MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

EWEBSITEAUCTION - SISTEM ONLINE DE LICITAȚII PENTRU SITE-URI WEB

LUCRARE DE LICENȚĂ

Absolvent: **Marius-Vlad CRIȘAN**

Coordonator ştiinţific: **Șef lucr. ing. Cosmina IVAN**

2013

MINISTERUL EDUCAȚIEI, CERCETĂRII, TINERETULUI ȘI SPORTULUI

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

VIZAT,
DIRECTOR DEPARTAMENT,
Prof. dr. ing. Rodica POTOLEA

DECAN,
Prof. dr. ing. Liviu MICLEA

Absolvent: **Marius-Vlad CRIȘAN**

EWEBSITEAUCTION - SISTEM ONLINE DE LICITAȚII PENTRU SITE-URI WEB

1. **Enunţul temei:** *Proiectul își propune realizarea unui sistem online, care să permită utilizatorilor cu acces la Internet sa creeze licitații și să cumpere site-uri web în siguranță.*

2. **Conţinutul lucrării:** *Cuprins, Introducere, Obiectivele proiectului, Studiu bibliografic, Analiză și fundamentare teoretică, Proiectare de detaliu și implementare, Testare și validare, Manual de instalare și utilizare, Concluzii, Bibliografie, Anexe.*

3. **Locul documentării:** Biblioteca Universității Tehnice din Cluj-Napoca.

4. **Consultanţi:** șef lucr. ing. Cosmina Ivan

5. **Data emiterii temei:** 1 noiembrie 2012

6. **Data predării:** 8 Iulie 2013

Absolvent: _____

Coordonator ştiinţific: _____

Cuprins

1. Introducere ... 6
1.1. Contextul General ... 6
1.2. Contextul Aplicaţiei ... 7
1.3. Contribuţii Personale ... 8
2. Obiectivele Proiectului ... 9
3. Studiu Bibliografic ... 11
3.1. Conceptul de Licitaţie ... 11
3.2. Sisteme similare ... 12
3.2.1. WebsiteBroker.com ... 13
3.2.2. DigitalPoint Marketplace ... 14
3.2.3. Flippa.com ... 14
4. Fundamentare Teoretică şi Analiză ... 16
4.1. Fundamentare Teoretică ... 16
4.1.1. Enterprise JavaBeans ... 16
4.1.2. JavaServer Pages ... 18
4.1.3. Spring MVC ... 19
4.1.4. Framework-ul TopLink ... 21
4.1.5. Alexa Rank API ... 22
4.1.6. Google API ... 23
4.1.7. PayPal Data Transfer API ... 23
4.1.8. WhoIS API ... 24
4.2. Analiza Sistemului ... 25
4.2.1. Cerinţele Funcţionale ale sistemului ... 26
4.2.2. Cerinţe Non-Funcţionale ale Sistemului ... 27
4.2.3. Cazuri de utilizare ... 28
5. Proiectare de detaliu şi implementare ... 36
5.1. Arhitectura Conceptuală ... 36
5.2. Nivelul de Prezentare ... 37
5.3. Nivelul de Controller ... 40
5.4. Nivelul de Model ... 43
5.5. Deployment Digram ... 46
5.6. Nivelul de Date ... 47
5.7. Interacţiunea dintre componente ... 51
6. Testare şi Validare ... 54
7. Manual de Instalare şi Utilizare ... 59
7.1. Manual de instalare ... 59
7.2. Manual de Utilizare ... 59

8. Concluzii 64
8.1. Realizări 64
8.2. Dezvoltări Ulterioare 65
9. Bibliografie 67
Anexa 1. Acronime 68
Anexa 2. Glosar de Termeni 69
Anexa 3. Lista de figuri și tabele 70

1. Introducere

Sistemul de licitaţie este unul dintre cele mai populare sisteme de schimb din lumea comerţului. Originile licitaţiilor datează din Roma antică, când un senator roman (după cum spune legenda), a licitat pe imperiul Roman, pentru a securiza controlul acestuia. Din Roma antică, licitaţiile, au început să se extindă şi în alte zone, adunând tot mai multă faimă o dată cu trecerea timpului, până au ajuns la nivelul actual. În momentul de faţă s-a ajuns ca guvernele, companiile şi oamenii de afaceri, să realizeze în mod regulat licitaţii pentru o gamă largă de produse şi servicii.

Licitaţiile online pot fi privite ca un mod prin care mediul de afaceri s-a dezvoltat, pentru a beneficia de toate serviciile oferite de Internet, odată cu apariţia şi extinderea lui la un nivel mondial. Datorită dezvoltării produselor software sofisticate, care permit mediului de afaceri să cumpere şi să vândă cu ajutorul Internetului, sub condiţii securizate, s-a ajuns la dezvoltarea de noi strategii potrivite pentru acest mediu şi la apariţia unor noi modele de business. Licitaţiile online au depăşit aşteptările pentru a duce la capăt tranzacţiile intermediare într-un mediu business-to-business(B2B), într-un mod eficient.

Sistemele de licitaţii online, au şi alte nume, precum: e-Licitaţii, licitaţii electronice, etc. Cerinţele unei licitaţii online, pot fi specificate într-un mod mai concis de către client, iar acest lucru ar trebui să devină o practică bună, când este realizată într-un mod transparent.

Licitaţiile online au devenit tot mai răspândite în multe domenii industriale. Acestea nu includ doar produsul, spre a fi vândut, ci şi servicii care pot fi oferite cumpărătorilor. Datorită costurilor mici, expansiunea a făcut ca sistemul să crească şi să se dezvolte foarte mult.

În sistemele de licitaţii online licitanţii pot fi salvaţi într-o bază de date, de unde pot fi monitorizaţi foarte uşor. Acest lucru permite crearea foarte uşor a unor multitudini de rapoarte privind situaţia sistemului şi reduce necesitatea folosirii hârtiei şi a timpului necesar pentru întreţinere.

1.1. Contextul General

În ziua de azi, Internetul este accesibil majorităţii populaţiei, iar datorită acestui fapt numărul de site-uri existente a crescut foarte mult. Deoarece site-urile tind să se dezvolte foarte rapid, tot mai multe persoane nu mai au resursele necesare să se ocupe de ele şi astfel a apărut necesitatea unei platforme în care aceştia să îşi poată expune şi într-un final vinde munca în siguranţă.

Serviciile unei astfel de platforme nu au restricţii geografice, făcând astfel disponibilă tranzacţionarea site-urilor între toţi utilizatorii de Internet, indiferent de zona în care se află. Din acest motiv, acest domeniu este într-o continuă creştere, realizându-se tot mai multe tranzacţii, în fiecare zi.

Este important de ţinut cont, că acesta este un domeniu relativ nou (apărut cu doar câţiva ani în urmă) şi de aceea trebuie să fim precauţi când dorim să vindem/cumpărăm site-uri. Prudenţa este necesară deoarece datele furnizate de vânzători, pot fi falsificate destul de uşor, iar pentru că utilizatorii nu au experienţă în acest domeniu, le va fi greu să decidă dacă datele sunt adevărate.

1.2. Contextul Aplicaţiei

Licitaţiile online, denumite şi licitaţii virtuale, reprezintă un serviciu în care participanţii pot realiza diverse tranzacţii, pentru a achiziţiona obiecte la un preţ mai bun decât cel de pe piaţă.

Acestea încearcă să copieze licitaţiile tradiţionale, neelectronice şi sunt în aşa maniera concepute încât să permită existenţa mai multor licitatori simultan.

Platforma de licitaţii care va fi implementa se va numi ***EWebsiteAuction***. Aceasta se va încadra în categoria licitaţiilor de tip ***„Silent”***, deoarece va permite existenţa mai multor licitaţii active în acelaşi moment, preţul licitaţiei va fi crescător, vor avea acces la licitaţie mai mulţi licitanţi, iar câştigător va fi desemnat cel cu oferta cea mai mare.

Există un număr foarte mare de avantaje al acestui model, iar câteva dintre acestea vor fi enumerate mai jos.

Pot exista mai mulţi licitatori: Deoarece licitaţiile online oferă potenţialul de a obţine un preţ mai mic decât cel obişnuit şi datorită uşurinţei cu care se poate licita(realizând doar câteva click-uri) din propria casă, fără a fi nevoie de a te deplasa la o anumită locaţie.

Pot exista mai mulţi vânzători: Aceştia au potenţialul de a obţine un preţ mai mare, în cazul în care 2 sau mai multe persoane doresc să cumpere site-ul respectiv. Un alt motiv reprezintă uşurinţa cu care se realizează tranzacţia, fără a fi nevoie să te întâlneşti personal cu cumpărătorul.

Nu există limită de timp: Faţă de licitaţiile obişnuite aici se pot crea licitaţii noi la orice oră, fără a fi nevoie să aştepţi un anumit interval orar, sau anumite zile.

Nu există limitări geografice: Aceasta este una dintre principalele motive pentru care sistemele de licitaţii online au atras din ce în ce mai mulţi utilizatori. Internetul permite cumpărătorilor şi vânzătorilor să participe într-o licitaţie online, din orice locaţie în care există o conexiune la Internet. Costul participării este practic zero, nefiind nevoie să te deplasezi în anumite locaţii, pentru a lua parte.

Abilitatea de a verifica vânzătorul: Datorită sistemului de feedback şi rating implementat de majoritatea sistemelor de licitaţii, utilizatorii, pot verifica profilul vânzătorului şi modul în care acesta a realizat tranzacţii înainte.

Descrierea detaliată a produsului: Licitatorii au acces în orice moment al licitaţiei, la o descriere detaliată a produsului, realizată chiar de către vânzător.

Deoarece sistemul de licitare online al site-urilor se desfăşoară pe Internet există numeroase riscuri în realizarea tranzacţiilor. Este dovedit faptul că cele mai multe fraude online se realizează în domeniul licitaţiilor online, deoarece există posibilitatea ca vânzătorul să nu mai livreze produsul, sau produsul să fie de o calitate inferioară faţă de cea specificată. Ţinând cont de acestea, am decis să enumăr câteva sfaturi, care au fost identificate în bibliografia de specialitate, care ar trebui să ajute atât cumpărătorii cât şi vânzătorii.

Verificare profil vânzător: Înainte de a licita pentru un anumit site, sau înainte de a încheia tranzacţia, este necesar să verificaţi profilul utilizatorului, pentru a vedea ce tranzacţii a făcut înainte. Majoritatea site-urilor de licitaţii au implementat un sistem de feedback şi comentarii, care au menirea să ajute utilizatorii în luarea deciziei.

Salvaţi datele tranzacţiei: Este important să salvaţi toate plăţile şi mesajele schimbate cu vânzătorul/cumpărătorul pentru a avea dovezi în cazul în care nu decurge totul conform înţelegerii.

Verificaţi datele site-ului: Este extrem de important să verificaţi datele oferite de platformele de licitaţii în legătură cu site-ul vândut, dar pe lângă acestea, trebuie să căutaţi şi dumneavoastră dovezi că datele oferite de vânzător sunt reale şi nu falsificate.

Alegeţi o metodă de plată sigură: Cumpărătorii ar trebui să folosească o metodă de plată care să poată fi urmărită, cum ar fi PayPal, MoneyBookers, Escrow. În cazul realizării plății prin PayPal, dacă tranzacţia nu decurge cum ar trebui, aveţi posibilitatea să iniţiati o dispută asupra plății, existând o mare şansă să recuperaţi toată suma plătită.

Citiţi descrierea produsului: Înainte de a licita asupra unui produs, trebuie să citiţi cu foarte mare atenţie descrierea produsului respectiv, pentru a nu avea surpriza descoperirii unor defecte specificate în descriere, dar pe care dumneavoastră nu le-aţi observat.

1.3. Contribuții Personale

Lucrarea de față reprezintă un sistem de licitații online destinat cumpărării și vinderii site-urilor web de către utilizatorii care au acces la Internet.

Pentru implementarea unui astfel de sistem, vor fi aduse următoarele contribuții:

- **Analiză Sisteme Similare**. Se vor selecta cele mai performante astfel de sisteme, pentru a se realiza o comparație cu sistemul ce se dorește a fi implementat, pentru a se putea descoperi funcționalitățile existente, îmbunătățirile care s-ar putea aduce și modul de funcționare.
- **Proiectarea conform cerințelor**. Sistemul va fi proiectat astfel încât să respecte toate cerințele funcționale și nonfuncționale studiate în lucrările de specialitate.
- **Implementarea Proiectului**. Se va realiza o implementare a tuturor cerințelor funcționale ale sistemului și o integrare a acestuia cu alte servicii, pentru a putea oferi utilizatorilor o experiență placută.
- **Testare și Validare**. Se va realiza o testare modulară a componentelor de business și a interfeței cu utiliatorul, folosind unelte de testare automată, pentru a asigura buna funcționare a sistemului și pentru a valida datele returnate.

2. Obiectivele Proiectului

Proiectul are ca scop realizarea unei platforme de licitaţii online: **EWebsiteAuction**, pentru cumpărarea şi vânzarea site-urilor web şi a domeniilor, care să inspire utilizatorilor siguranţă în realizarea tranzacţiilor. Utilizatorii acestui site ar trebui să poată realiza într-un mod cât mai rapid şi uşor operaţiile prezentate mai jos.

- **Înregistrare** . Utilizatorii trebuie să aibă posibilitatea de a-şi crea un cont pentru a putea beneficia de toate funcţionalităţile site-ului. La crearea contului va fi necesară verificarea adresei de email, pentru a putea oferi o siguranţă mai mare utilizatorilor.

- **Căutare avansată**. Utilizatorii ar trebui să poate realiza o căutare avansată a licitaţiilor existente, folosind mai multe cerinţe în acelaşi timp. Aceştea ar putea realiza căutări în funcţie de: preţul curent, numărul de vizitatori ai site-ului, profitul obţinut şi numărul de impresii.

- **Urmăreşte Licitaţii**. Utilizatorii logaţi trebuie să aibă posibilitatea de a urmări licitaţiile care li se par interesante, pentru a le putea găsi, fără a mai fi nevoie să le caute printre toate licitaţiile existente, în cazul în care vor dori să liciteze.

- **Licitare** . Utilizatorii logaţi trebuie să aibă posibilitatea de a licita la licitaţiile deschise şi să îşi asume responsabilitatea în cazul în care vor deveni câştigători.

- **Acceptare bid**. Creatorii unei licitaţii, trebuie să aibă posibilitatea de a accepta sau respinge suma licitată. Astfel vor avea posibilitatea verificării contului persoanei care a licitat, înainte ca suma licitată de aceasta să fie luată în considerare.

- **Trimite ofertă**. Creatorii unei licitaţii, trebuie să aibă posibilitatea negocierii preţului site-ului, cu persoana care a licitat cel mai mult, în cazul în care preţul atins la licitaţie este mai mic decât preţul rezervat.

- **Creare licitaţii**. Utilizatorii logaţi ar trebui să aibă posibilitatea să creeze licitaţii noi, care să poată avea setat un preţ de pornire, un preţ rezervat de la care site-ul va trebui vândut şi un preţ de cumpărare rapidă pentru a putea vinde site-ul mai repede în cazul unui cumpărător interesat.

- **Licitaţiile mele**. Creatorii licitaţiilor ar trebui să aibă o pagină separată cu toate licitaţiile create de ei, pentru a le putea ţine evidenţa mai uşor.

- **Feedback şi Rating**. Câştigătorii unei licitaţii trebuie să aibă posibilitatea lăsării unui rating şi mesaj vânzătorului, pentru a descrie modul în care s-a desfăşurat tranzacţia, îmbunătăţind astfel încrederea cumpărătorului în vânzător.

- **Trimite Mesaj**. Utilizatorii logaţi trebuie să aibă posibilitatea de a trimite mesaje altor utilizatori, pentru a putea realiza o mai bună evaluare a site-ului şi pentru un transfer mai rapid în cazul câştigării licitaţiei.

- **Management Useri**. Administratorii trebuie să aibă posibilitatea deactivarii contului unor useri, în cazul unui comportament necorespunzător, sau deactivarea/activarea unor licitaţii, în cazul în care vor exista probleme.

- **Setare Site** Administratorii site-ului ar trebui să poate realiza anumite setări ale site-ului, cum ar fi: setare adresa paypal pentru plată, setare preţ pornire licitaţie, setare preţ îmbunătăţiri licitaţie.

Proiectul realizat este destinat exclusiv realizării licitaţiilor de site-uri, fără a putea fi folosit şi pentru alte tipuri de produse. Am decis să îl specializez doar pentru o anumită categorie de licitaţii, pentru a putea oferi o calitate mai mare a serviciilor oferite de sistem. Platforma, nu poate fi folosită pentru tranzacţionarea şi altor tipuri de produse deoarece nu permite realizarea unor astfel de licitaţii, datorită câmpurilor necesare a fi completate în mod obligatoriu (gândite pentru a pune în evidenţă site-urile listate) în momentul creării unei noi licitaţii.

3. Studiu Bibliografic

În acest capitol vor fi prezentate principalele concepte şi tehnologii relevante pentru dezvoltarea şi implementarea cerinţelor funcţionale şi non-funcţionale ale proiectului.

În articolul [2] este prezentat un ghid al sistemelor de licitaţii şi a procesului de licitare pentru oamenii de ştiinţă din domeniul calculatoarelor. O mare parte a articolului face parte dintr-o subcategorie a teoriei jocului, numită "teoria licitaţiei".

În referinţa [7] sunt prezentate fundamentele preţului dinamic şi a licitaţiilor, explicând astfel modul de funcţionare al licitaţiilor şi tranzacţiilor, în funcţie de numărul de vânzători şi cumpărători.

În referinţa [5] sunt prezentate regulile fundamentale ale unui soft de licitaţii, procesele şi componentele implicate, precum şi modul de organizare a interfeţei, pentru o utilizare cât mai uşoară. Aceste informaţii au ajutat la înţelegerea mai bine a modului de funcţionare a licitaţiilor şi funcţionalităţilor importante care ar trebui realizate pentru o funcţionare normală şi în siguranţă a sistemului.

În referinţele [4] şi [6] sunt prezentate principalele aspecte ale licitării cu ajutorul Internetului şi a importanţei licitaţiilor online pentru extinderea pieţelor business-to-business.

În cartea [3] sunt prezentate şi exemplificate principalele paradigme de calcul distribuit şi tehnologiile cu care se pot realiza astfel de sisteme.

În referinţa [8] este prezentată o carte care prezintă modalitatea de funcţionare a JSP şi principalele beneficii aduse de această tehnologie, faţă de alte tehnologii existente. Articolul [1] prezintă principalele caracteristice ale EJB, principalele componente şi modul de funcţionare al acestuia. Aceste referinţă au ajutat la implementarea nivelelor de business şi interfeţei utilizator a sistemului.

În referinţa [9] este prezentat modul de funcţionare şi implementare a framework-ului Spring MVC, iar în referinţa [10] este prezentat modul de realizare a persistenţei folosind framework-ul TopLink, cât şi beneficiile aduse de acesta. Aceste articole m-au ajutat să realizez persistenţa datelor şi realizarea unei arhitecturi împărţită pe 3 nivele.

Referinţa [11] descrie modul de integrare şi funcţionare a API-ului furnizat de către Alexa, iar referinţa [12] descrie modul de funcţionare a funcţiei de hash Jenkins, folosită în implementarea API-ului furnizat de Google.

Referinţa [13] prezintă modul de funcţionare şi implementare a Api-ului de plată prin PayPal. Acesta a ajutat la integrarea unui modul de plată în sistemul de licitaţii, folosind PayPal.

Articolul [14] prezintă modul de funcţionare şi organizare a protocolului WhoIs, iar referinţa [15] descrie modul de integrare a API-ului pentru preluare unui thumbnail al unui site.

3.1. Conceptul de Licitaţie

Licitaţiile au existat într-o formă sau alta de foarte mulţi ani, dacă ar fi să ne gândim la licitaţii, ca un schimb de bani, putem spune că au apărut de la apariţia banilor. De la începuturi, până în ziua de azi, licitaţiile au fost într-o continuă dezvoltare şi extindere. Astfel că în ziua de azi, există un număr foarte mare de utilizatori care folosesc sistemele de licitaţii, datorită extinderii Internetului şi a sistemelor de licitaţii online, care au făcut că tot procesul să fie foarte uşor de realizat.

Conform referinţei [4] licitaţiile online reprezintă o nouă oportunitate de a stabili noi legături mai puternice, faţă de membri canalului. Campaniile nu mai trebuie să se îngrijoreze că

dependenţa faţă de un furnizor i-ar putea pune în situaţia de a nu avea destule produse sau că furnizorul va creşte preţul doar pentru acest motiv.

În bibliografia de specialitate [2] au fost identificate următoarele tipuri de licitaţii online:

Licitaţii Englezeşti

Acestea reprezintă standardul licitaţiilor şi au o singură dimensiune, deoarece singurul aspect al produsului licitat este preţul. Pot fi şi bazate pe vânzător, deoarece acesta poate fixa un preţ rezervat, sub care nu va vinde produsul. În cadrul acestei licitaţii, toată lumea este conştientă de bid-ul făcut de celelalte persoane, iar bid-urile vor fi într-o ordine crescătoare. Câştigătorul licitaţie, va fi desemnat cel care a plasat cel mai mare bid, iar durata poate fi una absolută (ex până la ora 4pm) sau relativă (după 10 minute de la plasarea ultimului bid).

Licitaţii Olandeze

Acestea au de asemenea o singură dimensiune şi sunt foarte asemănătoare cu cele englezeşti. Cea mai mare diferenţa dintre cele 2 tipuri de licitaţii este faptul că bid-urile sunt descrescătoare. Într-o licitaţie olandeză tipică, licitatorul începe de la un preţ foarte mare, care va fi scăzut rapid, iar prima persoană care acceptă preţul va fi desemnată câştigătoare. Licitaţiile olandeze sunt numite şi licitaţii "*ceas descendent*", deoarece în practică licitantul indica deseori preţul, folosind un ceas mare.

Licitaţii Japoneze

La acest tip de licitaţie, licitantul foloseşte un preţ ascendent, iar licitatorii se retrag când preţul devine prea mare pentru ei. Câştigătorul licitaţiei va fi desemnată ultima persoană care rămâne, devenind astfel şi ultimul licitator.

Licitaţii „Silent"

Acestea sunt cel mai des folosite în organizarea licitaţiilor cu scopuri caritabile. În cadrul acestei licitaţii, bid-urile sunt scrise pe foaie, împreună cu numele persoanei care licitează, iar licitaţia se încheie la o oră prestabilită. Licitaţiile mute sunt foarte asemănătoare cu cele englezeşti, singura mare diferenţa fiind faptul că se pot vinde mai multe produse simultan. Din acest motiv acest tip de licitaţie se aseamănă cel mai mult cu licitaţiile online actuale.

Abilitatea de a licita pentru produse, lucruri care să fie realizate şi contracte a revoluţionat modul în care companiile îşi desfăşoară afacerile, iar din acest motiv multe companii şi agenţii guvernamentale îşi înregistrează propriile procese şi proceduri de licitare. Acestea simt că posibilitatea de a realiza licitaţii pe Internet reprezintă o şansă în plus de a avea acces la mult mai mulţi participanţi decât înainte.[6]

3.2. Sisteme similare

Pentru a putea fi prezentate funcţionalităţile necesare unui sistem de licitaţii online, au fost studiate cercetările deja existente în implementarea şi dezvoltarea unei astfel de platforme. Conform articolului [7] s-a reuşit identificarea principalele funcţionalităţi, pe care ar trebui să le aibă un sistem de licitaţii.

Având în vedere că în acest domeniu totul se desfăşoară cu ajutorul Internetului, fără a fi nevoie să te întâlneşti faţă în faţă cu cealaltă parte a tranzacţiei, principala preocupare a tuturor sistemelor de acest tip este reprezentată de veridicitatea datelor introduse de vânzător şi asigurarea unui nivel de siguranţă pentru cumpărător.

Următoarea funcţionalitate reprezintă transmiterea cu uşurinţă a mesajelor, pentru a putea comunica cât mai uşor şi rapid cu cumpărătorul/vânzătorul şi pentru a putea realiza transferul

site-ului în siguranţă. Este necesar să existe un istoric al mesajelor, pentru a putea urmări cu uşurinţă toate mesajele schimbate între cele 2 părţi ale tranzacţiei.
Conform articolului [5] o platformă de licitaţii ar trebui să conţină următoarele procese:

- **Iniţializare**
- **Înregistrare**
- **Afişare Preţ**
- **Plasare Bid**
- **Determinare Câştigător**
- **Plată Tranzacţie**
- **Notificare Câştigător**

În funcţie de aceste criterii am ales câteva dintre cele mai bune sisteme de licitaţii, pentru a face o comparaţie cu aplicaţia dezvoltată: **EwebsiteAuction.**

3.2.1. WebsiteBroker.com

Aceasta este o platformă de licitaţii online, care permite cumpărarea/vinderea site-urilor şi domeniilor.

Dintre cele mai importante caracteristici, putem menţiona urmărirea licitaţiilor existente, căutarea licitaţiilor în funcţie de anumite criterii, oferirea resurselor pentru a cumpăra şi vinde site-uri. Cu ajutorul acestui site, utilizatorii pot căuta în funcţie de anumite categorii site-uri de care ar putea fi interesaţi să le cumpere.

Sistemul de licitaţie nu permite vizualizarea ofertelor, doar de către proprietarul site-ului, făcând astfel mai greu acest proces.

