

Міністерство освіти і науки України
Національний технічний університет України “Київський
політехнічний інститут”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Пояснювальна записка
до модульної контрольної роботи
“Технології розробки корпоративних застосувань ”.
тему:
“ Сервіс замовлення букетів ”

Виконав: студент 3 курсу
групи ІП-42 Усик Р. В.

Перевірив: ас. Подрубайло О. О.

Кількість балів: _____

ЗМІСТ

ЗМІСТ	2
АНОТАЦІЯ	3
ВСТУП	4
ЗАГАЛЬНІ ПОЛОЖЕННЯ	5
Характеристика об'єкту автоматизації	5
Постановка задачі	6
ВИМОГИ ДО СИСТЕМИ.....	7
Вимоги до системи.....	7
Вимоги до функцій системи	7
Вимоги до програмного забезпечення	7
Вимоги до технічного забезпечення.....	7
ОПИС ВИДІВ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	8
Моделі та засоби проектування	8
Технології та засоби розробки системи	8
Інформаційне забезпечення системи	9
Програмне забезпечення.....	10
Entity	11
Repository	11
Service	12
Web.....	14
ОПИС СИСТЕМИ.....	15
Опис структури системи та опис кожної підсистеми	15
Опис функціонування системи	15
ВИСНОВКИ.....	17
ЛІТЕРАТУРА.....	18

АНОТАЦІЯ

Дана модульна контрольна робота призначена для здобуття навичок розробки корпоративних застосунків за допомогою засобів JavaEE. В результаті повинна бути розроблена повноцінна система для купівлі букетів. Також повинні бути розроблені модулі, які використовують JPA, JTA, JSF, EJB, JMS, MDB, JAAS

ВСТУП

У даній модульній контрольній роботі розробляється система для купівлі букетів. Дана тема є актуальною, оскільки на сьогодні не багато систем забезпечують користувачів повним набором функціоналу для зручності замовлень. Система, що розробляється повинна відповідати наступним вимогам:

1. Можливість реєстрації користувачів.
2. Безпечна аутентифікація та авторизація користувачів.
3. Можливість перегляду, додавання, редагування та видалення букетів.
4. Можливість покрокового формування замовлення.
5. Можливість підтвердження чи скасування замовлення.
6. Можливість управління замовленнями.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Характеристика об'єкту автоматизації

Система «Сервіс замовлення букетів» повинна реалізовувати такі бізнес-процеси:

1. Реєстрація та авторизація користувачів.
2. Управління букетами: перегляд, додавання, редагування та видалення.
3. Інтерфейс для купівлі букетів.
 - 3.1. Інтерфейс для формування замовлення (додавання та видалення букетів до замовлення)
 - 3.2. Інтерфейс для підтвердження чи скасування замовлення
4. Інтерфейс для управління замовленнями (для адміністратора / дизайнера).

