

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 архівації робіт,
публікації співробітників кафедри і студентів

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

Staff's and students' works and publications archiving system

Проектирование и разработка информационного ресурса кафедры
на облачной платформе, подсистема архивации работ,
публикаций сотрудников кафедры и студентов

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

6.050102 – Комп'ютерна інженерія

(шифр і назва напряму підготовки, спеціальності)

Драбинка В.В.

(прізвище та ініціали)

Керівник

к.т.н., доц. Волощук Л.А.

(прізвище та ініціали)

Рецензент

к.ф.-м.н., доц. Крапивний Ю.М.

(прізвище та ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
№ від « » р.

Завідувач кафедри

Є.В. Малахов

(підпис)

(прізвище, ініціали)

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)

Одеса – 2017

АНОТАЦІЯ

У дипломній роботі проведено проектування і розробка підсистеми архівації даних інформаційного ресурсу кафедри.

Для формування вимог до інформаційного ресурсу розглянуті актуальні системи документообігу та автоматизації діяльності навчальних закладів, а також деякі аспекти питань архівування документації.

Підсистема архівування дозволяє виконувати різні дії з архівної документації, в тому числі такі як протоколювання процесів архівації та видалення з архіву, розширений пошук документів архіву за типами документів, за належністю до типових процесів діяльності кафедри, а також загальний пошук.

Інформаційний ресурс кафедри реалізований на основі клієнт-серверної сервіс-орієнтованої архітектури, може бути розміщений в будь-якому хмарному сховищі з можливістю територіально-розподіленого доступу співробітників кафедри відповідно до політики доступу до ресурсу.

ABSTRACT

In the graduation work, the design and development of a subsystem for archiving data from the information resource of the department was carried out.

To form the requirements for the information resource, the current systems of document circulation and automation of the activities of educational institutions, are considered as well as some aspects of the archiving of documentation.

The archiving subsystem allows you to perform various actions with archive documentation, including such as logging of archiving and deletion processes from the archive, advanced search of archive documents by document types, by membership in the standard processes of the department, by years, by authors.

Information resource of the department is realized on the basis of client-server service-oriented architecture, can be placed in any cloud storage with the possibility of geographically distributed access of the department employees in accordance with the policy of access to the resource.

АННОТАЦИЯ

В дипломной работе проведено проектирование и разработка подсистемы архивации данных информационного ресурса кафедры.

Для формирования требований к информационному ресурсу рассмотрены актуальные системы документооборота и автоматизации деятельности учебных заведений, а также некоторые аспекты вопросов архивирования документации.

Подсистема архивирования позволяет выполнять различные действия с архивной документацией, в том числе такие как протоколирование процессов архивации и удаления из архива, расширенный поиск документов архива по типам документов, по принадлежности к типовым процессам деятельности кафедры, по годам, по авторам.

Информационный ресурс кафедры реализован на основе клиент-серверной сервис-ориентированной архитектуры, может быть размещён в любом облачном хранилище с возможностью территориально-распределенного доступа сотрудников кафедры в соответствии с политикой доступа к ресурсу.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 ОГЛЯД АНАЛОГІЧНИХ СИСТЕМ І ПОСТАНОВКА ЗАВДАННЯ	10
1.1 Принципи архівації документації в сучасних інформаційних системах 10	
1.2 Загальна структура архіву документообігу	11
1.3 Процес перенесення документа в архів.....	14
1.4 Огляд існуючих систем документообігу.....	16
1.4.1 Система управління навчальним процесом «Магеллан».....	16
1.4.2 Система управління документообігом Directum RX.....	18
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОГО РЕСУРСУ КАФЕДРИ НА ХМАРНІЙ ПЛАТФОРМІ І ПІДСИСТЕМИ АРХІВАЦІЇ РОБІТ, ПУБЛІКАЦІЙ СПІВРОБІТНИКІВ КАФЕДРИ І СТУДЕНТІВ.....	21
2.1 Проектування архітектури інформаційного ресурсу кафедри.....	21
2.2 Формування моделі даних інформаційного ресурсу кафедри.....	24
2.2.1 Класифікація даних кафедрального документообігу	24
2.2.2 Проектування моделі даних кафедрального документообігу	25
2.3 Проектування бази даних підсистеми архівації	28
2.3.1 Побудова логічної схеми бази даних підсистеми архівації	28
2.4 Модуль конвертації даних з підсистеми документообігу в підсистему архівації	31
2.5 Функціональні вимоги до підсистеми архівації	36
2.6 Проектування шаблону інтерфейсу підсистеми архівації.....	38