Comparaţie între EWebsiteAuction şi Website Broker

Cele 2 site-uri se adresează aceloraşi utilizatori, iar principalele asemănări şi deosebiri sunt prezentate în tabelul de mai jos.

Funcţionalităţi Generale	EWebsiteAuction	Website Broker
Căutare Avansate	DA	DA
Urmăreşte Licitaţii	DA	DA
Trimite Mesaje	DA	DA
Vizualizare Profil	DA	NU
Trimite Feedback	DA	NU
Generare Informaţii Site	DA	NU
Zona Tranzacţionare	DA	NU
Verificare Telefon	NU	NU
Verificare IP	NU	NU
Verificare Email	DA	DA

Tabel 3.1 Comparaţie între EWebsiteAuction şi Website Broker

În urma analizei se poate observa că **EwebsiteAuction** oferă în plus funcţionalităţi importante, care fac procesul de licitare mult mai uşor. Spre deosebire de **EwebsiteAuction** acest sistem nu permite vizualizarea profilului vânzătorului, nu oferă posibilitatea trimiterii feedback şi nu oferă o zonă de tranzacţionare în care vânzătorul şi cumpărătorul să poată pune la punct toate detaliile tranzacţiei.

3.2.2. DigitalPoint Marketplace

DigitalPoint este un forum care furnizează în cadrul lui un sistem de licitaţii. Utilizatorii, pot crea licitaţii, destul de uşor, dar procesul de căutare este foarte anevoios, de multe ori fiind nevoie să verifici toate licitaţiile, pentru a găsi ceea ce te interesează.

Un mare dezavantaj al acestei platforme, este sistemul de feedback, care este foarte greu de folosit şi de cele mai multe ori putând fi influenţat destul de uşor.

Comparaţie între EWebsiteAuction şi DigitalPoint Marketplace

Funcţionalităţi Generale	EWebsiteAuction	DigitalPoint Marketplace
Căutare Avansate	DA	NU
Urmăreşte Licitaţii	DA	NU
Trimite Mesaje	DA	DA
Vizualizare Profil	DA	DA
Trimite Feedback	DA	NU
Generare Informaţii Site	DA	NU
Zona Tranzacţionare	DA	NU
Verificare Telefon	NU	NU
Verificare IP	NU	NU
Verificare Email	DA	DA

Tabel 3.2 Comparaţie între EWebsiteAuction şi DigitalPoint Marketplace

DigitalPoint, este util deoarece este uşor de folosit, de către vânzători, dar are un mare dezavantaj faţă de restul platformelor de acest tip, deoarece nu oferă siguranţă cumpărătorilor, datorită sistemului de feedback care poate fi manipulat.

În urma analizei realizate se poate observe că **EwebsiteAuction** este net superior faţă de DigitalPointMarketplace, deoarece oferă utilizatorilor mult mai multe servicii care au menirea să ajute utilizatorii în procesul de cumpărare/vânzare site.

3.2.3. Flippa.com

Acesta este unul dintre cele mai vechi şi mai mari sisteme de licitaţii online. Datorită faptului că cele mai multe fraude se fac în domeniul licitaţiile, site-ul este într-o continuă schimbare, făcând astfel experienta utilizatorului cât mai sigură.

Printre principalele elemente de siguranţă se enumeră, verificarea numărului de telefon în momentul realizării contului şi verificarea adresei de email. Site-ul este uşor de folosit, iar datorită sistemului de feedback şi rating, utilizatorii se simt foarte confortabil în realizarea unei tranzacţii.

Comparaţie între EWebsiteAuction şi Flippa

Funcţionalităţi Generale	EWebsiteAuction	Flippa
Căutare Avansate	DA	DA
Urmăreşte Licitaţii	DA	DA
Trimite Mesaje	DA	DA
Vizualizare Profil	DA	DA
Trimite Feedback	DA	DA
Generare Informaţii Site	DA	DA
Zona Tranzacţionare	DA	DA
Verificare Telefon	NU	DA
Verificare IP	NU	DA
Verificare Email	DA	DA

Tabel 3.3 Comparaţie între EWebsiteAuction şi Flippa

Flippa, unul dintre cele mai mari site-uri din domeniu, este foarte bine organizat, are un design uşor de folosit, oferă o zonă de tranzacţionare a site-urilor şi implementează numeroase sisteme de verificare a utilizatorilor, cum ar fi: verificare email, verificare ip, verificare număr telefon. Unul dintre dezavantajele acestui site, este procesul de creare a licitaţiilor, care necesită foarte mulţi paşi până a putea fi activate, ceea ce poate fi mai greu de realizat pentru utilizatorii neexperimentaţi.

EWebsiteAuction oferă un compromis a elementelor de securitate şi a modului de folosire a site-ului. Astfel utilizatorii pot crea licitaţii foarte uşor, realizând doar câteva click-uri, iar site-ul este foarte uşor de folosit chiar şi de către un începător. În acelaşi timp există şi numeroase sisteme de securitate implementate, cum ar fi: adăugare automată date domeniu, verificare email, sistem de feedback şi rating.

În concluzie, pentru implementarea sistemului **EwebsiteAuction** s-a încercat atingerea principalele funcţionalităţi ale celor mai mari site-uri de acest tip, existente pe piaţă. Deşi nu s-au implementat toate funcţionalităţile oferite de toate cele 3 site-uri, proiectul dezvoltat asigura toate nevoile utilizatorilor, pentru a putea realiza toate procesele enumerate la inceputul capitolului.

4. Fundamentare Teoretică şi Analiză

În acest capitol se va realiza o prezentare a tuturor tehnologiilor folosite în dezvoltarea aplicaţiei, motivând alegerea fiecarei tehnologii în parte. De asemenea va fi realizată şi o analiză a sistemului, care va permite observarea modului de funcţionare a sistemul şi ce funcţionalităţi de bază ar trebui să ofere.

4.1. Fundamentare Teoretică

În continuare vor fi detaliate tehnologiile care au fost folosite pentru a implementa proiectul.

Aplicaţia a fost dezvoltată folosind atât tehnologii open source, cât şi tehnologii care au nevoie de anumite licenţe pentru a funcţiona. Toate aceste tehnologii sunt bine documentate, larg răspândite şi au în spate comunităţi de dezvoltatori preocupaţi de utilizarea şi dezvoltarea acestora.

Alegerea făcută prezintă următoarele avantaje:

- Cost redus privind dezvoltarea aplicaţiei, deoarece nu este necesară achitarea unei sume de bani pentru a fi folosite toate tehnologiile, deşi unele tool-uri necesita realizarea unui cont pentru a putea funcţiona normal
- Există suport din partea comunităţilor care susţin folosirea tehnologiilor, prin realizarea unor tutoriale şi prin implicarea în forumuri de discuţii pentru a ajuta alţi dezvoltatori care au nevoie.

4.1.1. Enterprise JavaBeans

Enterprise JavaBeans defineşte o arhitectură pentru dezvoltarea componentelor software tranzacţionale, a obiectelor distribuite bazate pe aplicaţii şi a componentelor care aparţin de server. Componentele care aparţin de server, se numesc Enterprise Beans, sunt obiecte distribuite care aparţin de containerele EJB şi permit clienţilor accesul de la distanţă la serviciile oferite.[1]

Mediile EJB pot să fie organizate pe 2 sau 3 nivele. În mediile client-server pe 2 nivele aplicaţiile vor fi cuplate îndeaproape cu cerinţele sistemului. În mod obişnuit aplicaţiile accesează serviciile bazei de date sau serviciile tranzacţionale direct din aplicaţia client. Aceste aplicaţii sunt numite de obicei "*clienţi graşi*", deoarece logica lor se afla în client, făcându-l foarte mare şi complex.

Aplicaţiile client-server pe 3 nivele realizează un server intermediar, care operează între aplicaţia client şi serviciile bazei de date. Nivelul de mijloc conţine logica de business a sistemului şi coordonează interacţiunile clientului cu baza de date. Câteva dintre beneficiile acestei organizări, ar fi scalabilitatea, performanţa mărita şi creşterea flexibilităţii şi extensibilităţii sistemului.[1]

Portabilitatea în EJB, funcţionează, deoarece specificaţiile definesc un set de contracte bine definite între container-ul EJB şi componentele EJB(obiectul de business). Aceste contracte sau reguli care definesc exact ce servicii trebuie să facă disponibile un container unui enterprise bean şi ce alte API-uri trebuie să creeze dezvoltatorul.

EJB simplifica dezvoltarea, lansarea şi accesul la obiecte distribuite. Dezvoltatorul unui obiect EJB, trebuie doar să implementeze obiectul conform convenţiei şi protocolului stabilit pentru Enterprise JavaBean.

Container-ul EJB

Bean-urile Enterprise reprezintă componenta software, care rulează într-un mediu special numit container EJB [1]. Containerul depozitează şi manageriaza un Eneterprise Bean în aceeaşi modalitate în care Java Web Server depozitează un servlet sau în care un browser depozitează un applet Java. Un Bean Enterprise nu poate funcţiona în exteriorul container-ului EJB, deoarece acesta manageriaza fiecare aspect al unui bean enterprise în momentul rulării, incluzând accesul la distanţă, securitatea, persistenţa, tranzacţiile, concurenţa şi accesul la resurse.

Container-ul izolează bean-ul enterprise de la accesul direct de către aplicaţiile client. În momentul în care o aplicaţie client invocă o metodă la distanţă, dintr-un Bean Enterprise, container-ul interceptează invocarea, pentru a asigura persistenta.

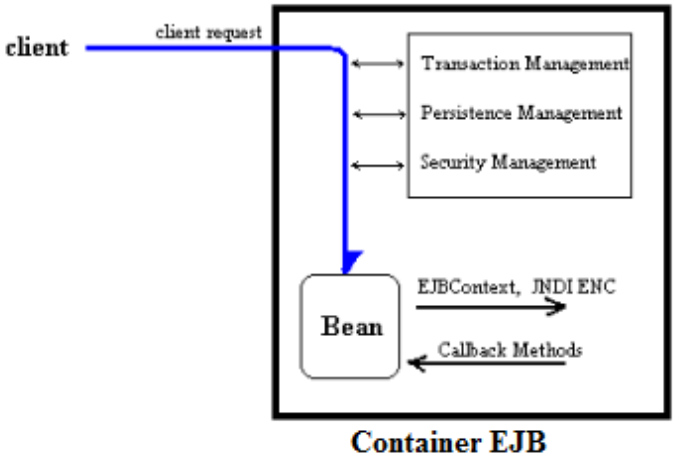

Figura 4.1 Container EJB (preluată din [1])

Bean-ul Enterprise

Pentru a crea o componentă EJB server-side, trebuie realizată o interfaţă care defineşte metodele de business ale bean-ului, plus implementarea acestuia. Clientul foloseşte interfaţa publică a bean-ului pentru a crea, manipula şi şterge bean-uri de pe serverul EJB. Clasa implementată, denumită şi clasa bean este instanţiată la run time şi devine un obiect distribuit. Un bean este o componentă server-side care reprezintă un concept de business precum un Client sau Manager.[1]

Bean-ul Entitate

Acesta este unul dintre cele două tipuri de bean existente: entitate şi sesiune. Bean-ul entitate este folosit pentru a reprezenta datele în baza de date. Pune la dispoziţie o interfaţă orientată pe obiect, pentru acces la date, care în mod normal ar fi accesate folosind JDBC sau alte API-uri. Bean-ul entitate furnizează un model de componentă care permită dezvoltatorilor să îşi concentreze atenţia asupra logicii de business, în timp ce container-ul se ocupă de managerierea persistenţei, tranzacţiilor şi controlului.

Există două tipuri de bază de bean entitate: container-managed persistence(CMP) şi bean-managed persistence(BMP). Cu ajutorul CMP, container-ul manageriază persistenţa bean-ului entitate, iar cu BMP bean-ul entitate conţine codul de acces la baza de date şi este responsabil

pentru scrierea şi citirea stării proprii în baza de date.

Bean-ul Sesiune

Este folosit pentru a manageria interacţiunile entităţii şi a altor bean-uri de sesiune, accesarea resurselor, şi pentru a realiza task-uri în numele clientului.

Există două tipuri de bean-uri de sesiune: stateless şi statefull. Bean-urile de sesiune stateless nu îşi menţin starea conversaţională cu clientul. Acestea îşi menţin starea conversaţională doar pe durata invocării metodei. Conform [3] bean-ul stateless ar trebui să fie folosit când:

- Starea bean-ului nu are date specifice despre client
- Bean-ul realizează informații generice pentru toţi clienţii printr-o singură invocare de metoda

Pentru Bean-ul statefull starea este salvată până când comunicarea dintre client şi bean se termină. Bean statefull ar trebui să fie folosit când[3]:

- Starea bean-ului reprezintă interacţiunea dintre bean şi un client specific
- Bean-ul trebuie să păstreze informaţii despre client pe durata a mai multor invocări de metode
- Bean-ul mediază între client şi alte componente ale aplicaţiei
- Bean-ul manageriază modul de desfăşurare a mai multor Enterprise Beans

Am decis să folosesc tehnologia EJB, deoarece:

- Securitatea este mărită datorită container-ului EJB
- Componentele se pot reutiliza. Enterprise Beans sunt portabile, dezvoltatorul aplicaţiei poate realiza aplicaţii noi folosind un set de bean-uri existente
- Separa logica de business de prezentare.

4.1.2. JavaServer Pages

JavaServer Pages sunt similare cu fişierele HTML, dar furnizează abilitatea de a afişa conţinut dinamic în interiorul paginii. Tehnologia JSP a fost dezvoltată de către Sun Microsystems, pentru a separa dezvoltarea conţinutului dinamic al paginii de design-ul static al paginilor HTML. Rezultatul acestei separări înseamnă că design-ul paginii se poate modifica, fără a fi nevoie să fie schimbat conţinutul dinamic al paginii. Acest lucru este util în timpul ciclului de dezvoltare, deoarece designerii paginilor web, nu trebuie să ştie cum să creeze conţinutul dinamic, ci doar să ştie unde să-l plaseze în cadrul paginii.[8]

Pentru a facilita integrarea conţinutului dinamic, JSP foloseşte un număr de tag-uri, care ajută designerul paginii să insereze proprietăţile unui obiect java şi elementele de script într-un fişier JSP.

Câteva dintre avantajele folosirii tehnologiei JSP, faţă de alte metode de creare a conţinutului dinamic sunt[8]:

- **Separarea conţinutului dinamic şi static.** Acest lucru permite separarea logicii aplicaţiei de design-ul paginii, reducând astfel complexitatea dezvoltării site-urilor şi făcând site-ul mai uşor de întreţinut.
- **Independenta faţă de platformă.** Deoarece tehnologia JSP este Java-based, aceasta este independentă de platformă. JSP poate rula pe aproape orice server de aplicaţii web, poate fi dezvoltat pe orice platformă şi vizualizat pe orice browser.
- **Tag-uri şi Script-uri.** JSP suporta atât cod Java Script cât şi tag-uri. JavaScript este

folosit de obicei, pentru a adăuga noi funcţionalităţi la nivel de pagină, iar tag-urile permit o integrare şi modificare mai uşoară a proprietăţilor JavaBean.

Funcţionarea Paginilor JSP

JavaServer Pages au devenit operabile, prin translatarea conţinutului lor, într-un servlet, de către server-ul aplicaţie[8]. Acest proces este responsabil pentru translatarea într-un servlet Java, care livrează browser-ului web, codul translatat cu ajutorul serverului, atât a elementelor statice, cât şi a celor dinamice

Deoarece JSP este o tehnologie server-side, procesarea elementelor statice şi dinamice a paginii se realizează în interiorul server-ului. Arhitectura unui site JSP este numită de multe ori "*client subţire*", deoarce logica de business este executată în mare parte pe server.

Următorul proces evidenţiază task-urile realizate în interiorul unei pagini JSP, la o primă invocare a fişierului, sau când fişierul este modificat[8]:

- Browser-ul face o cerere la pagina JSP
- Motorul JSP parsează conţinutul paginii JSP
- Motorul JSP crează un servlet temporar, responsabil pentru randarea elementelor statice
- Codul sursă al servlet-ului este compilat de către compilatorul Java într-un fişier servlet
- Servlet-ul este instanţiat
- Combinaţia elementelor statice şi dinamice este trimisă către browser-ul web

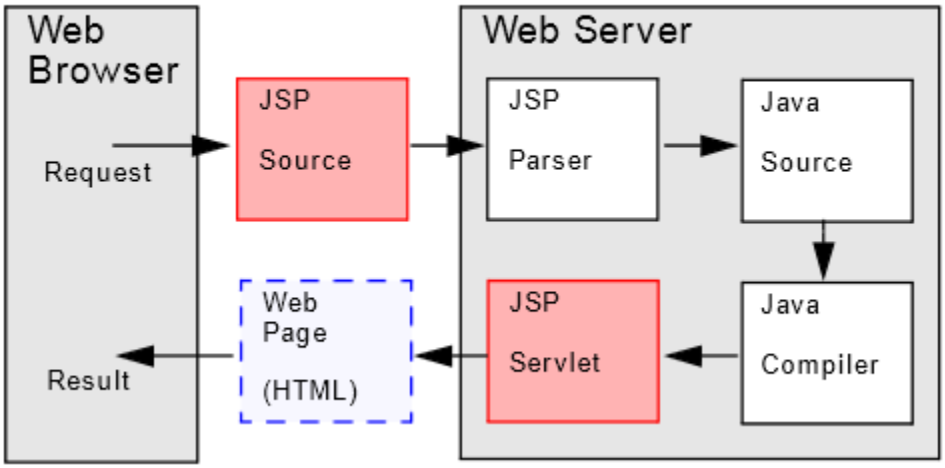

Figura 4.2 Ciclul de viaţă al paginii JSP la prima invocare(preluată din [8])

Am decis să folosesc tehnologia JSP, deoarece permite generarea conţinutului dinamic şi o bună separare a codului de business de interfaţă utilizatorului. Un alt motiv pentru alegerea acestei tehnologii este datorat rapidităţii cu care se pot crea noi pagini JSP.

4.1.3. Spring MVC

Framework-ul web Spring MVC este realizat în jurul unui DispatcherServlet, care trimite cereri claselor care se ocupă cu manipularea datelor. Clasele default pot fi interfeţe Controller foarte simple, care oferă o metodă ModelandView.

Acest framework oferă următoarele funcţionalităţi[9]:

- **Separarea clară a rolurilor**. Fiecare rol(controller, validator, obiect model, obiect comanda) poate fi îndeplinit de un obiect specializat
- **Adaptabilitate**. Se poate folosi orice subclasă controler, pentru un anumit scenariu, în locul derivării dintr-un singur controler toate funcţionalităţile
- **Refolosirea codului de business**. Se pot refolosi obiecte de business existente, ca şi comandă în locul oglindirii lor, pentru a putea extinde un anumit framework
- **Transfer flexibil al modelului**. Transferul modelului prin nume/valoare face posibilă o integrare uşoară cu orice tehnologie pentru vedere.
- **Librărie de tag-uri JSP**. Conţine o librărie simplă şi puternică de tag-uri JSP, denumită şi tag-uri Spring, care furnizează suport pentru funcţionalităţi ca legarea datelor. Tag-urile customizate permit o flexibilitate maximă.

DispatcherServlet

Framework-ul web MVC, este concentrat pe cerere, fiind realizat în jurul unui servlet central care trimite cereri controller-elor şi oferă alte funcţionalităţi care facilitează dezvoltarea aplicaţiilor web.[9]

În figură de mai jos este prezentat fluxul de lucru al procesării cererilor, de către Spring MVC.

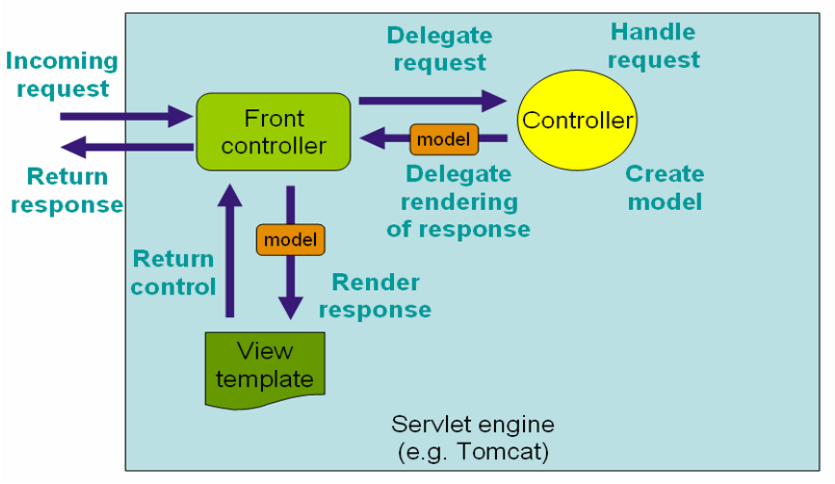

Figura 4.3 Fluxul de lucru al procesării cererilor de către MVC (prelută din [9])

DispatcherServlet este un servlet şi de accea este declarat în fişierul web.xml al aplicaţiei. Cererile care se doresc a fi manipulate de către DispatcherServlet vor trebui să fie mapate, folosind o mapare URL în acelaşi fişier web.xml.

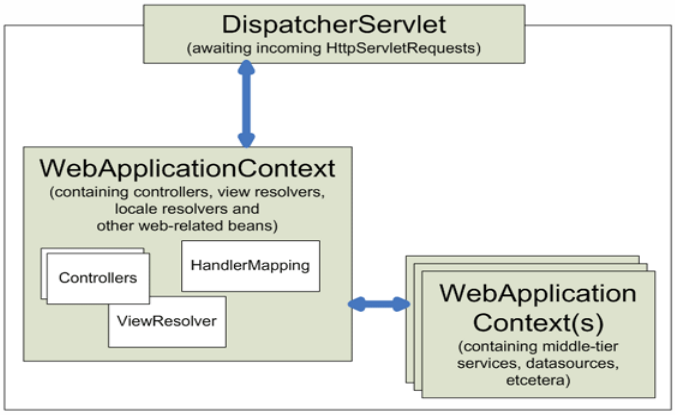

Figura 4.4 Ierarhia context Spring MVC (preluată din [9])

Am folosit framework-ul Spring MVC, deoarece este unul dintre cele mai populare framework-uri de organizare a aplicaţiei pe 3 nivele şi datorită tuturor avantajelor enumerate mai sus.

4.1.4. Framework-ul TopLink

Oracle TopLink este un framework avansat de persistare şi transformare a obiectelor, care furnizează dezvoltatorilor unelte şi capabilităţi run-time, menite să reducă eforturile de dezvoltare şi întreţinere. Acesta construieşte aplicaţii cu performanţă ridicată, care memorează obiecte persistente într-o bază de date relaţională. Reuşeşte să transforme cu succes obiecte orientate pe date în date relaţionale sau elemente XML[10].

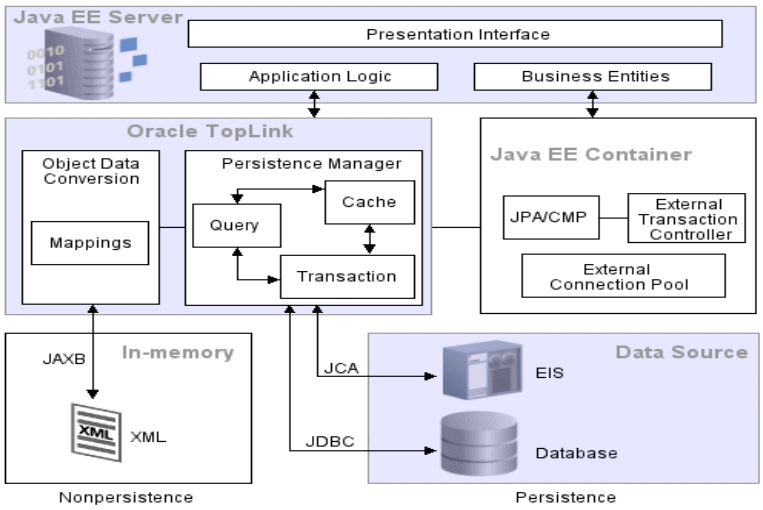

Figura 4.5 Arhitectura TopLink(preluată din [10])

TopLink furnizează un larg set de unelte, care pot fi folosite pentru a crea rapid, aplicaţii de o performanţă ridicată, care sunt atât scalabile cât şi uşor de întreţinut. Acest framework permite să se realizeze următoarele lucruri[10]:

- Persistă obiecte java la orice bază de date relaţională suportată de un driver JDBC
- Persistă obiecte java către orice sursă de date nonrelatională, cum ar fi XML

- Realizează conversii în memorie, între obiecte java şi scheme XML(XSD)
- Mapează orice model de obiecte, la orice schemă relaţională sau nonrelatională

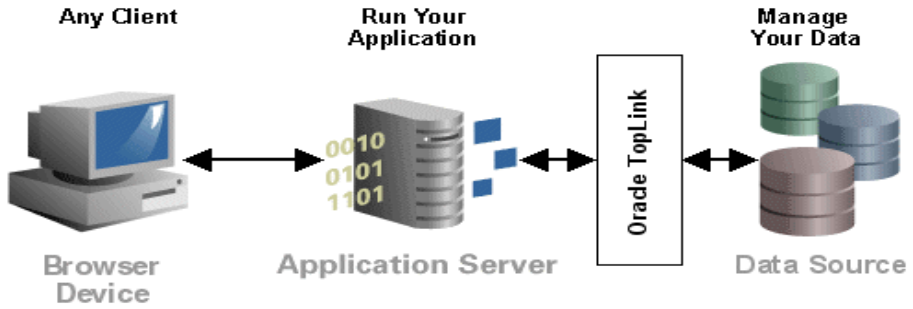

Figura 4.6 Arhitectura unei aplicaţiei care foloseşte TopLink(preluată din [10])

Am decis să folosesc TopLink, deoarece acesta este framework-ul de persistentă ideal, datorită folosirii mai multor produse software produse de Oracle. În acest fel am reuşit să obţin o omogenitate a aplicaţiei din privinţa produselor software folosite.

