

# RAPORTARE ȘTIINȚIFICĂ

## Proiect complex ReTeRom. Proiect component CoBiLiRo

### **Activitatea A2.1: Realizarea infrastructurii comune de calcul care va găzdui resursele și instrumentele de prelucrare și acces la corpusul bimodal**

Faza de predare: septembrie 2019

Autori: Dan Cristea, Cristian Pădurariu, Andrei Scutelnicu

Colaborator extern: Șerban Boghiu

#### 1. Rezumatul etapei

A doua etapă (2019) a proiectului CoBiLiRO prevede realizarea infrastructurii pentru gestionarea și stocarea corpusului bimodal. Activitățile complementare prevăzute includ descrierea și implementarea unor soluții de armonizare a reprezentărilor colecțiilor existente text/vorbire (metadata și adnotări). Ulterior, aceste soluții urmează să fie utilizate pentru a armoniza formatele colecțiilor existente și a le încărca pe platforma CoBiLiRO. Similar celorlalte etape, este prevăzută o activitate de diseminare atât la evenimente științifice cât și în mass-media. De interes special pentru platforma dezvoltată este respectarea drepturilor de autor și, atunci când este cazul, anonimizarea contributorilor de resurse vorbite în platformă.

#### 2. Rezumatul activității

Activitatea A2.1 a avut drept obiectiv implementarea specificațiilor de realizare a infrastructurii funcționale de upload și acces la resurse, care urmărește îndeaproape descrierea din livrabilul *A1.3 Proiectarea funcțională și arhitecturală a infrastructurii care va găzdui resursele și instrumentele de prelucrare și acces ale consorțiului și realizarea unui prototip*.

Respectivul livrabil descria: structura de pagini a Platformei și interfața utilizator (secțiunile *Acasă, Despre noi, Resurse și servicii de procesare a limbajului natural, Comunicări și publicații, Parteneri externi și Contact*), funcționalitatea Platformei CoBiLiRo (rolurile de administrator, curator resurse, donator și utilizator de resurse; modalitățile de interogare a bazei de date, lansarea în execuție a unei resurse ca serviciu web, încărcarea și ștergerea unei resurse, operațiile de upgradare/actualizare a unei resurse, conversiile de format, operațiile de back-up - salvare periodică a bazei de date a Portalului), funcționalități auxiliare, precum și propunerea de standard CoBiLiRO, care a ghidat realizarea convertoarelor de format.

#### 3. Descrierea științifică și tehnică

##### 3.1 Introducere

Scopul proiectului complex ReTeRom este ridicarea nivelului tehnologiei limbii române în câteva direcții esențiale: înțelegerea și sinteza vorbirii, prelucrările textuale aplicate limbii române, alinierea resurselor voce text românești, organizarea unui depozit de date de limbă română pentru cercetare și uz public. În particular, proiectul component CoBiLiRo va dezvolta un portal capabil să găzduiască un tezaur de resurse audio și textuale adnotate la diferite niveluri, furnizate inițial de membrii proiectului, ulterior și de alți contribuitori, care va deveni cea mai reprezentativă referință de acest tip pentru limba română. Cu precădere, resursele vor fi de natură bimodală, în care cele două componente sunt aliniată la nivel de propoziție, iar uneori și mai fin, la cuvânt și chiar la fonem. Toate resursele sunt însoțite de metadate, fișiere care descriu informații despre: sursa obiectului memorat în Platformă, identitatea vorbitorului (doar în anumite condiții și cu păstrarea confidențialității), tipul vocii înregistrate (vorbire spontană sau voce în lectură), condițiile tehnice la înregistrării, durata înregistrării, tipul de fișier rezultat (mp3 sau wav), nivelul de segmentare și aliniere (propoziție, cuvinte, foneme), adnotarea ori nu a prozodiei (accente, creșterea sau descreșterea frecvenței fundamentale), părecum și alte informații.