3	РОЗРОБКА ІНФОРМАЦІЙНОГО РЕСУРСУ КАФЕДРИ НА ХМАРНІЙ ПЛАТФОРМІ І ПІДСИСТЕМИ АРХІВАЦІЇ РОБІТ, ПУБЛІКАЦІЙ СПІВРОБІТНИКІВ КАФЕДРИ І СТУДЕНТІВ.....	40
3.1	Реалізація архітектури інформаційного ресурсу кафедри	40
3.1.1	Опис фреймворка Spring	41
3.1.2	Опис контейнера ІоС	41
3.1.3	Опис фреймворка Spring MVC	42
3.1.4	Опис фреймворка Spring Security	43
3.1.5	Опис JDBC	44
3.2	Опис СУБД PostgreSQL	44
3.2.1	Реалізація фізичної схеми бази даних.....	45
3.3	Реалізація модуля конвертації даних.....	45
3.4	Опис класів додатку	49
3.4.1	Реалізація Model паттерна MVC.....	50
3.4.2	Реалізація View паттерна MVC	51
3.4.3	Реалізація Controller паттерна MVC.....	53
3.5	Керівництво користувача.....	54
	ВИСНОВОК.....	62
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
	ДОДАТОК А. SQL-ЗАПИТИ НА ТРАНСЛЯЦІЮ ПРИВІЛЕЇВ І СТВОРЕННЯ ЕЛЕМЕНТІВ БАЗИ ДАНИХ	66
	ДОДАТОК Б. ПРИКЛАД РЕАЛІЗАЦІЇ КЛАСУ ENTITY	68
	ДОДАТОК В. ПРИКЛАД РЕАЛІЗАЦІЇ КЛАСУ CONTROLLER	70
	ДОДАТОК Г. АКТ ВПРОВАДЖЕННЯ.....	75

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

SOA — Service-Oriented Architecture.

MVC — Model-View-Controller.

REST — Representational State Transfer.

IoC — Inversion of Control.

JDBC — Java DataBase Connectivity.

НДР — науково-дослідна робота.

СУБД – система управління базами даних.

ВСТУП

В даний час все більше і більше підприємств, навчальних закладів переходять до формату електронного документообігу, відходячи від звичного формату роботи з документацією — з паперами безпосередньо. Даний перехід значно спрощує роботу з документами. Електронний варіант роботи з документацією дозволяє витратити менше часу і ресурсів на її обробку та аналіз. Більш того, в зв'язку з масовою появою і використанням програм електронного документообігу в кожній організації, що використовує такі програми, постає питання про закриття документів, виведенні їх з оперативного використання, об'єднанні в логічні структури і передачі їх на зберігання в архів.

Одним із способів підвищення ефективності управління навчальним процесом і документообігом є впровадження спеціалізованої інформаційної системи. Така система повинна систематизувати і спростити процес зберігання, редагування та аналізу документів кафедри.

Проте, жодна така система документообігу не може обійтися без підсистеми архівації для подальшого зберігання та аналітики документації. Для цього було вирішено реалізовувати окрему підсистему, яка буде займатися виключно архівацією, зберіганням даних що містяться в архіві документів, а також надавати доступ для їх перегляду, необхідні засоби для роботи з документами всередині архіву і містити функціонал для аналітики вмісту архіву з відповідним розмежуванням доступу. [1]

Метою даної дипломної роботи є проектування і розробка інформаційного ресурсу кафедри на хмарній платформі і підсистеми архівації робіт, публікацій співробітників і студентів випускаючої кафедри.

