



Universitatea POLITEHNICA București
Facultatea Automatică și Calculatoare
Departamentul Automatică și Informatică Industrială

LUCRARE DE LICENȚĂ

Concepte din jocuri aplicate în dezvoltarea software

Coordonator
prof.dr.ing. Liliana Dobrică

Absolvent
Butură Ștefan

2018

Cuprins

1. Introducere	3
2. Gamificarea	4
3. Cerințe funcționale	8
3.1. Definirea sarcinilor	9
3.2. Definirea recompenselor	12
3.3. Utilizatorii	13
4. Tehnologiile folosite	16
4.1. Docker	16
4.2. Git	18
4.3. Solr. MySQL	18
4.4. PHP	19
4.5. Nginx	19
4.6. Drupal. Drush	20
4.7. CSS. Twig	20
5. Implementare	22
5.1. Configurarea containerelor de Docker. Configurarea Nginx	23
5.2. Configurarea Drupal	26
5.3. Alegerea temei. Subteme	29
5.4. Taxonomia	31
5.5. Tipurile de conținut	33
5.6. Utilizatorii	34
5.7. Moduri de afișare	36
5.8. Sistemul de urmărire al utilizatorilor	37
5.9. Stările unei sarcini	38
5.10. Vizualizări	39
5.11. Crearea unui modul personalizat	41
6. Prezentarea aplicației finale. Dezvoltare ulterioară	44
7. Bibliografie	48
8. Anexe	50

1. Introducere

Domeniul dezvoltării de software a înregistrat creșteri masive în ultimii câțiva ani. În ultimele decenii, industria s-a focusat pe progres prin utilizarea de standarde precum CMMI (Capability Maturity Model Integration, ISO/IEC 2003) sau ISO 15504 (ISO/IEC 2003, ISO/IEC 2008). De asemenea, s-a pus mare accent pe incorporarea metodelor de dezvoltare agile (în special SCRUM). Dar, principalul aspect în dezvoltarea de software, spre deosebire de alte domenii, este factorul uman, cheia către succes fiind capacitatea omului de motivare și de concentrare pe termen lung. Managementul resursei umane este considerat unul dintre cele mai importante elemente. Cu toate acestea, natura obositoare și anostă a activităților din domeniul dezvoltării de software este un factor ce poate deteriora motivația și atenția tuturor participanților, fie ei manageri de proiect sau membri ai echipei.

În căutarea unei soluții pentru această problemă, vom considera proiectele software pot fi organizate ca un set de provocări a căror rezolvare necesită anumite abilități și efort colectiv. În acest proces sunt necesare aptitudini precum disciplina și agilitatea. Se poate vedea că exista o similaritate între dezvoltarea de software și jocuri: ambele constau în activități prin care jucătorul deprinde abilități noi și le folosește pentru a rezolva anumite provocări obținând premii sau pedepse în funcție de gradul de succes asociat activității [12].

Având în vedere acest lucru, putem afirma că aplicarea elementelor din jocuri în domeniul dezvoltării software, sau gamificarea (din termenul englez „gamification”) lui sună promițător. Obiectivul introducerii acestor elemente de joc este rezolvarea problemelor prezentate anterior prin crearea unui mediu de lucru mai prietenos pentru angajați, crescând motivația și gradul lor de implicare la locul de muncă.

În continuare, această lucrare își propune să prezinte gamificarea și să ofere detalii despre dezvoltarea unei platforme pentru aplicarea ei în domeniul dezvoltării de software.

2. Gamificarea

Termenul de „gamification” (gamificare) a fost folosit pentru prima data în anul 2002 de către dezvoltatorul britanic Nick Pelling pentru a descrie „accelerarea și atraktivizarea instrumentelor electronice utilizând interfețe grafice de tip joc”, conform [14]. Noțiunea a devenit populară abia în a doua parte a anului 2010. Sebastian Deterding, unul dintre pionierii și susținătorii conceptului de gamificare, afirma în 2011 că gamificarea reprezintă un termen „umbrelă” folosit pentru a descrie utilizarea elementelor prezente în jocuri video în afara mediului de joc pentru a spori experiența utilizatorilor și gradul lor de implicare, conform [9]. Transpunerea elementelor din jocuri în viața reală se face astfel încât diverse activități cotidiene să fie percepute de participanți într-un mod mult mai distractiv și antrenant, rezultatul final fiind îmbunătățit în anumite cazuri. Deterding a propus în [3] următoare definiție: „gamificarea este utilizarea elementelor din jocuri în contexte exterioare jocurilor”. Pentru a înțelege mai bine această definiție, Deterding a explicat cele 2 tipuri de joc, pentru care în limba engleză exista două cuvinte diferite. Cuvântul „play” se referă la o acțiune liberă, ludică, asociată în general copiilor. Opusul acestuia, „game”, este jocul cu reguli stricte, având anumite obiective și rezultate măsurabile.

Pentru a reprezenta mai bine elementele acestor definiții, Deterding a utilizat următoarea matrice:

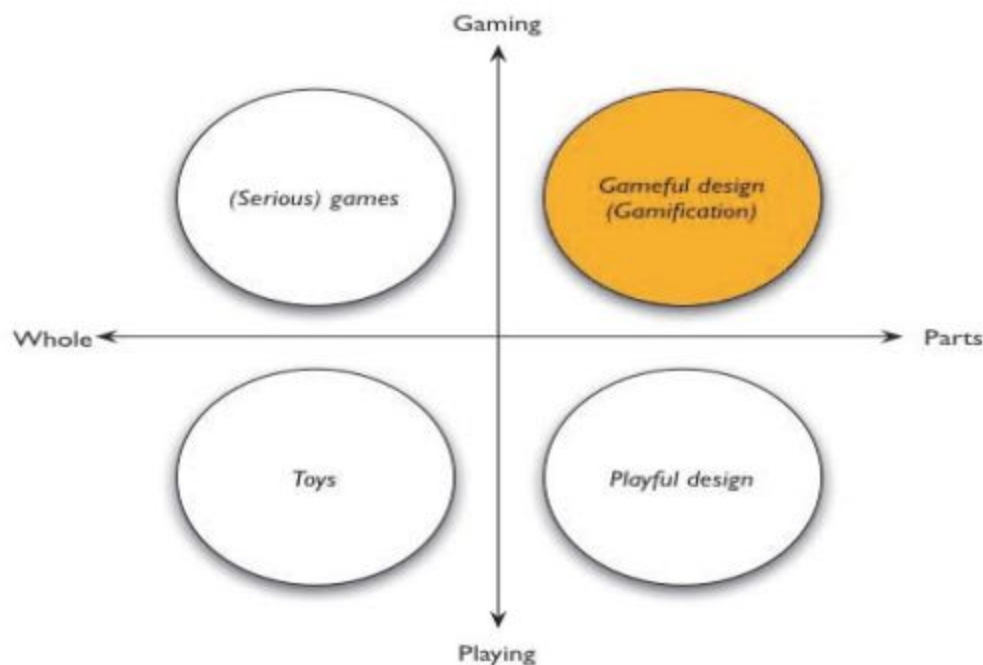


Fig. 1. Gamificarea și noțiunile aferente, conform [4]

Pe axa verticală sunt reprezentate cele 2 tipuri de joc explicate anterior iar pe axa orizontală este reprezentată tendința de completitudine a jocului, în limita din stânga având „whole”, jocul complet, iar în partea dreaptă, „parts”, reprezentând anumite elemente din joc. Se observă că gamificarea se folosește doar de câteva elemente din jocurile cu reguli, obiective și recompense.

Popularitatea gamificării a crescut semnificativ în ultimii ani, astfel că, cercetătorii din domeniul dezvoltării de software cunosc potențialele beneficii ale aplicării acestei metode la locul de muncă. Deși dezvoltarea de software are aspecte plăcute, majoritatea programatorilor văd anumite activități ca fiind plictisitoare sau anoste. Câteva exemple de astfel de activități includ scrierea de cod pentru părți ale sistemului care devin repetitive, precum scrierea de teste automate pentru funcționalități nu foarte complexe sau rezolvarea problemelor simple de mentenanță. Gamificarea permite companiilor să ofere recompense programatorilor pentru fiecare cerință îndeplinită de aceștia, pentru fiecare problemă rezolvată, fac mediul de lucru mult mai amuzant, crescând productivitatea. Elemente sociale precum misiuni colaborative sau clasamente în care angajații sau echipe de angajați să își compare progresul cu cel al colegilor lor

sunt câteva exemple ale aplicabilității gamificării. În acest mod, ei pot primi feedback permanent din care să rezulte importanța muncii lor pentru companie. Astfel, introducând elemente de joc în viața zilnică a unui programator poate ajuta organizațiile care se ocupa de dezvoltare de software să își motiveze mai bine angajații, rezultatul final al proiectelor

Principalele elemente de joc folosite în domeniu de dezvoltării software sunt combinațiile de provocare – recompensă, premiile fiind medalii, puncte, embleme și clasamentele. Câteva dintre aplicațiile care incorporează aceste metode încep deja să se extindă pentru a se potrivi domeniului dezvoltării software: *Gamify*, *RedCritic*, *PropsToYou*, *ScrumKnowsy*, *MasterBranch*, *CodeHunt*, extensia de joc pentru *Jenkins Continuous Integration* și altele.

Dar folosirea gamificării în domeniul ingineriei software este rară, iar lucrările de cercetare în domeniu sunt la început. De asemenea, incorporarea gamificării se face de obicei peste aplicații independente, care nu au fost inițial dezvoltate cu acest obiectiv.

Gamificarea este un domeniu de cercetare vast care a devenit atractiv în anumite domenii precum marketing-ul online, aplicațiile pentru mobil sau cel educațional [2]. Una dintre cele mai importante lucrări de cercetare se bazează pe evaluarea utilității gamificării prin analizarea mai multor situații în care a fost aplicată [11]. Concluzia articolului a fost: „gamificarea funcționează, dar există anumite îngrijorări”; majoritatea rezultatelor gamificării au fost pozitive, existând însă și anumiți factori de confuzie asupra rezultatului.

Folosirea acestor principii în dezvoltarea de aplicații web a fost analizată în [18], concluzia fiind că gamificarea s-a bazat pe folosirea superficială a elementelor de joc și că ar trebui folosite și aspecte mai avansate precum interacțiunea socială, suportul pentru aplicații mobile și date statistice mai complexe despre participanți.