4.1.5. Alexa Rank API

Alexa oferă o platformă pentru crearea soluţiilor şi serviciilor web, folosind vastele informaţii despre site-uri, ale Alexa, accesibile printr-un api.

Cererea de interogare este o simplă cerere HTTP GET, care returnează un XML cu datele dorite. Cu ajutorul acestui API, putem obţine o clasificare a site-urilor în funcţie de traficul pe care l-au avut în ultimele 3 luni. Site-ul cu cea mai mare combinaţie de vizitatori unici şi număr de vizualizări va ocupa locul 1[11].

Adresa de interogare are următoare structura: *dată.alexa.com/dată?cli=10&url=domainName*, unde valoarea parametrului *cli*, reprezintă informaţiile care să fie returnate, iar valoarea parametrului *url* reprezintă adresa site-ului care se doreşte a fi interogat.

Apelul acestui API, va returna un XML, care va conţine, printre altele poziţia site-ului interogat.

```
- <ALEXA VER="0.9" URL="ironmovies.com/" HOME="0" AID="=" IDN="ironmovies.com/">
  - <SD>
      <POPULARITY URL="ironmovies.com/" TEXT="962463" SOURCE="panel"/>
      <REACH RANK="956711"/>
      <COUNTRY CODE="IN" NAME="India" RANK="113333"/>
    </SD>
  </ALEXA>
```

Figura 4.7 Rezultatul apelului Alexa API

Am decis să utilizez API-ul furnizat de ALEXA, deoarece aceasta reprezintă singura modalitate pentru a obţine datele statistice dorite, despre un domeniu. Rank-ul furnizat de ALEXA reprezintă cel mai bun etalon pentru a măsură importanţa unui site.

4.1.6. Google API

PageRank-ul reprezintă o familie de algoritmi, pentru asignarea unei ponderi numerice paginilor web, indexate de un motor de căutare. Sistemul de PageRank este folosit de popularul motor de căutare Google, pentru a determina relevența şi importanţa unei pagini web.

Cu ajutorul acestui API, aplicaţiile pot obţine valoare Google PR, pentru orice domeniu, din unul din datacenter-ele folosite de Google şi care suportă cereri la adresa: "*toolbarqueries.google.com*".

Apelul API-ului se face, cu ajutorul unei cereri HTTP GET, către următoarea adresă: "*http://toolbarqueries.google.com/tbr?client=navclient-auto&hl=en&ch=6236440745*

&ie=UTF-8&oe=UTF-8&features=Rank&q=info:google.com".

Partea mai complicată o reprezintă parametrul *ch*, care foloseşte algoritmul de hashing Bob Jenkins, pentru a cripta adresa URL şi a obţine valoarea.

Rezultatul apelului va fi un string de formă:" *Rank_1:1:9"* , unde PageRank-ul domeniului va fi 9.

Funcţiile de hash Jenkins reprezintă o colecţie de funcţii hash non-criptografice, pentru chei multi-byte, realizate de către Bob Jenkins. Acestea pot fi folosite şi ca checksums, pentru a detecta coruperea accidentală a datelor, sau pentru a detecta înregistrările identice din baza de date[12].

Am decis să folosesc acest API, deoarece aceasta era singura modalitate de a obţine PR unui domeniu din data center-ele folosite de Google. Am dorit obţinerea acestei informaţii despre un domeniu, deoarece acesta este cel mai important factor de care se ţine cont pentru a stabili ierarhia rezultatelor din motorul de căutare Google.

4.1.7. PayPal Data Transfer API

Payment Data Transfer (PDT) furnizează comercianţilor abilitatea de a afişa detaliile tranzacţiei, cumpărătorilor, care vor fi redirecţionaţi spre site după finalizarea plăţii.

Pentru a putea folosi serviciul PDT, acesta trebuie activat mai întâi din contul de paypal. După activare, de fiecare dată când un cumpărător realizează o plată de pe site şi este redirecţionat înapoi spre site, un token al tranzacţiei va fi păsat ca o variabilă GET, la URL-ul returnat. Pentru a folosi în condiţii bune acest API, trebuie captat token-ul tranzacţiei, variabila "*tx*" şi preluarea detaliilor tranzacţiei de la Paypal, prin construcţia unui HTTP Post[13].

Apelul metodei POST trebuie să fie trimis la adresa: "*https://www.paypal.com/cgi-bin/webscr* " , iar acesta trebuie să conţină variabila *cmd*, cu valoarea: *_notify-synch*. Pe lângă datele obligatorii, mai putem trimite adresa de email a cumpărătorului, folosind variabilă: "*payer_email*", numele produsului, cantitatea şi preţul. În afară variabilelor predefinite putem folosi şi variabilă *custom*, pentru a trimite şi alte date care ne vor ajuta să procesăm tranzacţia.

Răspunsul primit de la PayPal va fi cu un singur cuvânt pe o linie, iar în mesajul răspunsului vom avea: "SUCCESS" sau "FAIL". Când răspunsul este "SUCCESS", restul mesajului va conţine detaliile tranzacţiei, câte unul pe linie, în formatul: cheie=valoare, unde cheie şi valoare sunt stringuri codificate în URL.

```
SUCCESS
first_name=Jane+Doe
last_name=Smith
payment_status=Completed
payer_email=janedoesmith%40hotmail.com
payment_gross=3.99
mc_currency=USD
custom=For+the+purchase+of+the+rare+book+Green+Eggs+%26+Ham
```

Figura 4.8 Mesaj cu răspunsul primit de la Paypal

În momentul în care un cumparator realizează o plată, acesta va fi redirecţionat înapoi spre site, după cum urmează: dacă tranzacţia să desfăşurat cu succes, atunci acesta va fi redirectat la pagina de succes a site-ului, iar dacă tranzacţia a fost anulată, sau nu a putut fi procesată, atunci cumpărătorul va fi redirecţionat spre o altă pagină a site-ului.[13]

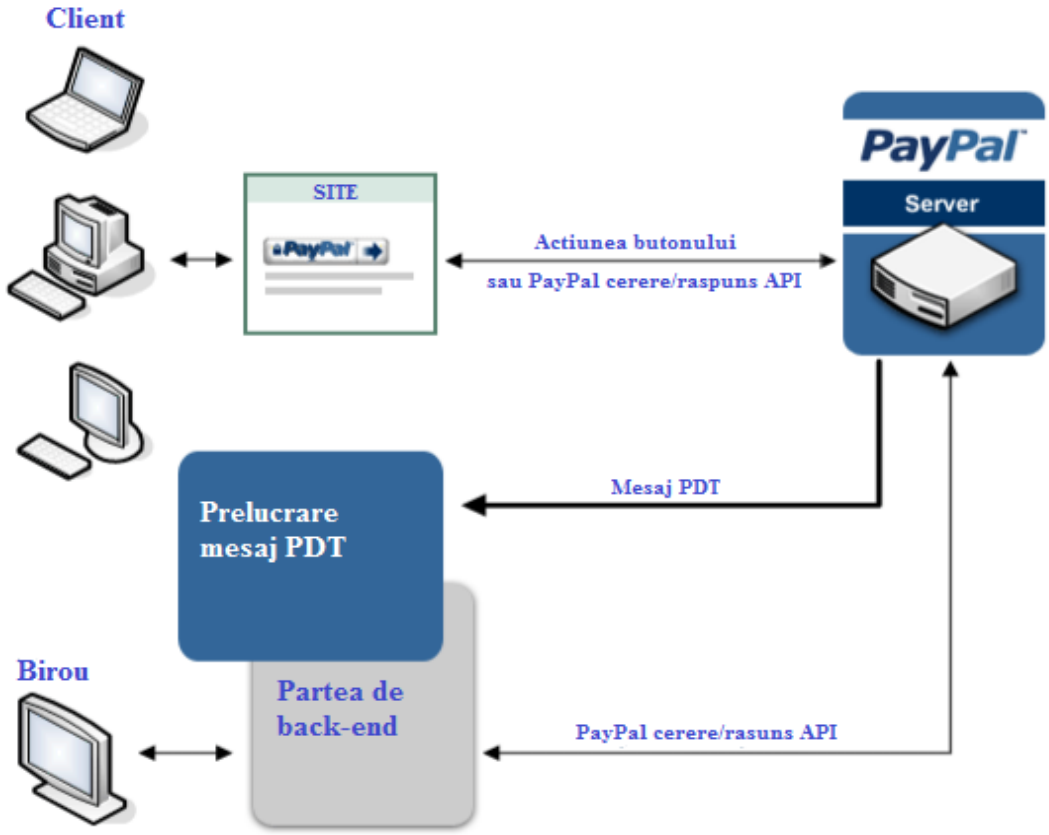

Figura 4.9 Model Funcţionare PDT

Am decis să folosesc API-ul furnizat de PayPal, deoarece acesta oferă posibilitatea de a redirecţiona clientul la anumite pagini în funcţie de modul în care s-a desfăşurat plata şi pentru uşurinţa cu care se poate customiza pentru a accesa diverse date necesare aplicaţiei.

4.1.8. WhoIS API

WhoIs este un protocol cerere şi răspuns, folosit de foarte multă lume pentru a interoga baze de date care stochează informaţiile despre resurse ale Internetului, cum ar fi: nume

domeniu, adresa ip, dată înregistrare şi expirare domeniu, informaţii utilizatori.

Protocolul WhoIs îşi are originile în protocolul Arpnet Nicname şi a fost implementat original pe Network Control Program, dar cel mai mult a fost utilizat în momentul în care suita TCP/IP a fost standardizata peste Arpanet şi apoi Internet.

Datele WhoIs pot fi memorate şi accesate conform oricăruia dintre modelele: „*thin*" sau „*thick*". Modelul „*thick*" presupune că un singur server WhoIs conţine toate informaţiile a tuturor persoanelor care au înregistrat un domeniu. În modelul „*thin*" serverul memorează doar numele altor servere WhoIs, care la rândul lor vor stoca datele despre persoanele care au înregistrat domenii.[14]

Pentru a obţine datele despre domeniu, am folosit serviciul WhoIs Api, pus la dispoziţie de site-ul *whoisxmlapi*. În urma cererii HTTP, care are ca atribut numele domeniului care se doreşte a fi interogat , API-ul returnează un fişier XML sau JSON bine format.


```
  <?xml version="1.0" encoding="utf-8" ?>
- <WhoisRecord>
    <createdDate>1995-01-18.</createdDate>
    <updatedDate>2009-01-19.</updatedDate>
    <expiresDate>2012-01-18.</expiresDate>
  - <registrant>
      <name>Domain Administrator</name>
      <organization>Yahoo! Inc.</organization>
      <street1>701 First Avenue</street1>
      <city>Sunnyvale</city>
      <state>CA</state>
      <postalCode>94089</postalCode>
      <country>US</country>
      <email>domainadmin@yahoo-inc.com</email>
      <telephone>+1.4083493300</telephone>
      <fax>+1.4083493300</fax>
      <rawText>Domain Administrator Yahoo! Inc. 701 Firs
      <unparsable>domainadmin@yahoo-inc.com Fax: +1.4
    </registrant>
```


Figura 4.10 Rezultatul unui apel al API-ului WhoIs

4.1.9. Thumbnail API

Thumbnail API, realizează o captură a site-ului, în timp real. În momentul în care un utilizator accesează o pagină care apelează acest API, este generată o captură a site-ului respectiv şi trimisă site-ului care a realizat apelul, spre a fi afişată.[15]

Link-ul de apelare are următoarea structura:" *api.webthumbnail.org?width=500&height=400&format=png&screen=1024&url=nume_domeniu*" , unde variabilă *url* reprezintă numele site-ului.

4.2. Analiza Sistemului

În acest subcapitol vor fi prezentate principalele funcţionalităţi care pot fi realizate de către utilizatori, principalele cerinţe nefuncţionale pe care ar trebui să le ofere sistemul, actorii sistemului şi principalele cazuri de utilizare care ar putea fi realizate de către aceştea.

4.2.1. Cerinţele Funcţionale ale sistemului

Aşa cum a fost specificat şi în primul capitol, principalul obiectiv al acestui proiect este acela de a realiza un sistem online de licitaţie pentru site-urilor web.

Interfaţa pe care o oferă sistemul, permite utilizatorilor să realizeze funcţionalităţi importante precum:

- **Creare Licitaţii**.

  Proiectul fiind o platformă de licitaţii online oferă utilizatorilor posibilitatea de a-şi vinde site-urile şi domeniilor prin intermediul unei licitaţii. Acesta îşi propune să conecteze vânzătorul cu potenţialii cumpărători. Pentru a crea o licitaţie, vânzătorul trebuie să ofere o descriere a site-ului, durata licitaţiei şi datele statistice ale site-ului. Pe lângă toate aceste lucruri, sistemul adaugă alte informaţii despre domeniu şi site, luate automat de pe Internet în momentul creării licitaţiei. Toate aceste informaţii au rolul de a uşura munca utilizatorilor în a lua o decizie şi pentru a le creşte încredere în platformă de licitaţii.
- **Plasare Bid**

  Utilizatorii pot să liciteze la orice licitaţie doresc, fără să existe anumite constrângeri. Aceştea trebuie să fie responsabili, iar pe baza datelor furnizate de vânzător şi de sistem, trebuie să ia decizia de a licita sau nu. Utilizatorii au afişat preţul curent al licitaţiei şi bid-ul minim pe care îl pot face, iar pe lângă acestea au şi opţiunea de cumpărare acum, pentru a cumpăra site-ul înainte de a se încheia licitaţia, asigurându-se astfel că vor fi câştigători.
- **Trimite Feedback şi Rating**

  În urma încheierii cu succes a unei licitaţii, membri implicaţi în tranzacţie au posibilitatea de a lăsa rating şi feedback unul altuia. Acest sistem este benefic deoarece permite o evaluare continuă a serviciilor oferite de vânzător, sau a comportamentului cumpărătorului chiar de către utilizatorii site-ului. Pentru că sistemul de feedback şi rating să fie credibil, acesta se poate lăsa doar o singură dată pentru o tranzacţie, iar după ce a fost trimis nu mai există posibilitate de modificare.
- **Tranzacţionarea site-ului**

  Această parte a platformei este utilă pentru a ajuta membri implicaţi într-o tranzacţie, în a transfera toate bunurile site-ului şi a explica modul de funcţionare şi de gestionare al site-ului vândut. Pagina de tranzacţionare permite realizarea plăţii site-ului, trimiterea de mesaje vânzătorului/cumpărătorului care vor fi afişate ca o conversaţie fiind astfel mai uşor de citit şi trimitere feedback, pentru a anunţa şi ceilalţi vizitatori despre modul în care s-a desfăşurat tranzacţia.
- **Căutare Avansată**

  Utilizatorii acestei platforme, pot realiza o căutare avansată a licitaţiilor existente, pentru a le afişa doar pe cele care respectă criteriile dorite de ei. Astfel ei pot realiza o căutare în care să specifice, dacă vor licitaţii active/terminate, licitaţii care au activată opţiunea de cumpăra acum, sau licitaţiile care se încadrează într-un anumit interval de profit, de vizitatori sau de impresii ale site-ului. Serviciul de căutare ţine cont de toate datele introduse, făcând astfel rezultatele afişate să fie cât mai filtrate şi uşor de gestionat.
- **Administrare site**

  Administratorii acestui site, pot să gestioneze foarte uşor utilizatorii şi licitaţiile cu ajutorul zonei de administrare. De acolo aceştea pot să baneze/debaneze utilizatori, să activeze/deactiveze licitaţii. Pe lângă aceste lucruri, pot de asemenea să fixeze preţul creării unei licitaţii, să seteze preţul îmbunătăţirilor aduse licitaţiei şi să seteze adresa de paypal în

care se vor face toate plăţile. Managerierea utilizatorilor şi licitaţiilor este importantă, pentru a opri utilizatorii care doresc doar să spam-eze site-ul, sau pentru a putea opri licitaţiile care nu sunt conforme cu politica platformei de licitaţii.

Pe lângă principalele servicii enumerate mai sus, utilizatorii sistemului mai pot realiza următoarele lucruri:

- **Înregistrare.** Utilizatorii îşi pot crea cont, pentru a putea beneficia de toate serviciile oferite de platformă.
- **Trimite Mesaje.** Utilizatorii au posibilitatea de a trimite şi primi mesaje de la alţi membri ai site-ului.
- **Urmăreşte Licitaţii.** Membrii site-ului pot urmări licitaţiile care li se par mai interesante, pentru a le putea găsi mai uşor.
- **Afişare Licitaţii Proprii.** Vânzătorii site-urilor, pot să vadă doar licitaţiile create de ei.
- **Acceptare Bid.** Creatorii licitaţiilor pot să accepte sau respingă bid-urile primite. Bid-urile neacceptate nu vor fi luate în considerare, până în momentul acceptării lor. Acest lucru permite o verificare a utilizatorilor, înainte de a participa la licitaţie
- **Trimite oferta.** În cazul în care există bid-uri, iar licitaţia nu a atins preţul rezervat, creatorul licitaţiei, poate trimite o nouă ofertă persoanei care a licitat cel mai mult. Persoana respectivă trebuie să accepte oferta, pentru că licitaţia să se considere terminată cu succes.

4.2.2. Cerinţe Non-Funcționale ale Sistemului

Arhitectura sistemului trebuie proiectată, astfel încât să asigure atât cerinţele funcţionale cât şi anumite constrângeri datorate indicatorilor de calitate, cum ar fi: fiabilitate, securitate, scalabilitate.

Indicatorii de calitate aparţin cerinţelor non-funcționale, iar prin intermediul acestora se verifică cât de bine au fost îndeplinite cerinţele funcţionale ale sistemului. Pentru ca cerinţele de calitate ale unui sistem să fie utile, acestea trebuie formulate într-un mod cât mai clar şi concret. Sistemul realizat respecta următoarele cerinţe non-funcționale:

- **Performanţă**

  Performanţa reprezintă o măsură care defineşte numărul de procesări pe care aplicaţia poate să îl realizeze şi durata necesară pentru a putea executa anumite cerinţe funcţionale cum ar fi: crearea unei licitaţii, trimiterea unui mesaj. Această cerinţă este influenţată şi de complexitatea interfeţei utilizator şi de numărul de clienţi care accesează site-ul la un moment dat. Performanţa sistemului va fi măsurată în capitolul Testare şi Evaluare.
- **Securitatea**

  Această cerinţă non-functionala specifică nivelul şi mecanismelor de securitate precum: autentificare, autorizare, care trebuie satisfăcute de către sistem.

  Sistemul, implementează cerinţa autentificării prin necesitatea utilizatorilor de a se identifica în cazul în care doresc să acceseze anumite funcţionalităţi.

  Autorizarea aplicaţiei, este asigurată prin existenţa a trei tipuri de roluri: utilizatori nelogati, utilizatori logaţi şi administratori.
- **Integritatea**

  Integritatea reprezintă o cerinţă care verifica cât de uşor poate fi încorporat sistemul într-un context de aplicaţii mai larg. De cele mai multe ori, valoarea unei aplicaţii poate fi mărită dacă datele produse de aplicaţie, pot fi folosite şi în alte moduri decât cele pentru care au fost prevăzute.

Integritatea a fost realizată prin stocarea datelor într-o bază de date relaţionala, astfel încât să poată fi accesate şi de alte aplicaţii.

- **Utilizabilitatea**

  Cerinţa aceasta se referă la uşurinţa cu care utilizatorii pot folosi toate funcţionalităţile furnizate de sistem. Site-ul de licitaţii are o interfaţă uşor de folosit, care nu solicită utilizatorilor cunoştinţe despre tehnologiile folosite, ci doar să cunoască modul de funcţionare al unui site web obişnuit.

  Pentru a realiza această cerinţă, aplicaţia foloseşte meniuri, butoane standard şi mesaje de eroare, care vor ajuta utilizatorul, în cazul în care datele introduse nu sunt corecte.

- **Reutilizabilitate**

  Aplicaţia trebuie să ofere posibilitatea reutilizării anumitor componente, conform cu practicile de dezvoltare a sistemelor software. Reutilizabilitatea a fost realizată prin folosirea pattern-ului Model-View-Controller, care oferă o delimitare a logicii de business de modelul de date şi implementarea interfeţei cu utilizatorul.

4.2.3. Cazuri de utilizare

Pentru a înţelege modul de funcţionare al sistemului de licitare trebuie să identificăm actorii care interacţionează cu acesta şi cazurile de utilizare care pot fi realizate. În sistemul realizat putem observa 4 actori: utilizatorul nelogat, utilizatorul logat, creatorul de licitaţii şi administratorul de sistem.

Utilizatorii nelogati pot beneficia doar de o mică parte a serviciilor oferite de site şi anume, pot doar să caute anumite licitaţii şi să vadă detaliile acestora.

Utilizatorii logaţi moştenesc funcţionalităţile utilizatorilor nelogati şi se bucura de toate serviciile oferite de platformă, având astfel posibilitatea de a participa în procesul de licitare al site-urilor.

Creatorii de licitaţii moştenesc beneficiile utilizatorilor logaţi şi mai pot de asemenea să creeze şi managerieze propria licitaţie.

Administratorii de sistem, reprezintă persoanele care trebuie să asigure o bună funcţionare a organizării licitaţiilor, ei trebuie să supravegheze licitaţiile create şi comportamentul utilizatorilor.

Cazurile de utilizare identificate pentru utilizatorii nelogati:

1. Căutare licitaţii în arhiva de licitaţii existente
2. Vizualizarea tuturor detaliilor unei licitaţii

Figura 4.11 Diagrama Cazuri de Utilizare pentru Utilizator Nelogat

Cazurile de utilizare pentru utilizatorii logaţi:

3. Crearea unei noi licitaţii
4. Vizualizarea tuturor licitaţiilor proprii
5. Adăugarea unei licitaţii la licitaţiile urmărite

6. Plasare bid în cadrul unei licitaţii
7. Plasare comentariu în cadrul unei licitaţii
8. Plasare feedback în urma realizării unei tranzacţii
9. Cumpărare credite
10. Trimitere mesaje altor utilizatori
11. Acceptarea câştigării licitaţiei, în cazul în care preţul rezervat nu a fost atins

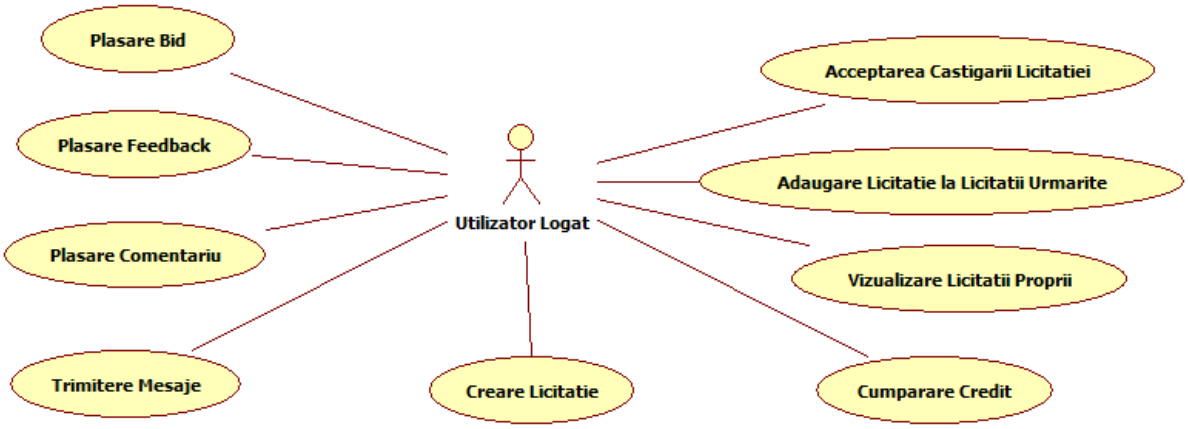

Figura 4.12 Diagrama Cazuri de Utilizare pentru Utilizator Logat

Cazuri de utilizare pentru creatorii de licitaţii:

12. Aprobă bid-uri primite în cadrul unei licitaţii
13. Respinge bid-uri primite în cadrul unei licitaţii
14. Desemnare câştigător în cazul în care preţul rezervat nu a fost atins

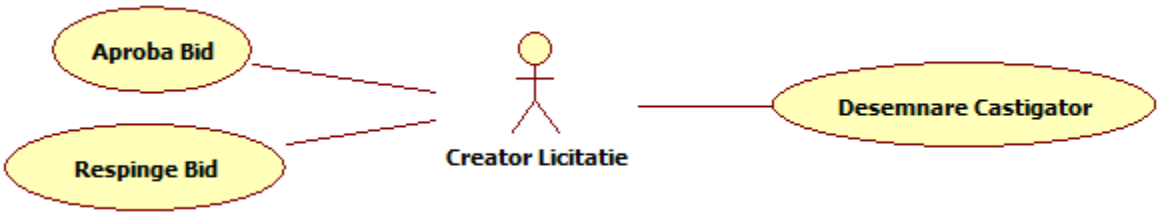

Figura 4.13 Diagrama Cazuri de Utilizare pentru Creatori Licitaţie

Cazuri de utilizare pentru administratori:

15. Operaţii CRUD asupra utilizatorilor
16. Operaţii CRUD asupra licitaţiilor
17. Realizare setări site

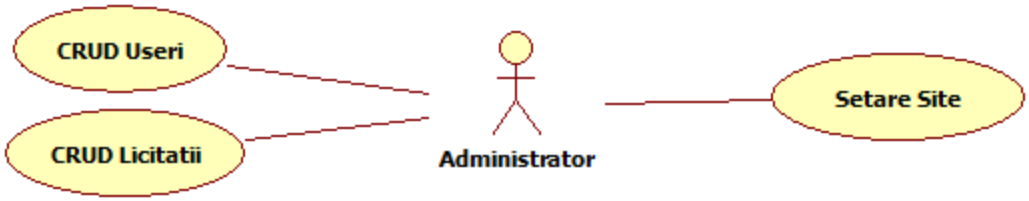

Figura 4.14 Diagrama Cazuri de Utilizare pentru Administrator

Cazuri de utilizare pentru sistem:

18. Porneşte licitaţia
19. Termină licitaţia
20. Trimite emailuri
21. Respingere automată a bid-urilor neacceptate, după o anumită perioadă
22. Preluare date domeniu
23. Validare cont

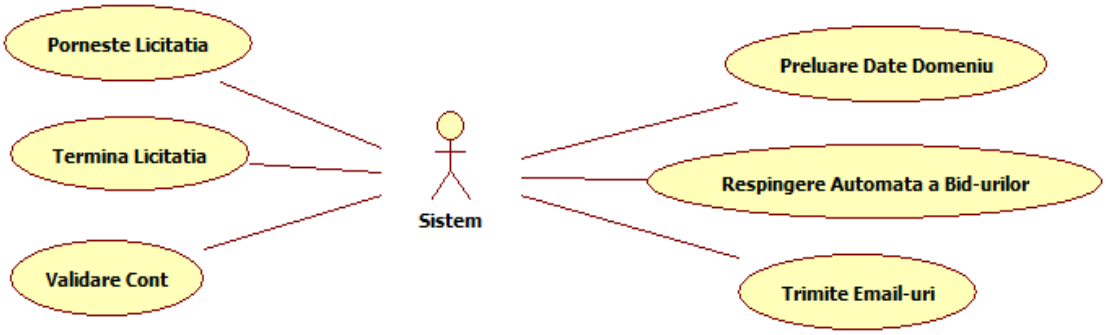

Figura 4.15 Diagrama Cazuri de Utilizare pentru Sistem

Caz de Utilizare: Logare Utilizator
Actorul Principal: Utilizatorul
Participanţii şi Interesele lor:

- Utilizatorul doreşte să se autentifice, pentru a putea realiza operaţiile care vor modifica starea sistemului.