Для досягнення даної мети необхідно вирішити такі завдання:

- 1) провести аналіз систем документообігу і процесів архівації в них;

- 2) провести аналіз існуючих принципів архівування документів в сучасних інформаційних системах;
- 3) провести проектування архітектури інформаційного ресурсу кафедри з урахуванням реалізації підсистеми архівації;
- 4) провести проектування моделі даних для предметної області «документообіг випускаючої кафедри»;
- 5) формування типових ролей користувачів і політики безпеки підсистеми архівації та інформаційного ресурсу кафедри;
- 6) виконати проектування функціональних вимог користувача архіватора;
- 7) виконати проектування бази даних для підсистеми архівації;
- 8) провести проектування конвертера документації між підсистемами документообігу та архівації;
- 9) виконати проектування шаблону сторінок і навігації підсистеми архівації;
- 10) розробити системну архітектуру інформаційного ресурсу кафедри і здійснити вибір технологій реалізації підсистеми архівації;
- 11) розробити базу даних і бізнес логіку підсистему архівації інформаційного ресурсу кафедри;
- 12) провести налагодження і тестування розробленої підсистеми архівації з метою розміщення в інтернет-хмарі.

ВИСНОВОК

У даній дипломній роботі був спроектований і реалізований інформаційний ресурс кафедри на хмарної платформі, а також підсистема архівації робіт, публікацій співробітників кафедри і студентів.

При виконанні даної роботи був проведений аналіз таких інформаційних систем як «Магеллан» і «Directum RX», розглянута реалізація функцій архівації систем, досліджені концептуальні підходи архівації документації та даних в сучасних інформаційних системах.

Проведено аналіз процесу документообігу випускаючої кафедри, який дозволив побудувати відповідну обраної предметної області модель даних. У моделі даних були виділені основні типи діяльності кафедри і відповідного їм документообігу:

- документообіг навчального процесу;
- документообіг наукового процесу;
- документообіг організаційного процесу;
- документообіг процесу матеріально-технічного забезпечення.

Визначено перелік документів і їх структура для типових процесів діяльності кафедри.

Проведено проектування архітектури інформаційного ресурсу кафедри, в якій визначені базові модулі підсистеми архівації: модуль конвертації даних і модуль підсистеми архівації.

Спроектовано і реалізовано підсистема архівації документів кафедри, в тому числі база даних підсистеми, бізнес-логіка та інтерфейс підсистемі, а також модуль конвертації документів з підсистеми документообігу в підсистему архівації.

Для реалізації використані сучасні технології, такі як мова програмування Java, фреймворк Spring, зокрема Spring MVC, СУБД PostgreSQL, фронт-енд фреймворк Bootstrap та інші.

Інформаційний ресурс кафедри і підсистема архівації розташовані за адресою <http://91.208.138.238> мережі Інтернет, що забезпечує територіально-розподілений доступ співробітників кафедри відповідно до прав доступу, що надані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 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.
2. 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.
3. Даниленко А.Ю, Подрабинович А.А., Основные принципы организации архивного хранения документов, выведенных из оперативного электронного документооборота / Арлазаров В.Л., Н. Емельянов // Сб. Организационное управление и искусственный интеллект – URSS, 2003
4. Система управления учебным процессом «Магеллан» [Электронный ресурс] – Режим доступа: <https://magellanius.ru/> – 06.04.2017.
5. Система управления документооборотом «Directum RX» [Электронный ресурс] – Режим доступа: <http://gx.directum.ru/> – 27.04.2017.
6. Компоненты сетевого приложения. Клиент-серверное взаимодействие и роли серверов. [Электронный ресурс] – Режим доступа: <http://www.4stud.info/networking/lecture5.html> – 06.12.2013.
7. Архитектурные особенности проектирования и разработки Веб-приложений [Электронный ресурс] — Режим доступа: <http://www.intuit.ru/studies/curriculum/956/courses/467/lecture/28784?page=3> – 14.04.2017.
8. IoC контейнер [Электронный ресурс] — Режим доступа: <https://habrahabr.ru/post/131993/> – 06.11.2011.