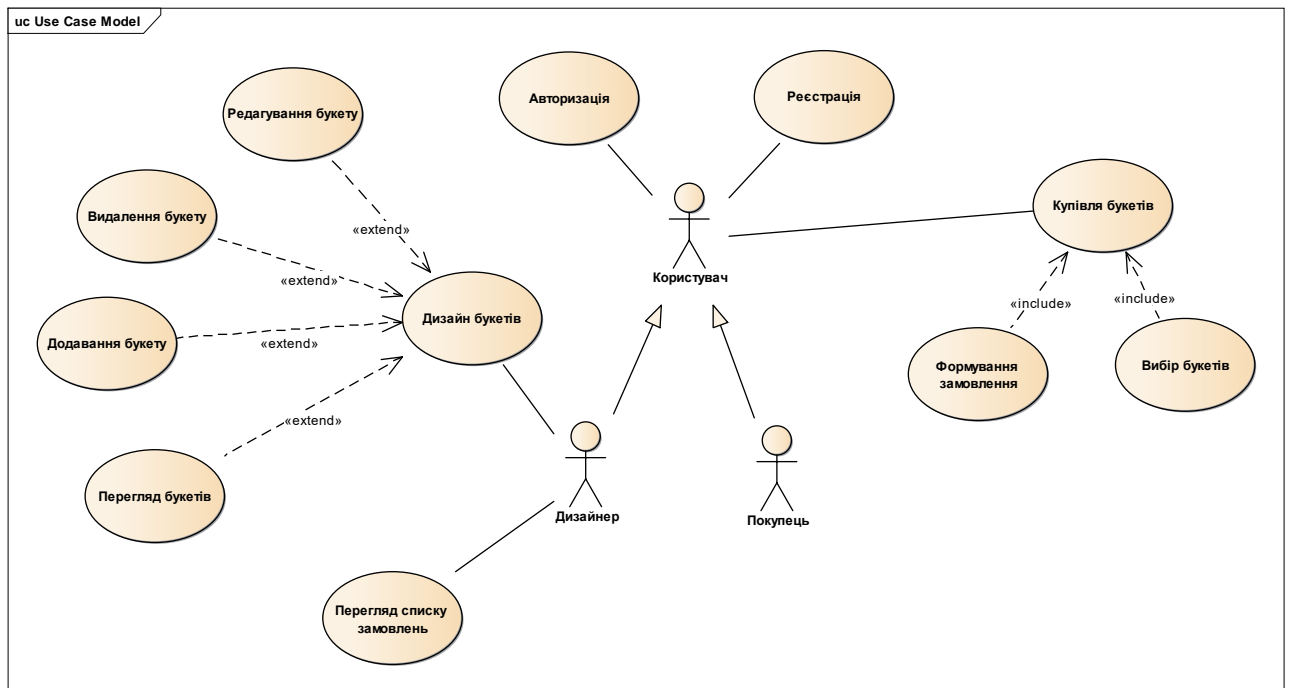


Рисунок 1.1. Use case діаграма

Постановка задачі

Метою створення системи «Сервіс замовлення букетів» є створення програмного застосунку, який дозволяє додавати та редагувати наявні в системі букети та надання користувачеві інтерфейсу для обрання та придбання цих букетів. В результаті розробки системи мають бути досягнуті наступні показники:

1. Можливість реєстрації користувачів.
2. Безпечна аутентифікація та авторизація користувачів.
3. Можливість перегляду, додавання, редагування та видалення букетів.
4. Можливість покрокового формування замовлення.
5. Можливість підтвердження чи скасування замовлення.
6. Можливість управління замовленнями.

ВИМОГИ ДО СИСТЕМИ

Вимоги до системи

Система «Сервіс замовлення букетів» повинна виконувати такі операції:

1. Дизайн та управління букетами.
2. Створення замовлення з букетів.
3. Можливість управління замовленнями.

Вимоги до функцій системи

Система «Сервіс замовлення букетів» повинна забезпечувати такі функції:

1. Реєстрація та авторизація користувачів.
2. Управління букетами: перегляд, додавання, редагування та видалення.
3. Інтерфейс для купівлі букетів.
4. Інтерфейс для управління замовленнями (для адміністратора / дизайнера).

Вимоги до програмного забезпечення

Для роботи системи необхідне таке програмне забезпечення:

1. Java 8 SE, Java EE 7.
2. MySQL.
3. WildFly 10.0.

Вимоги до технічного забезпечення

Для роботи системи необхідне таке технічне забезпечення:

1. Сервер зі встановленим програмним забезпеченням, що описано вище.
2. Браузер на стороні клієнта.

ОПИС ВИДІВ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Моделі та засоби проектування

На етапі проектування системи «Сервіс замовлення букетів» побудовані такі моделі:

1. Use case model.
2. ER-diagram.
3. Діаграма фізичної моделі даних.

Для побудови діаграм на етапі проектування використовувались такі CASE-засоби:

1. Enterprise Architect.
2. IntelliJ IDEA.
3. Workbench MySQL.

Технології та засоби розробки системи

При розробці системи «Сервіс замовлення букетів» були використані такі технології:

1. Java Servlet API
2. Java Persistence API
3. Java Server Faces
4. Enterprise Java Bean
5. Java Authentication and Authorization Service
6. AJAX
7. Java Message Service
8. Message Driven Beans
9. Log4j
10. Bootstrap

При розробці системи «Сервіс замовлення букетів» були використані такі засоби:

1. IntelliJ IDEA.
2. MySQL

Інформаційне забезпечення системи

При організації інформаційного забезпечення системи «Сервіс замовлення букетів» були використана база даних MySQL. Для розробки її схеми була використана технологія JPA з Hibernate реалізацією. На діаграмі нижче наведено ER-модель бази даних.

