

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені І.І.МЕЧНИКОВА

(повне найменування вищого навчального закладу)

Інститут математики, економіки та механіки

(повне найменування інституту, назва факультету (відділення))

Кафедра математичного забезпечення комп'ютерних систем

(повна назва кафедри (предметної, циклової комісії))

Дипломна робота

бакалавра

(освітньо-кваліфікаційний рівень)

на тему Проектування і розробка інформаційного ресурсу кафедри
на хмарній платформі, підсистема для обліку і зберігання
документації випускаючої кафедри

Designing and development of the department information resource
on the cloud platform.

Graduate department documents accounting and storage subsystem

Проектирование и разработка информационного ресурса кафедры
на облачной платформе, подсистема для учета и хранения
документации выпускающей кафедры

Виконав: студент 4 курсу, групи 1
напряму підготовки (спеціальності)

6.050102 – Комп'ютерна інженерія

(шифр і назва напряму підготовки, спеціальності)

Малахов В.Є.

(прізвище та ініціали)

Керівник

к.т.н., доц. Волощук Л.А.

(прізвище та ініціали)

Рецензент

к.ф.-м.н., доц. Антоненко О.С.

(прізвище та ініціали)

Рекомендовано до захисту:

Протокол засідання кафедри

№ від « » р.

Завідувач кафедри

Захищено на засіданні ЕК №

протокол № від « » р.

Оцінка

(за 4-х бальною шкалою, за шкалою ECTS, бал.)

Голова ЕК

О.О. Арсірій

(підпис)

(прізвище, ініціали)

(підпис)

(прізвище, ініціали)

АНОТАЦІЯ

В даній дипломній роботі розглянуті питання проектування і розробки елементів інформаційного ресурсу кафедри на хмарній платформі, зокрема підсистеми для обліку і зберігання документації випускаючої кафедри, користувачами якої є співробітники кафедри.

З метою проектування і розробки зазначеної системи на основі аналізу предметної області побудована модель даних, в якій виділено 36 типів документів відповідно з чотирма основними типами процесів діловодства кафедри.

Проведено проектування сервіс-орієнтованої архітектури інформаційного ресурсу кафедри, при реалізації якої інтегровані сервіси хмарної файлової системи і системи редагування текстових документів, електронних таблиць і презентацій.

Створена система дозволяє автоматизувати процес документообігу кафедри, суттєво полегшує формування та аналіз документації організаційного, учбового і наукового процесів діловодства кафедри.

Інформаційний ресурс кафедри розміщений за адресою 91.208.138.238 мережі інтернет, що дозволяє співробітникам кафедри отримувати територіально-розподілений доступ до роботи з документацією кафедри.

За підсумками дипломної роботи опубліковано 2 наукові статті та 2 тез конференцій, а також складений акт про використання результатів дипломної роботи в організаційному управлінні кафедрою.

АННОТАЦИЯ

В данной дипломной работе рассмотрены вопросы проектирования и разработки элементов информационного ресурса кафедры на облачной платформе, в частности подсистемы учёта и хранения документации выпускающей кафедры, пользователями которой являются сотрудники кафедры.

Для проведения проектирования и разработки указанной системы на основе анализа предметной области построена модель данных, в которой выделены 36 типов документов в соответствии с четырьмя основными типами процессов делопроизводства кафедры.

Проведено проектирование сервис-ориентированной архитектуры информационного ресурса кафедры, при реализации которой интегрированы сервисы облачной файловой системы и системы редактирования текстовых документов, электронных таблиц и презентаций.

Созданная система позволяет автоматизировать процесс документооборота кафедры, существенно облегчает формирование и анализ документации организационного, учебного и научного процессов деятельности кафедры.

Информационный ресурс кафедры размещён по адресу 91.208.138.238 сети интернет, что позволяет сотрудникам кафедры получать территориально-распределённый доступ к работе с документацией кафедры.

По результатам дипломной работы опубликовано 2 научных статьи и 2 тезисов конференций, а также составлен акт об использовании результатов дипломной работы в организационном управлении кафедрой.

ABSTRACT

In this graduation work the issues of designing and development of the department information resource on the cloud platform and graduate department documents accounting and storage subsystem, whose users are employees of the department, are considered.