Precondiţiile: Utilizatorul trebuie să acceseze pagina de login
Postcondiţiile: Utilizatorul este logat şi primeşte accesul la operaţiile dorite.
Scenariul Principal de Succes:

1. Utilizatorul accesează pagina de login a sistemului
2. Utilizatorul introduce datele cerute
3. Utilizatorul trimite datele spre server
4. Server-ul verifică datele primite.
5. Utilizatorul este autentificat.

Extensii posibile:

2a. Combinaţia de user şi parola introduse sunt invalide

1. Sistemul afişează un mesaj corespunzător

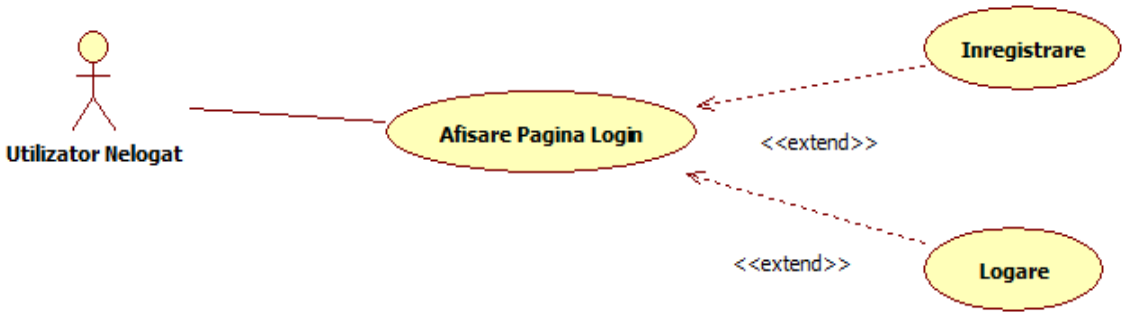

Figura 4.16 Diagrama cazurilor de utilizare pentru logare utilizator

Caz de Utilizare: Creare Licitaţie
Actorul Principal: Creator Licitaţie
Participanţii şi Interesele lor:
- Utilizatorul doreşte să creeze o licitaţie, pentru a vinde un site

Precondiţiile:
- Utilizatorul trebuie să fie autentificat
- Utilizatorul trebuie să acceseze pagina de creare a licitaţiilor

Postcondiţiile:
- O nouă licitaţie va fi programată pentru a începe

Scenariul Principal de Succes:

1. Actorul selectează pagina de creare a site-urilor
2. Sistemul îi afişează form-ul pentru realizarea licitaţiei
3. Actorul introduce detaliile site-ului
4. Actorul setează preţul de start
5. Actorul setează preţul rezervat
6. Actorul setează preţul de cumpărare acum
7. Actorul selectează dată de începere a licitaţiei şi durata
8. Actorul selectează îmbunătăţirile dorite
9. Actorul trimite datele la server
   9.1 Actorul plăteşte costurile licitaţiei
   9.2 Actorul trimite datele spre server
10. Server-ul crează o nouă licitaţie

Extensii posibile:

2-7a. Actorul decide să renunţe la crearea licitaţiei
   2. Sistemul rămâne nemodificat

8a. Nu sunt completate toate câmpurile
   1. Sistemul afişează un mesaj corespunzător. Licitaţia nu este creată

8b. Preţul rezervat este mai mic decât preţul de start
   1. Sistemul afişează un mesaj corespunzător. Licitaţia nu este creată

8c. Preţul de cumpăra acum este mai mic decât preţul de start
   1. Sistemul afişează un mesaj corespunzător. Licitaţia nu este creată

8d. Data începerii licitaţiei şi perioada sunt incorecte
   1. Sistemul afişează un mesaj corespunzător. Licitaţia nu este creată.

9.1a. Actorul nu plăteşte taxele licitaţiei
1. Licitaţia este creată, dar va fi dezactivată până în momentul plăţii

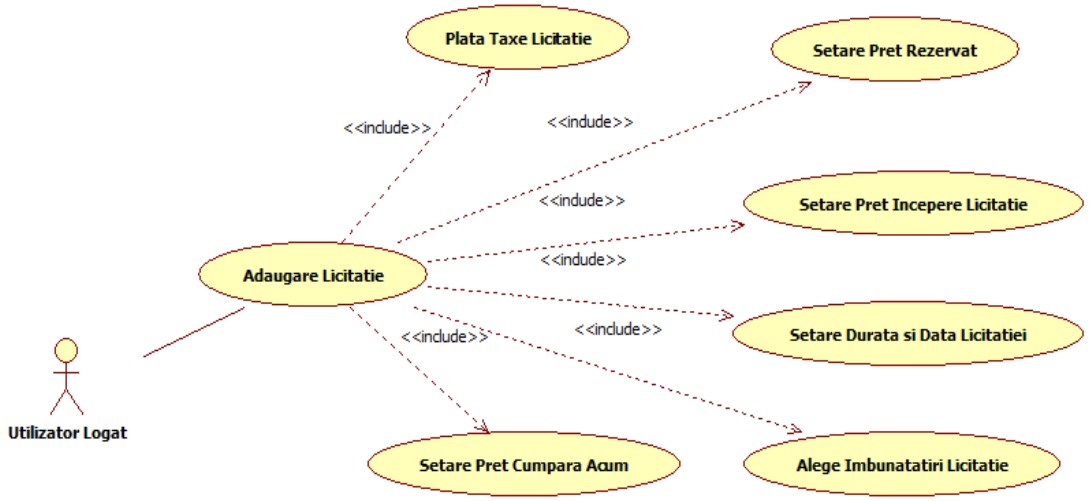

Figura 4.17 Diagrama cazurilor de utilizare pentru creare licitaţii

Caz de Utilizare: Afişare Licitaţii
Actorul Principal: Utilizator
Participanţii şi Interesele lor:
- Utilizatorul doreşte să vizualizeze licitaţiile existente în arhiva site-ului

Precondiţiile:
- Componentele site-ului trebuie să fie funcţionale
- Baza de date va conţine informaţii valide

Postcondiţiile:
- Utilizatorul va putea vizualiza licitaţiile în funcţie de anumite categorii: cele mai active, licitaţiile terminate cu succes, licitaţiile care nu au atins preţul rezervat

Scenariul Principal de Succes:

1. Utilizatorul navighează către modul de vizualizare al licitaţiilor
2. Aplicaţia încarcă din memorie licitaţiile existente
3. Aplicaţia lansează în ecranul principal o listă cu toate licitaţiile din categoria selectată, pentru fiecare categorie sunt afişate informaţii precum:
    - Titlu
    - Timp Rămas
    - Bid-uri
    - Număr Vizualizări
    - Preţ
4. Utilizatorul alege categoriile de licitaţii pe care vrea să le vadă
    1.1. Utilizatorul selectează licitaţiile active
    1.2. Utilizatorul selectează licitaţiile cele mai active

1.3.Utilizatorul selectează licitaţiile terminate carea au atins preţul rezervat
1.4.Utilizatorul selectează licitaţiile care nu au atins preţul rezervat
1.5.Utilizatorul selectează licitaţiile premium

Extensii posibile:

1 - 4a. Serverul un funcţionează corect

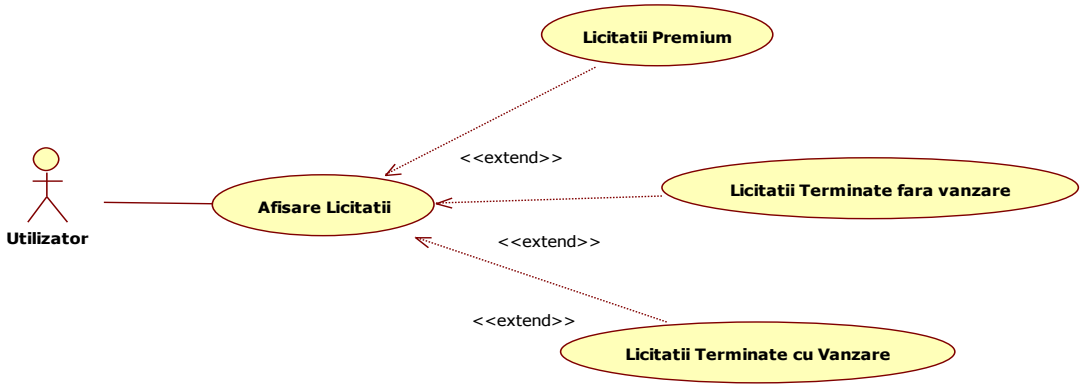

Figura 4.18 Diagrama cazurilor de utilizare pentru afişare licitaţii

Caz de Utilizare: Urmăreşte Licitaţii
Actorul Principal: Utilizatorul
Participanţii şi Interesele lor:

- Utilizatorul doreşte să urmărească o licitaţie, pentru a o putea găsi mai uşor

Precondiţiile:

- Utilizatorul trebuie să se logeze

Postcondiţiile:

- Licitaţia dorită va apărea în zona licitaţiilor urmărite de către utilizator

Scenariul Principal de Succes:

1. Utilizatorul se autentifică
2. Utilizatorul merge la licitaţia dorită
3. Utilizatorul apasă butonul "*Urmăreşte Licitaţia*"
4. Server-ul procesează cererea
5. Licitaţii este adăugată la licitaţiile urmărite de către utilizator

Extensii posibile:

1a. Autentificarea nu a fost realizată cu succes
1a1. Sistemul afişează un mesaj corespunzător
1a2. Utilizatorul încearcă din nou
4a. Serverul nu poate procesa cererea. Sistemul rămâne neschimbat

Caz de Utilizare: Gestionare Licitaţie
Actorul Principal: Creatorul unei licitaţii
Participanţii şi Interesele lor:

- Actorul poate să gestioneze licitaţia, prin acceptarea/respingerea bid-urilor şi prin plasarea unei oferte persoanei care a licitat cel mai mult, în cazul în care preţul rezervat

nu a fost atins

Precondiţiile:

- Actorul trebuie să fie logat în sistem
- Actorul trebuie să aibă o licitaţie activă

Postcondiţiile:

- Bid-urile vor fi acceptate/respinse în funcţie de dorinţa actorului

Scenariul Principal de Succes:

1. Actorul s-a autentificat cu succes în sistem
2. Actorul a creat cu succes o licitaţie
3. Actorul gestionează bid-urile primite
   3.1. Actorul accepta bid-ul primit
   3.2. Actorul respinge bid-ul primit
4. În cazul în care licitaţia nu atinge preţul rezervat actorul trimite oferta
5. Actorul tranzacţionează site-ul

Extensii posibile:

1a. Actorul nu a reuşit să se autentifice
   1a1. Sistemul afişează un mesaj corespunzător
   1a2. Actorul încearcă din nou mai târziu

2a. Actorul nu reuşeşte să creeze o licitaţie
   2a1. Sistemul afişează un mesaj corespunzător
   2a2. Actorul încearcă din nou mai târziu

3-5a. Actorul nu reuşeşte să realizeze comanda dorită
   3-5a1. Sistemul afişează un mesaj corespunzător

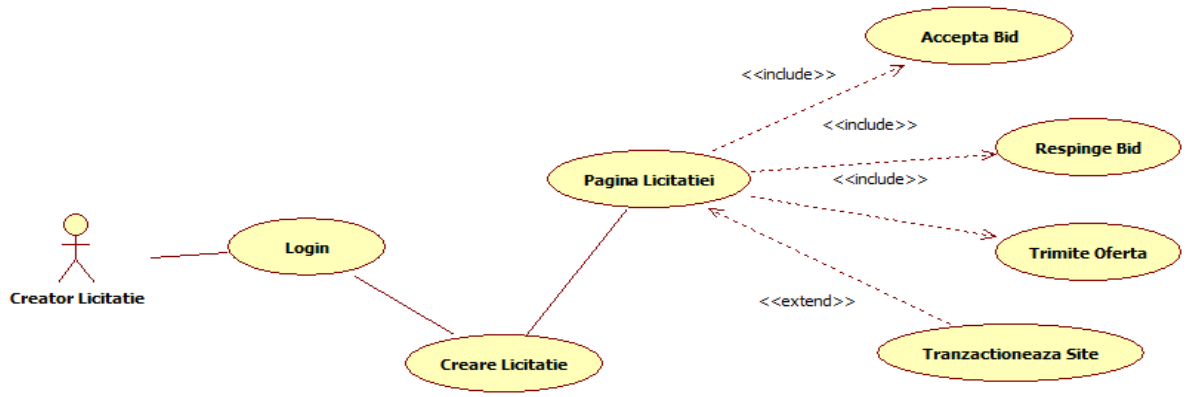

Figura 4.19 Diagrama cazurilor de utilizare pentru a manageria o licitaţie

Caz de Utilizare: Administrare Site

Actorul Principal: Administrator

Participanţii şi Interesele lor:

- Administratorul poate să realizeze operaţii CRUD asupra utilizatorilor şi licitaţiilor şi să facă anumite setări sistemului de licitaţii.

Precondiţiile:

- Actorul trebuie să fie logat în zona de administrator

Postcondiţiile:

- Setările realizate de administrator vor fi salvate în sistem

Scenariul Principal de Succes:

1. Actorul s-a autentificat în zona de administrator
2. Actorul merge la zona de administrare a site-ului

Extensii posibile:

2a. Actorul doreşte să acceseze zona de administrare a clienţilor

Precondiţii:

- Actorul se afla în modul de vizualizare al clienţilor

Scenariul Principal de Succes:

2aa. Aplicaţia încarcă utilizatorii din memorie
2ab. Actorul va realiza operaţii asupra utilizatorilor

2b. Actorul doreşte să acceseze zona de administrare a licitaţiilor

Precondiţii:

- Actorul se afla în modul de vizualizare a licitaţiilor

Scenariul Principal de Succes:

2ba. Aplicaţia încarcă licitaţiile din memorie
2bb. Actorul va realiza operaţii asupra licitaţiilor

2c. Actorul doreşte să acceseze zona de setare sistem licitaţii

Precondiţii:

- Actorul se afla în zona de setare site

Scenariul principal de succes:

2ca. Actorul completează form-ul
2cb. Actorul trimite datele spre sistem
2cc. Sistemul realizează setările

5. Proiectare de detaliu şi implementare

Aşa cum am precizat şi în capitolele anterioare, sistemul **EWebsiteAuction** îşi propune dezvoltarea unei platforme de licitare online a site-urilor. Sistemul este bazat pe o arhitectură client-server pe 3 nivele, folosind pattern-ul ModelViewController.

În acest capitol voi prezenta modul de implementare al funcţionalităţilor, modul de organizare al bazei de date şi interacţiunea dintre componente.

5.1. Arhitectura Conceptuală

Aplicaţia realizată reprezintă un sistem distribuit de tip client-server, realizat pe 3 nivele care au o responsabilitate diferită în arhitectura de ansamblu. Funcţionalităţile elementelor de pe un nivel superior depind de funcţionalităţile componentelor aflate pe un nivel inferior.

Cele 3 nivele sunt următoarele:

- Prezentare
- Controller
- Model

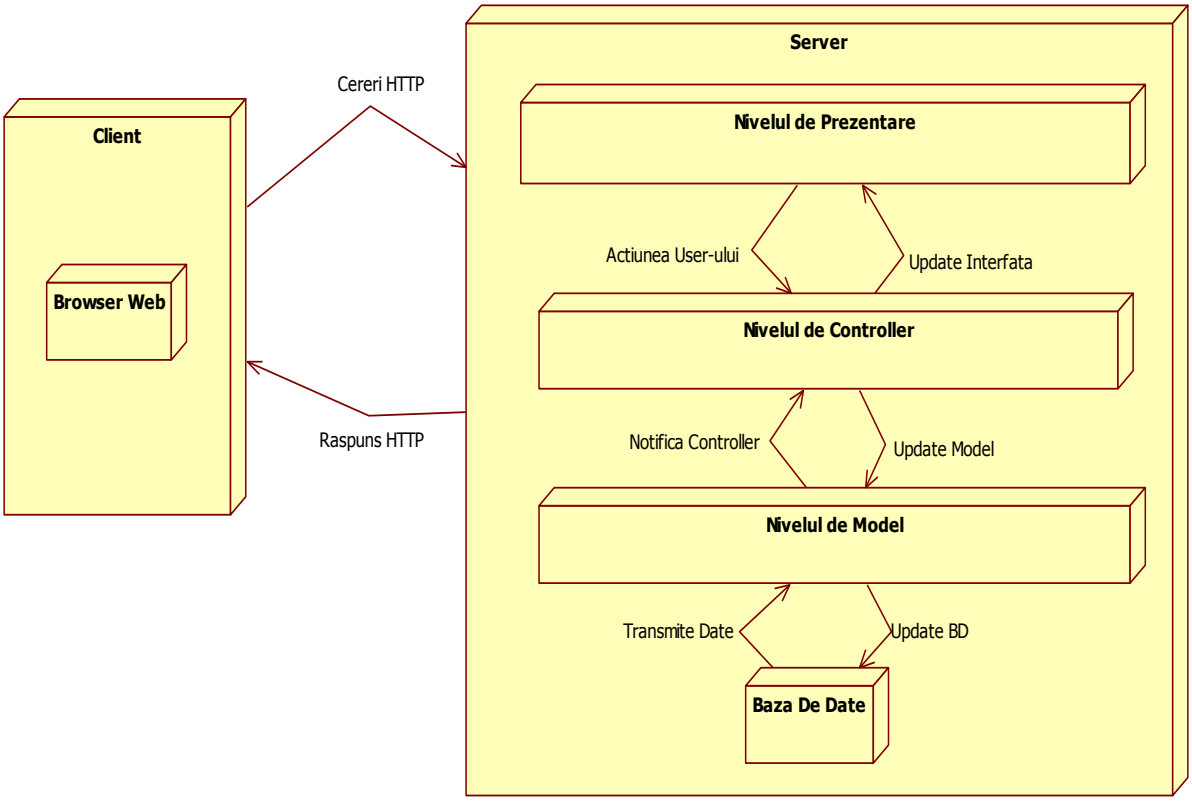

Figura 5.1 Diagrama Conceptuală

5.2. Nivelul de Prezentare

La acest nivel se află toate componentele responsabile de interacţiunea utilizatorului cu aplicaţia.

În cadrul aplicaţiei acest nivel a fost realizat folosind tehnologia JSP. Acest nivel conţine mai multe pagini JSP, care vor interacţiona cu clientul, pentru a putea fi realizate operaţiile dorite. Fiecare pagină va permite utilizatorului să realizeze anumite operaţii cum ar fi: creare licitaţie, trimitere mesaj, vizualizare licitaţie, etc.

Pentru funcţionarea nivelului de prezentare pentru fiecare pagină am mapat un controller folosind fişierul „dispatcher-servlet.xml", furnizat de framework-ul Spring MVC.

```
<bean id="urlMapping" class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">
    <property name="mappings">
        <props>
            <prop key="index.htm">indexController</prop>
            <prop key="UserLogin.htm">loginController</prop>
            <prop key="header.htm">headerController</prop>
            <prop key="Register.htm">registerController</prop>
            <prop key="confirm.htm">confirmController</prop>
            <prop key="forgot.htm">forgotController</prop>
            <prop key="logout.htm">logoutController</prop>
            <prop key="createauction.htm">createauctionController</prop>
            <prop key="newauction.htm">newauctionController</prop>
            <prop key="featuredauction.htm">featuredauctionController</prop>
            <prop key="mostactive.htm">mostactiveController</prop>
            <prop key="endingsoon.htm">endingsoonController</prop>
            <prop key="recentlysold.htm">recentlysoldController</prop>
            <prop key="unsold.htm">unsoldController</prop>
            <prop key="search.htm">searchController</prop>
            <prop key="listing.htm">listingController</prop>
            <prop key="approve.htm">approveController</prop>
            <prop key="reject.htm">rejectController</prop>
            <prop key="addwatch.htm">addwatchController</prop>
            <prop key="removewatch.htm">removewatchController</prop>
            <prop key="trimitemesaj.htm">trimitemesajController</prop>
            <prop key="mesajetrimise.htm">mesajetrimiseController</prop>
            <prop key="mesajeprimite.htm">mesajeprimiteController</prop>
            <prop key="user.htm">userController</prop>
            <prop key="watch.htm">watchController</prop>
            <prop key="adminlogout.htm">adminlogoutController</prop>
        </props>

    </property>
</bean>
```

Figura 5.2 Mapare URL la Controller

Fiecare controller asignat unei pagini foloseşte „ParameterizableViewController" şi de aceea trebuie să definim o mapare explicită a acestora, în fişierul „dispatcher-servlet.xml".

În fişierul „web.xml" am configurat pagina de start a aplicaţiei, durata sesiunii până la expirare şi extensia paginilor JSP, cu care vor putea fi accesate paginile.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/applicationContext.xml</param-value>
    </context-param>
    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>
    <servlet>
        <servlet-name>dispatcher</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <load-on-startup>2</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>dispatcher</servlet-name>
        <url-pattern>*.htm</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
    <welcome-file-list>
        <welcome-file>redirect.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

Figura 5.3 Setări Fişier web.xml

În momentul creării unui nou cont, utilizatorul este nevoit să-şi introducă adresa de email, un nume de utilizator, parola şi un număr valid de telefon. În cazul în care adresa de email sau numărul de telefon nu au un format valid, sau username-ul nu are lungimea necesară, sau parola de verificare nu se potriveşte, voi salva în sesiune problema apărută, iar apoi o voi afişa utilizatorului. După momentul creării contului, pentru a se putea loga în sistem utilizatorii trebuie să-şi verifice adresa de email, pentru a dovedi că datele furnizate sunt corecte.

Pentru a restricţiona accesul utilizatorilor, am salvat user-ul acestora în sesiune în momentul logării, iar în momentul în care accesează o pagină care necesită ca utilizatorul să fie autentificat am verificat datele salvate în sesiune. În cazul în care un utilizator nelogat, doreşte să acceseze o pagină, care necesită autentificare, atunci sistemul va afişă un mesaj corespunzător, care va conţine şi un link spre pagina de login.

