UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Live Auctions

Name: Usuc Alexandru
Group: 4

Table of Contents

Deliverable 1

Project Specification

Obiectivul principal al proiectului este de a creea o aplicatie web care sa le permita utilizatorilor sa participe la licitatii online de oriunde s-ar afla. Aplicatia va fi implementata in limbajul de programare Java utilizand framework-ul Spring, iar toate datele din cadrul aplicatiei vor fi salvate in baza de date MySql. Ca si mediu de dezvoltare se va folosi IDEA-ul Intellij. Cand site-ul web este accesat, utilizatorul va vedea o lista cu toate obiectele scoase la licitatie de catre alti utilizatori. Acesta poate sa vizualizeze postarile, sa liciteze si sa scrie comentarii aferente obiectelor. Utilizatorul va avea posibilitatea de a-si creea un cont si de a scoate la licitatie obiecte.

Functional Requirements

Utilizatorul va putea sa-si gestioneze contul usor prin functionalitatile de “Sign up”, “Log in” si “Log out”.

Cand este accesat site-ul prima data de cineva, acesta poate vizualiza lista de obiecte scoase la licitatie, poate accesa obiectele individual pentru a vedea detalii despre ele. Dupa crearea contului, utilizatorului ii va aparea o noua functionalitate si anume, crearea unei postari, in care acesta poate, prin intermediul unui HTML form, sa adauge o imagine, un titlu, o descriere, o categorie si un pret de inceput al obiectului pe care doreste sa-l scoata la licitatie.

- Dacă utilizatorul este conectat, acesta va putea adauga un obiect în „Watchlist”. Dacă articolul este deja in Watchlist, utilizatorul ar trebui să îl poată elimina.
- Dacă utilizatorul este conectat, acesta ar trebui să poată licita pentru articol. Oferta trebuie să fie cel puțin la fel de mare ca oferta de pornire și trebuie să fie mai mare decât orice alte oferte care au fost plasate (dacă există). Dacă oferta nu îndeplinește aceste criterii, utilizatorului ar trebui să i se prezinte o eroare.
- Dacă utilizatorul este autentificat și este cel care a creat postarea, utilizatorul ar trebui să aibă posibilitatea de a „închide” licitația din această pagină, ceea ce face ca ofertantul cel mai mare să devină câștigătorul licitației și face ca postarea să nu mai fie activă.
- Utilizatorii care sunt conectați ar trebui să poată adăuga comentarii pe pagina postarii. Aceasta va afisa toate comentariile care au fost făcute la postare.

Use Case Model 1

Use Cases Identification

Use-case: Utilizatorul isi poate creea cont (Sign up)
Level: Sub-function
Primary actor: Client
Main success scenario: Utilizatorul completeaza un formular de inregistrare, furnizand email-ul personal si porala de 2 ori. Dupa apasarea butonului “Submit” un nou cont a fost creat, iar datele aferente contului sunt stocate intr-o baza de date.
Extensions: Utilizatorul va primi un mesaj de atentionare in cazul in care email-ul si/sau parola nu sunt corecte.

Use-case: Utilizatorul se poate conecta la contul propriu (Log in)
Level: Sub-function
Primary actor: Client
Main success scenario: Utilizatorul completeaza un formular de conectare, furnizand email-ul personal si porala. Dupa apasarea butonului “Submit” utilizatorul va fi redirectionat la pagina principal a site-ului web.
Extensions: Utilizatorul va primi un mesaj de atentionare in cazul in care email-ul si/sau parola nu sunt corecte.

Use-case: Utilizatorul se poate deconecta de la contul propriu (Log out)
Level: Sub-function
Primary actor: Client
Main success scenario: Utilizatorul apasa pe butonul “Log out”, iar butonul de creare a unei postari dispare. Utilizatorul poate doar sa vizualizeze lista de obiecte scoase la licitatie.
Extensions: -

Use-case: Utilizatorul poate creea o noua postare
Level: User goal
Primary actor: Client
Main success scenario: Utilizatorul apasa pe butonul “Create listing” care il va redirectiona catre pagina de creare a postarii. Pe acesta pagina utilizatorul va putea furniza informatii cum ar fi o imagine a obiectului pe care doreste sa-l scoata la licitatie, un titlu, o descriere, o categorie si pretul de inceput.
Extensions: Utilizatorul va primi o notificare pentru a-l atentiona ca obiectul a fost postat cu succes.

Use-case: Utilizatorul poate vizualiza o lista cu toate obiectele scoase la licitatie
Level: User goal
Primary actor: Client
Main success scenario: Utilizatorul deschide pagina principala cu toate obiectele scoase la licitatie
Extensions: -

Use-case: Utilizatorul poate vizualiza detaliile unei postari
Level: User goal
Primary actor: Client
Main success scenario: Apasand pe un obiect de pe pagina principala, utilizatorul este redirectionat catre pagina obiectului unde se pot vedea detaliile obiectului cum ar fi: imagine, titlu, descriere, ultimul pret si comentariile aferente obiectului.
Extensions: -

Use-case: Utilizatorul poate adauga un obiect la un “Watchlist”
Level: User goal
Primary actor: Client
Main success scenario: Pe pagina fiecarui obiect va fi un buton cu rolul de a adauga obiectul respective la o lista de urmarire “Watchlist”
Extensions: -

Use-case: Utilizatorul poate sterge un obiect din “Watchlist”
Level: User goal
Primary actor: Client
Main success scenario: Utilizatorul deschide lista de urmarire “Watchlist” si sterge obiectele dorite apasand un buton
Extensions: -

