RAPORTARE ŞTIINŢIFICĂ

Proiect complex ReTeRom. Proiect component CoBiLiRo

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

Faza de predare: septembrie 2019
Autori: Dan Cristea, Cristian Pădurariu, Andrei Scutelnicu
Colaborator extern: Şerban Boghiu

1. Rezumatul etapei

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

2. Rezumatul activităţii

Activitatea A2.1 a avut drept obiectiv implementarea specificaţiilor de realizare a infrastructurii funcţionale de upload şi acces la resurse, care urmăreşte îndeaproape descrierea din livrabilul *A1.3 Proiectarea funcţională şi arhitecturală a infrastructurii care va găzdui resursele şi instrumentele de prelucrare şi acces ale consorţiului şi realizarea unui prototip*.

Respectivul livrabil descria: structura de pagini a Platformei şi interfaţa utilizator (secţiunile *Acasă*, *Despre noi*, *Resurse şi servicii de procesare a limbajului natural*, *Comunicări şi publicaţii*, *Parteneri externi* şi *Contact*), funcţionalitatea Platformei CoBiLiRo (rolurile de administrator, curator resurse, donator şi utilizator de resurse; modalităţile de interogare a bazei de date, lansarea în execuţie a unei resurse ca serviciu web, încărcarea şi ştergerea unei resurse, operaţiile de upgradare/updatare a unei resurse, conversiile de format, operaţiile de back-up - salvare periodică a bazei de date a Portalului), funcţionalităţi auxiliare, precum şi propunerea de standard CoBiLiRO, care a ghidat realizarea convertoarelor de format.

3. Descrierea ştiinţifică şi tehnică

3.1 Introducere

Scopul proiectului complex ReTeRom este ridicarea nivelului tehnologiei limbii române în câteva direcţii esenţiale: înţelegerea şi sinteza vorbirii, prelucrările textuale aplicate limbii române, alinierea resurselor voce text româneşti, organizarea unui depozit de date de limbă română pentru cercetare şi uz public. În particular, proiectul component CoBiLiRo va dezvolta un portal capabil să găzduiască un tezaur de resurse audio şi textuale adnotate la diferite niveluri, furnizate iniţial de membrii proiectului, ulterior şi de alţi contribuitori, care va deveni cea mai reprezentativă referinţă de acest tip pentru limba română. Cu precădere, resursele vor fi de natură bimodală, în care cele două componente sunt aliniate la nivel de propoziţie, iar uneori şi mai fin, la cuvânt şi chiar la fonem. Toate resursele sunt însoţite de metadate, fişiere care descriu informaţii despre: sursa obiectului memorat în Platformă, identitatea vorbitorului (doar în anumite condiţii şi cu păstrarea confidenţialităţii), tipul vocii înregistrate (vorbire spontană sau voce în lectură), condiţiile tehnice la înregistrării, durata înregistrării, tipul de fişier rezultat (mp3 sau wav), nivelul de segmentare şi aliniere (propoziţie, cuvinte, foneme), adnotarea ori nu a prozodiei (accente, creşterea sau descreşterea frecvenţei fundamentale), părecum şi alte informaţii.

Componentele vocale şi transcrierile lor provin din diferite surse: eBook-uri însoţite de texte, înregistrări din piese de teatru dublate de textele corespunzătoare, dialoguri TV sau radiio reproduse în volume tipărite, înregistrări din discursuri parlamentare însoţite de transcrierile stenogramelor etc. În puţine 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, intersecţiile comune fiind reţinute, iar restul fiind corectat manual.

3.2 Platforma CoBiLiRo

3.2.1 Aplicaţie

i. Arhitectură (Cristi)

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

Figura 1 prezintă elementele principale ale funcţionalităţii Portalului. Componentele vorbire-text sunt încărcate în platformă de contributori, în general sub formă de perechi de fişiere, iar metadatele sunt completate manual utilizând interfaţa on-line. În funcţie de formatul de intrare este apelat apoi unul dintre convertoare, rezultatul fiind un fişier care respectă formatul standard CoBiLiRo (descris în livrabilul A1.3). Doar asupra componentei textuale, Portalul

apelează apoi lanţul de prelucrări textuale TEPROLIN, care întoarce adnotări complexe. În final, cele patru componente, voce+text+metadate+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.

