**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗІНА**

**ФАКУЛЬТЕТ «КОМП’ЮТЕРНИХ НАУК»**

**КАФЕДРА НЕПОМНЮ КАК ОНА НАЗЫВАЕТСЯ**

**Пояснювальна записка**

до курсової роботи

на тему «Інтернет магазин мерчендайзу Govno»

Виконали студенти 3 курсу

Групи КС-31

Неліпа Олександр Дмитрович

Кочерещенко Захар Юрійович

Чудна Валерія Батьківна

Групи КС-32

Затерка Павло Батькович

Керівник:

ст. викл Лазурик В.М

Харків – 2020

**РЕФЕРАТ**

Звіт про виконання курсової роботи: 0 с., 0 рис.

Ключові слова:

**ЗМІСТ**

СПИСОК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

SQL (англ. Structured Query Language) – структурована мова запитів, призначена для управління базами даних;

UML (англ. Unified Modeling Language) – структурована мова проектування;

БД – база даних;

СУБД – система управління базами даних;

ООП – об’єктно-орієнтоване програмування;

JSP – Java Server Pages;

JSF – Java Server Faces;

ВСТУП

1 МЕТА РОБОТИ

Мета курсової роботи полягає у створенні інтернет-магазину. Одною з основних задач є проектування та розробка бази даних, де зберігається інформація про клієнтів магазину та товари, які вони можуть купувати.

Досягнення поставленої мети передбачає реалізацію таких етапів:

1. Проведення аналізу предметної області та засобів реалізації програмного забезпечення;

2. Аналіз наявних аналогів;

3. Проектування бази даних та основного функціоналу систем;

4. Вибір програмних засобів для реалізації;

5. Розробка програми;

6. Тестування працездатності створеної програми.

2 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ

3 ПРОГРАМНО-АПАРАТНІ РІШЕННЯ ДЛЯ РОЗРОБКИ

3.1 Вибір системи управління базами даних

Створення та управління базою даних – є одною з найважніших задач при розробці та підтримці будь-якого інтернет-магазину. Саме тому, одним з ключових питань перед початком розробки, було обрання системи управління базами даних (СУБД). У зв’язку з цим, нашою командою було проаналізовано різні класифікації СУБД:

* Ієрархічні;
* Мережеві;
* Реляційні;
* Об’єктно-орієнтовані;
* Об’єктно-реляційні.

Порівнювалися можливості використання MySQL і PostgreSQL серверів, бібліотеки SQLite, NoSQL підходу та інші. У результаті, було обрано MySQL – вільну реляційну клієнт-серверну СУБД компанії Oracle.

Переваги:

* Простота у використанні – MySQL досить легко інсталюється, а наявність безлічі плагінів і допоміжних додатків спрощує роботу з базами даних.
* Великий функціонал – система MySQL має увесь необхідний інструментарій, який може знадобитися в реалізації практично будь-якого проекту.
* Безпека – система створена таким чином, що безліч вбудованих функцій безпеки в ній працюють за замовчуванням.
* Масштабованість – будучи досить універсальною СУБД, MySQL в рівній мірі легко може бути використана для роботи і з малими, і з великими об’ємами даних.
* Швидкість – висока продуктивність системи забезпечується за рахунок спрощення деяких використовуваних у ній стандартів.

3.2 Засоби для створення та адміністрування бази даних

Для створення та адміністрування бази даних використовувалися 2 графічних менеджери:

SQLyog – графічний інтерфейс користувача для популярної системи реляційних баз даних MySQL. Cтворена компанією Webyog Softworks Pvt. Ltd. Із переваг можна виділити надійність, простоту, інтуїтивно зрозумілий інтерфейс, наявність community версії. Створення бази даних проводилося саме у SQLyog.

DataGrip – IDE-інструмент компанії JetBrains для роботи з базами даних MySQL, PostgreSQL, Oracle, SQL Server, Sybase, DB2, SQLite, HyperSQL, Apache Derby і H2. Рішення JetBrains DataGrip включає потужний текстовий редактор з мультикурсорами, забезпечує синтаксичне виділення коду, підтримує інтеграцію з системами контролю версій Git, Subversion і т. д.

3.3. Мова програмування, інструменти та бібліотеки

Для програмування backend частини веб-сайту що розробляється було використано мову програмування java в поєднанні з декількома бібліотеками та одним фреймворком. Java була обрана за низкою причин, по перше це мова яка реалізую принципи ООП, по друге є великий вибір бібліотек, отже є можливість підібрати бібліотеки які максимально відповідають нашим вимогам, і остання та не за значенням причина це наявність декількох систем збірки застосунку. За сервер застосунку було обрано Apache Tomcat, він процює у ролі контейнера сервлетів та реалізує специфікацію JSP та JSF.