Pentru a realiza header-ul, footer-ul şi meniul site-ului am decis să creez fişiere JSP care vor fi folosite ca un template. Astfel am realizat fişierele *header.jsp, footer.jsp* şi *menu.jsp* care vor conţine cod care va fi comun pentru mai multe pagini. Pentru a folosi fişierele respective în alte pagini a fost nevoie să le import folosind structura:

```jsp
<jsp:include page="/WEB-INF/jsp/header.jsp"/>
 <jsp:include page="/WEB-INF/jsp/menu.jsp"/>
<jsp:include page="/WEB-INF/jsp/footer.jsp"/>
```

Realizând această structură am reuşit să reutilizez codul respectiv de ori câte ori am avut nevoie, fără a fi nevoie să îl rescriu de fiecare dată, ci doar să import fişierele necesare.

Pentru a integra plată prin *PayPal* am creat un form, cu ajutorul căruia am trimis datele necesare tranzacţiei spre *PayPal*, cu ajutorul hidden fields.

Prin intermediul form-ului am specificat adresa de paypal, către care să se facă plată, obiectul pentru care se face plata, preţul produsului, moneda si pagina de redirecţionare în cazul în care plata se desfăşoară cu succes sau în cazul în care este anulată.

În cazul în care plata se desfăşoară cu succes, utilizatorul va fi redirecţionat spre pagina corespunzătoare, unde voi putea accesa datele primite de la Paypal, cum ar fi: numărul

produsului cumpărat, numele produsului, statusul plăţii, valoarea plătită şi adresa de email de pe care s-a plătit. Datele primite de la Paypal, vor fi trimise spre controller, pentru a putea fi procesată plată.

Pentru a porni şi opri licitaţiile, în momentele dorite de utilizator, apelez metodele ***startAuction()*** şi ***endAuction()*** de pe pagina de vizualizare a unei licitaţii şi de pe pagina de vizualizare a tuturor licitaţiilor. În acest mod sistemul va verifica de fiecare dată când un utilizator caută licitaţii, dacă nu sunt altele noi care ar trebui activate, sau dacă nu ar trebui oprite.

```
public void jspInit() {
    searC.startAuction();
    searC.endAuction();
  }
```

Pagina licitaţiei va putea fi accesată atât de utilizatorii logaţi cât şi de cei nelogati, dar în momentul în care aceştea vor dori să realizeze un bid sau comentariu, vor fi rugaţi să se logeze. Aceasta pagină este generate dinamic, în funcţie de id-ul licitaţiei, trimis prin intermediul link-ului de pe pagina vizualizării licitaţiilor. În momentul în care este accesata aceasta pagină, este citit username-ul utilizatorului din sesiune:
String username=(String)session.getAttribute("uşer_s");, iar apoi este citit id-ul licitaţiei din adresa URL: *auctionId=Integer.parseInt(request.getParameter("aucid"));*. Datele afişate pe pagină vor fi generate de către controllere, în funcţie de id-ul trimis ca parametru în momentul invocării metodelor necesare. În această pagină vor fi afişate datele licitaţiei, precum numărul de bid-uri primite, suma minimă care se poate licita, valoarea preţului rezervat, bid-urile primite până în momentul respectiv împreună cu username-ul utilizatorului şi data în care au fost realizate. Pe lângă aceste date, furnizate de utilizator, sau generate pe parcursul licitaţiei, vor fi afişate şi datele despre domeniu preluate de către sistem, în momentul creării licitaţiei.

La finalizarea unei licitaţii, în cazul în care aceasta se va termina cu succes, va fi afişat pe pagina licitaţiei respective, un mesaj care îndruma utilizatorii spre zona de tranzacţionare a site-ului. Acest mesaj va fi vizibil doar creatorului licitaţiei şi persoanei care a fost desemnată câştigător.

În cazul în care licitaţia s-a terminat, dar preţul rezervat nu a fost atins, vânzătorul va putea să facă prin intermediul paginii licitaţiei o ultimă ofertă, persoanei care a licitat cel mai mult. Persoana respectivă va trebui să accepte oferta, pentru a fi desemnată câştigătoare.

Zona de tranzacţionare a site-ului este o pagină care permite părţilor implicate să trimită mesaje, care vor putea fi vizionate sub forma unei conversaţii. Mesajele vor fi preluate din baza de date prin intermediul metodelor din controllere, care vor primi ca parametru, id-ul persoanelor implicate. Pe lângă acestea, tot din această zonă, se va putea trimite şi feedback. În momentul accesării paginii, dacă username-ul salvat în sesiune, vă fie echivalent cu username-ul utilizatorului câştigător, atunci acesta va putea lăsa feedback pentru vânzător şi să plătească site-ul achiziţionat, iar dacă username-ul este egal cu cel al vânzătorului, atunci acesta va putea lăsa feedback cumpărătorului.

Căutarea licitaţiilor se poate face prin selectarea opţiunilor dorite, şi prin introducerea intervalului dorit pentru numărul de vizitatori, impresii, preţ şi profit. În urma trimiterii datelor, form-ul va fi citit, iar valorile vor fi trimise spre controller, pentru a fi procesate. În urma căutării va fi afişat pentru fiecare site: titlul, timpul rămas, numărul de bid-uri, numărul de vizualizări şi preţul la care a ajuns licitaţia.

Zona de administrare a site-ului poate fi accesată doar de către administrator, după ce s-a logat cu succes în sistem. Această zonă este împărţită în:

- **Zona de administrare utilizatori**. Se pot bana/debana utilizatorii, încărcaţi din baza de date, prin intermediul controller-elor
- **Zona de administrare licitaţii**. Se pot activa/deactiva licitaţiile, încărcate din baza de date prin intermediul controller-elor
- **Zona de setare site**. Administratorul poate introduce adresa de paypal, care va fi folosită de sistem, costul creării unei licitaţii şi costul îmbunatăţirilor.

5.3. Nivelul de Controller

Face legătura între interfaţă şi domeniul aplicaţiei. Controlerul este responsabil de pregătirea datelor necesare spre a fi afişate în interfaţă şi de trimiterea datelor spre domeniu, pentru a fi salvate în baza de date.

Luând în considerare dimensiunea proiectului, s-au folosit mai multe controllere, pentru a nu avea o singură clasă controller, care într-un final va conţine un cod stufos şi greu de înţeles. În nivelul de controller am realizat operaţii precum: **crearea unei licitaţii**, **pornirea unei licitaţii la momentul dorit de utilizator**, **oprirea licitaţiei în momentul terminării acesteia**, **apelarea API-urilor folosite**, etc.

Descrierea principalelor metode

În momentul creării unei licitaţii, folosesc anumite atribute auxiliare precum: **payed**, **pending**, **finished** şi **hide** cu ajutorul cărora, voi decide dacă utilizatorul a plătit pentru a o crea, dacă licitaţia a fost banata, sau dacă licitaţia s-a termiat. Astfel, atributul **pending** va fi iniţializat cu 1, iar în momentul începerii licitaţiei va fi schimbat în 0, **payed** va fi iniţial 0, iar în momentul realizării platii va fi setat pe 1, iar **hide** va fi iniţializat cu 0 şi va fi schimbat în 1 doar în momentul în care administratorul decide să deactiveze licitaţia.

Pentru a activa licitaţiile create, am realizat metoda **startAuction()** în care verific dacă timpul de start al licitaţiei este mai mic decât timpul curent, dacă utilizatorul a plătit toate taxele necesare, dacă licitaţia nu a fost deactivata de administrator şi dacă licitaţia nu este terminată. În cazul în care toate aceste condiţii vor fi valide voi seta parametrul **pending** pe 0 şi astfel licitaţia va fi activă.

Pentru a termina o licitaţie am realizat metoda **endAuction()**, unde verific dacă timpul de sfârşit al licitaţiei este mai mic decât timpul curent, dacă licitaţia nu a fost deactivată şi dacă licitaţia este activă. În cazul în care toate aceste condiţii vor fi adevărate voi seta **finished** pe 0, iar licitaţia se va termina.

Metodele **startAuction()** şi **endAuction()** vor fi invocate din **nivelul de prezentare** şi vor fi executate de ori câte ori un utilizator va accesa o anumită licitaţie sau pagina de vizualizare a tuturor licitaţiilor.

Pentru a obţine durata unei licitaţii, am creat metoda **getTime(long ms),** care primeşte ca parametru timpul în milisecunde, obţinut prin scăderea din timpul de terminare a licitaţiei timpul curent. Această metodă va conţine un vector, care va reprezenta durata licitaţiei exprimată în secunde, minute, ore şi zile.

Descrierea Principalelor Clase

În figura de mai jos voi prezenta diagrama de clase a pachetului “controller”. Clasele din acest pachet vor comunica cu clasele din nivelul de domeniu şi cu paginile JSP.

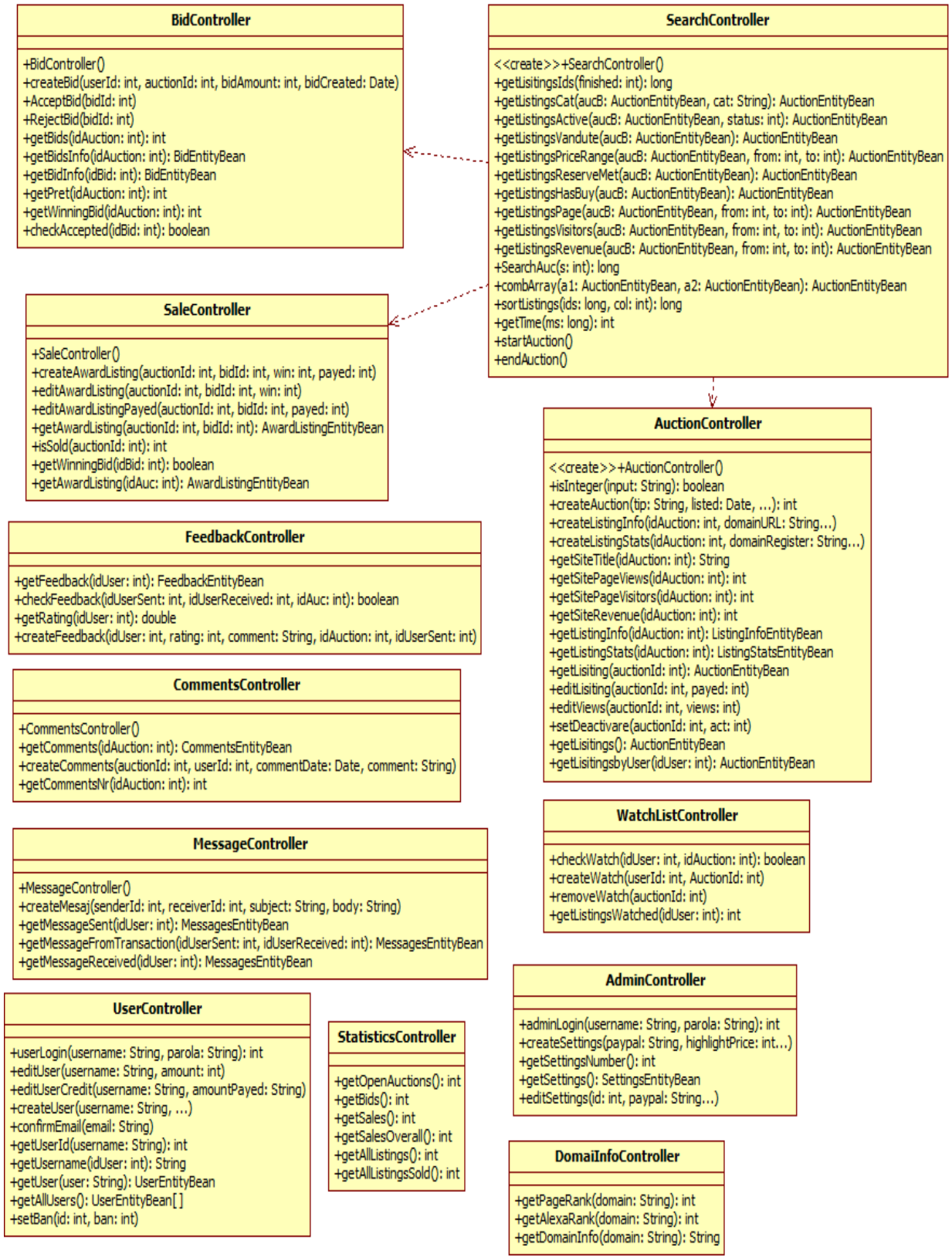

Figura 5.4 Diagrama de clase pentru pachetul Controller

Clasa **WatchListController**, este folosită de componentele aflate la nivelul de prezentare pentru a adăuga şi afişa licitaţiile urmărite de anumiţi utilizatori.

Principalele responsabilităţi ale acestei clase, sunt de a verifica dacă o anumită licitaţie, este pe lista celor urmărite de un anumit utilizator, de a adăuga noi licitaţii la cele urmărite, de a şterge o licitaţie de pe lista celor urmărite şi de a trimite lista acestora spre interfaţă, pentru a putea fi afişate.

Clasa **StatisticsController,** este folosită de componentele nivelului de prezentare pentru a calcula statistici precum: numărul de licitaţii deschise, numărul de bid-uri primite în ultimele 7 zile, valoarea vânzărilor realizate în ultimele 7 zile, procentajul licitaţiilor terminate cu succes şi valoarea totală a site-urilor vândute.

Principalele metode realizate sunt:

- **getOpenAuctions()**, care returnează un întreg cu numărul licitaţiilor active
- **getBids()**, returnează un întreg cu numărul de bid-uri din ultimele 7 zile
- **getSales()**, retunreaza un întreg cu vânzările realizate în ultimele 7 zile

Clasa **SearchController** este folosită pentru a procesa datele furnizate de la nivelul de prezentare pentru a realiza căutarea în arhiva de licitaţii.

Principala metoda a acestei clase este **SearchAuc(int[] s).** Aceasta primeşte ca parametru un vector de întregi, reprezentând valorile în funcţie de care să se realizeze căutarea şi va returna informaţiile licitaţiilor care respectă condiţiile necesare. În funcţie de valorile introduse, se vor invoca metodele care fac căutarea în funcţie de un anumit criteriu. În funcţie de criteriile după care se face căutarea, rezultatele metodelor invocate, vor fi concatenate sau vor fi transmise ca parametru altor metode de căutare. În final se va obţine o listă care va conţine rezultatele dorite, în funcţie de toate criteriile introduse de utilizator. Rezultatele returnate vor fi trimise nivelului de prezentare, unde vor fi afişate, pentru a fi vizualizate de către utilizator.

Clasa **SaleController,** este folosită de către nivelul de prezentare pentru a desemna câştigătorul licitaţiei.

Principala metoda a acestei clase este **createAwardListing(int auctionId, int bidId, int win, int payed),** aceasta primeşte ca parametri id-ul licitaţiei, id-ul bid-ului, valoarea 1 dacă licitaţia s-a terminat cu succes sau 0 în caz contrar şi valoarea 1 dacă preţul atins la licitaţie a fost plătit şi 0 în caz contrar. Sarcina acestei metode este de a trimite la domeniu aceste date, spre a fi introduse în baza de date. Pe baza acestor informaţii, se poate identifica bid-ul câştigător al unei licitaţii şi persoana care a câştigat licitaţia.

Clasa **MessageController** este folosită pentru a gestiona mesajele schimbate între utilizatori. Metodele acestei clase permit, salvarea unui mesaj, returnarea mesajelor trimise de o anumită persoană(**getMessageSent(int idUser)**), returnarea mesajelor primite de o anumită persoană(**getMessageReceived(int idUser)**) şi returnarea mesajelor schimbate în cadrul tranzacţionării unei licitaţii(**getMessageFromTransaction(int idUserSent, int idUserReceived)**).

Clasa **FeedbackController** primeşte datele de la nivelul de prezentare pentru a realiza operaţii asupra feedback-urilor.

Principalele metode ale acestei clase sunt:

- **getFeedback(int idUser).** Returnează feedback-urile existente pentru un anumit utilizator.
- **checkFeedback(int idUserSent, int idUserReceived, int idAuc).** Metoda verifică dacă un anumit utilizator a lăsat feedback altui utilizator, rezultatul returnat va fi de

tip boolean. Această metodă este folosită în zona de tranzacţionare pentru a permite unei persoane să lase feedback o singură dată, în cadrul unei tranzacţii.

- **getRating(int idUser).** Returnează media notelor lăsate unui anumit utilizator

Clasa **DomainInfoController** este folosită de către interfaţă pentru a obţine anumite detalii despre domenii.

Principalele metode ale acestei clase sunt:

- **getPageRank(String domain).** Returnează PageRank-ul unui domeniu. Pentru a obţine această valoare a fost folosită librăria ***PageRankService,*** care apelează ***toolbarqueries.google.com/***.
- **getAlexaRank(String domain).** Returnează rank-ul Alexa al unui domeniu.
- **getDomainInfo(String domain).** Returnează datele despre un domeniu.

Clasa **BidController**, este folosită pentru a realiza operaţii asupra bid-urilor.

Cele mai importante metode sunt:

- **AcceptBid(int bidId)**. Folosită pentru a permite creatorului unei licitaţii să accepte bid-urile primite
- **RejectBid(int bidId)**. Folosită pentru a permite creatorului unei licitaţii să respingă bid-urile primite

5.4. Nivelul de Model

La acest nivel a fost folosită folosit tehnologia EJB, pentru a defini domeniul problemei. Pentru fiecare Seassion Bean am creat o interfaţă Remote, pentru a putea fi accesată de către client. Acesta va putea accesa metodele definite în Seassion Bean, doar prin intermediul interfeţei.

Interfaţa Remote este folosită de clienţii remote care rulează într-un sistem diferit şi într-o maşină virtuală Java diferită. Locaţia bean-ului enterprise este transparentă faţă de cea a clientului remote.

Pentru a realiza interfeţele remote a fost folosită adnotarea @Remote

```
package businessSessionBeans;

import entities.UserEntityBean;
import javax.ejb.Remote;
/**
 *
 * @author Marius
 */
@Remote
public interface UserSeassionRemote {

    void edit(UserEntityBean uBean);
    public void create(String username, String password, String email, String phoneNumber, int rating,
    void destroy(Integer id);
    entities.UserEntityBean find(Object pk);
    java.util.List findAll();
}
```

Figura 5.5 Realizare Interfaţa Remote

Seassion Bean-urile sunt de tipul Stateless. Acestea nu păstrează starea conversaţiei cu clientul. Spre deosebire de Stateful Seassion Bean, acestea îşi păstrează starea conversaţiei, doar pe durata invocării metodei. Bean-urile Stateless au fost create folosind adnotarea @Stateless.

```
@Stateless
public class UserSeassionBean implements businessSessionBeans.UserSeassionRemote{

  @PersistenceContext
   EntityManager em;

   public UserSeassionBean() { }

   public void create(String username, String password, String email, String phoneNumber, int rating
      UserEntityBean uBean= new UserEntityBean();
      uBean.setUsername(username);
      uBean.setPassword(password);
      uBean.setEmail(email);
      uBean.setPhoneNumber(phoneNumber);
      uBean.setRating(rating);
      uBean.setAmount(amount);
      uBean.setVerified(verified);
      uBean.setBan(ban);
      em.persist(uBean);
   }
```

Figura 5.6 Creare Bean Stateless

Entităţile reprezintă obiecte de business într-un mecanism de stocare persistent. Fiecare entitate este reprezentată ca un tabel într-o bază de date relaţională. Pentru a defini entităţile şi cheile primare ale tabelelor au fost utilizate adnotări.

```
@Entity
public class AdminEntityBean implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Integer id;
    private String username;
    private String password;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }
```