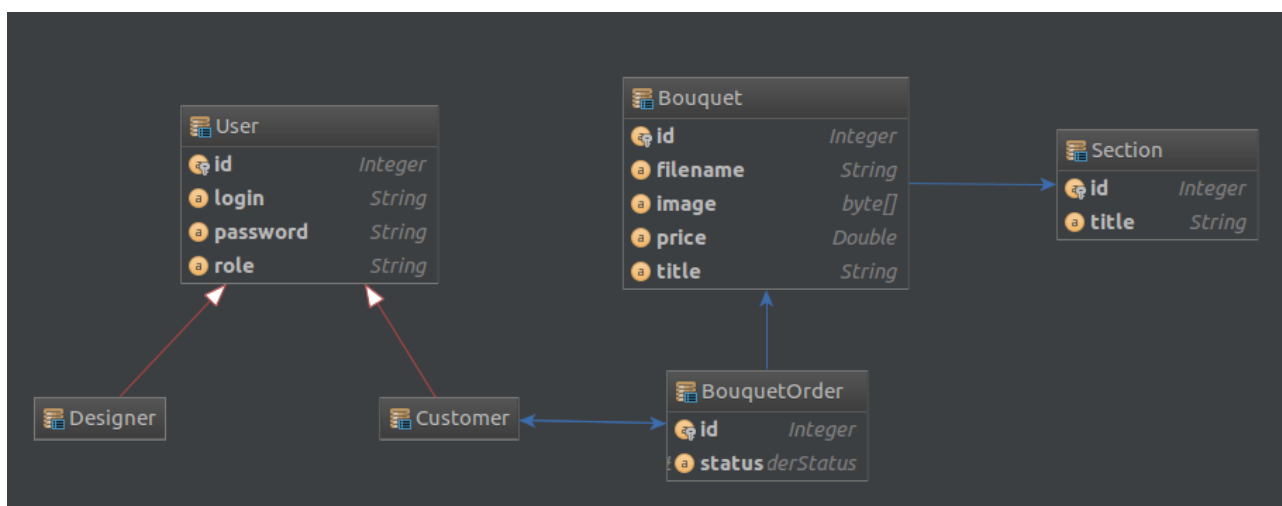


Рисунок 1.2. ER-модель бази даних

Відповідно, база даних, що була створена технологією JPA, може бути представлена наступною фізичною моделлю.