În ultimii ani, lucrările de cercetare s-au axat pe introducerea gamificării în domeniul ingineriei software. În [10] au fost analizate 29 de studii pe această temă, luându-se în considerare procesele implicate din dezvoltare, elementele de joc aplicate, metodologia aplicată pentru introducerea lor și dovezile empirice despre impactul lor asupra rezultatelor organizației. Cel mai vechi studiu descoperit datează din anul 2010. Principalele concluzii ale acestei analize au sunt:

- implementarea de software este procesul care a atras cel mai mult interes pentru gamificare, urmat de testare și de managementul cerințelor. Alt proces important inclus în aceste studii de caz este managementul proiectelor.

- elemente de joc aplicate sunt sistemele bazate pe recompense în puncte și embleme, clasamente și elemente de reputație socială

- jumătate dintre studii erau doar propuneri pentru astfel de sisteme, neavând o aplicație reală la bază care să justifice alegerile făcute

Concluzia analizei făcute în [10] este că gamificarea în domeniul dezvoltării software este într-un stadiu foarte puțin dezvoltat. S-a observat că adoptarea unei astfel de strategii s-a făcut mult mai rapid în alte domenii precum marketing, educație, sănătate și banking. De asemenea, au fost identificate următoarele probleme în aceste studii:

- lipsa unei metodologii pentru aplicarea gamificării, ceea ce afectează replicabilitatea propunerii în alte companii. Autorii propunerilor descriu câțiva pași simpli pentru acest lucru dar adeseori sunt incompleți.

- dificultatea integrării unui astfel de sistem în medii reale de lucru.

3. Cerințe funcționale

În continuare, lucrarea își propune dezvoltarea și prezentarea unei aplicații web menită organizațiilor din domeniul dezvoltării software, care să implementeze gamificarea. Aplicația va permite împărțirea cerințelor unui proiect în sarcini. O sarcină va avea următoarele caracteristici: un responsabil pentru completarea ei, stare, prioritate, categorie, durată estimată, descrierea activității, recompensă și termenul limită. Platforma va conține statistici relevante per sarcină sau per utilizator (ex: totalul de ore lucrate).

Introducerea conceptelor din jocuri are scopul de a crește motivația și performanțele utilizatorilor aplicației. Câteva dintre aceste concepte sunt:

- un sistem de punctare pentru completarea unor sarcini. Responsabilul pentru acea sarcină va primi puncte de experiență proporțional cu dificultatea ei
- un nivel asociat fiecărui cont. Creșterea nivelului se va face prin acumularea punctelor de experiență
- premii pentru terminarea unei sarcini, constând în embleme, monede virtuale, titluri, iconițe sau texte care să marcheze realizările utilizatorilor.
- pagini de profil personalizabile în care utilizatorii își pot afișa premiile câștigate
- o pagina de tip magazin în care angajații își pot folosi monedele virtuale câștigate pentru a primi anumite beneficii
- un sistem de urmărire al altor colegi
- clasamente globale sau între colegi
- misiuni de tipul – terminarea unei sarcini într-un timp mai scurt decât cel estimat.

În funcție de poziția lor în companie, utilizatorii vor avea un anumit rol în cadrul aplicației. Pentru buna funcționare a aplicației, am identificat 4 roluri ca fiind necesare. Administratorul de sistem poate configura toate setările aplicației și este de regula o persoană instruită pentru acest lucru. Managerul de proiect are rolul de a crea sarcini, de a le atribui altor angajați și de a evalua dacă acestea au fost completate cu succes. Artistul are rolul de a crea premii care pot fi asociate sarcinilor. Programatorul are rolul de a completa sarcinile atribuite.

Pentru a putea înțelege mai bine modul de utilizare al aplicației și ce poate oferi aceasta unei companii, ea va fi împărțită în componente. În continuarea acestui capitol, fiecare componentă va fi analizată și vor fi expuse toate cerințele funcționale.

3.1. Definirea sarcinilor

Cea mai importanta parte a soluției propuse este sistemul de sarcini. În contextul acestei aplicații, o sarcină reprezintă o cerință atomică din ciclul real de dezvoltare de software al unei companii. O astfel de sarcină are următoarele atribute:

- număr de identificare, generat automat de către aplicație în ordine crescătoare, începând de la 1. Acesta va fi folosit pentru ușurința accesării paginii unei sarcini. De exemplu, sarcina cu numărul 10 va fi disponibilă la adresa „/task/10”.
- titlu
- descrierea sarcinii. Acest câmp va putea conține text, imagini, legături către alte pagini și elemente de formatare.
- prioritate, care poate lua una dintre următoarele valori: scăzută, normală și ridicată.
- responsabilul cu rezolvarea sarcinii. Acest câmp va putea fi setat de către creatorul sarcinii cu o valoare din lista de utilizatori ai aplicației.
- categoria din care face parte sarcina; poate lua una dintre valorile: sarcină obișnuită, bug sau îmbunătățire.
- data termen pentru terminarea sarcinii

- timpul estimat necesar terminării sarcinii, în ore
- statusul sarcinii, care poate lua una dintre următoarele valori: nouă, în desfășurare, acceptanță, acceptată, încheiată/închisă. Diagrama pentru tranzițiile între aceste stări va fi prezentată ulterior.

- timpul total petrecut pentru sarcină, în ore
- puncte de experiență, un premiu primit pentru terminarea unei sarcini
- „aur”, un întreg reprezentând cantitatea de monedă virtuală primită de angajatul care duce sarcina la bun sfârșit
- recompensa, un câmp folosit pentru a aduna toate celelalte tipuri de premii date responsabilului acelei sarcini. Acesta poate fi de tipul iconiță, titlu primar sau secundar, insignă sau un text care să marcheze îndeplinirea sarcinii. Aceste tipuri de recompense vor fi descrise ulterior.

De asemenea, pe pagina unei sarcini va fi implementat un sistem de comentarii, un comentariu fiind descris de utilizatorul care l-a creat, data la care acesta a fost scris și conținutul lui.

Sarcinile vor putea fi create doar de către managerii de proiect, care vor completa toate câmpurile descrise anterior, care vor completa toate câmpurile descrise anterior, mai puțin numărul de identificare care este generat automat și de timpul total petrecut pe sarcină, care va fi incrementat pe măsură ce angajatul responsabil face un progres în completarea ei.

Programatorii au dreptul de a prelua singuri sarcini care nu au deja un responsabil desemnat. Aceștia își vor putea adăuga pe pagina sarcinii timpul petrecut pentru ea și vor putea schimba statusul ei. Stările pe care aceștia le pot alege pentru o sarcină sunt însă limitate. Managerii de proiect au rolul de a evalua dacă o sarcină a fost completată corect sau dacă nu mai este nevoie ca aceasta să fie terminată. Doar ei pot trece o sarcină în starea „acceptată” sau „încheiată/închisă”.

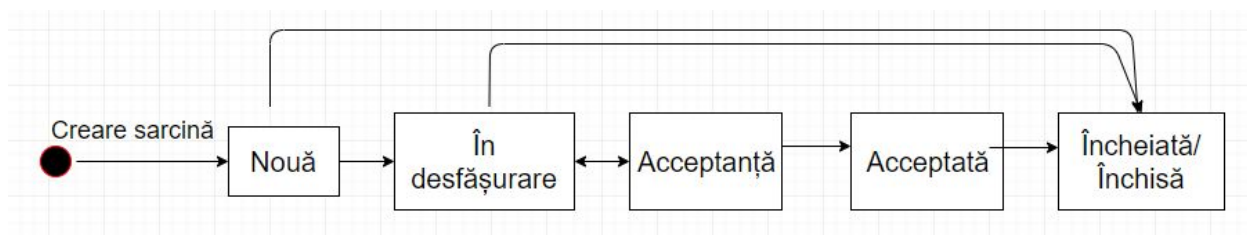


Fig. 3.1.1. Diagrama de stări a unei sarcini modificată de un manager de proiecte

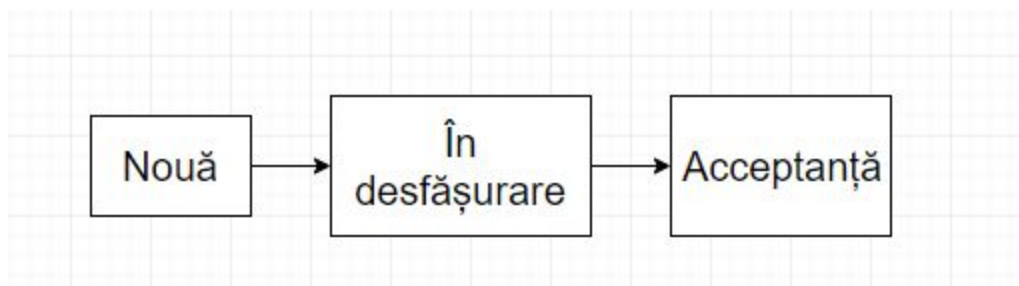


Fig. 3.1.1. Diagrama de stări a unei sarcini modificată de programator

Pagina principală a aplicației va conține lista sarcini, în ordinea descrescătoare a datei publicării lor. În această listă vor putea fi vizibile numărul de identificare, categoria, prioritatea, statusul, titlul și responsabilul pentru fiecare sarcină. Accesarea unei sarcini se va face prin apăsarea unui clic pe titlu. Pe pagina unei sarcinii vor putea fi vizualizate toate detaliile ei și i se vor putea modifica atributele. De asemenea, vor putea fi vizibile comentariile aferente ei și se vor putea adăuga comentarii noi. Pagina principală a site-ului va mai conține un formular pentru căutarea în sarcini și fațete pentru a putea filtra lista disponibilă.

3.2. Definirea recompenselor

Pentru ducerea la bun sfârșit a unei sarcini, un angajat poate primi unul sau mai multe dintre următoarele premii: puncte de experiență, aur, iconițe, titluri, embleme, sau realizări.

Punctele de experiență reprezintă principala metodă de progres în competiția cu alți angajați. Adunând un anumit număr de puncte, angajații pot trece la nivelul următor. Acest lucru va putea fi vizibil în clasamentele angajaților, în care ei vor fi ordinați după acest criteriu, crescând competitivitatea și motivația lor. Pentru a trece de la nivelul 1 la nivelul 2, sunt necesare 100 de puncte de experiență. Pentru fiecare nivel subsecvent va fi nevoie de 50 de puncte în plus față de nivelul anterior. De exemplu, pentru a trece la nivelul 6, va fi nevoie de 1000 de puncte.