Figura 5.7 Creare Entitate

Legătura aplicaţiei cu baza de date a fost realizată folosind fişierul glassfish-resources.xml. S-a definit în acest fişier, numele bazei de date, user-ul, parola şi driverul folosit.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE resources PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Resource Definitions//EN"
<resources>
    <jdbc-connection-pool allow-non-component-callers="false" associate-with-thread="false" connection-creat
        <property name="serverName" value="localhost"/>
        <property name="portNumber" value="3306"/>
        <property name="databaseName" value="auction"/>
        <property name="User" value="root"/>
        <property name="Password" value="root"/>
        <property name="URL" value="jdbc:mysql://localhost:3306/auction"/>
        <property name="driverClass" value="com.mysql.jdbc.Driver"/>
    </jdbc-connection-pool>
    <jdbc-resource enabled="true" jndi-name="auction2" object-type="user" pool-name="mysql_auction_rootPool"
```

Figura 5.8 Setare fişier glassfish-resources.xml

Clasele Entitate vor fi mapate în baza de date. Numele fiecărei clase entitate va corespunde unui tabel din baza de date, iar atributele vor corespunde coloanelor tabelului respectiv. Toate clasele entitate vor fi realizate în pachetul „entities”. EJB va fi structurat în 2 pachete: „entities” şi „businessSessionBeans”. Între clasele SessionBean şi entităţile corespunzătoare va exista o relaţie de dependenţa, iar între SeassionBean şi interfeţele

corespunzătoare va exista o relaţie de generalizare.

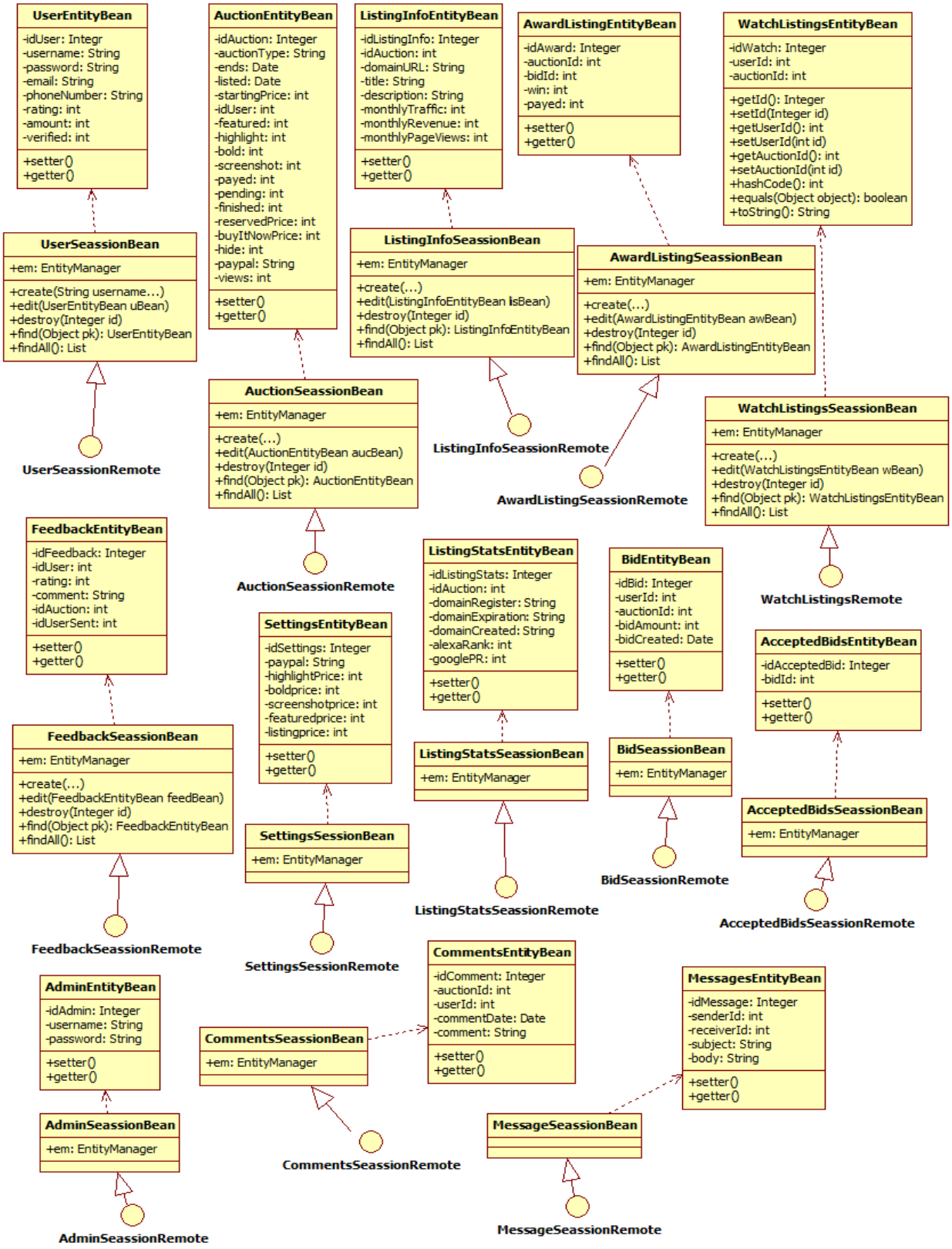

Figura 5.9 Diagrama de clase pentru EJB

5.5. Deployment Digram

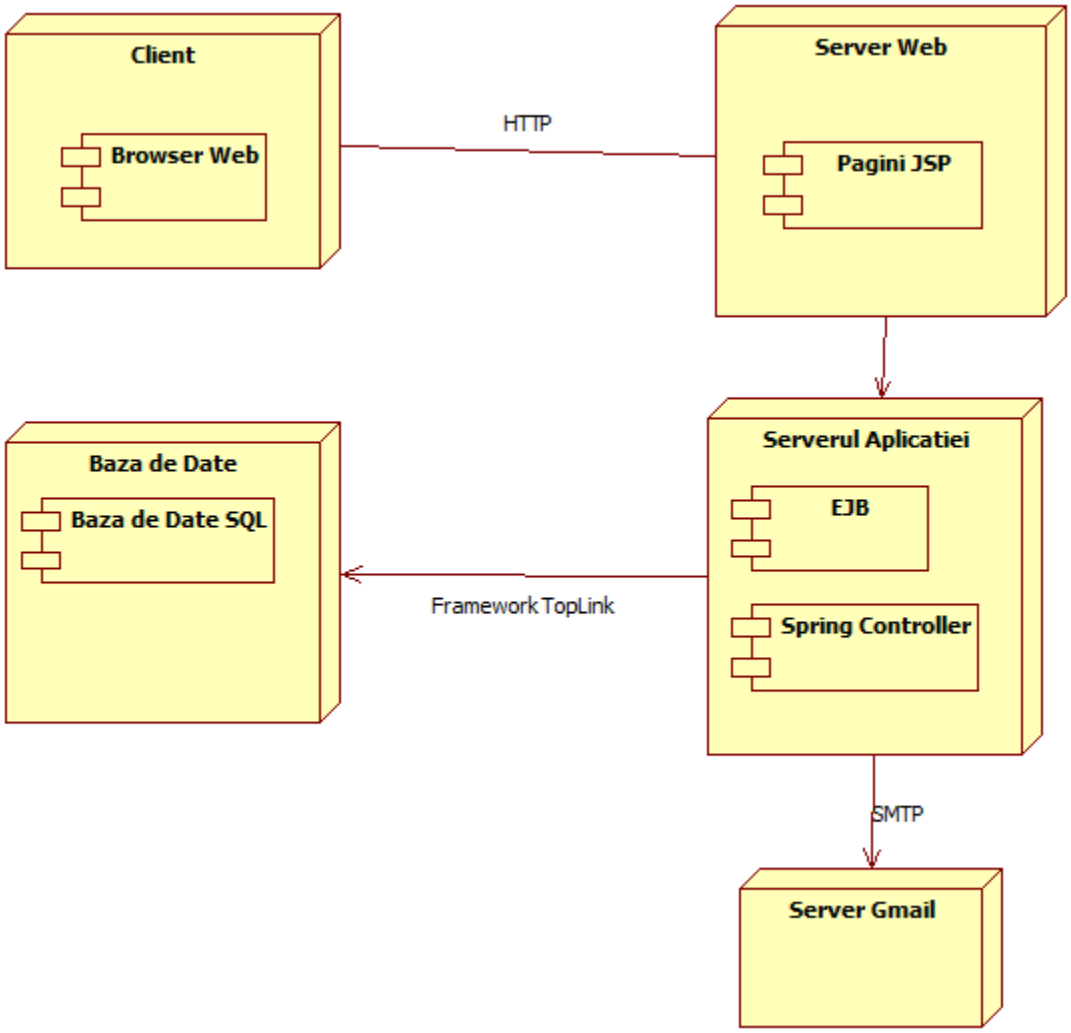

Figure 5.10 Digrama de Deployment

Diagrama de deployment, pentru sistemul de licitații online, conține un client, care va realiza cereri HTTP, către serverul web și care va conține paginile JSP ale site-ului. Serverul WEB va realiza invocarea metodelor de business conținute în serverul aplicației. Acesta din urmă, va fi format din tehnologia EJB și din Controllere Spring, care vor procesa cererile provenite de la nivelul interfației. Serverul aplicației se va conecta la serverul bazei de date, cu ajutorul framework-ului TopLink, iar la serverul Gmail, prin SMTP.

5.6. Nivelul de Date

Datele furnizate de utilizatori vor fi salvate într-o bază de date relaţionala, MySQL care conţine 13 tabele.

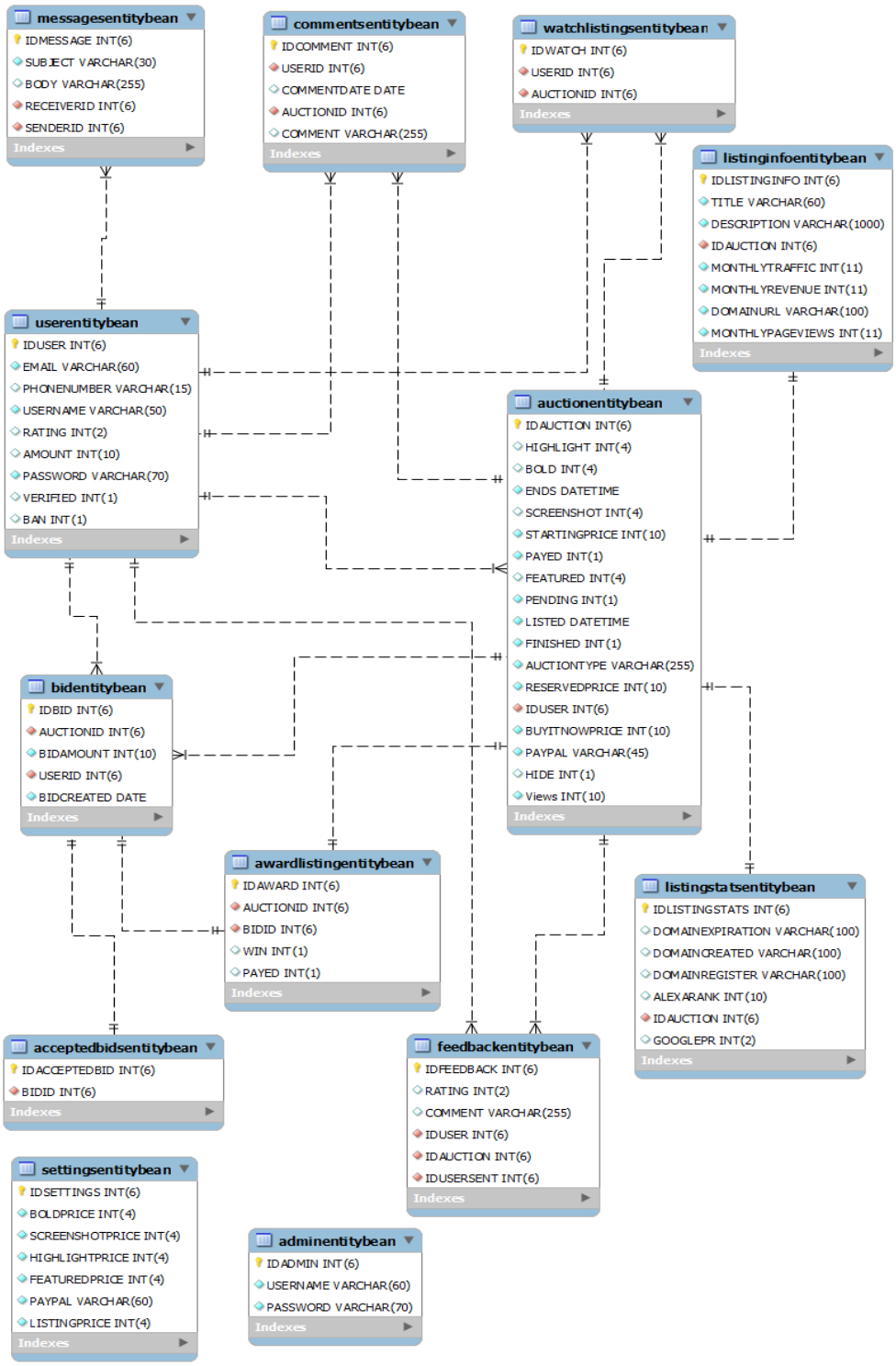

Figura 5.11 Structura Generală a Bazei de Date

Tabela UserEntityBean este folosită pentru a stoca informaţiile despre utilizatori şi are următoarele coloane:

- IdUser – reprezintă cheia primară a tabelei şi identifică în mod unic un utilizator
- Email – câmp în care se va stoca emailul utilizatorului
- Phonenumber – câmp în care se va stoca numărul de telefon al utilizatorului
- Username – câmp în care se va stoca username-ul utilizatorului
- Rating – câmp în care va fi salvată media rating-uui primit
- Amount – câmp în care va fi salvat numărul de credite al utilizatorului
- Password- câmp în care va fi salavată parola
- Veriefied – câmp în care va fi salvată valoarea 0, dacă contul nu are verificată adresa de email şi 1 dacă este verificată
- Ban – câmp care va fi salvată valoarea 0 dacă contul nu este banat şi 1 în caz contrar

Tabela MessageEntityBean este folosită pentru a salva mesajele schimbate între utilizatori şi conţine următoarele coloane:

- IdMessage câmp în care va fi salvată o valoare de tip întreg care va fi cheia primară a tabelului
- ReceiverId - câmp în care va fi salvat id-ul utilizatorului care va primi mesajul.
- Subject – câmp în care va fi salvat subiectul mesajului
- SenderId – câmp în care va fi salvat id-ul utilizatorului care a trimis mesajul
- Body – câmp în care va fi salvat conţinutul mesajului

Între tabelele UserEntityBean şi MessageEntityBean există o relaţie de 1 la n, deoarece id-ul unui utilizator din tabelul UserEntityBean, poate fi asociat cu mai multe mesaje din tabela MessageEntityBean.

Tabela CommentsEntityBean este folosită pentru a salva comentariile realizate de utilizatori în cadrul unei licitaţii. Aceasta tabelă conţine următoarele coloane:

- IdComment – câmp în care va fi salvat id-ul comentariului, cu ajutorul căruia se vor identifica comentariile. Reprezintă cheia primară a tabelului
- UserId – câmp în care va fi salvat id-ul user-ului care a lăsat comentariul
- CommentDate – câmp în care va fi salvată dată în care a fost realizat comentariul
- AuctionId – câmp în care va fi salvat id-ul licitaţiei în cadrul căreia a fost lăsat comentariul
- Comment – câmp în care va fi salvat conţinutul comentariului

Tabele UserEntityBean şi CommentsEntityBean, sunt în relaţie de 1 la n, deoarece id-ul user-ului din tabela UserEntityBean, poate fi asociat cu mai multe comentarii din tabela CommentsEntityBean.

Tabela AuctionEntityBean este în relaţie de 1 la n cu acest tabel, deoarece id-ul licitaţiei din tabela AuctionEntityBean, poate fi asociat cu mai multe commentarii din tabela CommentsEntityBean.

Tabela AuctionEntityBean este folosită pentru a salva informaţiile despre licitaţie şi are următoarele coloane:

- IdAuction – câmp în care este salvat id-ul licitaţiei şi reprezintă cheia primară a tabelului
- Highlight – câmp folosit pentru a salva preţul unei licitaţii cu titlul evidenţiat
- Bold – câmp folosit pentru a salva preţul necesar pentru a crea o licitaţie cu titlul boldat

- Ends – câmp folosit pentru a salva data în care se va termina licitaţia
- Screenshot – câmp folosit pentru a salva preţul afişării screenshot-ului pe pagina de căutare licitaţii
- StartingPrice – câmp folosit pentru a salva preţul de început al licitaţiei
- Payed – câmp care va avea valoarea 1 dacă utilizatorul a plătit taxele licitaţiei sau 0 în caz contrar
- Featured – câmp în care va fi salvat pretul licitaţiei premium
- Pending – câmp care va avea valoarea 0 dacă licitaţia este activa şi 1 dacă licitaţia este oprită
- Listed – câmp în care va fi salvată dată la care va începe licitaţia
- Finished – câmp care va avea valoarea 1 dacă s-a terminat timpul destinat licitaţiei
- AuctionType – câmp în care va fi salvat tipul licitaţiei
- ReservedPrie – câmp în care va fi salvat preţul rezervat al licitaţiei
- IdUser- câmp în care va fi salvat id-ul utilizatorului care a creat licitaţia
- BuyItNowPrice – câmp în care va fi salvat preţul de cumpără acum
- Paypal – câmp în care va fi salvată adresa de paypal folosită de vânzător
- Hide – câmp care va avea valoarea 1 dacă licitaţia este deactivată de către admin
- Views – câmp în care va fi salvat numărul de vizualizări al licitaţiei

Tabela UserEntityBean este în relaţie 1 la n cu tabela AuctionEntityBean, deoarece IdUser din tabela UserEntityean, poate avea asociate mai multe licitaţii în tabela AuctionEntityBean.

Tabela ListingInfoEntityBean este folosită pentru a salva datele despre site-ul listat la licitaţie şi conţine următoarele coloane:

- IdListing – câmp folosit pentru a salava id-ul tabelului. Reprezintă cheia primară a tabelului
- Title – câmp folosit pentru a sală titlul licitaţiei
- Description – câmp folosit pentru a salva descrierea licitaţiei
- IdAuction – câmp folosit pentru a salva id-ul licitaţiei în cadrul căreia vor fi folosite aceste informaţii
- MonthlyTraffic – câmp folosit pentru a salva numărul de utilizatori ai site-ului
- MonthlyRevenue – câmp folosit pentru a salva profitul site-ului
- DomainUrl – câmp folosit pentru a salva adresa URL site-ului
- MonthlyPageViews – câmp folosit pentru a salva numărul de impresii ale site-ului

Tabela AuctionEntityBean şi ListingInfoEntityBean sunt într-o relaţie 1 la 1, deoarece id-ul licitaţiei din tabelul AuctionEntityBean poate fi asociat o singură dată cu datele din tabelul ListingInfoEntityBean.

Tabela ListingStatsEntityBean este folosit pentru a salva datele furnizate de sistem, despre site-ul licitat şi conţine următoarele coloane:

- IdListingStats – câmp în care este salvat id-ul unic al tabelei. Reprezintă cheia primară.
- DomainExpiration – câmp în care va fi salvată dată de expirare a domeniului
- DomainCreated – câmp în care va fi salvată data în care a fost înregistrat domeniul
- DomainRegister – câmp în care este salvată denumirea companiei la care este înregistrat domeniul
- AlexaRank – câmp în care este salvat rank-ul Alexa

- IdAuction – câmp în care este salvat id-ul licitaţiei, pentru care sunt salvate datele
- GooglePr – câmp în care este salvat PR furnizat de Google

Între tabela AuctionEntityBean şi ListingStatsEntityBean există o relaţie de 1 la 1, deoarece id-ul licitaţiei din tabela AuctionEntityBean poate fi asociat doar o singură dată cu această tabelă.

Tabela WatchListingsEntityBean este folosită pentru a salva licitaţiile urmărite de utilizatori. Acest tabel are următoarele coloane:

- IdWatch – câmp folosit pentru a salva id-ul licitaţiilor urmărite. Acest câmp este cheia primară a tabelului.
- UserId – câmp folosit pentru a salva id-ul utilizatorului care urmăresc licitatia
- AuctionId – câmp folosit pentru a salva id-ul licitaţiei urmărite

Între tabela UserEntityBean şi tabela WatchlistingsEntityBean este o relaţie de 1 la n, deoarece câmpul IdUser poate fi asociat cu mai multe licitaţii urmărite în tabela WatchListingsEntityBean.

Între tabela AuctionEntityBean şi tabela WatchListingsEntityBean este o relaţie de 1 la n, deoarece câmpul IdAuction poate fi asociat cu mai multe licitaţii urmărite.

Tabela FeedbackEntityBean este folosită pentru a salva feedback-ul lăsat de utilizatori şi conţine următoarele coloane:

- IdFeedback – câmp în care este salvat id-ul unic al tabelului şi reprezintă cheia primară a acestuia
- Rating – câmp în care este salvat rating-ul lăsat de utilizator
- Comment – câmp în care este salvat feedback-ul lăsat utilizatorului
- IdUser- câmp în care este salvat id-ul utilizatorului pentru care a fost lăsat rating-ul
- IdAuction – câmp în care este salvat id-ul licitaţiei, în cadrul căreia a fost lăsat rating-ul
- IdUserSent – câmp în care este salvat id-ul utilizatorului care a lăsat rating-ul

Între tabela UserEntityBean şi FeedbackEntityBean există o relaţie de 1 la n, deoarece un utilizator poate avea asociate mai multe feedback-uri.

Între tabela AuctionEntityBean există o relaţie de 1 la n, deoarece în cadrul unei licitaţii se pot salva mai multe feedback-uri în tabela FeedbackEntityBean.

Tabela BidEntityBean este folosită pentru a salva detaliile bid-ului trimis de utilizatori şi conţine următoarele coloane:

- IdBid – câmp folosit pentru a salva id-ul bid-ului. Acest câmp este şi cheia primară a tabelului.
- AuctionId – câmp în care este salvat id-ul licitaţiei în cadrul căreia a fost plasat bid-ul
- BidAmount – câmp în care a fost salvată suma licitată de utilizator
- UserId – câmp în care este salvat id-ul utilizatorului care a plasat bid-ul
- BidCreated – câmp în care este salvată data la care a fost creat bid-ul

Între tabela UserEntityBean şi BidEtityBean există o relaţie de 1 la n, deoarece un utilizator poate plasa mai multe bid-uri care vor fi salvate în BidEntityBean.

Între tabela AuctionEntityBean şi BidEntityBean există o relaţie de 1 la n, deoarece în cadrul unei licitaţii se pot realiza mai multe bid-uri care vor fi salvate în tabelul BidEntityBean.

Tabela AcceptedBidsEntityBean a fost creată pentru a salva bid-urile acceptate de creatorul licitaţiei şi conţine următoarele coloane:

- IdAcceptedBid – câmp în care este salvat id-ul unic al tabelei. Acest câmp reprezintă şi

cheia primară.

- BidId – câmp în care a fost salvat id-ul bid-ului care a fost acceptat de către vânzător

Între tabela BidEntityBean şi AcceptedBidsEntityBean este o relaţie de 1 la 1, deoarece un bid poate fi acceptat o singură dată şi salvat in tabela AcceptedBidsEntityBean.

Tabela AwardListingEntityBean a fost creată pentru a face legătura între licitaţia câştigată şi bid-ul câştigător. Acest tabel conţine următoarele coloane:

- IdAward – câmp în care este salvat id-ul unic al tabelei. Reprezintă cheia primară.
- AuctionId – câmp în care este salvat id-ul licitaţiei pentru care s-a licitat
- BidId – câmp în care este salvat id-ul bid-ului cu valoarea cea mai mare din cadrul unei licitaţii
- Win – câmp care conţine valoarea 1, dacă licitaţia a fost câştigată sau 0 în caz contrar
- Payed – câmp care conţine valoarea 1 dacă câştigătorul a plătit licitaţia şi 0 în caz contrar

Între tabela BidEntityBean şi AwardListingEntityBean există o relaţie de 1 la 1, deoarece poate fi asociat un singur bid cu acest tabel.

Între tabela AuctionEntityBean şi AwardListingEntityBean există o relaţie de 1 la 1, deoarece id-ul unei licitaţii poate fi asociat o singură dată cu acest tabel.

Tabela SettingsEntityBean este folosită pentru a salva setările realizate de administrator. Acest tabel conţine următoarele coloane:

- IdSettings – câmp folosit pentru a salva id-ul tabelei. Reprezintă cheia primară.
- BoldPrice – câmp folosit pentru a sava preţul pentru licitaţiile cu titlul boldat
- ScreenshotPrice – câmp folosit pentru a salva preţul pentru licitaţiile cu screenshot pe pagina de căutare
- HighlightPrice – câmp folosit pentru a salva preţul pentru licitaţiile cu titlul evidenţiat
- FeaturedPrice – câmp folosit pentru a salva preţul licitaţiilor premium
- Paypal – câmp folosit pentru a salva adresa de paypal în care se vor face plăţile
- ListingPrice – câmp folosit pentru a salva preţul creării unei licitaţii

Tabela AdminEntityBean este folosită pentru a salva datele administratorului şi conţine următoarele coloane:

- IdAdmin – câmp în care este salvat id-ul acestei tabele. Reprezintă cheia primară.
- Username – câmp în care este salvat numele de utilizator al administratorului
- Password – câmp în care este salvată parola administratorului

5.7. Interacţiunea dintre componente

În acest subcapitol voi descrie interacţiunile dintre componente ce au loc pentru a satisface cazurile de utilizare enumerate anterior. Interacţiunile vor fi descrise folosind diagrama de secvențiere ce respectă standardul UML 2.x.

Pentru a licita în cadrul unei licitaţii active, au loc următoarele interacţiuni între componente:

Pentru a putea licita, utilizatorul trebuie să se logeze. În cazul în care logarea are loc cu succes, acesta va fi redirecţionat spre pagina principală a site-ului. De aici va apăsa pe pagina cu licitaţii active, unde va fi creat un obiect de tip *SearchController().* Cu ajutorul acestui obiect se va crea un obiect de tip AuctionSeassionRemote, care va realiza citirea datelor din baza de date.

Datele licitaţiilor vor fi afişate în Browser-ul web, unde utilizatorul va selecta licitaţia

dorită şi va fi redirecţionat spre pagina licitaţiei. În această pagină va apăsa pe butonul „*Bid*” şi se va crea un obiect de tip *BidController(),* care va crea un obiect de tip BidSeassionBean, pentru a realiza scrierea datelor în baza de date.

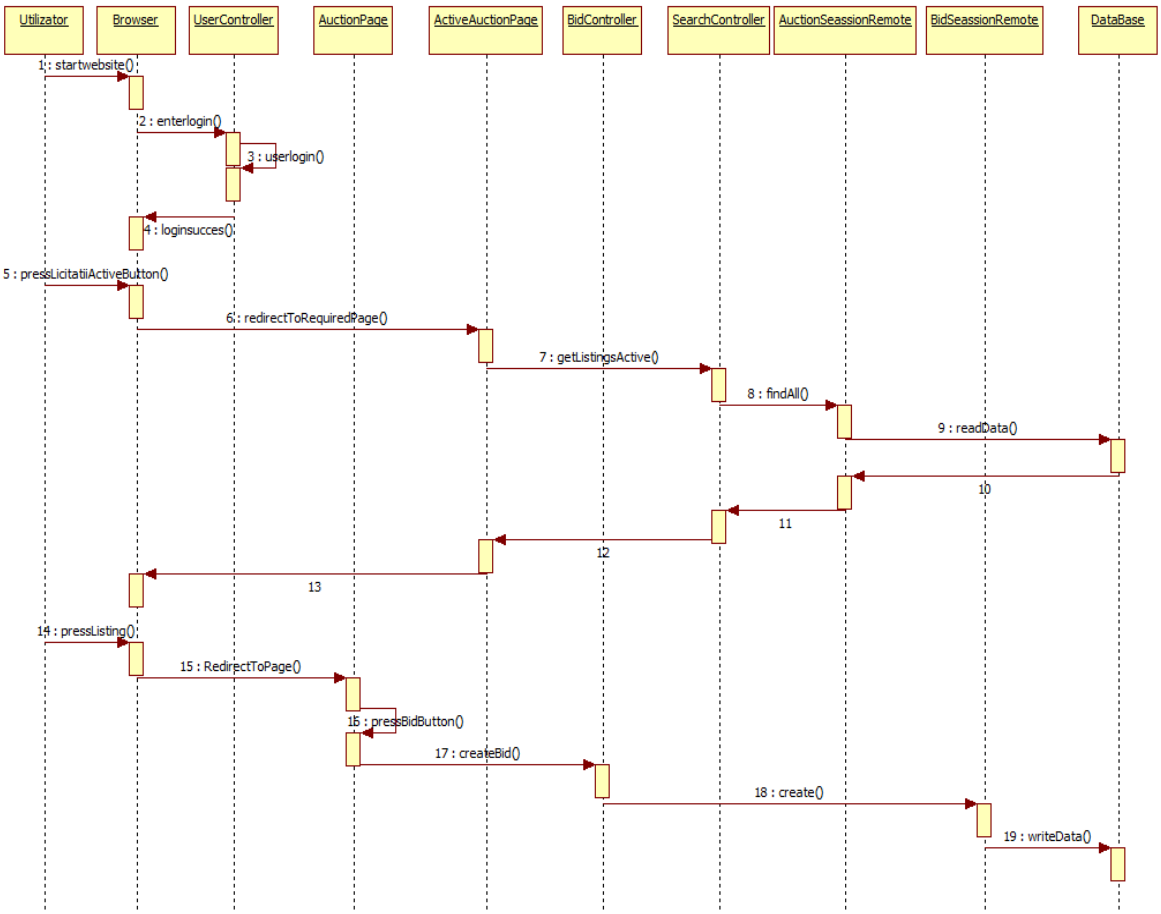

Figure 5.13 Diagrama de Secvenţe pentru plasare Bid

Pentru ca administratorul să vizualizeze conturile utilizatorilor au loc următoarele interacţiuni între componente:

Administratorul accesează platforma prin intermediul unui browser web. Acesta va completa datele de logare şi le va trimite spre a fi procesate pe server, iar în cazul în care vor fi valide, va fi redirecţionat spre pagina de administrator.

Pentru a vizualiza toţi utilizatorii, va accesa pagina utilizatorilor, unde se va crea un obiect de tip UserController, prin intermediul căruia se va crea un obiect de tip UserSeassionBean pentru a citi datele din baza de date. Acestea vor fi trimise pentru a fi afişate pe pagina de administrare a utilizatorilor.

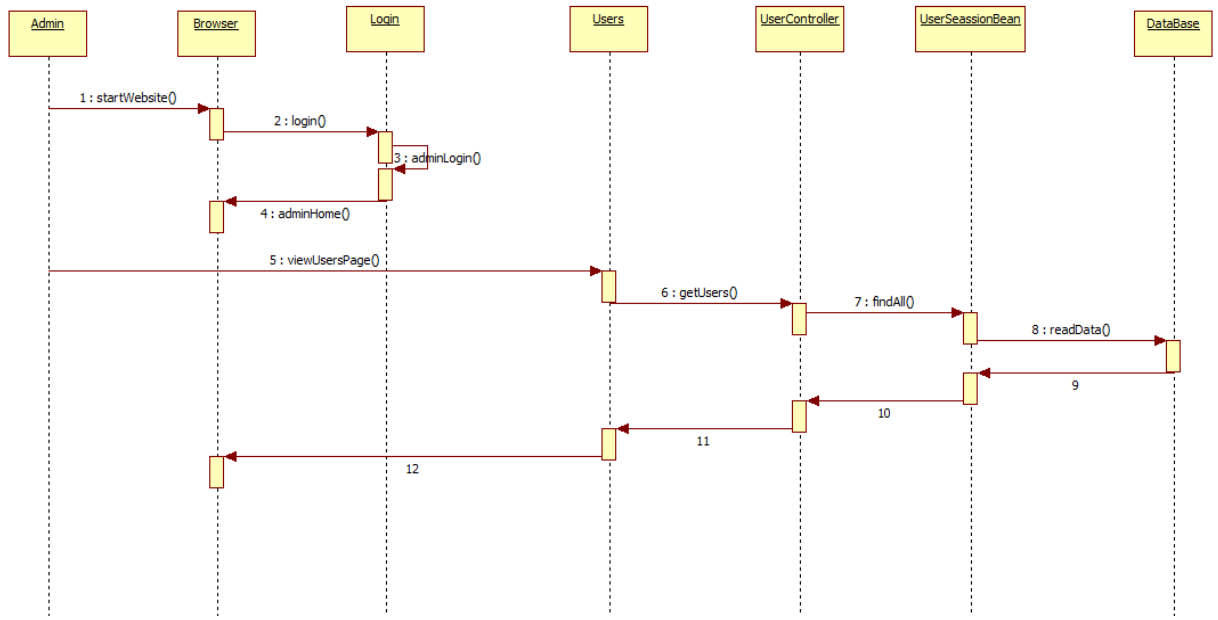

Figure 5.12 Diagrama de Secvenţe pentru afişare utilizatori

Cu ajutorul diagramelor de secvență a fost prezentată interactiunea dintre principalele componente ale aplicației, pentru a duce la indeplinire anumite functionalități, care pot fi realizate de către utilizatorii sistemului.

6. Testare şi Validare

Testarea software este procesul software folosit pentru a identifica corectitudinea, securizarea şi calitatea produsului software dezvoltat. Testarea este un proces de verificare tehnică, care are scopul de a descoperi informaţii despre produsul realizat cu respect la contextul în care este dorit a fi operat.

Testarea presupune procesul de executare a unui program cu scopul de a descoperi erorile existente. Calitatea nu este o cerinţă absolută, ci este o valoare definită de o persoană. Luând acest lucru în considerare, testarea nu poate stabili în totalitate corectitudinea unui produs software arbitrar. Prin intermediul procesului de testare realizăm o verificare a stării şi comportamentului aplicaţiei în funcţie de specificaţiile acestuia.

Indiferent de metodele de testare folosite, sau a nivelului de formalitate implicat rezultatul testării reprezintă un nivel de încredere în produsul realizat, astfel încât organizaţia poate fi încrezătoare că produsul o să aibă o rata acceptabilă a defectării. Această rată depinde de natura produsului software realizat.

O problemă a testării produselor software este dată de faptul că defectele care apar într-un produs software pot să fie foarte mari, iar numărul configurărilor produsului pot să fie şi mai mari. Problemele care apar foarte rar sunt greu de descoperit prin intermediul testării. O regulă de aur este aceea că un sistem, care doreşte să funcţioneze fără probleme pentru o anumită perioadă de timp ar trebui să fie testat pentru cel puţin jumătate din perioada de timp.

O practică des întâlnită a testării software este reprezentată de faptul că este realizată de un grup independent de testeri după fiecare funcţionalitate dezvoltată, dar înainte să fie livrată clientului.

Pentru testarea funcţionalităţilor sistemului de licitaţii am folosit framework-ul Junit, pentru testarea componentelor de business, fromework-ul JWebUnit, pentru testarea interfeţei şi profiler-ul furnizat de YourKit, pentru a studia performanta şi resursele consumate de sistem.

Testarea Componentelor de Business

Testarea componentelor de business a fost realizat cu ajutorul framework-ului de testare automată Junit 4.x. Acest framework permite realizarea testelor în limbajul de programare în care a fost realizat proiectul, testarea uşoară a obiectelor/claselor/metodelor individuale şi execuţia unui sau mai multor teste printr-o singură acţiune.

Am realizat cu ajutorul acestui framework, testarea principalele metode a claselor din pachetul Controller. Prin intermediul testelor am studiat comportamentul şi corectitudinea metodelor prin verificarea dacă datele returnate de către aceste metode sunt corecte.

Testele sunt separate de logica aplicaţiei, prin plasarea tuturor claselor de test într-un nou pachet. Am realizat metode de testare pentru următoarele componente:

Am testat principalele metode din clasa **AuctionController**, pentru a mă asigura că datele afişate utilizatorilor sunt corecte. Prin metodele *testGetSiteTitle(), testGetSitePageViews(), testGetSitePageVisitors(), testGetSiteRevenue(), testGetListingInfo(), testGetListingStats(), testGetLisiting(), testGetLisitings(), testGetLisitingsbyUser()*, am verificat dacă metodele care realizează furnizarea titlului, numărului de vizualizări, numărului de vizitatori, profitul site-ului, detaliile licitaţiei, detaliile domeniului sunt cele corecte. Pentru a realiza acest lucru am verificat dacă datele furnizate de aceste metode sunt aceleaşi cu cele existente în baza de date.

În urma executării acestor metode de test, toate rezultatele au trecut testul.

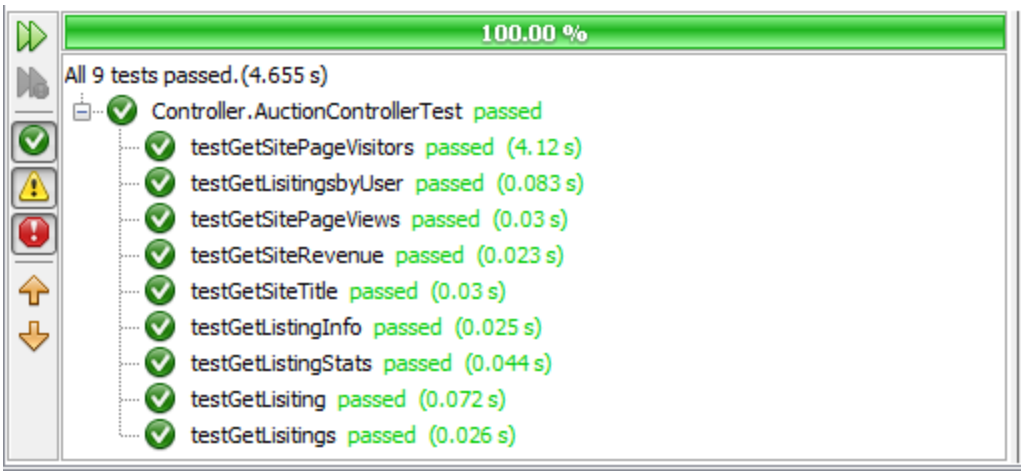

Figura 6.1 Rezultatul testelor pentru clasa AuctionController

Am realizat următoarele metode de test pentru clasa **BidController**: *testAcceptBid(), testGetBids(), testGetBidsInfo(), testGetBidInfo(), testGetPret(), testGetWinningBid(), testCheckAccepted()*. Prin intermediul acestor metode am verificat dacă rezultatele returnate de către principalele metode din clasa BidController sunt aceleaşi cu cele salvate în baza de date.

În urma evaluării condiţiilor, rezultatele au trecut toate testele.

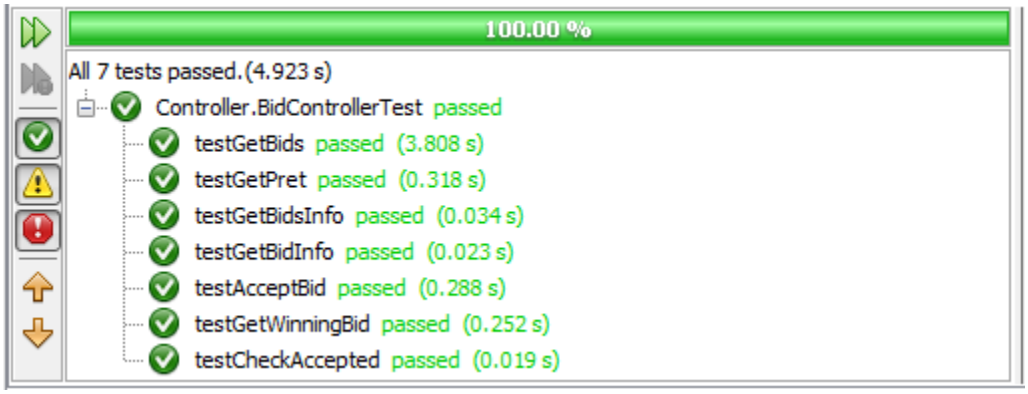

Figura 6.2 Rezultatul Testelor pentru clasa BidController

Pentru clasa SaleController, am creat următoarele metode de test: *testEditAwardListingPayed(), testGetAwardListing_int_int(), testIsSold(), testGetWinningBid()*, pentru a verifica dacă datele furnizate de metodele clasei corespund cu cele existente în baza de date.

În urma evaluării condiţiilor, rezultatele au trecut toate testele.

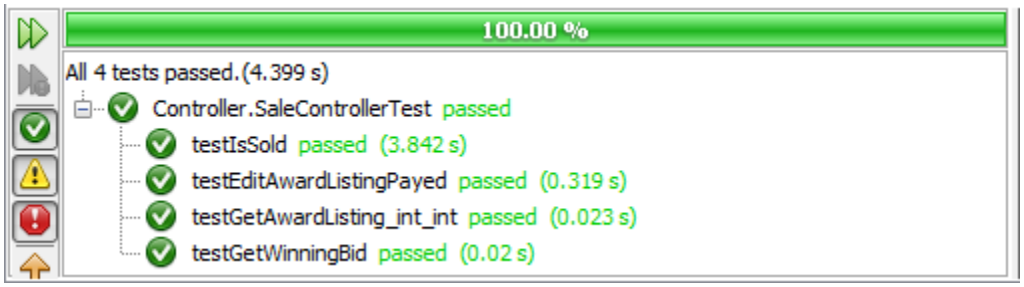

Figura 6.3 Rezultatele Testelor pentru clasa SaleController

Testarea Interfeţei Grafice

Pentru testarea interfeţei grafice am folosit framework-ul JWebUnit, pentru a testa dacă toate elementele interfeţei funcţionează corect. Acest framework foloseşte HtmlUnit şi Selenium, cu o interfaţă unificata simplă, pentru a permite verificarea corectitudinii aplicaţiei web.

JWebUnit furnizează un API de nivel înalt, pentru a naviga prin aplicaţia web şi a folosi un set de aserţiuni pentru a verifica corectitudinea aplicaţiei. Navigarea se poate realiza prin intermediul link-urilor, form-urilor, validarea conţinutului tabelelor şi alte funcţionalităţi ale aplicaţiei web.

Am realizat un test pentru a verifica corectitudinea link-urile de navigare. Astfel am setat link-ul de pornire şi am navigat pe mai multe pagini, verificând pentru fiecare dacă există anumite elemente, pentru a fi sigur că link-ul redirecţionează spre pagina care trebuie.