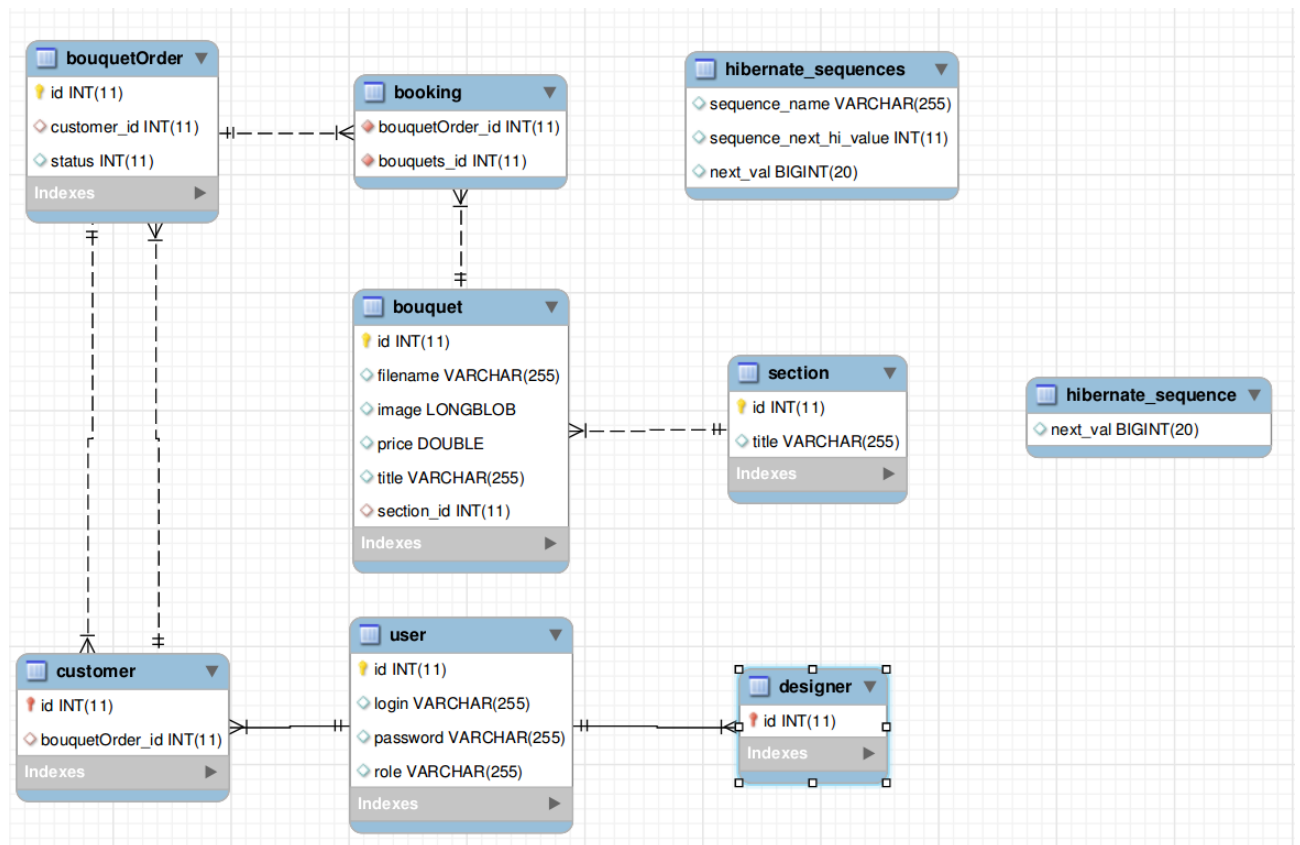


Рисунок 1.3. Фізична модель бази даних

Взаємодія між клієнтом та сервером організована за допомогою протоколу HTTP. Взаємодія компонентів серверної частини реалізована з використанням технологій JSF, EJB та JMS.

Програмне забезпечення

Архітектура системи «Сервіс замовлення букетів» має такі частини:

1. Entity
2. Repository
3. Service
4. Web

Далі наведено опис кожної з частин:

Entity

В цій частині описані сутності (Entity) у вигляді JPA Entity-класів. Всі вони мають структуру POJO. Далі наведено перелік всіх сутностей:

1. Bouquet.java
2. BouquetOrder.java
3. Customer.java
4. Designer.java
5. Section.java
6. User.java

Repository

В цій частині описані класи, які забезпечують CRUD-операції для JPA Entity-класів. Всі ці класи реалізують методи інтерфейсу DAO.java:

1. `Class<T> getEntityClass()` – повертає клас сутності.
2. `List<T> getAll()`
3. `void create(T entity)`
4. `T get(PK key)`
5. `void update(T entity)`
6. `void delete(T entity)`

Всі ці операції реалізовані в generic класі AbstractDAO.java. Кожному Entity відповідає клас з Repository, який конкретизує generic клас AbstractDAO.java:

1. BouquetDAO.java
2. BouquetOrderDAO.java
3. CustomerDAO.java
4. DesignerDAO.java
5. SectionDAO.java

6. UserDao.java

Service

В цій частині описані класи, які забезпечують логіку системи. Вони використовують класи з Repository та реалізують весь функціонал, необхідний для роботи системи.

1. BouquetService – клас для доступу до букетів. Основні методи:
 - 1.1. `public List<Bouquet> getAllBouquets()` – повертає всі букети.
 - 1.2. `public List<Bouquet> getAllBouquetsWithFilter()` – повертає всі букети зі вказаними параметрами фільтрації.
 - 1.3. `public void removeBouquetById()`
 - 1.4. `public void addBouquet()`
 - 1.5. `public void updateBouquet()`
2. ImageService – web-servlet для отримання зображень з бази даних.
Основні методи:
 - 2.1. `public void doGet(...)` – за кодом букету, що передається як параметр запиту, повертає зображення букету.
3. LoginService – клас, що призначений для хешування введеного користувачем паролю.
 - 3.1. `public static String getEncodedPassword(String password)` – повертає захешований за алгоритмом SHA-256 пароль.
4. LogoutService – клас, що забезпечує авторизацію та вихід з облікового запису користувача.
 - 4.1. `public void logout()`
 - 4.2. `public void logoutOrLogin(String username)` – виконує операцію logout або Login в залежності від того, чи авторизований користувач
5. NavLink – клас, який інкапсулює гіпертекстове посилання, тобто рядок-посилання та рядок-заголовок.

6. OrderDisplayService – клас для відображення списку замовлень.
 - 6.1. `public List<BouquetOrder> getAllOrders()`
 - 6.2. `public void removeOrderById(Integer bouquetOrderId)`
7. OrderListUpdater – клас для асинхронного оновлення списку замовлень.
 - 7.1. `public void onMessage(Message message)` – метод для асинхронного оновлення замовлення по коду, переданому в message.
8. OrderRepository – клас для відображення списку замовлень. Має такі поля:
 - 8.1. `private Map<Integer, BouquetOrder> orderMap` – потокобезпечна мапа для швидкого оновлення замовлень по ключу
 - 8.2. `private AtomicReference<List<BouquetOrder>> orderList` – список замовлень для відображення користувачевіТака схема збереження замовлень дозволяє асинхронно оновлювати з бази даних окремі замовлення по ключу, при цьому швидко відповідати на запити «Відобразити всі замовлення». Це реалізовано в методах:
 - 8.3. `private void list2map()` – конвертує список в мапу
 - 8.4. `private void map2list()` – конвертує мапу в список
 - 8.5. `public void updateOrderById(Integer bouquetOrderId)` – оновлює замовлення в мапі
 - 8.6. `public void removeOrderById(Integer bouquetOrderId)` – видаляє замовлення з мапи
 - 8.7. `public List<BouquetOrder> getOrderList()` – повертає список замовлень
9. OrderService – клас для роботи із замовленнями. Основні методи:
 - 9.1. `public void addBouquetToOrder(Integer bouquetId)`
 - 9.2. `public void removeBouquetFromOrder(Integer bouquetId)`
 - 9.3. `public void buyAll()` – підтверджує замовлення
 - 9.4. `public void clearAll()` – видалення всіх букетів із замовлення

