

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
Факультет інформаційних технологій

УДК 004.94:37.091.212.2

**«ПОГОДЖЕНО»**  
Дека́н факультету  
інформаційних технологій

**«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»**  
Завідувач кафедри комп'ютерних наук  
Голуб Б.Л., к.т.н., доцент

Глазунова О.Г., д.п.н., професор

2021 р. «30» листопада 2021 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему Інтелектуальна система підтримки торгівельної  
діяльності

Спеціальність 122 Комп'ютерні науки

(код і назва)

Освітня програма Інформаційні управляючі системи і технології

(назва)

Орієнтація освітньої програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

**Гарант освітньої програми «Інформаційні управляючі системи і технології»**

доктор техн. н., доцент

(науковий ступінь та вчене звання)

Бондаренко В.Є.

(підпис)

(ПІБ)

**Керівник магістерської кваліфікаційної роботи**

канд. техн. наук, доцент

(науковий ступінь та вчене звання)

погоджено

(підпис)

Дудник А.О.

(ПІБ)

**Виконав**

Недбайло Д.А.

НУБІП (ПДПС) **КІЇВ-2021** (ПІБ студента) України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
Факультет інформаційних технологій  
ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук \_\_\_\_\_

к.т.н., доцент Голуб Б.Д.  
(науковий ступінь, вчене звання) (підпис) (ПІБ)  
«29» жовтня 2020 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Недбайлу Дмитру Андрійовичу

(прізвище, ім'я, по батькові)

Спеціальність 122 «Комп'ютерні науки»

Освітня програма Інформаційні управляючі системи і технології

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Інтелектуальна система підтримки  
торгівельної діяльності

затверджена наказом ректора НУБіП України від «29» жовтня 2020 р. №1634 «С»

Термін подання завершеної роботи на кафедру 30 листопада 2021

р.

Вихідні дані до магістерської кваліфікаційної роботи \_\_\_\_\_

пр.

Перелік питань, що підлягають дослідженню:

№ з/п	Питання, що підлягає дослідженню	Строк виконання	Примітка
1.	Системний аналіз предметної області.	01.09.2021	
2.	Моделювання системи	10.09.2021	
3.	Розробка системи	17.09.2021	
4.	Результати дослідження	29.10.2021	
5.	Попередній захист	30.11.2021	
6.	Захист	14.12.2021	

Дата видачі завдання «29» жовтня 2020 р.

Керівник магістерської кваліфікаційної роботи \_\_\_\_\_

Дудник А.О.

(підпис)

(прізвище та ініціали)

Завдання прийняв до виконання \_\_\_\_\_

Недбайло Д.А.

(підпис)

(прізвище та ініціали студента)

# НУВБІП України

## ВСТУП

*Актуальність теми.* Зараз ця тема є дуже актуальною оскільки все більше і більше людей створюють свої приватні підприємства, не маючи при цьому розуміння роботи ринку, економіки і статистики. Ринок збуту це масштабний механізм, який отримує більшого розвитку з кожним днем. Тим самим був зроблений аналіз найперспективнішого напрямку розвитку і отримані результати, які висвітлюють тему табаку у найкращому вигляді. Звичайно мова йде про економічну вигоду і з кожним роком вона тільки зростає. На 2019 - 2020 роки, частка ВВП України, яку складає дохід від акцизи табаку виросла на 0.4% і скадала 5.6%

*Об'єктом дослідження* є інформаційна система для підтримки продажу.

*Предметом дослідження* є методи та алгоритми робочих процесів користувачів системи, що реалізується.

*Метою дослідження* є проаналізувати вимоги для реалізації інформаційної системи підтримки продажу.

Для написання даної роботи були використані такі та наукові джерела: інтернет-статті, електронні архіви.

*Практичне значення дослідження* полягає у надання користувачеві змогу купувати товар з будь-якої точки світу, в свою чергу власнику надати змогу формувати потрібну статистику, звітність і теоретичне прогнозування.

*Наукова новизна проекту*, на мою думку, проявляється у розроблених програмних модулях інформаційної системи, а саме:

НУБІП України

- Розроблено механізм імпорту даних користувачів та їх замовлень в окремий звіт, одразу на сторінці власника.

- Модуль C# скриптів для генерації унікального прогнозування можливого попиту на товар у наступному кварталі, який комплексно

НУБІП України

сформований на основі минулих вподобань користувачів, а також на основі зовнішніх факторів.

- Архітектура системи в цілому розроблена таким чином, що є розширюваною. Загальний та спеціальний функціонал винесено в

НУБІП України

окремі модулі, які можна за бажанням підключати чи відключати. Такі механізми як, наприклад, HTML-хелпери розроблені

універсально і для створення веб сторінки з полями-атрибутами БД

достатньо просто використати необхідний помічник .

НУБІП України

Дана робота має таку *структуру*: вступ, 4 розділи, висновки до розділів, висновки та список використаних джерел а також розділ Додатки.

НУБІП України

НУБІП України

НУБІП України

# НУБІП України

## 1 РОЗРОБКА СИСТЕМИ

### 1.1 Загальні характеристики інтернет-магазинів

На сьогоднішній день неможливо уявити світ без інтернету, ми все робимо використовуючи світову мережу, дивимось фільми, спілкуємось, навіть читаємо, і все це в «он-лайн». Згадайте, коли ви останній раз купували диск, або платівку щоб послухати музичний твір, так це було дуже давно. Більш того, ми замовляємо собі їжу, використовуючи при цьому свій браузер, а якщо конкретніше, то інтернет-магазин.

Інтернет-магазин - це web-сайт, що забезпечує продажі через Інтернет з використанням електронного каталогу або іншого способу представлення продукції.

Безумовно, зараз, під час максимальної діджиталізації, інтернет-магазини тільки нарощують свої клієнтські бази, і це не дивно, адже такі значні і фундаментальні речі можна знайти у перевагах «он-лайн» сервісів над іншими.

Кількість товарів, що пропонуються і продаються на одному сервері, може становити від декількох одиниць до десятків тисяч. Сама оплата товару здійснюється за допомогою інтерактивних платіжних систем, банківського переказу, поштового переказу, під час доставки товару покупцеві, у вигляді післяплати тощо. Інтернет-магазин допомагає клієнту отримати всю можливу інформацію про товари, а також не обтяжує покупця консультантами, які, зазвичай тільки заважають спокійно оцінювати переваги та недоліки того чи іншого продукту. Також варто зазначити, що за рахунок використання

комп'ютерних технологій стало можливим персоналізувати підхід до кожного покупця через дані історії його відвідувань магазину та зроблених раніше покупок.[1]

Серед найвідоміших світових інтернет-магазинів можна виділити такі, як Amazon, eBay, Rozetka, Aliexpress тощо.

### 1.2 Сфера кальянів у мережі інтернет

Кальян – це прилад для куріння східного походження, що представляє собою конструкції з ряду елементів. В апарат входять такі компоненти як колба, шахта, чаша, паланг та тарілка. Щодо виникнення кальянів, то найбільш ймовірна батьківщина кальяну – Індія. Місцем, звідки прийшли ці курильні пристрої також може бути Персія.

Раніше кальян використовувався східними народами для лікування. Люди заправляли в чашу суміш цілющих трав замість тютюну, тому сеанс куріння надавав знеболюючу дію на організм.

Зараз можна побачити кальян у кожному другому кафе або ресторані, більш того, найбільш популярними є місця, які спеціалізуються саме на створенні розслаблюючої атмосфери, із розслаблюючою музикою, гарячим чаєм і настільними іграми, а кальян якнайкраще вписується в такі заклади, і доповнює атмосферу густим і пахучим димом.

В ході розробки було досліджено декілька таких онлайн-магазинів, як dumok, shishastore, amushop. Для прикладу системи було обрано amushop, так як його панель інтерфейсу є дуже зручною, а функціоналу достатньо для обрання потрібного товару. Проте існує велике різноманіття зайвих функцій, які не потрібні користувачу, тому мій проєкт буде мати мінімум зайвого і максимум потрібного користувачу.[2]

### 1.3 Основні завдання виконання дипломного проєкту

Мета створення даного проекту полягає у створенні доступного додатку для клієнтів, які хочуть придбати кальян або його складові. Сьогодні дуже мало тих, хто дійсно знає про види кальянів, їх різницю і ціну, але цей додаток дозволить дізнатися все це за лічені хвилини, і одразу зробити замовлення, обравши той, що підходить саме вам.

Більш того система дає можливість відфільтрувати всі товари за вашим побажанням, що спрощує пошук бажаного товару. Також, після замовлення і отримання товару, кожен має змогу залишити свій коментар, як про конкретний товар, так і про додаток в цілому. Я хочу, щоб кожна людина мала змогу замовити собі і спробувати, обрала свій кальян, дізналась про цінову політику, мундштуки, вуглі, тощо. Кожен клієнт має залишитися задоволеним. [3]

Для розробки інформаційної системи треба виконати такі завдання:

- забезпечити надійне збереження інформації;
- надати доступ користувачеві до інформації;
- представляти актуальну інформацію на момент її отримання.

З метою забезпечення управління процесом продаж кальянів і їх складових, необхідно виконати наступні завдання:

- розробити адміністративну панель;
- розробити функцію авторизації;
- розробити панелі для редагування даних;
- розробити панель для керування замовленнями;
- розробити панель фільтрації товарів для зручного відбору.

#### 1.4 Аналіз системи



На початку аналізу системи визначимо основні об'єкти у системі. Предметом системи є продаж кальянів та їх складових через інтернет-магазин. Виходячи з предмету можна виділити два об'єкта у системі – це:

- Клієнт.
- Продавець.

Ці об'єкти є учасниками процесу продажу. Для опису об'єктів, що визначить тип, відповідальність, критерій успіху створюється профіль об'єктів.

Профілі об'єктів представленні у таблиці 1 та 2.

Профіль продавця.

Таблиця 1

Об'єкт	Продавець
Опис	Приймає замовлення та виконує дії для виконання замовлення
Тип	Користувач
Відповідальність	Виконання замовлення
Критерій успіху	Всі прийняті замовлення оброблені

Профіль клієнта

Таблиця 2

Об'єкт	Клієнт
Опис	Користувач, що обирає потрібний товар, замовляє його, оплачує та отримує

Тип	Користувач
Відповідальність	Правильне оформлення замовлення
Критерій успіху	Отримання замовлення

Процес продажу можна описати наступним чином. Клієнт обирає необхідні товари та замовляє їх. Під час оформлення замовлення клієнт повинен правильно вказати дані про себе. Основні дані для ідентифікації користувача наступні:

- ім'я;
- прізвище;
- номер телефону;
- електронна пошта.

Коли клієнт робить замовлення, продавець може побачити у своєму кабінеті що і коли замовив користувач. Спочатку клієнт визначає деталі замовлення, а саме кількість обраного товару, спосіб оплати, місце та спосіб доставки. Після погодження замовлення продавець готує видачу по замовленню.

Клієнт оплачує замовлення згідно з обраним на етапі погодження способом. Якщо замовлення оплачене продавець його видає. Таким чином процес продажу кальянів стає виконаним.

У процесі системного аналізу предметної області було визначено основні об'єкти та їх взаємодію між собою у процесі виконання продажу кальянів. Наступним кроком є моделювання системи, побудована на основі проведеного аналізу предметної області.

# НУБІП України

## 2 МОДЕЛЮВАННЯ СИСТЕМИ

### 2.1 Моделювання архітектури

Термін "моделювання" спочатку був введений для дослідження проблем, які не вдавалося відразу вирішити теоретичним або експериментальним методом. "Моделювання" - дослідження фізичних процесів на моделях. У найпростішому випадку модель відтворює досліджуване явище (оригінал) із збереженням його фізичної природи і геометричної подоби, а відрізняється від оригіналу лише тим (розмірами, швидкістю течії досліджуваних явищ і іноді матеріалом), що призводить до зміни ... параметрів".

Оскільки буде проводитися моделювання системи, що не є фізичним об'єктом, то така модель буде складатися з певної послідовності діаграм. Ці діаграми відобразатимуть роботу системи з різних сторін. Тобто ці діаграма і є певними моделями системи. [15]

При моделювання системи використовуватимуться UML-діаграми. UML - це спосіб візуалізації програмного забезпечення за допомогою низки певних діаграм. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. Сьогодні UML прийнята групою управління об'єктами, як стандарт для моделювання розробки програмного забезпечення.

Всі діаграми UML можна умовно розбити на дві групи, перша з яких - спеціальні діаграми. Загальні діаграми практично не залежать від предмета моделювання і можуть застосовуватися в будь-якому проекті без оглядки на предметну область, область рішень і т.д. [11]

# НУБІП України

При моделюванні системи будуть розроблені такі діаграми:

- прецедентів;
- послідовності;
- станів;
- розміщення;
- активності.

# НУБІП України

Діаграма прецедентів - це діаграма, на якій висвітлено усі можливі відношення між акторами(діючі ланки системи) і прецедентами. Діаграма прецедентів покликана відповісти на головне питання моделювання: що робить система в зовнішньому світі?

# НУБІП України

На цій діаграмі застосовуються два типи основних сутностей: прецеденти та дійові особи, між якими встановлюються такі основні типи відносин:

- асоціація між дійовою особою і прецедентом;
- узагальнення між діючими особами;
- узагальнення між прецедентами;
- залежності (різних типів) між прецедентами.

# НУБІП України

Діаграма прецедентів моделі системи представлено на рис 1. Згідно з основними об'єктами, які беруть участь у процесі продажу, дійовими особами у системі є: «Клієнт», «Продавець».

# НУБІП України

Клієнт взаємодіє з такими прецедентами:

- «Авторизація»;
- «Обрати товар»;
- «Обрати категорію товару»;
- «Коментування товару»;
- «Формування замовлення»;
- «Реалізація замовлення»;
- «Підтвердження замовлення»;
- «Перегляд товарів»;
- «Оцінювання товару»;

# НУБІП України

# НУБІП України

● «Вибір способу оплати»;  
 ● «Оплата»;  
 Продавець у свою чергу взаємодіє з:

● «Редагування характеристик товар»;  
 ● «Додати товар»;  
 ● «Формування рейтингу товару»;  
 ● «Формування фінансових звітів».

Також існує взаємодія прецедентів між собою. «Формування замовлення» включає в себе прецедент «Підтвердження замовлення» «Оплата» включає в себе прецедент «Вибір способу оплати». Прецедент «Обрати товар» розширюється прецедентом «Формування замовлення».

