

## Documentația aplicației Editor de text

În acest proiect am implementat un editor de text interactiv și personalizat în cadrul unui sistem de înregistrare/autentificare care acceptă funcții de editare de bază. Am folosit Java și Angular pentru implementare.

### **Limbaaj de programare și motivație:**

Această aplicație a fost realizată în limbajul de programare Java, Angular pentru partea de front-end, Postgres pentru crearea și stocarea bazei de date iar pentru utilizarea ei în mediul de dezvoltare am utilizat JPA.

Java a fost conceput pentru a fi ușor de utilizat și, prin urmare, este ușor de scris, compilat, depanat și învățat decât alte limbaje de programare. Java este orientat pe obiecte. Acest lucru permite crearea de programe modulare și cod reutilizabil. Java este independent de platformă.

Pentru partea frontală a aplicației am ales Angular deoarece este un cadru JavaScript open-source, scris în TypeScript. Google o întreține, iar scopul său principal este de a dezvolta aplicații cu o singură pagină. Ca și framework, Angular are avantaje clare, oferind și o structură standard cu care să lucreze dezvoltatorii.

Am utilizat Java Springboot pentru a configura automată ceea ce înseamnă că aplicația este inițializată cu dependențe prestabilite pe care nu trebuie să le configurem manual. Spring Boot ne-a ajutat să cream o aplicație care rulează în cadrul web. Mai exact, ne-a permis să cream aplicații autonome care rulează singure, fără a ne baza pe un server web extern, prin încorporarea unui server web Jetty în aplicație în timpul procesului de inițializare. Drept urmare, putem lansa aplicația pe orice platformă apăsând pur și simplu comanda Run.

PostgreSQL a fost utilizat ca depozit de date principal pentru multe aplicații web, mobile, geospațiale și de analiză și astfel am utilizat-o și noi în cadrul acestui proiect pentru a crea baza de date în care se stochează utilizatorii împreună cu detaliile fiecăruia, precum și fișierele text corespunzătoare utilizatorului aferent.

Pentru a mentine, modifica, șterge sau a aduce obiecte din baza de date am utilizat Java Persistence API (JPA).

În ceea ce privește partea de aspectul aplicației am folosit HTML și CSS. HTML este un limbaj de descriere structurală a paginilor Web. HTML (Hypertext Markup Language) și CSS (Cascading Style Sheets) sunt două dintre tehnologiile de bază

pentru construirea de pagini web. HTML oferă structura paginii, CSS aspectul (vizual și sonor), pentru o varietate de dispozitive. Alături de grafică și scripting, HTML și CSS sunt baza construirii paginilor web și aplicațiilor web.

Unul dintre cele mai semnificative avantaje ale aplicației noastre este capacitatea sa de a trece cu ușurință de la un sistem computerizat la altul. Capacitatea de a rula același program pe multe sisteme diferite este crucială pentru software-ul World Wide Web, iar aplicația noastră reușește în acest sens, fiind independentă de platformă atât la nivel sursă, cât și la nivel binar întrucât poate rula pe orice sistem computerizat atât timp cât are acces la internet, indiferent de caracteristici sau sistem de operare.

### **Motivatie:**

Există o mulțime de editori de text disponibili. Există aceia care rulează în terminal, într-un GUI, într-un browser și într-un motor de browser. Multe sunt foarte bune, iar unele sunt grozave. Dar uneori, cel mai satisfăcător răspuns la orice întrebare este cel pe care îl construiești singur.

## **Specificatii si analiza sistemului:**

Caracteristici generale ale aplicatiei:

- **Logarea noilor utilizatori:** adăuga posibilitatea utilizatorilor neînregistrați de a crea un cont în cadrul aplicației noastre astfel
  - 1) Se solicita utilizatorului o pereche nume de utilizator și parolă (împreună cu email) precum și repetarea parolei în vederea validării acesteia.
  - 2) Se creează un utilizator nou în cadrul aplicației.
  - 3) Se pune noul utilizator într-un tabel adecvat din baza de date, astfel încât să se poată accesa datele acestuia la nevoie.
  - 4) Se transmite ID-ul noului utilizator serverului pentru a acționa ca o cheie primară.
- **Logarea utilizatorilor deja existenți:**
  - 1) Se solicita utilizatorului numele de utilizator și parola
  - 2) Se verifică dacă această pereche există deja în baza de date a aplicației
  - 3) În cazul favorabil utilizatorului i se deschide fereastra dedicată editorului de text, în caz contrar i se da posibilitatea să-și modifice datele introduse
- **Cerere resetare parola uitată:**
  - 1) Se solicita utilizatorului numele de utilizator aferent parolei uitate
  - 2) În cazul în care acest nume de utilizator există în baza de date a aplicației un email va fi trimis către email-ul specific utilizatorului cu parola acestuia, în caz contrar

o eroare va fi afisata si utilizatorului i se va da optiunea de a rescrie numele de utilizator

- **Utilizarea editorului de text**

Caracteristicile specifice pe care le acceptă editorul nostru:

- **Afișare și derulare de bază:** editorul afișează conținutul unui fișier într-o fereastră împreună cu bare de defilare orizontale și verticale. Editorul funcționează corect când fereastra este redimensionată. Dimensiunea documentului reflectă în mod corespunzător înălțimea curentă și a ferestrei în raport cu înălțimea și lățimea totală a documentului; acest lucru se schimbă atunci când documentul este modificat sau fereastra este redimensionată.
- **Cursor de inserare:** când utilizatorul face clic în fereastră, editorul afișează o bară verticală intermitentă chiar în stânga celui mai apropiat caracter de clic. Bara clipește și se stinge: aprinsă timp de 0,5 secunde, apoi oprită timp de 0,5 secunde și așa mai departe. Dacă mouse-ul este tras cu butonul în jos, cursorul de inserare urmează mouse-ul. Dacă o fereastră pierde focalizarea de intrare (adică, faceți clic într-o altă fereastră) nu mai afișează cursorul; când fereastra își recapătă focalizarea, afișează din nou cursorul.
- **Tastare:** când utilizatorul tastează caractere imprimabile, cum ar fi „A” sau „%”, caracterele sunt inserate în locația cursorului și cursorul avansează pentru a rămâne imediat după caracterul care a fost introdus.
- **Linii noi:** dacă utilizatorul tastează Enter , acesta este inserat un caracter de linie nouă ("\\n") în document și afișează o linie nouă în fereastră.
- **Backspace și ștergere:** dacă utilizatorul introduce tasta Backspace, editorul șterge caracterul imediat dinaintea cursorului (cu excepția cazului în care există o selecție; vezi mai jos). Tasta Delete șterge caracterul din locația cursorului.
- **Selecție:** dacă utilizatorul apasă butonul mouse-ului și trage mouse-ul înainte sau înapoi, editorul creează o selecție corespunzătoare intervalului de caractere dintre caracterul pe care a fost apăsat mouse-ul și locația curentă a mouse-ului. Noua selecție înlocuiește orice selecție anterioară. Selecția trebuie să fie afișată folosind o formă de evidențiere, cum ar fi un fundal colorat .Odată ce a fost făcută o selecție, dacă utilizatorul apasă butonul mouse-ului în timp ce ține apăsată tasta Shift, atunci selecția ar trebui modificată ca și cum butonul mouse-ului nu ar fi fost niciodată eliberat (adică poziția curentă a mouse-ului specifică un capăt al selecției și celălalt capăt al selecției este determinat de locația în care a fost făcut clic cu mouse-ul fără tasta Shift în jos). Dacă butonul mouse-ului este apăsat și eliberat fără glisare, atunci orice selecție existentă este anulată.

- **Ștergerea selecției:** dacă un caracter sau o linie nouă este introdusă într-un moment în care există o selecție, atunci noul caracter trebuie să înlocuiască selecția. Dacă se tastează Backspace sau Delete atunci când există o selecție, atunci editorul ar trebui să omita comportamentul normal pentru aceste taste și pur și simplu să ștergă caracterele selectate.

- **Copiere și lipire:** dacă utilizatorul tastează Control-c, editorul ar trebui să copieze caracterele selectate în clipboard, unde pot fi lipite de alte aplicații. Dacă utilizatorul tastează Control-v, editorul trebuie să citească conținutul clipboard-ului și să îl copieze în locația cursorului (ștergând mai întâi orice caractere selectate).