Більш детально про бібліотеки, фреймвокр та систему збірки:

JUnit – бібліотека для юніт тестування;

Log4J – бібліотека для логування ходу роботи застосунка;

MySQL-connector – бібліотека що реалізує JDBC підключення до СУБД Mysql;

Був використаний spring framework, а саме дві його частини spring context та spring jdbc.Spring context забезпечує просту та зручну реалізацію останнього з SOLID принципів, а саме dependency inversion (інверсія залежностей).

У свою чергу spring jdbc надає абстракцію над рутинними діями з встановленням з’єднання з базою даних, утримання транзакції та таке інше, що дає змогу сфокусуватися на розробці SQL запитів, замість низькорівневого програмування.

Системою збірки застосунку було обрано maven, це класичне рішення для програм, що написані мовою програмування java.

Системою контроля версій був обран GIT, а саме його реалізація у вигляді GitHub’а. Це надало змогу нашій команді розподіллено працювати над проектом з належним рівнем ефективності.

// updated by Sasha

Для програмування back-end (програмно-апаратної) частини магазину було використано мову програмування Java в поєднанні з декількома бібліотеками та одним фреймворком (framework). Java була обрана за низкою причин:

1. Реалізація принципів ООП;

2. Великий вибір бібліотек, отже є можливість підібрати бібліотеки які максимально відповідають нашим вимогам

3. Наявність декількох систем збірки застосунку.

Більш детально про бібліотеки:

* JUnit – бібліотека для юніт (модульного) тестування;
* Log4J – бібліотека для логування ходу роботи застосунка;
* MySQL-connector – бібліотека що реалізує JDBC підключення до СУБД Mysql;

У якості фреймворка був використаний spring framework, а саме дві його частини: spring context та spring jdbc. Spring context забезпечує просту та зручну реалізацію останнього з SOLID принципів, а саме dependency inversion (інверсія залежностей).

У свою чергу, spring jdbc надає абстракцію над рутинними діями з встановленням з’єднання з базою даних, утримання транзакції та інше, що дає змогу сфокусуватися на розробці SQL запитів, замість низькорівневого програмування.

Системою збірки застосунку було обрано maven – це класичне рішення для програм, що написані мовою програмування java.

За сервер застосунку було обрано Apache Tomcat. Він процює у ролі контейнера сервлетів (servlet) та реалізує специфікацію JSP та JSF.

Системою контроля версій було обрано GIT, а саме його реалізація у вигляді GitHub. Це надало змогу нашій команді розподіллено працювати над проектом з належним рівнем ефективності.

// Паша, напиши про JS

4 ПРОЕКТУВАННЯ І РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

4.1 Проектування структури БД

Огляд основних сутностей і атрибутів:

1. users – сутність, що містить інформацію щодо користувачів інтернет-магазину.

Атрибути:

* id – унікальний ідентифікатор користувача;
* login – обране ім’я користувача, під яким він зможе заходити в особистий кабінет;
* password – умовне слово або набір символів, призначений для підтвердження особи користувача. Зберігати пароль у відкритому вигляді в БД небезпечно, тому при реєстрації, пароль користувача шифрується спеціальним алгоритмом;
* email – служба для отримання електронних повідомлень від інтернет-магазину. Використовується для інформування користувача щодо знижок, підтверджень оплати, появи нових товарів тощо;
* paymentMethod – дані платіжного методу користувача, що використовуються для онлайн оплати товару в інтернет-магазині. Зашифровані;
* contactNumber – номер користувача інтернет-магазину, за яким, у разі виникнення певних питань, з ним можна буде зв’язатися. Зашифрований.

2. categories – сутність, що використовується для створення ієрархії товарів та їх групування.

Атрибути:

* id – унікальний ідентифікатор категорії;
* title – назва категорії;
* description – містить додаткову інформацію про категорію.

3. vendors – сутність, що містить інформацію щодо виробника товару (у контексті тематичності нашого інтернет-магазину – виробником може бути: серія книг, кіно всесвіт, ігровий всесвіт і т.д). Його необхідно вказувати для того, щоб користувачі могли орієнтуватися на якість товару через рейтинги популярності бренду.

Атрибути:

* id – унікальний ідентифікатор виробника;
* title – назва виробника;
* description – містить додаткову інформацію про виробника.

4. shipping\_methods – сутність, що містить інформацію щодо компаній-посередників, які займаються доставкою товарів з інтернет-магазину.

Атрибути:

* id – унікальний ідентифікатор служби доставки;
* title – назва служби доставки;
* phone – номер технічної підтримки служби доставки, за якою користувачі зможуть звернутися у разі виникнення певних питань.

5. shipping\_addresses – сутність, що містить адресу користувачів, вказану при замовлені товарів з інтернет-магазину.

Атрибути:

* id – унікальний ідентифікатор адресу доставки;
* city – місто;
* region – область;
* street – вулиця;
* building – номер будинку;
* index – індекс міста.

6. product – сутність, що містить статичну (незмінну) інформацію щодо товару, розміщеному в інтернет-магазині.