Діаграма послідовності - це діаграма, яка показує роботу системи і проходить повний цикл роботи програми, починаючи з самого початку роботи застосування, використовуючи при цьому вхідні повідомлення та цикли. Також в цій діаграмі фігурують усі існуючі і задіяні об'єкти, які використовуються під час роботи з програмою. Ця модель у повному обсязі показує усі нюанси роботи інформаційної системи, її особливості і можливості.

Діаграми послідовності є одним із способів формалізації сценаріїв використання. Її перевага заключається в тому, що на ранніх стадіях опису сценаріїв можливо з'ясувати склад взаємодіючих компонентів та описати потік повідомлень від одних компонентів до інших. Ці компоненти та потоки повідомлень в подальшому будуть трансформовані в конкретні класи (об'єкти), методи цих об'єктів (якщо говорити термінологією мови Java). Відповідно, одразу ж з'ясується і модель системи подій (Actions), які дані класи (об'єкти) будуть підтримувати та обробляти. Більш детально можна ознайомитися на плакаті 4.

Діаграма послідовності моделі системи зображено на рис 1. Тут можна побачити повний цикл виконання програми з усіма уточненнями.

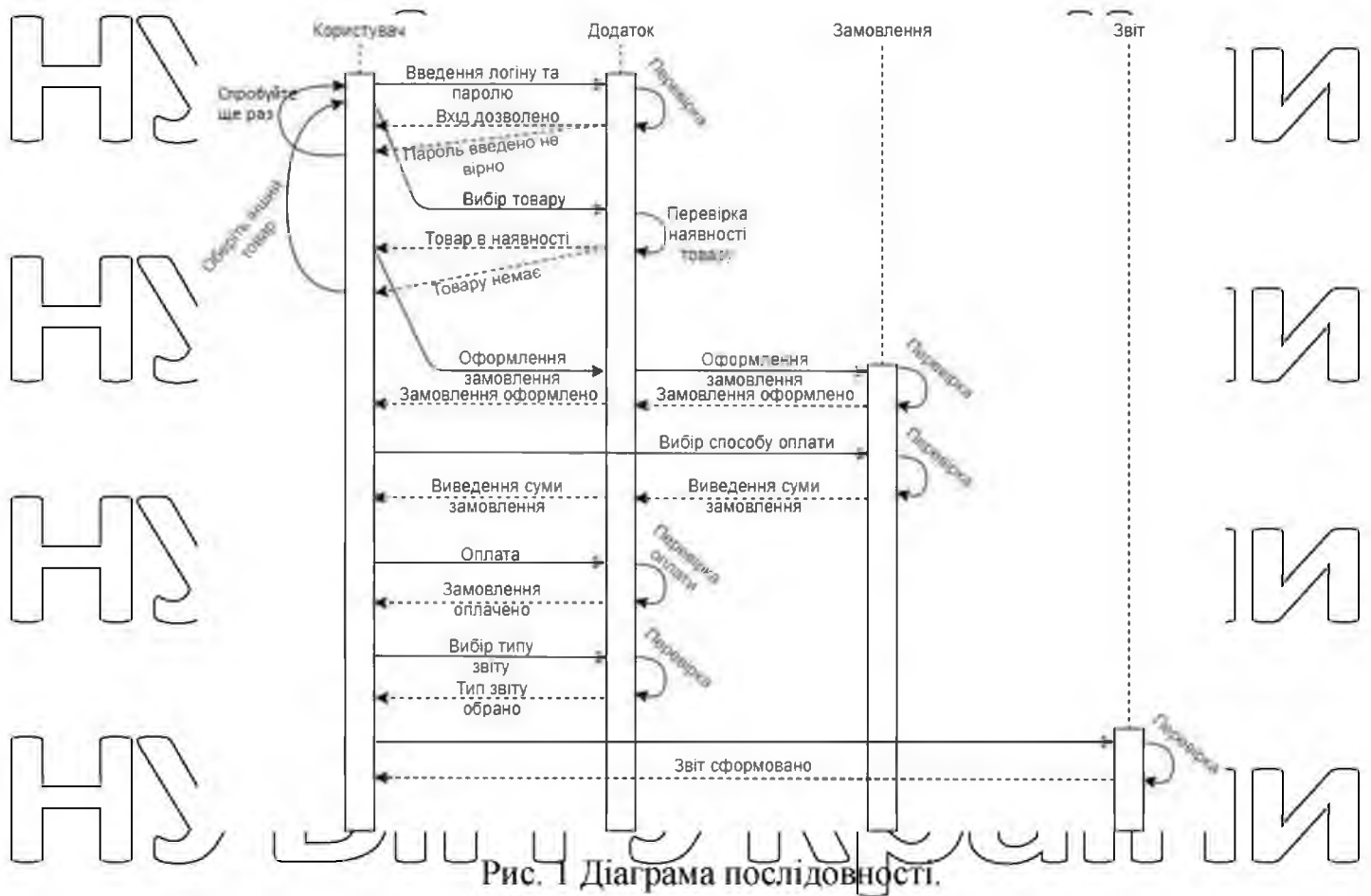


Рис. 1 Діаграма послідовності.

Під час роботи системи її користувач буде виконувати різні функції і дії.

Повна активність користувача будуються у діаграмі діяльності. Діаграма діяльності — діаграма, що показує певну активність того чи іншого об'єкта в рамках даної інформаційної системи, переходи здійснюються після завершення минулої дії, і є наслідком для кожної наступної.

Елементами діаграми є:

- Коло, що позначає початковий стан.
- Коло з маленьким колом всередині, що позначає кінцевий стан (якщо є).
- Округлений прямокутник, що позначає окрему діяльність. Верхня частина прямокутника містить назву стану, в середині може бути горизонтальна лінія, під якою записуються активності, що відбуваються в даному стані.
- Стрілка, що позначає перехід. Назва події (якщо є), що викликає перехід, відзначається над/під стрілкою. Вартовий вираз може бути доданий перед

«/» і укладений у квадратні дужки (назва події), він означає, що перехід відбувається лише за умови істинності виразу. Якщо при переході відбувається якась активність, то воно додається після «/» (назва події).

Побудована діаграма діяльності для процесу покупки товару клієнтом зображена на рис 2.

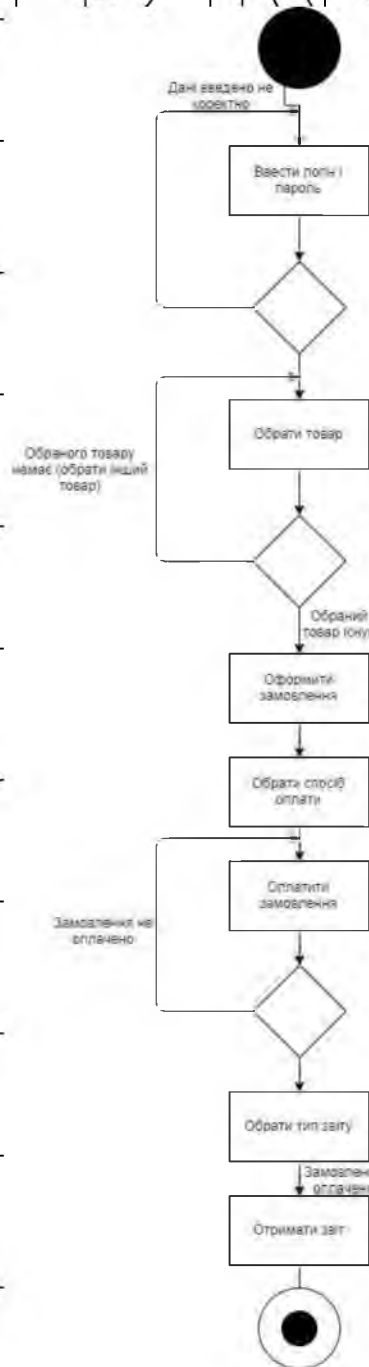


Рис. 2 Діаграма діяльності для процесу покупки товару.

У процесі вибору та покупки товару користувач повинен авторизуватися, а для цього йому потрібно ввести свій логін і пароль, після успішного введення даних, користувач обирає бажаний товар, і якщо він є в наявності то переходить до наступного процесу, яким є оформлення замовлення. Для оформлення замовлення користувачу потрібно підтвердити вибір обраних товарів і обрати спосіб оплати замовлення. Далі користувач обирає тип звіту (чеку) і отримує свій звіт.

Діаграма активності (activity diagram) - спосіб опису поведінки на основі вказівки потоків управління і потоків даних.

Діаграма діяльності - ще один спосіб опису поведінки, який візуально нагадує стару добру блок-схему алгоритму. Однак за рахунок модернізованих позначень, узгоджених з об'єктно-орієнтованим підходом, а головне, за рахунок нової семантичної складової (вільна інтерпретація мереж Петрі), діаграма діяльності UML є потужним засобом для опису поведінки системи.

На діаграмі діяльності застосовують один основний тип сутностей - дія, і один тип відносин - переходи (передачі управління і даних). Також використовуються такі конструкції як розвилки, злиття, з'єднання, розгалуження, які схожі на сутності, але такими насправді не є, а є графічний спосіб зображення деяких окремих випадків багатомісних відносин.

Діаграму активності для процесу оформлення замовлення клієнтом та виконання замовлення продавцем зображено на плакаті 2.

Діаграма розміщення (deployment diagram) поряд з відображенням складу і зв'язків елементів системи показує, як вони фізично розміщені на обчислювальних ресурсах під час виконання.

Перед тим як побудувати діаграму розміщення, спочатку треба визначити архітектуру системи. Існують такі види розподілених архітектур:

- архітектура «файл-сервер»;



# НУБІП України

- архітектура «клієнт-сервер»;
- архітектура веб-додатків.

Файл-серверна архітектура має на увазі наявність виділеного мережного ресурсу для зберігання даних. Такий ресурс називається «файловим сервером».

При такій архітектурі всі функціональні компоненти системи розташовані на користувальницькому комп'ютері, що називається «клієнтом», а самі дані перебувають на сервері. Така організація системи має такі переваги:

# НУБІП України

- багатокористувальницький режим роботи з даними, що зберігаються на сервері;

- централізоване керування правами доступу до загальних даних; низька вартість розробки;

# НУБІП України

- висока швидкість розробки.

Недоліки файл-серверної архітектури:

# НУБІП України

- послідовний доступ до загальних даних і відсутність гарантії їхньої цілісності;

- продуктивність (залежить від продуктивності мережі, клієнта й сервера).

Класичне подання файл-серверної архітектури зображено на рис 3

# НУБІП України



Рис. 3 Файл-серверна архітектура.

Архітектура «клієнт-сервер» являє собою мережеву інфраструктуру, в якій сервери є постачальниками певних сервісів (послуг), а клієнтські комп'ютери

виступають їхніми споживачами. Класичне подання клієнт-серверної архітектури має на увазі наявність у мережі сервера й декільком підключених до нього клієнтів. У таких системах сервер, в основному, відіграє роль

постачальника послуг з використання бази даних. Ця архітектурна модель називається дворівневою (two-tier architecture). Переваги даної архітектури:

- підтримка багатокористувальницької роботи; гарантія цілісності даних;
- наявність механізмів керування правами доступу до ресурсів сервера;
- можливість розподілу функцій між вузлами мережі.

Недоліки:

- вихід з ладу сервера може спричинити непрацездатність всієї системи;
- потреба високого рівня технічного персоналу;
- висока вартість устаткування.

Класичне подання архітектури «клієнт-сервер», зображено на рис 4.



Рис. 4 Архітектура «клієнт-сервер».

Архітектура web-додатків або архітектура web-сервісів має на увазі надання деякого сервісу, доступного в мережі Internet, через спеціальний додаток. Основою для надання таких послуг служать відкриті стандарти й протоколи SOAP, UDDI й WSDL. SOAP (Simple Object Access Protocol) визначає формат запитів до web-сервісів. Дані між клієнтом і сервісом передаються в SOAP-конвертах (envelops). WSDL (Web Service Description Language) служить для опису інтерфейсу надаваного сервісу. Перед розгортанням web-додатка

потрібно скласти його опис, вказати адресу, список підтримуваних протоколів, перелік припустимих операцій, а так само формати запитів і відповідей. UDDI (Universal Description, Discovery and Integration) являє собою протокол пошуку web-сервісів у мережі Internet. Пошук здійснюється за їхніми описами, які розташовані в спеціальному реєстрі. Архітектура таких сервісів схожа за концепцією з багатопанкової клієнт-серверною, однак, сервери додатків баз даних розташовуються в мережі Internet. [6]

У дипломній роботі ми використовуватимемо архітектуру web-додатків. Відповідно до обраної архітектури побудуємо діаграму розміщення системи. Побудована діаграма зображена на рис 5. Більш детально діаграму розгорнення можна побачити на плакаті 1.

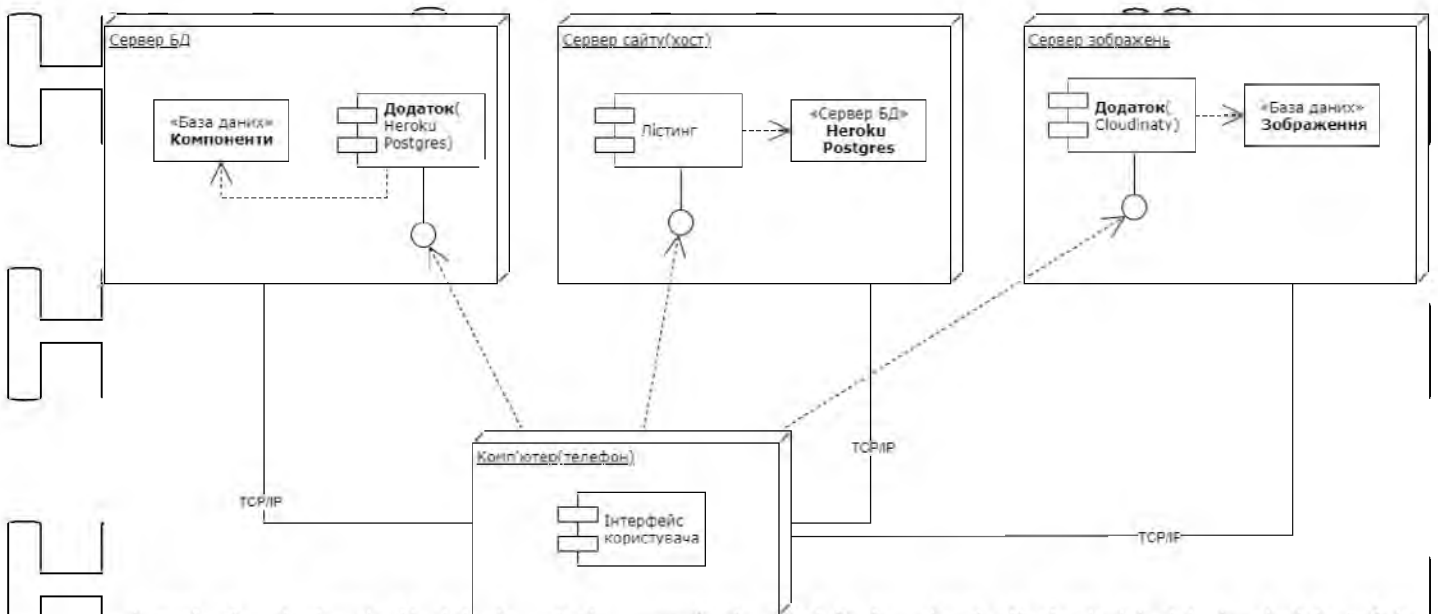


Рис. 5 Діаграма розміщення.

