

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та кібербезпеки

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

**Розробка створення клієнт-серверного додатку для служби доставки
вантажів логістичного підприємства**

Студента 4 курсу, 7 групи,
спеціальності (6. 050103
«Програмна інженерія»)
спеціалізації Інженерія
програмного забезпечення

підпис
студента

Мосійчука Євгена
Володимировича

Науковий керівник,
завідувач кафедри програмної
інженерії та кібербезпеки
доктор технічних наук,
професор
Гарант освітньої програми,
кандидат технічних наук,
доцент

підпис
керівника

Криворучко Олена
Володимирівна

підпис
керівника

Цензура Микола
Олександрович

Київ 2019

ЗМІСТ

ВСТУП.....	3
Розділ 1. Дослідження клієнт-серверної архітектури	5
1.1 Клієнт-серверна архітектура	5
1.1. Технічне завдання.....	7
1.2. Висновки до розділу 1	9
Розділ 2. Розробка моделі бази даних	11
2.1. Постановка завдання.....	11
2.3. Опис вхідних і вихідних даних	12
2.4 Розробка концептуальної та логічної моделі бази даних.....	12
2.5. Розробка фізичної моделі.....	17
2.6. Висновки до розділу 2.....	18
Розділ 3. Розробка клієнт-серверної частини для Web-додатку логістичного підприємства	20
3.1. Побудова серверної частини на базі фреймворку Spring.....	20
3.2. Висновки до розділу 3.....	27
Висновки та пропозиції.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	29

					<i>КНТЕУ 6.050103 07-13.БР</i>			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Акрушів</i>
<i>Зав. Каф.</i>		<i>Криворучко О.В</i>				<i>3</i>	<i>2</i>	<i>29</i>
<i>Керівник</i>		<i>Криворучко О.В.</i>				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група</i>		
<i>Гарант</i>		<i>Цензура М.О.</i>						
<i>Розробив</i>		<i>Мосійчук Є.В.</i>						
					<i>Зміст</i>			

ВСТУП

Тема дипломної роботи «Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства». Цілю дипломної роботи є розробка клієнт серверного застосування для здійснення онлайн замовлень доставки вантажу.

Мета дослідження дипломної проекту полягає в розробці Web сайту з надання послуг доставки вантажу, яка допоможе будь-якому користувачу легко організовувати і замовляти доставку вантажу.

Об'єктом дослідження є процес розробки баз даних і клієнт-серверного застосування для бізнесу.

Предметом дослідження є методи та алгоритми створення програмного забезпечення.

Завданнями дослідження є огляд існуючих концепцій баз даних і побудова власних концептуальних та логічних моделей. Огляд існуючих мов програмування та вибір найбільш прийнятних для вирішення задачі. Створення Web доданку та власної бази даних до нього. Проведення тестування і апробація бази даних.

Розроблений Web доданок має виконувати наступні задачі:

- зберігати відомості про службу доставки товарів, види вантажу, пункти призначення, тарифи ;
- формувати інформацію про заброньовані доставки;
- дозволяти в будь-який час отримувати інформацію про готельний комплекс;

					<i>КНТЕУ 6.050103 07-13.БР</i>			
Змн.	Арк.	№ докум.	Підпис	Дата	<i>Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства</i>	Стадія	Аркуш	Аркушів
Зав. Каф.		Криворучко О.В				В	3	29
Керівник		Криворучко О.В.				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група</i>		
Гарант		Цензура М.О.						
Розробив		Мосійчук Є.В.						

Розробка інформаційної моделі бази даних являє собою проведення докладного словесного опису об'єктів предметної області і реальних зв'язків, які присутні між описаними об'єктами. Нам потрібно розробити інформаційну базу даних, яка міститиме:

- відомості про службу доставки товарів, види вантажу, пункти призначення, тарифи;
- реєстрація клієнтів;
- ведення обліку клієнтів і дані про них.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		4

Розділ 1. Дослідження клієнт-серверної архітектури

1.1 Клієнт-серверна архітектура

У комп'ютерних науках клієнт-серверна архітектура – це модель програмної архітектури, яка складається з двох частин: клієнтська і серверних системи, обидві спілкуються по комп'ютерній мережі або на одному комп'ютері. Клієнт-серверна програма - це розподілена система, що складається як з клієнтського, так і з серверного програмного забезпечення. Застосування клієнтського сервера забезпечує кращий спосіб розподілу робочого навантаження. Процес клієнта завжди ініціює підключення до сервера, а серверний процес завжди чекає запитів від будь-якого клієнта. Клієнт-серверна архітектура складається з таких основних компонентів:

- набір серверів, які забезпечують інформацією чи іншими послугами програми, які звертаються до них (найпростіша ситуація, коли сервер лише один);
- набір клієнтів, які використовують сервіси надані серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

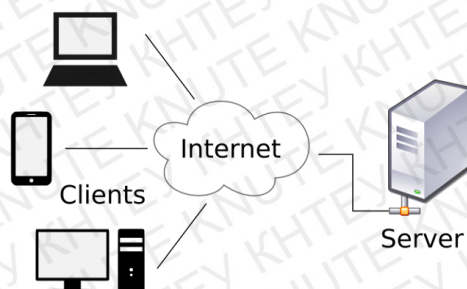


Рис. 1.1. Ілюстрація клієнт-серверної архітектури

					<i>КНТЕУ 6.050103 07-13.БР</i>			
Змн.	Арк.	№ докум.	Підпис	Дата	<i>Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства</i>	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В				<i>РІ</i>	5	29
Керівник		Криворучко О.В.				<i>Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група</i>		
Гарант		Цензура М.О.						
Розробив		Мосійчук Є.В.						
					<i>Розділ I</i>			

Сервер — програмний компонент обчислювальної системи, що виконує сервісні (обслуговуючі) функції по запиту клієнта, надаючи йому доступ до певних ресурсів або послуг.

Клієнт — робоча станція, що взаємодіє з користувачем, здатна виконувати необхідні обчислення і надає з'єднання з обчислювальними ресурсами та БД, засобами їх оброблення, а також засобами організації інтерфейсів.

