

ШЕЛУДЧЕНКО М.Д., АСТИТОВА Т.І.

**ДОСЛІДЖЕННЯ ТА РОЗРОБКА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ
МЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ З ВИКОРИСТАННЯМ
МОВИ ПРОГРАМУВАННЯ JAVA НА ОСНОВІ ТЕХНОЛОГІЙ SPRING ТА
HIBERNATE**

SHELUDCHENKO M.D., ASTISTOVA T.I.

**RESEARCH AND DEVELOPMENT OF MATHEMATICAL SOFTWARE FOR CONTENT
MANAGEMENT SYSTEM USING PROGRAMMING LANGUAGE JAVA BASED TECHNOLOGIES
SPRING AND HIBERNATE**

The main aim of the work was to set up frameworks known Java programming language and almost completely automate the installation of a content management system based instruments. The basis of organization and methods taken MVC concept which is one of the best to create the CMS. This model is designed to divide the whole of the system in three parts during Model- View- Controller. Each of them will be responsible for processing, display and control system. This interaction provides an opportunity to improve the system without having to rewrite it completely.

The research in this paper has made it possible to combine popular frameworks for Java environment to achieve the common goal of creating a single content management system to further facilitate the work of programmers and development companies.

Keywords: Content, Spring MVC, Spring Security, CMS, Hibernate. Keywords: Content, Spring MVC, Spring Security, CMS, Hibernate.

Вступ

Як правило, основна ціль кожної системи управління контентом – це автоматизація наповнення та створення сайту. Якщо система потребує значних до опрацювань та великої кількості правок для досягнення бажаного результату, то така система управління контентом немає попиту на ринку серед інших систем.

Саме тому, задля уникнення таких ситуацій і використовується математичні методи для автоматизації та інтелектуального підстроювання системи на етапі створення сайту. Основні методи що дають змогу вирішити більшість проблем це концепція MVC, допоміжні бібліотеки та фреймворки.

За основу було взято Веб фреймворк Spring MVC, фреймворк Spring Security, Бібліотеку Hibernate, MySQL в якості СУБД.

Постановка завдання

На основі фреймворку Spring MVC та Hibernate дослідити методи побудови систем управління контентом.

Основна частина

Spring найбільш відомий як джерело розширень, потрібних для ефективної розробки складних бізнес-додатків поза великовагових програмних моделей, які історично були домінуючими в промисловості.

Цей фреймворк пропонує послідовну модель і робить її придатною до більшості типів додатків, які вже створені на основі платформи Java. Вважається, що Spring Framework реалізує модель розробки, засновану на кращих стандартах індустрії, і робить її доступною в багатьох областях Java.

На сьогоднішній день багато уваги приділено роботі з базами даних а саме з реляційними базами. Завдання об'єктно-реляційного відображення даних на сьогодні вирішено створенням бібліотеки Hibernate.

Hibernate не тільки вирішує завдання зв'язку класів Java з таблицями бази даних (і типів даних Java з типами даних SQL), а й також надає можливості для автоматичної генерації і оновлення набору таблиць, побудови запитів і обробки отриманих даних і може значно зменшити час розробки, яке зазвичай витрачається на ручне написання SQL- і JDBC-коду. Hibernate автоматизує генерацію SQL-запитів і звільняє розробника від ручної обробки результуючого набору даних і перетворення об'єктів, максимально полегшуючи перенесення (портування) додатків на будь-які бази даних SQL.

Основна мета застосування MVC концепції полягає в відділенні бізнес-логіки (моделі) від її візуалізації (уявлення, виду). За рахунок такого поділу підвищується можливість повторного використання. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах і (або) з різних точок зору.

Концепція MVC дозволяє розділити дані (модель), уявлення і обробку дій (вироблену контролером) користувача на три окремих компоненти:

- Модель (Model):
 - Надає знання: дані та методи роботи з цими даними;
 - Реагує на запити, змінюючи свій стан;
 - Не містить інформації, як ці знання можна візуалізувати;
- Вид (View) - відповідає за

відображення інформації (візуалізацію). Часто в якості уявлення виступає форма (вікно) з графічними елементами;

- Контролер (Controller) - забезпечує зв'язок між користувачем і системою: контролює введення даних користувачем і використовує модель і уявлення для реалізації необхідної реакції.

Для налаштування та створення програмного продукту було виділено декілька етапів:

1. Створення та налаштування проекту в IDE.