## 3 РОЗРОБКА СИСТЕМИ

### 3.1 Вибір методів та засобів для реалізації інформаційного забезпечення системи

БД (База даних) - сукупність взаємопов'язаних даних, що зберігаються разом даних при наявності такої мінімальної надмірності, яка допускає їх використання оптимальним чином для одного або декількох додатків. Створення бази даних, її підтримка і забезпечення доступу користувачів до неї здійснюється централізовано за допомогою спеціального програмного інструментарію - системи управління базами даних.

СУБД (Система управління базами даних) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

Основні функції СУБД:

- управління даними в зовнішній пам'яті;
- управління буферами оперативної пам'яті;
- управління транзакціями;
- журналізація та відновлення БД після збоїв;
- підтримка мов БД.

Транзакція - послідовність пов'язаних дій, які переводять дану систему з одного стану в інший. Транзакція повинна завершитися повністю, або система повинна бути повернута в початковий стан.

Сутність (entity) - це реальний або уявний тип об'єкта, інформація про який повинна зберігатися і бути доступна. У діаграмах сутність представляється у вигляді прямокутника, що містить ім'я сутності. При цьому ім'я сутності - це ім'я типу, а не деякого конкретного примірника цього типу. Кожен екземпляр сутності (об'єкт) повинен відрізнятися від будь-якого іншого примірника тієї ж сутності.

Сутності мають властивості, які називаються атрибутами (attribute). Будь-який атрибут приймає значення з певної безлічі допустимих значень, що називається доменом атрибуту.

Ключ сутності (entity key), первинний ключ - це атрибут (або безліч атрибутів), що унікальним чином ідентифікує екземпляр сутності (об'єкт).

Зв'язок (relationship) - це графічно зображена асоціація, що встановлюється між двома сутностями. Зв'язок може існувати між двома різними сутностями або між сутністю і нею ж самою (рекурсивний зв'язок). [7]

Серед найбільш популярних СУБД виділяють такі, як MySQL, SQL Server, ORACLE та DB2. Перед тим, як обрати найбільш відповідну для виконання завдання дипломної роботи, ми проаналізували та порівняли їхні характеристики.

SQL Server відрізняється великою кількістю позитивних характеристик. Однак база даних, що розроблятиметься в дипломній роботі, є web-орієнтованою. Зважаючи на те, що SQL Server використовується здебільшого для розробки настільних додатків, ми не можемо обрати цю СУБД.

Іншою СУБД, що розглядалася нами, є Oracle. Серед позитивних характеристик можна виділити такі, як швидка обробка даних, можливість ефективно обробляти велику кількість даних, а також високий рівень масштабування. Також варто зважати на те, що Oracle має й певні недоліки, такі як висока вартість, потреба в якісному потужному обладнанні та додатковому залученні персоналу з високим рівнем спеціалізації для підтримки функціонування баз даних. Проаналізувавши позитивні сторони та недоліки ми прийшли до висновку, що Oracle не підходить для роботи.

Через те, що система, що створюється в рамках роботи над дипломним проектом, є web-орієнтованою, а також знаходиться на хостингу, який дає нам можливість використовувати безкоштовну версію PostgreSQL ми обрали для роботи саме її.

PostgreSQL - це система управління базами даних загального призначення та об'єктно-реляційною системою, найдосконалішою системою баз даних з

відкритим кодом. PostgreSQL був розроблений на базі POSTGRES 4.2 на кафедрі комп'ютерних наук Берклі, Каліфорнійський університет

PostgreSQL був розроблений для роботи на UNIX-подібних платформах.

Однак PostgreSQL також був розроблений як портативний, щоб він міг працювати на різних платформах, таких як Mac OS X, Solaris та Windows.

PostgreSQL - це безкоштовне програмне забезпечення з відкритим кодом. Його вихідний код доступний під ліцензією PostgreSQL, ліберальною ліцензією з

відкритим кодом. Ви можете користуватися, змінювати та поширювати PostgreSQL у будь-якій формі.

PostgreSQL вимагає дуже мінімальних зусиль, що підтримуються через свою стабільність. Тому, якщо ви розробляєте долатки на основі PostgreSQL,

загальна вартість володіння є низькою порівняно з іншими системами управління базами даних. [16]

Перед початком розробки бази даних треба створити її модель. Для цього у спеціальному середовищі ERWIN розробляється ER-діаграма. ER-модель, або

модель «сутність-зв'язок» (з англ. entity-relationship model) - це модель даних, що дає можливість за допомогою узагальнених конструкторних блоків описувати

концептуальні схеми баз даних. Зовнішній вигляд середовища ERWIN зображено на рис 6. Загальна модель бази даних зображена на плакаті 3.

НУБІП України

НУБІП України

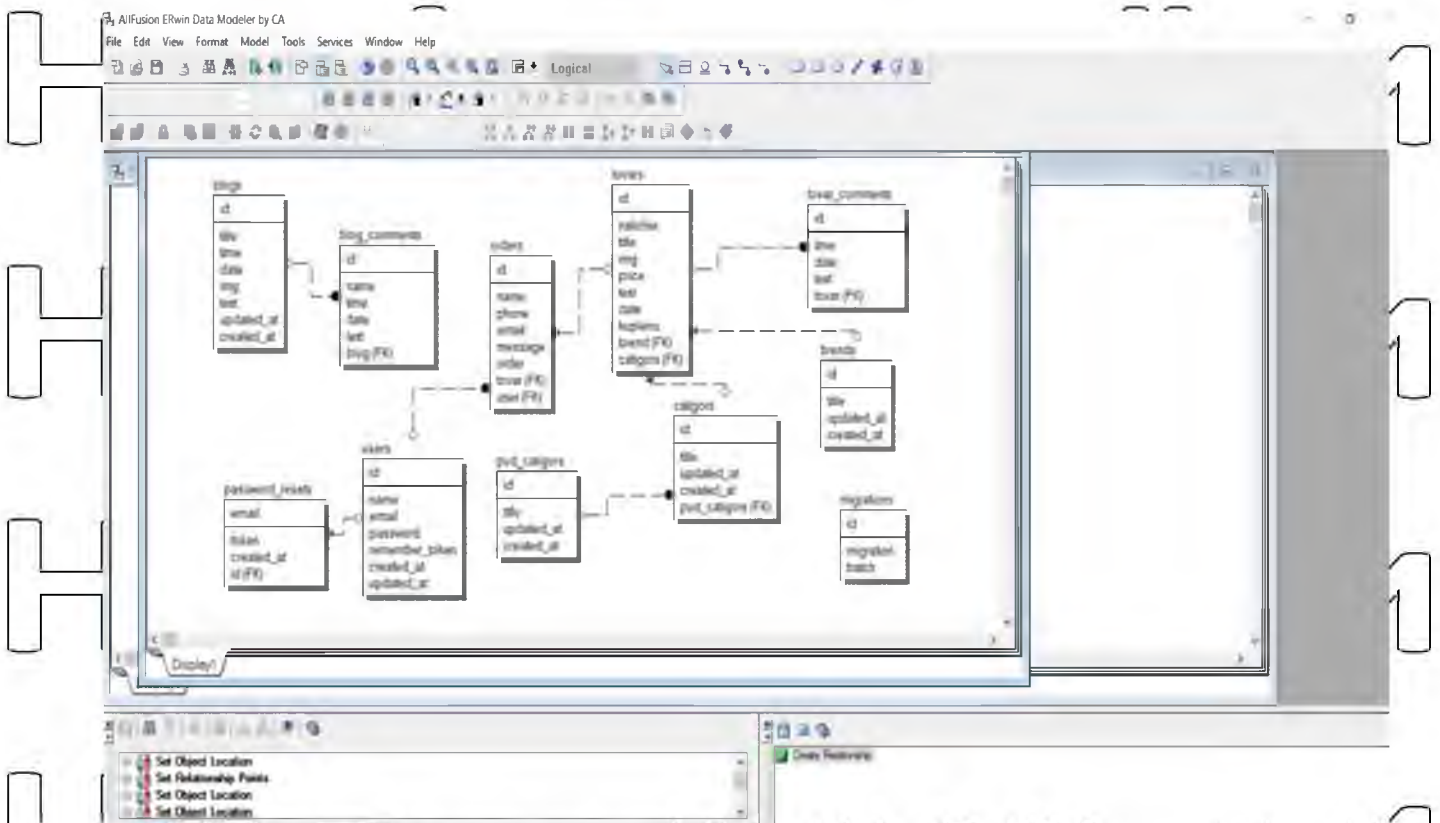


Рис. 6 Середовище ERWIN

### 3.2 Розробка структури інформаційного забезпечення системи

Структурно база даних складатиметься з таблиць даних, які зберігатимуть дані сутностей системи, та зв'язків між ними. Також база даних повинна бути нормалізована до 3 нормальної форми.

Третя нормальна форма (англ. Third normal form; скорочено 3NF) - одна з можливих нормальних форм відносини в реляційній базі даних. 3NF була спочатку сформульована Е. Ф. Коддом в 1971 році. Змінна відношення R знаходиться в 3NF тоді і тільки тоді, коли виконуються наступні умови:

- R знаходиться в другій нормальній формі.
- жоден неключових атрибут R не перебуває у транзитивній функціональній залежності від потенційного ключа R;

Відповідно до потреб системи у базі даних будуть створенні таблиці що зберігатимуть дані про:

НУБІП України

- кальяни, та їх
- Колір

- Країну виробника;

НУБІП України

- Діаметр трубки;
- Коментарі;

НУБІП України

- Коментарі;
- користувачів;
- складові кальянів;

НУБІП України

- замовлення.

Розроблена структура бази даних представлена на плакаті 3.

При створенні бази даних у середовищі «PostgreSQL» спочатку необхідно

зайти на наш хостинг і зареєструватись. Після цього кожен користувач має можливість використовувати сервіс бази даних PostgreSQL безкоштовно, потрібно лише обрати потрібну нам базу даних та натиснути «Provision».

Приклад можна побачити на рис 7.

НУБІП України

НУБІП України



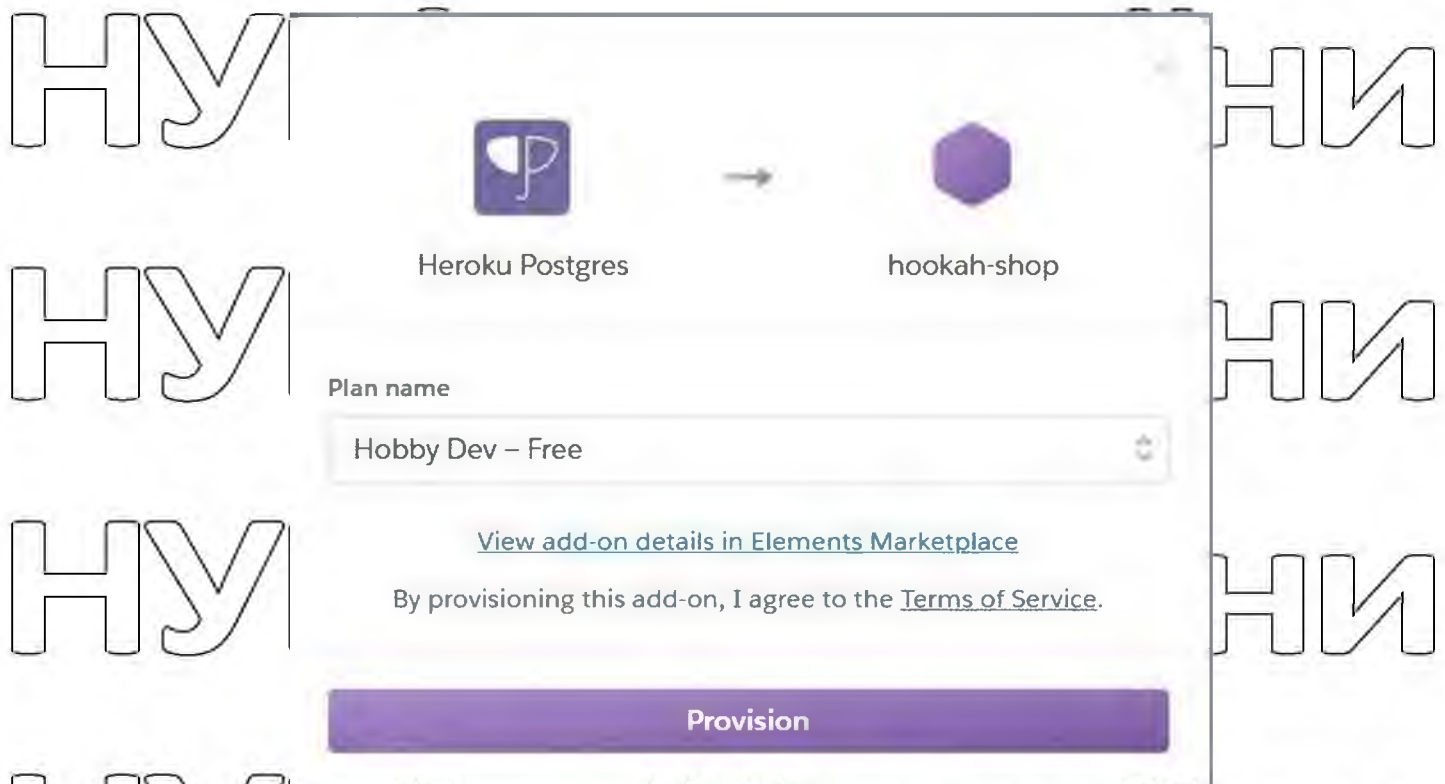


Рис. 7 Створення БД.

### 3.3 Вибір методів та засобів для реалізації прикладного програмного забезпечення

У дипломному проєкті розробляється web-орієнтована система web-архітектури. Тому для того, щоб система працювала необхідно встановити та запустити серверну платформу та обрати програмне середовище.