Сервери не залежать один від одного. Клієнти працюють паралельно і також незалежно один від одного. Загалом, клієнту не потрібно турбуватися про те, як сервер обробляє інформацію під час виконання запиту та отримання відповіді. Клієнт повинен лише інтерпретувати відповідь на основі відомого протоколу додатків, тобто вмісту та форматування даних для запитуваної служби.

Клієнти і сервери обмінюються повідомленнями в формі обміну повідомленнями запит-відповідь. Клієнт відправляє запит, сервер повертає відповідь. Цей обмін повідомленнями є прикладом взаємодії між процесами. Клієнт має знати спільну мову з сервером, і вони повинні слідувати правилам, щоб і клієнт, і сервер знали, чого очікувати. Мова і правила спілкування визначені в протоколі зв'язку. Всі клієнт-серверні протоколи працюють на рівні додатків. Протокол прикладного рівня визначає основні шаблони вимог. Сервер може реалізовувати інтерфейс прикладного програмування (API). API - це рівень абстракції для доступу до сервісу.

Сервер може отримувати запити від багатьох різних клієнтів за короткий проміжок часу. Система може обробляти лише обмежену кількість запитів за певний проміжок часу. Щоб запобігти зловживанням та максимально збільшити доступність, серверне програмне забезпечення

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		6

може обмежити доступність для клієнтів.

Мережі клієнт-серверної архітектури мають наступні переваги:

- Дозволяють створювати мережі з великою чисельністю робочих станцій;
- Забезпечують централізоване керування обліковими записами користувачів, доступом і безпекою;

- Швидкий та гнучкий доступ до мережевих ресурсів;

Між тим клієнт-серверна архітектура має і ряд недоліків:

- Несправність сервера може призвести до втрати мережевих ресурсів та припинення роботи сервісу;
- Необхідність мати кваліфікований персонал для адміністрування та підтримки мережі;
- Висока ціна мережевого обладнання та мереж.

1.1. Технічне завдання

Написання випускного кваліфікаційного проекту передбачає наступне технічне завдання:

Таблиця 1.1.

ЗАГАЛЬНІ ВІДОМОСТІ

Найменування системи	Mobile Delivery
Планові терміни початку та закінчення робіт	Серпень 2018 – Квітень 2019
Головний бенефіціар	Відсутній. ПП не має комерційного характеру.
Потенційні користувачі	Люди, зацікавлені в послугах сервісу по доставці товару.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		7

Продовження таблиці 1.1.

Призначення системи	Надати користувачеві якісний та зручний сервіс для замовлення доставки вантажу.
Мета створення системи	Опанування сучасних технологій для розробки веб-додатків та їх закріплення на практиці.

В ході дослідження поняття «клієнт-серверного додатку» було сформовано перелік основних (мінімальних) та додаткових вимог до кінцевого програмного продукту. Функціональні характеристики ПП перераховані у таблиці 1.2.

Таблиця 1.2.

ФУНКЦІОНАЛЬНІ ВИМОГИ ДО СИСТЕМИ

Основні	➤ реєстрація/авторизація користувача
	➤ онлайн калькулятор послуг
	➤ управління замовленнями
	➤ створення/видалення замовлень
	➤ розділення прав звичайного користувача та адміністратора
	➤ створення сучасного та адаптивного дизайну сайту

Вимоги до програмного забезпечення

Програмний комплекс повинен бути побудований на клієнт-серверній архітектурі з використанням веб-технологій, що не вимагають додаткового ліцензування. Серверна частина програмного комплексу має бути реалізована програмною мовою Java, з використанням фреймворків і бібліотек Spring, Hibernate, системи керування базами даних MySQL на базі веб-сервера Apache Tomcat. При проектуванні програмного комплексу необхідно застосувати архітектуру Model-View-Controller для

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		8

розмежування алгоритмічної частини від інтерфейсу та забезпечення спільного використання компонентів. Збірка проекту виконується за допомогою Apache Maven, вихідні тексти програми зберігаються в репозиторії Git.

Клієнтська частина повинна бути побудована на веб-технологіях, що дозволяють будувати багатофункціональні веб-компоненти. Сторінки інтерфейсу генеруються стандартом HTML5 з використанням бібліотеки Bootstrap 3, щоб створити адаптивний дизайн, оформлення здійснюється за допомогою таблиць стилів CSS, для додання динамічним елементів використовується JavaScript та JQuery, для асинхронної обробки запитів Ajax технологія.

Вимоги до технічного забезпечення

ПП розрахований на функціонування при такому наборі технічних засобів:

- мінімальна апаратна конфігурація сервера:
 - 2-ядерний процесор з тактовою частотою від 1.4ГГц;
 - оперативна пам'ять об'ємом не менше 4Гб;
 - попередньо встановлені програмні модулі – NPM (5<) та JRE (8<);
- мінімальна апаратна конфігурація комп'ютера-клієнта:
 - веб-браузер Mozilla FireFox (39<) або Google Chrome (42<);
 - підключення до мережі Інтернет.

Технічне забезпечення надається хостинговою веб-платформою (ресурси передбачені самою хмарою).

1.2. Висновки до розділу 1

В дослідження теоретичних відомостей була було розглянуто і описано концепцію клієнт-серверної архітектури, принцип її дії та основні переваги була переглянута інформація по різних мовам

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		9

програмування (C#, Java, C++) та системам управління базами даних (MySQL, MS SQL Server, PostgreSQL). На основі досліджень в якості засобів розробки було обрано об'єктно-орієнтовану мову Java і СКБД MySQL Workbench.

Аналізуючи результати проведеного вище дослідження, було розроблено технічне завдання ПП та висвітлено рекомендації щодо його реалізації.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		10

Розділ 2. Розробка моделі бази даних

2.1. Постановка завдання

Одним із завдань було розробити базу даних для роботи служби з доставки вантажів. Програма для роботи з інформацією повинна мати зрозумілий і дружній інтерфейс.

Було використано мову програмування Java, інтегровану середу розробки IntelliJ IDEA. Програма створена, як Web проект на базі фреймворку Apache Maven. Були використана технологія як Servlet. Для збереження усіх необхідних даних використовується MySQL Workbench. Було створено декілька таблиць зі зв'язками між ними. За допомогою SQL-запитів створювалася об'єднана з декількох таблиць реляційна таблиця, яка в повній мірі відображає інформацію про бронювання на даний момент.