9. Model-View-Controller [Электронный ресурс] — Режим доступа: <https://tproger.ru/articles/mvc/> – 17.07.2015.
10. Spring Security [Электронный ресурс] — Режим доступа: <https://projects.spring.io/spring-security/> – 03.05.2017.
11. Spring Security Reference [Электронный ресурс] — Режим доступа: <https://docs.spring.io/spring-security/site/docs/current/reference/htmlsingle> – 05.05.2017.
12. Java Database Connectivity [Электронный ресурс] — Режим доступа: <https://docs.oracle.com/javase/tutorial/jdbc/> – 29.04.2017.
13. Система управления базами данных PostgreSQL [Электронный ресурс] — Режим доступа: <https://www.postgresql.org/> – 21.04.2017.

ДОДАТОК А.
SQL-ЗАПИТИ НА ТРАНСЛЯЦІЮ ПРИВІЛЕЇВ І СТВОРЕННЯ ЕЛЕМЕНТІВ
БАЗИ ДАНИХ

```
//SQL-запрос на создание таблицы Document
```

```
CREATE TABLE "archive"."Document"  
(  
    documentId serial NOT NULL primary key,  
    documentName varchar NOT NULL,  
    documentAuthor varchar NOT NULL,  
    documentPath varchar NOT NULL,  
    documentDateOfArchiving date NOT NULL,  
    documentAttributes jsonb NOT NULL,  
    documentComments varchar,  
    documentArchivePath varchar NOT NULL,  
    documentArchivingTerm real NOT NULL  
);
```

```
//SQL-запрос на создание таблицы Protocol
```

```
CREATE TABLE "archive"."Protocol"  
(  
    protocolId serial NOT NULL primary key,  
    protocolName varchar NOT NULL,  
    protocolDate date NOT NULL,  
    protocolTime time NOT NULL,  
    protocolAction varchar NOT NULL,  
    protocolComments varchar,  
    protocolDocumentId integer NOT NULL references  
"archive"."Document"(documentId),  
    protocolUser varchar NOT NULL  
);
```

//SQL-запрос на создание таблицы ProtocolOfDelete

```
CREATE TABLE "archive"."ProtocolOfDelete"
(
    protocolDelId serial NOT NULL primary key,
    protocolDelName varchar NOT NULL,
    protocolDelDate date NOT NULL,
    protocolDelTime time NOT NULL,
    protocolDelComments varchar,
    protocolDelUser varchar NOT NULL,
    protocolDelDocName varchar NOT NULL,
    protocolDelDocAuthor varchar NOT NULL,
    protocolDelDocArchivePath varchar NOT NULL,
    protocolDelDocDateOfArchiving date NOT NULL,
    protocolDelDocArchivingTerm real NOT NULL,
    protocolDelDocComments varchar
);
```

//SQL-запрос на выдачу прав oc_root'у на использование схемы archive.

```
GRANT SELECT, INSERT, UPDATE, DELETE
ON ALL TABLES IN SCHEMA "archive"
TO oc_root;
```

//SQL-запрос на выдачу прав oc_root'у на использование последовательностей схемы archive.

```
GRANT USAGE, SELECT, UPDATE
ON ALL SEQUENCES IN SCHEMA "archive"
TO oc_root;
```

ДОДАТОК Б.
ПРИКЛАД РЕАЛІЗАЦІЇ КЛАСУ ENTITY

```
public class EntityDocument {

    private Integer id;
    private String name, author, path, comments, archivePath;
    private JSONObject attributes;
    private double archivingTerm;
    private String dateOfArchiving;

    public EntityDocument() {
    }

    public EntityDocument(Integer id, String name, String author,
String path, String comments, String archivePath, JSONObject
attributes, double archivingTerm, String dateOfArchiving) {
        this.id = id;
        this.name = name;
        this.author = author;
        this.path = path;
        this.comments = comments;
        this.archivePath = archivePath;
        this.attributes = attributes;
        this.archivingTerm = archivingTerm;
        this.dateOfArchiving = dateOfArchiving;
    }

    public EntityDocument(String name, String author, String
path, String comments, String archivePath, JSONObject attributes,
double archivingTerm, String dateOfArchiving) {
```



```
    this.name = name;
    this.author = author;
    this.path = path;
    this.comments = comments;
    this.archivePath = archivePath;
    this.attributes = attributes;
    this.archivingTerm = archivingTerm;
    this.dateOfArchiving = dateOfArchiving;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

//more getters and setters
```