Iconițele reprezintă imagini de mici dimensiuni. Utilizatorii vor putea alege din colecția proprie de iconițe pentru a afișa una pe pagina lor de profil.

Titlurile sunt texte care pot fi afișate pe pagina de profil și pot fi de două feluri: titluri primare și secundare. Combinația a 2 titluri diferite creează un titlu complet. De exemplu, titlul primar „șoimul” poate fi alăturat titlului secundar „agil” pentru a genera un titlu complet demn de o pagină de profil.

Emblemele sunt imagini simple. Angajații vor putea afișa maxim 3 embleme pe pagina personală de profil.

Realizările sunt texte decorative care marchează îndeplinirea unei sarcini complexe. De exemplu, pentru terminarea unei sarcini în jumătatea timpului propus pentru aceasta, un angajat ar putea primi realizarea „Acest utilizator termină sarcini cu o viteză fantastică”. Pe pagina de profil vor putea fi afișate trei astfel de realizări.

Aurul reprezintă moneda virtuală folosită în cadrul aplicației pentru a cumpăra recompense care nu sunt accesibile altfel.

3.3. Utilizatorii

Aplicația va putea fi accesată pe baza unor conturi create de către administratorii ei sau de către managerii de proiect. Fiecare utilizator își va putea accesa contul folosind un nume de utilizator și o parolă. Aceștia au o pagină de profil personalizabilă pe care își vor putea afișa premiile primite. De asemenea, pe paginile lor vor fi afișate statistici despre numărul de sarcini îndeplinite,, timpul total petrecut pentru îndeplinirea lor și nivelul. Astfel, conturile vor fi definite de următoarele caracteristici:

- realizări
- puncte de experiență
- aur
- iconiță
- nivel
- premiile de pe pagina de profil
- totalul de recompense
- rolul

Aplicația va avea 4 roluri cu permisiuni diferite, vizibile în figura 3.3.1.

Permisii \ Rol	Administrator aplicație	Manager de proiect	Artist	Programator
Schimbare setări aplicație	✓	X	X	X
Creare sarcini	✓	✓	X	X
Trecere sarcină în starea de „progres”	✓	✓	✓	✓

Trecere sarcină în starea de „acceptanță”	✓	✓	✓	✓
Trecere sarcină în starea „acceptată” și „închisă”	✓	✓	X	X
Modificare sarcini	✓	✓	X	X
Adăugare comentarii	✓	✓	✓	✓
Schimbare pagină de profil personală	✓	✓	✓	✓
Adăugare/modificare/ștergere conturi	✓	✓	X	X
Creare recompense	✓	X	✓	
Vizualizare sarcini, pagini de profil, clasamente	✓	✓	✓	✓

Fig.3.3.1. Rolurile și permisiunile lor

Dacă o acțiune nu apare în tabel, se va considera că fiecare utilizator are dreptul de a o desfășura.

În plus, pe fiecare pagina de profil va apărea opțiunea de „urmărire utilizator”, disponibilă. Va exista un clasament personalizat pentru fiecare persoană, în care își poate vedea progresul relativ la partenerii pe care îi urmărește.



Fig.3.3.2. Exemplu de pagină de profil,
<https://steamcommunity.com/profiles/76561198028874223/> (accesat 2018)

4. Tehnologiile folosite

Pentru implementarea acestei aplicații au fost utilizate următoarele tehnologii: Nginx, Docker, Git, Composer, MySQL PHP/Drupal, Drush, Twig, CSS, Solr. Acestea vor fi prezentate în continuare, subliniind rolul lor în cadrul proiectului.

4.1. Docker

Docker este un program gratuit, open-source, dezvoltat de către Docker, Inc. și folosit pentru virtualizarea nivelului „sistem de operare” sau împărțirea lui în containere. Aceste containere sunt folosite pentru a rula aplicații independent de sistemul de operare deja existent și sunt văzute ca fiind în sinea lor un sistem de operare de către aplicațiile pe care le găzduiesc. În mod implicit, programele care rulează în aceste containere au acces doar la resursele containerului și nu pot comunica cu exteriorul.

Docker devine din ce în ce mai popular printre dezvoltatorii de aplicații. În martie 2018, peste 3,5 milioane de aplicații au fost plasate în containere și peste 37 de miliarde de astfel de aplicații au fost descărcate.

Virtualizarea permite portabilitatea aplicației pe orice server indiferent de sistemul de operare sau resursele deja instalate pe acesta, singura condiție fiind ca el să poată rula Docker. Spre deosebire de mașinile virtuale, care emulează resurse de sistem, containerele, partajează resursele sistemului de operare, fiind mult mai eficient.

Un alt motiv pentru utilizarea acestei platforme este ușurința dezvoltării ulterioare. Toate cerințele de sistem necesare pentru a începe dezvoltarea unei aplicații sunt înlocuite cu descărcarea containerelor și rularea lor, proces care durează câteva minute.

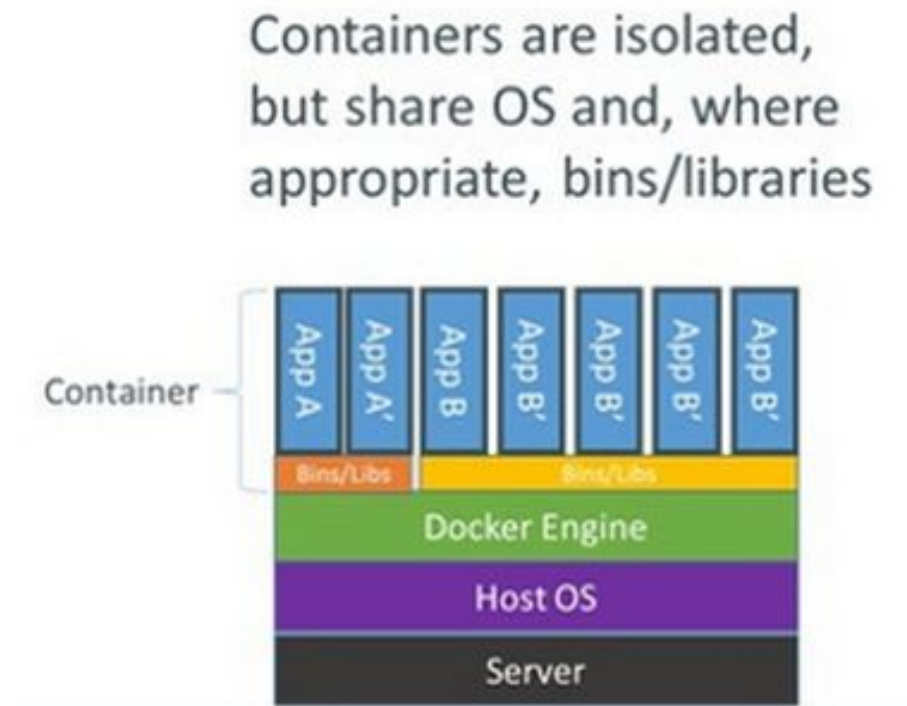


Fig. 3.1. Structura containerelor într-un sistem de operare [17]

4.2. Git

Git este un sistem open-source folosit pentru controlul reviziilor folosit în special pentru proiectele de dezvoltare de software de dimensiuni mari sau la care participa echipe numeroase.

În cadrul acestui proiect, versionarea codului este foarte importantă întrucât complexitatea lui este ridicată. Pentru siguranță, fiecare îmbunătățire adusă lui va duce la crearea unei versiuni noi pe Git, având posibilitatea de a mă întoarce oricând la o versiune anterioară în cazul în care apar probleme.

4.3. Solr. MySQL

Solr este platformă open-source, scrisă în Java, folosită pentru a interfața căutarea într-o bază de date. Principalele caracteristici includ: căutarea full-text, evidențierea cuvintelor căutate în rezultate, fațetare (facets), integrări cu baze de date obișnuite și tratarea documentelor de tip Word și PDF.

Proiectul își propune să aibă un sistem de căutare prin sarcinile disponibile pentru angajați având la dispoziție anumite filtre.

MySQL este un sistem open-source pentru gestionarea bazelor de date relaționale deținut de Oracle. Este folosit de multe companii mari, printre care și Google, însă nu pentru căutări, fiind destul de lent pentru acest tip de operații.

Este nevoie de un sistem de baze de date pentru a stoca datele generate de Drupal, conturile, conținutul site-ului și setările.

4.4. PHP

PHP este un limbaj de scripting dezvoltat de către Rasmus Lerdorf în anul 1995. La momentul actual este unul dintre cele mai populare astfel de limbaje. Este folosit, în special, pentru aplicațiile web cu conținut dinamic. Este ușor de folosit și foarte rapid, astfel că, peste 50% dintre aplicațiile web se bazează pe PHP. În mod normal, compilarea unui cod îngreunează destul de mult executarea lui, dar PHP minimizează această problemă, folosind cache-uri pentru a memora codul deja compilat anterior. Unele dintre site-uri importante care folosesc acest limbaj sunt: Facebook, Yahoo, Oracle.

Versiunea 4.0 a PHP a adus o îmbunătățire majoră – aceasta permite utilizarea programării orientată pe obiect. În prezent, PHP a ajuns la versiunea 7.3.

4.5. Nginx

Nginx, pronunțat „engine X”, este un server HTTP care necesită puține resurse, dezvoltat în Rusia. Acesta oferă performanțe ridicate și poate servi funcționalitatea de proxy pentru un server de email (IMAP/POP3/SMTP). Nginx este faimos pentru stabilitatea sa, configurația flexibilă și pentru faptul că resursele folosite sunt minime. Acesta poate rula Unix, Linux și Windows [8].

Comparativ cu Apache, un alt server popular pentru HTTP, Nginx consumă mai puține resurse și poate servi mai mulți clienți simultan. Acesta poate suporta până la 50000 de conexiuni concurente.

4.6. Drupal. Drush

Drupal este un sistem open-source de gestionarea conținutului (Content Management System – CMS). Acesta ușurează semnificativ munca dezvoltatorilor de aplicații web, permițându-le acestora să se axeze pe funcționalități noi, fără a mai fi nevoie să implementeze pe elemente de bază precum gestionarea bazelor de date sau a utilizatorilor.

