



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

ArtBidding - Platformă de licitații pentru obiecte de artă

LUCRARE DE LICENȚĂ

Absolvent: **Daniel-Vasile POP**

Coordonator Șef lucr. ing. Cosmina IVAN
științific:

2017



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE**

DECAN,
Prof. dr. ing. Liviu MICLEA

DIRECTOR DEPARTAMENT,
Prof. dr. ing. Rodica POTOLEA

Absolvent: **Daniel-Vasile POP**

ArtBidding - Platformă de licitații pentru obiecte de artă

1. **Enunțul temei:** Proiectați un sistem care simulează o casă de licitație și gestionează produse din domeniul artei.
2. **Conținutul 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:** Universitatea Tehnică din Cluj-Napoca
4. **Consultanți:** Șef lucr. ing. Cosmina Ivan
5. **Data emiterii temei:** 19 Martie 2017
6. **Data predării:** 14 Iulie 2017

Absolvent: Daniel-Vasile Pop

Coordonator științific: Asist.Prof.Ing. Cosmina Ivan

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE****Declarație pe proprie răspundere privind
autenticitatea lucrării de licență**

Subsemnatul _____, legitimat cu
_____ seria _____ nr. _____
CNP _____, autorul lucrării

_____ elaborată în vederea susținerii
examenului de finalizare a studiilor de licență la Facultatea de Automatică și
Calculatoare, Specializarea _____ din cadrul
Universității Tehnice din Cluj-Napoca, sesiunea _____ a anului universitar
_____, declar pe proprie răspundere, că această lucrare este rezultatul propriei
activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse
care au fost citate, în textul lucrării, și în bibliografie.

Declar, că această lucrare nu conține porțiuni plagate, iar sursele bibliografice au
fost folosite cu respectarea legislației române și a convențiilor internaționale privind
drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte
comisii de examen de licență.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile
administrative, respectiv, *anularea examenului de licență*.

Data

Nume, Prenume

Semnătura

Cuprins

Capitolul 1. Introducere	7
1.1. Contextul general.....	7
1.2. Contextul proiectului	8
1.3. Contribuții personale	9
Capitolul 2. Obiectivele proiectului.....	10
Capitolul 3. Studiu bibliografic	12
3.1. Conceptul de licitație	13
3.2. Sisteme similare.....	14
3.2.1. Ebay	15
3.2.2. QuiBids	16
3.2.3. ArtnetAuctions.....	17
3.2.4. Artmark.....	18
Capitolul 4. Analiză și fundamentare teoretică	20
4.1. Analiză.....	20
4.1.1. Cerințe funcționale	20
4.1.2. Cerinte non-funcționale	22
4.1.3. Cazuri de utilizare.....	23
4.2. Fundamentare teoretică.....	30
4.2.1. Hibernate Framework	31
4.2.2. Spring MVC	32
4.2.3. Serverele Tomcat și Glassfish	33
4.2.4. AngularJs	34
4.2.5. HTML, CSS și Javascript	35
4.2.6. Bootstrap.....	35
4.2.7. Paypal API.....	36
4.2.8. Google Maps API	37
4.2.9. Git	37
Capitolul 5. Proiectare de detaliu și implementare	39
5.1. Arhitectura conceptuală	39
5.2. Nivelul de date	40
5.3. Nivelul de model.....	42
5.4. Nivelul de controller	44

5.5. Nivelul de prezentare	48
5.6. Componente	49
5.6.1. Descriere	49
5.6.2. Interacțiune	50
5.7. Diagrama de deployment	51
5.8. Implementare proprie	52
Capitolul 6. Testare și Validare	54
Capitolul 7. Manualul de instalare și utilizare	56
7.1. Manualul de instalare	56
7.2. Manualul de utilizare	58
Capitolul 8. Concluzii	61
8.1. Realizări	61
8.2. Analiza	62
8.3. Dezvoltări ulterioare	62
Capitolul 9. Bibliografie	64
Anexa1. Tabele și figuri	66
Anexa2. Glosar de termeni	67

Capitolul 1. Introducere

Licitația reprezintă punerea în vânzare de bunuri sau servicii, după anumite reguli speciale, având ca finalitate atribuirea obiectului respectiv persoanei care a oferit prețul cel mai mare. Originea licitațiilor se regăsește încă din Roma antică, romanii fiind primii care au organizat vânzarea mărfurilor prin licitație, modalitate prin care dispuneau și comercializau bunurile. Totodată, romanii vindeau în cadrul licitațiilor bunurile confiscate în timpul războiului. În zilele noastre, atât instituțiile statului, cât și companii private sau oamenii de rând, inițiază licitații având o gamă largă de produse și servicii.

Licitațiile clasice au ajuns treptat să fie devansate de licitațiile în mediul virtual, motiv al dezvoltării continue a tehnologiei și a produselor necesare aferente licitațiilor online. Licitațiile online au contribuit la dezvoltarea mediului de afaceri, punând la dispoziție bunuri sau servicii într-un mod mult mai eficient și convenabil pentru client, dar și pentru vânzător. Contextul actual presupune realizarea unor strategii de dezvoltare astfel încât produsul finit să corespundă normelor actuale și să atragă cât mai mulți potențiali clienți.

Sistemele de licitații online existente reprezintă un mediu virtual în care atât vânzătorul, cât și participanții la licitații au la dispoziție o multitudine de informații care să elimine incertitudinile legate de veridicitatea licitației, a persoanelor implicate și modul de organizare.

În cadrul licitațiilor online, se ține o evidență asupra produselor puse în vânzare, a istoricului și evoluției licitației, precum și a participanților, pentru a putea fi îmbunătățită ulterior. Aceste entități sunt salvate în baza de date pentru a putea fi gestionate mai ușor, iar pentru gestiunea licitației în sine se generează rapoarte care constau în evoluția sistemului și a componentelor care interacționează în cadrul acestuia.

1.1. Contextul general

Contextul actual presupune o lume în care domeniul informatic e în plină expansiune, iar prin intermediul internetului, majoritatea oamenilor au acces la o multitudine de informații. Majoritatea bunurilor sau serviciilor au ajuns treptat să facă tranziția de la o expunere pur fizică, în care cumpărătorul își prezintă marfa direct clienților interesați într-un mediu specific (piață, instituție publică, magazin), la mediul virtual, unde sunt mai accesibile unei game variate de potențiali cumpărători.

Bunurile sau serviciile expuse online depășesc granița fizică, acestea putând fi achiziționate de către orice persoană, fără nicio restricție referitoare la zona geografică. Astfel, acest domeniu este în continuă expansiune, atragând din ce în ce mai mulți cumpărători, oferindu-i un caracter dinamic și concurențial.

Domeniul online reprezintă un mediu vast și captează interesul unui număr mai mare de oameni pe zi ce trece. Deși reprezintă un mediu în care majoritatea datelor de contact sunt expuse atât către vânzător, cât și către cumpărător, este necesară și o doză de prudență pentru a evita unele neplăceri. Platformele performante presupun o securitate sporită și o multitudine de pași pentru a se asigura de veridicitatea datelor unei persoane.

1.2. Contextul proiectului

Licitațiile din domeniul online reprezintă o alternativă mai rapidă și eficientă a licitațiilor clasice, în care participanții la licitație concurează între ei pentru a spera la achiziționarea produsului la un preț cât mai mic. Aceste licitații presupun o transpunere a licitațiilor tradiționale în mediul virtual, în care clienții au acces la licitație prin intermediul internet-ului și accesând site-ul corespondent, iar vânzătorul e reprezentat de casa de licitații, parctic platforma pe care rulează întreaga aplicație.

Licitația pe care doresc să o implementez constituie o **casă de licitații ArtBidding** care gestionează obiecte de artă. Aceasta se încadrează în categoria de **licitații deschise**, la care poate lua parte oricine. Prețul în cadrul licitației este crescător, miza oferită de următorul licitant trebuie să depășească oferta anterioară, numărul de persoane care pot lua parte simultan la licitație este nelimitat, iar câștigătorul este acea persoană care a avut oferta cea mai mare, iar plata o constituie suma aferentă unei oferte din istoricul licitației curente.

Avantajele unei astfel de abordări ale licitației sunt variate, având efecte benefice atât asupra vânzătorului, cât și asupra cumpărătorului. Mai jos voi enumera principalele avantaje, care sunt caracteristice acestui model.

Existența mai multor participanți: Acest fapt constituie un avantaj al vânzătorului, care astfel spera la un preț de vânzare ridicat al produsului. Participanții sunt atrași de produsul pus la licitație, iar casa de licitații oferă un mediu comod, rapid și sigur de tranzacționare.

Clienții pot deveni vânzători: Clienții pot propune la licitație anumite produse de artă proprii, iar un admin va procesa cererea și va decide dacă acceptă sau nu punerea produsului în cadrul unei licitații și va stabili detalii degate de prețul produsului, data plasării și alte detalii aferente.

Existența unui cadru global: Existența unui cadru global, în care nu există limitări geografice, constituie un motiv important pentru care licitațiile online au devenit atât de populare. Prin intermediul internetului și a unei platforme care gestionează licitațiile, vânzătorul și cumpărătorul sunt puși în legătură, astfel conturându-se conceptul de licitație. Avantajele unui cadru global sunt determinate de evitarea costurilor de deplasare, precum și comoditatea oferită participantului, care astfel își poate canaliza energia asupra licitației în sine.

Verificarea datelor despre casa de licitații: Casa de licitații dispune de o pagină de contact în care sunt expuse date despre aceasta precum regimul de funcționare, regulile folosite, sediul existent, data înființării, etc

Detalii despre produsele puse în vânzare: Produsele puse în vânzare sunt expuse utilizatorului pentru ca acesta să știe exact pentru ceea ce licitează. În plus, clientul are la dispoziție opțiune de căutare a unui produs sau a unei categorii de licitații.

Salvarea datelor: Datele din cadrul aplicației sunt salvate în baza de date pentru a putea fi ulterior prelucrate. Persistența datelor este necesară pentru a monitoriza istoricul și desfășurarea licitației, precum și pentru dezvoltarea ulterioară a acesteia.

Metoda sigură de plată: Clientul care castigă o licitație poate folosi metoda sigură de plată prin Paypal, acesta fiind redirecționată pe pagina de login a celor de la Paypal unde este completată suma și produsul achiziționat, acesta trebuie doar să se logheze și să inițieze plata, sau să-și facă un cont nou dacă este cazul.

Trimiterea unui email de informare: Clienții sunt informați de licitațiile care urmează să se desfășoare prin intermediul unui email, în care sunt prezentate produsele care urmează să se liciteze, prețul de pornire și data licitației. Totodată, informarea clienților câștigători se face tot prin intermediul unui email.

Generarea rapoartelor: Rapoartele reprezintă o sursă de monitorizare a flow-ului aplicației, precum și o modalitate de expunere a unor informații care vor putea fi folosite pentru a eficientiza licitațiile viitoare.

Atragerea clienților: Această casă de licitații atrage clienții prin limitarea constrângerilor (de natură geografică), dar și prin abordarea unui nou tip de licitație, care nu mai este prezent pe piață. Aceasta constă în faptul că persoana care a câștigat licitația prin plasarea celei mai mari oferte, va plăti suma aferentă unei oferte din istoricul de oferte al licitației câștigate.

1.3. Contribuții personale

Casa de licitații reprezintă o platformă care expune lucrări de artă și oferă publicitate artiștilor care le-au realizat, cu scopul de crea un mediu destinat cumpărării și vânzării acestora.

Contribuțiile personale asupra implementării casei de licitații sunt următoarele:

- **Analiza sistemelor similare.** Analiza sistemelor asemănătoare presupune o trecere în revistă a conceptelor și practicilor folosite pentru a crea un ansamblu cât mai performant, astfel putând fi construită o aplicație care să extindă funcționalitățile sistemelor similare sau să le îmbunătățească.
- **Proiectare.** Proiectarea sistemului presupune asamblarea acestuia conform unor cerințe funcționale și non-funcționale atent alese pentru a putea fi mapate pe sistemul dorit.
- **Implementare.** Implementarea constă în respectarea normelor actuale, a unor standarde regăsite în sistemele similare și integrarea unor servicii externe cu scopul de a realiza o aplicație modernă, mentenabilă, care poate fi dezvoltată ulterior pentru a oferi utilizatorilor o experiență din ce în ce mai bună.
- **Testare.** Testarea se referă la verificarea componentelor sistemului (unit test), precum și funcționalitatea când acestea sunt asamblate într-o singură unitate (integration test).
- **Validare.** Validarea presupune controlul anumitor date pe care utilizatorul le poate introduce în cadrul aplicației, aceasta fiind necesară pentru a evita deteriorarea aplicației și pentru a oferi utilizatorului o experiență cât mai bună.

Capitolul 2. Obiectivele proiectului

Scopul acestui proiect constă în proiectarea unei **case de licitații ArtBidding** care să gestioneze licitații pentru anumite obiecte din domeniul artei. Prin acest proiect se urmărește eficientizarea vânzării acestor produse, precum și economisirea timpului prețios al clienților și al vânzătorului prin transpunerea licitației în mediul virtual.

Clienții și vânzătorul au la dispoziție o multitudine de operații pe care le pot regăsi în cadrul aplicației, și anume:

- **Înregistrare.** Aplicația va permite clienților interesați de produsele de artă existente, să se înregistreze în cadrul aplicației, iar apoi să liciteze pentru produsele dorite.
- **Logare.** Clienții se loghează și vor fi redirecționați către pagina proprie, unde își pot vizualiza detalii legate de cont și pot naviga în continuare către alte pagini.
- **Licitare de pe orice dispozitiv.** Posibilitatea clienților să liciteze de pe orice dispozitiv (laptop, PC, telefon, tabletă) care dispune de conexiune la internet, deoarece aplicația este construită în așa fel încât să se plieze și să se comporte similar pe orice dispozitiv
- **Trimitere mail.** Adminul are la dispoziție posibilitatea de a trimite mail clienților pentru a le prezenta o licitație, produsele din cadrul acesteia și alte detalii specifice. Totodată, câștigătorul licitației va primi de asemenea un mail în care e informat că a câștigat un anumit produs și plata pe care trebuie să o onoreze.
- **Căutarea unei licitații sau a unui produs.** Clienții pot căuta o licitație de care sunt interesați, specificând categoria din care face parte sau pot realiza o căutare avansată introducând numele unui produs asupra căruia doresc să liciteze.
- **Crearea licitației.** Adminul poate crea o licitație și poate introduce obiectele de artă în sistem, precum și detalii despre acestea. Totodată, acesta poate actualiza sau șterge anumite date referitoare la o anumită licitație.
- **Management useri, produse și licitații.** Managementul datelor din aplicație sunt de asemenea gestionate de către admin, care poate efectua operații de creare, actualizare sau ștergere.
- **Realizare rapoarte.** Rapoartele reprezintă o modalitate de monitorizare a sistemului, precum și o sursă de expunere a unor informații care vor putea fi folosite ulterior pentru a-i oferi utilizatorului o experiență mai plăcută.
- **Cerere creare licitație.** Clienții au de asemenea posibilitatea de a propune plasarea în cadrul unei licitații a unor obiecte de artă, însă este necesară aprobarea adminului pentru ca această operație să se efectueze cu succes.
- **Plata sigură.** Clienții câștigători au la dispoziție mijloace sigure de plată, folosind serviciul extern furnizat de Paypal.

Totodată, aplicația este una sigură, în care s-a pus accentul și pe confidențialitatea datelor utilizatorilor. Spre exemplu, parola clienților este criptată în cadrul bazei de date, pentru a oferi protecție în cazul atacurilor rău intenționate.

Aplicația dispune și de o serie de validări ale datelor care pot fi introduse de utilizatori, astfel, dacă utilizatorii introduc date invalide, incomplete sau care se regăsesc deja în baza de date (același username sau email identic), sunt atenționați să reintroducă date valide.

Capitolul 3. Studiu bibliografic

Acest capitol urmărește documentarea tehnologiilor, precum și a conceptelor care duc la realizarea unei aplicații performante, conform standardelor actuale și ale cerințelor funcționale și non-funcționale.

Articolul [1] prezintă un model de date conceptual care poate fi extins pentru a crea o aplicație mai complexă. Acesta poate fi mapat pe sistemul propriu, oferind un exemplu de structură a datelor în cadrul bazei de date.

În articolul [2] sunt prezentate concepte ale licitațiilor, tipuri de licitații și regulile aferente, precum și identificarea ofertelor și stabilirea echilibrului în cadrul licitației dintre vânzător și cumpărător.

Articolul [3] constă într-o lucrare din domeniul Economic și al Managementului Artei în care se descriu concepte legate de casele de licitație și artă contemporană. Gama de obiecte vândute în cadrul licitației este foarte variată. Scopul acestei lucrări constă în studiul pieței artei contemporane din perspectiva vânzării în cadrul licitațiilor și investigarea prețului estimativ al obiectelor din cadrul licitației.

Articolul [4] reprezintă o lucrare de licență care surprinde management-ul unui sistem de licitații în care vânzătorii și cumpărătorii se întâlnesc și negociază pentru aproape orice. Articolul surprinde structura unui site de licitații, care este organizat pe 3 nivele – date, aplicație și client, precum și cazurile de utilizare specifice atât pentru clienți, cât și pentru admin.

În cadrul articolului [5] se specifică de ce participanții folosesc licitația în defavoarea altei metode de tranzacționare. Totodată, se precizează că adăugarea unui timp asupra unei licitații poate influența semnificativ atractivitatea ofertanților. Articolul surprinde deciziile de participare ale ofertanților și cum interacționează aceștia cu design-ul licitației.

Articolul [6] reprezintă o lucrare în care se specifică importanța tehnologiilor în cadrul business-ului online și cum îl revoluționează. Acesta prezintă un design al unei licitații și oferă un model de bază asupra căruia pot fi implementate alte funcționalități mai complexe.

Articolul [7] surprinde un model de bază de date în care se detaliază proiectarea acestuia și prezintă caracteristici care ajută la gestionarea de rapoarte și strategii de optimizare ale aplicației.

Sursa [8] oferă o listă de produse software care au caracter de licitație, iar sursa [9] exemplifică un software din această industrie, care oferă un API care poate fi integrat în sistemul de licitație propriu.

Articolul [10] constituie o lucrare care exemplifică construcția unui site web de licitații. Acesta este scalabil, suportând un număr mare de participanți simultan în cadrul unei licitații. Articolul [11] prezintă un sistem de licitație în timp real pentru comercializarea ceaiurilor, care gestionează de asemenea un număr mare de participanți.

3.1. Conceptul de licitație

Conform sursei [12], **licitația** presupune un proces de cumpărare și vânzare de bunuri sau servicii prin plasarea acestora spre a putea fi licitate (se oferă un bid), în care se acceptă bid-uri, iar în final produsul este cumpărat de către cumpărătorul care a oferit cel mai mare bid. Participanții licitațiilor oferă o miză asupra produsului aflat la licitație, iar aceasta trebuie să fie mai mare decât miza plasată anterior de către alți licitanți. Un vânzător de licitații anunță prețul de pornire al unui produs, iar participanții la licitație pot licita luând parte efectiv la licitația propriu zisă, sau pot licita electronic, având afișată miza curentă pentru a putea licita în concordanță.

Scopul licitației îl constituie faptul că licitatorul dorește să ofere un bun unuia dintre cumpărători. Interesul persoanei care oferă produsul la licitație este acela de a maximiza prețul, iar cel al cumpărătorului e de a-l minimiza. Sursa [14] precizează că acest fapt constituie conceptul esențial al licitațiilor, dezvoltat de matematicianul John Nash, denumit “conceptul echilibrului”.

Sursa [12] precizează următoarele tipuri de licitații:

- 4 mari categorii tradiționale de licitații care pot fi aplicate asupra unui produs:

- **Licitația de tip englez (English auction)**

Acesta este cel mai comun tip de licitație. Participanții mizează deschis unul împotriva celuilalt, fiecare ofertă fiind mai mare decât cea precedentă. Cel care deschide licitația începe de la prețul de start al produsului, iar ofertele cresc succesiv până când niciun alt participant nu mai oferă o sumă mai mare. Persoana care a oferit în final cea mai mare sumă este câștigătoarea licitației. În cazul în care obiectul pus la licitație deține un preț de rezervă (un preț minim setat înainte de startul licitației), iar cea mai mare ofertă nu depășește acest preț de rezervă, produsul rămâne nevândut. Cel mai distinctiv factor al acestui tip de licitație îl reprezintă faptul că cea mai mare miză posibilă este tot timpul disponibilă potențialilor licitanți până la terminarea efectivă a licitației.

- **Licitația de tip olandez (Dutch auction)**

Acest tip de licitație este utilizat atunci când un număr mai mare de produse de același tip este pus la licitație, în cadrul căreia nu va exista un singur cumpărător, ci mai mulți, fiecare având o cantitate diferită. Licitația olandeză presupune ca fiecare participant să facă o ofertă pentru cantitatea dorită de produse, iar ofertele se ordonează în ordine descrescătoare a sumei oferite. Câștigătoare sunt primele oferte (având suma cea mai mare), până când stocul produsului e epuizat. Prețul de vânzare va fi unic pentru toți cumpărătorii, constând în valoarea celei mai mici oferte din cadrul ofertelor câștigătoare.

- **Licitația sealed first-price sau Licitația oarbă (Blind auction)**

În cadrul acestui tip de licitație, participanții oferă mize în secret. Cu alte cuvinte, fiecare licitant își cunoaște doar propria miză, neștiind valoarea unei alte oferte a unui alt participant. Participanții pot plasa o singură ofertă inițială, iar oferta cea mai mare desemnează câștigătorul licitației.

- **Licitația Vickrey (sealed-bid second-price-auction)**

Acest tip de licitație presupune plasarea ofertelor secrete, iar câștigătorul este desemnat cel care a oferit miza cea mai mare. Valoarea pe care trebuie să o plătească pentru obiectul câștigat este reprezentată de valoarea celei de-a doua cele mai bune oferte.