```
@Test
public void testNavigation() {
    beginAt("index.htm");
    clickLinkWithText("Licitatii Premium");
    assertTitleEquals("Vizualizati licitatiile Premium");
    clickLinkWithText("Licitatii noi");
    assertTitleEquals("Vizualizati licitatiile noi");
    clickLinkWithText("Cele mai active");
    assertTitleEquals("Vizualizati cele mai active licitatii");
    clickLinkWithText("Termina Curand");
    assertTitleEquals("Vizualizati licitatiile care se termina in curand");
    clickLinkWithText("Nevandute");
    assertTitleEquals("Vizualizati licitatiile nevandute");
    clickLinkWithText("Vandute Recent");
    assertTitleEquals("Vizualizati licitatiile vandute recent");
    clickLinkWithText("Acasa");
    assertTitleEquals("Cumpara si vinde site-uri la licitatie");
    clickLinkWithText("Cumpara");
    assertTitleEquals("Vizualizati licitatiile noi");
    clickLinkWithText("Vinde");
    assertLinkPresentWithText("Apasati aici pentru logare");
}
```

Figura 6.4 Test pentru verificarea link-urilor de Navigare

Am realizat un alt test pentru a verifica modul de funcţionare a paginii de licitaţie, în funcţie de tipul de utilizator: logat sau nelogat. În momentul în care utilizatorul nelogat încearcă să trimită un comentariu, sau dacă încearcă să se log-eze cu date incorecte am verificat dacă apar pe această pagină mesaje corespunzătoare.

```
@Test
public void testAuctionPage() {
    beginAt("index.htm");
    clickLinkWithText("Vandute Recent");
    assertTitleEquals("Vizualizati licitatiile vandute recent");
    clickLinkWithText("Site cu poze hazlii");
    setTextField("com", "a");
    submit();
    assertTextPresent("Trebuie sa va logati");
    setTextField("username", "a");
    setTextField("password", "a222");
    assertButtonPresent("logid");
    clickButton("logid");
    assertTextPresent("Logarea a esuat.");
    setTextField("username", "a");
    setTextField("password", "a");
    assertButtonPresent("logid");
    clickButton("logid");
    assertTextPresent("Logout");
    clickLinkWithText("Logout");


}
```

Figura 6.5 Test pentru verificare funcţionare pentru Pagina Licitaţiei

Am realizat al treilea test, pentru a verifica dacă zona de administrator funcţionează în mod corect, pentru aceasta, am verificat ce se întâmplă dacă administratorul introduce date de logare incorecte şi modul de funcţionare al link-urilor de navigare.