Drupal este scris în PHP și se folosește de JavaScript pentru anumite funcționalități de front-end. Pentru ca acesta să funcționeze, trebuie folosit în combinație cu un sistem de baze de date, fiind acceptate MySQL, MariaDB sau PostgreSQL. Baza de date stochează atât conținutul site-ului cât și configurările (sistemul de configurări oferit de Drupal va fi prezentat în detaliu la capitolul legat de implementare). Datorită stratului de abstractizare oferit de către Drupal, indiferent de tipul de bază de date folosită, comenzile folosite pentru manipularea lor rămân aceleași, iar siguranța datelor este sporită, fiind eliminate atacurile de tip injecție SQL.

Drush este o interfață, în linia de comandă, ce ajută dezvoltatorii de Drupal să execute acțiuni asupra site-ului lor într-un singur loc.

4.7. CSS. Twig

CSS (Cascading Style Sheets) este un limbaj folosit pentru a descrie modul de prezentare al documentelor scrise în HTML sau XML. CSS indică modul în care elementele vor apărea pe ecranele utilizatorilor.

Twig este un template engine pentru PHP. Acesta este descris de autorii săi ca fiind rapid, sigur și flexibil [7]. Este foarte concis iar comenzile sunt simple. De exemplu, dacă se dorește

afișarea numelor unor utilizatori, care sunt transmiși prin variabila de tip array „users”, sintaxa este următoarea:

```
{% for user in users %}  
    {{ user.name }}  
{% else %}  
    Niciun utilizator nu a fost gasit  
{% endfor %}
```

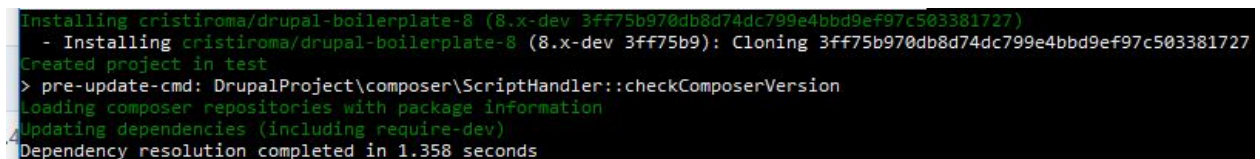
5. Implementare

Primul pas pentru implementarea acestei aplicații este descărcarea și configurarea unei distribuții de Drupal. Voi porni de la varianta de Drupal propusă de Cristian Romanescu, disponibilă la <https://github.com/cristiroma/drupal-boilerplate-8> (accesat 2018). După cum sugerează documentația acestei distribuții, pentru a începe un proiect nou folosind această bază este necesară utilizarea aplicației Composer.

Comanda folosită pentru a începe proiectul este:

```
composer -vv create-project cristiroma/drupal-boilerplate-8 licenta  
--stability dev --no-interaction
```

După executarea acestei instrucțiuni în Windows Powershell, un nou proiect a fost creat pe disc, conținând distribuția de Drupal 8 dorită.



```
Installing cristiroma/drupal-boilerplate-8 (8.x-dev 3ff75b970db8d74dc799e4bbd9ef97c503381727)  
- Installing cristiroma/drupal-boilerplate-8 (8.x-dev 3ff75b9): Cloning 3ff75b970db8d74dc799e4bbd9ef97c503381727  
Created project in test  
> pre-update-cmd: DrupalProject\composer\ScriptHandler::checkComposerVersion  
Loading composer repositories with package information  
Updating dependencies (including require-dev)  
Dependency resolution completed in 1.358 seconds
```

Fig 5.1. Instalarea proiectului de Drupal

5.1. Configurarea containerelor de Docker.

Configurarea Nginx

În mod obișnuit, pentru a putea găzdui această aplicație web, este necesară configurarea unui server și instalarea dependențelor, un proces de lungă durată. De aceea, vom lucra cu într-un mediu virtualizat, cu containere de Docker care au deja instalate librăriile necesare. În plus, acest lucru contribuie la ușurința dezvoltării ulterioare a aplicației.

Pentru proiectul prezentat în cadrul acestei lucrări este nevoie de următoarele servicii: Nginx, PHP 7.1, MySQL și Solr 6. Fiecare serviciu este plasat într-un container separat, cu caracteristici descrise în fișierul *docker-compose.yml*. Acesta are o structură de tipul cheie – valoare. În continuare vor fi explicate cheile din fișier și necesitatea lor:

- `container_name` – numele containerului
- `ports` – legătura între porturile sistemului de operare și porturile containerului
- `depends_on` – serviciul de care depinde un container pentru a rula în condiții normale
- `volumes` – partajarea resurselor de pe disc cu containerul

Pentru a putea comunica cu exteriorul, este nevoie să mapăm portul 80 al containerului de Nginx la portul 80 al sistemului de operare. Astfel, accesând din browser o adresă web, care în mod implicit este pe portul 80, se va putea accesa site-ul aflat în container. De asemenea, se vor partaja anumite fișiere de configurare și proiectul în sine.

În continuare, este prezentat fișierul *docker-compose.yml*:

```
nginx:
  container_name: licenta_proxy
  ports:
    - "8080:8080"
  image: nginx:latest
```

```
depends_on:
  - php71
volumes:
  - .docker/conf-nginx/globals.conf:/etc/nginx/conf.d/globals.conf:ro
  - .docker/conf-nginx/project.conf:/etc/nginx/conf.d/project.conf:ro
  - ../var/www/html/licenta:z
env_file:
  - docker.env
```

php71:

```
container_name: licenta_php
image: cristiroma/php-fpm
depends_on:
  - db
volumes:
  - ../var/www/html/licenta
```

db:

```
container_name: licenta_db
image: mariadb
env_file:
  - docker.env
volumes:
  - .docker/conf-mysqld/server.cnf:/etc/mysql/conf.d/server.cnf
```

solr6:

```
image: library/solr:6
container_name: licenta_solr
volumes:
  - .docker/conf-solr/core-template:/core-template
entrypoint:
  - docker-entrypoint.sh
  - solr-precreate
  - licenta
  - /core-template
```


Fișierul menționat *docker.env* conține o serie de setări particulare pentru fiecare dezvoltator sau mașină. Este important ca acest fișier să nu fie făcut public întrucât el poate conține informații secrete precum datele de acces la o bază de date MySQL. În continuare, este prezentată o parte din conținutul acestui fișier:

```
# Global
TZ=Europe/Zurich

# MySQL
MYSQL_ROOT_PASSWORD=root
MYSQL_DATABASE=licenta
```

5.1.1. Configurarea Nginx

Ca în cazul oricărei aplicații web, este necesar un server HTTP care să servească pagini vizitatorilor. Pentru acest lucru, în containerul de Nginx va fi creat un virtual host, disponibil pentru început la adresa <http://localhost:8080>. Aplicațiile de Drupal au nevoie de o serie de setări de configurare pentru Nginx pentru a funcționa corect, ele fiind disponibile la [15]. Câteva dintre aceste setări vor fi prezentate în continuarea acestei lucrări:

```
server {
    listen      8080;
    server_name localhost;
    root /var/www/html/licenta/docroot;

    location ~ ^/sites/.*/files/ {
        sendfile on;
```

```

    gzip on;
    try_files $uri $uri/ /index.php?$query_string;
}
location @rewrite {
    rewrite ^/(.*)$ /index.php;
}
}

```

Cheile „listen” și „server_name” descriu modul în care va fi accesată aplicația, adică la adresa `http://localhost:8080`. Cheia „root” este folosită pentru a seta calea către aplicația web de pe disc.

Arhitectura de Drupal este construită în așa fel încât fișierul „index.php” să servească fiecare cerere. Rutinele din „index.php” predau controlul către serviciul potrivit care mai apoi afișează pagina dorită. Din acest motiv, accesarea oricărei pagini din cadrul site-ului va trebui redirectată către index. În caz contrar, Drupal nu va mai funcționa corect sau poate chiar avea o mare breșă de securitate.

În cazul fișierelor statice, precum imagini sau fișiere, această redirectare către index nu mai este necesară, ci este servit fișierul cerut.

În plus, Nginx mai are nevoie de câteva setări globale precum timpul maxim acordat unui client până când este trimisă o eroare de timeout sau cantitatea maximă de date care poate fi trimisă către un client pentru o singură cerere. Aceste setări pot fi accesate în fișierul „„docker/conf-nginx/globals.conf”.

5.2. Configurarea Drupal

După ce am terminat cu setările legate de mediul în care va rula aplicația, este nevoie de configurarea pilonului care stă la baza ei, și anume Drupalul. Acest CMS își ține setările de bază

în fișierul „docroot/sites/default/settings.php”. Primul pas este copierea fișierului *default.settings.php* și redenumirea în *settings.php* pentru a avea configurația de bază. Buna practică indică ca fișierul *settings.php* să nu fie modificat decât prin adăugarea următoarelor linii:

```
if (file_exists($app_root . '/' . $site_path .
'/settings.local.php')) {
    include $app_root . '/' . $site_path . '/settings.local.php';
}
```

Când *settings.php* este încărcat, liniile de cod menționate anterior comandă încărcarea unui alt fișier, *settings.local.php*, în care se vor ține setări personale. Cea mai importantă modificare adusă la configurația de bază este cea care ține legătura cu baza de date:

```
$databases['default']['default'] = array(
    'database' => 'licenta',
    'username' => 'root',
    'password' => 'root',
    'prefix' => '',
    'host' => 'db',
    'port' => '3306',
    'namespace' => 'Drupal\\Core\\Database\\Driver\\mysql',
    'driver' => 'mysql',
);
```

Fără aceasta modificare, Drupal nu poate funcționa, întrucât este necesară conexiunea la o bază de date. Pe lângă parametrii obișnuiți pentru o astfel de legătura, precum numele bazei de date, numele de utilizator și parola, dar și portul la care poate fi accesată, este important de remarcat parametrul *host*. În mod obișnuit, baza de date rulează pe serverul local, iar acest parametru este egal cu „localhost”. În cazul unei aplicații care folosește Docker, *host*-ul bazei de date este chiar containerul pe care ea se află, iar în setări va fi folosit numele lui, în acest caz el fiind „db”.

Din acest moment, Drupal poate să ruleze. În continuare, se va trece prin procesul obișnuit de instalare al Drupal 8. Pentru acest lucru, se va accesa pagina „http://localhost:8080/core/install.php”.