- Alte categorii de licitații precum:

- **licitație multiunit** – licitație în cadrul căreia se vând mai multe obiecte de același tip, evitându-se astfel o licitație separată pentru fiecare articol

- **licitație all-pay** – licitație în care toate persoanele care au licitat trebuie să plătească miza pe care au oferit-o, indiferent dacă au câștigat sau nu licitația

- **licitație “lumânare”** – licitația a fost folosită în Anglia pentru vânzarea ambarcațiunilor, în care cea mai mare miză existentă pe masă până când o lumânare arde complet, este câștigătoare

- **licitație bidding fee** – licitația presupune ca fiecare participant să plătească o sumă fixă de fiecare dată când plasează o miză. Când timpul licitației expiră, cea mai mare miză câștigă și trebuie plătit prețul final. Spre deosebire de licitațiile convenționale, prețul final este mult sub valoarea produsului, dar toți participanții trebuie să plătească pentru fiecare bid plasat

- **licitație buyout** – licitație care are un preț adițional “buyout”, care poate fi acceptat de către orice licitant în orice moment, astfel terminând licitația și câștigând produsul

- **licitația japoneză** – asemănătoare cu licitația engleză, diferența constând în faptul că odată ce licitația a început, niciun participant nou nu mai poate lua parte la ea

- **licitația misterioasă** – licitație în care participanții licitează pentru cutii sau plicuri care conțin potențiale obiecte interesante sau de valoare

- **licitația senior** – licitație în care cele mai bune 2 trebuie să plătească pentru obiectul licitat, care îi revine doar celui cu oferta cea mai mare

- **licitația silențioasă** – similară cu licitația engleză, dar ofertele sunt scrise pe o foaie de hârtie

- **licitația top-up** – licitație în care pierzătorii trebuie să plătească diferența dintre miza oferită și următoarea cea mai mică miză. Este folosită în special pentru evenimente de caritate

3.2. Sisteme similare

Sistemele similare contribuie la dezvoltarea unui sistem propriu standardizat, care poate fi dezvoltat ulterior după bunul plac. Funcționalitățile sistemelor actuale sunt analizate, comparate și vor fi integrate în sistemul propriu.

Sistemele actuale se concentrează pe transpunerea licitațiilor tradiționale în mediul virtual, oferind participanților siguranța și o experiență cât mai plăcută. Aceste sisteme au un design care presupune gestiunea mai multor clienți în cadrul aceleiași licitații, oferind totodată mecanisme care păstrează confidențialitatea datelor.

Articolul [14] conturează 6 pași importanți care trebuie efectuați pentru a realiza o licitație de succes:

- Evaluarea strategiei inițială de management
- Deciderea asupra unui model de licitație
- Construirea unui model de licitație
- Menținerea clienților interesați
- Rularea licitației propriu zise, având feedback de la participanți
- Analiza sistemului

Acești pași au fost urmăriți și în realizarea **căsei de licitații ArtBidding**, adaptați la cerințele sistemului propriu.

3.2.1. Ebay

Ebay reprezintă unul dintre cele mai vechi site-uri de licitație online și oferă o gamă variată de produse, totul de la îmbrăcăminte, la produse de artă. Cumpărătorii pot plasa o miză în cadrul licitației sau pot cumpăra produsul instant, aceștia putând deveni și vânzători prin plasarea la licitație a produselor personale.

După vizitarea site-ului, așa cum este și specificat în sursa [15], felul cum decurge o licitație online la eBay diferă destul de mult de licitațiile tradiționale. Două exemple de diferențe:

- Ofertantul/furnizorul trebuie să stabilească sfârșitul licitației sale pentru o anumită zi, oră, minut și secundă;
- Afișajul nivelului curent la care s-a ajuns diferă de ultima ofertă făcută. Regulile exacte ale afișajului sunt însă publicate pe site.

În Ebay există însă și vânzări la un preț fix, oferte permanente precum și alte variante de vanzare ale produselor. Unele companii de renume își oferă produsele și serviciile și pe platforma eBay.

Ebay dispune de o categorie specială dedicată artei, unde utilizatorii pot lua parte la licitații live sau pot licita la anumite produse din variate domenii ale artei.

Comparatie între Ebay și casa de licitații

Principalele asemănări și deosebiri între aceste 2 site-uri de licitații sunt prezentate în tabelul următor:

Caracteristici	Ebay	Casa de licitații
Pagina principală	✓	✓
Pagina de contact	✓	✓
Pagina de register	✓	✓
Pagina de login	✓	✓
Personalizare pagină	✓	X
Plasare licitație	✓	✓
Plata online	✓	✓

Vizualizare informații	✓	✓
Selecție categorie	✓	✓
Căutare produs	✓	✓
Produse de artă	Conține categorie	✓
Teritoriu	Global	Global
Generare rapoarte	X	✓
Trimitere mail	X	✓

Tabel 3.1. Comparație între Ebay și casa de licitații

În urma comparației, se observă faptul că implementarea proprie oferă în plus funcționalitatea de generare a rapoartelor și trimitere de mail către clienți. Casa de licitație se concentrează

3.2.2. QuiBids

Precum este precizat și în sursa [16], QuiBids este unul dintre cele mai mari site-uri de licitații online care funcționează pe principiul de licitație „penny”, în care licitantul plătește o sumă predefinită (0.6 \$) pentru fiecare bid, chiar dacă acesta nu este câștigător. Miza în cadrul licitațiilor variază de la între 1 și 20 de unități, în care incrementul este de obicei reprezentat de 5 unități. Acest tip de licitație presupune necesitatea utilizatorului de a plăti pentru mizele plasate chiar dacă nu este câștigător, iar pe deasupra trebuie să plătească și prețul final al obiectului. Practic suma finală plătită de utilizator este compusă din suma mizelor plasate cumulată cu suma finală a produsului. Câștigătorul are la dispoziție o săptămână pentru a-si revendica premiul, altfel acesta îl va pierde.

Comparație între QuiBids și casa de licitații ArtBidding

Caracteristici	QuiBids	Casa de licitații
Pagina principală	✓	✓
Pagina de contact	✓	✓
Pagina de register	✓	✓
Pagina de login	✓	✓
Plata online	✓	✓
Vizualizare informații	✓	✓
Selecție categorie	✓	✓
Căutare produs	✓	✓
Produse de artă	Conține categorie	✓
Teritoriu	Global	Global
Feedback	✓	✓
Generare rapoarte	X	✓
Trimitere mail	X	✓
Informare câștigător	X	✓

Tabel 3.2. Comparație între QuiBids și casa de licitații

QuiBids atrage participanții prin faptul că are o implementare a conceptului de licitație diferită față de alte site-uri, este ușor de folosit și are o interfață grafică simplă, intuitivă, care actualizează produsele în fiecare moment al licitației.

În urma analizei tabelului anterior se observă că propriul produs dispune de mai multe caracteristici și rezolvă unele neajunsuri care se regăsesc pe platforma QuiBids.

3.2.3. ArtnetAuctions

Din sursa [17] am tras concluzia că ArtnetAuctions reprezintă un site de licitații dedicat exclusiv obiectelor de artă, în care acestea sunt evaluate de niște specialiști înainte de a fi plasate la licitație. Fiecare produs plasat la licitație oferă informații referitoare la proveniența, detalii despre produs, piața de artă în care se încadrează. Cumpărătorul onorează suma corespunzătoare câștigării licitației către vânzător, iar ArtnetAuctions încasează între 5% și 15% din prețul acesta, procent care este revendicat pentru oferirea serviciilor premergătoare plasării produsului în cadrul licitației.

Comparație între ArtnetAuctions și casa de licitații ArtBidding

Caracteristici	ArtnetAuctions	Casa de licitații Artbidding
Pagina principală	✓	✓
Pagina de contact	✓	✓
Pagina de register	✓	✓
Pagina de login	✓	✓
Personalizare pagină	X	X
Plasare licitație	✓	✓
Plata online	✓	✓
Vizualizare informații	✓	✓
Selecție categorie	✓	✓
Căutare produs	✓	✓
Produse de artă	✓	✓
Teritoriu	Global	Global
Feedback	✓	✓
Generare rapoarte	X	✓
Trimitere mail	✓	✓
Informare câștigător	✓	✓
Ghid pentru câștigător	✓	X
Asigurare livrare	✓	X
Catalog virtual	✓	X

Tabel 3.3. Comparație între ArtnetAuctions și casa de licitații

ArtnetAuctions oferă participanților posibilitatea de deveni vânzători în cadrul licitației, punând la dispoziție utilizatorilor diferite metode de a-și plasa produsul în

cadrul licitației: email cu datele specifice despre produsul de artă sau prezentare la sediul central cu produsul dorit, iar experții vor face analiza.

În urma analizei tabelului anterior am ajuns la concluzia că ArtnetAuctions oferă un cadru complex de transpunere a licitațiilor, care însă duce lipsă de rapoarte amănunțite care pot îmbunătăți experiența utilizatorilor sau a unor grafice care să prezinte înclinația utilizatorilor asupra unor anumite produse din cadrul licitațiilor.

3.2.4. Artmark

Artmark constituie un cadru în care se evaluează și afișează opere de artă autentice, iar activitatea presupune atât licitații publice, cât și vânzări private. Concluzia la care am ajuns după ce am studiat sursa [18] este că Artmark reprezintă o casa de licitații cu renume din România, care a fost înființată imediat după 1989. La ora actuală, Artmark este liderul licitațiilor din România, având aproximativ 80% din cota de piață.

Comparație între Artmark și casa de licitații ArtBidding

Caracteristici	Artmark	Casa de licitații
Pagina principală	✓	✓
Pagina de contact	✓	✓
Pagina de register	✓	✓
Pagina de login	✓	✓
Personalizare pagină	X	X
Plasare licitație	✓	✓
Plata online	✓	✓
Vizualizare informații	✓	✓
Selecție categorie	✓	✓
Căutare produs	✓	✓
Produse de artă	✓	✓
Teritoriu	Global	Global
Feedback	✓	✓
Rapoarte publice	✓	X
Trimitere mail	✓	✓
Catalog virtual	✓	X
Licitație live	✓	X
Știri	✓	X
Tip de licitație	clasica	Implementare proprie

Tabel 3.4. Comparație între Artmark și casa de licitații ArtBidding

Artmark conturează un mediu de licitație foarte bine organizat, cu o structură complexă care a acaparat piața din România prin numeroasele facilități pe care le pune la dispoziția clienților.

Tabelul anterior surprinde complexitatea sistemului Artmark și multitudinea de funcționalități de care dispune. **Casa de licitații ArtBidding** proprie s-a axat pe atragerea clienților prin implementarea unei metode de licitație unică, astfel neglijând unele funcționalități suplimentare.

Pentru a implementa **casa de licitație ArtBidding** proprie, s-a pus accentul pe studiul sistemelor existente și pe realizarea principalelor funcționalități. Sistemul s-a axat pe atragerea clienților prin realizarea unei licitații mai deosebite, diferită de cele existente pe piața. Totodată, se încearcă oferirea unei experiențe cât mai plăcută a utilizatorului de pe orice dispozitiv, oferindu-i posibilitatea de a accesa site-ul prin intermediul telefonului mobil, a tabletei, laptop-ului sau a PC-ului.

Capitolul 4. Analiză și fundamentare teoretică

Acest capitol prezintă tehnologiile folosite pentru realizarea **casei de licitații ArtBidding**, precum și o comparație cu alte alternative existente pe piață. Sistemul va fi analizat atât în ansamblu, urmărind funcționalitatea fiecărei componente în parte, cât și comportamentul unitar, când componentele sunt conectate și formează întreaga aplicație. Se face o analiză atât a tehnologiilor folosite, precum și a felului cum acestea influențează buna funcționare a aplicației.

4.1. Analiză

Analiza sistemului presupune o trecere în revistă a funcționalităților de care dispune sistemul, a tipurilor de utilizatori care pot folosi aplicația, a cerințelor non-funcționale și funcționale ale sistemului, precum și cazurile de utilizare pe care utilizatorii le pot realiza.

4.1.1. Cerințe funcționale

Cerințele funcționale prezintă funcționalitățile sistemului sau ale componentelor acestuia. Obiectivul principal îl reprezintă realizarea unei **case de licitații ArtBidding** online care transpune licitațiile tradiționale în mediul online.

Principalele funcționalități ale sistemului care le pune la dispoziția utilizatorilor sunt următoarele:

- **Înregistrare și logare**
Persoanele care doresc să beneficieze de serviciile acestei **case de licitații** au la dispoziție serviciul de înregistrare unde își introduc datele personale, iar după o verificare și validare a acestora, sunt integrați în sistem. Acum devin utilizatori ai acestui site și se pot loga pentru a fi redirecționați către pagina personală, de unde pot naviga către paginile de interes și pot efectua operațiunile dorite.
- **Adaugare licitație**
Această platformă implică adaugarea unei licitații și selectarea produselor aferente, pentru care participanții vor concura pentru a plasa cea mai bună ofertă. Adăugarea licitației și a produselor se face de către utilizatorii care au dreptul de admin, aceștia putând totodată efectua operații de modificare sau ștergere. Totodată, simplii utilizatori fără drepturi speciale care doresc să ofere un produs pentru a putea fi adăugat într-o licitație, pot trimite un mesaj admin-ului precizând detaliile despre produs, iar după o analiză amănunțită, acesta decide dacă acceptă sau nu produsul. În caz afirmativ, produsul va fi plasat într-o licitație viitoare, iar în caz negativ, adminul refuză cererea, iar utilizatorul va fi informat printr-un email.

-
- **Participare la licitație**
Participarea în cadrul licitației se poate realiza de către orice utilizator al sistemului, neexistând nici un fel de constrângere. Utilizatorii pot plasa miza dorită la orice produs de artă din cadrul oricărei licitații, pe baza informațiilor puse la dispoziție și a constrângerilor sistemului. Participanții au afișat în dreptul produsului prețul actual și trebuie să plaseze o sumă care depășește acest preț. În cazul în care licitația este terminată, utilizatorul care a plasat miza cea mai mare este determinat câștigător și primește un mail în care îi este adus la cunoștință acest fapt și suma care trebuie plătită pentru produsul câștigat.
 - **Managementul produselor**
Gestiunea produselor din cadrul sistemului este realizată de către admin, care poate efectua operații de adăugare, modificare sau ștergere. Produsele sunt asociate unei licitații, fiecare produs în parte având o descriere proprie și alte detalii caracteristice.
 - **Căutarea unui produs sau a unei licitații**
Pentru a îmbunătăți experiența utilizatorilor acestei platforme, problema căutării unui anumit produs sau a unei anumite licitații este rezolvată prin adăugarea unei funcționalități de căutare avansată. Utilizatorul introduce numele produsului sau a licitației dorite, iar apoi este redirecționat către pagina de interes corespunzătoare.
 - **Trimitere mail**
Adminii au la dispoziție posibilitatea de a trimite mail utilizatorilor platformei pentru a le prezenta detalii legate de licitațiile viitoare și a le expune oferta variată de produse. Totodată, trimiterea mail-ului către câștigătorul unei licitații se efectuează automat în cadrul sistemului, acesta fiind informat despre produsul câștigat și suma pe care trebuie să o onoreze.
 - **Feedback**
Participanții la licitații pot evalua desfășurarea licitației la finalul acesteia prin acordarea unui comentariu sau rating. Aceștia pot efectua această acțiune o singură dată, la finalul licitației, astfel oferind informații care îi ajută pe alți participanți să-și facă o părere despre experiența pe această platformă.
 - **Implementare tip de licitație unic**
Această platformă rulează un tip de licitație unic, o implementare ingenioasă care este menită pentru a atrage un număr mai mare de participanți. Aceasta constă în faptul că persoana care va oferi miza cea mai mare este câștigătoare, dar va plăti pentru produsul câștigat la licitație suma corespunzătoare unei mize din istoricul de bid-uri. Astfel se urmărește atragerea clienților care, fie sunt raționali și își calculează șansele de câștig ale produsului la un preț mai mic decât cel al pieței, fie sunt dornici să riște și să spere la un preț avantajos.

- **Generare rapoarte**

Rapoartele sunt o sursă de gestionare și verificare a funcționării aplicației, precum și o modalitate de prelucrare a unor informații care vor putea fi folosite pentru a eficientiza licitațiile viitoare și a le oferi participanților o experiență mai plăcută.

4.1.2. Cerinte non-funcționale

Sistemul trebuie să respecte cerințele de ansamblu stabilite, atât cele funcționale, cât și cele non-funcționale, pentru a corela funcționalitățile dorite cu indicatorii existenți pe piață pentru gestionare sau monitorizare ale acestora.

Cerințele non-funcționale specifică criteriile care pot fi folosite pentru a monitoriza funcționarea unui sistem, mai degrabă decât descrierea componentelor. Cerințele trebuie să se mapeze pe sistemul actual și trebuie să fie clare și caracteristice pentru a putea duce la o implementare cât mai fidelă a sistemului dorit. **Casa de licitații ArtBidding** prezintă următoarele cerințe non-funcționale:

- **Performanța**

Performanța constă într-o caracteristică a aplicației care definește cantitate de informație care trebuie procesată. **Casa de licitații ArtBidding** prezintă un timp de răspuns scăzut pentru procesarea datelor, o rată de procesare ridicată și o utilizare redusă a resurselor sistemului. Performanța este necesară pentru a oferi utilizatorilor o experiență cât mai plăcută când utilizează aplicația și efectuează operații precum: logare, creare licitație, plasare miza, trimitere mesaje.

- **Mentenabilitate**

Aplicația prezentă este mentenabilă, poate fi întreținută fără eforturi substanțiale. Mentenabilitatea reiese din structura platformei și din modul de implementare, care se regăsesc în standardele actuale. Mentenabilitatea constă în corecția unor defecte, a unor flow-uri incomplete, prevenția unor condiții de funcționalitate neașteptate, maximizarea eficienței și a duratei de utilizare a produsului.

- **Scalabilitate**

Această cerință non-funcțională presupune gestionarea creșterii volumului de date și a potențialului de mărire a aplicației. Un sistem este considerat scalabil atât timp cât își extinde cantitatea de date atunci când se suplimentează resursele (de obicei hardware). Aplicația prezintă scalabilitate verticală, aceasta presupunând o creștere a volumului de date odată cu adăugarea resurselor.

- **Securitate**

Această caracteristică presupune protecția și rezistența împotriva atacurilor. Regăsirea acestei cerințe non-funcționale în aplicație se află în mecanismele securitate implementate: înregistrare, logare, precum și restricționarea utilizatorilor în funcție de rolul acestora. Totodată, utilizatorii pot accesa doar pagina proprie, iar în funcție de rol (user sau admin), pot accesa pagini care au la dispoziție funcționalități diferite.

-
- **Compatibilitate**
Aplicația este compatibilă cu sistemele actuale, iar utilizatorii o pot rula folosind orice dispozitiv (PC, laptop, tabletă, telefon). Experiența utilizatorului nu este afectată de dispozitivul folosit, platforma rulând în mod similar. Totodată, aplicația este una responsive, care se adaptează la dimensiunile ecranului.
 - **Extensibilitate**
Această cerință non-funcțională este luată în considerare pentru o viitoare dezvoltare a aplicației. Adăugarea de noi caracteristici va fi ușoară și vor putea fi integrate cu succes în cadrul unui proiect bine structurat.
 - **Uzabilitate**
Platforma are în componența sa acest atribut calitativ, care se referă la cât de ușor le este utilizatorilor să folosească această aplicație. Acest site este ușor de folosit, având o interfață intuitivă, care este adaptată la toate dispozitivele actuale. Această cerință non-funcțională mai presupune rapiditatea cu care utilizatorii învață să folosească funcționalitățile acestei aplicații, cât de bună este experiența resimțită și cât de predispuși la erori sunt când introduc date în sistem. Pentru a reduce și evita erorile, s-au conceput validări care controlează și ghidează utilizatorul.

4.1.3. Cazuri de utilizare

Pentru a folosi corect aplicația și pentru a recunoaște toate funcționalitățile de care dispune, caracterstice tipului de utilizator existent în sistem, trebuie să analizăm cazurile de utilizare, care constituie flow-ul aplicației, precum și actorii implicați. **Casa de licitații ArtBidding** contine 3 tipuri de actori: utilizatorul neînregistrat, utilizatorul înregistrat în cadrul sistemului și adminul sistemului.

Utilizatorul neînregistrat nu poate beneficia de serviciile acestei platforme, decât de acela de înregistrare, acesta putând să-si realizeze un cont. Acesta trebuie să-si introducă datele personale și după ce este acceptat în sistem, se poate loga și poate folosi facilitățile de care dispune această platformă. Utilizatorul nelogat are acces doar la pagina principală, unde poate vizualiza informații legate de **casa de licitații**.

Utilizatorul înregistrat în cadrul sistemului este reprezentat de utilizatorul din cadrul sistemului care are acces la serviciile furnizate de **casa de licitații**. Acesta poate participa la licitații, vizualiza informații despre produse și poate plasa o cerere către administrator pentru a adăuga un produs personal în cadrul licitației viitoare.

Adminul sistemului este reprezentat de un utilizator înregistrat în sistem, care are privilegii suplimentare precum: managementul licitației (adaugare licitație, adăugare produse, modificare date despre licitație, ștergere licitație), managementul utilizatorilor, acceptarea sau respingerea cererii utilizatorilor de a adăuga un produs în cadrul unei licitații, trimiterea de mail-uri de informare în legătură cu licitațiile viitoare, generarea și vizualizarea de rapoarte.