Use-case: Utilizatorul plasa oferte “bids” obiectelor
Level: User goal
Primary actor: Client
Main success scenario: Utilizatorul deschide pagina obiectului si poate plasa o contra oferta pretului curent
Extensions: -

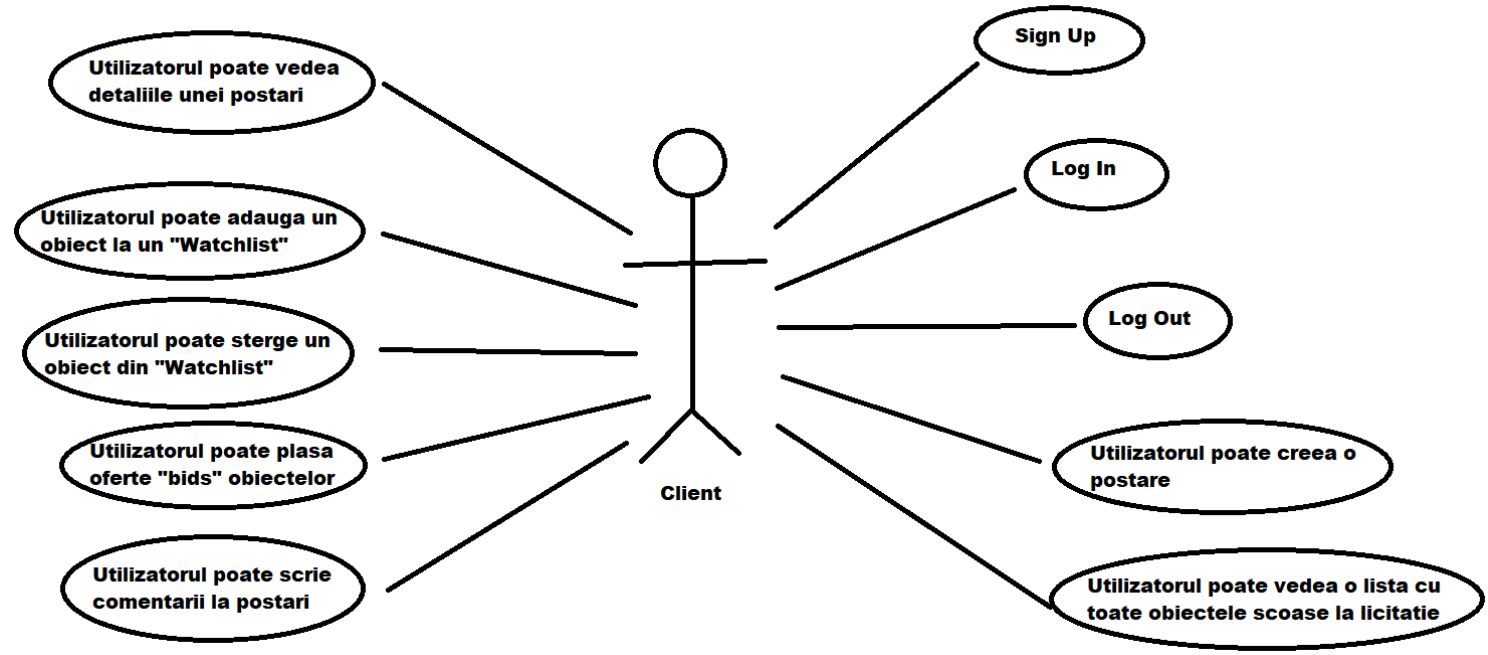
Use-case: Utilizatorul poate scrie comentarii la postari
Level: User goal
Primary actor: Client
Main success scenario: Utilizatorul deschide pagina obiectului unde va exista sectiunea de comentarii. Acesta poate adauga comentarii care vor fi vazute si de alti utilizatori. Comentariile, de asemenea, se vor salva in baza de date MySql.
Extensions: -

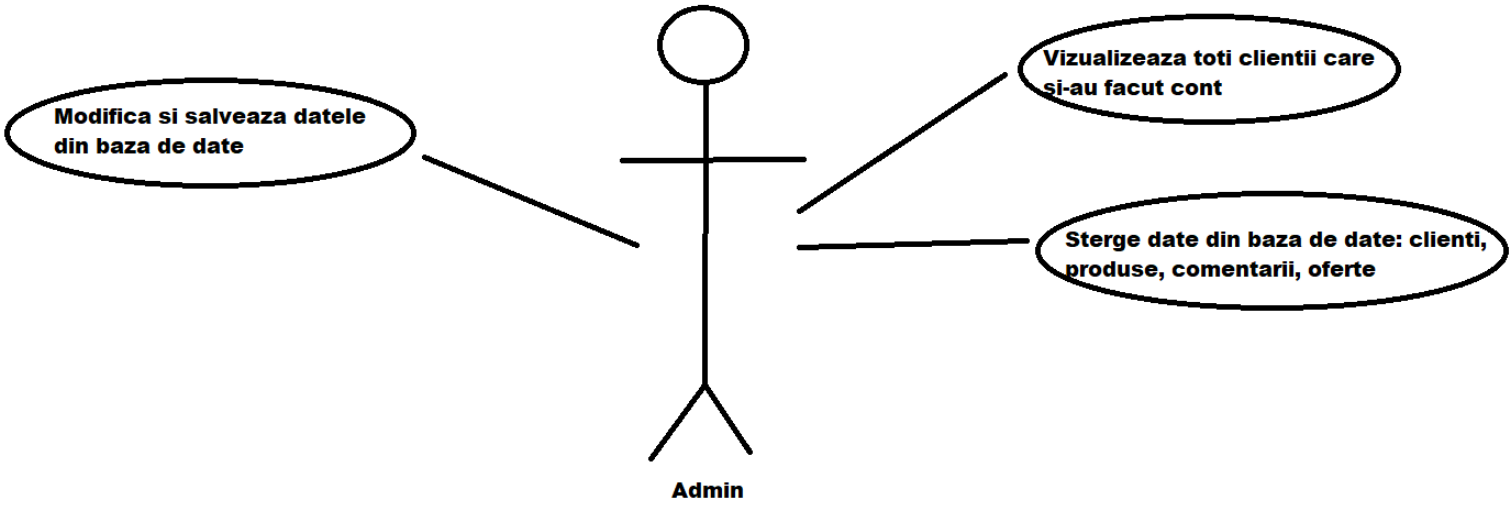
Use-case: Admin-ul vizualizeaza toti clientii
Level: User goal
Primary actor: Admin
Main success scenario: Admin-ul deschide baza de date unde se afla toate informatiile dorite.
Extensions: -