În urma procesului de instalare, o versiune curată de Drupal va fi disponibilă. În continuarea aceste lucrări va fi prezentat procesul de configurare al Drupal și pașii prin care s-a trecut pentru a ajunge la aplicația dorită.



Fig.5.2.1. Instalarea Drupal

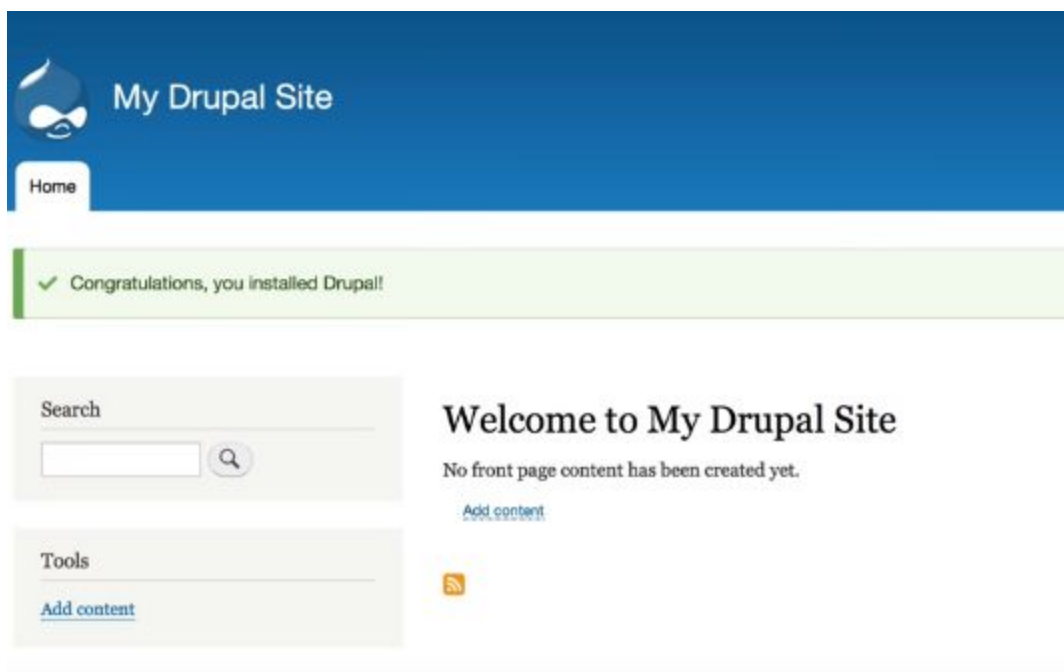


Fig.5.2.2. Pagina principală a unui site nou creat cu ajutorul Drupal

5.3. Alegerea temei. Subteme

În Drupal, toate componentele de design precum structura blocurilor în pagini, culorile, fonturile sau modul în care sunt afișate diverse tipuri de conținut intră sub denumirea de temă. Temele de Drupal sunt gratuite și disponibile pe site-ul oficial: https://www.drupal.org/project/project_theme (accesat 2018).

Pentru acest proiect a fost aleasă tema „Bulma CSS” (<https://www.drupal.org/project/bulma>, accesat 2018) bazată pe Bulma, un framework open-source de CSS care se bazează pe flexbox și care este folosit de peste 100000 de dezvoltatori din toată lumea [6]. Este compatibilă cu majoritatea instrumentelor de navigare web, oferind flexibilitate aplicației. Am ales Bulma CSS datorită culorilor vii, aspectului modern și datorită structurii blocurilor.

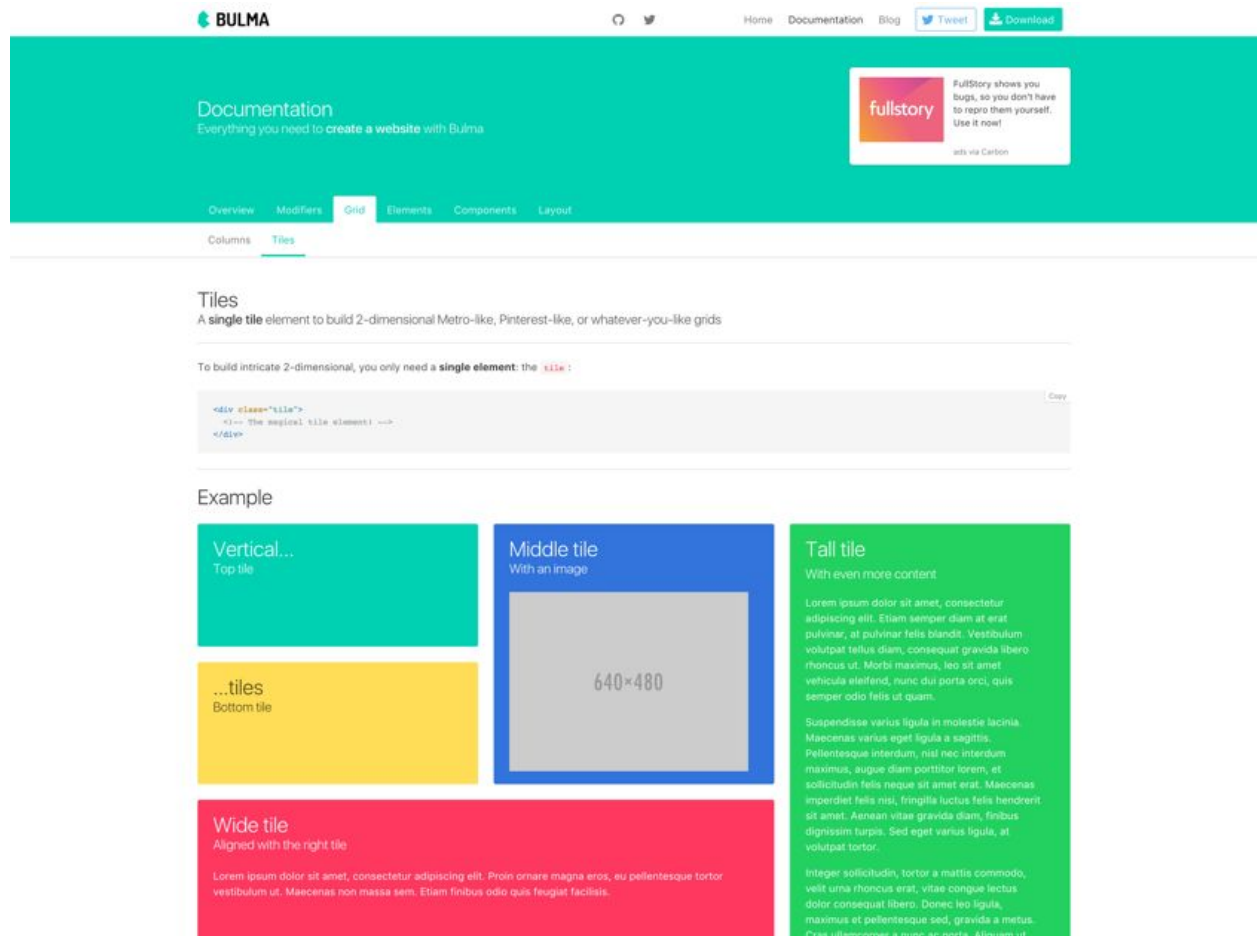


Fig 5.3.1. Tema Bulma CSS

Pentru instalarea unei teme este nevoie să fie rulate în containerul de PHP câteva comenzi. Accesarea terminalului unui container se face cu comanda „docker exec –ti nume_container bash”. Pentru acest caz, nume_container se va înlocui cu *licenta_php*.

Comenzile care se vor rula în directorul *docroot* din interiorul containerului de PHP sunt:

```
drush dl -y bulma
drush en -y bulma.
```

Prima comanda este folosită pentru a descărca tema iar a doua pentru a o activa. Setarea ei ca temă principală se poate face în două feluri: Primul mod este navigarea pe pagina */admin/appearance* și selectarea ei din meniul de teme disponibile prin apăsarea butonului „Set

as default”. Acest lucru se mai poate face și din consolă rulând comanda „drush config-set system.theme default bulma”.

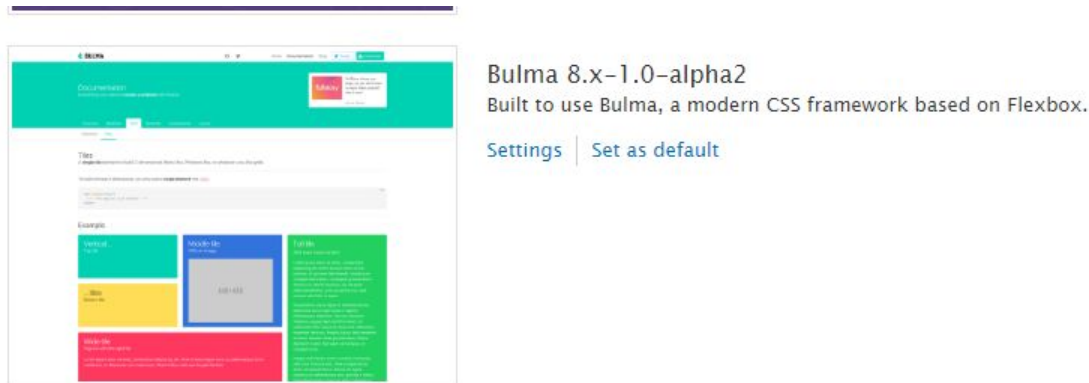


Fig.5.3.1. Selectarea temei

O subtemă se comportă ca o temă obișnuită, având o singură particularitate; acestea moștenesc resursele temei părinte. Nu există limite în ceea ce privește abilitatea de a crea subteme, astfel că o subtemă poate avea părinte o altă subtemă.

Pentru acest proiect voi crea o subtemă care să extindă tema Bulma, pentru a duce câteva modificări temei de bază. Pentru a extinde tema Bulma, dezvoltatorii au pus la dispoziție comanda „drush bulma nume_tema”.

5.4. Taxonomia

Taxonomia este o disciplină științifică care se ocupă cu stabilirea unor legi de clasificare și sistematizare a unor domenii reale, caracterizate printr-o structură complexă . În contextul Drupal, taxonomia („taxonomy”) este un modul de bază care permite crearea unor vocabulare de

termeni cu scopul de a împărți conținutul unui site în categorii sau pentru a face legătura între anumite concepte.