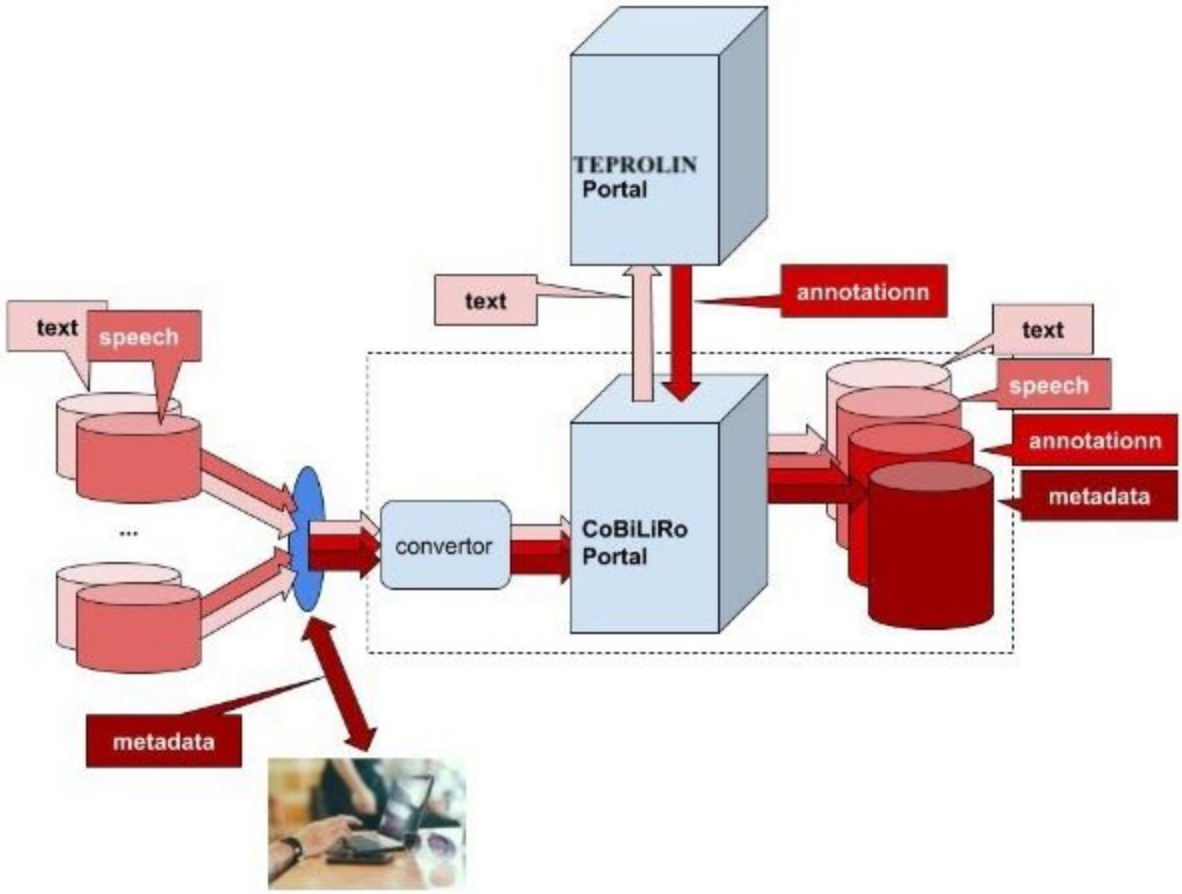

Figura 1: Funcţionarea de principiu a Portalului CoBiLiRo

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 şi încărcată de către modulul de upload. Metadatele specifice resursei sunt stocate în baza de date MySQL, iar fişierele sunt trimise în continuare spre modulul de conversie. În funcţie 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 şi fişierele rezultate în urma conversiei vor fi stocate pe server.