Cazurile de utilizare pentru utilizatorii neînregistrați:

1. Vizualizarea paginii principale
2. Vizualizarea paginii de contact
3. Vizualizarea produselor care au fost vândute în cadrul licitațiilor terminate
4. Posibilitatea de a se înregistra sau loga

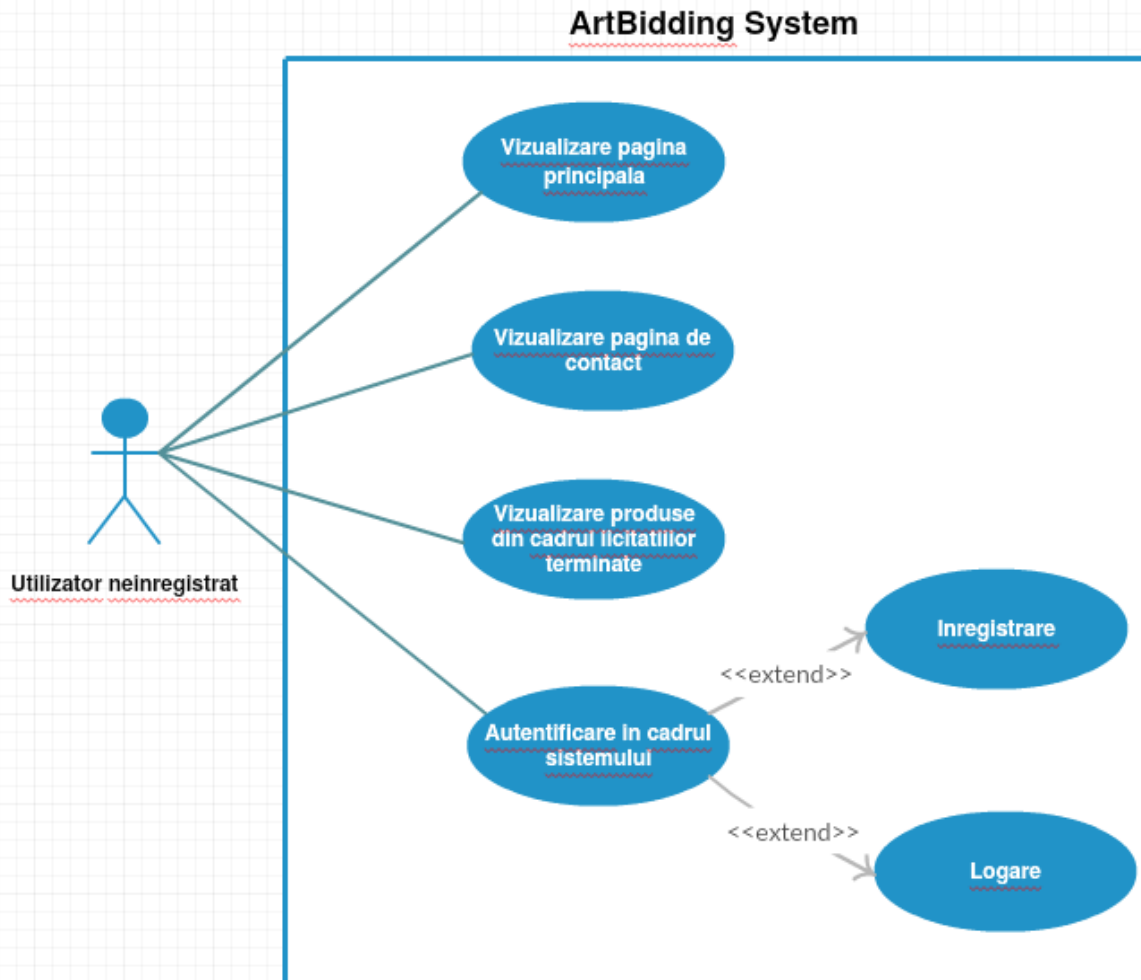


Figura 4.1. Cazuri de Utilizare pentru Utilizatorul neînregistrat

Cazurile de utilizare pentru utilizatorii logați:

5. Căutarea unui produs sau a unei licitații
6. Plasarea mizei dorite în cadrul unei licitații
7. Adăugare comentariu în cadrul unei licitații
8. Evaluare licitații (rating)
9. Efectuare plata online
10. Cerere pentru plasarea produsului propriu în cadrul unei licitații

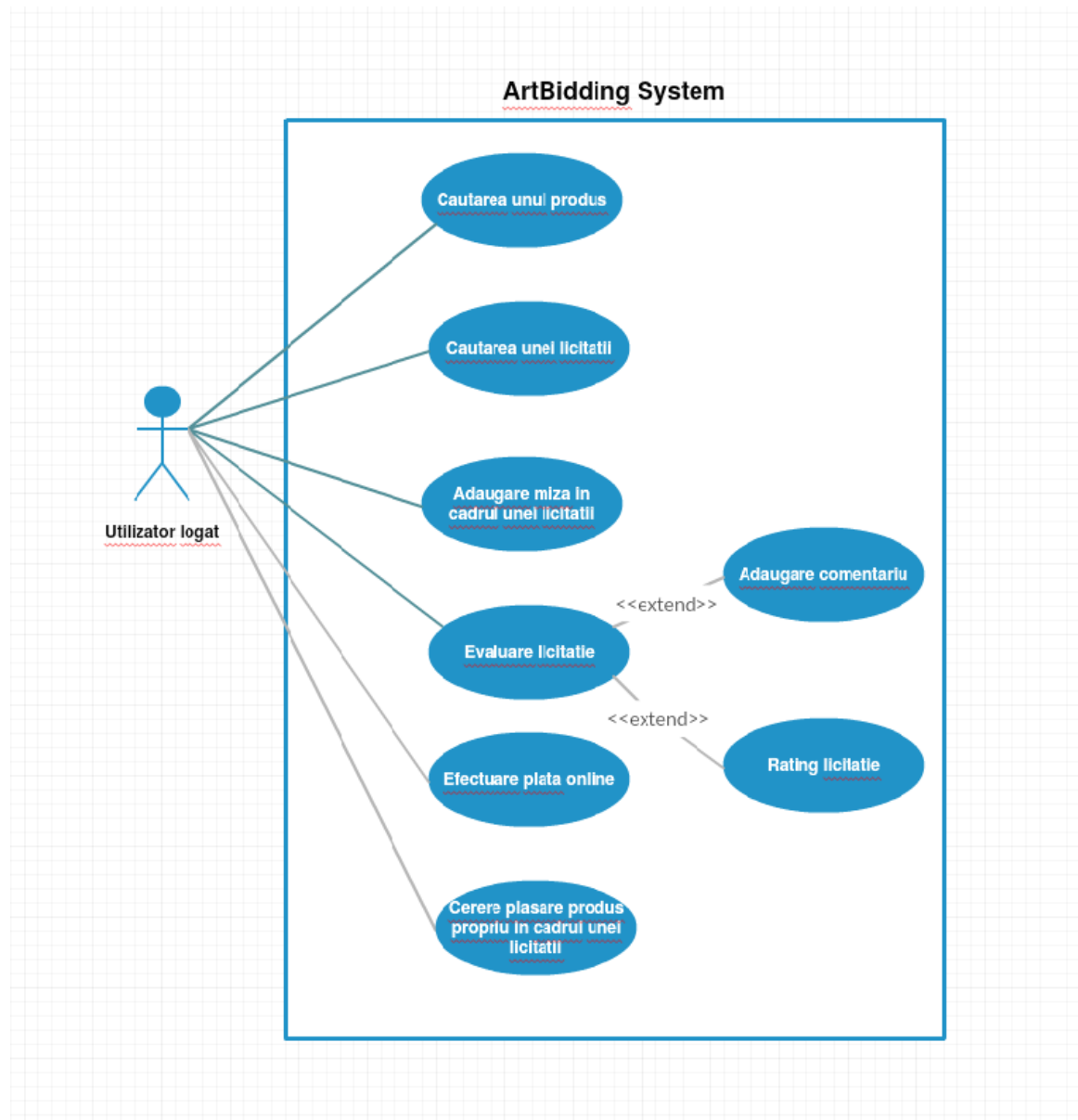


Figura 4.2. Cazuri de Utilizare pentru Utilizatorul logat

Cazurile de utilizare pentru admin:

11. Management-ul utilizatorilor
12. Management-ul licitațiilor și a produselor
13. Acceptarea sau respingerea cererii utilizatorului pentru a plasa un produs în cadrul unei licitații
14. Trimiterea mail-urilor de informare către utilizatori
15. Generarea rapoartelor
16. Schimbarea rolului unui utilizator

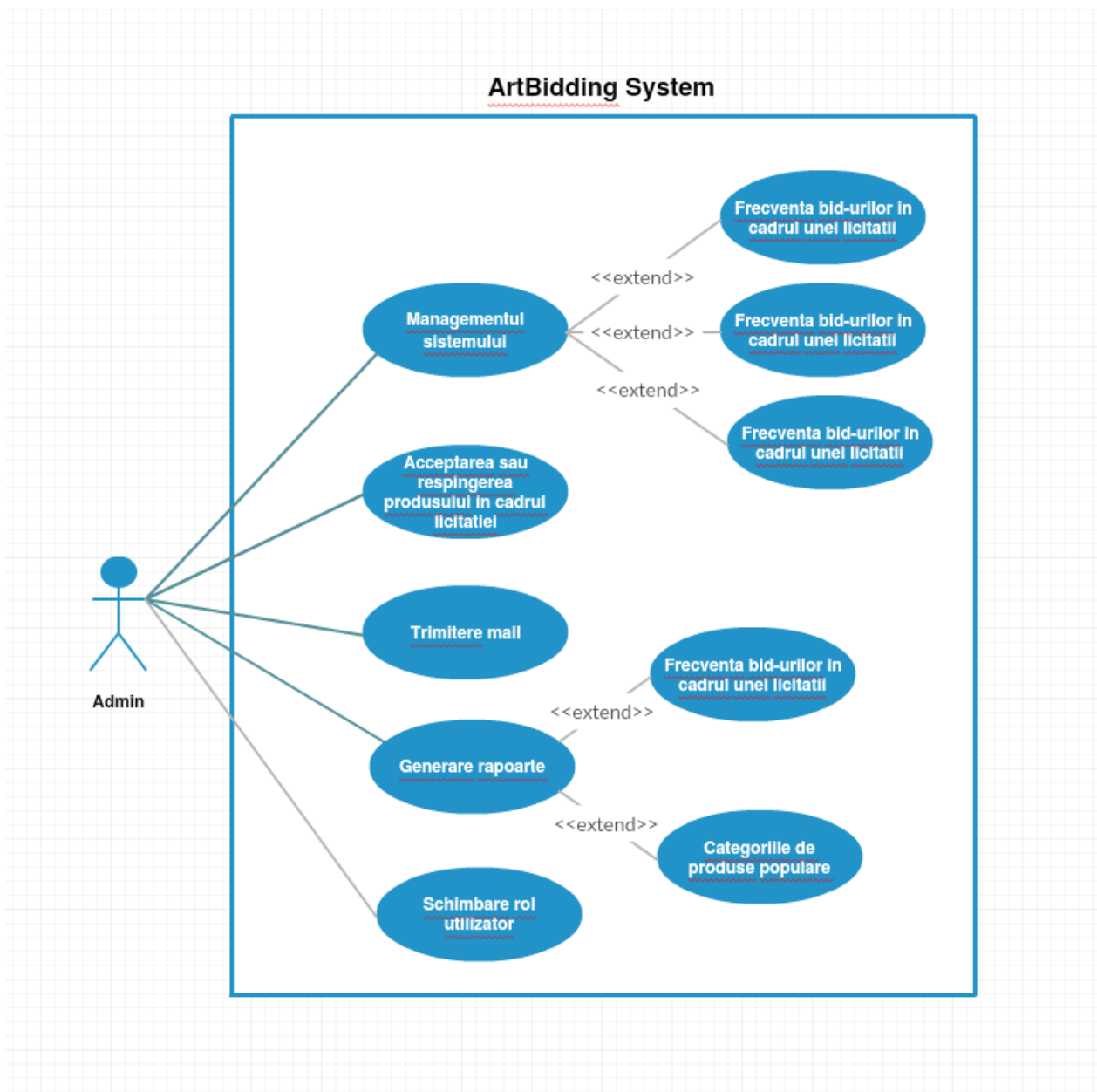


Figura 4.3. Cazuri de Utilizare pentru admin

Cazurile de utilizare ale sistemului:

- 17. Validarea contului și a datelor introduse de utilizator
- 18. Pornirea licitației
- 19. Terminarea licitației
- 20. Înregistrarea bid-urilor utilizatorilor
- 21. Trimiterea mail-ului de informare către câștigătorul licitației
- 22. Salvarea datelor în baza de date

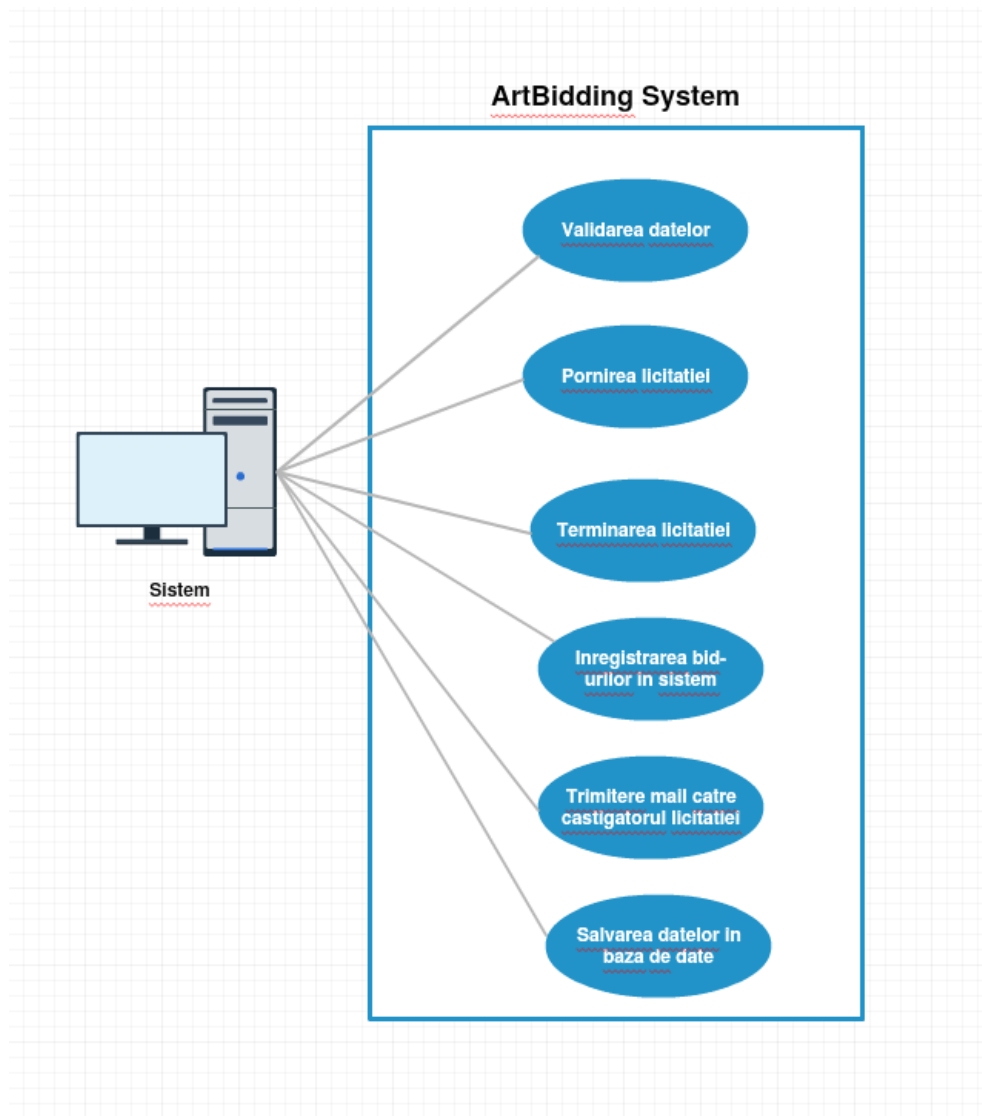


Figura 4.4. Cazuri de Utilizare pentru sistem

Exemplificarea pașilor

Caz de utilizare 1: Logarea utilizatorului

Actor: Utilizatorul

Scop: Autentificarea utilizatorului pentru a avea acces la facilitățile pe care le pune la dispoziție aplicația

Precondiții: Accesarea paginii de login de către utilizator

Postcondiții: Logarea utilizatorului în sistem și asigurarea privilegiilor specifice

Scenariu favorabil:

1. Accesarea paginii de login de către utilizator
2. Introducerea datelor în câmpurile puse la dispoziție
3. Trimiterea datelor prin apăsarea butonului “Login”
4. Prelucrarea datelor în cadrul sistemului
5. Autentificarea utilizatorului în sistem

Scenariu nefavorabil:

- 2.n. Introducerea unor date nevalide
 1. Sistemul afișează un mesaj corespunzător

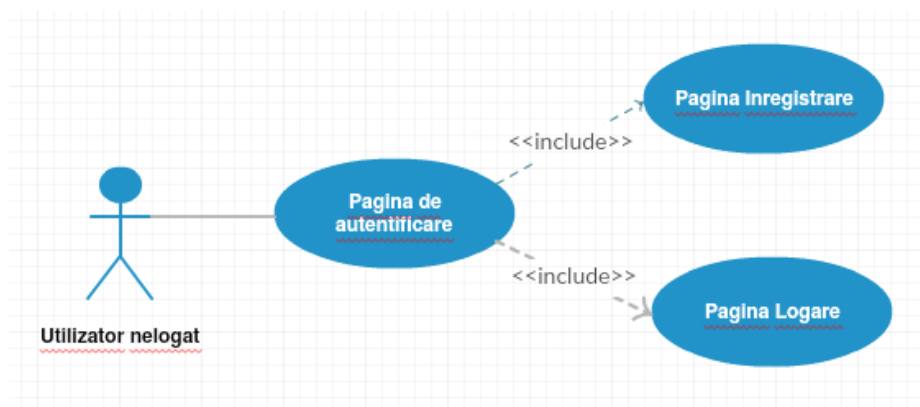


Figura 4.5. Scenariu caz de utilizare pentru Utilizatorul neînregistrat

Caz de utilizare 2: Crearea unei licitații

Actor: Adminul

Scop: Plasarea unei licitații pentru ca utilizatorii să poată licita asupra produselor dorite

Precondiții:

- Logarea adminului în sistem
- Accesarea paginii de creare a licitației

Postcondiții: Afișarea în cadrul sistemului a licitației și a produselor conținute

Scenariu favorabil:

1. Selectarea paginii de creare a licitației
2. Introducerea datelor în câmpurile puse la dispoziție
 - 2.1. Atribuirea unui nume sugestiv pentru licitație
 - 2.2. Setarea tipului de licitație
 - 2.3. Selecția datei de start a licitației
 - 2.4. Selecția momentului în care licitația se termină
 - 2.5. Adăugarea unei descrieri sugestive

3. Trimiterea datelor prin apasarea butonului “Creează licitație”
4. Adăugarea unei noi licitații în sistem

Scenariu nefavorabil:

- 2.n. Introducerea unor date nevalide sau incomplete
 - 2.n.1. Nu sunt completate toate câmpurile
 1. Sistemul afișează un mesaj corespunzător
 - 2.n.2. Data de terminare a licitației este calendaristic înaintea datei de start
 1. Sistemul afișează un mesaj corespunzător
 - 2.n.3. Perioada de start sau terminare a licitației sunt calendaristic înaintea datei curente
 1. Sistemul afișează un mesaj corespunzător
- 3.n. Erori neprevăzute în trimiterea datelor
 1. Sistemul informează utilizatorul că operația a eșuat

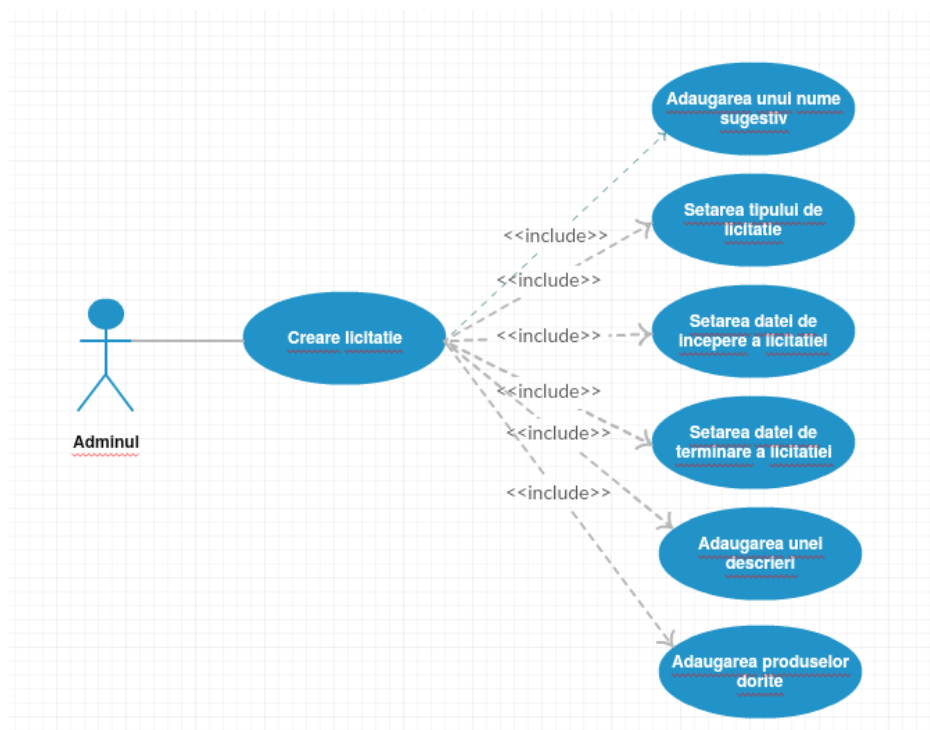


Figura 4.6. Scenariu caz de utilizare pentru crearea unei licitații

Caz de utilizare 3: Gestionarea aplicației (utilizatori, licitații, produse)