Componentele vocale și transcrierile lor provin din diferite surse: eBook-uri însoțite de texte, înregistrări din piese de teatru dublate de textele corespunzătoare, dialoguri TV sau radiio reproduse în volume tipărite, înregistrări din discursuri parlamentare însoțite de transcrierile stenogramelor etc. În puține cazuri componenta textuală rezultă din transcrieri făcute de sisteme de interpretare automată a vorbirii, când arhitecturi S2T diferite se presupun a face erori de natură diferită, intersecțiile comune fiind reținute, iar restul fiind corectat manual.

## **3.2 Platforma CoBiLiRo**

### **3.2.1 Aplicație**

#### **i. Arhitectură (Cristi)**

Următoarele criterii au fost avute în vedere la proiectarea Platformei CoBiLiRo:

Figura 1 prezintă elementele principale ale funcționalității Portalului. Componentele vorbire-text sunt încărcate în platformă de contribuitori, în general sub formă de perechi de fișiere, iar metadatele sunt completate manual utilizând interfața on-line. În funcție de formatul de intrare este apelat apoi unul dintre convertoare, rezultatul fiind un fișier care respectă formatul standard CoBiLiRo (descriș în livrabilul A1.3). Doar asupra componentei textuale, Portalul apelează apoi lanțul de prelucrări textuale TEPROLIN, care întoarce adnotări complexe. În final, cele patru

componente, voce+text+metadata+adnotări, sunt stocate în Portal. Accesul la aceste componente poate fi făcut de către utilizatori în mai multe maniere, după cum vom descrie mai jos.