ДОДАТОК В.
ПРИКЛАД РЕАЛІЗАЦІЇ КЛАСУ CONTROLLER

```
@Controller
@RequestMapping("/archive/archivation")
public class ControllerConvert {

    private ApplicationContext context;

    private la3v.logic.entities.archive.EntityDocument
documentToArchive;
    private EntityDocument documentFromDocumentSystem;
    private SimpleDateFormat dateFormat = new
SimpleDateFormat("yyyy-MM-dd");
    private SimpleDateFormat timeFormat = new
SimpleDateFormat("HH:mm:ss");
    private Map<String, IAttributes> attributeMap = new
AttributesMap().getAttributeMap();
    private Map<String, String> attributeRusEng = new
AttributesTranslate().getAttributesRusEng();

    // Ініціалізація логера
    private static final Logger log =
LoggerFactory.getLogger(ControllerArchive.class);

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

```

    @RequestMapping(value =("/{id}", method = RequestMethod.GET)
    public String showArchivation(Model model,
    @PathVariable("id") Integer docId) throws Exception {
        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        documentFromDocumentSystem =
repositoryDocument.findById(docId);

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

        model.addAttribute("documentToArchive",
documentFromDocumentSystem);
        model.addAttribute("addressPart", docId);
        model.addAttribute("date", dateFormat.format(new
Date()));

        return "archive/documentToArchive";
    }

    @RequestMapping(value =("/{id}", method = RequestMethod.POST)
    public String toArchive(@ModelAttribute("documentToArchive")
la3v.logic.entities.archive.EntityDocument documentFromForm,
        BindingResult bindingResult, Model
model,
        @PathVariable("id") Integer docId)
throws Exception {

        documentToArchive = documentFromForm;

```

```
        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        EntityDocument document =
repositoryDocument.findById(docId);

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

        String docType =
attributeRusEng.get(document.getDocType().toLowerCase());

        if(docType == null)
            throw new Exception("There is no such attribute in
attribute map!");
        return "redirect:{id}/attributes/" + docType;
    }

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

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

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

```

attributes.getFromJson(documentFromDocumentSystem.getAttributes()
);

        model.addAttribute("documentFromSystem",
documentFromDocumentSystem);
        model.addAttribute("documentFromSystemAttr",
attributes.getFromJson(documentFromDocumentSystem.getAttributes()
));

        model.addAttribute("attributes", attributes);
        model.addAttribute("documentToArchive",
documentToArchive);
        model.addAttribute("addressPart", documentId);

        log.info(String.format("IN /archivation/{id}/attributes/"
+ docType + " GET"));

        return "archive/attributes/" + docType.toLowerCase();
    }

    private void archiveDocument(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();

        documentToArchive.setAttributes(obj);

```

```

        IRepositoryArchive repositoryArchive =
context.getBean(IRepositoryArchive.class);
        IRepositoryDocument repositoryDocument =
context.getBean(IRepositoryDocument.class);
        repositoryArchive.insertDocument(documentToArchive);
        Integer maxProtocolId =
repositoryArchive.getMaxProtocolId() + 1;
        EntityProtocol entityProtocol = new EntityProtocol(
            principal.getName(),
            "Протокол " + maxProtocolId,
            "Добавление в архив",
            "Добавление в архив документа " +
documentToArchive.getName() + " пользователем " +
principal.getName() + " в " + dateFormat.format(new Date()) + " "
+ timeFormat.format(new Date()),
            dateFormat.format(new Date()),
            timeFormat.format(new Date()),

repositoryArchive.getDocument(documentToArchive).getId()
        );
        repositoryArchive.insertProtocol(entityProtocol);

repositoryDocument.deleteFromCoauthor(Integer.parseInt(documentId
.substring(1,documentId.length() - 1)));

repositoryDocument.deleteFromProcess(Integer.parseInt(documentId.
substring(1,documentId.length() - 1)));

repositoryDocument.deleteFromDocument(Integer.parseInt(documentId
.substring(1,documentId.length() - 1)));    }

```

ДОДАТОК Г. АКТ ВПРОВАДЖЕННЯ