2.2. Опис предметної області

Сервіс служби доставки вантажів, який ми будемо розглядати в даній роботі, буде містити базу даних про користувачів і можливості самої служби. Служба дає можливість клієнтам замовити доставку, веде бюлетень по покупцям. Для того, щоб продати зареєструвати клієнта, необхідні наступні дані: прізвище, ім'я, електронна адреса. База даних зберігає інформацію про тарифи служби, види пакування і вантажу, які можна доставляти.

База даних для служби доставки вантажів логістичного підприємства повинна виконувати такі основні завдання:

- зберігати відомості про всіх клієнтів;

					КНТЕУ 6.050103 07-13.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В.				P2	11	29
Керівник		Криворучко О.В.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Мосійчук Є.В.						
					Розділ 2			

- зберігати відомості про здійснені замовлення на доставку вантажу;

Кожна доставка характеризується вагою, маршрутом, датою відправки і отримання, ціною, видом пакування. Підтвердження замовлення здійснюється адміністратором і користувач отримує данні про дату доставки та ціну.

Для цього буде використана спільна база даних, котра включає необхідну інформацію. Програма є актуальною на сьогодні, оскільки вона надає можливість онлайн користування і автоматизує роботу з базою даних.

2.3. Опис вхідних і вихідних даних

У базі даних для служби доставки вантажів логістичного підприємства використовуються наступні вхідні дані:

- інформація про клієнтів;
- інформація про види пакування;
- інформація про поштові пункти;
- інформація про точки маршруту;
- інформація про маршрути;
- інформація про доставки;

У проєктованій базі даних необхідно створити такі обмеження:

- заборонено вводити негативні числові значення;
- доступ до бази даних мають лише адміністратори сервісу доставки.

Вихідною інформацією є результати роботи запитів, вивід загальної інформації щодо замовлення доставки.

2.4 Розробка концептуальної та логічної моделі бази даних

Після аналізу предметної області, наступним етапом є створення концептуальної моделі бази даних. Це графічне зображення в термінах

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		12

предметної області. Концептуальний рівень – це опис на високому рівні інформаційних потреб підприємства. Вона зазвичай включає лише основні поняття та основні відносини між ними. Як правило, це перша модель, з недостатньою деталізацією для створення фактичної бази даних. Цей рівень описує структуру всієї бази даних для групи користувачів. Концептуальна модель також відома як модель даних, яка може бути використана для опису концептуальної схеми при реалізації системи баз даних. Він приховує внутрішні деталі фізичної пам'яті і цілі щодо опису сутностей, типів даних, відносин і обмежень. Інформацію про проект зображають з використанням інфологічної моделі даних "Сутність-Зв'язок".

Логічна модель даних є моделлю даних конкретної предметної області, що виражається незалежно від конкретного сервісу керування базою даних або технології зберігання (фізична модель даних), але з точки зору структури даних, таких як реляційні таблиці і стовпці, об'єктно-орієнтовані класи, або теги XML. Це на відміну від концептуальної моделі даних, яка описує семантику організації без посилання на технологію.

Створення логічної моделі – це ітераційний процес, що складається з етапів оцінювання, аналізу та проектування. На кожному кроці ітерації виникають нові правила. Ефективні засоби проектування баз даних мають бути гнучкими, а організація роботи з ними – ефективною. ER-діаграми мають доповнюватися більш детальною інформацією про бізнес, правила та обмеження, а також давати змогу управляти наочним поданням деталей моделі.

Проаналізувавши предметну область було вирішено розробити базу даних, яка складається з семи таблиць:

- «Доставки»;

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		13

- «Користувачі»;
- «Вантажі»;
- «Пакування»;
- «Маршрути».
- «Точки маршруту»;
- «Поштові відділення».

Таблиця «Вантажі» повинна містити в собі такі поля:

- ідентифікатор вантажу;
- назва вантажу;
- ціна за кілограм;

Таблиця «Точки маршруту» повинна містити в собі такі поля:

- ідентифікатор точки маршруту;
- ідентифікатор поштового відділення;
- ідентифікатор маршруту;

Таблиця «Користувачі» повинна містити в собі такі поля:

- ідентифікатор запису про клієнта;
- прізвище;
- ім'я;
- електронна адреса;
- пароль;
- роль;
- рахунок;

Таблиця «Поштові відділення» повинна містити в собі такі поля:

- ідентифікатор поштового відділення;
- назва відділення;

Таблиця «Доставки» повинна містити в собі такі поля:

- ідентифікатор доставки;
- ідентифікатор вантажу;

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		14

- ідентифікатор пакування;
- ідентифікатор користувача;
- ідентифікатор маршруту;
- дата відправки;
- дата отримання;
- вага вантажу;
- ціна за доставку;
- статус замовлення;
- статус оплати;

Таблиця «Маршрути» повинна містити в собі такі поля:

- ідентифікатор маршруту;
- днів на доставку;

Таблиця «Пакування» повинна містити в собі такі поля:

- ідентифікатор пакування;
- назва пакування;
- ціна пакування;

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		15

На основі зроблених досліджень була створена логічна модель бази даних:

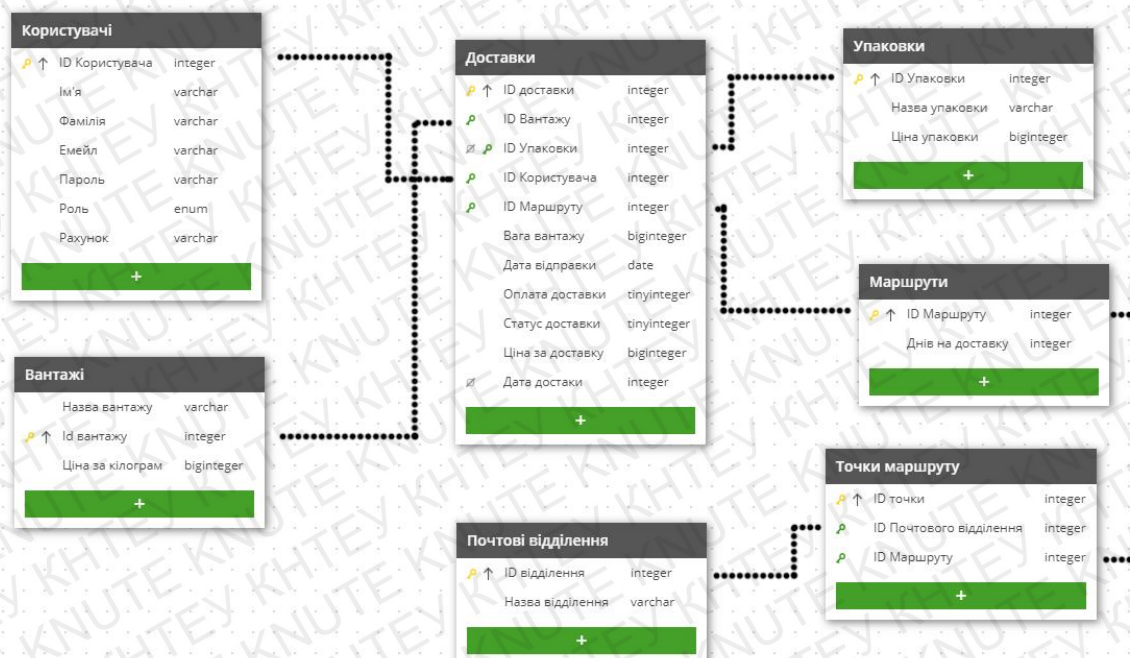


Рис. 2.2. – Логічна модель бази даних логістичного підприємства

Дана схема демонструє усі зв'язки між таблицями і їх атрибути.

Одне з найважливіших достоїнств реляційних баз даних полягає в тому, що можна зберігати логічно згруповані дані в різних таблицях і задавати зв'язки між ними, об'єднуючи їх в єдину базу. Така організація даних дозволяє зменшити надмірність збережених даних, спростує їх введення і організацію запитів і звітів.

В процесі проектування завдання була розроблена БД для логістичного підприємства, яка містить 7 таблиць та зв'язки між ними. Таблиці, в свою чергу, зберігають максимально повну характеристику, інформацію та опис об'єктів згідно з предметної області.

2.5. Розробка фізичної моделі

Фізична модель даних представляє, як модель буде побудована в базі даних. Фізична модель бази даних показує всі структури таблиць, включаючи ім'я стовпця, тип даних стовпця, обмеження стовпців, первинний ключ, зовнішній ключ і відносини між таблицями.

Стадія фізичного проектування БД включає:

- здійснення вибору способу організації БД;
- розробку засобами моделі даних специфікації внутрішньої схеми БД;
- опис концептуальної схеми БД у внутрішній структурі управління файлами.

Після створення даталогічної моделі фізичне проектування включає:

- вибір СКБД;
- опис структури таблиць;
- визначення типів полів для розподілу атрибутів сутностей;
- створення певних додаткових об'єктів як тригери (обробники подій), індекси і процедури, що зберігаються та полегшують обробку даних та пошук в таблицях.

Система керування базами даних надає цілковитий контроль над процесом ідентифікації даних, їх обробки та використанням. СКБД оптимізує та полегшує обробку великих масивів даних, які зберігаються в багаточисленних таблицях.

Різноманітні засоби СКБД надають можливість виконання трьох базових функцій: обробка даних, зберігання даних, оперування даними.

На основі розробленої концептуальної і логічної моделі даних було створено фізичну модель:

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		17

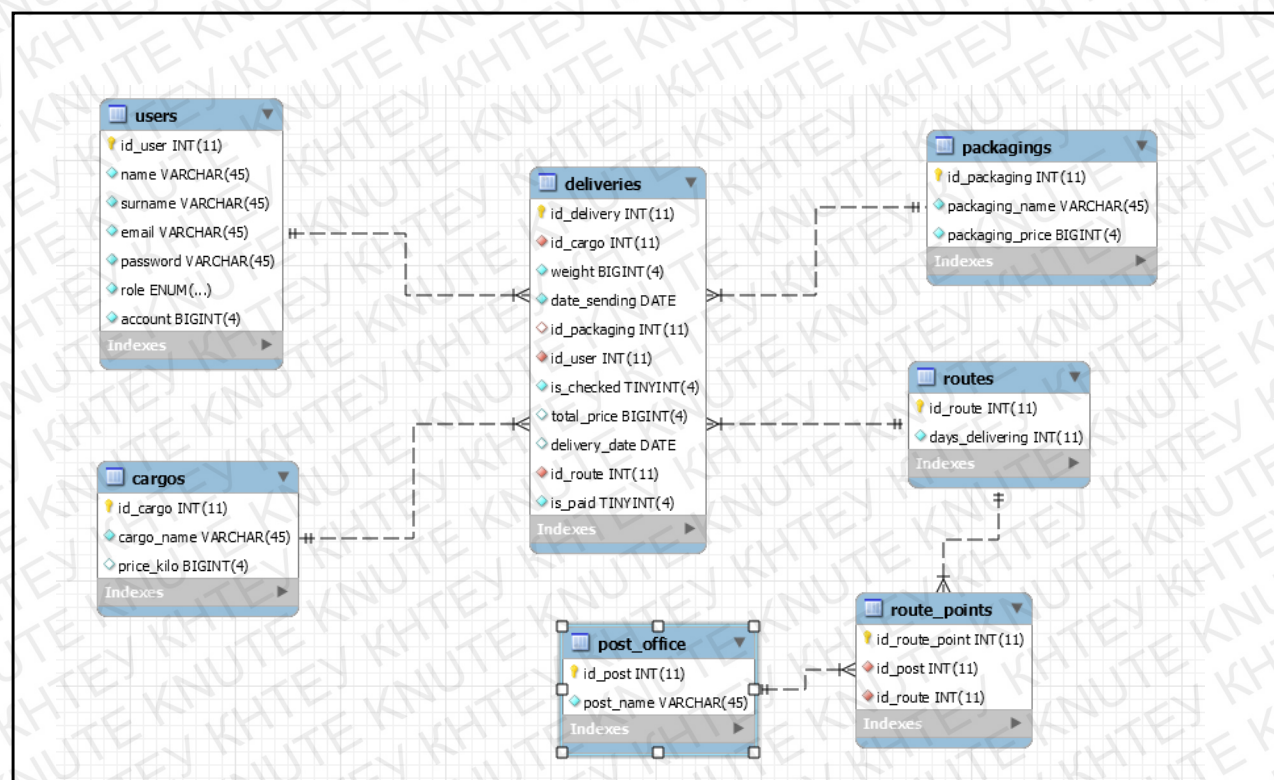


Рис. 2.3. – Логічна модель бази даних логістичного підприємства

Дана схема демонструє усі зв'язки між таблицями, також поля і типи даних, які їм присвоєно.

