

ВОЙТОВИЧ В.О., АСТІСТОВА Т.І., КРАСНИТСЬКИЙ С.М.,
**СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ДИНАМІЧНОГО
САЙТУ**

VOITOVICH V.O., ASTISTOVA T.I., KRASNITSKIY S.M.
DYNAMIC WEBSITE CONTENT MANAGEMENT SYSTEM

The basic methods of research were analysis of existent information, in relation to all aspects of the put theme, treatment of her, for determination of main moments. but research of architecture and functionality programmatic to addition, basic of for development network protocols used and model of baseю Keywords: control system by content, software for a management content.

Вступ

Об'єктом дослідження нашої роботи є програмне забезпечення для управління контентом в ситуації відсутності готового рішення для потреб компанії при встановленні системи управління проектом на свої власні сервери, з даного приводу розглядаються концепції MVC, фреймворки Spring і Hibernate[1,4,5]. Вихідні дані наших досліджень плануються бути впровадженими в компанії по розробці програмних продуктів, а також в інші компанії, що потребують планування проектів та контроль за їх створенням.

Основна частина

Як правило, основна ціль кожної системи управління контентом – це автоматизація наповнення та створення сайту. Якщо система потребує значних доопрацювань та великої кількості правок для досягнення бажаного результату, то така система управління контентом немає попиту на ринку серед інших систем управління контенту. 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. Шаблон MVC дозволяє створювати додатки, різні аспекти яких (логіка введення, бізнес-логіка і логіка інтерфейсу) розділені, але досить тісно взаємодіють один з одним. Ця схема вказує розташування кожного виду логіки в додатку. Призначення для користувача інтерфейс розташовується в уявленні. Логіка введення розташовується в контроллері. Бізнес-логіка знаходиться в моделі. Це розділення дозволяє працювати із складними структурами при створенні додатка, оскільки забезпечує одночасну реалізацію тільки одного аспекту. Наприклад, розробник може сконцентруватися на створенні представлення окремо від бізнес-логіки. Зв'язок між основними компонентами додатка MVC також полегшує паралельну розробку. Наприклад, один розробник може створювати представлення, інший - логіку контроллера, а третій - бізнеслогіку моделі Найочевидніша перевага, яку ми отримуємо від використання концепції

MVC, - це чітке розділення логіки представлення (інтерфейсу користувача) і логіки додатка. Підтримка різних типів користувачів, які використовують різні типи пристроїв є загальною проблемою наших днів. Інтерфейс, що надається, повинен розрізнятися, якщо запит приходить з персонального комп'ютера або з мобільного телефону. Модель повертає однакові дані, єдина відмінність полягає в тому, що контроллер вибирає різні види для виведення даних. Окрім ізолювання видів від логіки додатка, концепція MVC істотно зменшує складність великих застосувань. Код виходить набагато більше структурованим, і, тим самим, полегшується підтримка, тестування і повторне використання рішень. Розглянемо один із найпоширеніших видів фреймворків реалізуючого технологію MVC, фреймворк Spring Framework. Ймовірно, найбільш відомий як джерело розширень (features), потрібних для ефективної розробки складних бізнес-додатків поза великоагових програмних моделей, які історично були домінуючими в промисловості. Ще одна його перевага в тому, що він ввів раніше невикористовувані функціональні можливості в сьогоденні найголовніші методи розробки, навіть поза платформи Java.). Spring це не просто фреймворк, а ціле сімейство фреймворків, які застосовуються для побудови якісної архітектури на основі мови Java. Це: 1. Spring Core – основний фреймворк сімейства, який відповідає за впровадження залежностей, що дозволяє розробити гнучку архітектуру завдяки тому, що компоненти програмного додатку зв'язані між собою на рівні інтерфейсів і впроваджені в код із контейнера Spring. 2. Spring Web MVC – надбудова над технологіями Java Servlet. Фреймворк визначає основну архітектуру додатка і розділяє її на три слої (модель, вид та контролер). Це розділення важливе для відокремлення логіки додатку від генерування інтерфейсу користувача, також завдяки цьому в майбутньому буде дуже просто перейти на інший вид інтерфейсу або змінити модель без змін у інтерфейсі користувача. Також фреймворк бере на себе роботу з обробки всіх HTTP запитів клієнта. Фреймворк Spring Framework пропонує послідовну модель і робить її придатною до більшості типів додатків, які вже створені на основі платформи Java.[3, 7]. 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 Інформаційні технології в науці, виробництві та підприємстві Київський національний університет технологій та дизайну 235 відображення даних на сьогодні вирішено створенням бібліотеки Hibernate. Метою Hibernate є звільнення розробника від значного обсягу порівняно низькорівневого програмування щодо забезпечення зберігання об'єктів в реляційній базі даних. Розробник може використовувати Hibernate як в процесі проектування системи класів і

таблиць «з нуля», так і для роботи з вже існуючою базою даних. Hibernate являється надбудовою над JDBC (стандарт взаємодії Java з базами даних). Він допомагає побудувати зв'язки між об'єктами Java і таблицями реляційних баз даних[2,5]. 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. Hibernate забезпечує прозору підтримку збереження даних (persistence) для «POJO» (тобто для стандартних Java-об'єктів); єдина суворона вимога для зберігається класу - наявність конструктора за замовчуванням (без параметрів). Для коректної поведінки в деяких додатках потрібно також приділити увагу методам equals () і hashCode (). Для зберігання даних було обрано СУБД MySQL. Переваги MySQL у її масштабованості, переносності, зв'язності, безпеці, швидкості та зручності.

Висновки

Вихідні дані досліджень можуть бути впровадженими в компанії по розробці програмних продуктів, а також в інші компанії, що потребують планування систем для управління контентом та контроль за їх створенням. Для розробки програмного забезпечення для роботи з контентом було обрано фреймворк Spring і Hibernate .

Література

1. Астістова Т.І., Підгайний М.О. Дослідження та розробка математичного забезпечення для управління контентом на основі технологій spring та hibernate // Інформаційні технології в науці, техніці та підприємстві, К: Освіта України, 2020, с. 232 – 236.
2. Хеффельфингер Д., Разработка приложений Java EE 6 в NetBeans 7 М.: ДМК Пресс, 2015. – 330 с.
3. Spring 5 для профессионалов, 5-е издание /Кларенс Хо, Роб Харроп, Крис Шефер. – К.: Видавництво «Вільямс». – 2016. – 1120 с.
4. Електронний ресурс: [Documentation - 6.1 - Hibernate ORM](#)
5. Хорстманн К.С., Библиотека профессионала. Java 2. Том 1. Основы. / К.С Хорстманн , Г Корнелл– М.: Издательский дом «Вильямс», 2013. – 848 с.:
6. Кларенс Хо , Spring 3 для профессионалов./ Хо Кларенс, Роб Харроп : М.: ООО «И.Д. Вильямс», 2016. – 880 с.