Actor: Adminul

Scop: Managementul entităților, efectuarea operațiilor de creare, actualizare și ștergere, precum și setarea unor date în cadrul aplicației

Precondiții:

- Logarea adminului în sistem
- Accesarea paginii de management

Postcondiții: Actualizarea datelor manipulate în cadrul sistemului

Scenariu favorabil:

1. Selectarea paginii de management

2. Selecția paginii corespundente (utilizator, licitație, produs)
 - 2.1. Utilizator
 - 2.2. Licitație
 - 2.3. Produs
3. Selectarea operației dorite
 - 3.1. Adăugare
 - 3.2. Modificare
 - 3.3. Ștergere
4. Introducerea datelor în câmpurile puse la dispoziție
5. Trimiterea datelor
6. Efectuarea modificărilor în sistem

Scenariu nefavorabil:

- 4.n. Introducerea unor date nevalide sau incomplete
 - 4.n.1. Nu sunt completate toate câmpurile (ramura 3.1)
 1. Sistemul afișează un mesaj corespunzător
 - 4.n.2. Datele introduse se regăsesc în sistem (ramura 3.2)
 1. Sistemul afișează un mesaj corespunzător
 - 4.n.3. Entitatea nu se regăsește în sistem (ramura 3.3)
 1. Sistemul afișează un mesaj corespunzător
- 5.n. Erori neprevăzute în trimiterea datelor
 1. Sistemul informează adminul că operația a eșuat

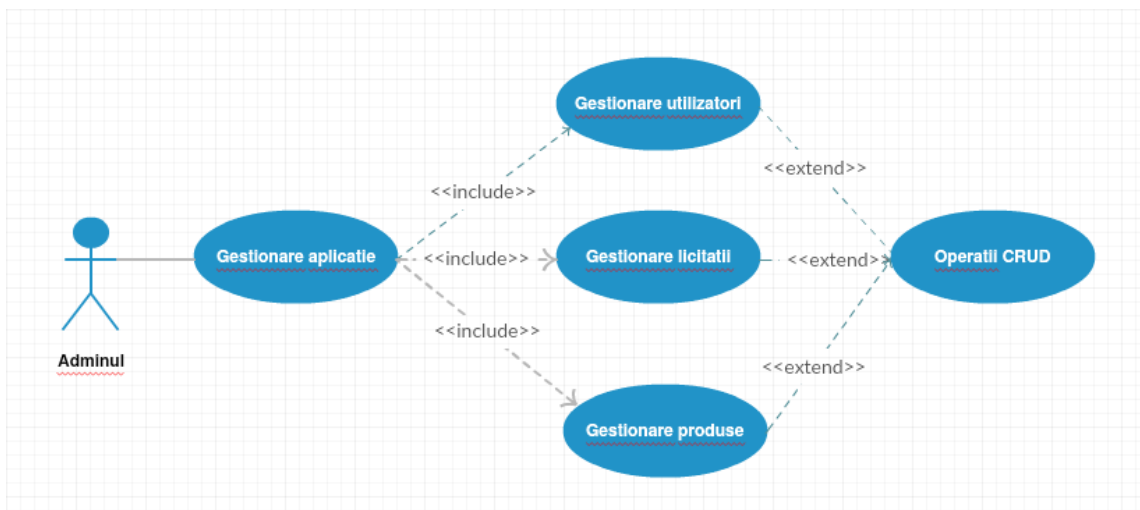


Figura 4.7. Scenariu caz de utilizare pentru gestionarea aplicației

4.2. Fundamentare teoretică

Fundamentarea teoretică preupune descrierea tehnologiilor utilizate în cadrul realizării **casei de licitații ArtBidding**, o comparație între alternativele existente pe piață, precum și o motivare a folosirii acestora.

Dezvoltarea aplicației s-a bazat pe folosirea tehnologiilor actuale regăsite pe piață, care dispun de suport din partea comunității pentru noi dezvoltări și implementări.

Alegerea acestor tehnologii s-a bazat pe dorința de a aplica și folosi standardele actuale ale pieței, ușurința integrării acestora în cadrul platformei, precum și multitudinea de facilități pe care le pun la dispoziție.

4.2.1. Hibernate Framework

Precum este specificat și pe site-ul oficial [19], Framework-ul ORM (Obeject Relational Mapping) Hibernate permite programatorilor să construiască aplicații într-un mod mai simplu, în cadrul cărora datele rezultate în urma proceselor sunt salvate. Hibernate se axează pe persistența datelor și cum sunt ele gestionate în cadrul bazei de date.

Pe lângă API-ul propriu de care dispune, Hibernate reprezintă și o implementare a specificațiilor Java Persistence API, astfel putând fi folosit în orice mediu care suportă JPA, incluzând aplicații diferite Java. Hibernate permite programatorului să dezvolte clase persistente, bazate pe principiile și paradigmele OOP: polimorfism, moștenire, compoziție și colecțiile Java. Hibernate nu necesită o interfață sau o clasă de bază pentru persistența claselor și permite oricarei clase sau structuri să fie persistente.

Arhitectura Hibernate, precum se specifică și în sursa [17], este împărțită în layere pentru a izola . Hibernate se bazează pe baza de date și configurările create pentru a oferi servicii de persistență în cadrul aplicației.

În continuare este prezentată o detaliere a Arhitecturii Hibernate cu principalele clase importante:

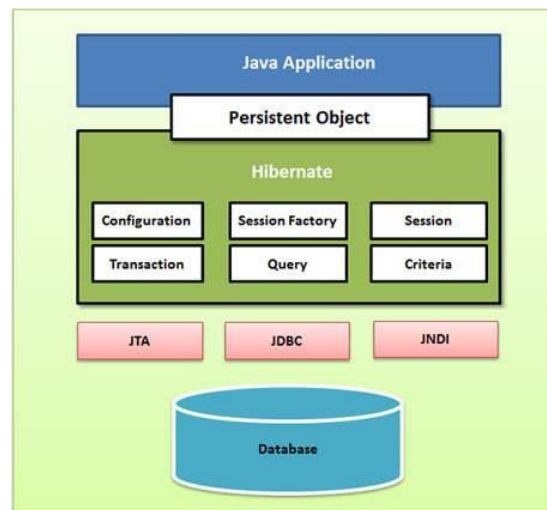


Figura 4.8 Arhitectura Hibernate (preluata din sursa [20])

Hibernate folosește o varietate de API-uri Java, precum JDBC, Java Transaction API (JTA) și Java Naming and Directory Interface (JNDI). JDBC oferă un nivel de abstractizare rudimentar al funcționalităților comune ale bazei de date relaționale.

O alternativă “lightweight” a Hibernate-ului o reprezintă OrmLite. În sursa [18] se precizează că OrmLite oferă o modalitate mai simplă pentru persistența obiectelor Java în

baza de date SQL, evitând complexitatea pachetelor standard ORM existente. OrmLite este ușor de folosit și oferă caracteristici precum setarea claselor prin adăugarea adnotărilor, abstractizarea claselor DAO (Database Access Object), suport pentru MySQL, Postgres, Microsoft SQL Server, suport pentru tranzacții, suport pentru configurarea tabelor și câmpurilor și generarea automată de cod SQL pentru crearea și ștergerea tabelor în baza de date.

Am ales să folosesc Hibernate ORM pentru persistența datelor deoarece reprezintă un framework complex, un standard pe piață cu o multitudine de servicii la dispoziție și suport, care ofera performanță ridicată, scalabilitate, stabilitate, extensibilitate, este ușor de configurat și gestionat în cadrul aplicației.

4.2.2. Spring MVC

Așa cum se precizează și în sursa [22], Spring MVC (model-view-controller) framework este construit în jurul unui DispatcherServlet, care trimite cereri claselor pentru a gestiona, configura mapările și a oferi suport.

Modulul spring web include caracteristici unice precum:

- **Separare clara a rolurilor**
Fiecare rol – controller, validator, obiect, DispatcherServlet, mapare, poate fi îndeplinit de către un obiect specializat.
- **Configurare**
Spring MVC oferă un mecanism puternic și complex de configurare atât a framework-ului, precum și a claselor din cadrul aplicației. Configurarea include referințe contextuale, de la controllere web la obiecte business și validatori.
- **Reutilizarea codului**
Această caracteristică presupune eliminarea codului duplicat și folosirea obiectelor business existente, în defavoarea dublicării acestora pentru a extinde clase de bază particulare.
- **Adaptabilitate și flexibilitate**
Aceste două caracteristici presupun definirea în cadrul unui controller a unei semnături dorite, folosind pentru parametrii adnotări, caracteristici scenariului dorit (@RequestParam, @PathVariable).
- **Validări și asocieri personalizabile**
Erorile din cadrul aplicației sunt gestionate, prin mecanisme speciale care detectează erori de validare, a tipurilor de legături existente și a conversiilor din cadrul aplicației.

DispatcherServlet reprezintă un Servlet (moștenit de la clasa de baza HttpServlet) și este declarat în cadrul aplicației în fișierul web.xml. Programatorul trebuie să mapeze cererile dorite către DispatcherServlet pentru a le gestiona, prin folosirea mapării URL.

Următoarea figură descrie fluxul procesării cererilor gestionate de Spring MVC.

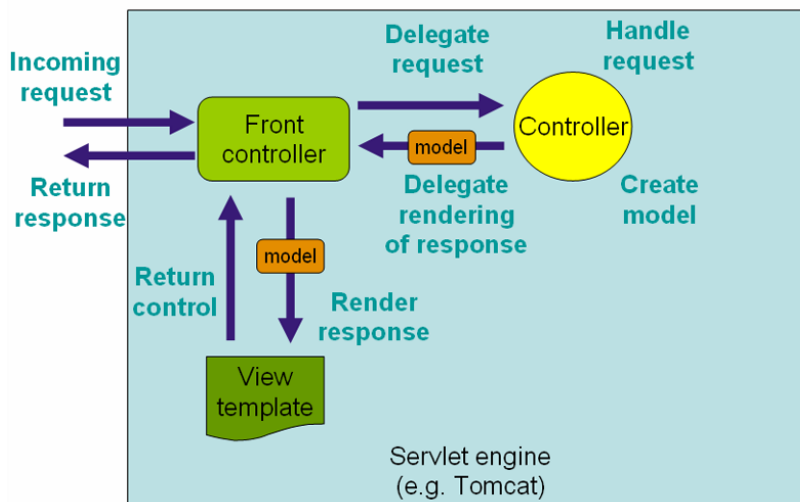


Figura 4.9. Fluxul procesării cererilor MVC (preluata din sursa [22])

O alternativă existentă pe piață care poate înlocui Spring MVC este Struts2, care reprezintă un framework Java bazat pe cereri care permite crearea de aplicații web. Struts2 urmează structura de bază a pattern-ului MVC, iar un avantaj care îl are față de Spring MVC este acela că are propriul limbaj de vizualizare, astfel putând construi aplicația integral. O altă facilitare de care dispune Struts2 este aceea că este ușor de configurat și rulat.

Am ales folosirea Spring MVC deoarece este un framework care oferă utilizatorului facilități unice, este ușor de învățat, stabil și robust care poate fi integrat ușor în cadrul aplicației dorite.

4.2.3. Serverele Tomcat și Glassfish

Apache Tomcat este un web server open source și container servlet dezvoltat de Apache Software Foundation (ASF). Glassfish este de asemenea un web server open source, proiect început de Sun Microsystems pentru platforma Java Enterprise și în moment este sponsorizat de Oracle.

Aceste servere sunt destinate pentru a deployment-ul aplicațiilor dezvoltate, parte de back-end fiind gestionată de către serverul Tomcat, iar front-end-ul aplicației de către Glassfish. Am ales separarea aplicației două părți pentru a abstractiza cele două concepte, pentru a oferi posibilitatea dezvoltării independente a celor două părți, precum și pentru o testare și analiză mai ușoară, atât a funcționalităților celor două părți – back-end și front-end, precum și comporatarea celor două servere.

Ambele părți ale aplicației puteau fi plasate în cadrul aceluiași server, dar pentru o înțelegere mai amanunțită a conceptelor și a modului de funcționare, am decis separarea

acesteia. Totodată, o eventuală eroare în cadrul aplicației este mai ușor de depistat și putem gestiona eroarea mai eficient și rapid.

Am ales folosirea acestor 2 servere pentru a studia comportamentul acestora, fiind 2 servere care sunt frecvent utilizate în momentul actual pe piață, dispun de facilități care ajută programatorul să depisteze erori și să le remedieze, precum și faptul că sunt open source și sunt îmbunătățite și întreținute de comunitate.

4.2.4. AngularJs

AngularJs este un framework foarte puternic de Javascript. Așa cum se precizează și în sursa [23], acesta este folosit în cadrul proiectelor Single Page (SPA). Acesta extinde DOM-ul (Document Object Model) HTML, oferind attribute adiționale care îl face mai sensibil la acțiunile utilizatorului. AngularJs este open source, gratuit și este folosit de către mii de programatori din întreaga lume.

Caracteristicile principale ale framework-ului AngularJs sunt următoarele:

- **Data binding**
Această caracteristică reprezintă sincronizarea automată a datelor dintre model și view
- **Scope**
Obiectele care au ca atribut scope se referă la datele din model. Acestea actionează ca o legătură între controller și view.
- **Controller**
Controller-ul reprezintă o funcție Javascript care este asociată unui scope particular.
- **Servicii**
Serviciile reprezintă obiecte din cadrul aplicației care sunt instantiate o singura data în aplicație și oferă anumite servicii. AngularJs oferă servicii predefinite (de exemplu: \$http – care gestionează cereri HTTP), dar îi oferă posibilitatea utilizatorului de a-și crea alte servicii personalizate.
- **Filtre**
Filtrele reprezintă niște caracteristici care sunt aplicate asupra unei liste, având ca rezultat o nouă listă cu obiectele care îndeplinesc aceste cerințe.
- **Directive**
Directivele sunt markere pentru elementele din DOM. Acestea sunt folosite pentru a crea tag-uri HTML personalizate. AngularJs dispune de directive predefinite precum ngModel, ngView, ngShow, ...
- **Rutare**
Rutarea reprezintă conceptul de tranziție între view-uri.

AngularJS este un framework bazat pe pattern-ul MVC. MVC este popular deoarece izolează logica aplicației de layerul asociat interfeței utilizatorului pentru o separare clară a conceptelor. Model-ul reprezintă nivelul responsabil pentru menținerea datelor, view-ul este responsabil pentru afișarea tuturor datelor sau a unei părți din acestea utilizatorului, iar controller-ul este o porțiune de cod care controlează interacțiunea dintre model și view.

Pe piața actuală există o diversitate de alternative ale framework-ului AngularJs. Două importante alternative sunt reprezentate de Vue.js și React. Vue reprezintă un framework lightweight care a fost menit să organizeze și simplifice dezvoltarea aplicațiilor web. Acesta poate fi integrat ușor în proiect, fiind un framework adaptabil, capabil să realizeze aplicații single-page performante, care este versatil, simplu, cu funcționalități minimale dar puternice și care oferă o performanță ridicată.

A doua alternativă pentru AngularJS o reprezintă React, o bibliotecă javascript care este folosită pentru a crea interfețe grafice. React ușurează realizarea interfețelor grafice interactive, proiectând view-uri simple pentru fiecare stare a aplicației. Acesta va modifica și randa eficient doar componentele dorite când datele se modifică. View-urile declarative ajută la realizarea unui cod predictibil și mai ușor de gestionat.

Am ales folosirea Framework-ului Angular Js deoarece este un framework complex, care implementează modelul MVC, este intuitiv de folosit, se integrează ușor, se mapează pe sistemul dorit, este flexibil, oferă o multitudine de facilități predefinite precum directive, servicii, este foarte popular și recomandat de către o mulțime de programatori cu experiență.

4.2.5. HTML, CSS și Javascript

HTML este un limbaj de marcare standard pentru crearea paginilor web. HTML (Hyper Text Markup Language) descrie structura paginilor web folosindu-se de marcare, iar elementele HTML sunt blocurile constitutive ale paginilor HTML. Elementele HTML sunt reprezentate prin tag-uri, fiind structurate în entități precum „heading”, „paragraph”, „table”, etc. Browser-ul nu afișează tag-urile HTML, ci le folosește pentru a randa conținutul paginilor, pe care le vizualizează utilizatorul.

CSS este un limbaj care descrie stilul unui document HTML. CSS (Cascading Style Sheet) descrie cum trebuie să fie afișate elementele HTML. CSS salvează o mare parte din munca programatorului, care poate controla layout-ul multiplelor pagini web simultan, prin folosirea anumitor stiluri care să fie aplicate asupra paginilor dorite.

Javascript este un limbaj de programare care este folosit pentru a face paginile web interactive. Acesta se utilizează pentru a specifica comportamentul unor pagini web sau a elementelor acestora.

În cadrul implementării **casei de licitații ArtBidding**, am folosit aceste 3 limbaje pentru a crea pagini web interactive, care au un design deosebit și au funcționalități proprii, pentru a asigura o experiență cât mai plăcută pentru utilizator.

4.2.6. Bootstrap

Bootstrap reprezintă un front-end web framework pentru design-ul site-urilor web și al aplicațiilor web. Acesta conține design-uri de template-uri bazate pe HTML, CSS care constau în pagini, butoane, navigare sau alte componente similare, precum și extensii opționale de Javascript. Bootstrap este cel mai popular framework de HTML, CSS și Javascript pentru dezvoltarea aplicațiilor web responsive, care sunt ușor și eficiente

scalate folosind un singur cod de bază, care se aplică de la telefon până la PC, folosind media queries.

Așa cum se poate observa și pe site-ul principal [24], bootstrap oferă o multitudine de facilități pentru a ajuta la construirea unui site modern, cu o interfață grafică deosebită, într-un timp cât mai scurt. Acesta dispune de teme prestabilite pe care le putem integra în aplicația proprie și adapta după bunul plac. Fiecare temă este însoțită de o documentație în care sunt descrise componentele, precum și o mulțime de caracteristici specifice Bootstrap, plugin-uri, build tools, etc.

Bootstrap reprezintă o platformă care a început să fie folosită în majoritatea aplicațiilor online datorită multitudinii de teme web, aplicații și plugin-uri pe care le pune la dispoziția utilizatorului.

Un concurent de seala al Bootstrap-ului îl reprezintă Foundation. Acesta este considerat ca fiind cel mai avansat framework responsive pentru front-end din lume. Aceste două framework-uri complexe sunt la ora actuală cele mai populare și folosite, ambele oferind diverse facilități și suport, astfel încât este destul de dificil de spus care dintre ele este mai performant. Ambele framework-uri au potențial enorm, iar folosirea acestora depinde de preferințele personale. Foundation are un sistem grid mai robust decât cel al Bootstrap-ului, iar Bootstrap este mai ușor de customizat.

Personal am optat pentru Framework-ul Bootstrap deoarece este ușor de folosit și integrat, este open source, conține elemente responsive, iar prin folosirea elementelor de design predefinite am salvat foarte mult timp necesar customizării paginilor web.

4.2.7. Paypal API

Pentru a facilita câștigătorilor posibilitatea de a efectua plata online, am ales folosirea unui serviciu de plată extern pe care l-am integrat în sistemul propriu, și anume Paypal. Folosirea reprezintă o modalitate ușoară și sigură de plată online, care îi oferă utilizatorului siguranță și încredere că serviciile dorite se vor executa cu succes.

În cadrul sursei [25] sunt prezentate toate caracteristicile acestui API. Programatorul poate integra aceste funcționalități în cadrul site-ului propriu pentru a le permite clienților plata online. Aceștia își pot crea un cont nou PayPal dacă dispun de unul existent, sau se pot loga introducând datele proprii și efectua plata rapid, la câteva click-uri distanță. Utilizatorului îi este pus la dispoziție serviciul imediat după câștigarea unei licitații, sau poate efectua plata ulterior, utilizând altă metodă de plată dorită.

Alternative de servicii online care oferă plata în mod sigur și rapid a clienților sunt nenumărate, una dintre ele fiind Stripe. Acesta reprezintă un alt standard în cadrul plăților online și reprezintă o platformă software care pune la dispoziție o multitudine de caracteristici precum condiții de testare înainte de a plasa produsul pe piață pentru a vedea comportamentul în diferite scenarii, suport pentru aplicațiile mobile, gestionarea datelor business, instrumente care previn fraudă, diagrame, grafice și multe altele.

Deși pune la dispoziție o multitudine de servicii, Stripe nu este open source și este necesară cumpărarea acestor facilități. Din acest motiv am decis alegerea Paypal API, care este ușor de integrat și nu necesită achitarea unei sume pentru folosirea acestuia.

4.2.8. Google Maps API

Acest serviciu prezintă o soluție de expunere a localizării globale a **casei de licitații ArtBidding**, precum și o modalitate de ghidare pentru clienții care doresc să vizualizeze zona în care se regăsește sediul central sau doresc să se deplaseze la acesta. Integrarea acestui API extern a fost ușoară, codul fiind pus la dispoziție de către Google, iar pentru accesul la resursele interne, a fost necesară o cerere pentru a fi pusă la dispoziție o cheie unică.

Google Maps API este gratuit pentru majoritatea cazurilor de utilizare, având niște caracteristici standard. Elementul central îl reprezintă realizarea hărții: **map=new google.maps.Map(document.getElementById("googleMap"), mapProp)**, această comandă fiind plasată în cadrul unui document HTML. MapProp reprezintă caracteristicile hărții create, având diferite tipuri de afișare și coordonate.

Am ales folosirea Google Maps API deoarece este ușor de integrat, gratuit, cel mai popular la momentul actual și reprezintă o soluție la îndemână pentru a expune utilizatorilor locația **casei de licitații**, precum și a zonei adiacente.