2.6. Висновки до розділу 2

База даних (скорочено — БД) — це організована структура, метою якої є зберігання, зміна і обробка певної інформації, переважно великих обсягів. У технічному розумінні також і система керування базами даних.

Головне призначення БД — зберігання величезних масивів інформації та надання доступу до неї прикладній програмі або ж користувачеві. База даних складається з двох частин: системи керування інформацією та самими даними. На основі проведеного аналізу предметної області була побудована схема функціональних залежностей, що містить у собі всі атрибути, визначені в ході дослідження, та їх об'єднання.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		18

Під час розробки концептуальної моделі було створено схему зв'язків між таблицями. Згодом було розроблено концептуальну, логічну та фізичну модель даних. На основі отриманих результатів було створено власну базу даних з 7 таблицями, які об'єднані між собою різноманітними зв'язками.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		19

Розділ 3. Розробка клієнт-серверної частини для Web-додатку логістичного підприємства

3.1. Побудова серверної частини на базі фреймворку Spring

Spring Framework - це платформа Java, яка забезпечує всебічну підтримку інфраструктури для розробки додатків Java. Spring обробляє інфраструктуру, щоб програмісти могли зосередитися на розробці додатку.

Spring дозволяє створювати програми із "звичайних старих об'єктів Java" (POJO) і застосовувати корпоративні послуги неінвазивно до POJO. Ця можливість застосовується до моделі програмування Java SE і до повної і часткової Java EE.

Spring web MVC framework забезпечує архітектуру модель-уявлення-контролер (MVC) і готові компоненти, які можуть бути використані для розробки гнучких і слабо пов'язаних веб-додатків. Шаблон MVC призводить до поділу різних аспектів застосування (обробка даних, логіка UI та бізнес-логіка), забезпечуючи при цьому слабкий зв'язок між цими елементами.

1. Model зберігає дані і складається з POJO класів, які інкапсулюють інформацію про сутності доменної області.
2. View відповідає за представлення даних моделі користувачам і в цілому він генерує HTML, який браузер клієнта може інтерпретувати.
3. Controller обробляє запити користувачів і створює необхідні дані для виведення у View і демонстрації користувачеві.

Spring Web model-view-controller (MVC) framework розроблений навколо DispatcherServlet, який обробляє всі HTTP-запити і відповіді.

					КНТЕУ 6.050103 07-13.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства	Стадія	Аркуш	Акрюшів
Зав. Каф.		Криворучко О.В.				РЗ	20	29
Керівник		Криворучко О.В.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Мосійчук Є.В.			Розділ 3			

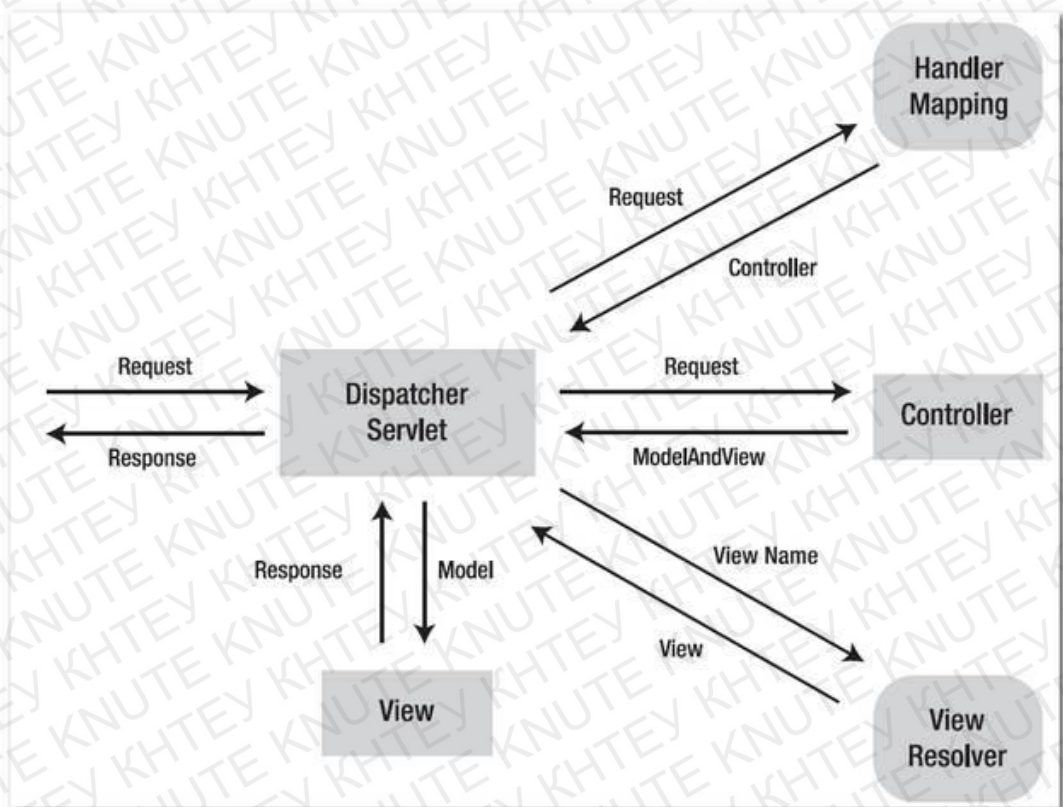


Рис. 3.1. – Ілюстрація процесу обробки HTTP запиту в Spring MVC.