У якості збірки для розгортання серверної платформи було обрано «Heroku: Cloud Application Platform». Це портативна серверна платформа і програмне середовище, створене спеціально для веб-розробників з урахуванням їх побажань та рекомендацій.

Програмний інструмент має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості з адміністрування та налаштування компонентів.

Платформа широко використовується з метою розробки, налагодження і

тестування веб-проектів, а так само для надання веб-сервісів в локальних мережах.

Хоча спочатку програмні продукти, що входять до складу комплексу, не розроблялись спеціально для роботи один з одним, така зв'язка стала дуже популярною серед користувачів Windows, в першу чергу через те, що вони отримували безкоштовний комплекс програм з надійністю на рівні «Linux» серверів. За час свого існування «Heroku» зарекомендував себе як першокласний і надійний інструмент необхідний кожному розробнику.

Панель керування серверною платформою є зручною та компактною. Зовнішній вигляд панелі зображено на рис 8.



Рис. 82 Зовнішній вигляд панелі керування "Heroku".

Java - це об'єктно-орієнтована мова програмування, розроблений компанією Sun Microsystems (в подальшому придбаній компанією Oracle) Дата офіційного випуску - 23 травня 1995 року.

Програми на Java транслюються в байт-код, який потім виконується віртуальною машиною Java (JVM). JVM - це програма, яка обробляє байтовий код і передає інструкції обладнанню як інтерпретатор. Перевагою такої реалізації є незалежність байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує JVM.

Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною

машиною. Будь-які операції, які перевищують повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання. Також було

використано Java JPA. Java Persistence API (JPA) - це java-специфікація визначає

доступ, збереження і управління даними між Java-об'єктами (Entity класами) і реляційної базою даних. JPA була визначена як частина специфікації EJB 3.0 в якості заміни специфікації EJB 2 CMP Entity Beans. JPA тепер вважається

стандартним галузевим підходом до об'єктно-реляційному відображенню (ORM)

в Java Enterprise Edition (Java EE).

JPA сама по собі є лише специфікацією і визначає набір інтерфейсів вимагають реалізації. Існують комерційні варіанти реалізації JPA і вільні з

відкритим вихідним кодом. Будь-сервер додатків підтримує Java EE повинен забезпечувати і наявність реалізації JPA.

З боку клієнта для розробки веб-сторінок використовуватиметься мова HTML. HTML є найбільш широко використовуваною мовою розмітки

документів, що використовується для написання веб-сторінок. HTML широко

використовується для форматування веб-сторінок за допомогою різних тегів, доступних у мові HTML

Також використовуватиметься CSS - каскадні таблиці стилів, що

використовуються для опису зовнішнього вигляду документа, написаного

мовою розмітки. Зазвичай CSS-стилі використовуються для створення і зміни

стилю елементів веб-сторінок і призначених для користувача інтерфейсів,

написаних мовами HTML і XHTML, але також можуть бути застосовані до будь-

якого виду XML-документа, в тому числі XML, SVG і XUL.

Спеціальним фреймворком зі стандартизованими написаними компонентами, що використовуватиметься для налаштування стилю сторінок, є

Bootstrap. Bootstrap є найбільш популярним HTML, CSS, та JS фреймворком для

зовнішнього вигляду веб-сторінок, він дозволяє будувати чутливі, найперше –

мобільні проекти. Зовнішній вигляд стартової сторінки зображено на рис 9. Для

перевірки правильності введення даних з боку клієнта використовуватимуться JavaScript-розширення.

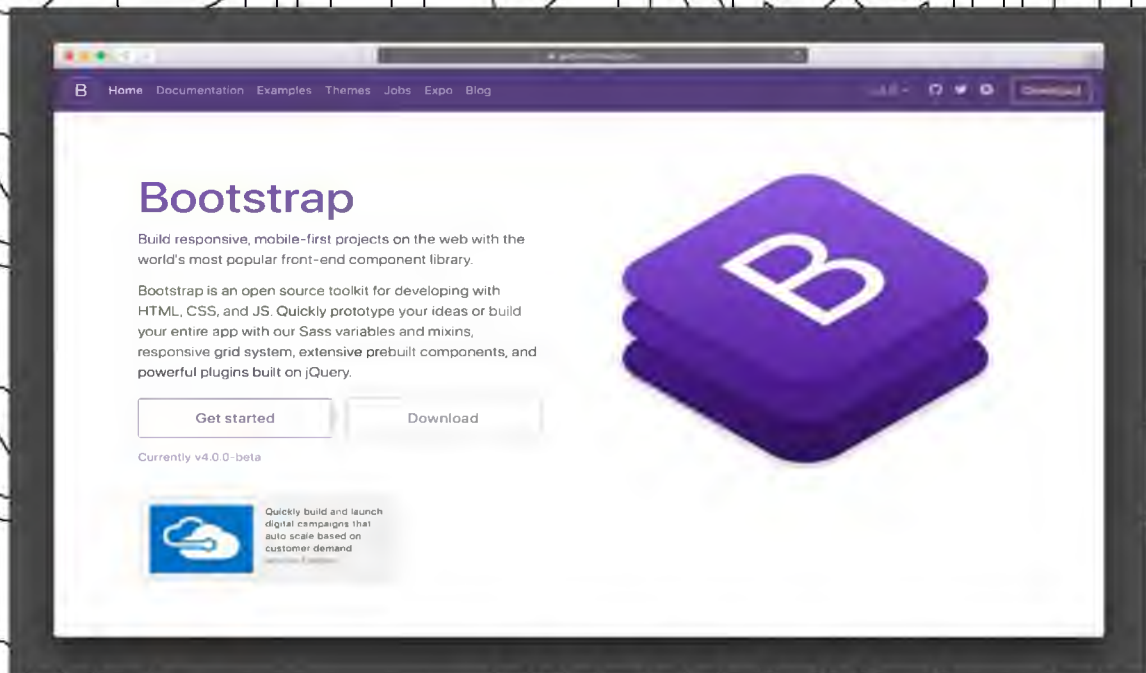


Рис. 9 Стартова сторінка "Bootstrap"

Для полегшення програмування мовою Java використовувався фреймворк «Spring». Spring Framework пропонує комплексну модель програмування та конфігурації сучасних корпоративних Java-програм на будь-якій платформі розгортання.

Ключовим елементом Spring є інфраструктурна підтримка на рівні додатків: Весна фокусується на «сантехніці» корпоративних додатків, щоб команди могли зосередитись на бізнес-логіці на рівні додатків, без зайвих зв'язків із конкретними середовищами розгортання.

Код для файлів сайту можна писати у звичайному текстовому редакторі, та я обрав Visual Studio Code, оскільки це більш сучасна і адаптована програма, яка має додаткові функції, і робить написання потрібної програми якнайбільш зручним. Приклад роботи в цьому середовищі можна побачити на рис 10.

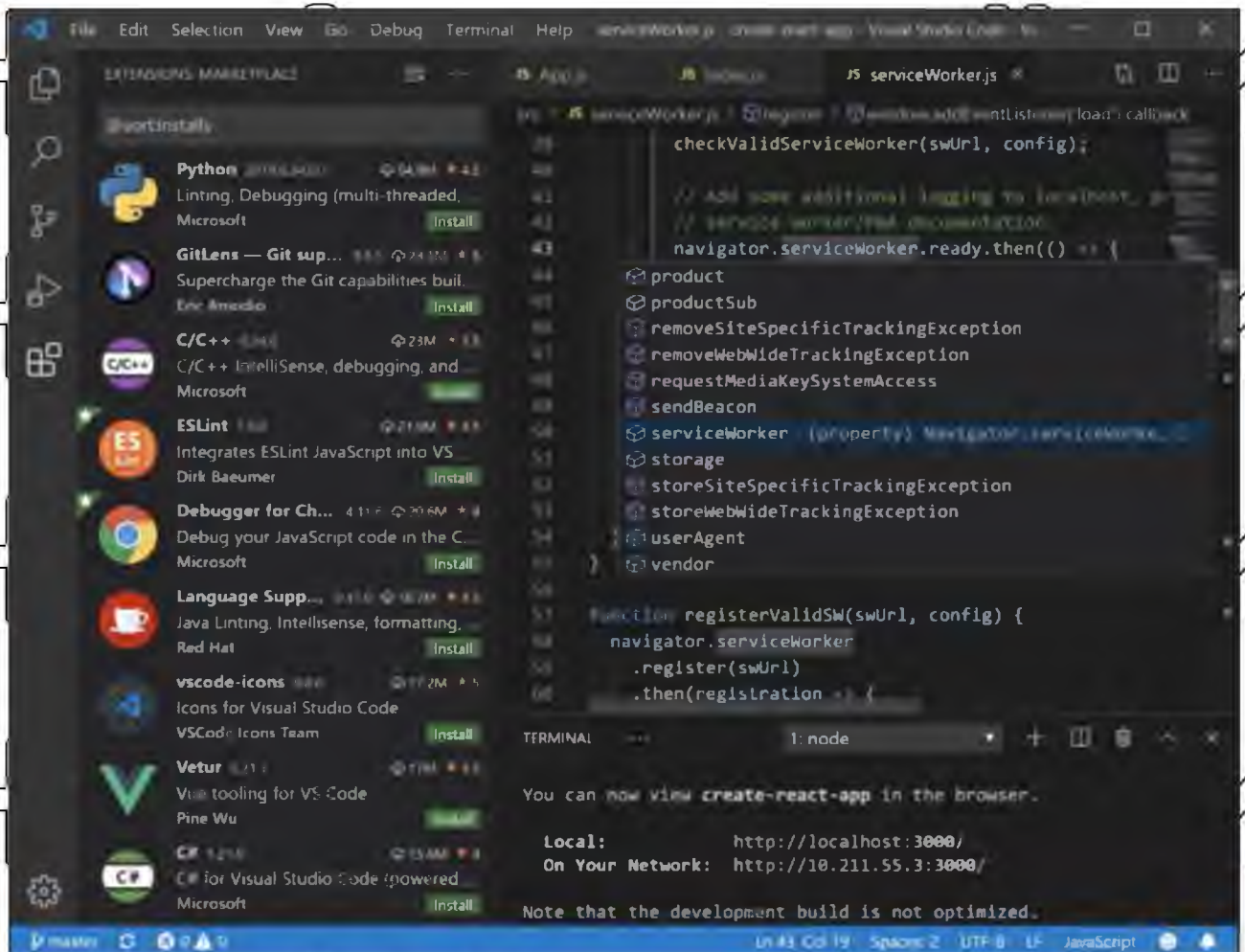


Рис. 103 Редактор "Visual Studio Code".

### 3.4 Розробка та реалізація прикладного програмного забезпечення

Першим кроком у розробці web-орієнтованої системи є створення шаблону зовнішнього вигляду онлайн магазину. Для його створення скористаємося графічним редактором «GIMP». GIMP - багатфункціональний графічний редактор. В основному працює з растровими зображеннями, проте має деякі векторні інструменти.

Розробимо вигляд сторінки перегляду кальянів. Сторінка буде складатися з шапки сторінки, навігаційного меню та основної частини розділеної на два блоки. У лівому блоці будуть виведені списки категорій, підкатегорій та бренду для фільтрації доступних кальянів. Процес створення шаблону зображено на рис 11.

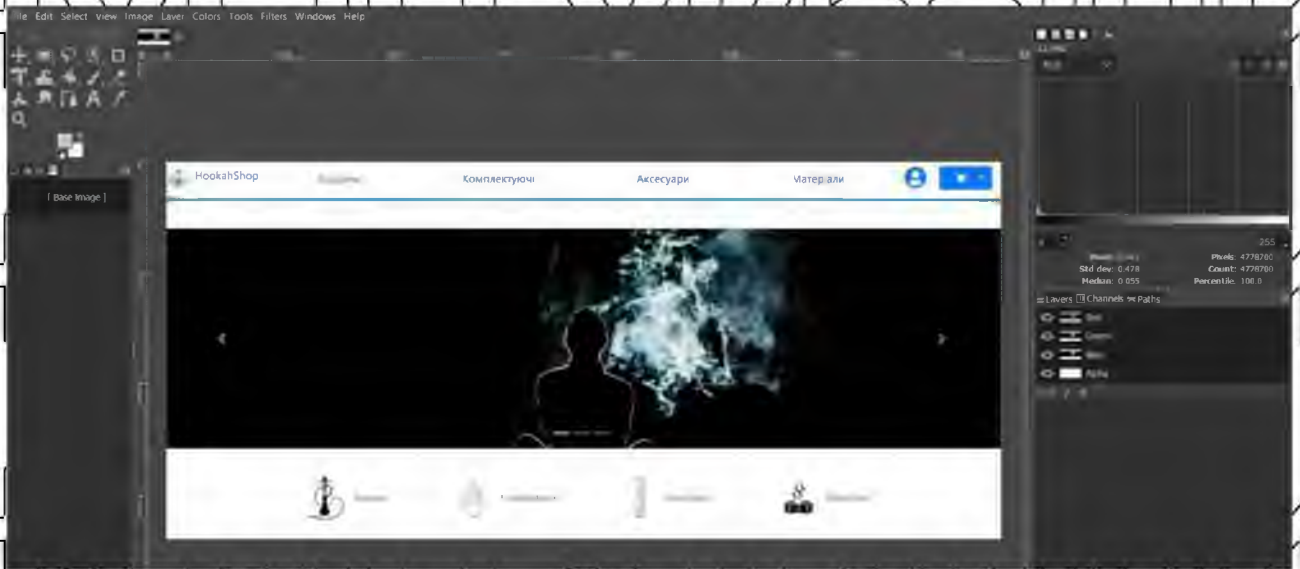


Рис. 11 Процес створення шаблону сторінки перегляду кальянів.

Після створення шаблону сторінки настає етап її верстування. Для цього спочатку треба зробити розмітку сторінки та вивести необхідні дані на неї, використовуючи мову HTML та Java. Написана розмітка представлена у додатку Б. У результаті отримаємо чисту сторінку без стилів. Зовнішній вигляд без стилів представлено на рис 12.



Рис. 12 Зображення сторінки без стилю.

Щоб зовнішній вигляд сторінки відповідав побудованому шаблону необхідно розробити систему стилів «CSS» та під'єднати її до сайту відповідним

рядком. Приклад рядка підключення: «<link href="http://hookahshop/css/main.css" rel="stylesheet">». Лістинг системи стилів представлений у додатку В. У результаті отримуємо потрібну сторінку. Результат роботи представлено на рис

13.