În cadrul acestui proiect, este nevoie de următoarele trei vocabulare:

- statusuri sarcini
- categorii sarcini
- priorități sarcini

Vocabularele pot fi create sau modificate pe pagina „/admin/structure/taxonomy”. În mod normal, fiecare termen este definit de un titlu și o descriere, modulul oferind posibilitatea de a adăuga și alte câmpuri.

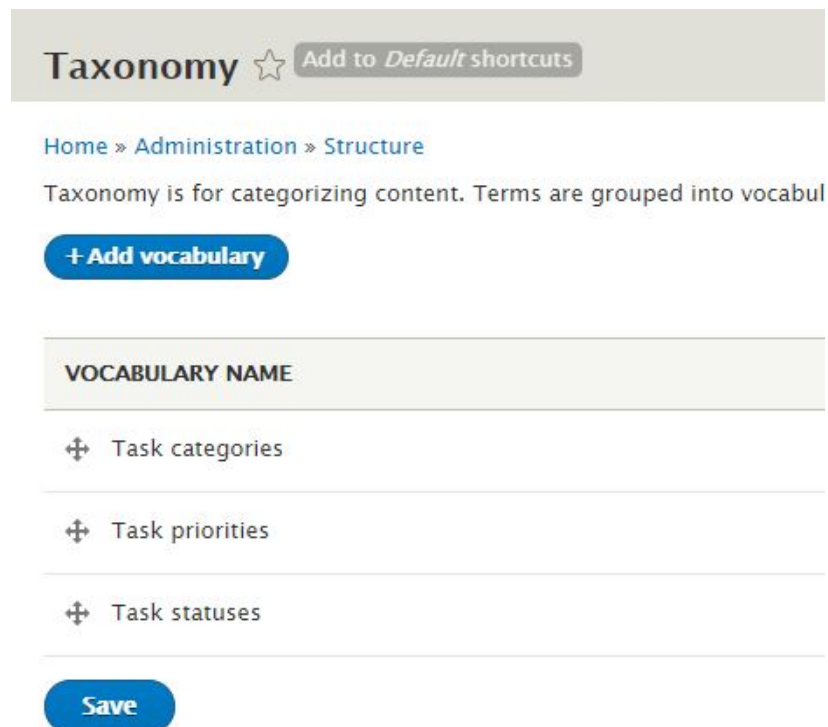


Fig.5.4.1.Vocabulare în Drupal 8

NAME
+ Acceptance
+ Approved
+ Closed
+ New
+ On hold
+ Under work

Save Reset to alphabetical

Fig 5.4.2. Termenii din vocabularul „statusuri sarcini” („Task statuses”)

5.5. Tipurile de conținut

În Drupal, conținutul unui site este împărțit în tipuri. Fiecare tip de conținut are caracteristici proprii, marcate prin câmpuri. Câmpurile („fields”) pot fi de diverse tipuri în funcție de valorile pe care le acceptă. Câteva dintre tipurile de câmpuri acceptate sunt: număr, text, dată sau referință către o altă entitate.

Pentru implementarea acestei aplicații, tipurile de conținut folosite vor fi:

- realizare
- emblemă
- iconiță
- notificare

- titlu primar
- titlu secundar
- timp petrecut
- sarcină

Detalierea acestor tipuri de conținut și a câmpurilor lor este disponibilă în anexa 1.

Home » Administration » Structure » Content types » Task

[+ Add field](#)

LABEL	MACHINE NAME	FIELD TYPE
Assignee	field_assignee	Entity reference
Body	body	Text (formatted, long, with summary)
Category	field_category	Entity reference
Comments	comment	Comments
Due date	field_due_date	Date
Estimated time	field_estimated_time	Number (decimal)
Experience points	field_experience_points	Number (integer)
Gold	field_gold	Number (integer)
Priority	field_priority	Entity reference
Rewards	field_rewards	Entity reference
Spent time	field_spent_time	Number (decimal)
Status	field_workflow_status	Workflow state
Task number	field_task_number	Number (integer)

Fig.5.5.1. Câmpurile unei sarcini

5.6. Utilizatorii

Utilizatorii sunt o entitate de bază în Drupal și nu necesită configurări suplimentare pentru a putea folosi un sistem de conturi. Aceste entități implementează câteva câmpuri de bază:

nume utilizator, parolă, email și imagine, cu posibilitatea de a fi extinse. Totuși, pentru a putea respecta cerințele expuse în capitolul 3, vor fi adăugate câteva câmpuri noi: iconiță, experiență, nivel, recompense, aur, embleme, realizări, timp petrecut, titlu primar, titlu secundar, Aceasta sunt descrise în anexa 1.

Drupal implementează pentru utilizatori un sistem de roluri. Fiecare rol are un set de permisiuni granulare.

Aplicația va avea cele 4 roluri prezentate la capitolul cerințe, urmând a selecta permisiunile potrivite pentru fiecare rol din ecranul de configurare de pe pagina „/admin/people/permissions”.

PERMISSION	ANONYMOUS USER	AUTHENTICATED USER	PRINCE	WARRIOR
Block				
Administer blocks	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Comment				
Administer comment types and settings <i>Warning: Give to trusted roles only; this permission has security implications.</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Administer comments and comment settings	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Edit own comments	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Post comments	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Skip comment approval	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fig.5.6.1. Ecranul de configurare al permisiunilor din Drupal

5.7. Moduri de afișare

Drupal permite personalizarea modului în care este afișat conținutul prin utilizarea modurilor de afișare. Administratorii pot crea mai multe astfel de moduri de afișare pentru diferite contexte.

Primul pas în a configura un mod de afișare este crearea lui și selectarea câmpurilor care se doresc a fi afișate. Drupal caută fișiere „template” în cadrul proiectului, care descriu exact pozițiile fiecărui element în pagină. Fișierele „template” sunt în format .html.twig și implementează limbajul twig, un limbaj simplist dar sigur.

De exemplu, pentru schimbarea modului de bază în care sunt afișate sarcinile, am creat fișierul „licenta-base-task-full.html.twig”, conținutul acestuia fiind disponibil în anexa 2.

The screenshot shows a Drupal task page for a project named "Licenta". The page has a teal header with the project name and a "Home" link. Below the header is a search bar and a navigation menu with options: View, Edit, Delete, Manage display, Revisions, Devel, and Workflow. The main content area features a status filter bar with buttons for "New", "Under work", "Acceptance", "Approved", and "Closed". Below this are three panels: "Details", "Rewards", and "Description". The "Details" panel shows task information: Assignee (admin), Category (Bug), Status (New), Priority (Normal), and Spent time (0.00 hours). The "Rewards" panel shows Experience points (150), Gold (0), and a Primary title (The swift). The "Description" panel contains a text area with the placeholder "Test body". At the bottom, there is a "Comments" section with a user profile picture and a timestamp "Friday, June 1, 2018, 12:00".

Assignee:	Category:	Status:	Priority:	Spent time:	Add spent time
admin	Bug	New	Normal	0.00 hours	

Experience points:	Gold:	Primary title
150	0	The swift

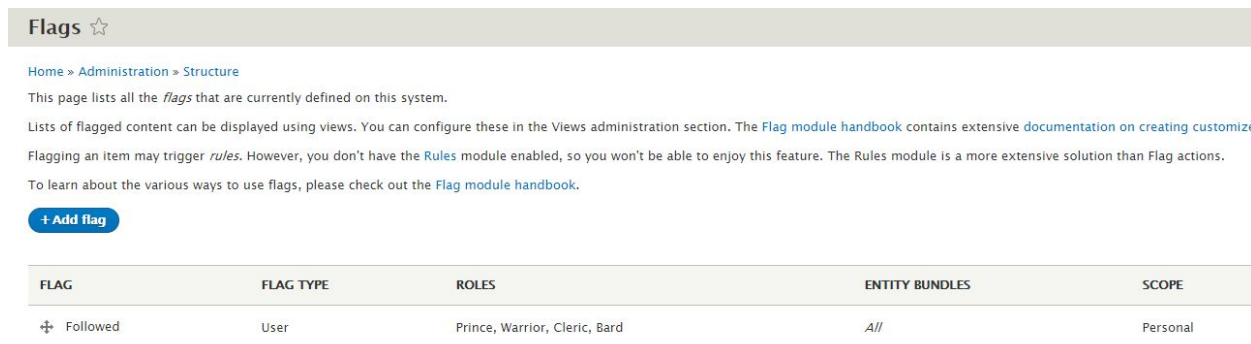
Fig.5.7.1. Pagina unei sarcini

5.8. Sistemul de urmărire al utilizatorilor

Se dorește implementarea unui sistem de urmărire al utilizatorilor pentru ca angajații unei companii să își poată compara progresul cu persoane relevante pentru ei. În mod obișnuit, Drupal nu oferă o astfel de funcționalitate, astfel că este necesară implementarea unui modul contribuit de comunitate. Un modul care se potrivește cerințelor aplicației este „flag” (steag), disponibil la adresa <https://www.drupal.org/project/flag> (accesat 2018). El permite adăugarea unor steaguri pe fiecare entitate a cărei valoare poate fi globală sau individuală.

Pentru implementarea sistemului de urmărire, se vor adăuga steaguri personale pe entitățile de tip utilizator. Astfel, fiecare utilizator va putea marca un alt utilizator ca urmărit.

Modul prin care „flag” reușește să aducă aceste funcționalități este prin introducerea în baza de date a unei intrări cu 2 valori: utilizatorul care a făcut marcajul și entitatea marcată.