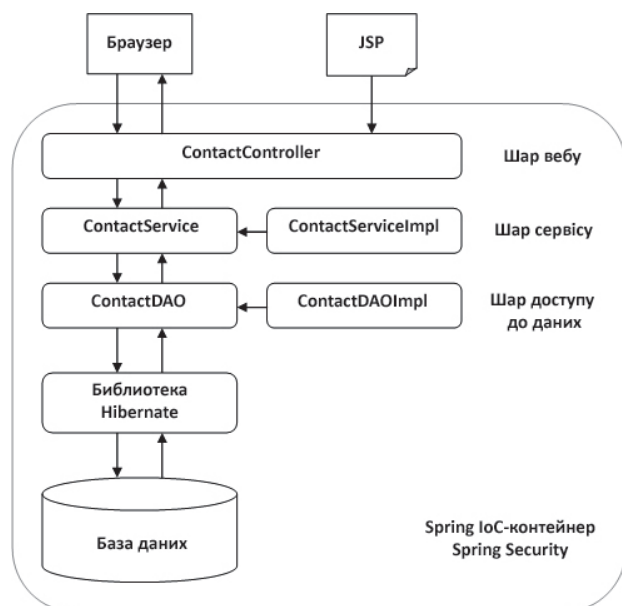


Рисунок 1 Схема організації CMS

2. Створення структури пакетів.
3. Додавання класу суті в модель домену.
4. Налаштування поясу доступу до даних.
5. Створення про шарку сервіс.
6. Додавання веб моделі.
7. Створення та налаштування «Контролер».
8. Створення та налаштування «Вид».
9. Запуск програми.
10. Дослідження та доопрацювання системи на проблеми безпеки.

Приклад коду «Контролера»:

```
package net.schastny.contactmanager.web;
import java.util.Map;
import net.schastny.contactmanager.domain.Contact;
import net.schastny.contactmanager.service.ContactService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
@Controller
public class ContactController {
    @Autowired
    private ContactService contactService;
    @RequestMapping("/index")
    public String listContacts(Map<String, Object> map) {
        map.put("contact", new Contact());
        map.put("contactList", contactService.listContact());
        return "contact";    }
    @RequestMapping("/")
    public String home() {
        return "redirect:/index";    }
    @RequestMapping(value = "/add", method = RequestMethod.POST)
    public String addContact(@ModelAttribute("contact") Contact contact,
        BindingResult result) {
        contactService.addContact(contact);
        return "redirect:/index";    }
    @RequestMapping("/delete/{contactId}")
    public String deleteContact(@PathVariable("contactId") Integer contactId) {
        contactService.removeContact(contactId);
        return "redirect:/index";
```

Висновки

Нам вдалось поєднати популярні фреймворки для середовища Java задля досягнення єдиної мети, створення єдиної системи управління контентом. Дане програмне забезпечення покликане зменшити час на

розробку системи управління контентом як для однієї особи так і для компаній що займаються розробкою веб сайтів для корпоративних мереж. Подальший розвиток і вдосконалення продукту дасть змогу повністю автоматизувати встановлення системи управління контентом та керування самим контентом.

Література

1. Spring Framework Reference Documentation - Rod Johnson , Juergen Hoeller , Keith Donald - 2016 рік.
2. https://ru.wikipedia.org/wiki/Spring_Framework
<https://www.infoq.com/presentations/Mastering-Spring-MVC-3>

ЩЕРБАНЬ В.Ю., МАРЧЕНКО К.О., ЛОПАЧУК Ю.Є.,
КОТОНОС Д.О., САМІЛИК Є.С., СЕМЕНЮК В.В.

СТРУКТУРНО-ЛОГІЧНА СХЕМА САПР ОБЛАДНАННЯ ТА ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ

SCHERBAN V.Ju., MARCHENKO K.O., LOPACHUK Ju.E.,
KOTONOS D.O., SAMILUK E.S., SEMENUK V.V.

STRUCTURALLY LOGICAL CHART OF SAPR OF EQUIPMENT AND TECHNOLOGICAL PROCESSES

Single must plug CADD of equipment and technological processes of light and textile industry in itself two subsystems (PS) for the equipment of PS CADD Equipment and PS CADD the Technological process (TP). For an equipment basic will be mechanical and thermal processes. Between two subsystems there is connection of two-sided orientation.

It is necessary to mark that given CADD on a sign «Complication of planning object» will belong to to CADD of very difficult objects with the number of component parts from 10^4 to 10^6 .

On the level of computer-aided design it will belong to to среднеавтоматизированным CADD (from 25 to 50%). On a sign «Complexity of computer-aided design» it will behave to the complex systems.

Вступ

Як об'єкти проектування обираються з одного боку устаткування, а з іншого боку технологічні процеси легкої і текстильної промисловості. Дані об'єкти тісно взаємозв'язані між собою [1]. Це виявляється в тому, що відповідне технологічне устаткування завжди розробляється або удосконалюється під конкретний технологічний процес. Кожна зміна технологічного ланцюжка приводить до обов'язкової модернізації устаткування, яке використовується на даному переході[2]. Дані факти дозволяють оцінювати об'єкти проектування як єдину систему, у якій присутні тісні зв'язки між об'єктами. Причому, дані зв'язки мають двосторонню спрямованість. Наприклад, недосконалість робочих органів технологічного устаткування може привести до серйозних порушень при виконанні певної технологічної операції [1-3].