Use-case: Admin-ul sterge date din baza de date
Level: User goal
Primary actor: Admin
Main success scenario: Admin-ul vizualizeaza baza de date si sterge datele dorite
Extensions: -

Use-case: Admin-ul modifica si salveaza baza de date
Level: User goal
Primary actor: Admin
Main success scenario: Admin-ul vizualizeaza baza de date si modifica informatiile dorite. Apoi salveza modificarile pentru a fi vazute si de ceilalti utilizatori.
Extensions: -

UML Use Case Diagrams





Supplementary Specification

Non-functional Requirements

[Choose 4 NF for your system, describe them and explain why these NF are suitable for your implementation.]

- Interfata grafica va fi intuitiva si usor de folosit de catre utilizator.
- Aplicatia va avea un nivel ridicat de Securitate. Utilizatorii nu vor avea acces la baza de date. Parolele utilizatorilor vor fi criptate pentru a nu fi vazute nici de admin care are acces la baza de date. Va fi folosita o functie hash care va transforma parolele intr-o serie de cifre, litere si simboluri.
- Aplicatia va fi fiabila si de incredere. Pentru a obține o fiabilitate ridicată, se vor elimina toate erorile care pot influența siguranța codului și problemele cu componentele sistemului.
- Performanta ridicata. Se vor folosi algoritmi optimi pentru imbunatatirea experientei utilizatorului.

Design Constraints

[This section needs to indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

Aplicatia va fi implementata in limbajul de programare Java utilizand framework-ul Spring, iar toate datele din cadrul aplicatiei vor fi salvate in baza de date MySql. Ca si mediu de dezvoltare se va folosi IDEA-ul Intellij.

Interfata grafica va fi realizata utilizand HTML si CSS.

Glossary

[Present the noteworthy terms and their definition, format and validation rules if appropriate.]

Ca si baza de date pentru stocarea informatiilor despre user si produse am folosit MySql. Baza de date SQL sau baza de date relațională este o colecție de tabele foarte bine structurate, în care fiecare rând reflectă o entitate de date și fiecare coloană definește un câmp de informații specific. Bazele de date relaționale sunt construite folosind limbajul de interogare structurat (SQL) pentru a crea, stoca, actualiza și prelua date.

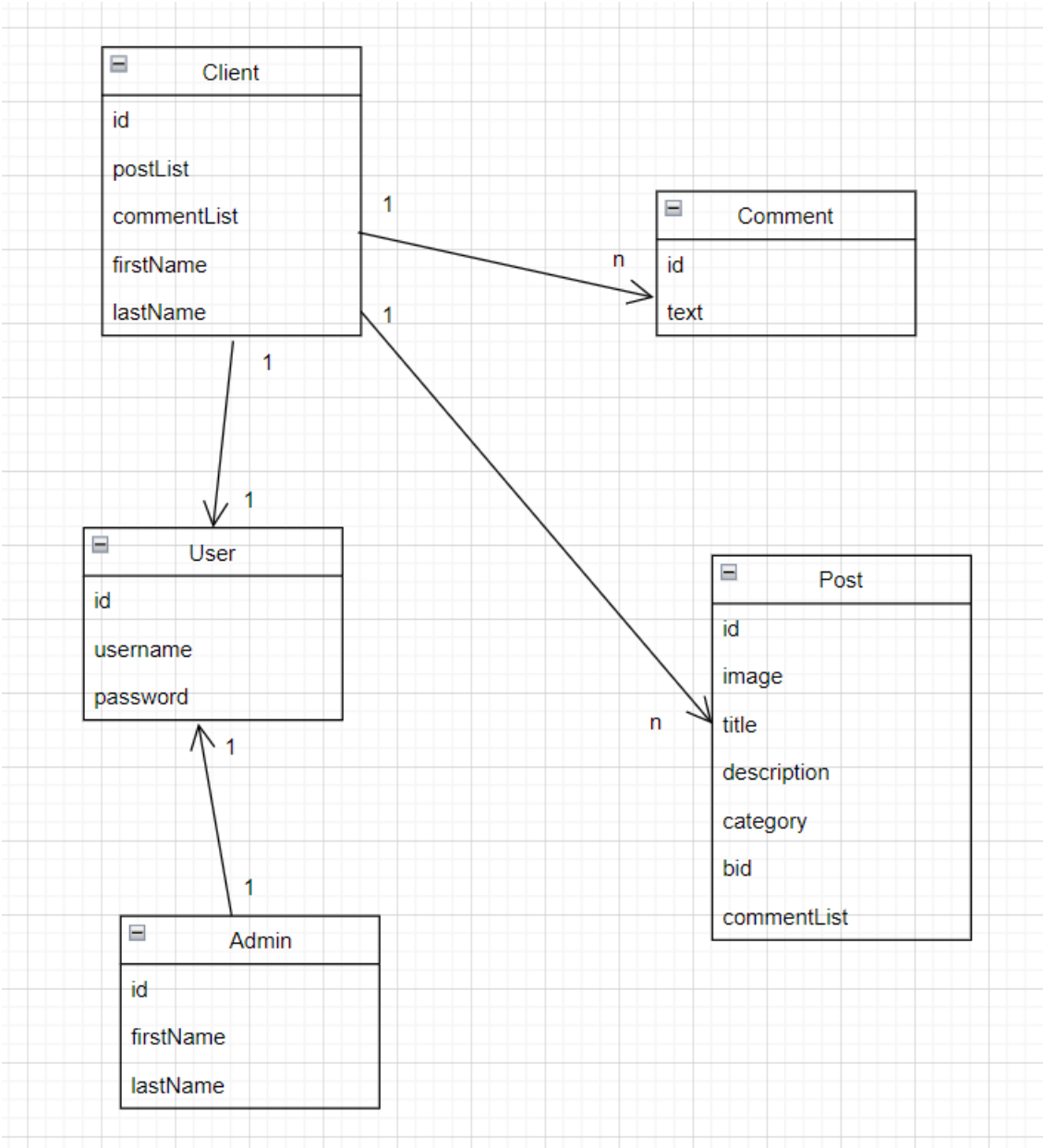
SQL injection (SQLi) este o vulnerabilitate de securitate web care permite unui atacator să interfereze cu interogările pe care o aplicație le face în baza sa de date. În general, permite unui atacator să vadă date pe care în mod normal nu le poate prelua.