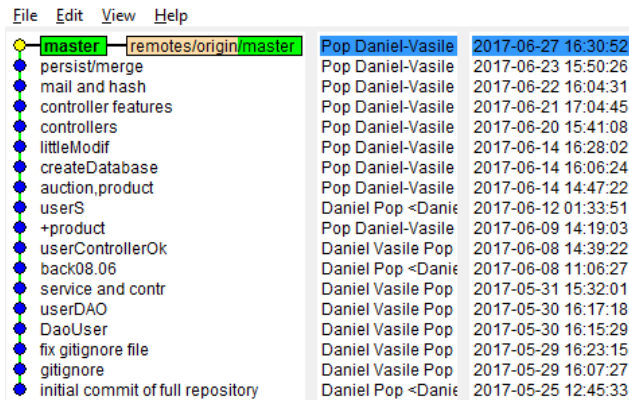


Figura 4.10. Google Maps API

4.2.9. Git

Git reprezintă un sistem de control al versiunilor pentru urmărirea modificărilor și coordonarea activității fișierelor în cadrul unui proiect mai complex. Sistemul de control al versiunilor permite să avem "versiuni" ale unui proiect, care arată schimbările care au fost făcute din timp în timp și permite revenirea la o versiune anterioară dacă este necesar. Git permite structurarea mai eficientă a codului și o gestiune mai ușoară a acestuia, iar în proiecte mai ample ușurează și permite munca în echipă.

În contextul Git, un commit reprezintă o înregistrare a modificărilor din repository (proiect). Alăturat este prezentat istoricul commit-urilor individuale, pentru a trasa principalele funcționalități ale sistemului:



Commit Message	Author	Date
initial commit of full repository	Daniel Pop <Daniel@pop.ro>	2017-05-25 12:45:33
gitignore	Daniel Vasile Pop	2017-05-29 16:07:27
fix gitignore file	Daniel Vasile Pop	2017-05-29 16:23:15
DaoUser	Daniel Vasile Pop	2017-05-30 16:15:29
userDAO	Daniel Vasile Pop	2017-05-30 16:17:18
service and contr	Daniel Vasile Pop	2017-05-31 15:32:01
back08.06	Daniel Pop <Daniel@pop.ro>	2017-06-08 11:06:27
userControllerOk	Daniel Vasile Pop	2017-06-08 14:39:22
+product	Pop Daniel-Vasile	2017-06-09 14:19:03
userS	Daniel Pop <Daniel@pop.ro>	2017-06-12 01:33:51
auction,product	Pop Daniel-Vasile	2017-06-14 14:47:22
createDatabase	Pop Daniel-Vasile	2017-06-14 16:06:24
littleModif	Pop Daniel-Vasile	2017-06-14 16:28:02
controllers	Pop Daniel-Vasile	2017-06-20 15:41:08
controller features	Pop Daniel-Vasile	2017-06-21 17:04:45
mail and hash	Pop Daniel-Vasile	2017-06-22 16:04:31
persist/merge	Pop Daniel-Vasile	2017-06-23 15:50:26
master	Pop Daniel-Vasile	2017-06-27 16:30:52

Figura 4.11. Istoricul commit-urilor in Git

Una din alternativele Git este reprezentată de Mercurial SCM. Mercurial este un instrument gratuit de gestionare a sursei de control al surselor din cadrul unui proiect. Acesta efectuează eficient proiectele de orice dimensiune și oferă o interfață ușor de folosit și intuitivă. Mercurial gestionează eficient proiectele de orice mărime, iar diferențele între acesta și Git nu sunt foarte mari, folosirea unuia depinzând de contextul proiectului și preferințele personale.

Personal, am ales folosirea Git deoarece este cel mai popular sistem de control al versiunilor la ora actuală, este gratuit și open source și gestionează totul de la proiecte mici la proiecte ample cu rapiditate și eficiență.

Capitolul 5. Proiectare de detaliu și implementare

Principalul obiectiv al acestei aplicații este transpunerea licitațiilor tradiționale în mediul online sub forma unui site de licitații. Design-ul acestei platforme constă într-o arhitectură modernă, client-server, grupată pe 3 nivele, la baza acestora aflându-se pattern-ul MVC (Model-View-Controller). Acest pattern este responsabil pentru separarea conceptelor din cadrul aplicației.

Acest capitol prezintă nivelele din cadrul aplicației, modul de implementare și organizare, precum și interacțiunea dintre componentele sistemului.

5.1. Arhitectura conceptuală

Arhitectura conceptuală se referă la modul de organizare al componentelor în cadrul aplicației și modul acestora de funcționare. Aplicația proprie este construită sub forma unui sistem distribuit de tip client-server, structurat pe 3 nivele – model, view, controller. În cadrul modelului se gestionează obiecte Java care conțin date pe care le modelăm ulterior în aplicație, view-ul constituie vizualizarea acestor date conținute în model, iar controller-ul gestionează datele din flow-ul aplicației și actualizează view-ul atunci când datele din model sunt modificate. Fiecare nivel folosește doar funcționalitățile de pe nivelul anterior, acestea propagându-se de la nivel la nivel pentru a parcurge întregul flow al aplicației.

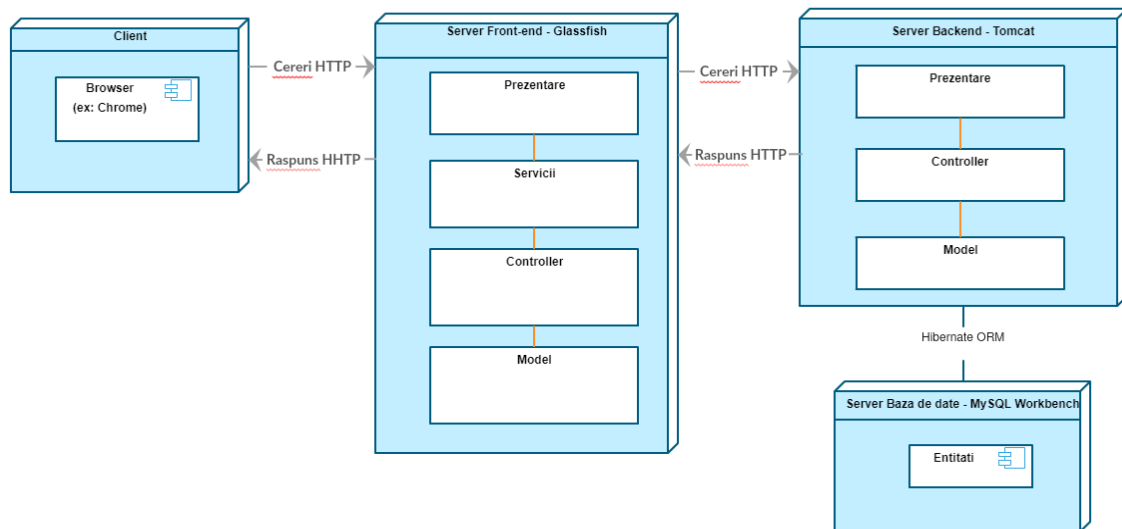


Figura 5.1. Diagramă conceptuală

Tabela user stochează datele despre utilizatorii sistemului și conține următoarele câmpuri:

- **userId** – reprezintă cheia primară a tabelului, prin care se identifică un utilizator în mod unic; tipul câmpului: int
- **address** – stochează adresa utilizatorului; tip: Varchar
- **email** – stochează email-ul utilizatorului; tip: Varchar
- **name** – reține numele utilizatorului; tip: Varchar
- **username** – gestionează username-ul utilizatorului; tip: Varchar
- **password** – stochează parola utilizatorului; tip: Varchar
- **phone** – reține numărul de telefon; tip: Varchar
- **role** – specifică rolul utilizatorului; tip: Varchar

Tabela usercomments reține comment-urile utilizatorilor, precum și data când au fost plasate. Aceasta conține următoarele câmpuri:

- **userCommId** – reprezintă cheia primară a tabelului, prin care se identifică un comment al unui utilizator în mod unic; tipul câmpului: Int
- **commDate** – stochează adresa utilizatorului; tip: DateTime
- **comment** – reține conținutul comment-ului utilizatorului; tip: Varchar
- **userId** – reține id-ul utilizatorului care a plasat comment-ul; acesta reprezintă cheia străină către tabelul user; tip: Int

Între tabela usercomment și tabela user se află relația many-to-one, aceasta subliniind faptul că un utilizator poate avea mai multe comment-uri.

Tabela userrole specifică rolurile utilizatorului (admin sau user) și este în relație one-to-one cu tabela user.

Tabela auction reține date despre licitațiile desfășurate în cadrul aplicației.. Aceasta conține următoarele câmpuri:

- **auctionId** – reprezintă cheia primară a tabelului, prin care se identifică o licitație în mod unic; tipul câmpului: Int
- **auctionName** – înregistrează numele licitației; tip: Varchar
- **auctionState** – reține starea curentă a licitației; tip: Varchar
- **description** – conține o descriere a licitației; tip: Varchar
- **startDate** – reprezintă data de start a licitației; tip: DateTime
- **endDate** – reprezintă data de terminare a licitației; tip: DateTime
- **auctionType** – reprezintă tipul licitației; tip: Varchar
- **userId** - reține id-ul utilizatorului care a participat la licitație; acesta reprezintă cheia străină către tabela user; tip: Int

Între tabela auction și tabela user se află relația many-to-one, aceasta subliniind faptul că un utilizator poate participa la mai multe licitații.

Tabela auctionType specifică tipul de licitații disponibile (implementare proprie sau un tip de licitație tradițională) și este în relație one-to-one cu tabela auction.

Tabela auctionState reprezintă starea licitației curente (începută, în desfășurare, terminată) și este în relație one-to-one cu tabela auction.

Tabela product conține date despre produsele din cadrul licitației. Aceasta conține următoarele câmpuri:

- productId – reprezintă cheia primară a tabelului, prin care se identifică un produs în mod unic; tipul câmpului: Int
- creatorName – precizează artistul care a realizat produsul; tip: Varchar
- description – conține o descriere a produsului; tip: Varchar
- name – reprezintă numele licitației; tip: Varchar
- photo – conține o fotografie a produsului; tip: Blob
- price – reprezintă prețul produsului; tip: Int
- productType – reprezintă tipul produsului; tip: Varchar
- auctionId - reține id-ul licitației în cadrul căreia produsul este plasat; acesta reprezintă cheia străină către tabela auction; tip: Int

Între tabela product și tabela auction se află relația many-to-one, aceasta reprezentând faptul o licitație poate conține mai multe produse.

Tabela productType specifică tipul de produse disponibile în cadrul licitației (antichitate, pictură, sculptură) și este în relație one-to-one cu tabela product.

Tabela creatorName specifică proprietarului unui obiect de artă și conține detalii despre acesta și este în relație one-to-one cu tabela product.

Tabela bid conține date despre bid-urile plasate în cadrul sistemului de către utilizatori. Aceasta conține următoarele câmpuri:

- bidId – reprezintă cheia primară a tabelului; tipul câmpului: Int
- biddingDate – precizează data la care a fost plasat un bid; tip: DateTime
- value – constituie valoarea unui bid; tip: int
- auctionId - reține id-ul licitației în cadrul căreia s-a plasat miza și produsul dorit; acesta reprezintă cheia străină către tabela auction; tip: Int
- userId - reține id-ul utilizatorului care a plasat bid-ul în cadrul licitației; acesta reprezintă cheia străină către tabela user; tip: Int

Între tabela bid și tabela auction se află relația many-to-one, aceasta reprezentând faptul în cadrul unei licitații se pot plasa mai multe bid-uri.

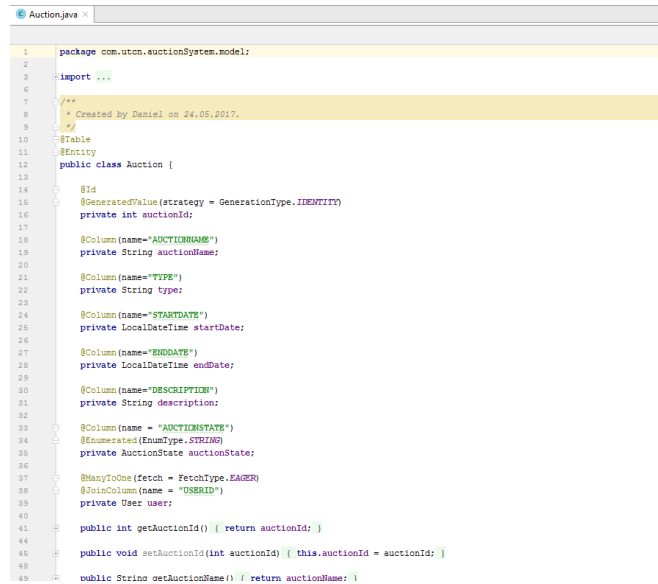
Între tabela bid și tabela user se află relația many-to-one, aceasta reprezentând faptul un user poate plasa mai multe bid-uri.

5.3. Nivelul de model

În cadrul acestui nivel, am folosit Framework-ul Hibernate pentru definirea domeniului problemei și asocierea obiectelor cu modelul din baza de date. Pentru fiecare obiect în parte am creat corespondentul, care este și POJO (plain old Java object), și DTO (Data Transfer Object), precum și Java bean. Un POJO reprezintă un simplu obiect Java, care nu prezintă caracteristici speciale sau alte restricții. Un DTO reprezintă un obiect

care este destinat transferului datelor între procesele din cadrul aplicației. În cadrul aplicației proprii, datele sunt transferate între layer-urile adiacente. Un Bean reprezintă un obiect care este instanțiat și asamblat de către Spring.

Următoarea figură prezintă un exemplu de model din cadrul aplicației.



```
1 package com.utcn.auctionSystem.model;
2
3 import ...
4
5
6
7 /**
8  * Created by Daniel on 24.05.2017.
9  */
10 @Table
11 @Entity
12 public class Auction {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int auctionId;
17
18     @Column(name="AUCTIONNAME")
19     private String auctionName;
20
21     @Column(name="TYPE")
22     private String type;
23
24     @Column(name="STARTDATE")
25     private LocalDateTime startDate;
26
27     @Column(name="ENDDATE")
28     private LocalDateTime endDate;
29
30     @Column(name="DESCRIPTION")
31     private String description;
32
33     @Column(name="AUCTIONSTATE")
34     @Enumerated(EnumType.STRING)
35     private AuctionState auctionState;
36
37     @ManyToOne(fetch = FetchType.LAZY)
38     @JoinColumn(name = "USERID")
39     private User user;
40
41     public int getAuctionId() { return auctionId; }
42
43     public void setAuctionId(int auctionId) { this.auctionId = auctionId; }
44
45     public String getAuctionName() { return auctionName; }
```

Figura 5.3. Clasa model (Auction)

Spring oferă un model de programare și configurare pentru aplicațiile moderne. Un element cheie al framework-ului Spring este suportul infrastructural la nivelul aplicației - se axează pe asamblarea componentelor, astfel încât programatorul să se poată concentra asupra logicii aplicației.

Caracteristicile importante ale Spring-ului sunt dependency injection, care este o tehnică în care un obiect gestionează dependențele altui obiect și oferirea unui cadru de dezvoltare a aplicațiilor web MVC și a serviciilor pe REST (stil arhitectural).

Spring oferă o serie de adnotări care se aplică clasei sau a obiectelor din cadrul aplicației. În continuare sunt prezentate adnotările folosite în nivelul de model al aplicației:

```
@Component
public class AuctionDao extends BaseDao
```

```
@Autowired
private AuctionDao auctionDao;
```

@Component indică faptul că o clasă adnotată este o "componentă". Astfel de clase sunt considerate ca fiind candidați pentru detectarea automată atunci când se utilizează configurația bazată pe adnotări și scanarea de clasă.

Adnotarea @Autowired oferă un control mai precis asupra locului și modului în care ar trebui să se realizeze management-ul bean-urilor. Autowiring-ul se realizează prin plasarea unei instanțe a unei bean în câmpul dorit într-o instanță a unui alt bean. Ambele clase ar trebui să fie bean-uri, adică ar trebui să fie definite pentru a trăi în contextul aplicației.

Clasele de model prezente în aplicația dezvoltată sunt:

- User – gestionează fluxul informațiilor referitoare la utilizatorii aplicației
- Product – constituie modelul care modelează produsele din cadrul licitației
- Auction – prezintă modelul unei licitații
- Bid – gestionează informațiile despre mizele plasate

Aceste modele sunt gestionate și corelate cu entitățile regăsite în baza de date. Astfel aplicația permite persistența datelor și reținerea acestora pentru, precum și un management mai eficient al fluxului de informații din sistem.

Următoarea diagramă prezintă clasele de model prezente în sistem, precum și o expunere a câmpurilor și metodelor constituente.

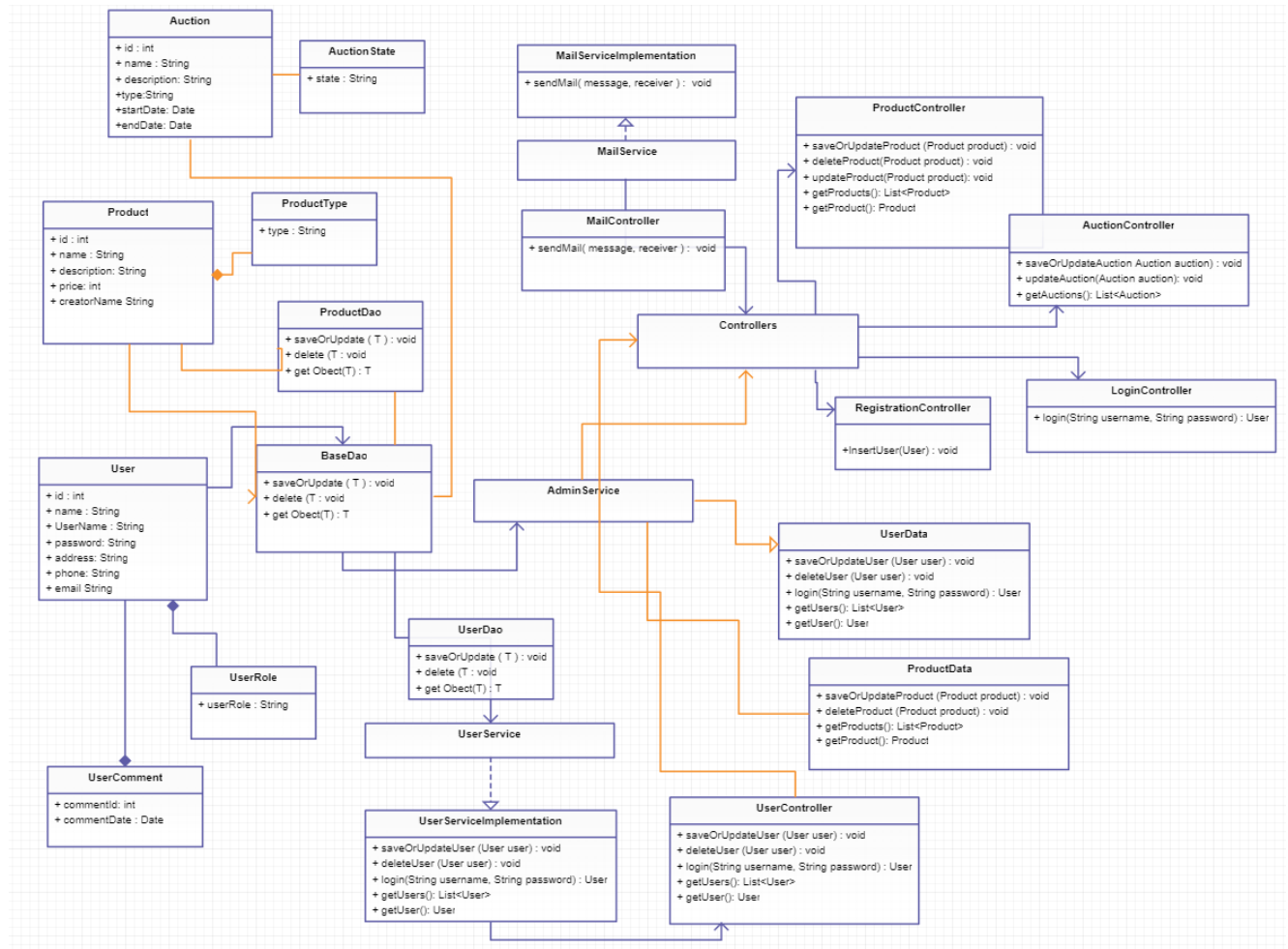


Figura 5.4. Clasele de model și interacțiunea în cadrul sistemului

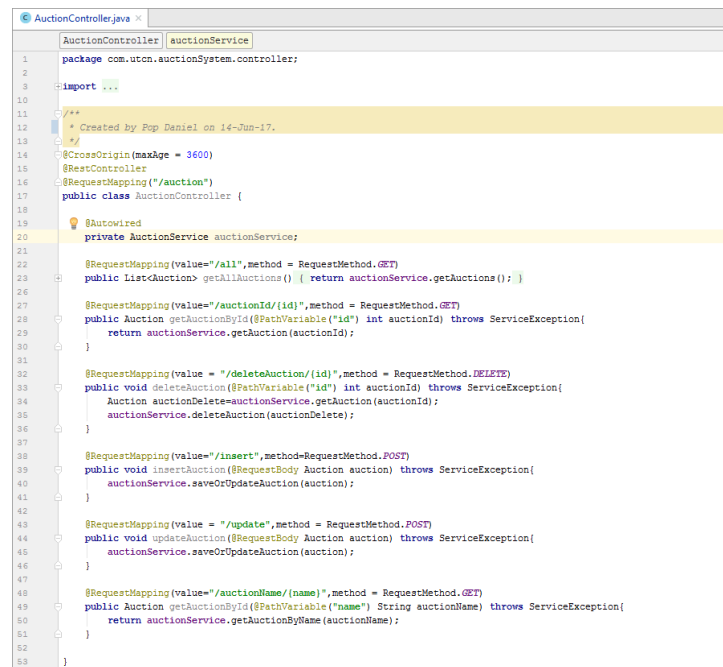
5.4. Nivelul de controller

Acest nivel constituie o punte de legătură între nivelul de model al aplicației și cel de prezentare. Controller-ul pune la dispoziție serviciile și metodele aplicate asupra datelor regăsite în modelul aplicației pentru a putea fi gestionate în nivelul de prezentare.