Нижче наводиться послідовність подій, відповідних вхідному запиту HTTP до DispatcherServlet:

1. Після отримання запиту HTTP, DispatcherServlet перевіряє theHandlerMapping для виклику відповідного Контролера.
2. Контролер приймає запит і викликає відповідні методи сервісів на основі використовуваного методу HTTP. Метод сервісу встановлює дані моделі, засновані на певній бізнес-логіці і повертає ім'я View в DispatcherServlet.
3. DispatcherServlet буде приймати допомогу від ViewResolver та вілобразжати певний View для запиту.
4. Після того, як View завершено, DispatcherServlet передає дані моделі у браузер.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		21

WebAppInitializer клас Spring MVC, який відповідає за зчитування конфігурація, запуск Web-додатку і налаштування DispatcherServlet.

```
public class WebAppInitializer implements
WebApplicationInitializer {

    public void onStartup(ServletContext servletContext) throws
ServletException {
        AnnotationConfigWebApplicationContext context = new
AnnotationConfigWebApplicationContext();
        context.register(SpringConfig.class, WebConfig.class,
SpringSecurityConfig.class);
        context.setServletContext(servletContext);

        ServletRegistration.Dynamic dispatcher =
servletContext.addServlet("dispatcher",
new DispatcherServlet(context));

        dispatcher.setLoadOnStartup(1);
        dispatcher.addMapping("/");
    }
}
```

Багаторівнева архітектура – це організація структури проекту в трьох основних категоріях: сервіси, рівень DAO, контролери . Кожен з шарів містить об'єкти, які несуть відповідальність за функціонал того рівня, який він представляє.

Всі залежності йдуть в одному напрямку, від веб шару до інфраструктури. Логіка, пов'язана з проблемою одного шару, не повинна розміщуватися в іншому шарі. Наприклад, в інтерфейсі користувача не повинно здійснюватися бізнес логіка або запити до бази даних.

Рівень контролерів - це верхній шар веб-програми. Він несе відповідальність за обробку вхідних даних користувача і повернення правильного відповіді користувачеві. Веб-шар також повинен обробляти винятки, викинуті іншими шарами. Оскільки рівень контролерів є точкою входу нашої програми, він повинен піклуватися про автентифікацію і виступати в якості першої лінії захисту від неавторизованих користувачів.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		22

Сервісний шар знаходиться нижче веб-шару. Він діє як межа транзакцій і містить послуги інфраструктури. Служби прикладних програм надають відкритий API рівня обслуговування. Вони також

Рівень DAO є найнижчим шаром веб-програми. Він відповідає за взаємодію з використовуваною базою даних.

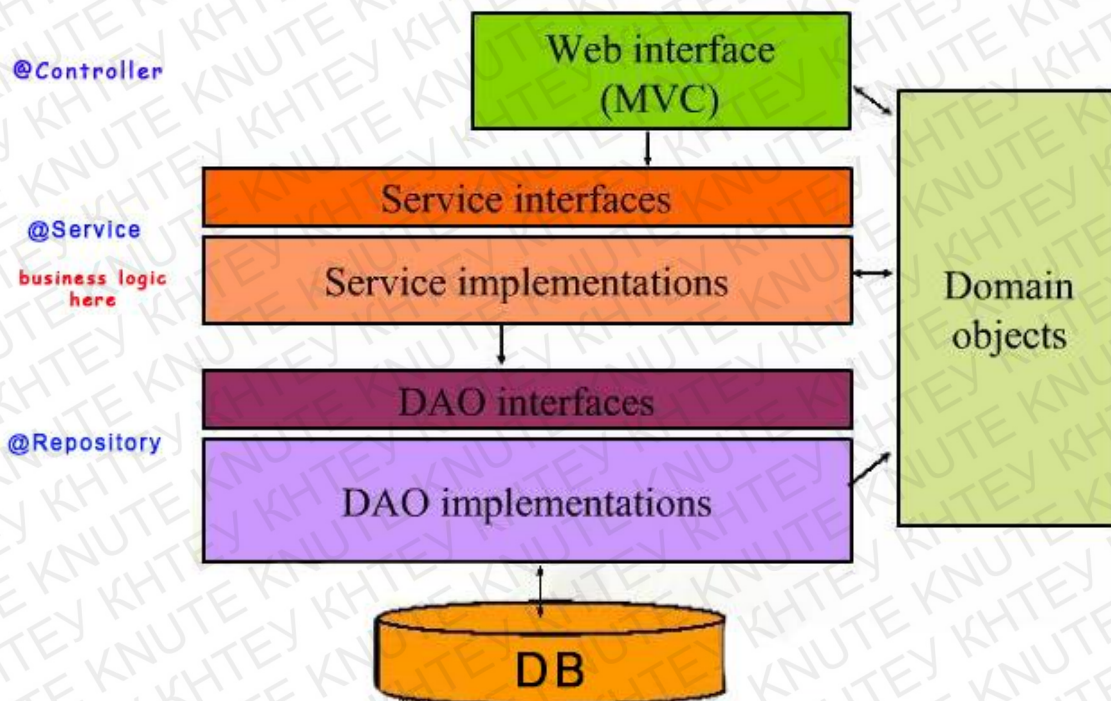


Рис. 3.2. – Багаторівнева архітектура Web-доданку.

Таким чином виглядає структура пакетів проекту з Java класами та їх відношення між собою.

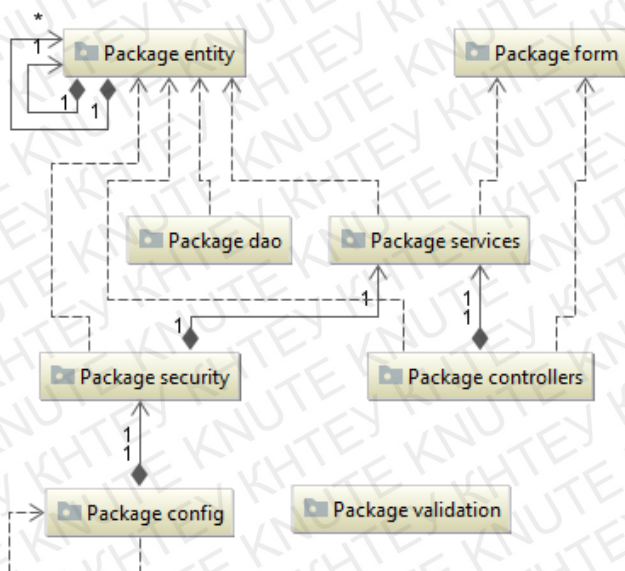


Рис. 3.3. – Структура пакетів проекту.