Spring Framework este o aplicație open-source care oferă suport de infrastructură pentru dezvoltarea aplicațiilor Java. Unul dintre cele mai populare framework-uri Java Enterprise Edition (Java EE), Spring îi ajută pe dezvoltatori să creeze aplicații de înaltă performanță.

Deliverable 2

Domain Model

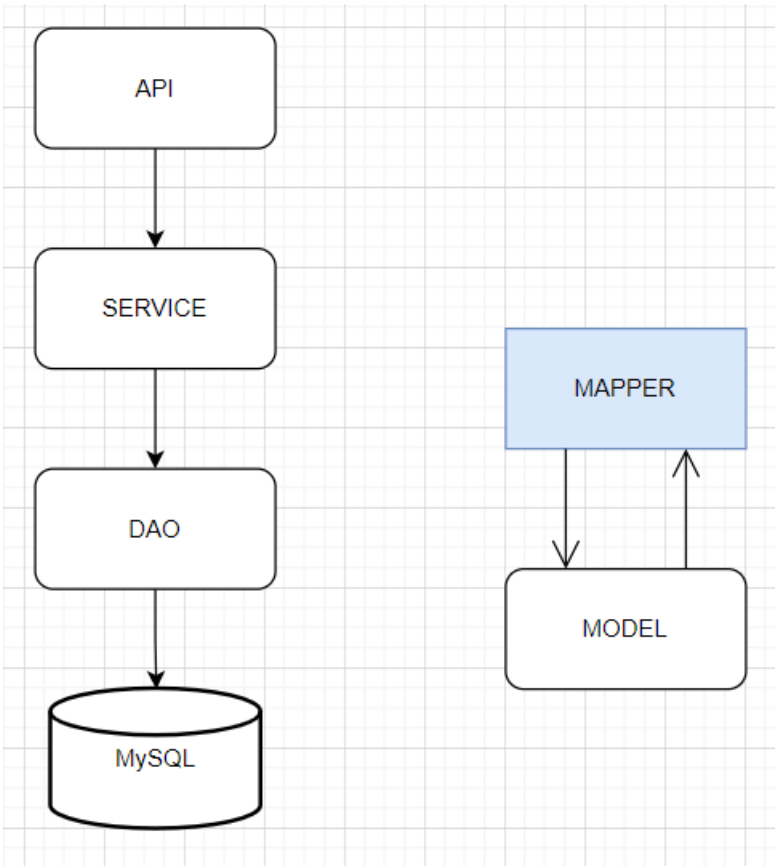
[Define the domain model and create the conceptual class diagrams]



Architectural Design

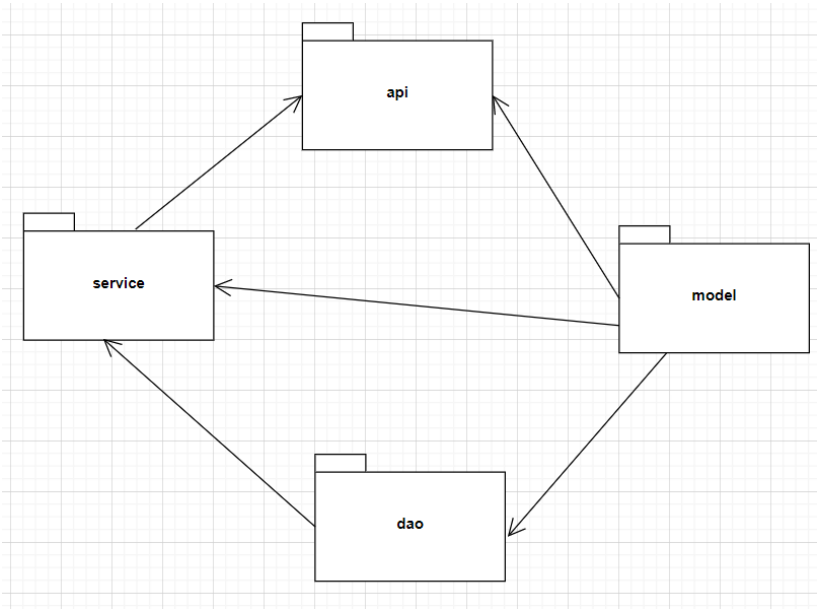
Conceptual Architecture

[Define the system's conceptual architecture; use an architectural style and pattern - highlight its use and motivate your choice.]