În cadrul aplicației dezvoltate, controllere-le sunt prezente atât în partea de backend, cât și în cea de front-end. În fiecare din părțile constituente controllerele își au rolul bine stabilit, de a controla flow-ul aplicației și a face legătura dintre model și view.

În partea de back-end a aplicației, gruparea controllere-lor se bazează pe funcționalitățile pe care le pun la dispoziție, iar în partea de front-end se păstrează aceeași structură, datele preluându-se din partea de back-end prin realizarea de cereri http.

Un exemplu de realizare al unui controller este exemplificat în figura următoare.



```
1 package com.utcn.auctionSystem.controller;
2
3 import ...
4
5 /**
6  * Created by Pop Daniel on 14-Jun-17.
7  */
8
9 @CrossOrigin(maxAge = 3600)
10 @RestController
11 @RequestMapping("/auction")
12 public class AuctionController {
13
14     @Autowired
15     private AuctionService auctionService;
16
17     @GetMapping(value="/all", method = RequestMethod.GET)
18     public List<Auction> getAllAuctions() { return auctionService.getAuctions(); }
19
20     @GetMapping(value="/auction/{id}", method = RequestMethod.GET)
21     public Auction getAuctionById(@PathVariable("id") int auctionId) throws ServiceException{
22         return auctionService.getAuction(auctionId);
23     }
24
25     @DeleteMapping(value = "/deleteAuction/{id}", method = RequestMethod.DELETE)
26     public void deleteAuction(@PathVariable("id") int auctionId) throws ServiceException{
27         Auction auctionDelete=auctionService.getAuction(auctionId);
28         auctionService.deleteAuction(auctionDelete);
29     }
30
31     @PostMapping(value="/insert", method=RequestMethod.POST)
32     public void insertAuction(@RequestBody Auction auction) throws ServiceException{
33         auctionService.saveOrUpdateAuction(auction);
34     }
35
36     @PostMapping(value = "/update", method = RequestMethod.POST)
37     public void updateAuction(@RequestBody Auction auction) throws ServiceException{
38         auctionService.saveOrUpdateAuction(auction);
39     }
40
41     @GetMapping(value="/auctionName/{name}", method = RequestMethod.GET)
42     public Auction getAuctionById(@PathVariable("name") String auctionName) throws ServiceException{
43         return auctionService.getAuctionByName(auctionName);
44     }
45 }
```

Figura 5.5. Nivelul de controller (Auction)

Nivelul de controller cuprinde 6 controllere, fiecare având funcționalități specifice, puse la dispoziție pentru a putea folosite și în partea de front-end a aplicației prin maparea corespunzătoare. Gruparea controllere-lor reprezintă o modalitate de bună structurare și gestionare a aplicației și întărește principiul de separare a principiilor în aplicațiile complexe.

În continuare sunt descrise controllere-le din cadrul aplicației, metodele specifice puse la dispoziție de fiecare în parte, precum și maparea corespunzătoare a acestora.

UserController reprezintă controller-ul care este responsabil cu management-ul datelor referitoare la utilizatorii aplicației. Maparea acestuia pentru a putea procesa cererile este `/user` și pune la dispoziție următoarele funcționalități:

- **getAllUsers** – această metodă returnează o listă cu toți utilizatorii aplicației. Aceasta poate fi apelată doar de către adminii aplicației, care utilizează maparea `/all`
- **getUserById** – returnează un utilizator pe baza id-ul. Maparea corespunzătoare este `/userId/{id}`, unde `{id}` reprezintă un `PathVariable`
- **insertUser** – inserează un utilizator în baza de date. Acesta se apelează folosind `/insert` și așteaptă un `RequestBody` de tip `User`
- **deleteUser** – șterge un utilizator din sistem. Se utilizează apelând `/deleteUser/{id}`, unde `{id}` reprezintă un `PathVariable`

- **updateUser** – actualizează un utilizator în sistem. Se introduc noile date ale utilizatorului, iar metoda se execută cu succes dacă datele sunt valide și nu se mai regăsesc în sistem. Mapare: **/update**
- **makeAdmin** – această metodă oferă unui utilizator obișnuit statutul de admin. Maparea specifică: **/makeAdmin/{id}**, unde id-ul reprezintă identificatorul utilizatorului asupra căruia se efectuează modificarea rolului
- **login** – metoda este folosită în cadrul autentificării utilizatorilor în sistem. Maparea specifică: **/login**

UserCommController constituie controller-ul care gestionează comment-urile lăsate de către utilizatorii sistemului. Maparea caracteristică în cadrul sistemului este **/userComm**. Acest controller pune la dispoziție următoarele metode:

- **insertUserComm** – facilitează utilizatorului posibilitatea de a adăuga un comentariu. Mapare: **/insert**
- **deleteUserComm** – permite ștergerea unui comentariu. Mapare: **/deleteUserComm/{id}**, unde id-ul este reprezentat de identificatorul utilizatorului care a plasat un comment
- **updateUserComm** – permite modificarea unui comentariu de către același utilizator. Mapare: **/update**

ProductController este controller-ul care gestionează produsele din cadrul sistemului. Maparea acestuia este **/product** și dispune de următoarele funcționalități:

- **getAllProducts** – returnează o listă cu toate produsele existente. Maparea specifică este: **/all**
- **insertProduct** – metoda folosită pentru a adăuga un produs în cadrul sistemului. Mapare: **/insert**
- **deleteProduct** – ștergerea unui produs care are id-ul specificat. Maparea este următoarea: **/deleteProduct/{id}**
- **getProductById** – metoda returnează produsul cu id-ul specificat și are maparea **/productId/{id}**
- **getProductByName** – metoda este similară cu cea prezentată anterior, dar întoarce produsul căutat pe baza numelui introdus. Mapare: **/productName/{name}**

AuctionController reprezintă controller-ul care gestionează licitațiile din sistem. Acesta are maparea **/auction** și prezintă următoarele metode:

- **getAllAuctions** – returnează o listă cu licitațiile existente. Maparea specifică este: **/all**
- **insertAuction** – metoda folosită pentru a adăuga o licitație în cadrul sistemului. Mapare: **/insert**
- **deleteAuction** – ștergerea unei licitații care are id-ul specificat. Maparea este următoarea: **/deleteAuction/{id}**
- **getAuctionById** – metoda returnează licitația cu id-ul specificat și are maparea **/auctionId/{id}**
- **getAuctionByName** – metoda este similară cu cea prezentată anterior, dar întoarce licitația căutată pe baza numelui introdus. Mapare: **/auctionName/{name}**

BidController constituie controller-ul care gestionează mizele oferite în cadrul licitației. Acest controller poate fi accesat doar de către adminii licitației, care pot folosi datele pentru a genera rapoarte și a-i oferi utilizatorului o experiență mai plăcută. Acesta are maparea **/bid** și prezintă următoarele metode:

- **getAllBids** – returnează o listă cu mizele plasate în cadrul tuturor licitațiilor sistemului. Maparea specifică este: **/all**
- **insertAuction** – metoda folosită pentru a adauga o miza în cadrul unei licitații, pentru un produs specificat. Mapare: **/insert**
- **getWinner** – metoda returnează câștigătorul licitației cu id-ul specificat și are maparea **/getWinner/{id}**
- **getSum** – metoda returnează suma pe care trebuie să o onoreze câștigătorul licitației. Mapare: **/getSum/{id}**, unde id-ul este reprezentat de identificatorul licitației câștigate.
- **getBidValues** – metoda returnează toate mizele plasate în cadrul unei licitații cu id-ul specificat și are maparea **/getBidValues/{id}**

MessageController este controller-ul care gestionează trimiterea mesajelor. Acesta poate fi apelat de către un admin sau poate fi integrat în sistem pentru a trimite automat mesaje atunci când se determină un câștigător al unei licitații. Acesta are maparea **/message** și pune la dispoziție următoarea metodă:

- **sendMessage** – metoda este mapată astfel: **/sendMessage** și are rolul de a trimite un mesaj utilizatorului specificat

Controller-ele pun la dispoziție metode pentru a putea fi prelucrate în nivelul de prezentare, acestea făcând legătura între model și view.

În continuare sunt prezentate controllere-le și metodele pe care le conțin, precum și actorul care are acces la aceste funcționalități.

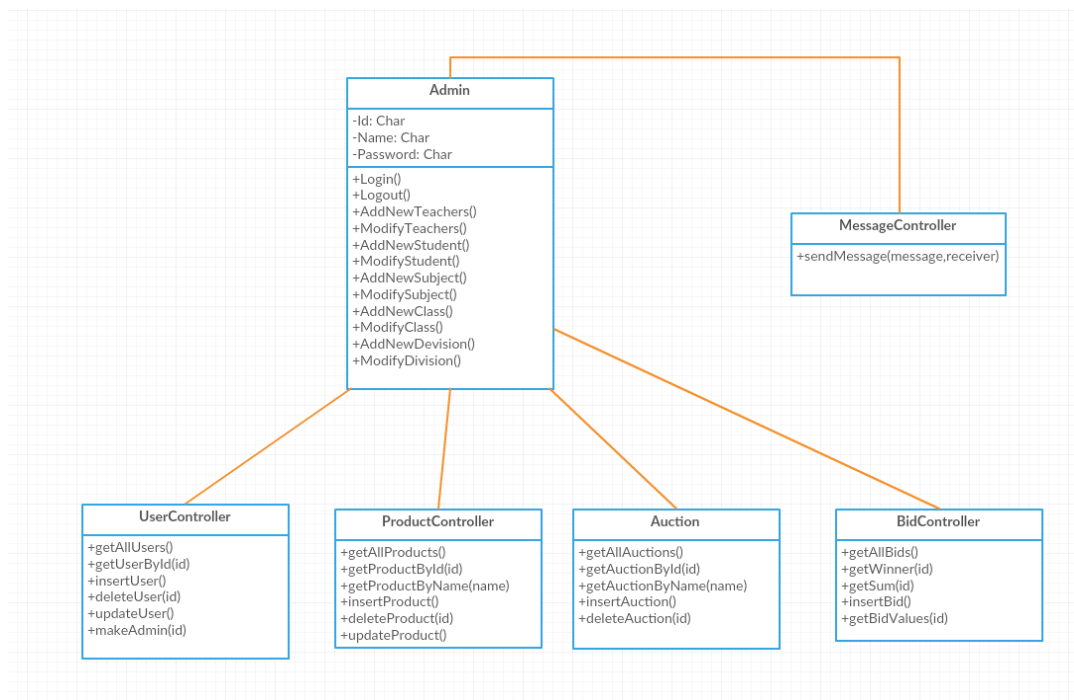


Figura 5.6. Controllerele și actorul principal

5.5. Nivelul de prezentare

Nivelul de prezentare reprezintă puntea de legătură dintre aplicația dezvoltată și utilizatorii sistemului. Nivelul de prezentare oferă interfața grafică cu utilizatorul, cuprinzând funcționalitățile sistemului descrise anterior.

Structura acestui nivel presupune modelul Single Page Application (SPA), în care scopul este acela de a-i oferi utilizatorului o experiență similară aplicațiilor desktop. În aplicația proprie, tot codul necesar de HTML, Javascript și CSS este procesat în cadrul unei singure încărcări a paginii, iar pentru a modifica pagina se modifică doar conținutul.

Acest nivel conține librării, controllere și servicii, după cum se observă și în figura următoare.

```
<!-- LIBS -->
<script src="assets/js/angular.min.js"></script>
<script src="assets/js/angular-route.min.js"></script>

<!-- APP -->
<script src="app/config/app.js"></script>
<script src="app/config/config.js"></script>
<!-- CONTROLLERS -->
<script src="app/controllers/navControllers.js"></script>
<script src="app/controllers/userControllers.js"></script>
<script src="app/controllers/registerControllers.js"></script>
<script src="app/controllers/loginControllers.js"></script>
<script src="app/controllers/productControllers.js"></script>
<script src="app/controllers/messageCtrollers.js"></script>
<script src="app/controllers/bidControllers.js"></script>

<!-- SERVICES -->
<script src="app/services/userService.js"></script>
<script src="app/services/registerService.js"></script>
<script src="app/services/loginService.js"></script>
<script src="app/services/productService.js"></script>
<script src="app/services/messageService.js"></script>
<script src="app/services/bidService.js"></script>
```

Figura 5.7. Nivelul de prezentare

În cadrul aplicației se regăsesc o multitudine de pagini la care utilizatorul are acces, în funcție de privilegiile și rolul deținut.

Pagina de înregistrare reprezintă pagina pe care o pot accesa utilizatorii neînregistrați pentru a-și putea introduce datele personale și a avea acces la site-ul de licitații. În momentul în care un client nou al **casei de licitații ArtBidding** dorește să beneficieze de serviciile puse la dispoziție, acesta este nevoit să completeze câmpurile regăsite pe pagina de înregistrare, și anume: nume și prenume, numele utilizatorului, adresa de email, numărul de telefon, adresă și parola.

Pagina de login pune la dispoziție utilizatorului posibilitatea de a se loga în sistem prin introducerea username-ului și a parolei. În cazul în care acestea sunt valide, utilizatorul este redirectionat către pagina proprie. În caz contrar, se afișează un mesaj de eroare corespunzător.

Pagina de contact pune la dispoziție tuturor utilizatorilor informații despre locul unde se află casa de licitații, precum și o hartă pentru a vizualiza zona.

Pagina care oferă informații despre **obiectele vândute** în cadrul licitațiilor terminate ajută utilizatorul să-si facă o părere despre obiectele vândute în cadrul licitațiilor desfășurate pe această platformă, precum și prețul de vânzare al acestora.

Pagina personală a utilizatorului oferă informațiile personale, precum și o listă de navigare către celelalte pagini din cadrul sistemului. Utilizatorul își poate modifica datele și naviga la paginile dorite pentru a plasa o miză în cadrul unei licitații sau pentru a propune către admin un obiect care să fie adăugat în cadrul unei licitații viitoare.

Pagina de licitație conține informații despre licitațiile în desfășurare sau cele viitoare, precum și despre produsele aferente acestor licitații. Utilizatorul poate plasa miza dorită pentru un produs și concurează cu alți utilizatori pentru a obține produsul, acesta fiind câștigat de către cel care oferă miza cea mai mare.

Pagina de gestiune a utilizatorilor, produselor și licitațiilor din sistem este dedicată exclusiv adminilor. Aceștia pot realiza operații de adăugare, modificare sau ștergere, precum și vizualiza anumite informații relevante care pot ajuta la dezvoltarea platformei.

Pagina de mesaje constituie pagina la care au acces adminii sistemului pentru a trimite mesaje utilizatorilor în legătură cu licitațiile viitoare sau informații de organizare. Aceștia pot selecta căror utilizatori să trimită anumit mesaj, precum și conținutul acestuia.

Nivelul de prezentare este atent structurat și construit, pentru a le oferi utilizatorilor o experiență cât mai plăcută. Aceștia se pot bucura de o interfață prietenoasă și ușor de folosit de pe orice dispozitiv, design-ul site-ului fiind unul responsive, care se adaptează la dispozitivul folosit.

5.6. Componente

În cadrul acestui subcapitol se reprezintă componentele sistemului și interacțiunea dintre ele. Modul de funcționare și integrarea al componentelor pentru a satisface flow-ul aplicației este regăsit în cazurile de utilizare și urmează a fi detaliat în continuare.

5.6.1. Descriere

Componentele modelează sistemul, iar îmbinarea acestora duce la contruirea unui ansamblu unitar. În continuare se vor descrie principalele componente, precum și rolul acestora în cadrul aplicației.

Utilizatorul reprezintă persoana care folosește aplicația și beneficiază de funcționalitățile acesteia. Utilizatorul se împarte în 3 categorii: neînregistrat, user și admin, fiecare dintre aceștia având anumite privilegii, precum și constrângeri. Pentru a utiliza aplicația, se urmează cazurile de utilizare pentru fiecare utilizator în parte.

Browser-ul constituie interfața grafică pentru accesarea paginilor de Web. Acesta permite utilizatorilor să afișeze text, elemente grafice și alte elemente specifice. Browser-ul funcționează pe baza unor protocoale (exemplu: http), care îl leagă pe utilizator de paginile web stocate pe servere web specializate.

Controller-ele ofera serviciile și metodele aplicate asupra datelor din cadrul aplicației pentru a putea fi gestionate în nivelul de prezentare. Acestea sunt grupate în funcție de entitatea asupra căreia se refera în: UserController, ProductController, AuctionController, BidController, MessageController.

Entitățile (bean-urile) reprezintă modelul aplicației, clasele Java care sunt modelate pentru a putea realiza funcționalitățile dorite. Acestea sunt manipulate prin intermediul setter-elor sau getter-elor.

Baza de date reprezintă locul în care sunt păstrate datele din cadrul aplicației. Persistența datelor este necesară pentru a nu se pierde informațiile referitoare la entitățile existente.

5.6.2. Interacțiune

Interacțiunea dintre componente reprezintă legăturile dintre acestea și flow-ul pe care îl realizează atunci când sunt interconectate.

În continuare vor fi prezentate 2 flow-uri, în care sunt descrise componentele și interacțiunea dintre ele.

Pentru ca utilizatorul să poată plasa un bid în cadrul unei licitații, acesta trebuie mai întâi să se logheze. După ce logarea s-a efectuat cu succes, acesta este redirecționat către pagina personală unde are la dispoziție navigarea către pagina de licitații. După ce apasă pe butonul aferent, se face o cerere către AuctionController pentru a lista toate licitațiile active. Datele sunt citite din baza de date și transmise către browserul web pentru a putea fi vizualizate de către utilizator. Dacă utilizatorul dorește să plaseze o miza pentru unul dintre produsele aflate la licitație, introduce suma dorită în câmpul pus la dispoziție și apasă butonul „Licitează”. Acesta apelează din BidController funcția „insert”, care adauga informația corespunzătoare în baza de date.

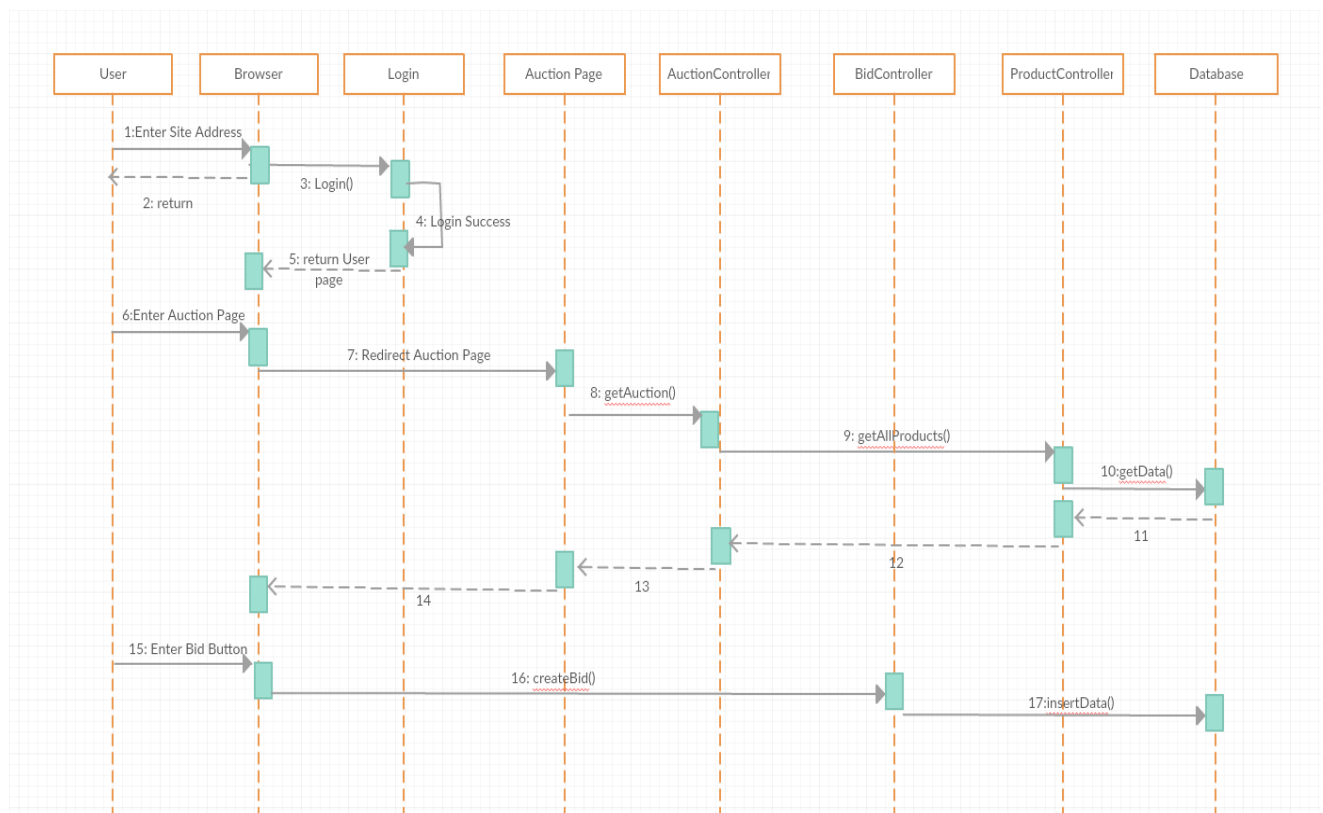


Figura 5.8. Diagrama de Secvențe pentru plasarea mizei

Pentru a vizualiza toate produsele din cadrul aplicației, adminul se va loga în sistem și va accesa pagina de management a produselor. Când apasă pe butonul de vizualizare a tuturor produselor din sistem, se realizează cerere în cadrul ProductController și se apelează metoda getAllProduct, care întoarce din baza de date o listă cu toate produsele și le afișează în browser-ul web.