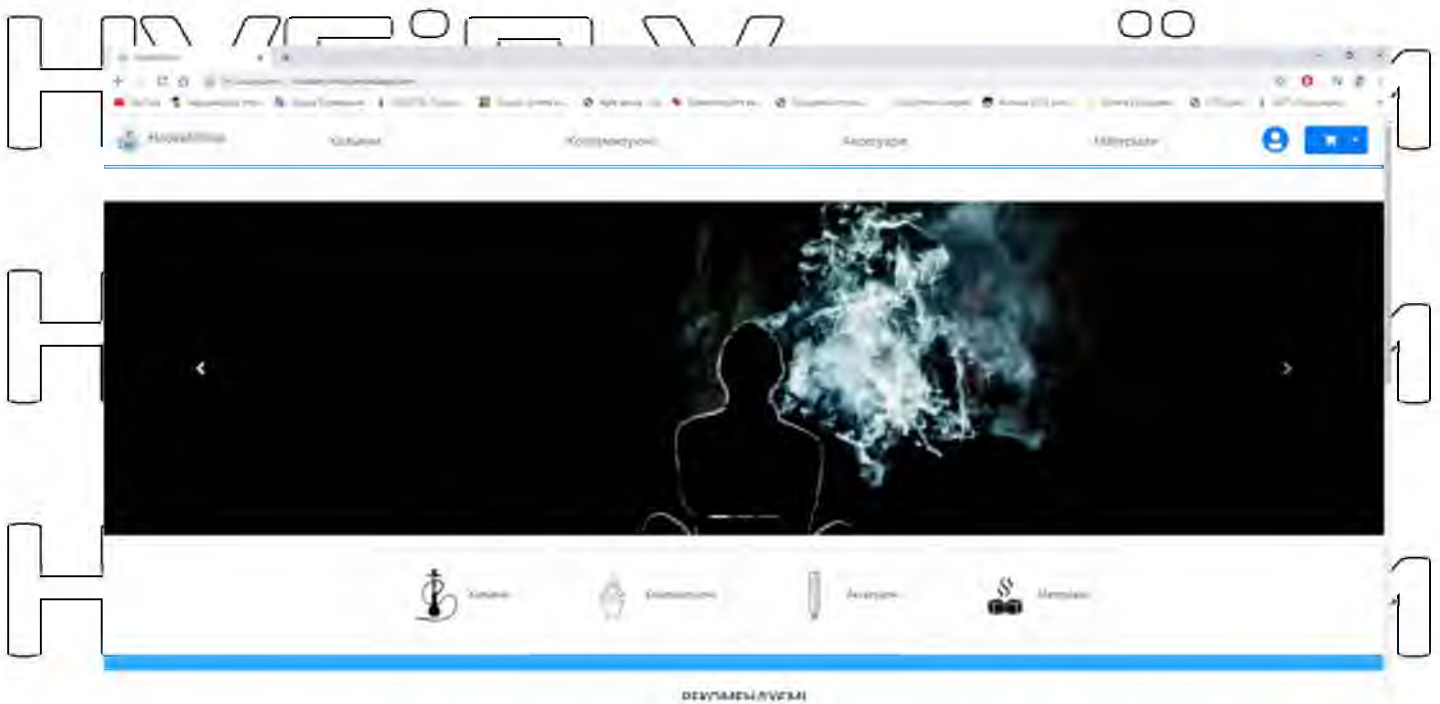


Рис. 13 Розроблена сторінка.

НУБІП України

НУБІП України

НУБІП України

## 4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 4.1 Вимоги до апаратного та програмного забезпечення

Щоб користуватись системою користувачу достатньо мати будь-який обчислювальний прилад, такий як:

- комп'ютер;
- смартфон;
- планшет;
- мобільний телефон;
- ноутбук.

Головною вимогою для користування є лише наявність доступу до інтернету та встановлений будь-який браузер. Ця вимога пов'язана з тим, що система є web-орієнтованою, не потребує попереднього встановлення на клієнтський браузер, і починає працювати при переході за посиланням у браузері.

### 4.2 Склад інсталяційного пакету для встановлення розробленої системи

На стороні клієнта система не потребує додаткових встановлень, лише наявність будь-якого браузера.

Адмін також повинен зайти на сайт, який знаходиться на сервері і зайти з свого облікового запису.

### 4.3 Тестування системи

Дана система знаходиться у вільному доступу, тому кожен користувач має змогу зайти на сайт і провести тестування. Відкривши сайт користувач може продивитися доступні товари, зробити замовлення, залишити коментар.



Протестуємо алгоритм роботи замовлення товару клієнтом на сайті. Коли клієнту заходять на сайт, відкривається сторінка перегляду доступних товарів. Якщо користувач натискає на кнопку «Купити», то обраний товар потрапляє до корзини, рис 14.



Рис. 4 Натискання кнопки «Купити»

Крім того, користувачу доступна фільтрація за наданими характеристиками, рис 15.

NUBIP Україна

NUBIP Україна

НУБІ

Ціна:

Від

1639,0

НУБІ

До

3000,0

НУБІ

Наявність

- Є в наявності
- Немає в наявності

НУБІ

Країна

- США
- Німеччина

НУБІ

Діаметр шахти

- 15мм
- 14мм

НУБІ

Рис. 15 Фільтрація

Відображення кількості доданих товарів до кошика зображено на рис 16.

НУБІП України

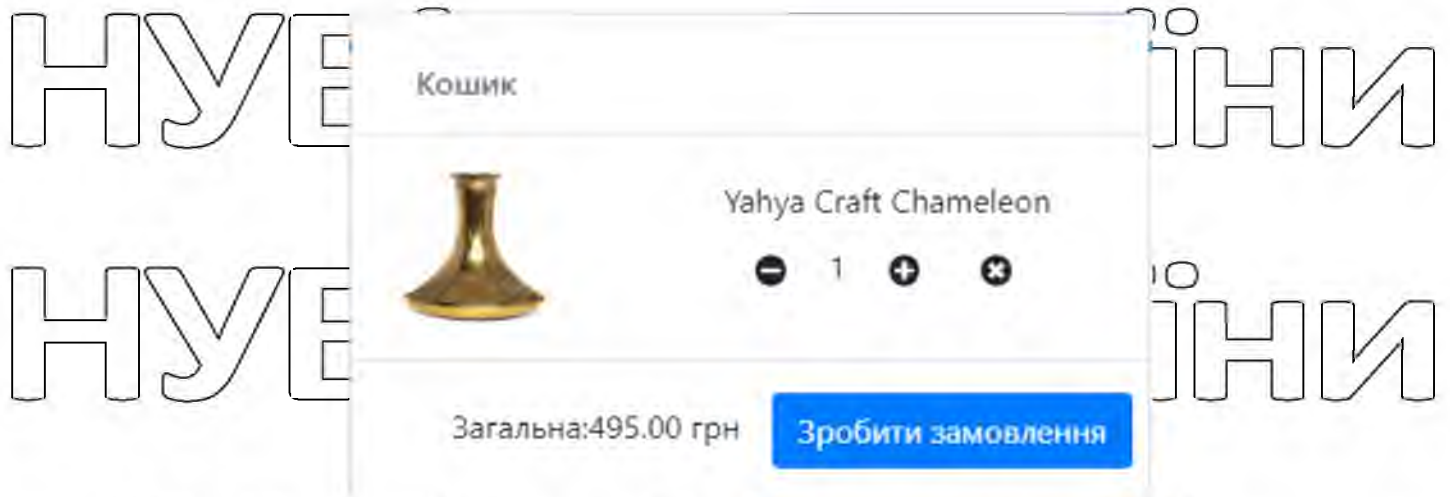


Рис. 16 Відображення кількості товарів у кошику

Для того щоб оформити замовлення клієнт переходить до кошику, рис 17.

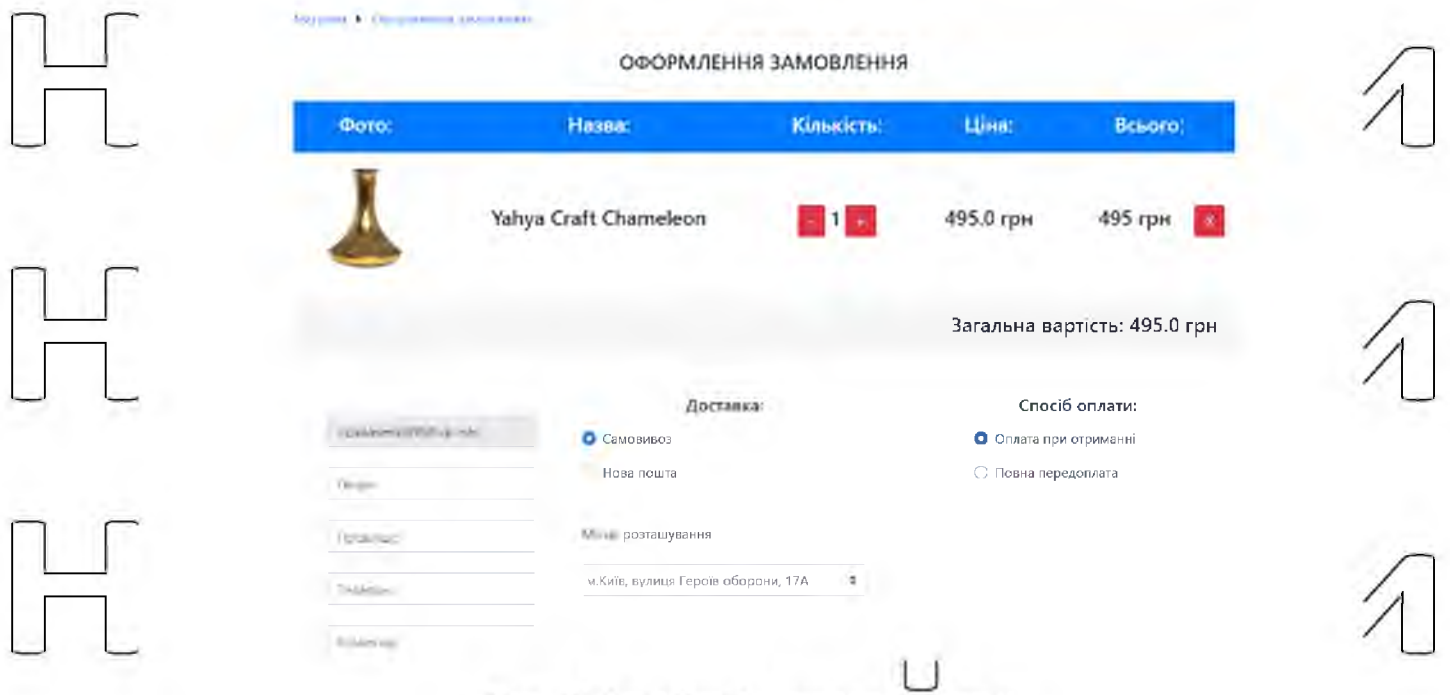


Рис. 17 Зовнішній вигляд кошику.

Далі користувач обирає або місце з якого він буде забирати замовлення або обирає доставку «Новою поштою», де вказує своє місто і відділення...

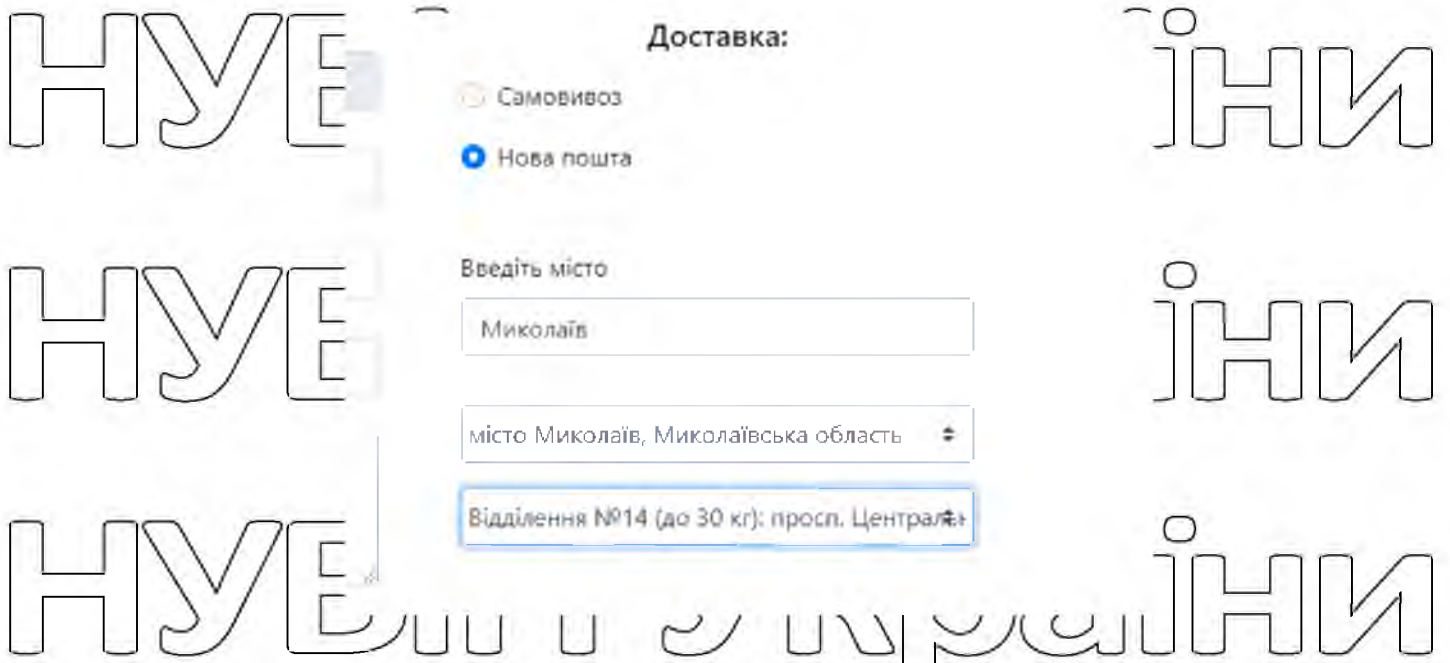


Рис. 185 Вікно вибору типу та місця доставки.

Клієнт повинен ввести дані про себе та натиснути кнопку підтвердити, рис 19.