**Stilizarea textului:** utilizatorul poate alege dintr-o gama de fonduri si stiluri cu scopul customizarii textului

## Fundamente teoretice:

Un editor de text este o aplicație software (un program) folosită pentru editarea (crearea și modificarea) de fișiere text, mai mult sau mai puțin complexă. Spre deosebire de procesoarele de texte, un editor de text nu oferă funcțiile complexe (de exemplu formatarea paginilor) specifice creării asistate de calculator a publicațiilor (numite aplicații de tip DeskTop Publishing, DTP).

Editoarele de text se folosesc îndeosebi pentru scrierea (dezvoltarea) de programe, pagini web, administrarea sistemelor de operare. Un editor de text este, în general, inclus în fiecare sistem de operare.

Designul uman, experiența prietenoasă a utilizatorului sunt foarte importante pentru interfața aplicației. Proiectarea interfeței aplicației este o direcție foarte importantă pentru interacțiunea om-calculator.

Cererea de înregistrare și autentificare poate fi cauzată de necesitatea de a stabili un mediu utilizator sau de nevoia de a colecta informații despre utilizator sau de necesitatea operațiilor stabilite de utilizator etc. Realizarea funcțiilor de „înregistrare” și „autentificare” pot juca, de asemenea, o influență importantă, iar o astfel de muncă poate necesita un design atent și precaut în consecință.

## **Valoarea funcțiilor de înregistrare și autentificare**

În primul rând, am analizat valoarea funcțiilor de înregistrare și autentificare. Analiza poate fi împărțită în trei niveluri de utilizator, funcție de produs și afacere:

### **1. Nivel de utilizator**

Din perspectiva utilizatorului, funcțiile de înregistrare și autentificare vor fi: utilizatorul trebuie să obțină un ID independent în acest produs, pentru a-și reprezenta în mod corespunzător existența independentă. Este posibil ca utilizatorul să fie nevoie să-și înregistreze acțiunile anterioare, cum ar fi utilizarea progresului, documente anterioare, conținutul preferat etc.

### **2. Nivelul funcției produsului**

Din perspectiva funcției produsului, funcția de înregistrare și autentificare va fi: pentru a aduce comoditate pentru sincronizarea datelor pe diferite echipamente, aduce, de asemenea, comoditate pentru comutarea diferitelor conturi în același echipament și menținerea stării anterioare a contului precum și pentru securitate.

## **4. Proiectarea aplicației**

### **4.1 .Descrierea arhitecturii sistemului**

Structurarea aplicației are în vedere:

- Minimizarea cuplării între module (modulele nu trebuie să cunoască detalii despre alte module, astfel schimbările ulterioare sunt mai ușor de implementat)
- Maximizare coeziune pentru module (conținutul unui modul izolează un concept bine definit)

Aplicația folosește adesea o arhitectură stratificată – un șablon arhitectural care permite dezvoltarea de sisteme flexibile în care componentele au un grad ridicat de independență.

- Fiecare strat comunică doar cu stratul imediat următor (depinde doar de stratul imediat următor)
- Fiecare strat are o interfață bine definită (se ascund detaliile), interfață folosită de stratul imediat superior

Arhitectura stratificată conține 4 straturi:

- **Nivel de prezentare** (User interface / Presentation)

Nivelul de prezentare este stratul superior al arhitecturii, fiind partea frontală a aplicației. Acest strat reprezintă interacțiunea cu utilizatorul

- **Nivel logic** (Application Logic/ Service)