For the purpose of designing and development of this system, based on the analysis of the subject area, a data model has been constructed, in which 36 types of documents have been identified in accordance with the four main types of departmental records management processes.

A service-oriented architecture of the information resource of the department was designed. When implementing the architecture, the services of the cloud file system and the system for editing text documents, spreadsheets and presentations were integrated.

The created system allows to automate the process of document circulation of the department, significantly facilitates the formation and analysis of documentation of the organizational, educational and scientific processes of the department.

Information resource of the department is located at the address 91.208.138.238 of the Internet, which allows the staff of the department to obtain territorially-distributed access to work with the department's documentation.

According to the results of the graduation work, 2 scientific articles and 2 theses of conferences are published, and also an act on the use of the graduation work results in the department organizational management has been drawn up.

ЗМІСТ

	ст.
ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД АНАЛОГІЧНИХ СИСТЕМ.....	10
1.1 Інтегрована система керування «ІСУ ВУЗ»	10
1.2 Автоматизована система керування ВНЗ «Паллада».....	12
1.3 Рішення «Галактика Управління ВУЗом»	13
1.4 Порівняльний аналіз розглянутих аналогічних систем	14
2 ПРОЕКТУВАННЯ ПІДСИСТЕМИ ОБЛІКУ І ЗБЕРІГАННЯ ДОКУМЕНТАЦІЇ ВИПУСКАЮЧОЇ КАФЕДРИ ІНФОРМАЦІЙНОГО РЕСУРСУ КАФЕДРИ НА ХМАРНІЙ ПЛАТФОРМІ	16
2.1 Проектування архітектури хмарної інформаційної системи «Інформаційний ресурс кафедри»	16
2.2 Формування функціональних задач по ролях користувачів	18
2.3 Модель даних інформаційної системи	19
2.4 Проектування підсистеми обліку і зберігання документації	23
2.4.1 Побудова логічної схеми бази даних	23
2.4.2 Модуль формування атрибутів документа	27
2.5 Проектування клієнтського інтерфейсу	28
3 РОЗРОБКА ПІДСИСТЕМИ ОБЛІКУ І ЗБЕРІГАННЯ ДОКУМЕНТАЦІЇ ВИПУСКАЮЧОЇ КАФЕДРИ ІНФОРМАЦІЙНОГО РЕСУРСУ КАФЕДРИ НА ХМАРНІЙ ПЛАТФОРМІ	30
3.1 Вибір технологій реалізації та системної архітектури.....	30
3.2 Вибір СУБД Postgresql і фреймворка взаємодії.....	31
3.3 Використовувані сторонні сервіси.....	32
3.3.1 Хмарний файловий сервіс «ownCloud»	32
3.3.2 Хмарний сервіс редагування документів «OnlyOffice».....	35
3.4 Реалізація фізичної схеми БД документів	36
3.5 Реалізація системної архітектури додатку	36

3.5.1 Реалізація рівня Model.....	37
3.5.2 Реалізація рівня View.....	37
3.5.3 Реалізація рівня Controller.....	38
3.6 Модуль формування атрибутів документа з файлу	40
3.7 Посібник користувача	41
ВИСНОВОК.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	53
ДОДАТОК А ЗАПИТИ СТВОРЕННЯ ТРИГЕРІВ	55
ДОДАТОК Б ЗАПИТИ НА СТВОРЕННЯ ТАБЛИЦЬ ТА ВИДАЧІ ПРИВІЛЕЇВ	57
ДОДАТОК В КОД РЕАЛІЗАЦІЇ РІВНЯ MODEL	59
ДОДАТОК Г КОД РЕАЛІЗАЦІЇ РІВНЯ CONTROLLER	62
ДОДАТОК Д АКТ ВПРОВАДЖЕННЯ	67

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення:

ІС — інформаційна система;

НОЗ ВПО — недержавний освітній заклад вищої професійної освіти;

ERP — Enterprise Resource Planning;

ECM — Enterprise content management;

SOA — Service-Oriented Architecture;

MVC — Model-View-Controller;

JDBC — Java Database Connectivity;

SaaS — Software as a Service.

ВСТУП

У цей час автоматизована обробка інформації та необхідність у доступі до інформації, яка оброблюється, для користувачів незалежно від їхнього місця розташування здобуває вкрай важливе значення. Це обумовлене вимогою до актуальності одержуваних даних і своєчасному реагуванню на повідомлення для найбільш швидкого прийняття рішень. Це стосується, у тому числі, такої предметної області, як адміністративне управління навчальним процесом і документообігом випускаючої кафедри [1].