В структуру проекту входять пакети з Java класами, файли клієнтської частини (HTML, CSS, JavaScript та бібліотеки), файли конфігурації та локалізації, а також файли підключення сторонніх залежностей у проект і пов'язані з віддаленим репозиторієм GitHub.

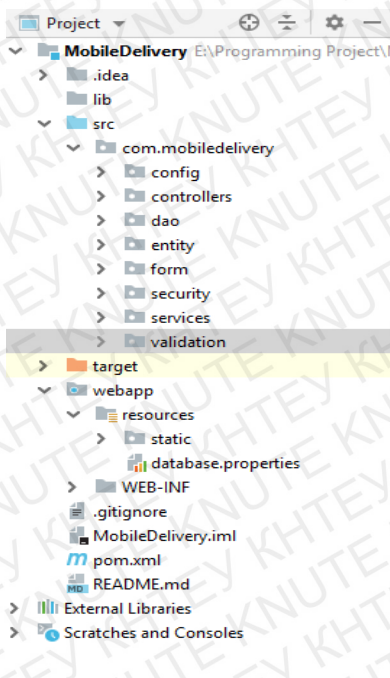


Рис. 3.4. – Структура проекту.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		24

Приклад діаграми класів одного з пакетів:

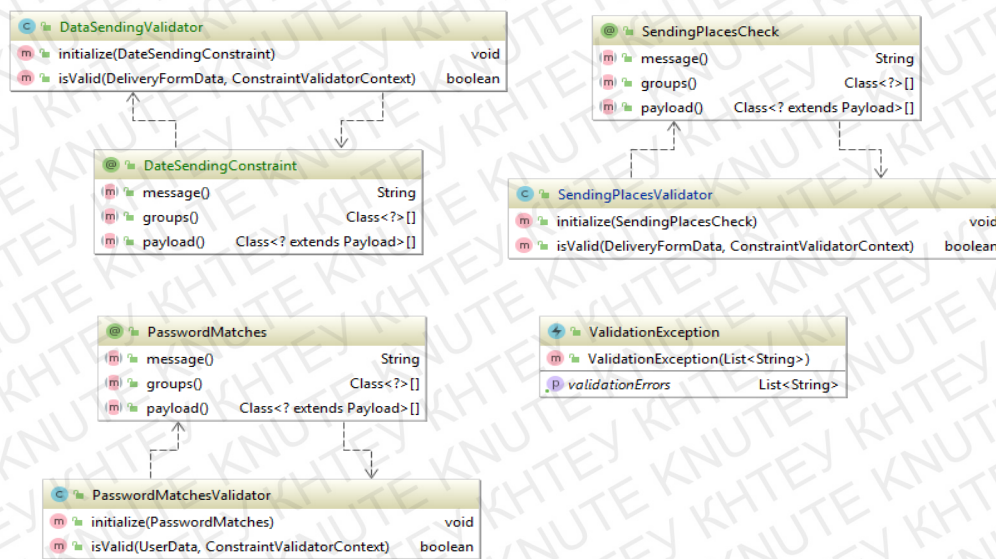


Рис. 3.4. – Діаграма класів пакету com.mobiledelivery.validation.

Щоб зрозуміти як працює обробка запитів в Spring MVC розберемо на прикладі процес реєстрації нового користувача.

Користувач заповнює форму реєстрації, далі за допомогою асинхронного запиту ми передаємо інформацію з форми і робимо запит на сервер для додання нового користувача в базу даних. Знизу зображений процес реєстрації за допомогою діаграми послідовностей.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		25

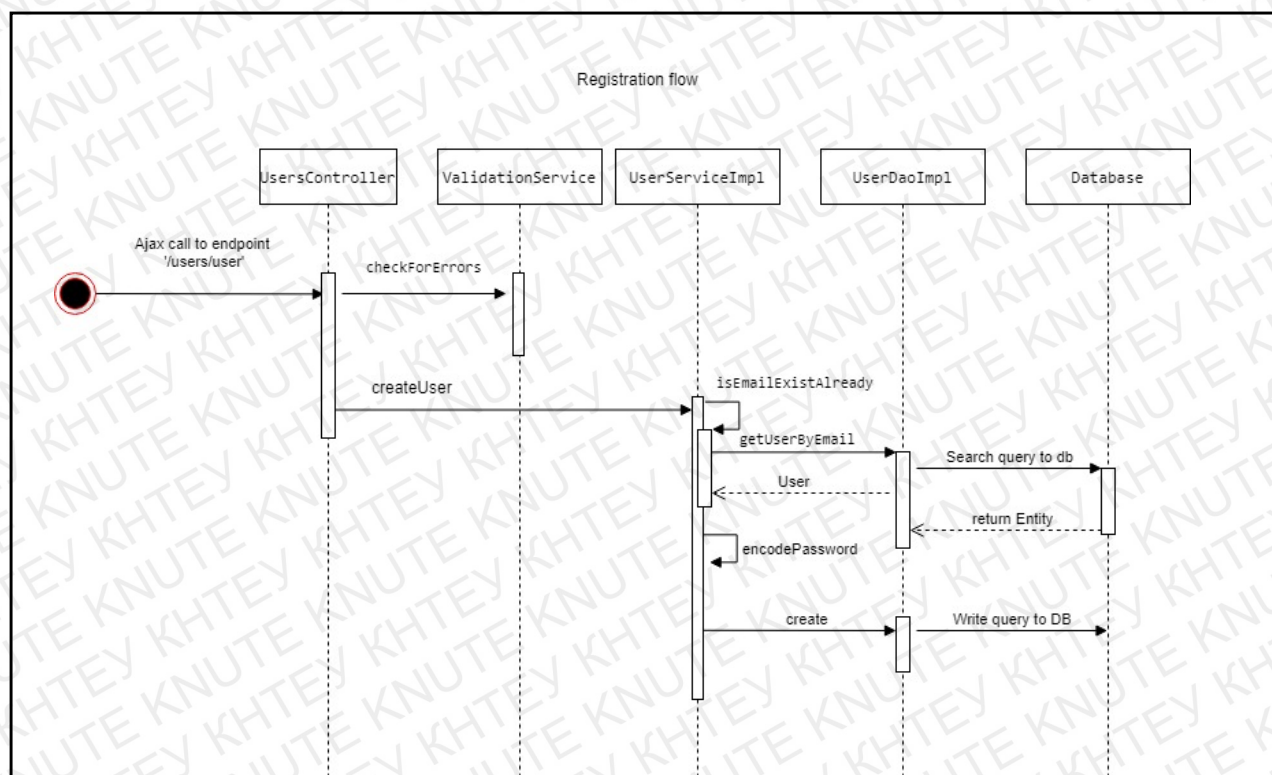


Рис. 3.5. – Діаграма послідовності для процесу реєстрації користувача.