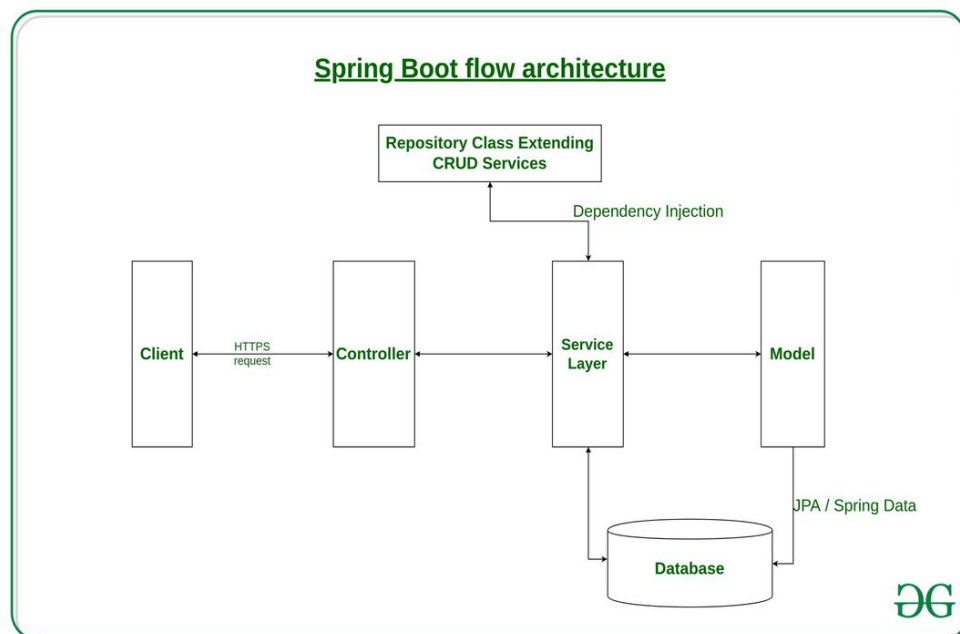
Nivelul logic conține toată logica aplicației, fiind compus din clase de servicii care oferă funcții determinate de cazurile de utilizare. Acest strat se ocupă și de validare și autorizare.

- **Infrastructură** (Repository)

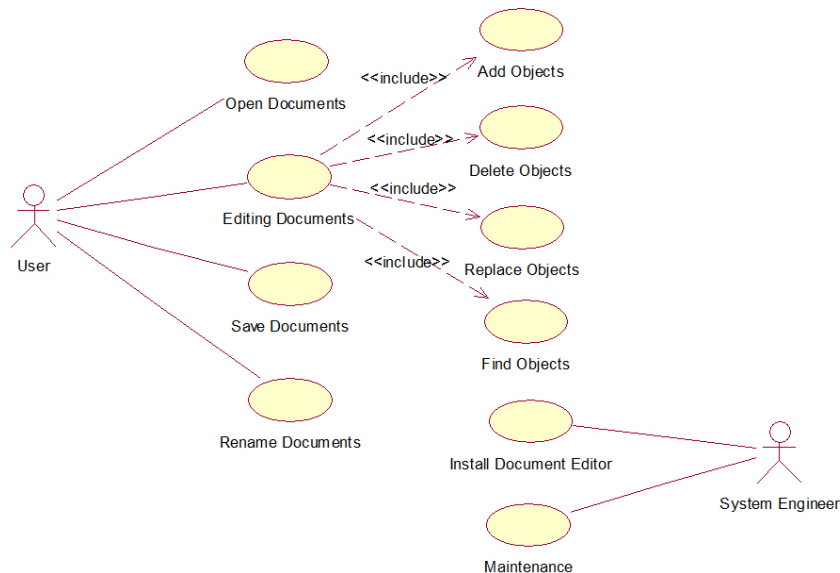
Acest strat poate conține mai multe baze de date. Este responsabil pentru efectuarea operațiunilor CRUD.

- **Coordonatorul aplicației** (Controller)

Se ocupă de solicitările HTTP și realizează autentificarea. Este responsabil de conversia obiectelor Java în JSON și invers.