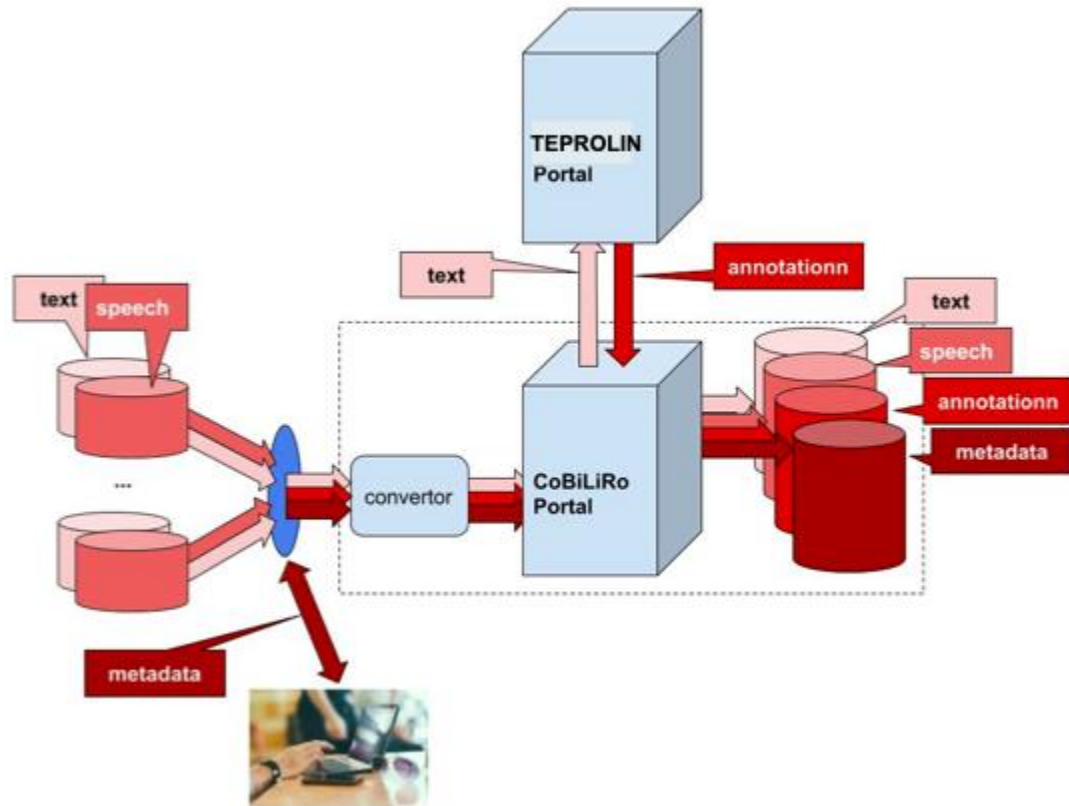


Figura 1: Funcționarea de principiu a Portalului CoBiLiRo

Figura 2 detaliază arhitectura Portalului. Orice acces în upload se face după ce modulul de autentificare efectuează verificările de rigoare. Modulul de autentificare permite utilizatorului acces pe platformă. O data autentificat, acesta poate încărca o resursă bimodală ce va fi preluată și încărcată de către modulul de upload. Metadatele specifice resursei sunt stocate în baza de date MySQL, iar fișierele sunt trimise în continuare spre modulul de conversie. În funcție de formatul de intrare pe care îl are resursa, acesta va fi prelucrat de către unul dintre cele 3 convertoare (descrise în A2.3). După conversie resursa rămâne pe serverul CoBiLiRo în formatul intern al Platformei, iar partea textuală a resursei va fi procesare de serviciul TEPROLIN. Rezultatul procesării și fișierele rezultate în urma conversiei vor fi stocate pe server.