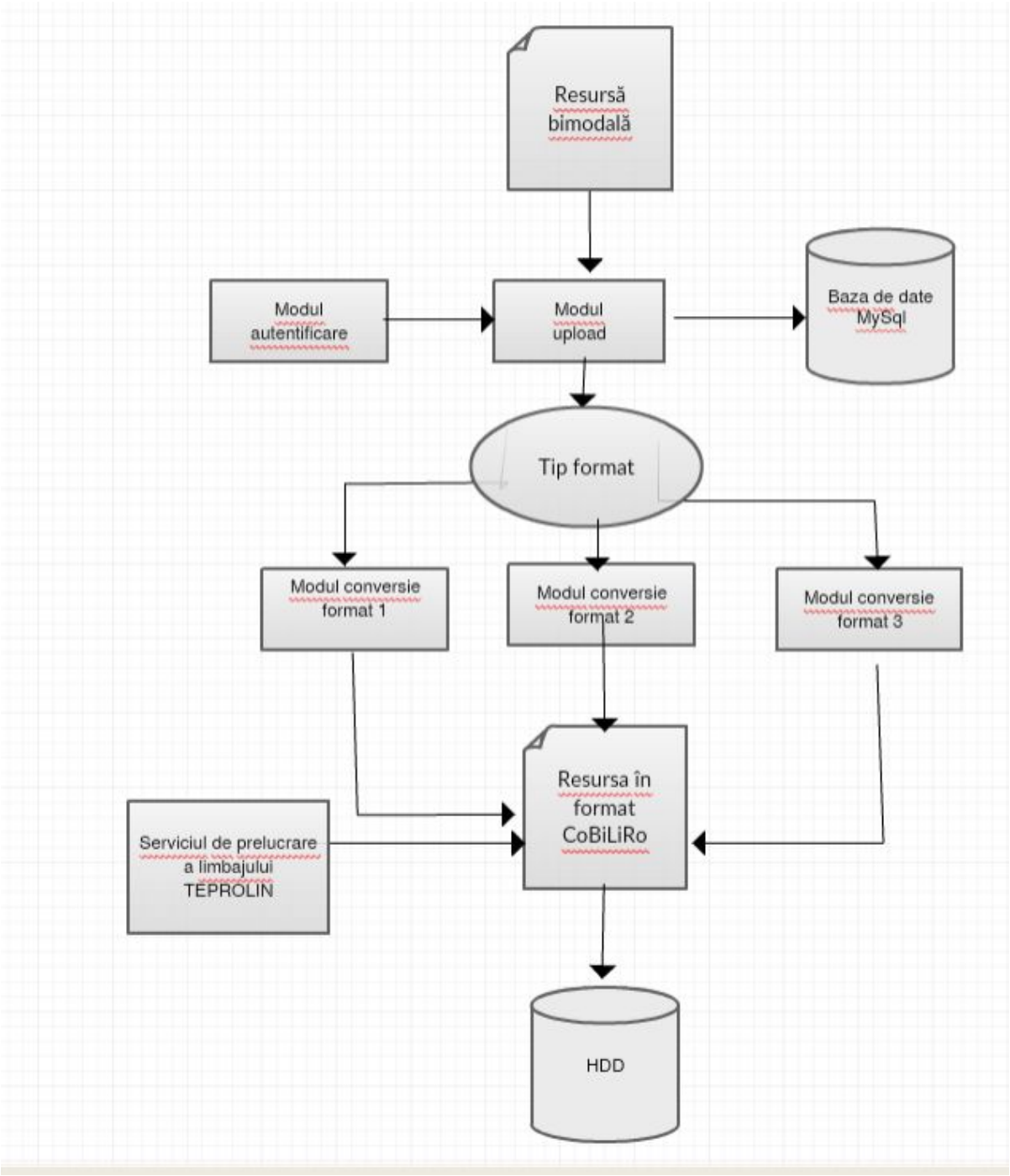

Figura 2: Arhitectura aplicaţiei

3.2.2 Baza de date - schema

Figura 3 detaliază schema bazei de date.