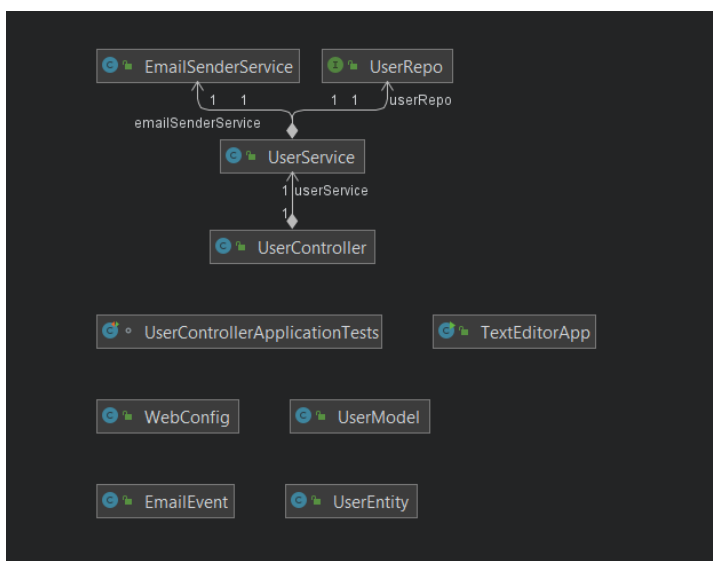


## 4.2. Identificarea funcționalităților aplicației (diagrama de cazuri de utilizare)



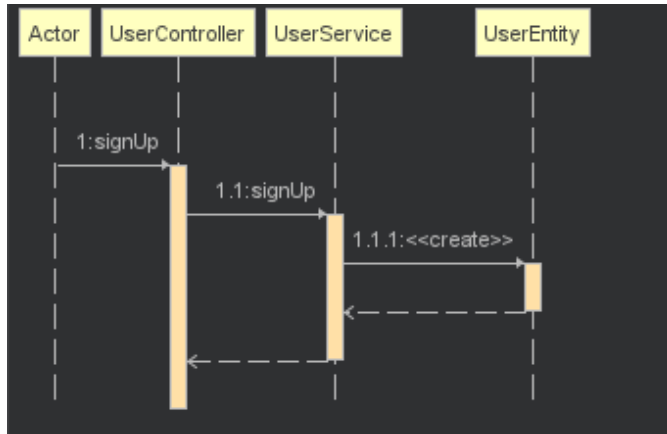
## 4.3. Diagrama de clase

Unified Modeling Language sau UML pe scurt este un limbaj standard pentru descrierea de modele și specificații pentru software. UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțele acestora (numite și obiecte). Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT.

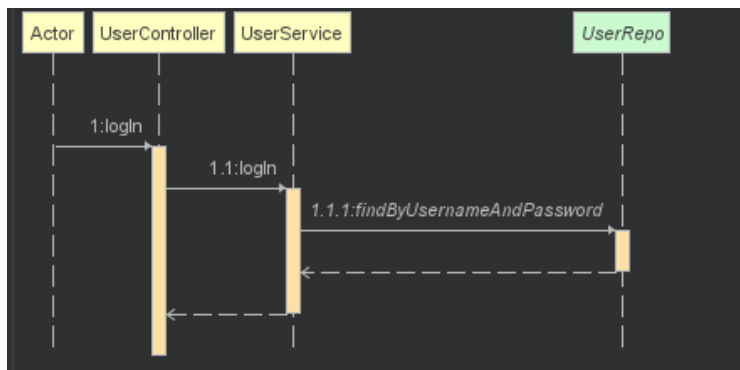


## 4.4.Diagrame de interacțiune:

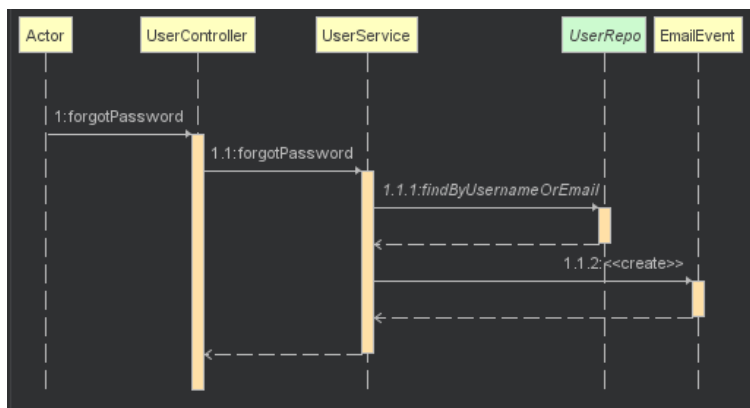
Inregistrarea unui nou user:



Logarea unui user in aplicatie:



Resetarea parolei:





## 4.5.Detalii de implementare