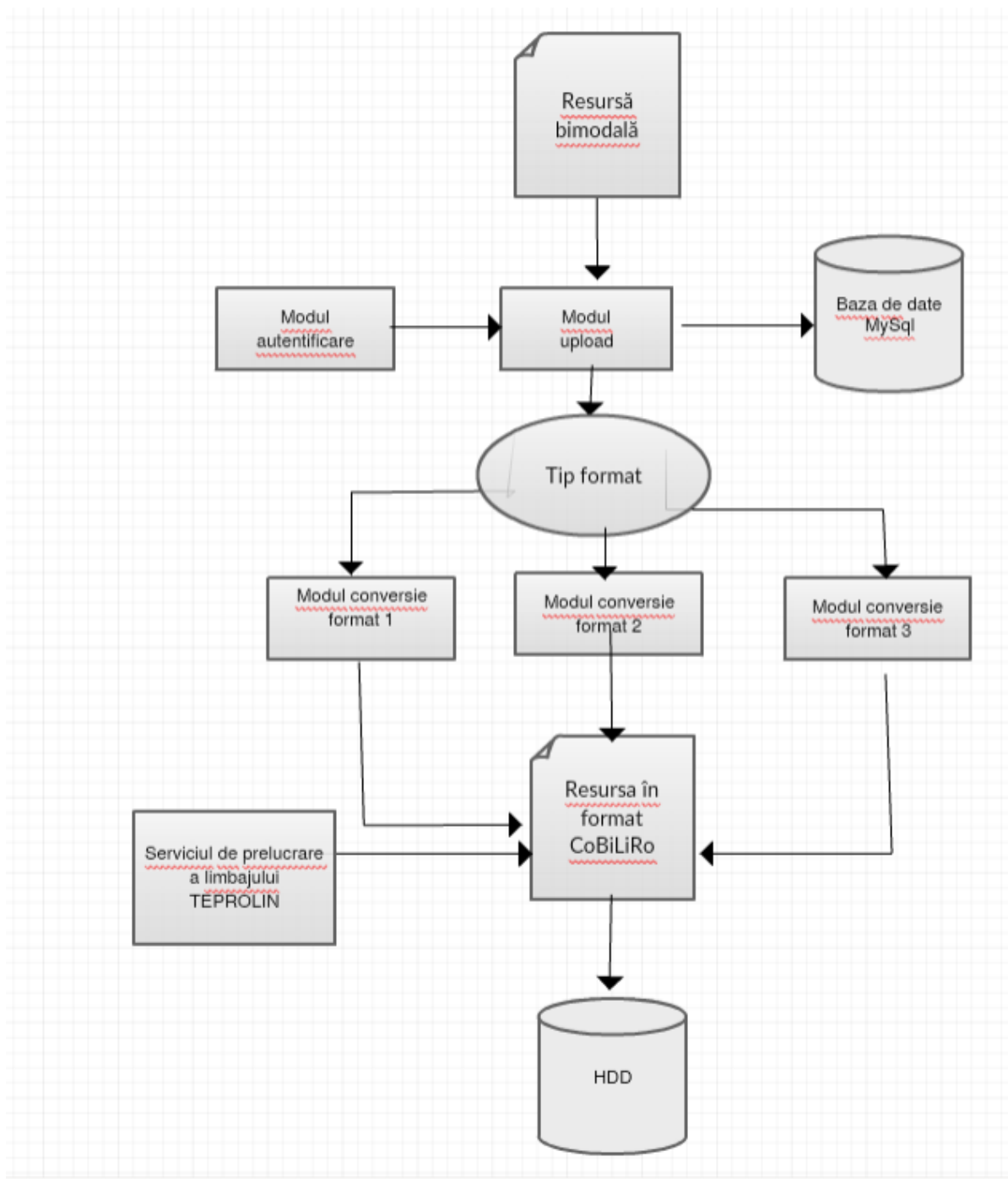


Figura 2: Arhitectura aplicației

### 3.2.2 Baza de date - schema

Figura 3 detaliază schema bazei de date.