Одним із способів підвищення ефективності управління навчальним процесом і документообігом є впровадження спеціалізованої інформаційної системи, яка буде націлена не тільки на розв'язання базових завдань управління в освітніх процесах, але і дозволить автоматизовано підтримувати повний життєвий цикл управління документами.

Інформаційна система — це програмний комплекс, функції якого складаються з підтримки надійного зберігання інформації в пам'яті комп'ютера, виконання специфічних для даного додатка перетворень інформації та обчислень, надання для користувача зручного та легко освоюваного інтерфейсу. Звичайно обсяги інформації, з якими доводиться мати справа таким системам, досить великі, а сама інформація має досить складну структуру.

Однак реалізація ІС на хмарній платформі і перенесення усіх обчислень на бік сервера дозволить забезпечити захист і розподілений доступ до ІС.

Метою даної дипломної роботи є проектування та розробка елементів інформаційного ресурсу кафедри на хмарній платформі і підсистеми для обліку і зберігання документації випускаючої кафедри.

Для досягнення наведеної мети необхідно розв'язати наступні задачі:

- 1) виконати аналіз предметної області, пов'язаної зі створенням, зберіганням і веденням документів кафедри;

- 2) виконати аналіз аналогічних систем з метою визначення актуальних вимог і завдань проектування хмарного інформаційного ресурсу кафедри, підсистеми для обліку і зберігання документації випускаючої кафедри;
- 3) виконати проектування архітектури хмарного інформаційного ресурсу;
- 4) визначити вимоги до даних і їх структурі, функціональні вимоги типових користувачів інформаційного ресурсу кафедри;
- 5) забезпечити розмежування доступу з боку різних категорій користувачів і захист від несанкціонованого доступу до документів;
- 6) провести проектування БД підсистеми для обліку і зберігання документації випускаючої кафедри;
- 7) вибрати технології реалізації та здійснити системну інтеграцію хмарних сервісів і операційної платформи для реалізації інформаційного ресурсу кафедри;
- 8) реалізувати підсистему для обліку і зберігання документації випускаючої кафедри, яка, крім перегляду, надасть можливості зручного зберігання і редагування документів;
- 9) провести розміщення і тестування ресурсу на хмарному сервері.

ВИСНОВОК

У даній дипломній роботі спроектовані і реалізовані елементи інформаційного ресурсу кафедри на хмарній платформі і підсистема обліку і зберігання документації випускаючої кафедри.

Основною метою є створення системи, що дозволяє істотно полегшити формування та аналіз документації різних процесів діяльності кафедри, а також реалізувати можливість комфортного зберігання і редагування документів.

В результаті проведеного аналізу існуючих систем управління кафедрою і систем зберігання документації отримані рекомендації до розроблюваної системи. Прийнято рішення реалізувати можливість редагування документів, а для проведення аналітики ввести розширене уявлення документа за допомогою метаданих.

З метою проектування і розробки зазначеної системи на основі аналізу предметної області побудована модель даних, в якій виділено 36 типів документів відповідно з чотирма основними типами процесів діловодства кафедри.

При проектуванні обрана клієнт-серверна сервіс-орієнтована архітектура, в рамках якої, в систему інтегровані і налаштовані два хмарних сервісу:

- сервіс файлової системи ownCloud;
- сервіс спільного редагування файлів OnlyOffice.

Для реалізації системи обрані технології фреймворка Spring, включаючи патерн MVC.

В рамках реалізації підсистеми обліку і зберігання документації виконана інтеграція таблиць підсистеми в БД сервісу ownCloud і реалізовані безпосередні зв'язки з чотирма таблицями даного сервісу.

Для даної підсистеми розроблені модулі авторизації і формування атрибутів документа.

У розробленій підсистемі обліку і зберігання документації були виконані всі поставлені до неї вимоги і реалізовані основні механізми роботи з документами. Дана підсистема має перспективи для подальшої доробки та розширення функціональних можливостей, у тому числі формування атрибутів документів автоматично з файлу документа і використання отриманих атрибутів для формування модуля аналітики.

