Пројектни задатак

XML и веб сервиси

Пројектовати, имплементирати и тестирати **информациони систем за рад са захтевима за ауторска и сродна права.**

У оквиру овог информационог система постоје следеће корисничке улоге: **грађанин** и **службеник**.

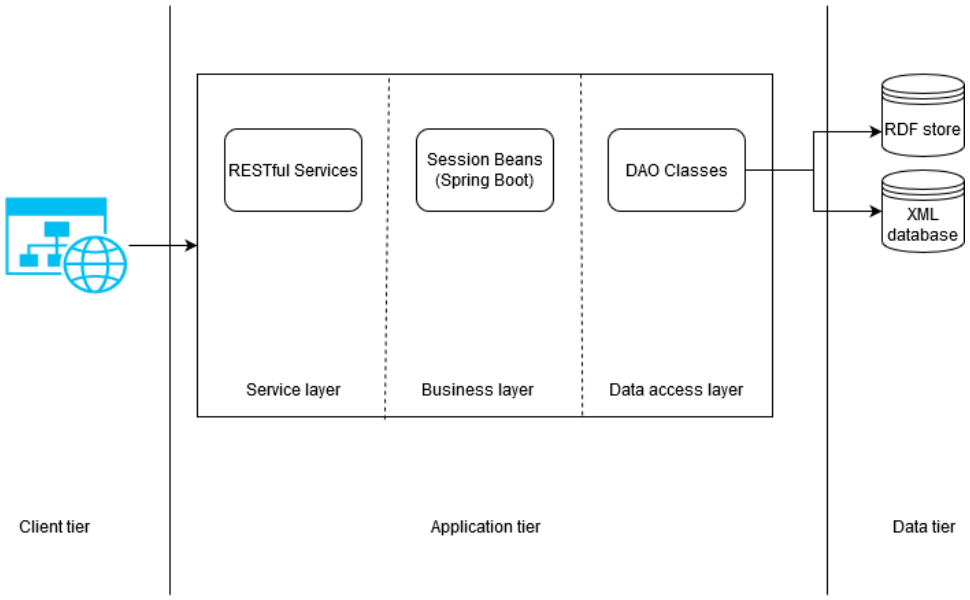
Образац А-1 са којим је потребно радити се налази у прилогу уз пројекат. Такође, у прилогу се налази и закон о ауторским и сродним правима ради додатног информисања.

Информациони систем треба да омогући:

- Регистрација грађанина (службеници су предефинисани и не могу се накнадно додати).
- Пријава корисника (грађана и службеника).
- Попуњавање захтева за ауторска права у виду обрасца А-1 од стране грађанина путем форме на веб страници.
- Службеник може да претражује све поднете захтеве (на чекању), док сви корисници сервиса могу да претражују већ заведена ауторска права (прихваћене захтеве). Претрага би требало да обухвата:
    - Претрагу по текстуалном садржају докумената (тј. основну претрагу) - имплементирати кроз засебну форму на којој је корисницима омогућен унос фраза или кључних речи по којима се врши претрага над свим елементима документа који садрже дате појмове;
    - Претрагу по метаподацима (тј. напредна претрага);
    - Претрагу по више метаподатака реализовати употребом логичких оператора “И”, “ИЛИ” и “НЕ”.
- Службеник може да прегледа захтеве (сваки документ може да се преузме у XHTML и PDF формату).
- Службеник може да преузиме метаподатаке свих докумената у RDF и JSON формату.
- Службеник може да поднесе решење о сваком захтеву. Приликом подношења решења о сваком захтеву од стране службеника, обавештава се грађанин путем мејла у ком се налази копија решења у PDF формату. Решење садржи:
    - Датум одобравања захтева, шифру под којом је ауторско дело заведено, име и презиме службеника и референцу на сам захтев, у случају одобравања захтева;
    - Датум одбијања захтева, образложење, име и презиме службеника и референцу на сам захтев, у случају одбијања захтева.
- Службеник може да генерише **извештај у PDF формату**. Извештај би требао да садржи број поднетих захтева, број прихваћених захтева и број одбијених захтева за одабрани временски период.

Документе, метаподатке докумената и идентификаторе докумената моделовати по W3C стандардима. Документе (обрасце, решења, извештаје итд.) моделовати као XML типове докумената у XML Schema језику (за сваки XML документ дефинисати по XML шему). Метаподатке докумената моделовати у RDF Schema језику. Идентификаторе докумената моделовати као URL шему. Потребно је метаподатке чувати у бази метаподатака. Све остале податке чувати у виду XML докумената у бази XML докумената.

Апликацију пројектовати по трослојној софтверској архитектури, а комуникацију између серверске и клијентске стране реализовати помоћу RESTful веб сервиса. Избор програмског језика и платформе за имплементацију серверске и клијентске стране препуштен је студентима (примери на вежбама су у Јави). Слика 1 (графички приказ на дијаграму) приказује пример архитектуре апликације Портала за подршку при обради захтева за права интелектуалне својине у Java/Spring Boot технологији. Слика 2 приказује архитектуру целокупног система који се састоји од више апликација. Сви подаци који се размењују преко мреже (између клијентске и серверске апликације и различитих сервиса међусобно) **морају бити у XML формату.**



Слика 1 - Вишеслојна архитектура Апликације

Слика 2: Архитектура система

Бекенд апликација Информационог система представљају вишеслојну апликацију које се састоји из слојева приказаних дијаграмом (слика 1). Апликациони слој реализовати кроз три подслоја (*data access*, *business* і *service layer*) на следећи начин:

Data access layer

- генерички слој апликације енкапсулира CRUD (*create*, *retrieve*, *update*, *delete*) операције за приступ подацима у XML и RDF базама података;
- DAO класе имплементирати или као наменске компоненте (npr. SLSB) или као обичне Јава класе (POJO).

Business layer

- средњи слој апликације који имплементира фасаду над слојем за приступ подацима
- Посредује између *data access* и *service layer*-а, стављајући сервисном слоју на располагање неопходан скуп функционалности кроз методе пословне логике;
- *business layer* имплементирати као наменске компоненте.

Service layer

- крајњи слој апликације који дефинише јавно доступан API имплементираних функционалности крајњем кориснику;

- сервисни слој имплементирати као RESTful веб сервисе које референцирају компоненте из *business layer*-а употребом *dependency injection* механизма.