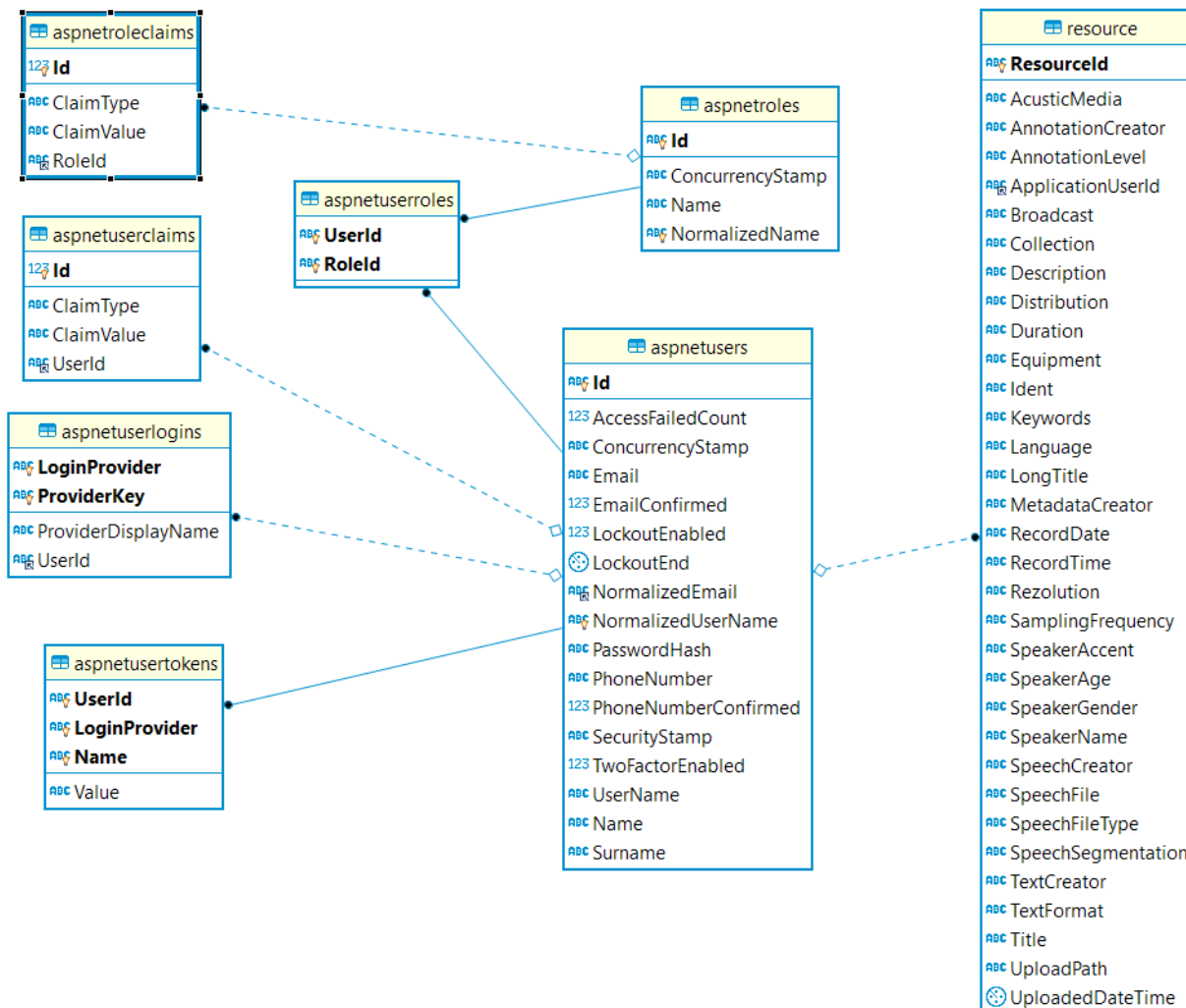


Figura 3: Schema bazei de date

Tabelul *aspnetusers* are ca scop stocarea datelor personale ale unui utilizator (nume, prenume, email etc.). Fiecare utilizator poate avea un singur rol. Aceste roluri sunt definite în tabelul *aspnetroles* (admin, contributor, utilizator simplu). Maparea dintre fiecare utilizator și rolul acestuia în cadrul aplicației este făcută în tabelul *aspnetuserroles*. Permisunile specifice fiecărui rol sunt stocate în tabelul *aspnetroleclaims*. Tabelul *aspnetuserlogins* reține informații legate de login-uri făcute prin alte metode decât logarea obișnuită de pe site, cum ar fi logări cu Facebook, Google, Twitter etc. (această opțiune nu este încă suportată de platforma CoBiLiRo).

### 3.2.3 Tehnologiile folosite

#### i. Limbaje și tehnologii de programare

Platforma este construită folosind limbajul C# din framework-ul ASP.NET Core. Joburile din background se execută folosind platforma Hangfire. Pentru găzduirea serverului web este utilizat Kestrel.

## **ii. Baza de date**

Ca suport pentru bazele de date aplicația folosește serverul MariaDB, pentru că este unul dintre cele mai populare servere de baze de date din lume. Este realizat de dezvoltatorii originali ai MySQL și garantat să rămână open source (gratuit). Deoarece serverul pe care rulează aplicația funcționează sub sistemul de operare Linux, acest server virtual pentru baze de date este printre puținele care poate rula pe Linux. Un alt motiv pentru care s-a ales MariaDB este securitatea informației. Sistemul de securitate al informației este foarte bine pus la punct, atât prin prisma sistemului de operare - Linux, cât și din configurația mașinii virtuale pe care va rula MariaDB.

## **iii. Servicii pentru procesarea limbajului**

Pentru prelucrarea textelor, încărcate ca parte a resurselor bimodale, se folosește serviciul TEPROLIN (<http://89.38.230.23:5000/>), dezvoltat de partenerii de la RACAI. Acest serviciu permite aplicarea mai multor operații asupra textelor cum ar fi:

- Restaurarea diacriticelor
- Transcrierea fonetică a cuvintelor
- Scrierea numeralelor
- Împărțire în propoziții
- Tokenizare
- POS-tagging
- Lematizare
- Recunoașterea entităților nume proprii (NER)
- Chunking
- Parsare sintactică (DP)

Rezultatele acestor prelucrări sunt stocate în format JSON pe serverul CoBiLiRo, ca parte componentă a resursei.

### **3.2.4 Funcționalități**

#### **a. Permiuni roluri**

Platforma CoBiLiRo permite 3 tipuri de utilizatori, fiecare cu anumite niveluri de permisiuni:

- Utilizatorul (user) simplu poate doar vizualiza și descărca resurse.
- Contributorul, pe lângă drepturile unui simplu user, mai poate și adăuga resurse noi.
- Administratorul, pe lângă drepturile unui contributor, poate adăuga/șterge/modifica informații despre un utilizator și poate crea noi roluri.

### b. Modulul de autentificare

Autentificarea se face pe baza adresei de email și a parolei. Aceste date sunt trimise la server, care interoghează baza de date și returnează un răspuns corespunzător. Securitate, crearea și distrugerea sesiunilor etc. sunt asigurate de framework-ul .NET.