Package Design

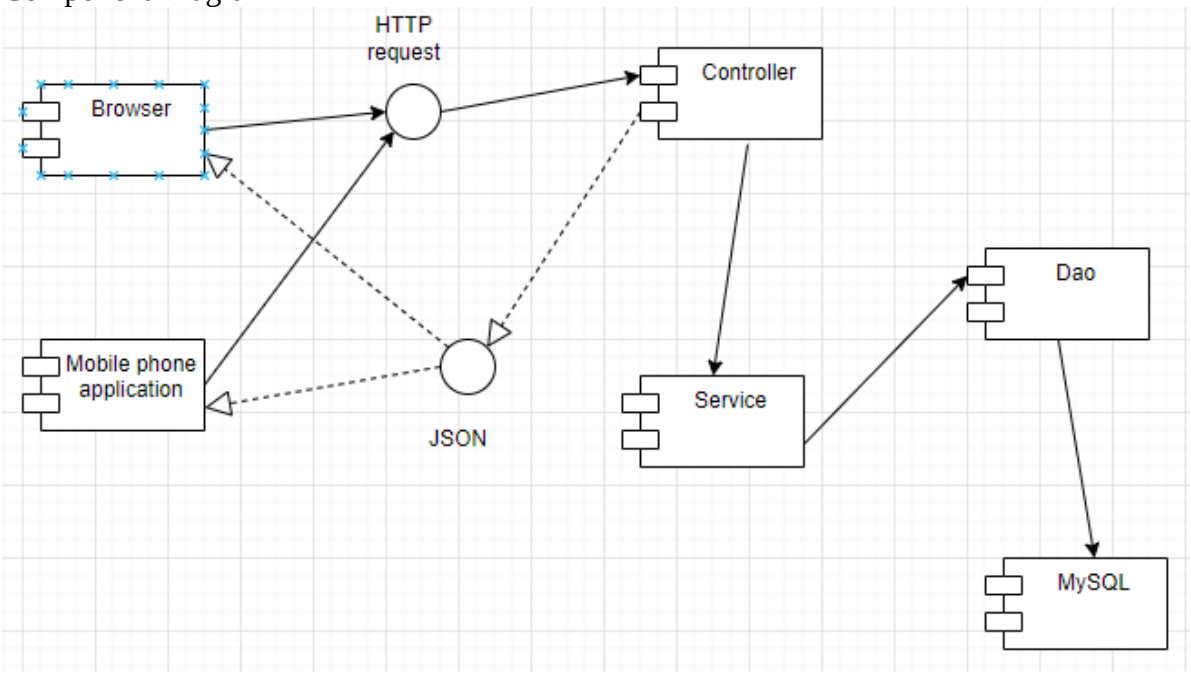
[Create a package diagram]