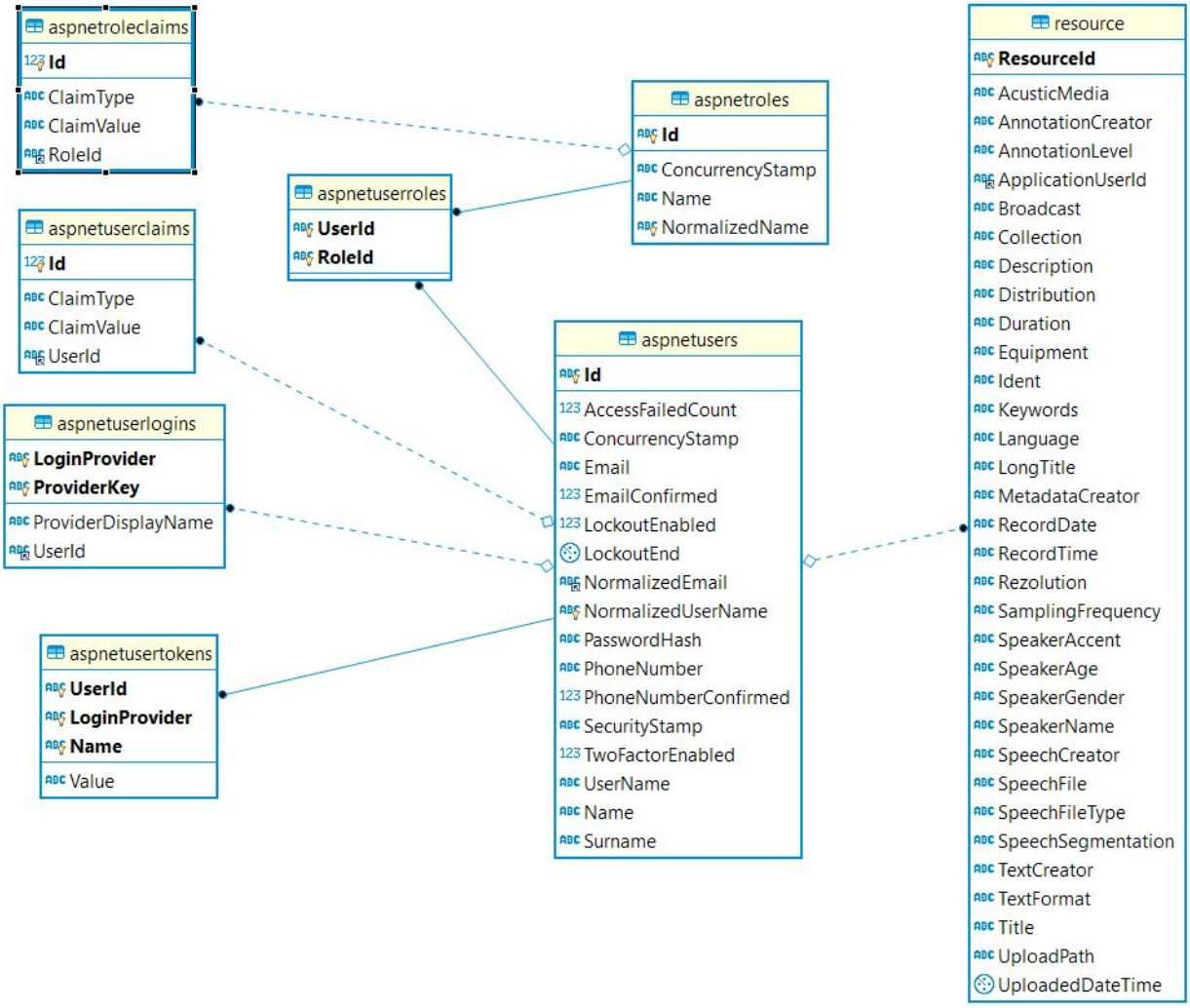

Figura 3: Schema bazei de date

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 şi rolul acestuia în cadrul aplicaţiei este făcută în tabelul *aspnetuserroles.* Permisiunile specifice fiecărui rol sunt stocate în tabelul *aspnetroleclaims.* Tabelul *aspnetuserlogins* reţine informaţii legate de login-uri făcute prin alte metode decât logarea obişnuită de pe site, cum ar fi logări cu Facebook, Google, Twitter etc. (aceasta opţiune nu este încă suportată de platforma CoBiLiRo).

3.2.3 Tehnologii folosite

i. Limbaje şi 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 aplicaţia foloseşte serverul MariaDB, pentru că este unul dintre cele mai populare servere de baze de date din lume. Este realizat de dezvoltatorii originali ai MySQL şi garantat să rămână open source (gratuit). Deoarece serverul pe care rulează aplicaţia funcţionează sub sistemul de operare Linux, acest server virtual pentru baze de date este printre puţinele care poate rula pe Linux. Un alt motiv pentru care s-a ales MariaDB este securitatea informaţiei. Sistemul de securitate al informaţiei este foarte bine pus la punct, atât prin prisma sistemului de operare - Linux, cât şi din configuraţia maşinii virtuale pe care va rula MariaDB.

iii. Servicii pentru procesarea limbajului

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

- Restaurarea diacriticelor
- Transcrierea fonetică a cuvintelor
- Scrierea numeralelor
- Împărţire în propoziţii
- Tokenizare
- POS-tagging
- Lematizare
- Recunoaşterea entităţilor 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 Funcţionalităţi

a. Permisiuni roluri

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

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

b. Modulul de autentificare

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

c. Modulul de încărcare fişiere

Contributorul va accesa pagina de creare a resursei şi va completa formularul asociat, vizibil în Figura 4. Acesta va încărca fişierele asociate resursei, iar în final va acţiona butonul "Crează" ce va permite transmiterea informaţiei către server. Aplicaţia salvează informaţiile din formular într-un obiect (*ResourceViewModel*), iar fişierele resursă într-o listă (*List<IFormFile>*).