За результатами даної роботи опубліковано дві статті і представлені доповіді на всеукраїнській конференції, а також складений акт про використання результатів дипломної роботи в організаційному управлінні кафедрою математичного забезпечення комп'ютерних систем ОНУ імені І.І. Мечникова (Додаток Д).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Малахов В.Е., Анализ систем управления высшими учебными заведениями / Малахов В.Е, Драбинка В.В, Волощук Л.А // Тезисы докладов 14 Всеукраинской конференции студентов и молодых учёных «Информатика, информационные системы и технологии». – Одесса. – 2017. – С.167-168.
2. Интегрированная система управления ВУЗом. [Электронный ресурс] – Режим доступа: http://isuhelp.tisbi.ru/index.php/Заглавная_страница – 23.07.2014.
3. Эмилова О.А., Структура информационной системы «Наука» в АСУ «Паллада» / Эмилова О.А., Свиридова А.С., Сержантова М.В. // Актуальные проблемы авиации и космонавтики. 2015. №11.
4. Решение «Галактика Управление Вузом». [Электронный ресурс] – Режим доступа: <https://www.galaktika.ru/vuz/> – 24.03.2012.
5. Мамду Ибрахим. Сервис-ориентированная архитектура и архитектура предприятия. [Электронный ресурс] – Режим доступа: <https://www.ibm.com/developerworks/ru/library/ws-soa-enterprise1/index.html/> – 25.03.2008.
6. Драбинка В.В., Модель данных информационного ресурса кафедры / Драбинка В.В, Малахов В.Е, Волощук Л.А // Тезисы докладов 14 Всеукраинской конференции студентов и молодых учёных «Информатика, информационные системы и технологии». – Одесса. – 2017. – С.151-152.
7. Малахов Е.В. Организация баз данных: конспект. – О.: ОНУ, 2014. – 169с.
8. Уваров А.Н. Инверсия управления и внедрение зависимостей // Символ науки. – 2016. – №10-1 – С.28-32.
9. Spring MVC — основные понятия, архитектура. [Электронный ресурс] – Режим доступа: <http://javastudy.ru/spring-mvc/spring-mvc-basic/> – 9.12.2016.

10. Гунченко Ю.О., Концепція побудови комплексу програмного забезпечення для дослідження завадостійких кодів. / Гунченко Ю.О., Ємельянов П.С. Щербакова Т.О., Малахов В.Е. // Збірник наукових праць військово-го інституту КНУ імені Тараса Шевченка. – Київ, 2017. – Вып. 55. – С. 125-134.
11. Jon Ellis, Linda Ho. JDBC 3.0 Specification // Символ науки. – 2001 – С.190.
12. ownCloud Documentation Overview. [Електронний ресурс] – Режим доступу: <https://doc.owncloud.org/> – 17.05.2017.
13. Maria G. Glava, Information systems reengineering approach based on the model of information systems domains. / Maria G. Glava, **Valery E. Malakhov** // International Journal of Software Engineering & Computer System. – Kuantan (Malaysia), 2017.
14. Установка версії Сервера документів для Linux. [Електронний ресурс] – Режим доступу: <http://helpcenter.onlyoffice.com/ru/server/linux/document/linux-installation.aspx> – 31.05.2017.