Flags ☆				
Home » Administration » Structure				
This page lists all the <i>flags</i> that are currently defined on this system.				
Lists of flagged content can be displayed using views. You can configure these in the Views administration section. The Flag module handbook contains extensive documentation on creating customize				
Flagging an item may trigger <i>rules</i> . However, you don't have the Rules module enabled, so you won't be able to enjoy this feature. The Rules module is a more extensive solution than Flag actions.				
To learn about the various ways to use flags, please check out the Flag module handbook .				
+ Add flag				
FLAG	FLAG TYPE	ROLES	ENTITY BUNDLES	SCOPE
✚ Followed	User	Prince, Warrior, Cleric, Bard	All	Personal

Fig.5.8.1. Adăugarea unui steag în modulul „flag”

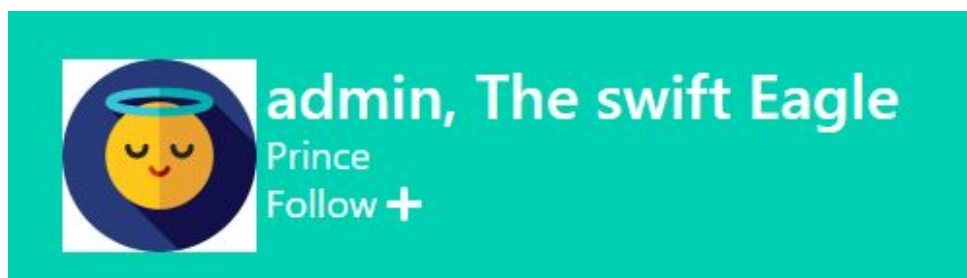


Fig.5.8.2. Butonul de urmărire („follow”) al unui utilizator

5.9. Stările unei sarcini

În cadrul acestei aplicații, o sarcină poate trece prin cele 5 stări descrise la capitolul 3. Tranzițiile între stări sunt restricționate atât de starea curentă cât și de rolul utilizatorului care face modificarea. Pentru a implementa această funcționalitate, se va folosi modulul contribuit „Workflow”, disponibil la <https://www.drupal.org/project/workflow> (accesat 2018). Acesta permite setarea facilă a stărilor unui tip de conținut și a permisiunilor. Astfel, pe tipul de conținut „sarcină” se va adăuga un nou câmp de tip „workflow”.

FROM \ TO	NEW	UNDER WORK	ACCEPTANCE
New		<input checked="" type="checkbox"/> (Author) <input type="checkbox"/> Anonymous user <input type="checkbox"/> Authenticated user <input checked="" type="checkbox"/> Prince <input checked="" type="checkbox"/> Warrior <input checked="" type="checkbox"/> Cleric <input checked="" type="checkbox"/> Bard	<input checked="" type="checkbox"/> (Author) <input type="checkbox"/> Anonymous user <input type="checkbox"/> Authenticated user <input checked="" type="checkbox"/> Prince <input checked="" type="checkbox"/> Warrior <input checked="" type="checkbox"/> Cleric <input checked="" type="checkbox"/> Bard
Under work	<input checked="" type="checkbox"/> (Author) <input type="checkbox"/> Anonymous user <input type="checkbox"/> Authenticated user <input checked="" type="checkbox"/> Prince <input checked="" type="checkbox"/> Warrior <input checked="" type="checkbox"/> Cleric <input checked="" type="checkbox"/> Bard		<input checked="" type="checkbox"/> (Author) <input type="checkbox"/> Anonymous user <input type="checkbox"/> Authenticated user <input checked="" type="checkbox"/> Prince <input checked="" type="checkbox"/> Warrior <input checked="" type="checkbox"/> Cleric <input checked="" type="checkbox"/> Bard
Acceptance	<input checked="" type="checkbox"/> (Author) <input type="checkbox"/> Anonymous user <input type="checkbox"/> Authenticated user <input type="checkbox"/> Prince <input checked="" type="checkbox"/> Warrior <input type="checkbox"/> Cleric <input checked="" type="checkbox"/> Bard	<input checked="" type="checkbox"/> (Author) <input type="checkbox"/> Anonymous user <input type="checkbox"/> Authenticated user <input type="checkbox"/> Prince <input checked="" type="checkbox"/> Warrior <input type="checkbox"/> Cleric <input checked="" type="checkbox"/> Bard	

Fig.5.9.1. Ecranul de configurare al tranzițiilor pentru statusul unei sarcini în modulul „workflow”

5.10. Vizualizări

În Drupal, vizualizările („views”) reprezintă interogarea bazei de date pentru obținerea de conținut și afișarea lui în liste, tabeluri, raporturi, blocuri, etc. Pot fi afișate diverse tipuri de conținut.

Această funcționalitate se găsește în modulul „Views” din nucleul Drupal care vine la pachet cu modulul „Views UI”, care pune la dispoziție o interfață grafică pentru configurarea vizualizărilor. Conținutul poate fi ordonat, filtrat și pus pe orice pagină.

Pentru a respecta cerințele funcționale, vor fi implementate următoarele vizualizări:

- lista de sarcini
- lista de recompense din magazin
- sarcinile completate recent de un utilizator
- clasamentul global
- clasamentul personal
- lista de noutăți

În continuarea lucrării, va fi detaliată implementarea uneia dintre aceste vizualizări: clasamentul personal. Vizualizarea va fi de tip bloc, acest lucru permițând afișarea mai multor vizualizări pe o singură pagina. Tipul de entitate afișat va fi „utilizator”, pentru care se selectează următoarele câmpuri: iconiță, nume, titlu primar, titlu secundar, nivel și puncte de experiență.

Câmpul „titlu primar” este ascuns iar câmpul „titlu secundar” este rescris pentru a concatena și titlul primar. Astfel, cele două câmpuri vor fi afișate în aceeași coloană.

În secțiunea criteriilor de sortare, se va alege o sortare în funcție de experiență, descrescător.

Displays

Block* [+ Add](#)

Display name: Block

TITLE

Title: [Personal leaderboard](#)

FORMAT

Format: [Table](#) | [Settings](#)

FIELDS [Add](#) ▼

User: [Icon \(Icon\)](#)

User: [Name \(Name\)](#)

User: [Primary title \[hidden\]](#)

User: [Secondary title \(Title\)](#)

User: [Level \(Level\)](#)

User: [Experience points \(Experience points\)](#)

FILTER CRITERIA [Add](#) ▼

User: [User status \(= Yes\)](#)

SORT CRITERIA [Add](#) ▼

User: [Experience points \(desc\)](#)

BLOCK SETTINGS

Block name: [None](#)

Block category: [Lists \(Views\)](#)

Allow settings: [Items per page](#)

Access: [Permission](#) | [View user information](#)

HEADER

FOOTER

NO RESULTS BEHAVIOR

PAGER

Use pager: [Full](#) | [Paged, 20 items](#)

More link: [No](#)

Link display: [None](#)

LANGUAGE

Rendering Language: [Content language of view row](#)

Fig 5.10.1. Setările pentru vizualizarea clasamentului personal

O altă setare importantă este adăugarea unei relații cu tabela de steaguri, pentru a filtra doar utilizatorii marcați ca urmăriti de cel curent.

Configure Relationship: Flags

Limit results to only those entity flagged by a certain flag; Or display information about the flag s

► **ADMINISTRATIVE TITLE**

☒ Include only flagged content

If checked, only content that has this flag will be included. Leave unchecked to include all con

Flag *

☒ Followed

By

☒ Current user

☐ Any user

[Apply](#) [Cancel](#) [Remove](#)

Fig.5.10.2. Setările pentru relația cu tabela de steaguri

Leaderboards				
Search				
View	Edit	Delete	Manage display	Devel



Friends leaderboard				
Icon	Name	Title	Level	Experience points
	user1		1	0
	user2		1	0

Fig.5.10.3. Clasamentul personal așa cum apare în aplicație

5.11. Crearea unui modul personalizat

Există o serie de cerințe în cadrul acestui proiect care nu își găsesc rezolvarea în niciun modul, fie el din nucleul de Drupal, sau contribuit. Din acest motiv, este necesară dezvoltarea unui modul pentru a îndeplini aceste cerințe.

Modului care conține rezolvarea problemelor impuse de cerințele funcționale se numește „licenta_base”. Pentru a crea acest modul, este nevoie în primul rând de scrierea fișierului „licenta_base.info.yml”, în folderul „docroot/modules/custom/licenta_base”, fișier ce conține descrierea modului:

```
type: module
name: 'Licenta base'
description: 'Base functionality.'
core: 8.x
```

Una dintre problemele principale ale proiectului este implementarea unor acțiuni personalizate atunci când un nod de tip „sarcină” este modificat. Pentru acest lucru, s-a implementat o funcție de tip „hook_node_save”. La salvarea unui nod, Drupal caută toate funcțiile de tipul „nume_modul_node_save()” din fișierele cu extensia „module” și le execută. În cazul acestui proiect, următoarele acțiuni se doresc a fi realizate la modificarea unei sarcini:

- dacă este o sarcină nouă, i se va atribui un număr de identificare
- dacă statusul ei s-a schimbat în „Acceptată”, i se vor trimite responsabilului recompensele asociate sarcinii, și se va crea o notificare
- dacă un utilizator nou a fost setat ca responsabil cu acea sarcină, i se va trimite o notificare.

Alte funcționalități care apar la salvarea unor tipuri de conținut includ trimiterea unei notificări utilizatorului care devine urmărit de către alt utilizator și incrementarea timpului total petrecut pentru rezolvarea unei sarcini atunci când o entitate de tip „timp petrecut” este salvată

În continuare, este prezentat codul pentru generarea unui număr de identificare pentru o sarcină. Inițial se verifică tipul nodului salvat. Dacă acesta este de tip „sarcină”, se execută o căutare în baza de date care returnează un vector cu toate nodurile de tip sarcină. Sarcinii curente îi este asociat un număr de identificare egal cu **max+1**, unde **max** este valoarea maximă a numerelor de identificare existente.

```
function licenta_base_node_presave(Node $node) {
  if ($node->bundle() == 'task') {
    // Automatically set task number.
    if (empty($node->field_task_number->value)) {
```

```
        $nids = \Drupal::entityQuery('node')->condition('type',
'task')->execute();
        $nodes = \Drupal\node\Entity\Node::loadMultiple($nids);
        $max = 0;
        foreach ($nodes as $task) {
            if ($task->field_task_number->value > $max) {
                $max = $task->field_task_number->value;
            }
        }
        $node->field_task_number->value = $max + 1;
    }
}
```

6. Prezentarea aplicației finale. Dezvoltare ulterioară

Utilizând puterea de construire a unei aplicații web oferită de Drupal, a fost posibilă implementarea unei aplicații care să permită companiilor de dezvoltare software să aducă gamificarea în mediul lor de lucru. De asemenea, site-ul creat oferă facilități în organizarea sarcinilor iar prin datele statistice prezentate, se poate verifica ușor productivitatea angajaților.

