile/challenge-urile întreprinse de utilizator și va primi un grafic ce descrie acest progres.

Dacă alege Challenge-urile atunci va intra într-un meniu unde poate să aleagă între a vedea challenge-urile deschise, a vedea un top general al concurenților și a vedea un istoric al provocărilor personale. Dacă nu există challenge-uri active va primi un mesaj de eroare.

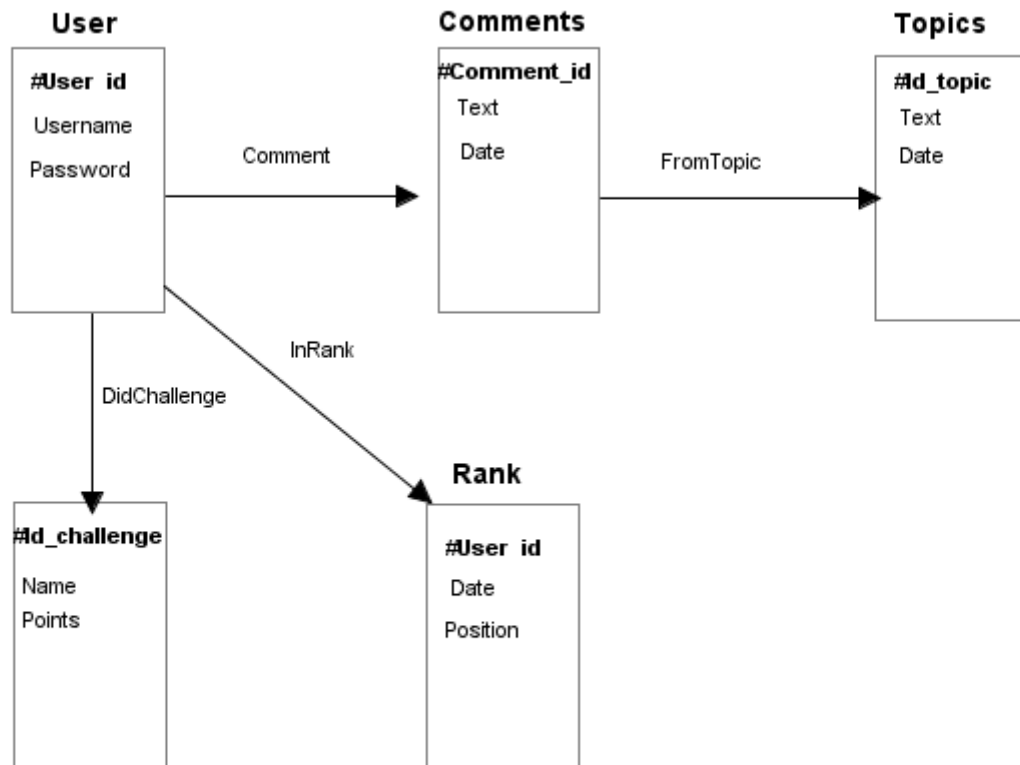
Dacă știrile atunci îi se va arăta o listă cu titlurile știrilor în ordine cronologică. Utilizatorul va putea să scrolleze prin meniu și să poată citi și titluri mai vechi, până la o anumită limită.

Dacă alege forumul atunci utilizatorul va putea să vadă topicurile de discuție, să comenteze pe baza acestora, dar să adauge și el topicuri spre dezbateri.

Dacă alege motorul de căutare atunci îi se va arăta o casuță de dialog unde va putea să caute un anumit utilizator după username. Dacă utilizatorul căutat nu figurează în baza de date, un mesaj corespunzător va fi afișat.

6. Modelul de date

Baza de date folosită de aplicația noastră respectă modelul relațional și are următoarea diagramă reprezentativă pentru a descrie modul în care noi ne structurăm datele.



7. Perspectiva fizică

Aplicația mobile funcționează pe platforma Android. Arhitectura este una de tip client – server peste rețea folosind protocolul de nivel aplicație HTTP. Serverul Web pe care îl folosim este IIS de la Microsoft deoarece folosim tehnologii .NET. Aplicația de front-end este scrisă în Java iar serverul, back-end este scris în C# (sharp). Baza de date pe care o folosim este un relațională, MySQL.

8. Perspectiva de dezvoltare

Dezvoltarea aplicației este împărțită în două părți principale, front-end-ul și back-end-ul. Deoarece suntem o echipă de patru, doi dintre noi se ocupă de front-end și doi de back-end. Am ales o dezvoltare incrementală a proiectului, plecând de la funcționalități de bază și crescând spre un produs mai complex. Integrarea o realizăm după fiecare funcționalitate adăugată astfel încât back-endul și front-endul să fie sincronizate și să nu existe o dezvoltare prea lungă de tip mock-up din partea ambelor părți.

Mergem pe o metodologie agile, cu sprinturi pe o săptămână în care vedem care este stadiul aplicației, ce amers bine, ce merge mai puțin bine, ce putem adăuga, și dacă suntem pe drumul corect spre o soluție. Modelul de date îl modificăm incremental la nevoie. Pentru fiecare funcționalitate a aplicației ce necesită baza de date modelul nostru relațional se poate schimba, astfel încât să poată răspunde nevoilor de ultimă oră.

Noi componente ale aplicației le putem adăuga în funcție de noile cazuri de utilizare pe care le constatăm de-

a lungul dezvoltării acesteia.

9. Perspectiva de testare

Testarea este o componentă importantă a proiectului nostru, aceasta a fost luată în calcul încă de la început, din fazele primitive ale aplicației noastre. Testele unitare sunt realizate de fiecare dintre noi în parte pe unitățile mici de program. La fiecare integrare săptămânală realizăm testele de integrare, pentru a vedea dacă componentele individuale se interconectează corespunzător. În cazul în care întâmpinăm probleme, prin faptul că aplicația pică unele dintre teste, este important să detectăm componenta sau componentele ce aduc aceste probleme și să încercăm să le remediem împreună, chiar dacă acestea au fost scrise de o singură persoană. De asemenea, încercăm să evităm problemele ce pot apărea prin code-review pe codul pe care un coleg îl încarcă pe repository-ul echipei.

La finalul proiectului, o să realizăm o testare de sistem. Aceasta o să fie făcută intern de către noi dar și extern de colegi de la facultate, pentru a vedea dacă aplicația îndeplinește standardele înalte pe care ni le-am propus.