ДОДАТОК А

ЗАПИТИ СТВОРЕННЯ ТРИГЕРІВ

```
-- триггер додавання файлу в БД
CREATE OR REPLACE FUNCTION "public".add_filecache_func()
RETURNS TRIGGER AS $$
DECLARE
    testId int;
    testName varchar;
    testWay varchar;

BEGIN

    testID = NEW.fileid;
    testName = NEW.name;
    testWay = NEW.path;

--    RAISE NOTICE 'Get ID (%)', testID;
--    RAISE NOTICE 'Get title (%)', testName;

    IF((testName like '%.doc'
        or testName like '%.docx'
        or testName like '%.xls'
        or testName like '%.xlsx'
        or testName like '%.pdf')
        and testWay like '%files/Documents%')
    THEN
        INSERT INTO "document"."document" (doc_id, doc_title,
doc_type_id, doc_attributes) VALUES
            (testID, testName, 1, '{}');
        INSERT INTO "document"."doc_proc"(doc_id, proc_id)
VALUES
            (testID, 1);
    END IF;

RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER add_filecache
BEFORE INSERT ON "public"."oc_filecache" FOR EACH ROW
EXECUTE PROCEDURE "public".add_filecache_func()
;

ALTER FUNCTION "public".add_filecache_func()
OWNER TO oc_root;
```

```

-- триггер удаления файлов из БД
CREATE OR REPLACE FUNCTION "public".delete_filecache_func()
RETURNS TRIGGER AS $$
DECLARE
    testID int;
    testName varchar;
    testWay varchar;

BEGIN

    testID = OLD.fileid;
    testName = OLD.name;
    testWay = OLD.path;

--     RAISE NOTICE 'Get ID (%)', testID;
--     RAISE NOTICE 'Get title (%)', testName;

    IF((testName like '%.doc'
        or testName like '%.docx'
        or testName like '%.xls'
        or testName like '%.xlsx'
        or testName like '%.pdf')
        and testWay like '%files/Documents%')
    THEN
        DELETE FROM "document"."doc_author" WHERE doc_id = testID;
        DELETE FROM "document"."doc_proc" WHERE doc_id =
testID;
        DELETE FROM "document"."document" WHERE doc_id =
testID;

        END IF;

RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER delete_filecache
BEFORE DELETE ON "public"."oc_filecache" FOR EACH ROW
EXECUTE PROCEDURE "public".delete_filecache_func()
;

ALTER FUNCTION "public".delete_filecache_func()
OWNER TO oc_root;

```

ДОДАТОК Б

ЗАПИТИ НА СТВОРЕННЯ ТАБЛИЦЬ ТА ВИДАЧІ ПРИВІЛЕЇВ

```

-- создание таблицы типов документов
create table "document"."type"
(
    type_id serial not null primary key,
    type_name varchar not null
);
-- создание таблицы документов
create table "document"."document"
(
    doc_id int not null primary key,
    doc_title varchar null,
    doc_type_id int null references "document"."type"(type_id),
    doc_date int null,
    doc_attributes jsonb null,
    doc_description varchar null
);
-- создание таблицы процессов документов
create table "document"."proc"
(
    proc_id serial not null primary key,
    proc_name varchar not null,
    proc_default_way varchar not null
    -- /var/www/owncloud/data/[main/root/...]/files/
);
-- создание таблицы авторов документов
create table "document"."author"
(
    author_id serial not null primary key,
    author_name varchar not null
);

create table "document"."doc_proc"
(
    doc_id int not null references
"document"."document"(doc_id),
    proc_id int not null references "document"."proc"(proc_id),
    primary key (doc_id, proc_id)
);

create table "document"."doc_author"
(
    doc_id int not null references
"document"."document"(doc_id),
    author_id int not null references
"document"."author"(author_id),
    primary key (doc_id, author_id)

```

```
);  
-- передача всех прав на таблицы системной роли oc_root и  
владельцу БД – ownCloud  
ALTER TABLE "document"."type" OWNER TO oc_root;  
ALTER TABLE "document"."document" OWNER TO oc_root;  
ALTER TABLE "document"."proc" OWNER TO oc_root;  
ALTER TABLE "document"."author" OWNER TO oc_root;  
ALTER TABLE "document"."doc_proc" OWNER TO oc_root;  
ALTER TABLE "document"."doc_author" OWNER TO oc_root;  
  
GRANT ALL ON ALL TABLES IN SCHEMA public TO oc_root;  
GRANT ALL ON ALL TABLES IN SCHEMA public TO owncloud;  
  
GRANT ALL ON ALL TABLES IN SCHEMA document TO oc_root;  
GRANT ALL ON ALL TABLES IN SCHEMA document TO owncloud;
```