Атрибути:

* id – унікальний ідентифікатор товару;
* title – назва товару;
* categoryID – ідентифікатор сутності **categories**, що вказує до якої категорії відносять той чи інший товар;
* vendorID – ідентифікатор сутності **vendors**, що вказує хто є виробником товару;
* description – містить будь-яку додаткову інформацію про товар.

7. product\_attributes – сутність, що містить основну інформацію щодо товару та його додаткові властивості (колір, розмір, вагу тощо).

Атрибути:

* id – унікальний ідентифікатор товару з поточними атрибутами;
* productID – ідентифікатор сутності **product**, що вказує на основний товар;
* сolor – колір, один з атрибутів товару;
* size – розмір, один з атрибутів товару;
* price – ціна товару з поточними атрибутами;
* photo – фото товару з поточними атрибутами;
* amountAvailable – кількість доступного для замовлення товару.

8. products\_orders – сутність, що містить інформацію щодо поточного замовлення користувача.

Атрибути:

* productOrderID – ідентифікатор поточного замовлення користувача;
* productsAttributesID – ідентифікатор сутності **product\_attributes**, що вказує який товар та з яким поточним атрибутом замовив користувач інтернет-магазину;
* amount – кількість замовленого товару;
* currentPrice – поточна ціна замовленого товару (фактично, ціна \* кількість).

9. active\_orders – сутність, що містить інформацію щодо усіх замовлень користувача.

Атрибути:

* id – унікальний ідентифікатор замовлення;
* userID – ідентифікатор сутності **user**, що вказую хто (який користувач) замовив товар;
* productOrder – ідентифікатор сутності **products\_orders**, що вказує на поточне замовлення товару;
* shippingMethodID – ідентифікатор сутності **shipping\_methods**, що вказує на обрану службу доставки;
* shippingAddressID – ідентифікатор сутності **shipping\_addresses**, що вказує на адресу доставки;
* price – фінальна ціна всього замовлення (з урахуванням доставки, знижок);
* currentStatus – статус замовлення (комплектується, у дорозі, доставлено тощо);
* datePlaced – дата і час замовлення;
* comment – коментар (додаткова інформація).

Схему бази даних наведено на рис. N:



Рисунок N – схема бази даних (изменю когда будет финальный вариант БД)

4.2 Створення таблиць

1. Таблиця users.

Таблиця 4.2.1

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| login | varchar (50) | ім’я користувача |
| password | varchar (50) | пароль користувача |
| email | varchar (50) | електрона пошта користувача |
| payment Method | varchar (50) | метод оплати користувача |
| contactNumber | varchar (15) | контактний номер користувача |

2. Таблиця categories.

Таблиця 4.2.2

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| title | varchar (50) | назва категорії |
| description | varchar (255) | опис категорії |

3. Таблиця vendors.

Таблиця 4.2.3

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| title | varchar (50) | назва поставника |
| description | varchar (255) | додаткова інформації |

4. Таблиця shipping\_methods.

Таблиця 4.2.4

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| title | varchar (50) | назва служби доставки |
| phone | varchar (15) | телефон технічної підтримки |

5. Таблиця shipping\_addresses

Таблиця 4.2.5

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| city | varchar (30) | місто |
| region | varchar (30) | область |
| street | varchar (30) | вулиця |
| building | varchar (30) | будівля |
| index | varchar (30) | індекс |

6. Таблиця product

Таблиця 4.2.6

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| title | varchar (100) | назва товару |
| categoryID | int | зовнішній ключ категорії |
| vendorID | int | зовнішній ключ виробника |
| description | varchar (255) | опис |

7. Таблиця product\_attributes.

Таблиця 4.2.7

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| productID | int | зовнішній ключ продукту |
| color | varchar (20) | колір |
| size | varchar (20) | розмір |
| price | double | ціна |
| photo | varchar (255) | фотографія |
| amountAvailable | int | кількість доступно |

8. Таблиця products\_orders

Таблиця 4.2.8

|  |  |  |
| --- | --- | --- |
| productOrderID | int | поточне замовлення |
| productsAttributesID | int | зовнішній ключ товару атрибута |
| amount | int | кількість товару |
| currentPrice | double | поточна ціна |

9. Таблиця active\_orders

Таблиця 4.2.9

|  |  |  |
| --- | --- | --- |
| id | int | первинний ключ таблиці |
| userID | int | зовнішній ключ користувача |
| productOrder | int | зовнішній ключ поточного замовлення |
| shippingMethodID | int | зовнішній ключ методу доставки |
| shippingAddressID | int | зовнішній ключ адресу доставки |
| price | double | повна ціна замовлення |
| currentStatus | varchar (50) | статус замовлення |
| datePlaced | date | дата замовлення |

4.3 Реалізація запитів на SQL

Будет, когда будут запросы

5 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ КЛІЄНТА ДО БД

Клиента, к сожалению, пока еще нет.

ВИСНОВОК

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