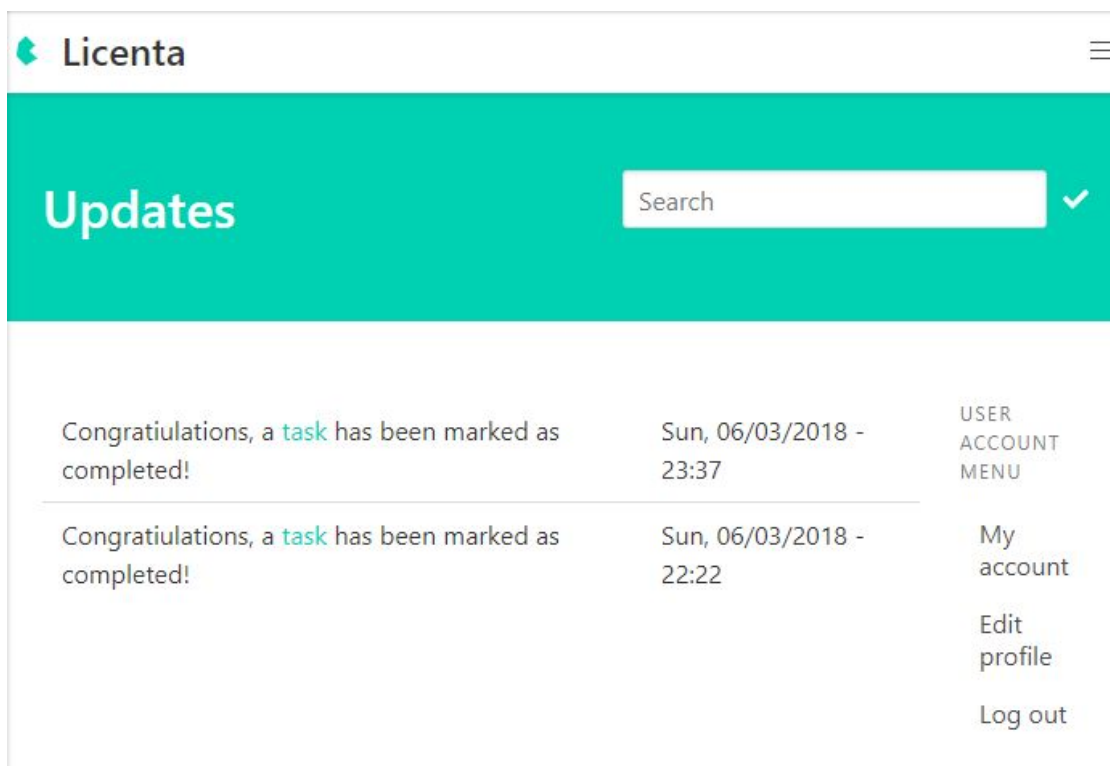
Consider că aplicația dezvoltată are un potențial bun de a crește motivația și gradul de atenție al angajaților unei companii, dacă este folosită corect.

Există totuși o serie de îmbunătățiri care pot fi aduse aplicației. Traducerea în mai multe limbi lipsește deocamdată, dar Drupal oferă suport pentru acest lucru. Procesul ar putea fi însă îndelungat, întrucât fiecare șir de caractere din cadrul site-ului trebuie tradus manual de către un traducător.

O altă îmbunătățire posibilă ar fi introducerea unui sistem de misiuni. O misiune reprezintă rezolvarea unor cerințe dificile în scopul primirii unor recompense. Un exemplu ar fi rezolvarea a cel puțin 5 sarcini într-o zi. Implementarea unui astfel de sistem ar putea fi însă destul de complicată. Spre deosebire de adăugarea unor cerințe noi și a recompenselor, care se face ușor din interfața grafică a aplicației, implementarea unei misiuni noi este o acțiune strict programatică, un responsabil fiind nevoit să scrie cod PHP și să îl testeze pentru a crea acest tip de conținut.

Pentru dezvoltarea ulterioară a aplicației, tot codul este disponibil pe GitHub, la adresa <https://github.com/stefanbutura/licenta> (accesat 2018). Programatorii noi vor descărca

codul folosind git și vor emula containerele folosind Docker. Setările de Drupal din baza de date sunt exportate ca fișiere cu extensia „.yaml” și pot fi importate folosind comanda „drush cim sync -y”. După acești pași simpli, dezvoltarea ulterioară poate începe.



The screenshot shows the 'Licenta' application interface. At the top, there is a teal header with the 'Licenta' logo and a search bar. Below the header, the main content area displays two notifications, each stating 'Congratulations, a task has been marked as completed!'. To the right of these notifications is a 'USER ACCOUNT MENU' with options: 'My account', 'Edit profile', and 'Log out'.

Notification	Timestamp	User Account Menu
Congratiulations, a task has been marked as completed!	Sun, 06/03/2018 - 23:37	My account
Congratiulations, a task has been marked as completed!	Sun, 06/03/2018 - 22:22	Edit profile
		Log out

Fig.6.1. Pagina de actualizări relevante pentru utilizatorul curent

Icon shop



Price: 111 gold

[Buy](#)



Price: 222 gold



[Buy](#)

Fig.6.2. Pagina cu magazinul de recompense

Leaderboards

[View](#) [Edit](#) [Delete](#) [Manage display](#) [Devel](#)

Friends leaderboard

Icon	Name	Title	Level	Experience points
	xd		1	0
	xdd		1	0

Global leaderboard



Icon	Name	Title	Level	Experience points
	admin	The swift Eagle	2	150

Fig.6.3. Pagina cu clasamentele utilizatorilor

Licetia

Home Le



admin, The swift Eagle
Prince
Follow +

Search

View Shortcuts Edit Manage display Devel

Statistics

Total completed tasks	1
Time spent	

Recent tasks

[Test task 3](#)

Achievements

- Finished 350 tasks
- Has been rank 1 for 2 months in a row
- Spent 200 hours on tasks in one month

Badges




Fig.6.4. Pagina de profil a unui utilizator

7. Bibliografie

1. *Apache Solr*. Wikipedia. https://en.wikipedia.org/wiki/Apache_Solr Accesat 2018.
2. Cunningham, C., Zicherman, G. (2011). *Gamification By Design*. O'Reilly Media.
3. Deterding, S., Khaled, R., Nacke, L., Dixon, D. (2011). *Gamification: Toward a definition*. Proceedings of CHI 2011 Workshop Gamification: Using Game Design Elements in Non-Game Contexts. 6-9.
4. Deterding, S., Dixon, D., Khaled, R., Nacke, L. (2011). *From Game Design Elements to Gamefulness: Defining Gamification*. Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011. 11. 9-15.
5. *Docker (software)*. Wikipedia. [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software)) Accesat 2018
6. Documentație Bulma. <https://bulma.io/> Accesat 2018.
7. Documentație Twig. <https://twig.symfony.com/> Accesat 2018.
8. Documentație Nginx. <http://wiki.nginx.org/Main> Accesat 2018.
9. Élthes, Z. (2016). *Utilizarea gamificarii în eLearning*. "Babeș-Bolyai" University, Faculty of Economics Sciences.
10. Garcia, F., Pedreira, O., Piattini, M., Cerdeira-Pena, A., Penabad, M. (2017). *A Framework for Gamification in Software Engineering*. *Journal of Systems and Software*. 132.
11. Hamari, J., Koivisto, J., Sarsa, H. (2014). *Does Gamification Work? — A Literature Review of Empirical Studies on Gamification*. *Proceedings of the Annual Hawaii International Conference on System Sciences*. 10.

12. Humphrey, W.S. (2000). *The Personal Software Process (PSP)*. Carnegie Mellon Software Engineering Institute.
13. *Operating-system-level virtualization*.
https://en.wikipedia.org/wiki/Operating-system-level_virtualization Accesat 2018.
14. Pelling, N. (2011). *The (short) prehistory of gamification*. Funding Startups(& other impossibilities).
15. Rețetă de configurare Nginx pentru Drupal.
<https://www.nginx.com/resources/wiki/start/topics/recipes/drupal/> Accesat 2018.
16. *Taxonomie*. Wikipedia. <https://ro.wikipedia.org/wiki/Taxonomie> Accesat 2018.
17. Vaughan-Nichols, S., *What is Docker and why is it so darn popular?*. ZDNet.
<https://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/> Accesat 2018.

8. Anexe

8.1. Detalierea entităților și a câmpurilor componente

Utilizator			
Câmp	Tip	Tip referențiat	Cardinalitate
field_icon	referință	Iconiță	1
field_primary_title	referință	Titlu primar	nelimitat
field_secondary_title	referință	Titlu secundar	nelimitat
field_experience	Integer		1
field_level	Integer		1
field_rewards	referință	Iconiță, titlu primar/secundar, realizare	nelimitat
field_gold	Întreg		1
field_profile_badges	referință	Emblemă	3
field_profile_achievements	referință	Realizare	3
field_spent_time	întreg		1
Sarcină			
Câmp	Tip	Tip referențiat	Cardinalitate
field_status	referință	termen - statusuri sarcini	1
field_priority	referință	termen - priorități sarcini	1
field_asignee	referință	utilizator	1
field_category	referință	termen - categorii sarcini	1
field_due_date	Date		1
field_estimated_time	Time		1
body	text		1
field_experience_reward	Integer		1
field_rewards	referință	Iconiță, titlu primar/secundar, realizare	nelimitat

Timp petrecut			
Câmp	Tip	Tip referențiat	Cardinalitate
field_task	referință	sarcină	1
field_user	referință	utilizator	1
field_spent_time	număr fracționar		1
Emblemă			
Câmp	Tip	Tip referențiat	Cardinalitate
field_image		Imagine	1
Realizare, titlu primar/secundar			
Câmp	Tip	Tip referențiat	Cardinalitate
Icon			
Câmp	Tip	Tip referențiat	Cardinalitate
field_image		Imagine	1
field_price		Întreg	1
Notificare			
Câmp	Tip	Tip referențiat	Cardinalitate
body	text		1
field_assignee	referință	utilizator	1

8.2. Fișierul template folosit pentru afișarea unei sarcini

```

<div class="columns">
  <div class="column">
    <h2 class="panel-heading">{{ 'Details' | trans }}</h2>
    <div class="panel-block">
      {{ left }}
    </div>
  </div>
</div>

```

```

<a href="{{ '/task/' ~
content['#node'].field_task_number.value ~ '/add-time' }}">{{ 'Add
spent time' | trans }}

```