ДОДАТОК В

КОД РЕАЛІЗАЦІЇ РІВНЯ MODEL

```
// Объект содержащий информацию об авторе

public class EntityDocumentAuthor {

    private Integer author_id;
    private String author_name;

    public EntityDocumentAuthor(Integer author_id, String
author_name) {
        this.author_id = author_id;
        this.author_name = author_name;
    }

    public EntityDocumentAuthor() {
    }

    public Integer getAuthor_id() {
        return author_id;
    }

    public void setAuthor_id(Integer author_id) {
        this.author_id = author_id;
    }

    public String getAuthor_name() {
        return author_name;
    }

    public void setAuthor_name(String author_name) {
        this.author_name = author_name;
    }
}
```



```

private static final String FIND_ONLY_DOC_TABLE =
    "SELECT doc_id, doc_title, doc_type_id, doc_date,
doc_attributes, doc_description\n" +
    "\tFROM \"document\".\"document\"\n" +
    "WHERE doc_id = ?;";

//region Document update
private static final String UPD_DOCUMENT_WITH_ATTRIBUTES =
    "UPDATE \"document\".\"document\"\n" +
    "\tSET doc_attributes = ?::JSONB\n" +
    "WHERE doc_id = ?;";

private static final String UPD_DOCUMENT_WITHOUT_ATTRIBUTES
=
    "UPDATE \"document\".\"document\"\n" +
    "\tSET doc_title = ?, doc_type_id = ?,
doc_date = ?, doc_description = ?\n" +
    "WHERE doc_id = ?;";

//endregion
//...

```

ДОДАТОК Г

КОД РЕАЛІЗАЦІЇ РІВНЯ CONTROLLER

```

@Controller
@RequestMapping("/document")
public class ControllerDocument {

    // Инициализация логера
    private static final Logger log =
LoggerFactory.getLogger(ControllerUpload.class);

    la3v.logic.entities.document.EntityDocument
entityDocumentBase;

    private ApplicationContext context;
    private final String regexAuthor =
"(\p{L}+?\s\p{L}\.\s??\p{L}\.)([\\;]??";

    private Map<String, IAttributes> attributeMap = new
AttributesMap().getAttributeMap();
    private Map<String, String> attributeRusEng = new
AttributesRusEng().getAttributesRusEng();

    @Autowired
    public void context(ApplicationContext context) {
        this.context = context;
    }

    @RequestMapping(value = "/all", method = RequestMethod.GET)
    public String showDocumentList(Model model, Principal
principal) {
        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);

        List<EntityDocument> entityDocumentList =
repositoryDocument.getAllDocumentList(principal.getName());
        model.addAttribute("entityDocumentList",
entityDocumentList);

        return "doc/document";
    }

    @RequestMapping(value = "/view/{id}", method =
RequestMethod.GET)
    public String showViewDocumentForm(Model model,
@PathVariable("id") Integer docId) throws Exception {

```

```

        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        la3v.logic.entities.document.EntityDocument
entityDocument = repositoryDocument.findById(docId);

        JSONObject jsonObj = entityDocument.getAttributes();

        if (entityDocument == null)
            throw new Exception("Database error! There is no
document with such ID!");

        if
(attributeMap.get(entityDocument.getDocType().toLowerCase()) !=
null) {
            IAttributes attributes =
attributeMap.get(entityDocument.getDocType().toLowerCase()).getF
romJson(jsonObj);
            model.addAttribute("entityDocumentAttribute",
attributes.toListString());
        }

        model.addAttribute("entityDocument", entityDocument);
        return "doc/attributes";
    }

    @RequestMapping(value = "/view/{id}", method =
RequestMethod.POST)
    public String nextViewDocumentForm(Model model,
@PathVariable("id") Integer docId) throws Exception {

        return "redirect:/document/edit/" + docId;
    }

    @RequestMapping(value = "/edit/{id}", method =
RequestMethod.GET)
    public String showEditDocumentForm(Model model,
@PathVariable("id") Integer docId) throws Exception {

        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        la3v.logic.entities.document.EntityDocument
entityDocument = repositoryDocument.findById(docId);
        List<EntityDocumentType> entityDocumentTypeList =
repositoryDocument.getAllDocumentType();
        List<EntityDocumentProc> entityDocumentProcList =
repositoryDocument.getAllDocumentProc();

        model.addAttribute("entityDocument", entityDocument);
        model.addAttribute("entityDocumentTypeList",
entityDocumentTypeList);
        model.addAttribute("entityDocumentProcList",
entityDocumentProcList);
    }