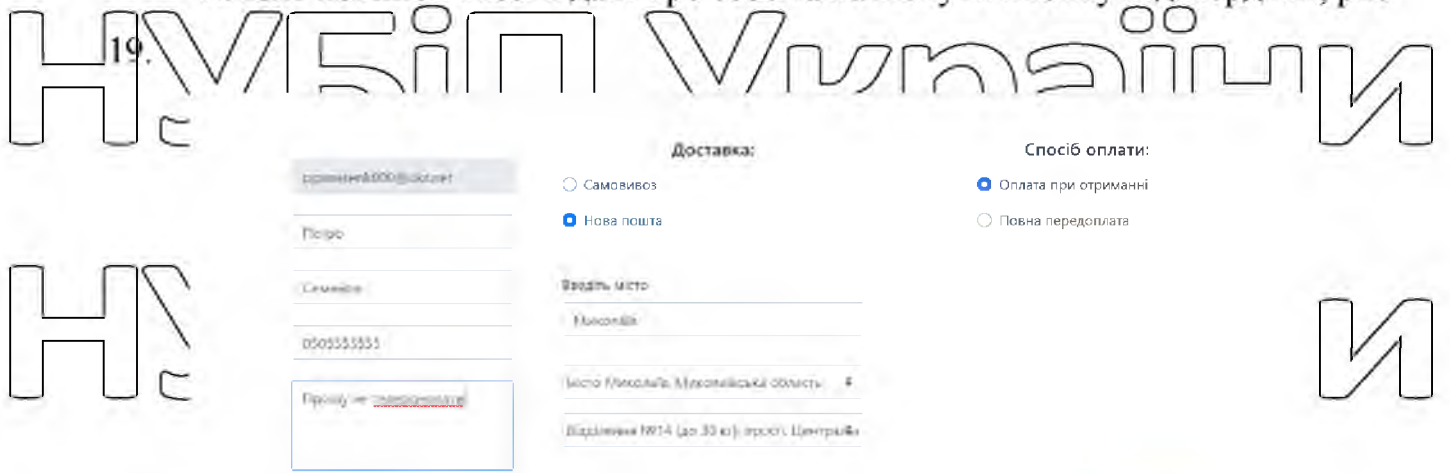
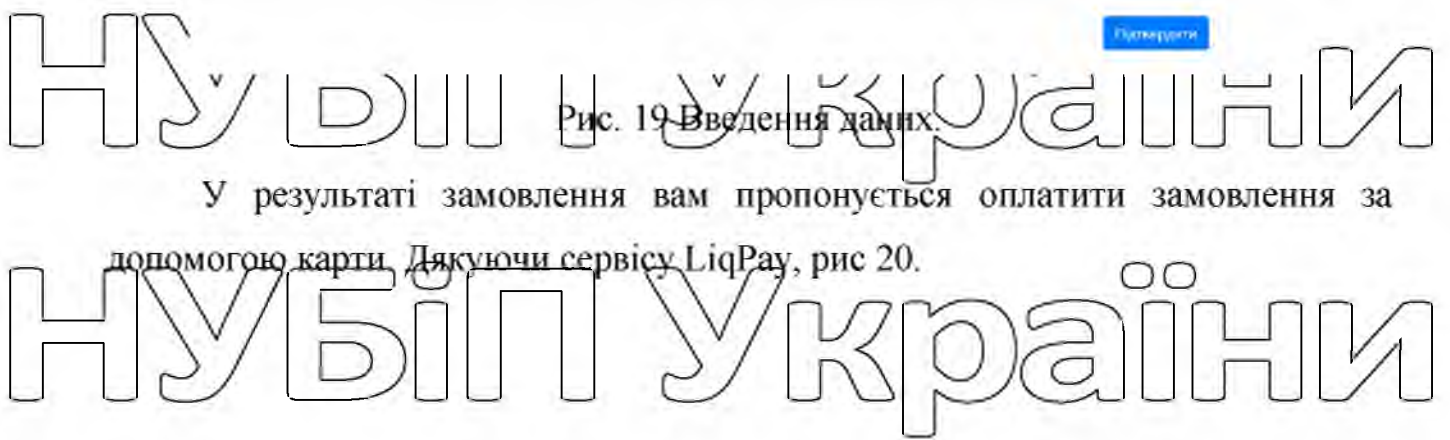



Рис. 19 Введення даних.

У результаті замовлення вам пропонується оплатити замовлення за допомогою карти. Дякуючи сервісу LiqPay, рис 20.



Тестовый режим

QR-код для оплаты



Используйте Privat24

К оплате: **495.00 UAH**  
Замовлення №4 на загальну суму: 495.0 грн

Оплатить через Приват24

**24 Pay**

или

Карта **Другой способ**

Номер карты  
[input type="text"]

Срок действия CVV2  
[input type="text"] [input type="text"]

Нажимая на кнопку «Оплатить», Вы принимаете Пользовательское соглашение

**Оплатить**

Отменить оплату

Рис. 206 Відображення оплати.

Після оплати товару, а отже успішного замовлення, на пошту користувача приходить лист, який сповіщає про зроблене вами замовлення, рис 21.

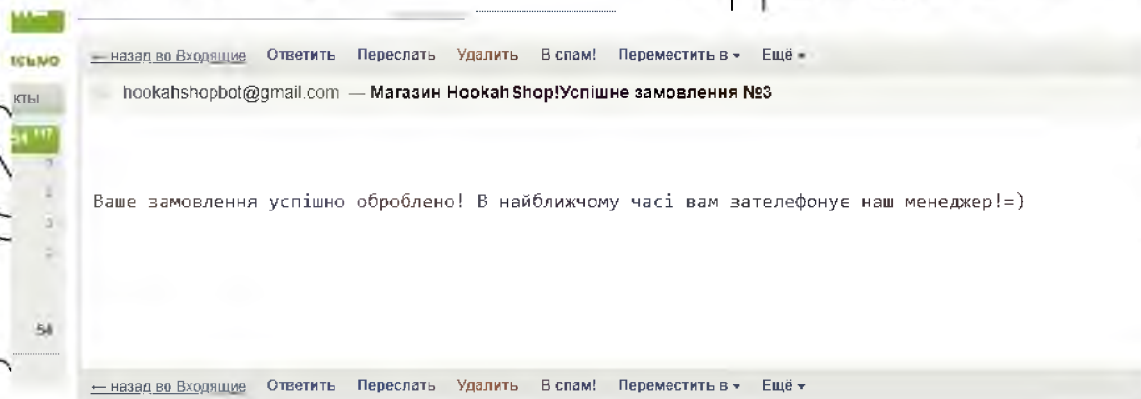


Рис. 21 Лист сповіщення.

Після успішного замовлення користувач має змогу залишити свій коментар, а також поставити свою оцінку тому чи іншому товару, попередньо пройшовши перевірку на не робота, в свою чергу продавець має змогу видалити небажані коментарі, рис 22.

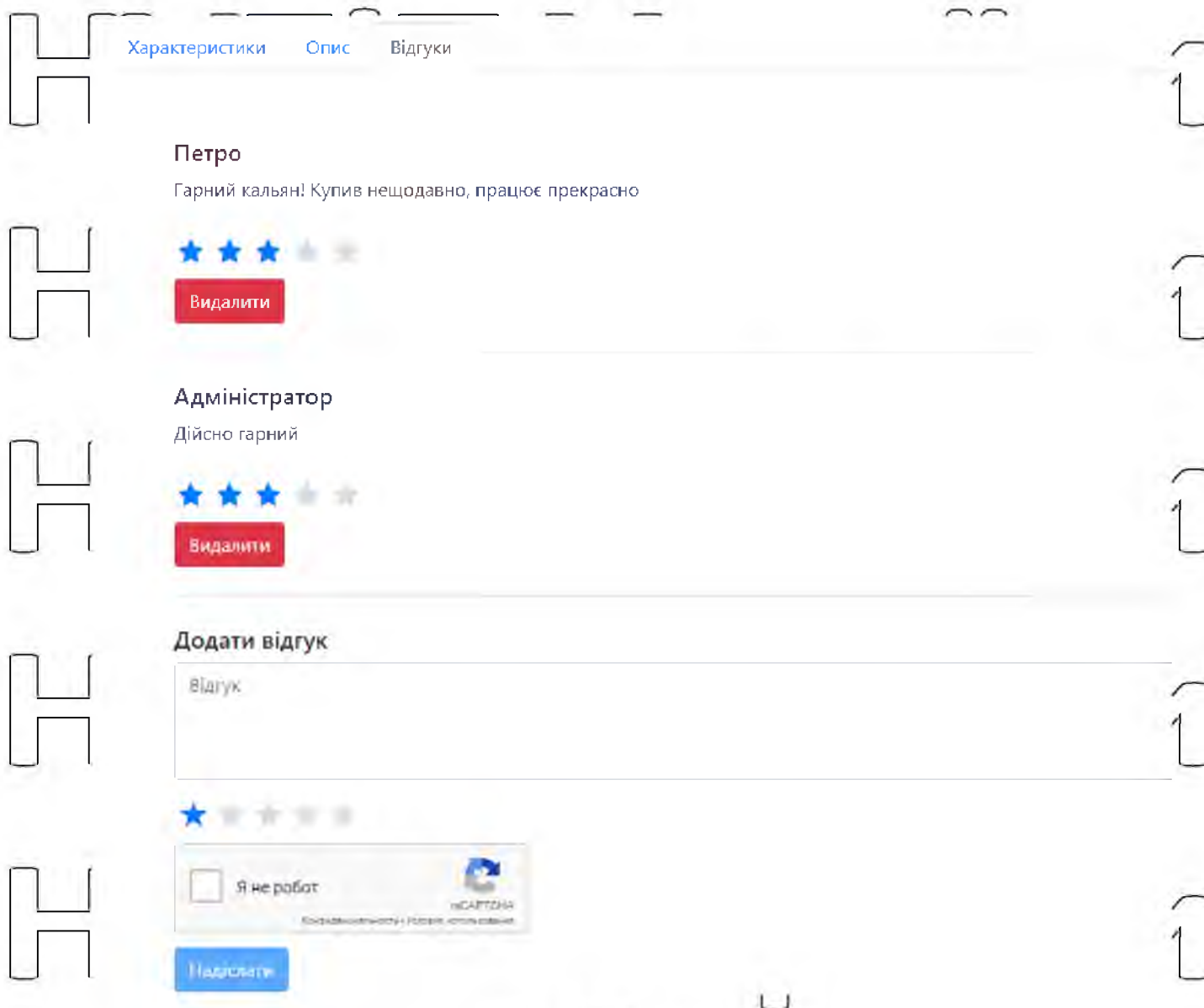


Рис. 22 Коментарі

Отже, алгоритм замовлення кальяну спрацював правильно. Оскільки в системі є продавець, прогестуємо функцію додавання товару. Спочатку продавець повинен пройти авторизацію у системі. Після авторизації у адмін панелі продавець повинен перейти у пункт редагування товарів та створити новий товар, рисунок 23 та 24.

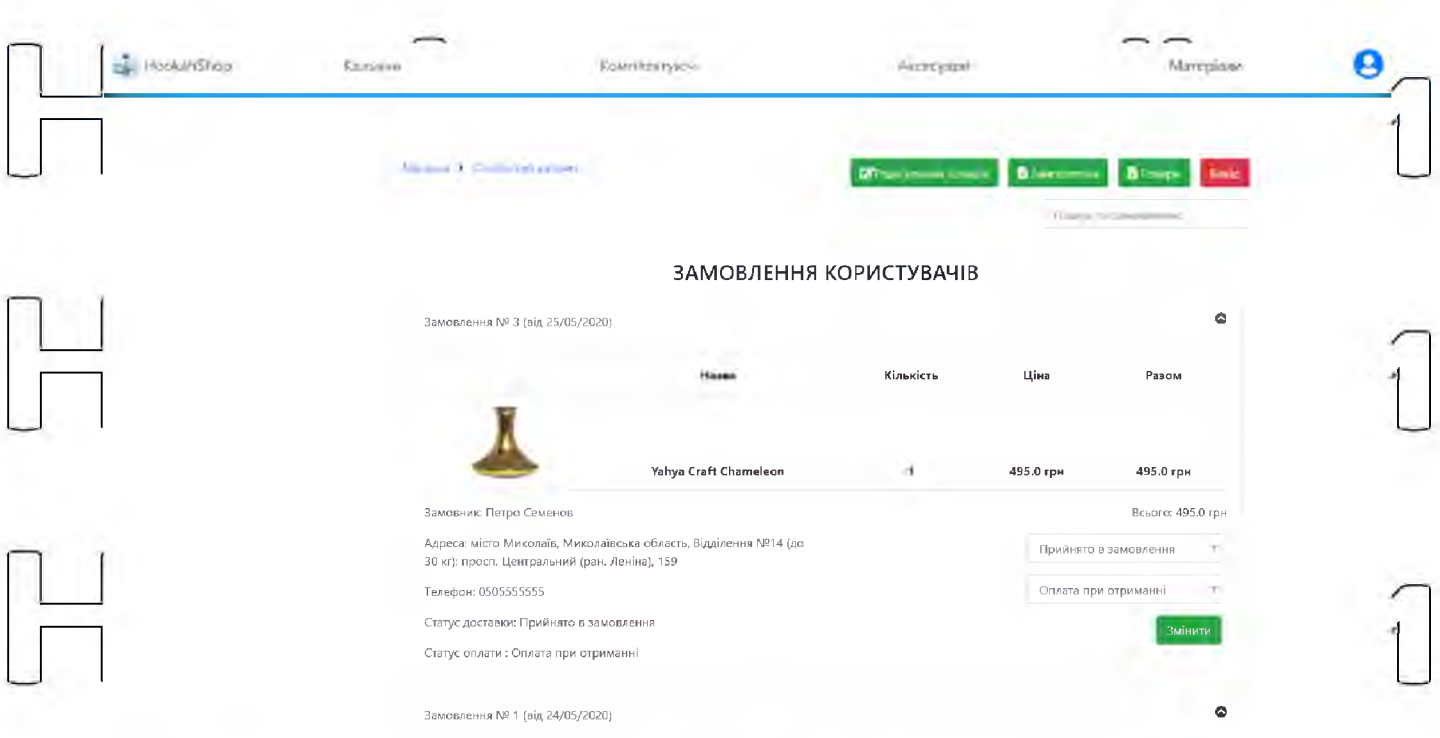


Рис. 7 Приклад кабінету продавця.

Як бачимо, адміністратор (продавець) має змогу змінювати статус замовлення, рис 24.