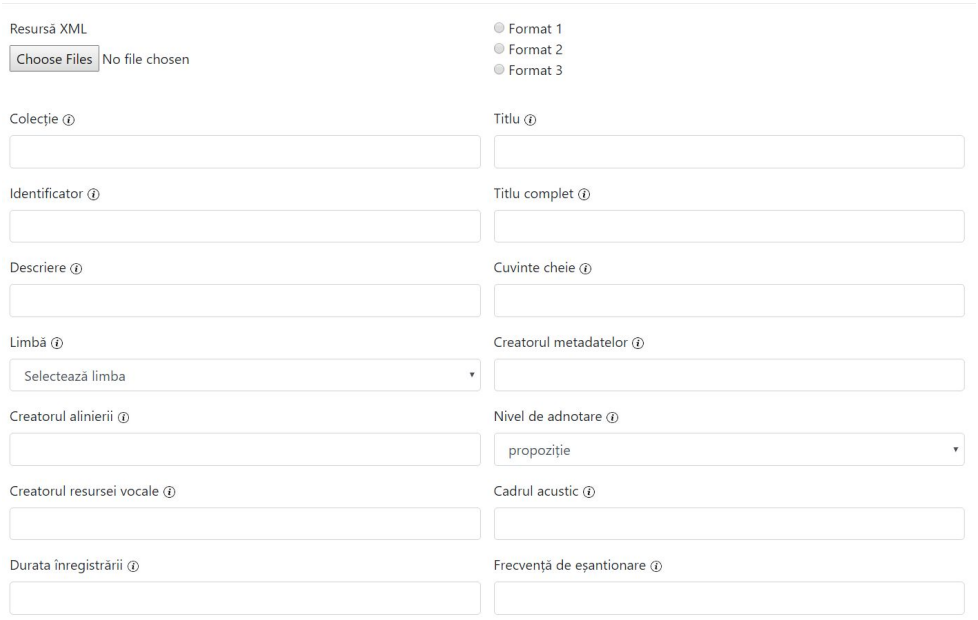

Figura 4: Formularul de descriere a unei resurse

d. Modulul de conversie a fişierelor

Acest modul are ca scop prelucrarea fişierelor şi 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 aparţinând unor clase diferite, pe baza unor reguli definite de programator. Utilizând această opţiune se realizează conversia de la obiectele *ResourceViewModel* şi *List<IFormFile>* la obiecte de tip *Tei*, care conţin un *TeiHeader* şi 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. Funcţionalităţi 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 sa ştie când s-a terminat upload-ul, când a început prelucrarea TEPROLIN şi când s-a terminat salvarea fişierelor pe disk.

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

3.2.5 Manual de utilizare

a. Autentificarea

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

b. Meniul administrativ

i. Adaugă user nou

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

ii. Vizualizare / editare useri

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

iii. Adaugare de noi roluri

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

iv. Vizualizare / editare roluri

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

c. Meniul utilizator

i. Spaţiul de lucru

Orice utilizator are acces la resursele deja încărcate pe site. Acesta îşi poate crea propriile colecţii sau poate vizualiza colecţiile celorlalţi 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 şi resursei i se va crea xml-ul corespunzător. Totodată, utilizatorul poate alege dacă resursa va trece prin lanţul de prelucrare TEPROLIN sau nu. În cazul în care utilizatorul a ales această opţiune, 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 şi statistici utilizate cu privire la activitatea fiecărui utilizator sau a grupurilor diferite de utilizatori. Aceste rapoarte se încarcă şi 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 credenţialele de acces au fost introduse corect, utilizatorul este autorizat să acceseze pagina de resurse. Aici acesta poate *descărca, modifica* sau chiar *şterge* o resursă, în funcţie de permisiunile asociate lui.

3.5 Concluzii

Livrabilul de faţă descrie în amănunt arhitectura funcţională şi realizarea tehnică a Platformei CoBiLiRo, schema bazei de date care găzduieşte resursele încărcate în Platformă, tipurile de activităţi permise în Platformă, precum şi funcţionalităţi, discutate în cadrul întâlnirilor consorţiului 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.Contribuitor,
       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))
```