Спочатку ми отримуємо інформацію з форми, перевіряємо дані на валідність та визиваємо сервіс для подальшої обробки. Метод рівня контролера виглядає таким чином:

```

@PostMapping(value = "/user",
    consumes = MediaType.APPLICATION_JSON_VALUE)
@ResponseStatus(HttpStatus.CREATED)
public void registerUser(@Valid @RequestBody UserData userData, BindingResult bindingResult) {
    validationService.checkForErrors(bindingResult);
    User user = mapUser(userData);
    userService.createUser(user);
}
  
```

Далі на рівні сервісів ми перевіряємо чи не існує вже користувач з такою електронною поштою, у разі необхідності створюємо помилку і повертаємо на рівень контролерів, щоб потім сповістити користувача про помилку.

```

@Transactional
public void createUser(User user) throws UserEmailDuplicationException {
    if(isEmailExistAlready(user.getEmail())){
        throw new UserEmailDuplicationException(user.getEmail());}
}
  
```



```

user.setPassword(passwordEncoder.encode(user.getPassword()));
userDao.create(user);
}

```

Якщо немає перешкод для створення нового користувача ми звертаємося до рівня DAO, де відбувається створення запиту до бази даних і безпосереднє його виконання.

```

@Override
public boolean create(User entity) {
    Session session = sessionFactory.getCurrentSession();
    session.persist(entity);
    LOG.info("User with email {} was created.", entity.getEmail());
    return true;
}

```

3.2. Висновки до розділу 3

В ході написання цього розділу було створено серверну частину Web-додатку для взаємодії з додатком Служби доставки вантажів. Розробка здійснювалася у інтегрований середі розробки IntelliJ Idea з автоматизованою збіркою проекту в Maven.

Використовувалися такі елементи: Servlet, *.css файли, *.jsp файли, ServletFilter, патерни проектування Command, Builder, DAO, MVC, Singleton, SqlConnection, JDBC, JSTL, на основі бази даних «Готельний комплекс».

Було створено структуру проекту згідно патерна проектування MVC.

Розроблено модель, контролер і представлення.

Отже, в ході написання цього розділу було розроблено базовий функціонал серверної частини та створено усі необхідні елементи архітектури. Було отримано фінальну версію програми, готову до тестування.

					КНТЕУ 6.050103 07-13.БР	Лист
Изм.	Арквш	№ Докум	Подпись	Дат		27

Висновки та пропозиції

Метою даної дипломної роботи була розробка клієнт-серверного застосування з надання послуг доставки вантажу, яка допоможе будь-якому користувачу легко організовувати і замовляти доставку вантажу.

В ході дослідження теоретичних відомостей була переглянута інформація по різних засобах розробки серверної частини та додаткові бібліотеки для покращення ефективності й швидкості розробки. На основі досліджень було сформовано перелік технологій за допомогою яких буде написана backend частина.

На основі проведеного аналізу предметної області були описані задачі та створено технічне завдання для проекту.

Під час розробки концептуальної моделі було створено схему зв'язків між таблицями. Згодом було розроблено логічну модель даних. В кінці отримано готову фізичну модель. На основі отриманих результатів було створено власну базу даних з 7 таблицями, які об'єднані між собою різноманітними зв'язками.

В фінальному етапі розробки програмного забезпечення створено клієнтську частину для Web-додатку служби доставки вантажів логістичного підприємства, розроблено базовий функціонал та створено усі необхідні елементи інтерфейсу. Було отримано фінальну версію програми, готову до тестування.

Дана розробка відповідає усім сучасним стандартам, використовувався широкий спектр технологій і було застосовано та вивчено на практиці багато навичок.

					КНТЕУ 6.050103 07-13.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В.				ВП	28	29
Керівник		Криворучко О.В.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Мосійчук Є.В.						
					Висновки та пропозиції			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основний:

1. Ian Dees. Java // Seven More Languages in Seven Weeks. Languages That Are Shaping the Future / Bruce Tate, Fred Daoud, Jack Moffitt, Ian Dees. — The Pragmatic Bookshelf, 2015. — С. 1—48. — 320 с. — ISBN 978-1941222157.
2. Roberto Ierusalimsky. Programming in Java. — 3-nd ed.. — 2012. — ISBN 9788590379850.
3. Programming Java / Zigurd Mednieks, G. Blake Meike, Лейрд Дорнін, Masumi Nakamura. — O'Reilly, 2011. - 736 с.

Інтернет ресурси:

1. Документація Spring MVC [Електронний ресурс] – Режим доступу: <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html>
2. Документація Spring MVC [Електронний ресурс] – Режим доступу: <https://spring.io/docs>
3. Документація Hibernate [Електронний ресурс] – Режим доступу: <https://hibernate.org/orm/documentation/5.4/>
4. RESTful API Designing guidelines [Електронний ресурс] – Режим доступу: <https://hackernoon.com/restful-api-designing-guidelines-the-best-practices-60e1d954e7c9>
5. Spring Security Taglibs [Електронний ресурс] – Режим доступу: <https://www.baeldung.com/spring-security-taglibs>

					КНТЕУ 6.050103 07-13.БР			
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка створення клієнт-серверного додатку для служби доставки вантажів логістичного підприємства	Стадія	Аркуш	Акрушів
Зав. Каф.		Криворучко О.В				СП	29	29
Керівник		Криворучко О.В.				Факультет обліку, аудиту та інформаційних систем, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розробив		Мосійчук Є.В.						
					Список використаних джерел			