### c. Modulul de încărcare fișiere

Contributorul va accesa pagina de creare a resursei și va completa formularul asociat, vizibil în Figura 4. Acesta va încărca fișierele asociate resursei, iar în final va acționa butonul “Crează” ce va permite transmiterea informației către server. Aplicația salvează informațiile din formular într-un obiect (*ResourceViewModel*), iar fișierele resursă într-o listă (*List<IFormFile>*).

Resursă XML

Choose Files No file chosen

Format 1  
Format 2  
Format 3

Colecție <sup>?</sup>

Titlu <sup>?</sup>

Identificator <sup>?</sup>

Titlu complet <sup>?</sup>

Descriere <sup>?</sup>

Cuvinte cheie <sup>?</sup>

Limbă <sup>?</sup>

Creatorul metadatelor <sup>?</sup>

Creatorul alinierii <sup>?</sup>

Nivel de adnotare <sup>?</sup>

Creatorul resursei vocale <sup>?</sup>

Cadru acustic <sup>?</sup>

Durata înregistrării <sup>?</sup>

Frecvență de eșantionare <sup>?</sup>

Selectează limba

propoziție

Figura 4: Formularul de descriere a unei resurse

#### **d. Modulul de conversie a fișierelor**

Acest modul are ca scop prelucrarea fișierelor și metadatelor oferite spre a putea fi disponibile în formatul descris în livrabilul A1.3.

Pentru conversie se utilizează un *AutoMapper*, care realizează conversia automată între două obiecte aparținând unor clase diferite, pe baza unor reguli definite de programator. Utilizând această opțiune se realizează conversia de la obiectele *ResourceViewModel* și *List<IFormFile>* la obiecte de tip *Tei*, care conțin un *TeiHeader* și o listă de *TeiUnits*. Regulile mapărilor sunt definite sub formă de perechi de tip *clasa1.atribut => clasa2.atribut*, care sugerează corespondența dintre atributul sursă și atributul destinație. Detaliile de implementare se pot observa în anexa 1, iar o descriere detaliată a procesului de conversie este prezentă în livrabilul A2.3.

#### **e. Funcționalități de adăugat în viitor**

În viitor se dorește adăugarea unei funcționalități de căutare și filtrare a resurselor. De asemenea, trebuie luat în considerare crearea unui sistem robust de tratare a tuturor tipurilor de erori ce pot apărea la momentul încărcării unei resurse cum ar fi: lipsa unor fișiere, formatarea acestora etc.

Trebuie implementat un sistem care să ofere contributorului un feedback legat de stadiul în care se află procesul de upload. Acesta ar trebui să știe când s-a terminat upload-ul, când a început prelucrarea TEPROLIN și când s-a terminat salvarea fișierelor pe disk.

De asemenea, avem în vedere implementarea unui mecanism de gestionare a încărcării fișierelor foarte mari (>1GB) folosind un sistem similar torrentelor.

### **3.2.5 Manual de utilizare**

#### **a. Autentificarea**

Pentru autentificare se vor folosi conturile deja create, care au fost trimise fiecărui departament prin email. Acestea au asociate anumite roluri și, respectiv, permisiuni ce reglementează accesul și colaborarea pe platformă. Utilizatorii pot fi, de asemenea, invitați folosind o adresă de e-mail care ulterior trebuie confirmată.

#### **b. Meniul administrativ**

##### **i. Adaugă user nou**

Administratorii au opțiunea de a adăuga un utilizator nou folosind o adresă de email validă.

##### **ii. Vizualizare / editare useri**



Administratorii au dreptul de a vizualiza și edita userii și datele lor personale de pe platformă.

### **iii. Adaugare de noi roluri**

Administratorii pot modifica rolurile utilizatorilor existenți, adăugând noi roluri, fiecare cu permisiuni specifice, ce pot fi configurate.

### **iv. Vizualizare / editare roluri**

Administratorii pot edita și vedea statusul actual al rolurilor de pe platformă.