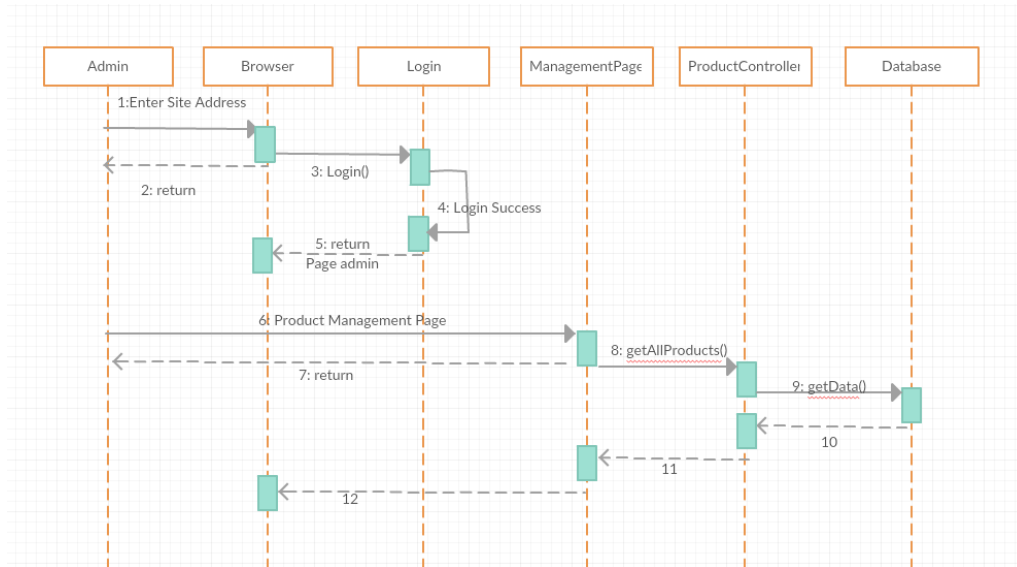


Figura 5.9. Diagrama de Secvențe pentru vizualizarea produselor

5.7. Diagrama de deployment

O diagramă de deployment arată ce componente hardware există (de exemplu: servere), ce componente software se regăsesc în sistem (de exemplu: baza de date, aplicație web) și modul de conectare al componentelor (de exemplu: JDBC, REST, RMI).

Următoarea figură reprezintă diagrama de deployment a sistemului implementat:

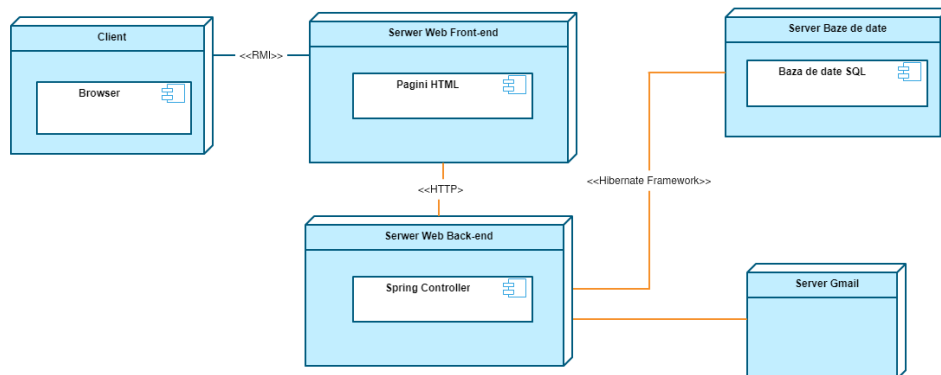


Figura 5.10. Diagrama de deployment

Sistemul dezvoltat presupune un client care dorește să acceseze serviciile puse la dispoziție, acesta efectuând cereri HTTP către serverul web (Glassfish), care gestionează interfața grafică. Serverul Glassfish va realiza la rândul lui cereri către partea de back-end a aplicației pentru a pune la dispoziție datele necesare pentru a putea afișa pagina web în conformitate cu datele existente. Serverul care este destinat pentru managementul părții de back-end, Tomcat, oferă serviciile către partea de front-end prin controller-e Spring, care apelează metodele de business și oferă doar un răspuns cu datele solicitate. Serverul de back-end este conectat și la baza de date a sistemului, prin intermediul Framework-ului Hibernate, precum și la serverul extern de Gmail, prin SMTP (Simple Mail Transfer Protocol).

5.8. Implementare proprie

Sistemul dezvoltat conține un tip de licitație care nu este regăsit în sistemele similare de pe piață, acesta reprezentând o implementare proprie cu scopul de a atrage mai mulți clienți și pentru a dinamiza procesul de licitație.

Ideea proprie extinde licitația de tip englez și îi oferă o latură de incertitudine, câștigătorul neștiind exact suma pe care o va plăti pentru produsul câștigat. Aceasta presupune ca participanții să mizeze deschis unul împotriva celuilalt, ofertele crescând succesiv până când niciun alt participant nu mai oferă o suma mai mare. Persoana care a oferit în final cea mai mare sumă este câștigătoarea licitației. Aceasta este nevoită să plătească suma echivalentă cu o miză aleasă aleator din cadrul istoricului de bid-uri ale licitației câștigate.

De exemplu, dacă pentru un produs se licitează mizele 5, 10, 25, 50, 85, câștigătoare este persoana care a mizat 85 și va plăti o valoare aleatoare din cadrul acestor mize.

Această implementare atrage persoanele care sunt dornice să riște și speră să câștige un produs la un preț foarte avantajos.

Studiu de caz

Presupunem următorul scenariu: Figura [5.8.1] reprezintă repartitia mizelor în cadrul unei licitații. Prețul produsului pe piață este reprezentat cu portocaliu, 4% dintre ofertanți plasând mize apropiate acestei valori. Produsele din cadrul licitației nu au un preț rezervat, astfel încât câștigătorul știe cu certitudine că va câștiga produsul, dar nu și prețul pe care trebuie să îl achite pentru acesta. Având în vedere că în cadrul unei licitații produsul se vinde de obicei la un preț inferior decât cel al pieței, această diagramă modelează perfect comportamentul licitanților și al repartției mizelor. 54% dintre participanți oferă o miză sub prețul pieței, 4% ating pragul prețului real, 28% supralicitează produsul, iar 14% din mize sunt mult peste valoarea prețului real al obiectului, însă participantul consideră că este o șansă foarte bună de a achiziționa produsul la un preț bun (în 54% din cazuri).

Un alt scenariu îl reprezintă supralicitarea produsului încă de la primele mize oferite, astfel sperând înlăturarea concurenței. De exemplu, dacă valoarea unui produs este 500, iar miza primului ofertant este 200, prin supralicitarea produsului la miza 800 se elimină concurența deoarece probabilitatea de a obține produsul la un preț avantajos crește semnificativ cu fiecare ofertă care urmează. Al doilea licitant are șansa 50% de a

câștiga produsul la un preț foarte avantajos (cu 300 unități mai ieftin decât prețul pe piață), dar aceeași șansă de a achiziționa produsul la un preț cu 300 de unități mai mare. Dacă un alt licitant intervine și plasează încă o miză (810), șansă de a obține produsul la un preț avantajos scade la 33%, iar în 66% din cazuri obține produsul la un preț dezavantajos, mult peste valoarea lui pe piață.

Astfel, prin implementarea proprie am creat un sistem atractiv, competitiv și complex, care îi oferă posibilitatea utilizatorului de a-și crea strategii pentru a-i descuraja pe ceilalți participanți să liciteze și pentru a avea o șansă cât mai mare să obțină produsul la un preț avantajos.

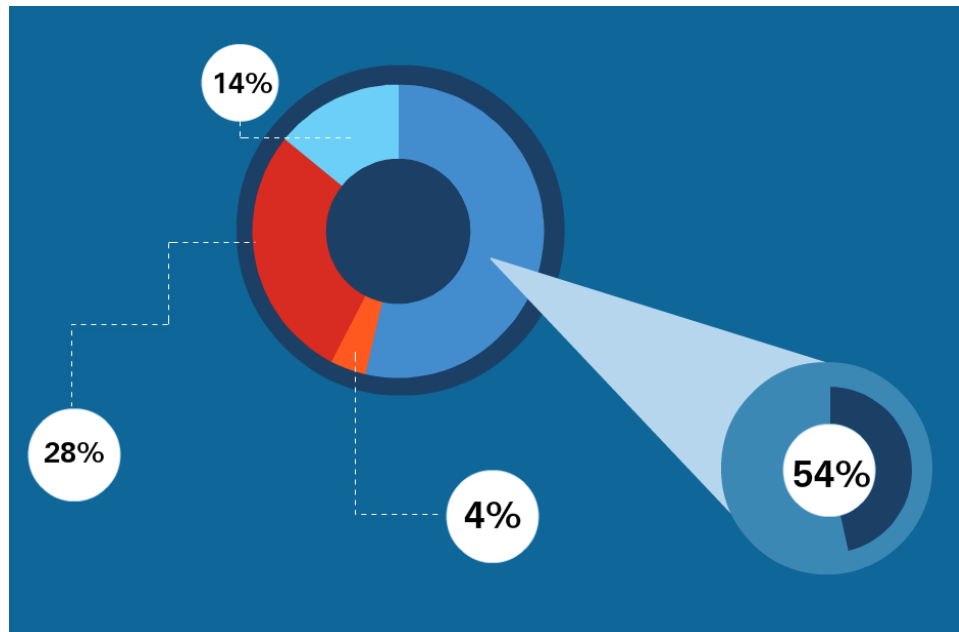


Figura 5.11. Diagrama studiu de caz pentru implementarea proprie

Capitolul 6. Testare și Validare

Testarea software reprezintă un control al aplicației cu scopul de a oferi informații referitoare la calitatea produsului sau a serviciilor, luând în considerație contextul în care acesta va urma să fie folosit. Testarea software pune la dispoziție o viziune obiectivă și independentă asupra produsului în dezvoltare, oferind astfel businessului posibilitatea de a înțelege și evalua riscurile asociate implementării.

În contextul testării, nu există un astfel de produs care permite identificarea tuturor defectelor posibile ce le poate conține o aplicație. De aceea, am decis să folosesc pe lângă testele manuale efectuate în cadrul aplicației, 2 tool-uri externe și anume: Postman și iMacros (alternativa a Selenium-ului).

Testarea API este un tip de testare software care implică testarea interfețelor de programare a aplicațiilor (API) direct și ca parte a testelor de integrare pentru a determina dacă îndeplinesc așteptările privind funcționalitatea, fiabilitatea, performanța și securitatea. Pentru aceasta, am folosit Postman, care permite dezvoltarea API mai rapidă și mai ușoară. Postman beneficiază de o interfață grafică simplă și intuitivă, care păstrează istoricul cererilor, permite testarea automată, monitorizarea timpilor de execuție și a performanței sistemului.

Datele introduse inițial în sistem sunt pentru a testa aplicația, ulterior adaugând date concrete despre utilizatori, produse și licitație.

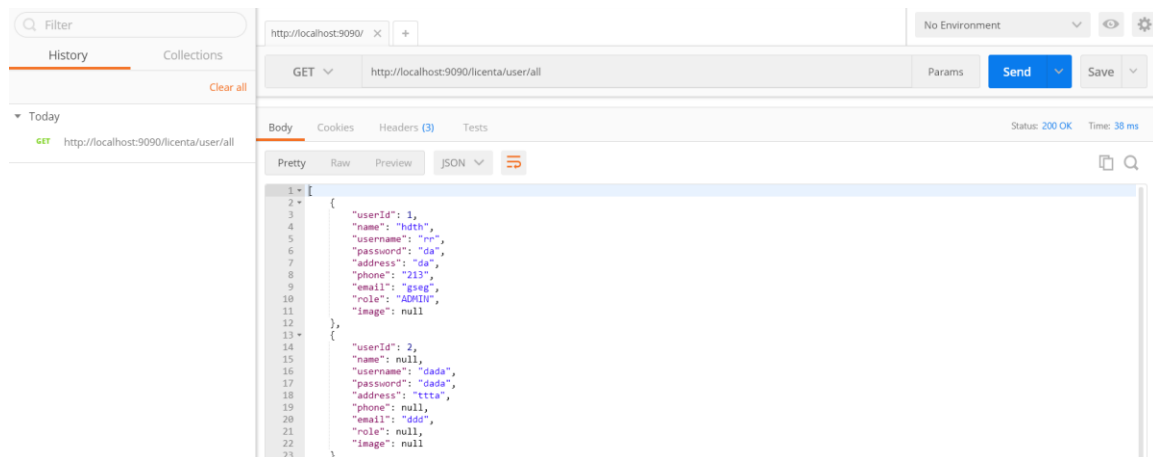


Figura 6.1. Postman – metoda GET

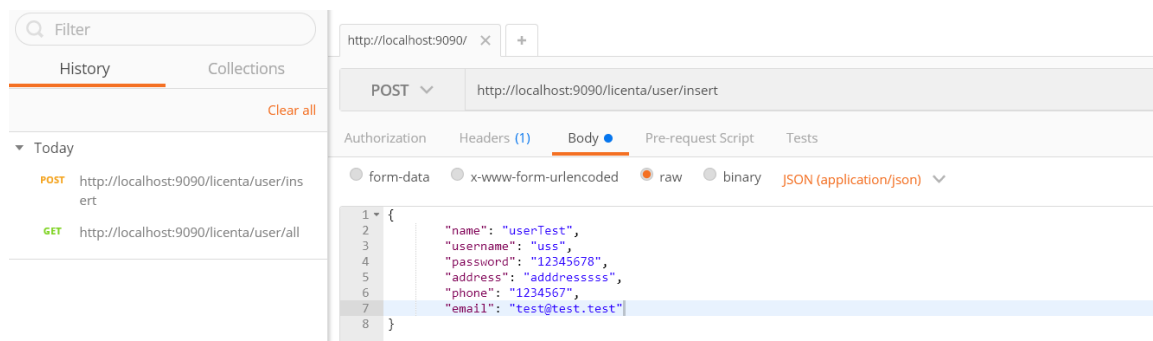


Figura 6.2. Postman – metoda POST

IMacros este o extensie pentru browserele Mozilla Firefox, Google Chrome și Internet Explorer, care adaugă o funcționalitate de înregistrare și reluare similară celei găsite în testarea web. IMacros este proiectat pentru a automatiza sarcinile pe web și a testa comportamentul acestora. În exemplul ce urmează, am înregistrat pașii pentru parcurge paginile importante din cadrul aplicației. Inițial, am setat o pagina de pornire neutră (google.ro), iar ulterior am introdus adresa corespunzătoare **căsei de licitații ArtBidding** pentru a mă redirecționa către pagina de pornire a aplicației dezvoltate. Am intrat pe rând pe pagina de login, unde am introdus datele proprii (valide) pentru a putea fi redirecționat către pagina proprie. Ulterior am vizualizat și pagina de contact, produse și register, iar testarea pașilor urmați prin reproducerea de către tool-ul pus la dispoziție s-a realizat cu succes.

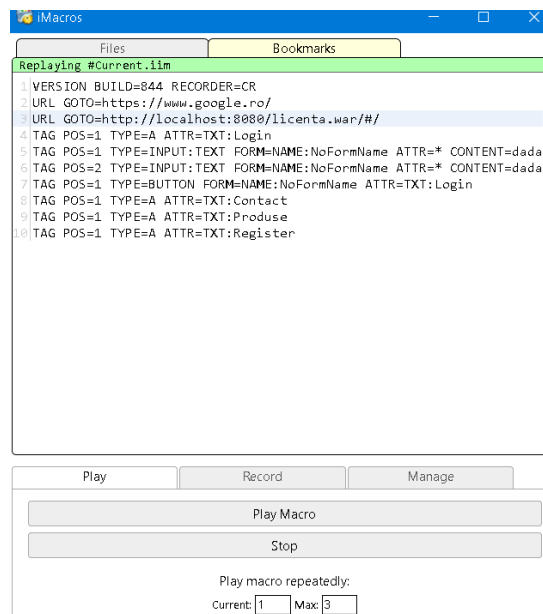


Figura 6.3. iMacros

Validarea datelor din aplicație s-a realizat în partea de front-end, unde s-au gestionat eventualele erori de introduce a datelor sau neconformitatea acestora. În cazul în care utilizatorul introduce date invalide, sau care se regăsesc în sistem (username, email), este afișată o casuță pop-up care specifică eroarea creată.

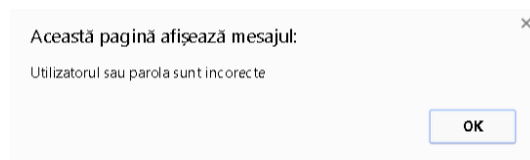


Figura 6.4. Validarea datelor

Capitolul 7. Manualul de instalare și utilizare

În cadrul acestui capitol se descriu resursele necesare rulării proiectului, pașii care trebuie urmați, precum și modul de utilizare al **căsuței de licitații ArtBidding** dezvoltate.

7.1. Manualul de instalare

În continuare sunt descrise principalele resurse pe care trebuie să le conțină sistemul.

Cerințe sistem

Pentru a-i oferi utilizatorului o experiență cât mai plăcută și pentru a funcționa în condiții optime, sistemul trebuie să beneficieze de următoarele:

- 4GB memorie RAM
- Spațiu pe disc 1GB

JAVA

Sistemul trebuie să conțină Java SE Development Kit 8:

- JDK 1.8.0
- JRE 1.8.0

Pentru instalare, se accesează pagina oficială de unde se poate descarca varianta pentru Windows (64 biti) :

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

IntelliJ IDEA (Ultimate)

Se descarcă mediul de programare IntelliJIDEA Ultimate de pe site-ul oficial:

<https://www.jetbrains.com/idea/> - secțiunea download

Tomcat

Se descarcă serverul Tomcat8, în care e gestionat backend-ul aplicației de pe site-ul oficial: <https://tomcat.apache.org/download-80.cgi>

Glassfish

Se descarcă serverul Glassfish4, unde e gestionată partea de front-end a aplicației dezvoltate de pe site-ul oficial: <https://javaee.github.io/glassfish/>

Maven

Se descarcă build tool-ul maven de pe site-ul oficial, de unde se urmăresc și pașii de instalare: <https://maven.apache.org/>

GIT

Se integrează Git în sistem. Acesta se poate downloada de pe site-ul principal, unde sunt explicați și pașii de instalare: <https://git-scm.com/downloads>

1) Pentru a descărca partea de backend a aplicației se urmează pașii:

- Creați un folder gol în spațiul de lucru de pe computer
- Click dreapta în folder și selectați **GitBash**
- Scrieți comenzile:
 - `git init`
 - `git remote add origin https://qe331@bitbucket.org/qe331/auctionsystem.git`
 - `git pull origin master`

2) Pentru a descărca partea de front-end a aplicației se urmează pașii:

- Creați un folder gol în spațiul de lucru de pe computer
- Click dreapta în folder și selectați **GitBash**
- Scrieți comenzile:
 - `git init`
 - `git remote add origin https://qe331@bitbucket.org/qe331/auctionsystemFrontend.git`

`git pull origin master`

Pașii 1) și 2) pot fi evitați, prin copierea surselor de pe CD-ul pus la dispoziție în spațiul de lucru personal.

MySQL

Se descarcă MySQL Workbench 6 de pe site-ul oficial <https://dev.mysql.com/downloads/workbench/> și se urmează pașii de instalare puși la dispoziție tot în cadrul acestui site.

Se creează o schema nouă în baza de date cu numele “licență”.

După urmarea pașilor descriși anterior, se deschid aplicațiile în IntelliJ Ultimate accesând meniul File – Open – LicentaBackend, respectiv File – Open – Open in New Window – LicentaFrontend. Se setează corespunzător serverele pe care urmează să fie deploy-ata aplicația, respectiv Tomcat pentru partea de backend (portul 9090) și Glassfish pentru partea de front-end (portul 8080). După setarea corespunzătoare, se pornește backend-ul, apăsând pe triunghiul verde, iar după ce a pornit, rulăm asemănător și partea de front-end. Utilizatorul poate accesa acum aplicația la adresa <http://localhost:8080/licenta.war/#/> și poate urmări manualul de utilizare din secțiunea 7.2. pentru a urmări principalele utilizări.

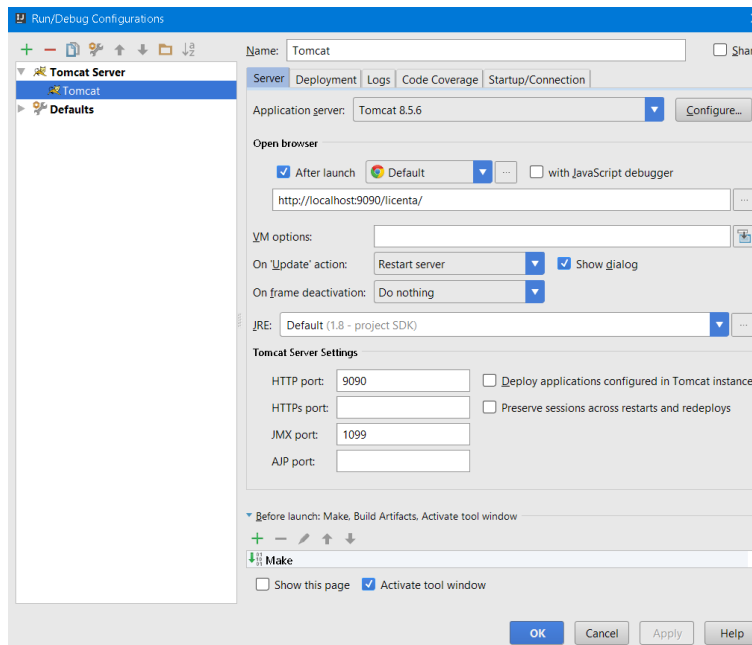


Figura 7.1. Configurare Tomcat în IntelliJ IDEA

7.2. Manualul de utilizare

Utilizarea aplicației este intuitivă, la îndemana utilizatorului, care poate urma pașii descriși în capitolele precedente. Use case-urile constituie un punct de plecare solid pentru utilizatorii neexperimentați, care pot urma cazurile de utilizare pas cu pas pentru a efectua operațiile dorite în cadrul sistemului și pentru a naviga de la o pagină la alta.