```

```

        return "doc/editBase";
    }

    @RequestMapping(value = "/edit/{id}", method =
RequestMethod.POST)
    public String
nextEditDocumentForm(@ModelAttribute("entityDocument")
la3v.logic.entities.document.EntityDocument entityDocument,
Model model, BindingResult result, Principal principal,
@PathVariable("id") Integer docId) throws Exception {

        entityDocument.setDocId(docId);
        entityDocumentBase = entityDocument;

        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        la3v.logic.entities.document.EntityDocument
entityDocumentOld = repositoryDocument.findById(docId);

        List<String> authors = new ArrayList<String>();

        EntityDocumentAuthor entityDocumentAuthor;
        EntityDocumentProc entityDocumentProc =
repositoryDocument.getDocumentProcById(Integer.parseInt(entityDo
cument.getDocProcess()));
        EntityDocumentType entityDocumentType =
repositoryDocument.getDocumentTypeById(Integer.parseInt(entityDo
cument.getDocType()));

        // обновляем документ

repositoryDocument.setUpdDocumentWithoutAttributes(entityDocumen
t);

        if (!Objects.equals(entityDocumentOld.getDocProcess(),
entityDocumentProc.getProcName()))
            // обновляем тип процесса

repositoryDocument.setUpdDocumentProc(entityDocument);

        if (!Objects.equals(entityDocumentOld.getAuthorName(),
entityDocument.getAuthorName())) {

repositoryDocument.deleteFromAuthor(entityDocument.getDocId());

            //region Получаем список авторов из строки
            Pattern pattern = Pattern.compile(regexAuthor);
            Matcher matcher =
pattern.matcher(entityDocument.getAuthorName());
            while (matcher.find()) {

```

```

        for (int i = 1; i <= matcher.groupCount(); i++)
    {
        authors.add(matcher.group(i));
    }
    //endregion

    //region Добавляем авторов и связи
    for (String str : authors) {
        entityDocumentAuthor =
repositoryDocument.getDocumentAuthorByName(str);

        if (entityDocumentAuthor != null) {

repositoryDocument.insertNewDocumentDocAuthor(entityDocument.get
DocId(), entityDocumentAuthor.getAuthor_id());
        } else {

repositoryDocument.insertNewDocumentAuthor(str);
        entityDocumentAuthor =
repositoryDocument.getDocumentAuthorByName(str);

repositoryDocument.insertNewDocumentDocAuthor(entityDocument.get
DocId(), entityDocumentAuthor.getAuthor_id());
        }
    }
    //endregion
}

    JsonObject jsonObj = entityDocument.getAttributes();
    if
(attributeMap.get(entityDocumentType.getType_name().toLowerCase(
)) != null) {

        String docType =
attributeRusEng.get(entityDocumentType.getType_name().toLowerCase(
e()));

        if (docType == null)
            throw new Exception("There is no such attribute
in attribute map!");

        log.info(String.format("Get docType: %s", docType));

        return "redirect:{id}/attributes/" + docType;
//
        return "redirect:" + entityDocument.getDocId() +
"/attributes/" + docType;
    }

    return "redirect:/document/all";
}

```

```

    @RequestMapping(value = "/edit/{id}/attributes/{docType}",
method = RequestMethod.GET)
    public String showListAttributes(Model model,
@PathVariable("id") Integer docId, @PathVariable("docType")
String docType) throws Exception {

        //IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        //la3v.logic.entities.document.EntityDocument
entityDocument = repositoryDocument.findById(docId);

        IAttributes attributes =
attributeMap.get(docType.toLowerCase());

        if (attributes == null)
            throw new Exception("There is no such attribute in
attribute map!");

        model.addAttribute("attributes", attributes);
        model.addAttribute("entityDocumentBase",
entityDocumentBase);
        model.addAttribute("addressPart", docId);

        return "doc/editViews/" + docType.toLowerCase();
    }

    private void editDocument(IAttributes attributes, String
documentId, Principal principal) {
        log.info(String.format("documentId: %s", documentId));
        Gson gson = new Gson();
        JsonParser parser = new JsonParser();
        JsonObject obj =
parser.parse(gson.toJson(attributes)).getAsJsonObject();

        entityDocumentBase.setAttributes(obj);

        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        repositoryDocument.setUpdDocumentWithAttributes
(entityDocumentBase);
    }

    @ExceptionHandler(Exception.class)
    public ModelAndView handleAllException(Exception ex) {
        ModelAndView model = new
ModelAndView("err/pageControllerError");

        model.addObject("errMsg", ex.getMessage());
        return model;}

```

ДОДАТОК Д
АКТ ВПРОВАДЖЕННЯ