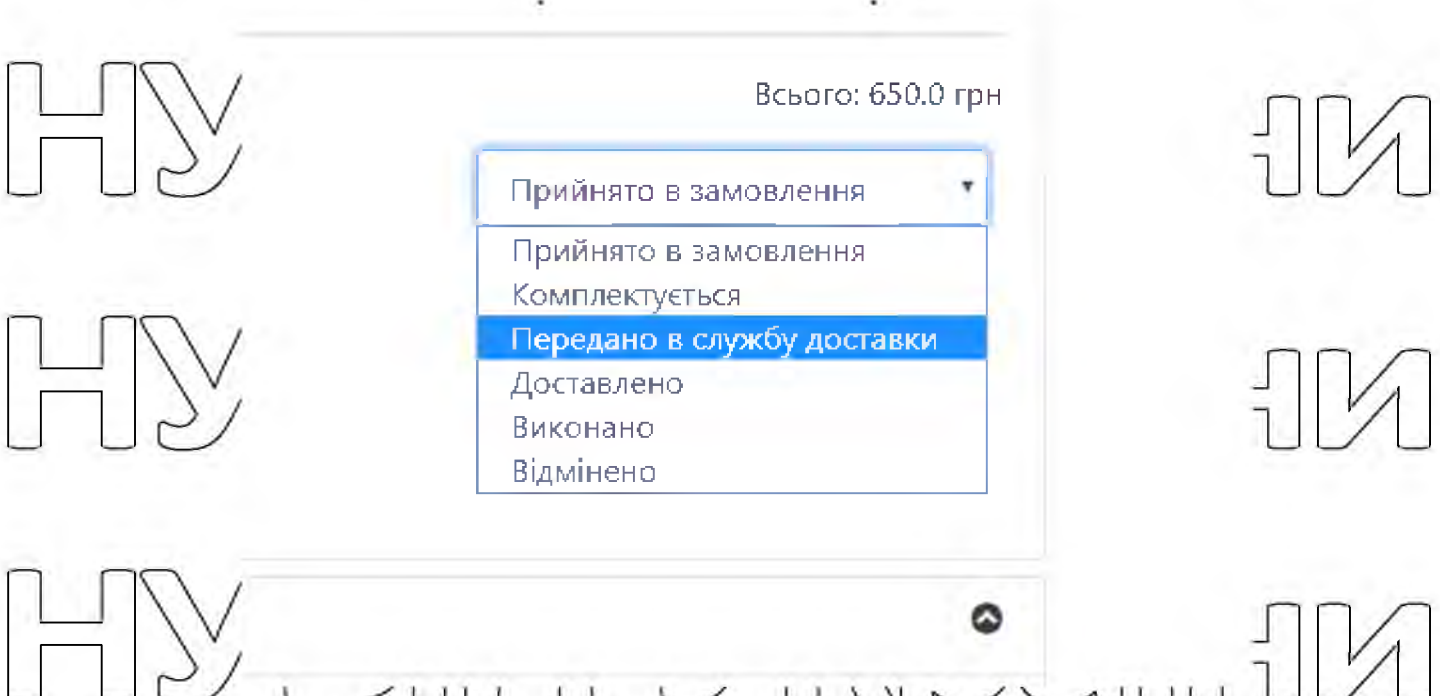


Рис. 24 Статус замовлення

Рис. 25 Додавання нового товару

У результаті створення товар можна переглянути у списку усіх товарів, рис

26.

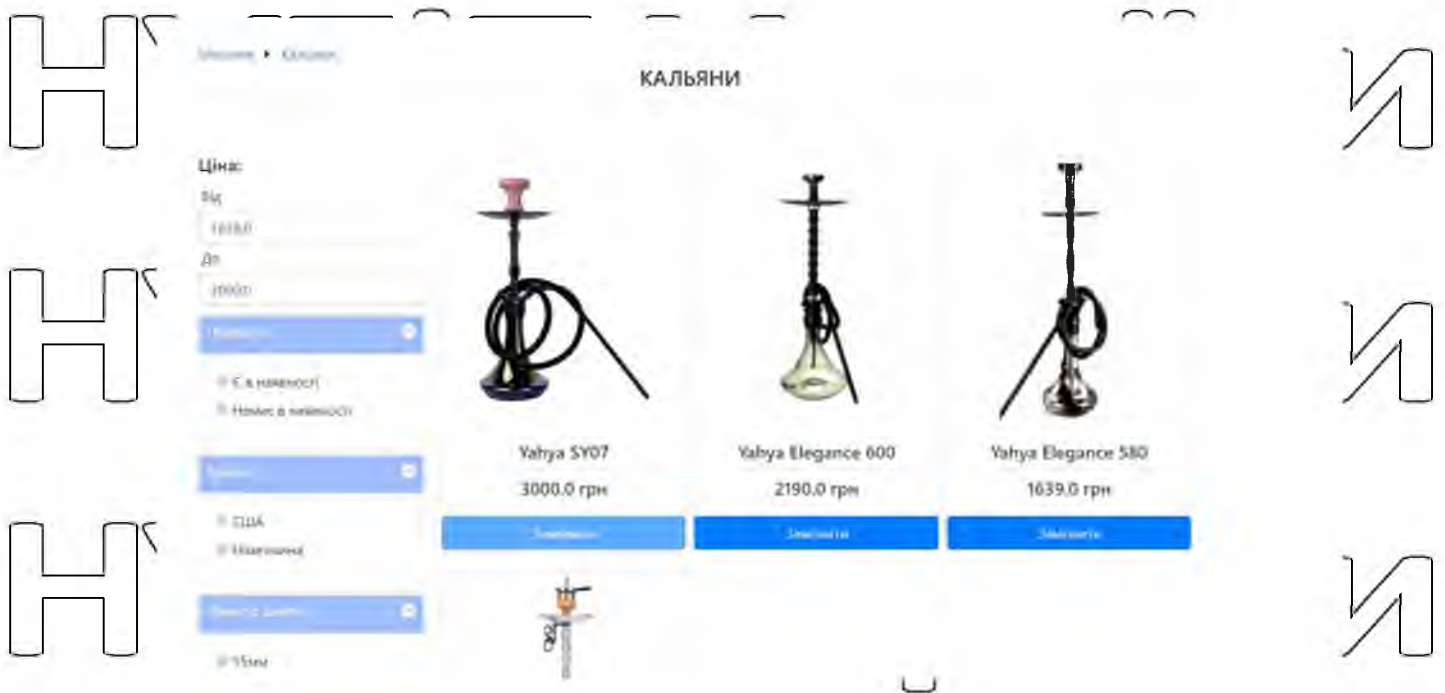


Рис. 26 Список товарів.



#### 4.3.1 Тестування системи користувачами.

Систему тестувало 13 чоловік, з яких 7 навчаються на факультеті інформаційних технологій, а також 6 інших здобувачів вищої освіти.

Під час тестування самої системи, користувачі перевірили наступні речі:

- коректність вибору товару
- правильність вибору категорій
- процес оплати
- деякі проблеми з доданими картами

Також майже всі користувачі вносили свої пропозиції до покращення системи і зручності використання сайту.

#### 4.3.2 Результати тестування.

За результатами тестування сайту було отримано наступні результати.

З 13 чоловік 12 зробили успішне замовлення з обраними ними товарами, не мали майже ніяких проблем, проте 3 з них повідомили про недоступність одного з товарів, це було виправлено у базі даних, а тринадцята людина, яка тестувала сайт зіткнулася з деякими проблемами з оплатою, але це було виправлено у коді, після чого все запрацювало стабільно.

#### 4.3.3 Висновки користувачів.

Після проведення тестування користувачами, було зроблено опитування, в ході котрого вони виділили деякі переваги і недоліки.

З боку переваг було виділено зручний та цікавий інтерфейс, приємний зовнішній вид, і доступність залишення коментарів.

Щодо недоліків, то отримано деякі зауваження з приводу функціональної роботи системи, які були виправлені в ході розробки, також були змінені деякі положення об'єктів для інтуїтивного розуміння сайту.

# НУБІП України **ВИСНОВКИ**

У ході розробки дипломного проекту нами була розроблена інформаційна

система для управління продажами в онлайн-магазині. Також стало зрозуміло,

що кожен клієнт – це індивідуальність, і до кожного потрібен свій підхід. Ця

система була розроблена з ціллю спрощення купівлі та отримання того чи іншого

товару в даній сфері.

Важливість цього додатку полягає у легкодоступності, і в точності підходу

до нової сфери, яка стає більш популярним з кожним днем. Клієнт має

можливість дуже швидко і зручно ознайомитися з інформацією про товар, та

його характеристики, зробити замовлення і отримати його в найближче

відділення пошти. Більш того, кожному клієнту доступна авторизація, що

спрощує роботу з додатком. Фільтрація зроблена для максимально ефективного

підбору бажаного товару.

Під час виконання дипломного проекту було досліджено предметну

область розроблюваної системи, за допомоги діаграм UML. Спроектовано

взаємодію з даними в системі за допомоги баз даних, інструментально

розроблено саму інформаційну систему «Підтримки продаж з веб - додатком».

Отже, внаслідок роботи над завданнями дипломної роботи ми розробили

повністю функціонуючу та апробовану інтелектуальну систему підтримки

торгівельної діяльності.

# НУБІП України

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Успенский И. В. ИНТЕРНЕТ-МАРКЕТИНГ / И. В. Успенский., 2003.
2. ЗАКОН УКРАЇНИ Про електронну комерцію [Електронний ресурс] – Режим доступу до ресурсу: <http://zakon2.rada.gov.ua/laws/show/675-19>.
3. The 19 Ecommerce Trends + 147 Online Shopping Stats Fueling Sales Growth in 2018 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bigcommerce.com/blog/ecommerce-trends>.
4. Saunders R. Business the Amazon.com Way: Secrets of the Worlds Most Astonishing Web Business / Rebecca Saunders., 1999. – 213 с. (Big Shots Series).
5. How Amazon and Independent Ecommerce Brands Grew Online Sales 18,233% in 20 Years [Електронний ресурс] – Режим доступу до ресурсу: <https://www.bigcommerce.com/blog/amazon-timeline-infographic/>.
6. Проектування інформаційних систем [Електронний ресурс] – Режим доступу до ресурсу: <https://goo.gl/W99HEA>.
7. Основные понятия баз данных [Електронний ресурс] – Режим доступу до ресурсу: [http://inf.susu.ac.ru/Klinachev/lc\\_sga\\_26.htm](http://inf.susu.ac.ru/Klinachev/lc_sga_26.htm).
8. Робсон Э. Изучаем HTML, XHTML и CSS / Э. Робсон, Э. Фримен., 2014. – 720 с. – (Head First (O'Reilly)).
9. Nixon R. Learning PHP, MySQL & JavaScript, 4th Edition With jQuery, CSS & HTML5 / Robin Nixon., 2014. – 810 с.
10. Ларман К. Применение UML 2.0 и шаблонов проектирования 3-е изд. / К. Ларман., 2013. – 736 с.
11. Unified Modeling Language (UML) Resource Page [Електронний ресурс] – Режим доступу до ресурсу: <http://www.uml.org/>.
12. Galbraith P. Developing Web Applications with Apache, MySQL, memcached, and Perl / Patrick Galbraith., 2009. – 888 с.
13. Markup Validation Service [Електронний ресурс] – Режим доступу до ресурсу: <https://validator.w3.org/>.
14. Шварц Б. High Performance MySQL / Б. Шварц, А. Ленц., 2010. – 832 с.

15. Поняття про моделювання систем, класифікації підходів і методів моделювання [Електронний ресурс] – Режим доступу до ресурсу: [https://stud.com.ua/24997/meredzhment/ponyattva\\_modelyvannva\\_sistem\\_klasifikatsivi\\_pidhodiv\\_metodiv\\_modelyvannva](https://stud.com.ua/24997/meredzhment/ponyattva_modelyvannva_sistem_klasifikatsivi_pidhodiv_metodiv_modelyvannva).

16. Навчальний посібник по використанню баз даних [Електронний ресурс] – Режим доступу до ресурсу: <https://www.postgresqltutorial.com/what-is-postgresql/>

17. СУБД SQL Server [Електронний ресурс] : інформаційний веб-сайт METANIT – Режим доступу: <https://metanit.com/sql/sqlserver/>.

18. Підприємство [Електронний ресурс] : Рішення для бізнесу - Режим доступу: <https://1c.ru/>

19. Інтелектуальні інформаційні системи [Електронний ресурс] : <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/25833/1/%D0%9A%D0%BE%D0%BD%D0%81%D0%BF%D0%B5%D0%BA%D1%82%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9.pdf>

20. Бандоріна Л. М. Методи багатокритеріальної оцінки ефективності інформаційно-інтелектуальних технологій [Електронний ресурс] : автореф. дис... канд. екон. наук : 08.03.02 / Бандоріна Лілія Миколаївна ; Дніпропетр. нац. університет. – Дніпропетровськ, 2005. – 20 с. – Режим доступу [Електронний ресурс] [http://www.irbis-nbuv.gov.ua/cgi-bin/irbis-nbuv.cgi?irbis=64.exe?C21COM=2&I21DBN=ARD&P21DBN=ARD&IMAGE\\_FILE\\_DOWNLOAD=1&Z21ID=&Image\\_file\\_name=DOC/2005/05blmiit.zip](http://www.irbis-nbuv.gov.ua/cgi-bin/irbis-nbuv.cgi?irbis=64.exe?C21COM=2&I21DBN=ARD&P21DBN=ARD&IMAGE_FILE_DOWNLOAD=1&Z21ID=&Image_file_name=DOC/2005/05blmiit.zip)

НУБІП України ДОДАТОК А

НУБІП України

НУБІП України

НУБІП України КОД ДОДАВАННЯ ТАБЛИЦЬ

НУБІП України Сторінок - 2

НУБІП України

НУБІП України КІЇВ - 2021

```

List<SubCatalog> subCatalogList = new ArrayList<>();
List<ShopProduct> shopProducts = new ArrayList<>();

```

```

// USER #1
ShopUser user = new ShopUser(1L, "Петро", "opanasenk000@ukr.net",
passwordEncoder.encode("user"),

```

```

Role.USER, null, null, null, null);
service.addUser(user);
// CART FOR USER #1
Cart cart = new Cart(1L, null);

```

```

service.addCart(cart);
service.addUser(user);
service.addOrder(new ShopOrder(1L, null, null, null,
null, null, null, null, null, cart, user, null, null));

```

```

// USER #2 ADMIN
ShopUser user1 = new ShopUser(2L, "Адміністратор",
"nedbail00098@gmail.com", passwordEncoder.encode("admin"),

```

```

Role.ADMIN, null, null, null, null);
service.addUser(user);
// CART FOR USER #2

```

```

Cart cart1 = new Cart(2L, null);
service.addCart(cart1);
service.addUser(user1);

```

НУБІП України Сторінка - 2  
service.addOrder(new ShopOrder(2L, null, null, null,  
null, null, null, null, null, cart1, user1, null, null));