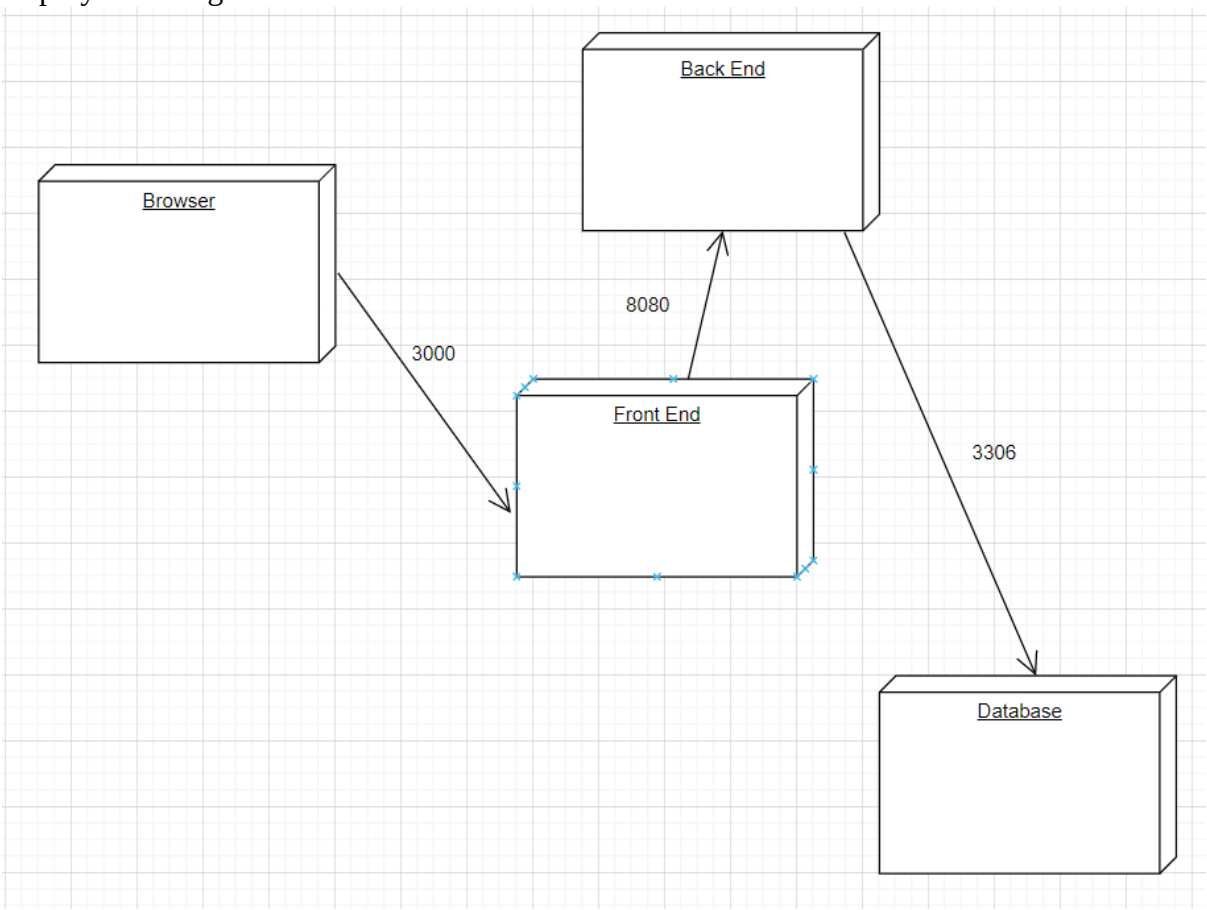
Component and Deployment Diagram

[Create the component and deployment diagrams.]