## **c. Meniul utilizator**

### **i. Spațiul de lucru**

Orice utilizator are acces la resursele deja încărcate pe site. Acesta își poate crea propriile colecții sau poate vizualiza colecțiile celorlalți utilizatori.

### **ii. Încărcarea resurselor**

Pentru a încărca o resursă, utilizatorul trebuie să completeze câmpurile obligatorii din formularul de upload. Acesta trebuie să aleagă unul din formatele existente. Pe baza formatului ales, Platforma va selecta convertorul necesar și resursei i se va crea xml-ul corespunzător. Totodată, utilizatorul poate alege dacă resursa va trece prin lanțul de prelucrare TEPROLIN sau nu. În cazul în care utilizatorul a ales această opțiune, output-ul rezultat pe baza call-ului făcut la acest serviciu va fi stocat într-un câmp separat din baza de date.

### **iii. Rapoarte generate**

Utilizatorii pot găsi pe platformă rapoarte și statistici utilizate cu privire la activitatea fiecărui utilizator sau a grupurilor diferite de utilizatori. Aceste rapoarte se încarcă și se actualizează în timp real. Printre caracteristicile enumerate putem găsi: resursă cu cele mai multe vizualizări, resursă cu cele mai multe descărcări, userul care a încărcat cele mai multe resurse etc.

### **iv. Accesul la resursele existente**

Accesul la resursele existente se face din postura rolului de utilizator simplu. După ce credențialele de acces au fost introduse corect, utilizatorul este autorizat să acceseze pagina de resurse. Aici acesta poate *descărca*, *modifica* sau chiar *șterge* o resursă, în funcție de permisiunile asociate lui.

## **3.5 Concluzii**

Livrabilul de față descrie în amănunt arhitectura funcțională și realizarea tehnică a Platformei CoBiLiRo, schema bazei de date care găzduiește resursele încărcate în Platformă, tipurile de activități permise în Platformă, precum și funcționalități, discutate în cadrul întâlnirilor consorțiului proiectului, de avut în vedere pentru extinderi pe viitor.

**Toate obiectivele incluse în plan la această activitate au fost realizate.**

**Anexa 1: Fragmente de cod**

```
CreateMap<CreateResourceViewModelF1, TeiFile>()
  // Header attributes
  .ForPath(m => m.TeiHeader.Title.Ident,
    opt => opt.MapFrom(src => src.Resource.Title))
  .ForPath(m => m.TeiHeader.Collection,
    opt => opt.MapFrom(src => src.Resource.Collection))
  .ForPath(m => m.TeiHeader.Contributor,
    opt => opt.MapFrom(src => src.Resource.ApplicationUser))
  .ForPath(m => m.TeiHeader.Title.LongTitle,
    opt => opt.MapFrom(src => src.Resource.LongTitle))
  .ForPath(m => m.TeiHeader.Description,
    opt => opt.MapFrom(src => src.Resource.Description))
  .ForPath(m => m.TeiHeader.Keywords,
    opt => opt.MapFrom(src => src.Resource.Keywords))
  .ForPath(m => m.TeiHeader.Language,
    opt => opt.MapFrom(src => src.Resource.Language))
  .ForPath(m => m.TeiHeader.Distribution,
    opt => opt.MapFrom(src => src.Resource.Distribution))
  //Data section
  .ForPath(m => m.TeiHeader.DataSection.MetadataCreator,
    opt => opt.MapFrom(src => src.Resource.MetadataCreator))
  .ForPath(m => m.TeiHeader.DataSection.AnnotationCreator,
    opt => opt.MapFrom(src => src.Resource.AnnotationCreator))
  .ForPath(m => m.TeiHeader.DataSection.AnnotationLevel,
    opt => opt.MapFrom(src => src.Resource.AnnotationLevel))
  //Text section
  .ForPath(m => m.TeiHeader.TextSection.TextCreator,
    opt => opt.MapFrom(src => src.Resource.TextCreator))
  .ForPath(m => m.TeiHeader.TextSection.TextFormat,
    opt => opt.MapFrom(src => src.Resource.TextFormat))
  .ForPath(m => m.TeiHeader.Language,
    opt => opt.MapFrom(src => src.Resource.Language))
```