Pagina principală a aplicației conține legătura către pagina de login, înregistrare, pagina de contact și pagina unde sunt listate produsele vândute în cadrul licitațiilor terminate. Următoarea figură reprezintă pagina principală a **căsei de licitații ArtBidding**.

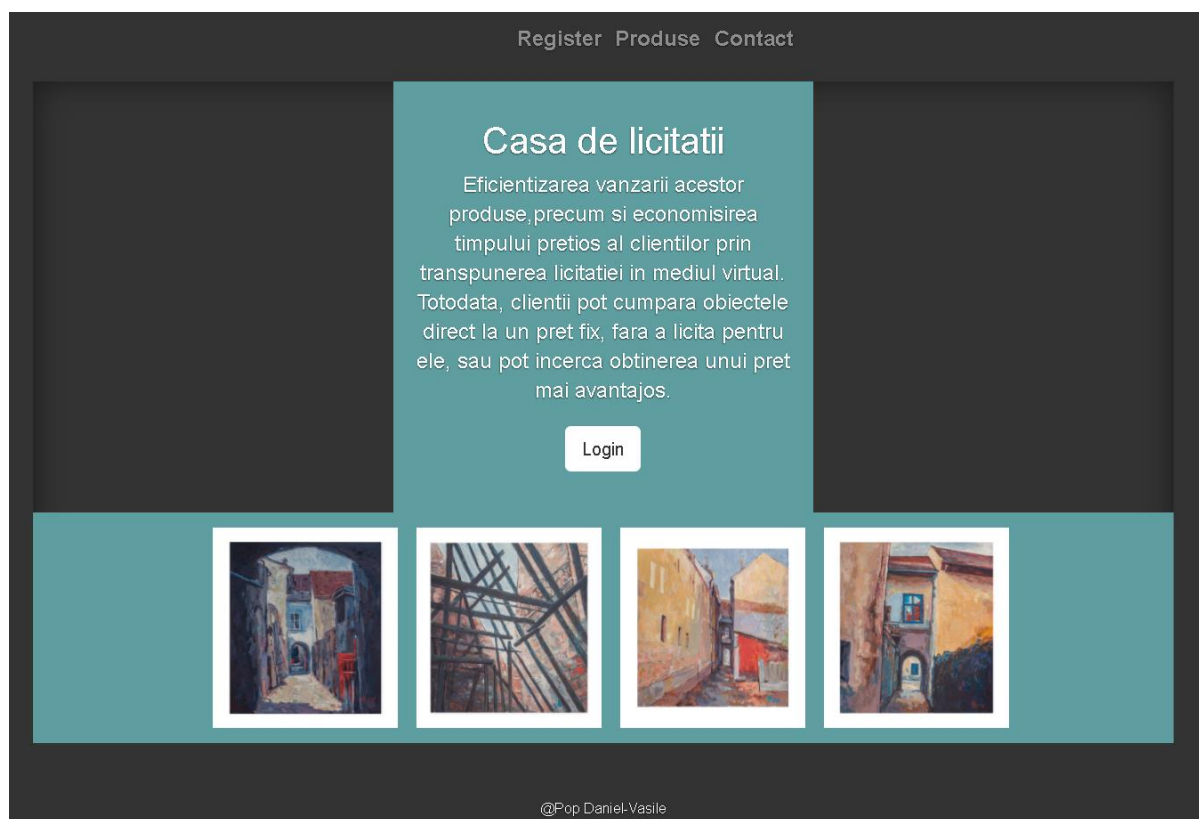
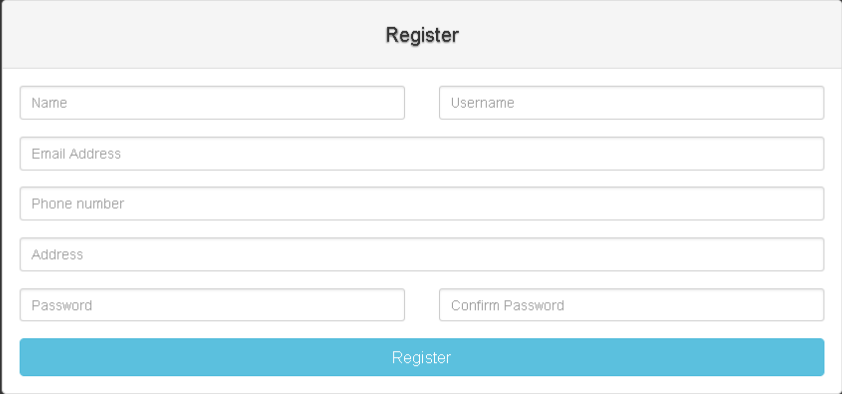


Figura 7.2. Pagina principală

Pentru utilizatorii neautentificați, pagina principală conține în meniul principal pagina de înregistrare, unde aceștia pot intra, își introduc datele personale, iar dacă acestea sunt valide, sunt înregistrați în cadrul sistemului și se pot loga în sistem.



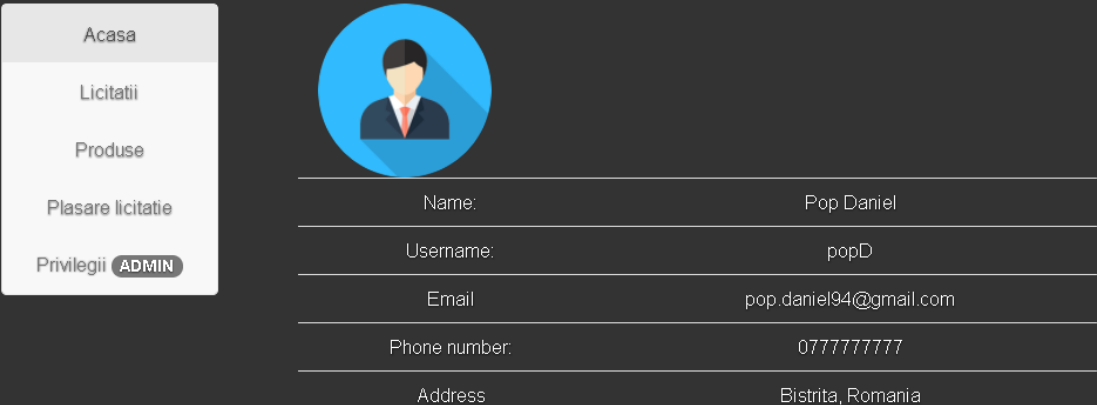
The image shows a 'Register' form with the following fields: Name, Username, Email Address, Phone number, Address, Password, and Confirm Password. A blue 'Register' button is at the bottom.

Figura 7.3. Pagina de register

Pagina de login permite utilizatorului să se logheze în sistem și să beneficieze de funcționalitățile puse la dispoziție de acesta. După logare, utilizatorul este redirecționat către pagina proprie, de unde poate naviga în continuare pe alte pagini dorite.

Pagina de contact prezintă o hartă Google Maps, care afișează locația casei de licitații. Utilizatorul poate manipula această hartă și explora împrejurimile.

Pagina personală prezintă informații despre utilizator, precum și legături către celelalte pagini de interes din cadrul aplicației. Utilizatorul este redirecționat către pagina personală dacă logarea s-a efectuat cu succes. Următoarea imagine reprezintă o vizualizare a paginii personale a unui utilizator.



The image shows a user profile page. On the left is a sidebar menu with items: Acasa, Licitatii, Produse, Plasare licitatie, and Privilegii (ADMIN). The main content area shows a user profile card with a blue circular avatar icon and the following fields:

Name:	Pop Daniel
Username:	popD
Email	pop.daniel94@gmail.com
Phone number:	0777777777
Address	Bistrita, Romania

Figura 7.4. Pagina principală a aplicației

De pe pagina personală, utilizatorul poate naviga către pagina de licitații, unde sunt expuse produsele de artă disponibile pentru care se poate licita. Utilizatorul adaugă în dreptul produsului dorit miza și apasă pe butonul “Licitează”. Miza este înregistrată în sistem, iar ceilalți participanți observă schimbarea prețului curent.

Adminul sistemului are la dispoziție pagini specifice, de management a utilizatorilor, produselor și a licitațiilor, precum și o pagină care generează rapoarte. Acestea sunt ușor de folosit, intuitive, unde adăugarea, modificarea sau ștergerea datelor din sistem se efectuează completând câmpurile puse la dispoziție. Pagina de generare a rapoartelor presupune afișarea unor grafice cu frecvența mizelor pe o perioadă specificată, precum și categoria de obiecte cele mai licitate în cadrul platformei.

Manualul de utilizare este succint, utilizarea platformei fiind foarte ușoară, deoarece s-a pus accentul pe o experiență cât mai plăcută a utilizatorului și o utilizare cât mai intuitivă a aplicației.

Capitolul 8. Concluzii

În acest capitol se prezintă concluziile în urma dezvoltării **casei de licitații ArtBidding**, se menționează realizările efectuate, o analiză a sistemului, precum și posibilitățile de dezvoltare ulterioară.

8.1. Realizări

Casa de licitații ArtBidding reprezintă o modalitate de transpunere a licitațiilor clasice în mediul virtual. Rolul acesteia e de a pune în legătură cumpărătorii și vânzătorii de obiecte de artă, într-un mod simplu și eficient pentru ambele părți. Platforma elimină constrângerile globale, sistemul putând fi folosit de către orice persoană, indiferent de zona geografică în care se află. Utilizatorii sunt scutiți de costurile la care erau expuși în momentul în care participau la o licitație clasică (transport, provizii), iar prin utilizarea acestei aplicații economisesc timp prețios, știind exact momentul de începere și de terminare al unei licitații.

Realizarea principală îl constituie site-ul dezvoltat, un ansamblu unitar compus din mai multe componente individuale. Acestea interacționează între ele pentru a realiza funcționalitățile promise și permite o bună structurare pentru o posibilă dezvoltare ulterioară.

Sistemul implementat respectă atât cerințele funcționale, cât și cele non-funcționale descrise în capitolele anterioare. Aceasta se mapează pe structura descrisă și le oferă utilizatorilor facilitățile promise, precum și un site ușor de folosit, intuitiv, care le oferă o experiență plăcută.

Principalele caracteristici puse la dispoziție de către sistemul implementat sunt:

- **Înregistrare** – utilizatorii care doresc să utilizeze platforma online de licitații se pot înregistra și devin utilizatori cu drepturi depline
- **Logare** – Logarea presupune redirectionarea utilizatorului pe pagina personală de unde poate naviga pe alte pagini și efectua operațiile dorite
- **Plasare Bid** – utilizatorii pot licita pentru produsul dorit, iar cel care a plasat cea mai mare miză este câștigătorul licitației
- **Trimitere mail** – adminii sistemului pot trimite mail de informare către utilizatori pentru a-i informa în legătură cu licitațiile viitoare
- **Generare rapoarte** – adminul poate genera rapoarte pentru a îmbunătăți experiența utilizatorilor și pentru a vedea produsele cele mai căutate
- **Management-ul site-ului** – management-ul site-ului este realizat de către admin, care poate crea, modifica sau șterge date despre utilizatori, produse și licitație
- **Serviciu de localizare** – pagina de contact prezintă o hartă pusă la dispoziție prin Google Maps API, care prezintă locația **casei de licitații ArtBidding** și a zonei adiacente
- **Metoda sigură de plată** – este realizată prin utilizarea Paypal API, care oferă utilizatorilor posibilitatea de a achita online prețul produsului câștigat în cadrul licitației

8.2. Analiza

Sistemul este unul modern, care se ridică la standardele actuale ale pieței. Acesta a fost dezvoltat folosind ultimele tehnologii, iar structurarea acestuia a fost atent gândită, pe baza sistemelor similare și a normelor actuale. Aplicația a fost gândită pe layere, unde au fost grupate funcționalitățile similare.

Aplicația beneficiază de o caracteristică unică față de alte platforme similare, și anume implementarea unui tip de licitație propriu, care atrage mai mulți clienți și creează un cadru competitiv.

În urma analizei amănunțite a sistemului, acesta aduce următoarele beneficii utilizatorilor:

- Reducerea costurilor prin transpunerea licitației în mediul online
- Construirea unui cadru global, care apropie cumpără de vânzător
- Plata sigură și ușoară pentru produsul achiziționat
- Economisirea timpului, fiind cunoscute data de pornire și de terminare a licitației
- Căutarea unei licitații sau a unui produs într-un mod foarte ușor
- Cadru dinamic și competitiv prin crearea unui tip de licitație propriu

Platforma se concentrează pe o experiență cât mai bună a utilizatorului, fiind ușor de folosit, păstrând caracteristicile și funcționalitățile de pe orice dispozitiv este accesată spre deosebire de alte aplicații similare studiate.

8.3. Dezvoltări ulterioare

Casa de licitații ArtBidding poate fi îmbunătățită prin adăugarea unor facilități noi, a unor servicii care pot extinde funcționalitățile actuale, precum și prin adăugarea unor mecanisme de verificare și protecție împotriva utilizatorilor rău intenționați.

Aplicația poate avea următoarele funcționalități suplimentare:

- **Verificare date introduse de către utilizator** – în momentul înregistrării, datele introduse de către utilizator trebuie să fie verificate înainte de introducerea lor în sistem
 - Număr de telefon – Pentru verificarea numărului de telefon, se poate trimite un SMS cu cod unic care trebuie introdus de către utilizator pentru a termina procesul de înregistrare
 - Date Personale – Pentru a verifica validitatea datelor, utilizatorul este nevoit să trimită o poză cu documentul de identitate sau alte documente oficiale
- **Integrare platforme social media** – sistemul va permite logarea în cadrul sistemului folosind mijloacele social media populare precum Facebook. Astfel se evită înregistrarea în cadrul sistemului, contul creat având datele corespunzătoare profilului de pe Facebook.

-
- **Verificare informațiilor despre vânzător** – utilizatorul trebuie informat în legătură cu datele vânzătorului și istoricul acestuia în cadrul platformei, pentru decide dacă este o persoană de încredere sau nu
 - **Adăugare chat** – sistemul poate implementa un chat pentru a permite comunicarea dintre vânzător și cumpărător
 - **Personalizarea paginii proprii** – această caracteristică presupune oferirea posibilității utilizatorului de a-și modela pagina proprie după bunul plac, alegând culorile dorite, tipul de meniu dorit, fontul, etc
 - **Posibilitatea de selecție a limbii** – deoarece platforma nu prezintă constrângeri geografice, o posibilă dezvoltare ulterioară e reprezentată de adăugarea unui meniu din care utilizatorul poate alege limba dorită

Pe lângă aceste dezvoltări, sistemul trebuie să conțină un mecanism de protecție împotriva atacurilor asupra sistemului care l-ar putea face să funcționeze incorect sau să nu mai fie disponibil. Totodată, sistemul trebuie să înlăture potențialele înțelegeri ale utilizatorilor prin urmărirea rapoartelor privind mizele plasate, iar unde se găsesc anomalii, să se efectueze o cercetare amănunțită.

Alte metode care trebuie combătute sunt următoarele:

- **Adăugarea unui comentariu nejustificat**
- **Neefectuarea plății în momentul câștigării unui produs**
- **Înțelegerea între utilizatori**
- **Licitarea de pe mai multe conturi**

Capitolul 9. Bibliografie

[1] Barry Williams, A Database for an Auction Web Site, 25 Februarie 2001,
Online: http://databaseanswers.org/data_models/auction/index.htm

[2] Petru FILIP, Marcel Ioan BOLOȘ, Cristian Ioan OTGON, Echilibrul Bayes-Nash și teoria jocurilor în angajarea cheltuielilor publice, Volumul XVIII, 2011
Online: http://store.ectap.ro/articole/591_ro.pdf

[3] Anna Kalmykova, Auction Houses and Contemporary Art. Jönköping, 2010
Online: <http://www.diva-portal.org/smash/get/diva2:352090/FULLTEXT01.pdf>

[4] Michele, On line auction system, Free University of Bolzano, Faculty of Computer Science, 2003/2004
Online: <http://pro.unibz.it/library/thesis/00001270.pdf>

[5] Milgrom P, Putting Auction Theory to Work. Cambridge: Cambridge University Press. - chapter 6, 2004
Online:
[ftp://nozdr.ru/biblio/kolxo3/G/GA/Milgrom%20P.%20Putting%20auction%20theory%20to%20work%20\(CUP,%202004\)\(ISBN%200521536723\)\(393s\)_GA_.pdf](ftp://nozdr.ru/biblio/kolxo3/G/GA/Milgrom%20P.%20Putting%20auction%20theory%20to%20work%20(CUP,%202004)(ISBN%200521536723)(393s)_GA_.pdf)

[6] IPCSIT, Online Auction Software Fundamentals, International Conference on Computer Engineering and Applications, vol.2, 2011
Online: <http://www.ipcsit.com/vol2/47-B128.pdf>

[7] Ken Goldberg, Online Auction Database Project, IEOR 115, 2014
Prezentare Powerpoint: <http://www.ieor.berkeley.edu/~ieor115/labs/DP-fa2014/project-team-7-115-f14.pdf>

[8] Best Auction Software | 2017 Reviews of the Most Popular Systems, Capterra
Online: <http://www.capterra.com/auction-software/>

[9] BidsOnline Auction Systems, Paul Taylor, 2014
Online: <http://www.capterra.com/auction-software/>

[10] Shanthi Potla, Online auctioning system, A Thesis Presented to the Faculty of San Diego State University, 2011
Online:
http://sdsu-dspace.calstate.edu/bitstream/handle/10211.10/1377/Potla_Shanthi.pdf

[11] Design of a Real-Time Auction System, Department of Computer Science and Engineering, Indian Institute of Technology Guwahati
Online: https://www.iitg.ernet.in/gb/papers/auct_paper.pdf

[12] Wikipedia – Licitație

Online: <https://en.wikipedia.org/wiki/Auction>

[13] Echilibrul licitațiilor

Online: <https://www.historia.ro/sectiune/portret/articol/matematicianul-john-nash-o-minte-sclipitoare-si-un-nobel-controversat>

[14] Dave Harman, Pași pentru implementarea unei licitații, 2013

Online: <http://www.esourcingforum.com/archives/2013/10/24/6-steps-to-implement-a-successful-auction/>

[15] Ebay, <https://ro.wikipedia.org/wiki/EBay>

[16] QuiBids, <https://www.quibids.com>

[17] ArtnetAuctions, <https://www.artnet.com/auctions/>

[18] Artmark, <http://www.artmark.ro/>

[19] Hibernate Framework, <http://hibernate.org/orm/>

[20] Arhitectura Hibernate,

https://www.tutorialspoint.com/hibernate/hibernate_architecture.htm

[21] OrmLite, <http://ormlite.com/>

[22] Spring MVC framework,

<https://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html>

[23] Angular Js, <https://www.tutorialspoint.com/angularjs/index.htm>

[24] Bootstrap, <http://getbootstrap.com/>

[25] Paypal API, <https://developer.paypal.com/docs/api/payments/>

Anexa1. Tabele și figuri

Capitolul 1. Introducere	-
Capitolul 2. Obiectivele proiectului	-
Capitolul 3. Studiu bibliografic	Tabel 3.1. Comparație între Ebay și casa de licitații Tabel 3.2. Comparaie între QuiBids și casa de licitații Tabel 3.3. Comparație între ArtnetAuctions și casa de licitații Tabel 3.4. Comparație între Artmark și casa de licitații
Capitolul 4. Analiză și fundamentare teoretică	Figura 4.1. Cazuri de Utilizare pentru Utilizatorul neînregistrat Figura 4.2. Cazuri de Utilizare pentru Utilizatorul logat Figura 4.3. Cazuri de Utilizare pentru admin Figura 4.4. Cazuri de Utilizare pentru sistem Figura 4.5. Scenariu caz de utilizare pentru Utilizatorul neînregistrat Figura 4.6. Scenariu caz de utilizare pentru crearea unei licitații Figura 4.7. Scenariu caz de utilizare pentru gestionarea aplicației
Capitolul 5. Proiectare de detaliu și implementare	Figura 5.1 Diagramă conceptuală Figura 5.2. Structura bazei de date Figura 5.3. Clasa model (Auction) Figura 5.4. Clasele de model și interacțiunea în cadrul sistemului Figura 5.5. Nivelul de controller (Auction) Figura 5.6. Controllerele și actorul principal Figura 5.7. Nivelul de prezentare Figura 5.8. Diagrama de Secvențe pentru plasarea mizei Figura 5.9. Diagrama de Secvențe pentru vizualizarea produselor Figura 5.10. Diagrama de deployment Figura 5.11. Diagrama studiu de caz pentru implementarea proprie
Capitolul 6. Testare și validare	Figura 6.1. Postman – metoda GET Figura 6.2. Postman – metoda POST Figura 6.3. iMacros Figura 6.4. Validarea datelor
Capitolul 7. Manual de instalare și utilizare	Figura 7.1. Configurare Tomcat în IntelliJ IDEA Figura 7.2. Pagina principală Figura 7.3. Pagina de register Figura 7.4. Pagina principală a aplicației
Capitolul 8. Concluzii	-

Anexa2. Glosar de termeni

Bid = sumă plasată în cadrul unei licitații

Platformă online = mediu digital care promovează comerțul electronic între părțile implicate

Framework = structură conceptuală reprezentată de o arhitectură software care modelează relațiile generale ale entităților domeniului

Application Programming Interface (API) = set de definiții de subprograme, protocoale și unelte care ajută la dezvoltarea aplicațiilor

Tag = cuvânt cheie sau termen folosit la asignarea unei bucăți de informații

Responsive = capacitatea de adaptare a design-ului paginilor în funcție de mediul de vizionare (PC, laptop, tabletă, telefon)