Component Diagram



Deployment diagram



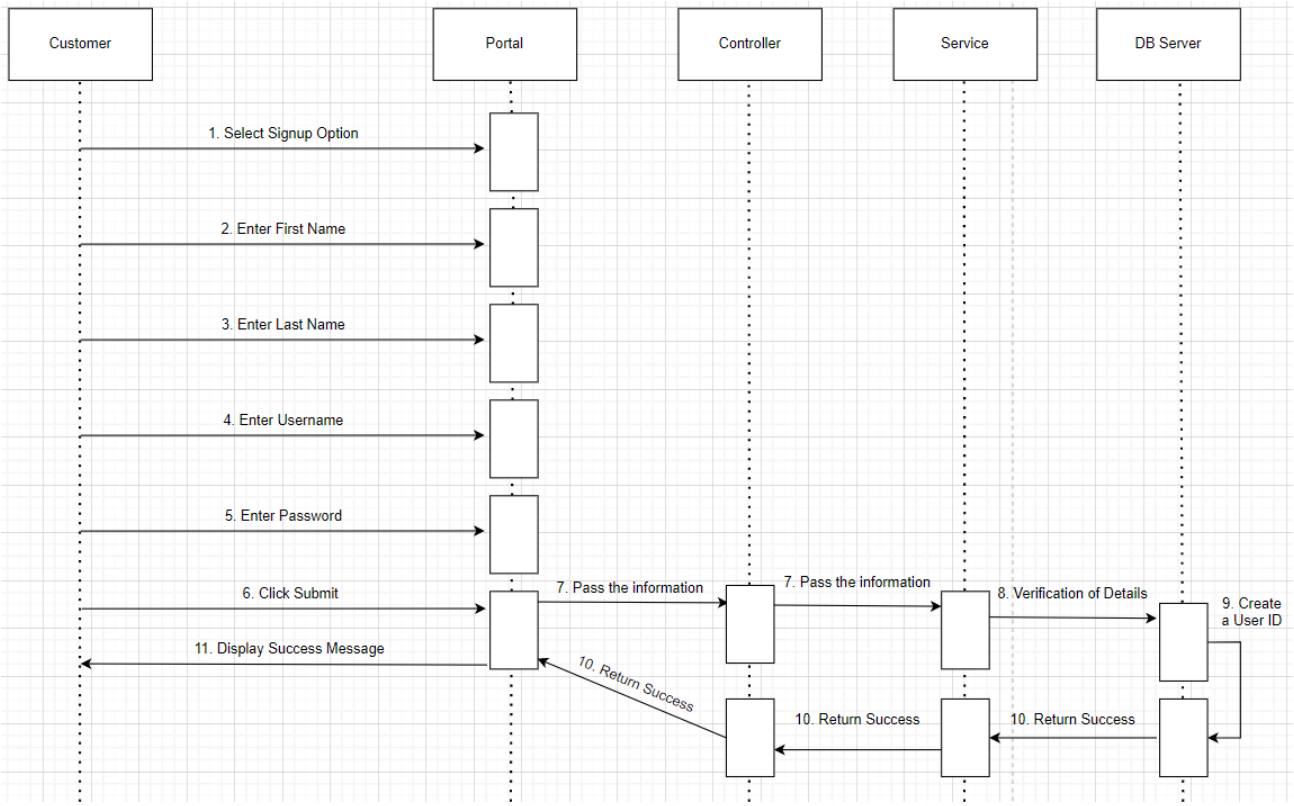
Deliverable 3

Design Model

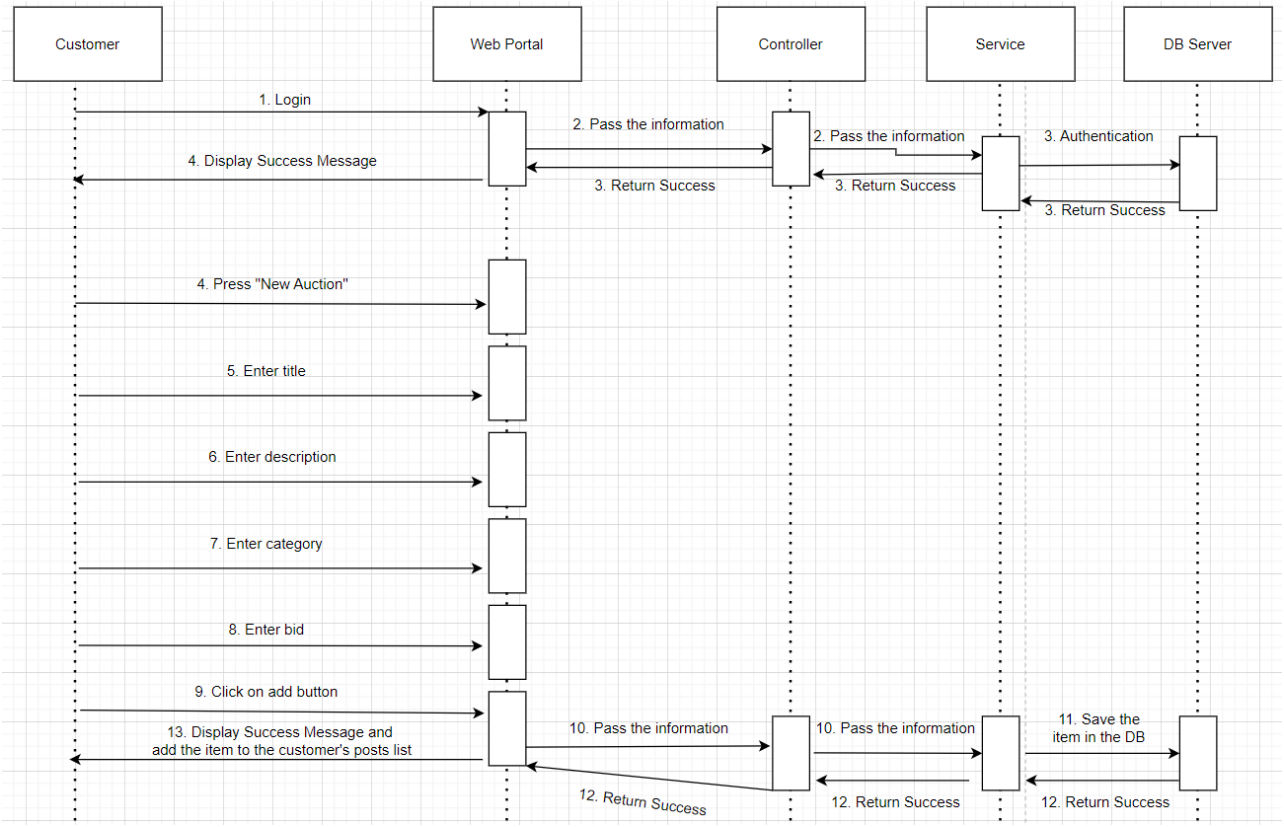
Dynamic Behavior

[Create the interaction diagrams (2 sequence diagrams) for 2 relevant scenarios]

1. Sign up functionality



2. The customer lists a new item for auction

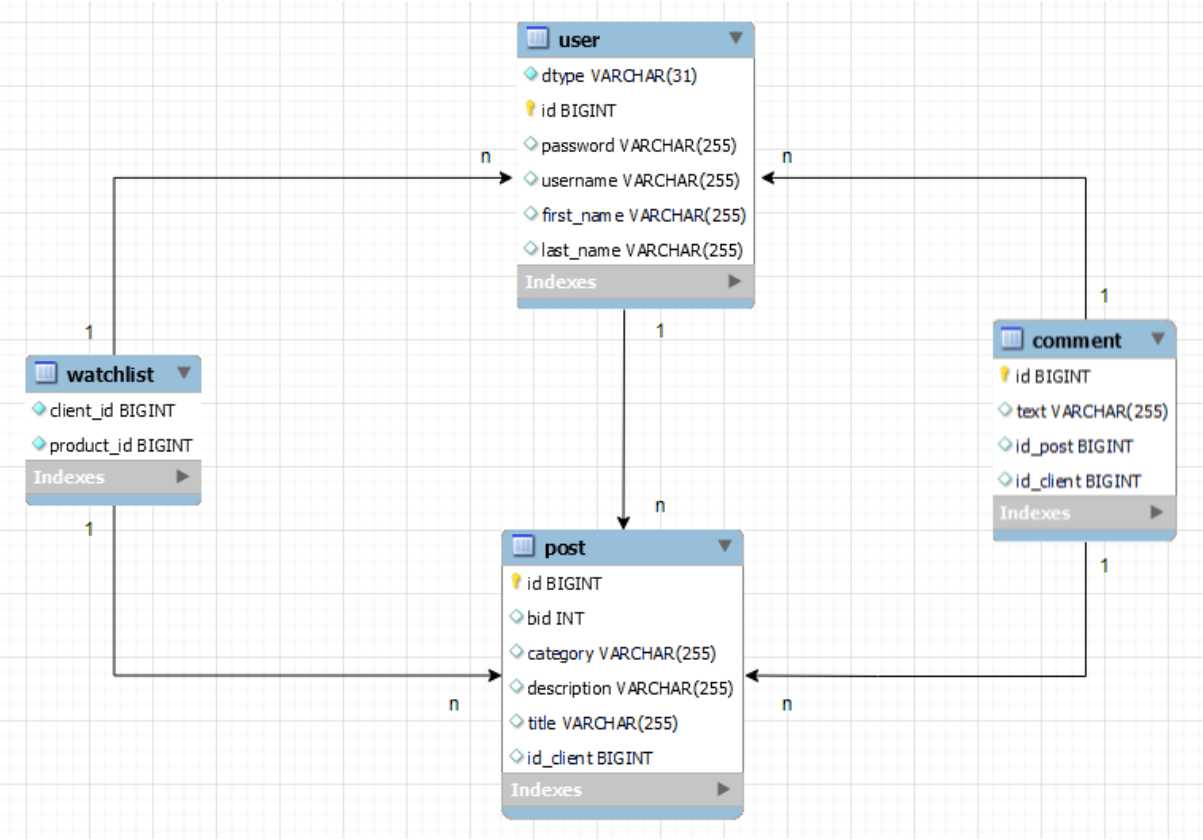


Class Diagram

[Create the UML class diagram; apply GoF patterns and motivate your choice]

Data Model

[Create the data model for the system. Database diagram. It can be generated]



System Testing

[Describe the testing methides and some test cases.]