9.5. `public List<Bouquet> getAllBouquetsInOrder()`

10. `SectionService` – клас для отримання категорії букету.

11. `UserNavBarList` – клас для отримання допустимих сторінок в меню для кожної ролі.

11.1. `public List<NavLink> getNavigationList(String role)` – повертає список допустимих сторінок в меню для заданої ролі

12. `UserService` – клас для доступу до користувачів.

12.1. `public void registerCustomer()`

12.2. `public User getUserByLogin(String login)`

Web

В цій частині описані веб-форми, які забезпечують відображення елементів системи. Всі вони є JSF-формами, деякі використовують технологію AJAX.

ОПИС СИСТЕМИ

Опис структури системи та опис кожної підсистеми

Система «Сервіс замовлення букетів» складається з таких підсистем:

1. Реєстрація та авторизація користувачів – забезпечує реєстрацію, вхід та вихід із системи.
2. Управління букетами – створення, оновлення, видалення букетів.
3. Формування замовлень – створення, редагування та підтвердження замовлень.

Опис функціонування системи

1. Реєстрація та авторизація користувачів
 - 1.1. Реєстрація користувачів.
 - 1.2. Аутентифікація користувачів.
 - 1.3. Авторизація користувачів.
 - 1.4. Вихід із системи.
2. Управління букетами
 - 2.1. Перегляд букетів.
 - 2.2. Створення букетів.
 - 2.3. Оновлення букетів.
 - 2.4. Видалення букетів.
3. Формування замовлень
 - 3.1. Створення замовлень.
 - 3.2. Редагування замовлень.
 - 3.3. Підтвердження замовлень.

Взаємодію підсистем можна поділити на дві категорії.

1. Взаємодія між підсистемою «Реєстрація та авторизація користувачів» та іншими підсистемами. Реалізується через JSF форми та механізм JaaS: користувач звертається до підсистеми, потрапляє до підсистеми «Реєстрація та авторизація користувачів», виконує вхід або реєстрацію. Подібним чином реалізовано і вихід з системи.
2. Взаємодія між підсистемою «Управління букетами» та підсистемою «Формування замовлень» реалізована за допомогою JMS з використанням MDB. При додаванні чи модифікації замовлення підсистема «Формування замовлень» генерує повідомлення, яке обробляється MDB, який редагує потрібне замовлення та інформує про це підсистему «Управління букетами». Таким чином підсистема «Формування замовлень» завжди має актуальну репліку таблиці замовлень без необхідності постійного звернення до бази даних.

ВИСНОВКИ

Після процесу проектування класів та визначення взаємодії між ними, а також програмної реалізації нашої системи, ми можемо стверджувати, що розроблена система може у повній мірі задовольнити функціонал, що був описаний у третьому розділі даної роботи. Так, вона може автоматизувати купівлю букетів.

JIITEPATYPA

1. JSF 2.0 Tutorials. - <http://www.coreservlets.com/JSF-Tutorial/jsf2/>
2. JSF for nonbelievers: The JSF application lifecycle. – <http://www.ibm.com/developerworks/library/j-jsf2>
3. Mike Keith, Merrick Schincariol. Pro JPA 2: Mastering the Java™ Persistence API. – APress: 2009. – 538 p.
4. Bruce W. Perry. Java Servlet & JSP Cookbook. - O'Reilly. - 2004. - 746p.
5. Bob McCune. Exploring the Java Persistence API. - <http://www.intertech.com/resource/usergroup/Exploring%20the%20JPA.pdf>
6. J2EE BluePrints. Designing Enterprise Applications with the J2EE Platform.. http://www.adobe.com/support/jrun/documentation/pdf/j2ee_ent_app_design
7. Deby Panda, Reza Rahman, Derek Lane. EJB 3 in Action. – Manning: 2007.- 677p.