```
@Test
public void testAdmin() {
    beginAt("adminlogin.htm");
    setTextField("user", "a");
    setTextField("parola", "a");
    submit();
    assertTextPresent("Logarea a esuat. Userul sau parola sunt incorecte");
    setTextField("user", "admin");
    setTextField("parola", "admin");
    submit();
    assertTextPresent("Bun venit");
    clickLinkWithText("Management Useri");
    assertTitleEquals("Management Useri");
    clickLinkWithText("Management Licitatii");
    assertTitleEquals("Management Licitatii");
    clickLinkWithText("Setari Site");
    assertTitleEquals("Setare Site");
    clickLinkWithText("Log Out");
}
```

Figura 6.6 Test pentru verificare zona Administrator

În urma executării acestor trei teste, toate verificările au fost corecte, rezultând astfel ca paginile funcţionează în mod corect.

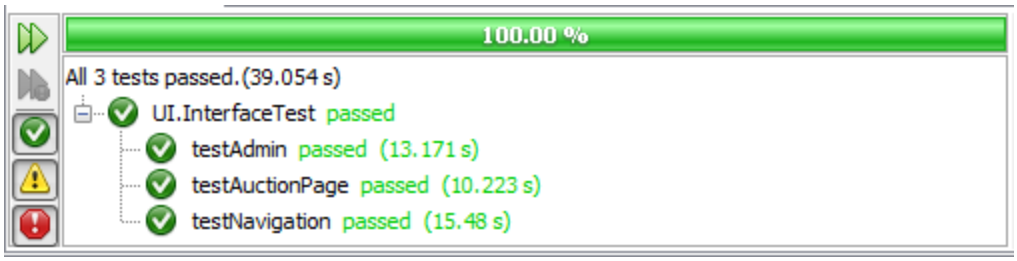

Figura 6.7 Rezultatele execuţiei testelor pentru interfaţa

YourKit Profiler

Am realizat un profil al site-ului folosind unealta YourKit Profiler pentru a analiza performanta sistemului. Unele dintre problemele greu de identificat într-o instalare a aplicaţiei web pe server sunt problemele de performanţă. Acestea pot apărea datorită unor probleme a produsului software, sistemului de back-end care nu funcţionează cum ar trebui, o problemă de reţea sau alţi factori interni sau externi.

Folosind profiler-ul YourKit am analizat resursele folosite de aplicaţia web.

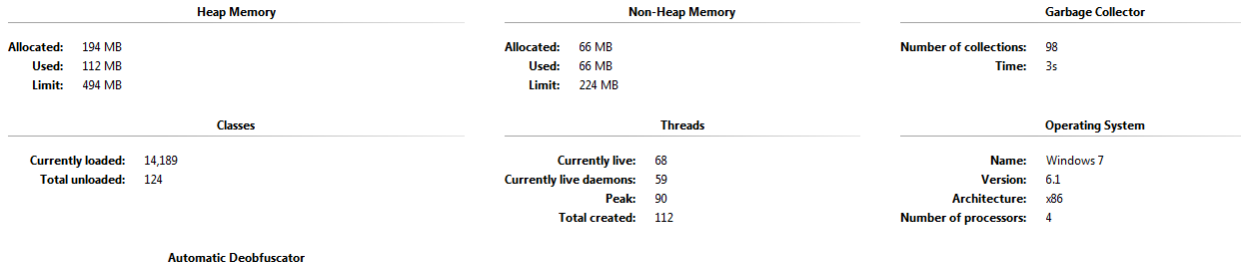

Figura 6.8 Analiza resurselor folosind profiler-ul YourKit

7. Manual de Instalare şi Utilizare

În cele ce urmează voi prezenta principalele resurse necesare pentru funcţionarea site-ului, precum şi paşii necesari pentru a instala aplicaţiile necesare şi pentru a folosi platforma de licitaţii.

7.1. Manual de instalare

Cerinţe Sistem

Pentru ca site-ul să funcţioneze în condiţii optime avem nevoie de un server care suportă aplicaţii java, cu următoarele cerinţe minime:

- 128 MB de RAM
- Spaţiu pe disc de 150 MB

Aplicaţiile necesare pentru a fi instalate sunt:

Java EE 7 SDK

Instalaţi Java EE 7 SDK, urmând paşii de mai jos:

1. Mergeţi la CD-ul furnizat o dată cu aplicaţia şi localizaţi fişierul java_ee_sdk-6u3-jdk7-windows-ml, sau la adresa: http://www.oracle.com/technetwork/java/javaee/downloads/index.html
2. Urmaţi instrucţiunile de pe ecran

NetBeans IDE 7.0.1

După aceasta vom instala NetBeans IDE, urmând următorii paşi:

1. Mergeţi la CD-ul furnizat cu aplicaţia şi localizaţi fişierul: netbeans-7.0.1-ml-javaee-windows, sau la adresa: https://netbeans.org/downloads/7.0.1/
2. Urmaţi instrucţiunile de pe ecran
3. Din lista de opţiuni disponibile, alegeţi alegeţi versiunea JDK 7

MySql 5.5

Pentru a instala MySql 5.5 trebuie să urmaţi paşii:

1. Mergeţi la CD-ul furnizat o dată cu aplicaţia şi localizaţi fişierul: mysql-5.5.12-win32, sau la adresa : http://dev.mysql.com/downloads/mysql/
2. Urmaţi instrucţiunile de instalare
3. Introduceţi parola de administrator

După instalarea tuturor aplicaţiilor, deschideţi proiectul în netbeans folosind funcţia file -> open project, iar după aceasta importaţi toate librăriile din folderul lib de pe CD în proiectul tocmai deschis.

7.2. Manual de Utilizare

Pentru utilizarea aplicaţiei, voi relua unele scenarii ale cazurilor de utilizare descrise în capitolele precedente.

Pagina de „*Home*” a site-ului permite utilizatorilor să vizualizeze statistice ale sistemului de licitaţii şi să acceseze tipul de licitaţii dorit: cele mai active, premium, terminate cu succes, nevândute, care se termina curând şi să creeze noi licitaţii.

Pentru a se loga utilizatorul trebuie să acceseze pagina de „*login*”, unde va trebui să insereze adresa de email şi parola. În cazul în care datele vor fi corecte, utilizatorul va fi

redirecţionat spre pagina principală, iar în caz contrar va fi afişat un mesaj de eroare.

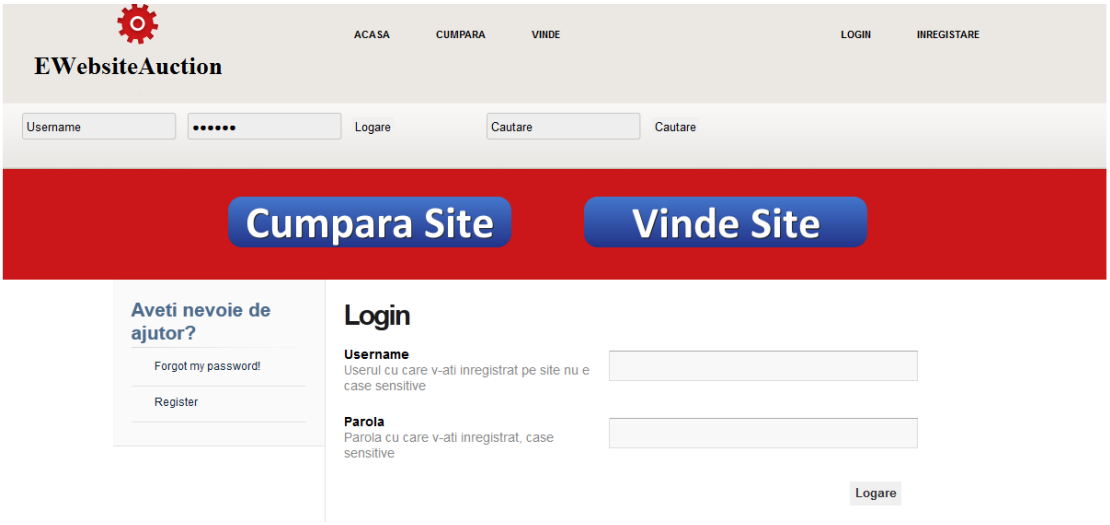

Figura 7.1 Pagina de Login a sistemului

Pentru a crea o nouă licitaţie utilizatorii vor accesa Pagina „*Vinde Site*”, aici ei vor trebui să introducă datele despre site-ul dorit a fi tranzacţionat şi să aleagă îmbunătăţirile dorite. Dacă datele introduse vor fi corecte, aceştea vor fi redirecţionaţi spre pagina de plată a taxelor de creare a licitaţiei.

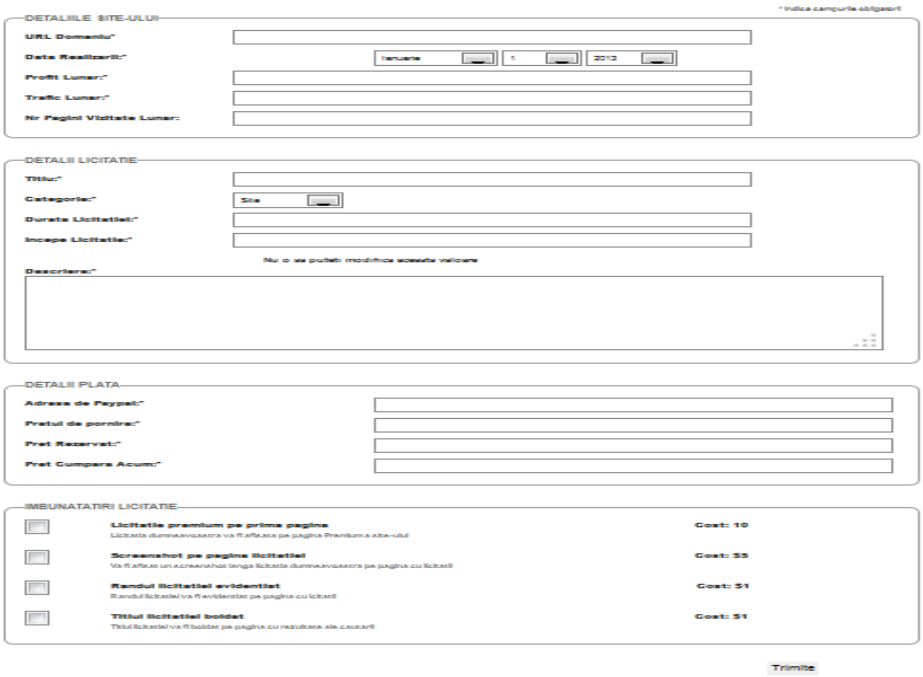

Figura 7.2 Câmpurile necesare pentru crearea unei licitaţii

În urma completării datelor necesare creării licitaţiei, şi trimiterii lor spre server, utilizatorii vor fi redirecţionaţi spre pagina de realizare a plăţii.

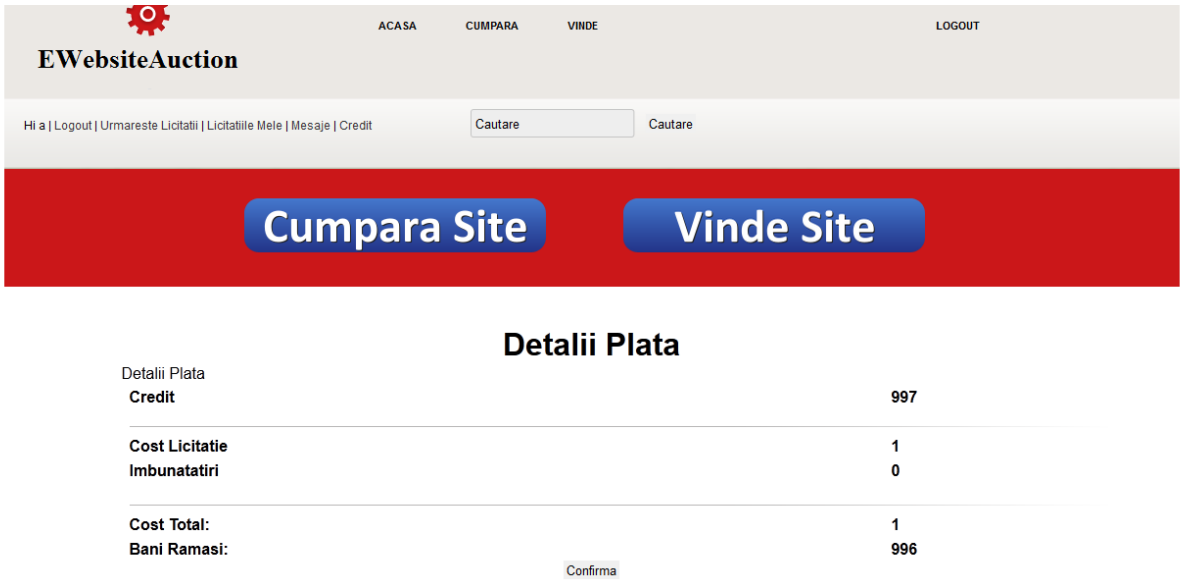

Figura 7.3 Realizare Plată Taxe Licitaţie

În momentul realizării plăţii, utilizatorul va fi redirecţionat spre pagina licitaţiei, unde va putea accepta/respinge bid-urile primite, să verifice sau să trimită comentarii noi, să vizualizeze detaliile licitaţiei, să verifice profilul vânzătorului şi să urmărească licitaţia.

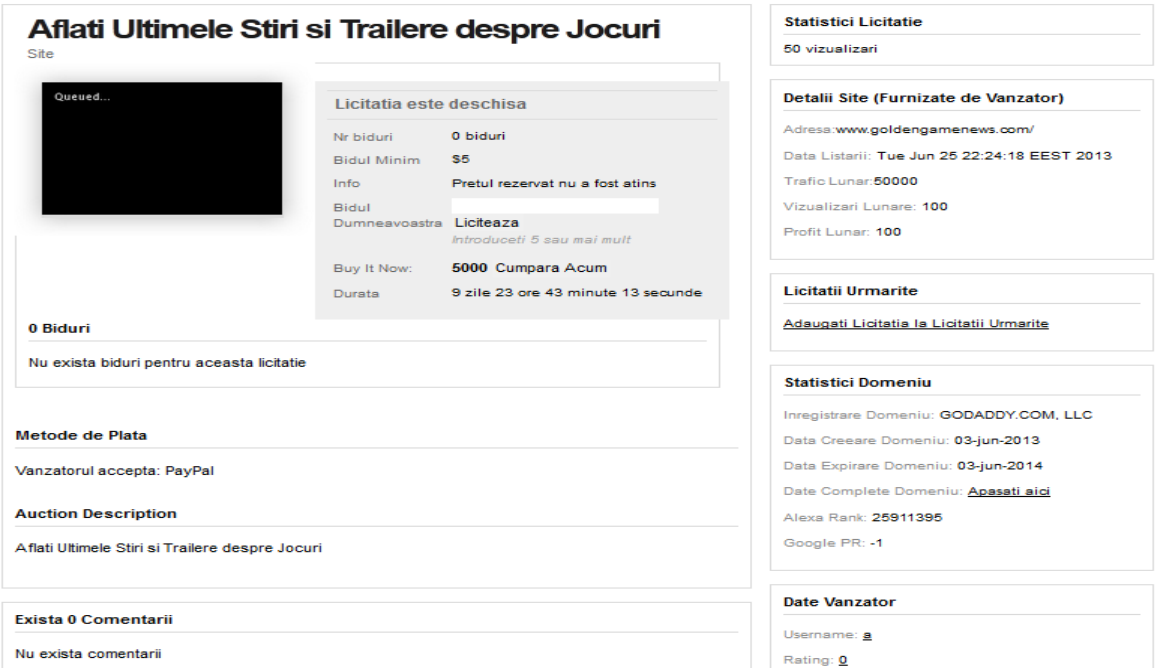

Figura 7.4 Pagina Licitaţiei

De pe pagina licitaţiilor vândute recent, se pot vizualiza toate licitaţiile care s-au terminat cu succes. Prin apăsarea pe titlul licitaţiei veţi fi duşi la pagina licitaţiei, pentru a vedea toate detaliile. Din partea dreaptă a site-ului se poate realiza căutarea licitaţiei dorite, în funcţii de anumite criterii.

Figura 7.5 Pagina Licitaţiilor Vândute Recent

De pe Pagina "*Urmăreşte Licitaţii*" se pot vizualiza toate licitaţiile urmărite de un anumit utilizator. Titlurile licitaţiilor vor fi link-uri care vor redirecţiona utilizatorii spre pagina licitaţiei, pentru a putea vizualiza mai multe detalii şi pentru a putea licita. În cazul în care o licitaţie urmărită s-a terminat, atunci titlul acesteia va fi cu o culoare ştearsă, pentru a se evidenţia de cele active.

Figura 7.6 Pagina Licitaţiilor Urmărite

Pentru a trimite mesaj, utilizatorii trebuie să acceseze pagina „Trimite Mesaj", acolo trebuie să completeze toate câmpurile necesare, iar apoi să trimită datele spre server, apăsând butonul Trimite Mesaj.

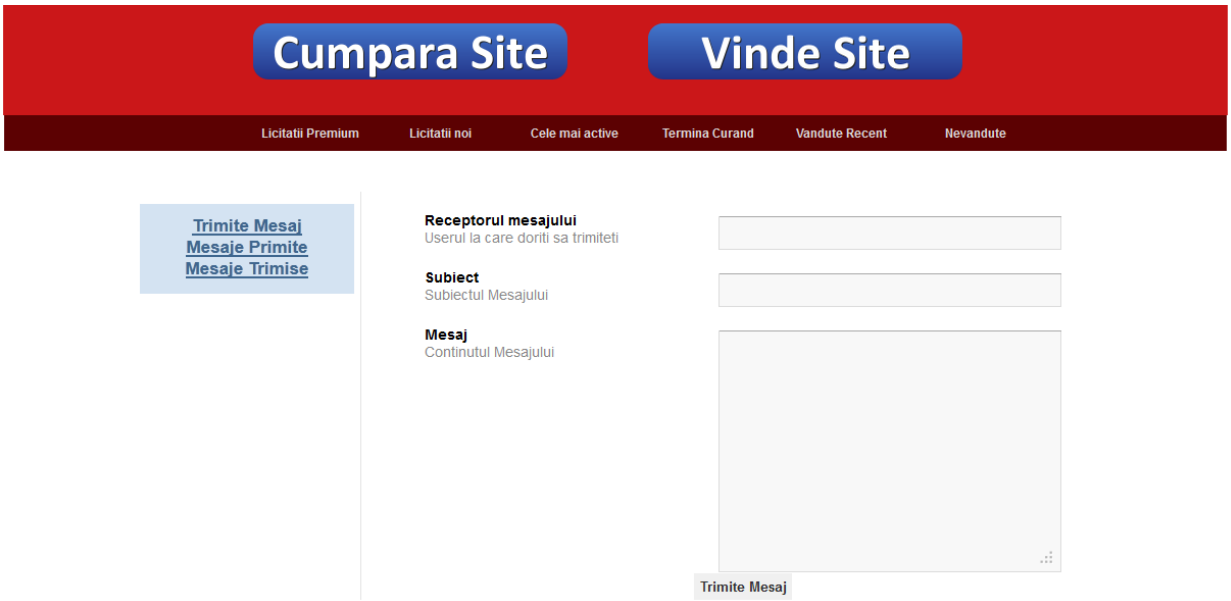

Figura 7.7 Pagina Trimite Mesaj

Pagina de tranzacţionare a site-ului conectează vânzătorul cu cumpărătorul şi le permite acestora să trimită mesaje, să vadă datele licitaţiei şi să lase feedback unul altuia.

Figura 7.8 Pagina de Tranzacţionare Site

8. Concluzii

În acest capitol voi prezenta concluziile obţinute în urma dezvoltării acestui proiect. În prima parte voi prezenta un rezumat al realizărilor care au condus la construcţia sistemului de licitaţii, iar apoi voi prezenta o serie de dezvoltări şi îmbunătăţiri care ar putea fi realizate ulterior.

8.1. Realizări

Aplicaţia implementată reprezintă un bun mod de a conecta cumpărătorii şi furnizorii de produse şi servicii. Rolul aplicaţiei se rezumă la a furniza un sistem de organizare a licitaţiile care să poată fi folosit prin intermediul Internetului de către toţi utilizatorii, indiferent de zonă geografică în care se afla. În acest mod utilizatorii nu vor mai fi nevoiţi să se deplaseze în anumite locaţii, pentru a participa la o licitaţie, sau să aştepte anumite intervale orare pentru a putea crea o licitaţie.

Datorită folosirii unor tehnologii noi, companiile care au folosit în trecut licitaţiile obişnuite se îndreaptă acum spre licitaţiile online, pentru a-şi creşte piaţa de desfacere.

Indiferent de cum priveşti situaţia, licitaţiile online reprezintă tipul de comerţ al viitorului, iar acestea vor fi analizate până când întrebările răspunse în teoria economiei tradiţionale vor fi răspunse şi în mediul virtual. Următorii ani, ne vor arăta ce tipuri de licitaţii vor supravieţui, industriile care vor lua parte în acest proces şi ce tipuri de licitaţii vor fi conduse de cumpărători sau vânzători .

Sistemul de licitaţie dezvoltat respectă toate cerinţele funcţionale şi nonfuncţionale specificate în capitolul 4 şi permite utilizatorilor să realizeze o gamă largă de operaţii, care îi vor ajuta să creeze licitaţia dorită sau să aleagă site-ul pentru care să liciteze.

Principalele servicii integrate în sistemul de licitaţie sunt:

- **PayPal Payment Data Transfer**. Utilizat pentru realizarea plăţilor de către utilizatori pentru a achita valoarea licitaţiilor câştigate şi pentru a plăti crearea licitaţiilor şi îmbunătăţirile pe care vor dori să le adauge.
- **Alexa Rank API.** Am integrat acest serviciu pentru a oferi utilizatorilor o idee despre importanţa site-urilor la un nivel global.
- **Google PageRank API.** Am integrat acest serviciu în aplicaţie pentru a arăta utilizatorilor importanţa pe care motorul de căutare Google o oferă site-ului. Cu cât este mai mare PR unui site, cu atât va fi mai important pentru motorul de căutare Google site-ul respectiv.
- **WhoIs API.** Am utilizat acest serviciu pentru a oferi utilizatorilor informaţii exacte despre domeniul unui site.
- **ThumbShot API.** Am utilizat acest serviciu pentru a afişa pe pagina licitaţiei o mică poză cu site-ul dorit a fi licitat.

Datorită integrării tuturor acestor servicii şi realizării tuturor obiectivelor specificate în capitolul 4 sistemul de licitaţii aduce următoarele beneficii utilizatorilor săi:

- Costuri reduse pentru realizarea unei licitaţii
- Crearea propriei licitaţii din propria casă, fără a exista nevoia deplasării în anumite locaţii
- Găsirea foarte uşor a site-urilor care respectă cerinţele dorite de dumneavoastră şi posibil obţinerea unui preţ mai bun decât cel de pe piaţă
- Cumpărarea unui site, fără negocieri directe între cumpărător şi vânzător

- Obţinerea unei audienţe globale, pentru licitaţiile create

Deoarece sistemul de licitaţii dezvoltat este specific licitaţiilor de tip „Silent”, este simplu de folosit şi concis în comparaţie cu celelalte tipuri de organizare a licitaţiei descrise în literatura de specialitate. Sistemul este dezvoltat folosind programarea orientată pe obiect şi am descris pentru el principalele procese folosite pentru organizarea unei licitaţii, navigarea web şi organizarea bazei de date.

8.2. Dezvoltări Ulterioare

Sistemul realizat poate fi îmbunătăţit prin utilizarea mai multor servicii şi prin îmbunătăţirea sistemului de verificare al utilizatorilor.

Pentru ca sistemul de licitaţii online să poate oferi o gamă mai largă de cerinţe funcţionale se pot integra următoarele module:

- **Verificare număr telefon**. În momentul creării unui nou cont, sistemul ar trebui să valideze numărul de telefon al acestora, înainte de a li se permite accesul. Pentru a verifica numărul de telefon sistemul, va trimite prin sms un cod unic de identificare asignat fiecărui utilizator în momentul creării contului. Utilizatorii vor trebui să trimită codul primit platformei, iar dacă va fi corect li se va permite accesul la toate funcţionalităţile sistemului.
- **Verificare Ip**. Sistemul va verifica IP-ul tuturor utilizatorilor de fiecare dată când se vor loga în sistem. În cazul în care un utilizator se logeaza într-un interval scurt de timp de pe teritoriul mai multor ţări, sistemul va afişă un mesaj de alertă pe pagina de profil al acestora. Sistemul va trebui să detecteze şi folosirea proxy-urilor, deoarece acestea permit utilizatorilor să îşi schimbe IP-ul.
- **Verificare Deţinător Site**. În momentul creării unei licitaţii, sistemul ar trebui să verifice, dacă persoana care organizează licitaţia este şi proprietarul site-ului respectiv.
- **Integrare Social Media**. Sistemul ar putea permite utilizatorilor să se logeze folosind platformele de social media Facebook şi Twitter. În acest fel nu va fi nevoie ca utilizatorii să îşi creeze un cont nou manual, deoarece contul lor va fi creat folosind datele preluate de la platformele de social media.
- **Integrare Google Analytics**. Sistemul ar trebui să permită utilizatorilor să îşi integreze propriile conturi de Google Analytics în momentul creării unei licitaţii. În acest mod, datele despre trafic ale site-ului vor putea fi accesate direct din profilul de analytics al site-ului licitat. În acest fel se asigura că datele vor fi reale.
- **Integrare Google Adsense**. Sistemul ar trebui să ofere utilizatorilor posibilitatea să îşi integreze propriile conturi de Adsense, pentru a putea obţine automat datele privind profitul obţinut prin programul Adsense de către site-ul licitat.

Extinderea sistemului pentru dispozitivele mobile, ar trebui realizat pe viitor, deoarece acestea sunt într-o continuă dezvoltare, iar numărul de utilizatori continua să crească într-un ritm alert. Pentru a realiza acest lucru, va trebui realizată o redimensionare a tuturor imaginilor şi structurilor folosite, pentru a fi compatibile cu dimensiunea telefoanelor mobile şi o reorganizare a afişării elementelor pe pagină pentru o utilizare mai uşoare. În momentul în care un utilizator vizitează site-ul, sistemul va verifica tipul de sistem folosit: dacă va fi sistem mobil va fi redirecţionat spre varianta pentru mobil a site-ului, iar dacă va fi sistem desktop, va fi redirecţionat spre site-ul obişnuit.

Permiterea accesării platformei de pe dispozitivele mobile va aduce următoarele beneficii:

- **Comoditate**. Utilizatorii pot folosi sistemul aflaţi în mişcare şi din orice locaţie
- **Intimitate.** Telefoanele mobile sunt mai private decât calculatoarele şi sunt tot timpul la îndemână.
- **Simplitate şi Viteza**. Deoarece licitaţiile necesită o cantitate limitată de informaţii, este relativ uşor pentru a le adapta pentru dispozitivele mobile

Limitările sistemelor mobile se datorează calităţii vizuale, datorită ecranului mai mic, capacităţii de memorie şi securitatea, în momentul folosirii reţelelor wireless.

Pe lângă toate acestea sistemul ar trebui să poată recunoaşte şi combate în mod automat **încercările de abuzare şi fraudare a licitaţiilor**. Fraudă se poate detecta prin analizarea tuturor informaţiilor despre utilizatori şi identificarea asimetriei dintre vânzător şi cumpărător.

Principalele metode de fraudă care ar trebui combătute sunt:

- **Falsificarea Bid-urilor**. Înţelegerea offline cu un alt utilizator să liciteze doar pentru a creşte preţul licitaţiei
- **Falsificarea Rating-ului**. Înţelegerea cu alţi utilizatori pentru a creşte rating-ul vânzătorului
- **Ameninţare pentru Feedback**. Ameninţarea utilizatorilor cu feedback negativ în schimbul unor beneficii
- **Neplata după câştigarea licitaţiei**
- **Licitarea unei sume mari, iar apoi neplata pentru licitaţie**
- **Neperformanta Tranzacţiei**. Acceptarea plăţii pentru un site, iar apoi evitarea transferului către noul cumpărător

9. Bibliografie

[1] **Haefel Richard Monson**, *"Enterprise JavaBeans Fundamentals"*. [Online]. Disponibil: http://www.freejavaguide.com/ejb.pdf

[2] **PARSONS SIMON, JUAN A. RODRIGUEZ-AGUILAR, KLEIN MARK**, "*Auctions and bidding: A guide for computer scientists*". 2011 [Online]. Disponibil: http://www.sci.brooklyn.cuny.edu/~parsons/projects/mech-design/publications/bluffers-final.pdf

[3] **Salomie Ioan, Cioară Tudor, Anghel Ionuț, Salomie Tudor**, „*Distributed Computing and Systems a Practical Approach"***,** Ed. Albastră, 2008.

[4] **Sashi C.M.**, **O'Leary Bay**, "*The role of Internet auctions in the expansion of B2B markets*", 2000. [Online]. Disponibil: http://www.itu.dk/~rold/economics/Sashi.pdf

[5] **Trevathan Jarrod, Read Wayne, Balingit Rodel**, "*Online Auction Software Fundamentals*", 2011. [Online]. Disponibil: http://www.ipcsit.net/vol2/47-B128.pdf

[6] **Turban Efraim**, "*Auctions and Bidding on the Internet*", 1997. [Online]. Disponibi: http://www.electronicmarkets.org/fileadmin/user_upload/doc/Issues/Volume_07/Issue_04/V07I4_Auctions_and_Bidding_on_the_Internet__an_Assessment.pdf

[7] **Turban E.** , *More on Electronic Auctions*, 2010 [Online], Disponibil: http://wps.prenhall.com/wps/media/objects/260/267260/online_appendices/Turban-Appendix2A.pdf

[8] **Wahli Ueli**, **Fielding Mitch**, **Mackown Gareth**, **Shaddon Deborah**, **Hekkenberg Gert**, "*Servlet and JSP Programming*", 2000. [Online]. Disponibil: http://www.redbooks.ibm.com/redbooks/pdfs/sg245755.pdf

[9] "Web MVC framework". [Online]. Disponibil: http://static.springsource.org/spring/docs/2.5.6/reference/mvc.html

[10] "*Oracle Fusion Middleware Developer's Guide for Oracle TopLink*". [Online]. Disponibil: http://docs.oracle.com/cd/E14571_01/web.1111/b32441/toc.htm

[11] "*Alexa Web Information Service*", 2005. [Online]. Disponibil: http://docs.aws.amazon.com/AlexaWebInfoService/latest/awis-dg-20050711.pdf

[12] *Jenkins hash function Wikipedia Section*. [Online]. Disponibil: http://en.wikipedia.org/wiki/Jenkins_hash_function

[13] "*Instant Payment Notification Guide".* [Online]. Disponibil: https://cms.paypal.com/cms_content/GB/en_GB/files/developer/IPNGuide.pdf

[14] *Whois Wikipedia Section*. [Online]. Disponibil: http://en.wikipedia.org/wiki/Whois

[15] "*Thumbnail Documentation*". [Online]. Disponibil: http://webthumbnail.org/how-it-works

Anexa 1. Acronime

API – Application Programming Interface
B2B – Business-to-Business
CRUD – Create Update Delete
EJB – Enterprise Java Beans
HTML - Hyper Text Markup Language
HTTP - HyperText Transfer Protocol
IDE - Integrated Development Environment
JAVA EE – Java Enterprise Edition
JDBC - Java DataBase Conncetivity
JDK - Java Development Kit
JSP – Java Server Pages
JSON – JavaScript Object Notation
MB – Megabyte
MVC – Model View Controller
PDT – PayPal Data Transfer
PR – PageRank
RAM - Random Access Memory
SDK - Software Development Kit
TCP – Transmission Control Protocol
UML - Unified Modeling Language
URL - Uniform Resource Locator
XML - EXtensible Markup Language
XSD - XML Schema Definition

Anexa 2. Glosar de Termeni

Bid = suma licitată în cadrul unei licitații active.

JavaBean = clase serializabile, care încapsulează mai multe obiecte în unul singur și permit accesul prin metode getter și setter.

Tag = termen non-ierarhic asociat unei părti a informației

Framework(software)= structură conceptuală ce reprezintă o arhitectură de software care modelează relațiile generale ale entităților domeniului

Alexa Rank = o evaluare a site-ului în funcție de restul site-urilor existente, luând în calcul date legate de trafic din ultimele 3 luni.

PageRank = algoritm de analiză a hiperlegăturilor din Internet, fiind folosit de motorul de căutare Google, pentru a acorda o pondere fiecărui element dintr-o mulțime de documente interconectate

Whois = protocol folosit pentru a interoga bazele de date care conțin informații despre utlizatori sau resurse ale Internet-ului

Checksum = numărul biților dintr-o transmisie, fiind și el inclus în transmisie. Folosit pentru detectarea erorilor.

Funcții Hash = algoritm care mapeaza data de lungime variabilă în dată de lungime fixă.

Anexa 3. Lista de figuri și tabele

Capitolul 1 - Introducere	-
Capitolul 2 – Obiectivele proiectului	-
Capitolul 3 – Studiu bibliografic	Tabel 3.1 Comparaţie între EwebsiteAuction şi Website Broker Tabel 3.2 Comparaţie între EWebsiteAuction şi DigitalPoint Marketplace Tabel 3.3 Comparaţie între EWebsiteAuction şi Flippa
Capitolul 4 – Analiză și fundamentare teoretică	Figura 4.1 Container EJB (preluată din [1]) Figura 4.2 Ciclul de viaţă al paginii JSP la prima invocare (preluată din [8]) Figura 4.3 Fluxul de lucru al procesării cererilor de către MVC (prelută din [9]) Figura 4.4 Ierarhia context Spring MVC (preluată din [9]) Figura 4.5 Arhitectura TopLink(preluată din [10]) Figura 4.6 Arhitectura unei aplicaţiei care foloseşte TopLink (preluată din [10]) Figura 4.7 Rezultatul apelului Alexa API Figura 4.8 Mesaj cu răspunsul primit de la Paypal Figura 4.9 Model Funcţionare PDT Figura 4.10 Rezultatul unui apel al API-ului WhoIs Figura 4.11 Diagrama Cazuri de Utilizare pentru Utilizator Nelogat Figura 4.12 Diagrama Cazuri de Utilizare pentru Utilizator Logat Figura 4.13 Diagrama Cazuri de Utilizare pentru Creatori Licitaţie Figura 4.14 Diagrama Cazuri de Utilizare pentru Administrator Figura 4.15 Diagrama Cazuri de Utilizare pentru Sistem Figura 4.16 Diagrama cazurilor de utilizare pentru logare utilizator Figura 4.17 Diagrama cazurilor de utilizare pentru creare licitaţii Figura 4.18 Diagrama cazurilor de utilizare pentru afişare licitaţii Figura 4.19 Diagrama cazurilor de utilizare pentru amanageria o licitaţie
Capitolul 5 – Proiectare de detaliu și implementare	Figura 5.1 Diagrama Conceptuală Figura 5.2 Mapare URL la Controller Figura 5.3 Setări Fişier web.xml Figura 5.4 Diagrama de clase pentru pachetul Controller Figura 5.5 Realizare Interfaţa Remote

	Figura 5.6 Creare Bean Stateless Figura 5.7 Creare Entitate Figura 5.8 Setare fişier glassfish-resources.xml Figura 5.9 Diagrama de clase pentru EJB Figura 5.10 Diagrama de Deployment Figura 5.11 Structura Generală a Bazei de Date Figure 5.12 Diagrama de Secvenţe pentru afişare utilizatori Figure 5.13 Diagrama de Secvenţe pentru plasare Bid
Capitolul 6 – Testare și validare	Figure 6.1 Rezultatul testelor pentru clasa AuctionController Figure 6.2 Rezultatul Testelor pentru clasa BidController Figure 6.3 Rezultatele Testelor pentru clasa SaleController Figure 6.4 Test pentru verificarea link-urilor de Navigare Figure 6.5 Test pentru verificare funcţionare pentru Pagina Licitaţiei Figure 6.6 Test pentru verificare zona Administrator Figura 6.7 Rezultatele execuţiei testelor pentru interfaţa Figura 6.8 Analiza resurselor folosind profiler-ul YourKit
Capitolul 7 – Manual de instalare și utilizare	Figura 7.1 Pagina de Login a sistemului Figura 7.2 Câmpurile necesare pentru crearea unei licitaţii Figure 7.3 Realizare Plată Taxe Licitaţie Figura 7.4 Pagina Licitaţiei Figura 7.5 Pagina Licitaţiilor Vândute Recent Figura 7.6 Pagina Licitaţiilor Urmărite Figura 7.7 Pagina Trimite Mesaj Figure 7.8 Pagina de Tranzacţionare Site
Capitolul 8 – Concluzii	-