Pentru testarea funcționalităților aplicației a fost folosit Junit. Au fost testate operațiile CRUD pe baza de date pentru entitațile User, Post și Comment.

Testarea software-ului identifică erori și probleme în procesul de dezvoltare, astfel încât acestea să fie remediate înainte de lansarea produsului. Această abordare asigură că numai produse de calitate sunt distribuite consumatorilor, ceea ce, la rândul său, crește satisfacția și încrederea clienților.

În cadrul entității Post au fost testate următoarele funcționalități:

- aducerea postărilor din baza de date
- crearea unei postari
- actualizarea unei postări

În cadrul entității Client au fost testate urmatoarele funcționalități:

- aducerea clienților din baza de date
- crearea unui client
- stergerea unui client
- actualizarea unui client

În cadrul entității Comment au fost testate urmatoarele functionalitati:

- aducerea comentariilor din baza de date
- crearea unui comentariu
- actualizarea unui comentariu

Alte tipuri de teste:

1. **End-To-End tests:** Testarea end-to-end reproduce comportamentul utilizatorului în cadrul aplicației. Aceasta verifică dacă anumite scenarii funcționează conform așteptărilor, cum ar fi încărcarea unei pagini web, autentificarea, sau scenarii mult mai complexe, verificarea notificărilor prin e-mail, plăți online etc...
2. **Performance testing:** Testele de performanță evaluează modul în care un sistem funcționează într-o anumită sarcină de lucru. Aceste teste ajută la măsurarea fiabilității, vitezei, scalabilității și receptivității unei aplicații. De exemplu, un test de performanță poate observa timpii de răspuns atunci când se execută un număr mare de solicitări sau poate determina modul în care un sistem se comportă cu o cantitate semnificativă de date.

Future Improvements

[Present some features that apply to the application scope.]

Considerand functionalitatile implementate pana in acest moment, aplicatia ar putea fi imbunatatita astfel

1. Adaugarea comentariilor la postari.
2. Comunicarea intre utilizatori prin live chat.
3. Evaluarea si selectia licitatiilor castigatoare. Notificarea castigatorului si gestionarea procesului de finalizare a licitatiei.
4. Feedback si evaluare. Utilizatorii ar trebui sa poata oferii feedback si sa evalueze licitatiile, serviciile si experienta generala cu aplicatia.
5. Sesiuni multiple intre utilizatori
6. Plati si facutare.

Conclusion

In concluzie, in urma implementarii acestei aplicatii web mi-am imbunatatit abilitatile in ceea ce priveste utilizarea limbajelor de programare java, JavaScript, CSS, html, dar si utilizarea unor framework-uri foarte populare cum ar fi Spring Boot pentru backend si ReactJs pentru frontend.

Am avut sansa de a invata lucruri noi cum ar fi securitatea datelor in backend, criptarea parolelor, dar si utilizarea websocket-urilor pentru notificarea utilizatorilor de fiecare data cand un nou obiect a fost scos la licitatie.

Ambele tehnologii, Java și ReactJS, ofera un nivel ridicat de flexibilitate și scalabilitate. Java este un limbaj robust și versatil care poate gestiona aplicatii complexe, in timp ce ReactJS este un framework JavaScript popular, cunoscut pentru performanta sa in construirea interfetelor de utilizator interactive. Combinatia celor doua tehnologii poate permite dezvoltarea unei aplicatii flexibile, usor de extins si de scalat în functie de cerintele in crestere ale proiectului.

Bibliography

https://dev.to/muratcanyuksel/using-websockets-with-react-50pi
https://www.baeldung.com/java-websockets
https://stackoverflow.com/questions/60269965/why-java-session-getbasicremote-sendtext-not-working-but-has-no-error-messa
https://www.appsloveworld.com/reactjs/200/173/react-hooks-typeerror-cannot-read-property-firstname-of-null
https://refine.dev/blog/common-usestate-mistakes-and-how-to-avoid/
https://www.w3schools.com/tags/att_input_readonly.asp
https://www.w3schools.com/jsref/prop_text_defaultvalue.asp
https://stackoverflow.com/questions/21168101/input-auto-complete-initialize
https://www.geeksforgeeks.org/what-are-inline-conditional-expressions-in-reactjs/https://blog.logrocket.com/react-conditional-rendering-9-methods/
https://www.w3schools.com/js/js_array_iteration.asp
https://www.pluralsight.com/guides/access-data-from-an-external-api-into-a-react-componenthttps://www.bannerbear.com/blog/what-is-a-cors-error-and-how-to-fix-it-3-ways/#solution-1-configure-the-backend-to-allow-cors
https://upmostly.com/tutorials/how-to-refresh-a-page-or-component-in-reacthttps://www.youtube.com/watch?v=vtPkZShrvXQ
https://www.youtube.com/watch?v=HMN8NCSm98s
https://www.youtube.com/watch?v=eI-4990XzC8
https://www.youtube.com/playlist?list=PL4cUxeGkcC9gZD-Tvwfod2gaISzfRiP9d
https://www.youtube.com/playlist?list=PL0Zuz27SZ-6PRCpm9clX0WiBEMB70FWwd