НУБІП України  
// ADDRESS  
Address address = new Address(1L, "м.Київ, вулиця Героїв оборони, 17А");

НУБІП України  
Address address2 = new Address(2L, "м.Київ, вулиця Пушкіна, 15");  
Address address3 = new Address(3L, "м.Київ, вулиця Степана Бендери, 4");  
service.addAddress(address);  
service.addAddress(address2);

НУБІП України  
service.addAddress(address3);  
// SHOP DELIVERY  
ShopDelivery shopDelivery = new ShopDelivery(1L, "Нова пошта", null);

НУБІП України  
service.addDelivery(shopDelivery);  
ShopDelivery shopDelivery2 = new ShopDelivery(2L, "САМОВИВОЗ",  
Arrays.asList(address, address2, address3));  
service.addDelivery(shopDelivery2);

НУБІП України  
// ORDER STATUS  
ShopOrderStatus shopOrderStatus = new ShopOrderStatus

НУБІП України

НУБІП України ДОДАТОК Б

НУБІП України

НУБІП України

НУБІП України КОДИ СИСТЕМИ

НУБІП України Сторінка - 4

НУБІП України

НУБІП України КИЇВ - 2021



```
public class EmailSender {
    @Value("${spring.mail.username}")
```

```
private String username;
```

```
@Autowired
```

```
private MailSender mailSender;
```

```
public String send(String email, String subject, String message){
```

```
SimpleMailMessage mailMessage = new SimpleMailMessage();
```

```
mailMessage.setFrom(username);
```

```
mailMessage.setTo(email);
```

```
mailMessage.setSubject(subject);
```

```
mailMessage.setText(message);
```

```
mailSender.send(mailMessage);
```

```
return "redirect:";
```

```
}
```

```
public static ByteArrayInputStream getExcelOrdersList(List<ShopOrder>
```

```
shopOrderList){
```

```
String columns[] = {"id", "Ім'я", "Прізвище", "Телефон", "Email", "Адреса",
    "Статус доставки", "Статус замовлення", "Дата", "Загальна сума"};
```

```
try{
```

```
Workbook workbook = new XSSFWorkbook();
```

```
ByteArrayOutputStream out = new ByteArrayOutputStream();
```

```
Sheet sheet = workbook.createSheet("Замовлення");
```

```
Font headerFont = workbook.createFont();
```

```
headerFont.setBold(true);
```

```
headerFont.setColor(IndexedColors.AQUA.getIndex());
```

```
CellStyle headerCellStyle = workbook.createCellStyle();
```

```
headerCellStyle.setFont(headerFont);
```

```
Row headerRow = sheet.createRow(0);
for (int i = 0; i < columns.length; i++){
```

```
Cell cell = headerRow.createCell(i);
```

```
cell.setCellValue(columns[i]);
cell.setCellStyle(headerCellStyle);
}
```

```
int rowId = 1;
```

```
for (ShopOrder order : shopOrderList){
Row row = sheet.createRow(rowId++);
row.createCell(0).setCellValue(order.getId());
```

```
row.createCell(1).setCellValue(order.getName());
```

```
row.createCell(2).setCellValue(order.getSurname());
```

```
row.createCell(3).setCellValue(order.getPhone());
```

```
row.createCell(4).setCellValue(order.getEmail());
```

```
row.createCell(5).setCellValue(order.getAddress().getAddress());
```

```
row.createCell(6).setCellValue(order.getIsPaid() ? "Оплачено" : "Оплата при
отриманні");
```

```
row.createCell(7).setCellValue(order.getStatus().getStatusName());
```

```
row.createCell(8).setCellValue(order.getDate());
```

```
row.createCell(9).setCellValue(order.getSummary());
```

```
}
Row row = sheet.createRow(rowId);
```

```
row.createCell(8).setCellValue("Разом");
```

```
row.createCell(9).setCellFormula(String.format("SUM(%s:%s)", "J2", "J"+rowId));
```

```
CellStyle yellowStyle = workbook.createCellStyle();
```

```
yellowStyle.setFillForegroundColor(IndexedColors.YELLOW.getIndex());
```

```
yellowStyle.setFillPattern(FillPatternType.SOLID_FOREGROUND);
```

row.getCell(9).setCellStyle(yellowStyle);

# НУБІП України

Сторінка - 3

```
autoSizeColumns(workbook);
```

```
workbook.write(out);
return new ByteArrayInputStream(out.toByteArray());
}
```

# НУБІП України

```
}

public LiqPay cnb_form(Map<String, String> params, String publicKey, String
privateKey) {
String data =
```

# НУБІП України

```
base64Encoder(JSONObject.toJSONString(withSandboxParam(withBasicApiParams
(params, publicKey))));
```

```
String signature = createSignature(data, privateKey);
return new LiqPay(data, signature);
}
```

# НУБІП України

```
public TreeMap<String, String> withBasicApiParams(Map<String, String> params,
```

```
String publicKey) {
TreeMap<String, String> tm = new TreeMap<>(params);
tm.put("public_key", publicKey);
```

# НУБІП України

```
tm.put("version", API_VERSION);
```

```
return tm;
}

public TreeMap<String, String> withSandboxParam(TreeMap<String, String>
```

# НУБІП України

```
params) {
```

```
if (params.get("sandbox") == null) {
TreeMap<String, String> tm = new TreeMap<>(params);
tm.put("sandbox", "1");
```

# НУБІП України

```
return tm;
```

НУБІП України Сторінка – 4

```
return params;
```

```
}
```

```
public String createSignature(String base64EncodedData, String privateKey) {  
    return base64Encoder(sha1(privateKey + base64EncodedData + privateKey));  
}
```

```
public String getStatus(Map<String, String> params, String publicKey, String
```

```
privateKey) throws Exception {
```

```
String data =
```

```
base64Encoder(JSONObject.toJSONString(withBasicApiParams(params,  
publicKey)));
```

```
String signature = createSignature(data, privateKey);
```

```
// data : base64Encoder(json{order_id, action, version, public key})
```

```
// signature : base64Encoder(sha1(privateKey + data + privateKey))
```

```
// 2 parameters data and signature must be entered
```

```
HashMap<String,String> result = new HashMap<>();
```

```
result.put("data", data);
```

```
result.put("signature", signature);
```

```
// from liqpay api sdk
```

```
String urlParameters = "";
```

```
for (Map.Entry<String, String> entry : result.entrySet())
```

```
urlParameters += entry.getKey() + "=" + URLEncoder.encode(entry.getValue(),  
"UTF-8") + "&;
```

```
RestTemplate restTemplate = new RestTemplate();
```

```
HttpHeaders httpHeaders = new HttpHeaders();
```

```
httpHeaders.setContentType(MediaType.APPLICATION_JSON);
```

```
HttpEntity<String> request = new HttpEntity<>(urlParameters, httpHeaders);
```

```
}
```

НУБІП України

ДОДАТОК В

НУБІП України

НУБІП України

НУБІП України

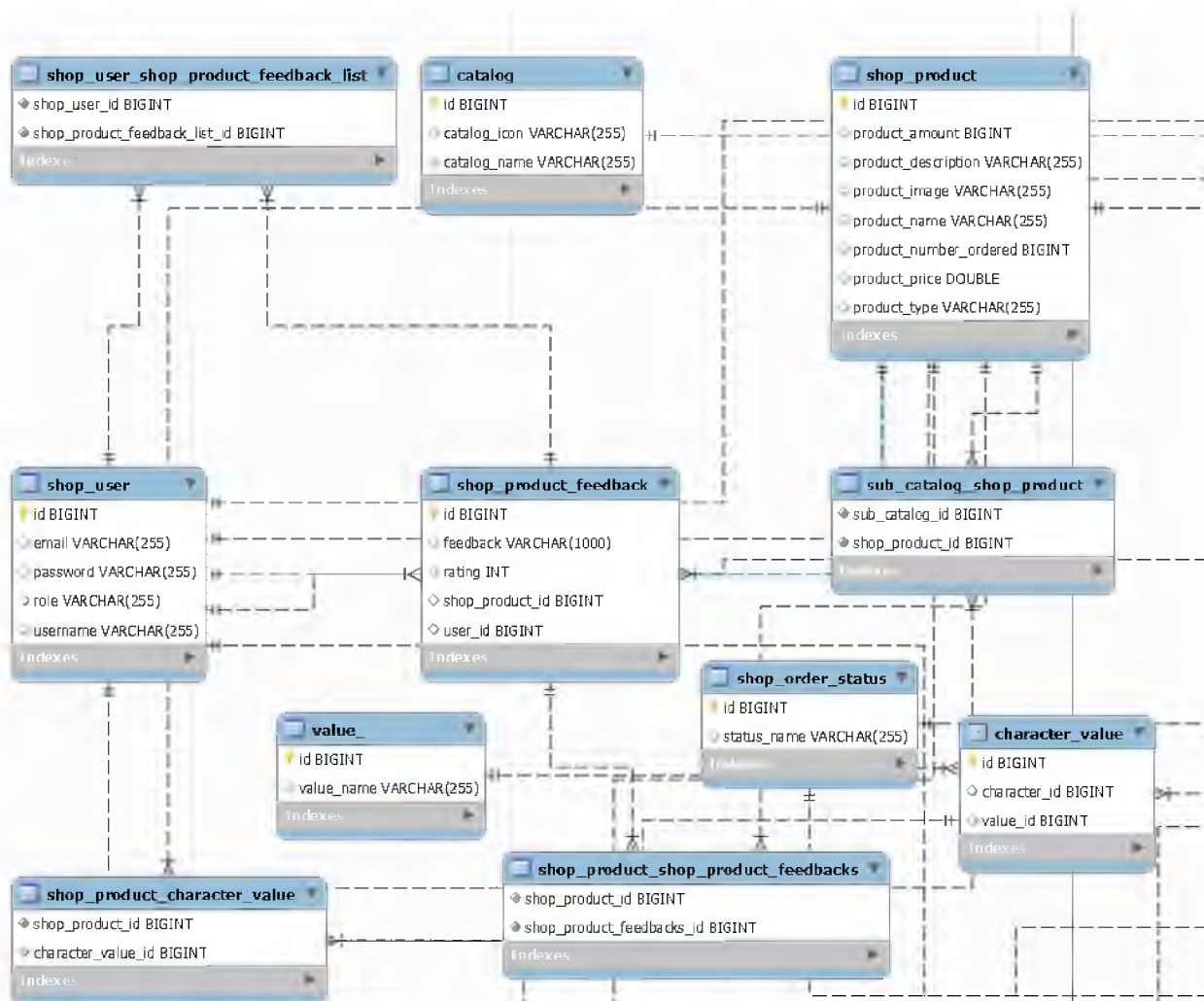
ДІАГРАМА КЛАСІВ

НУБІП України

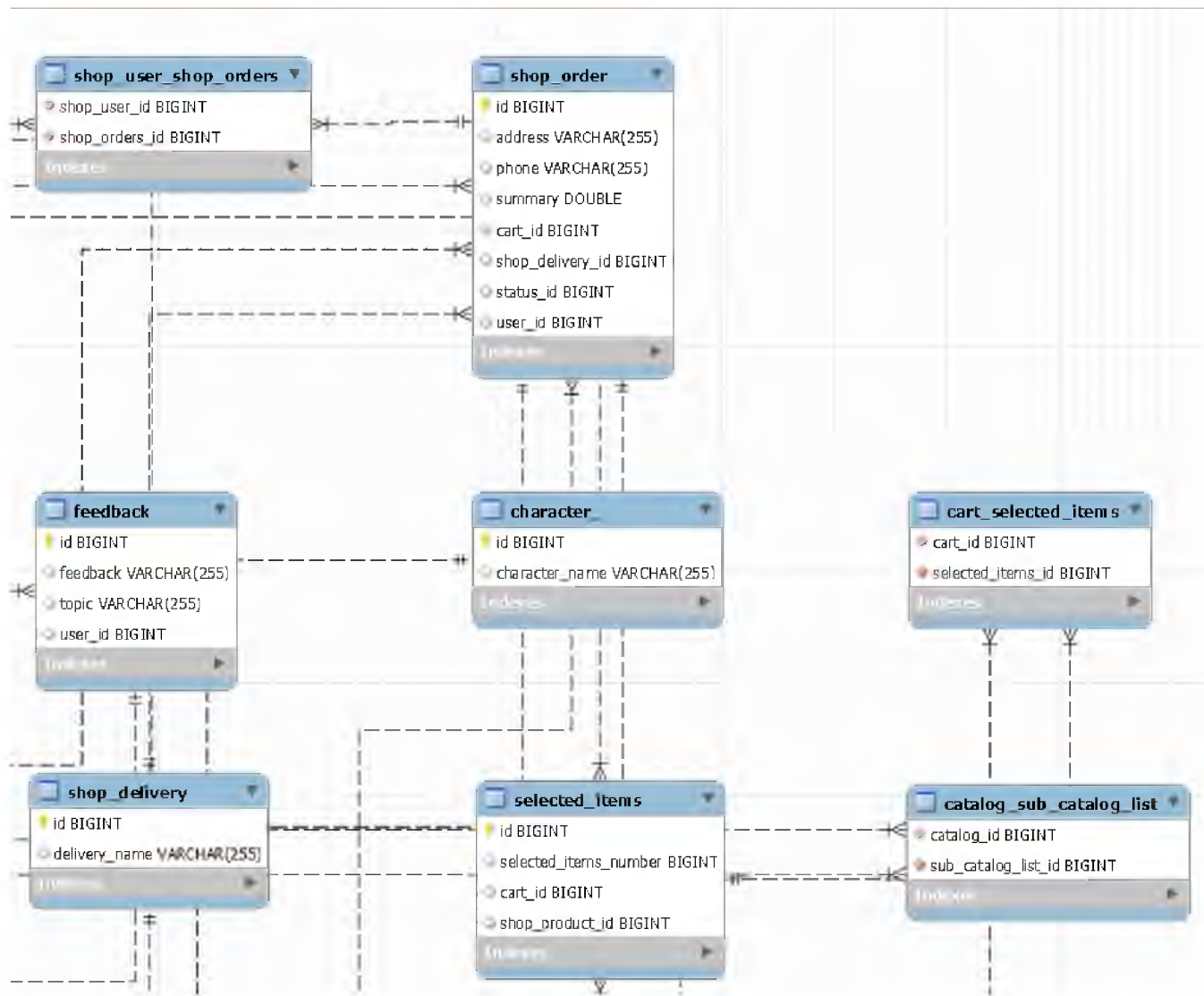
НУБІП України

Сторінок 13

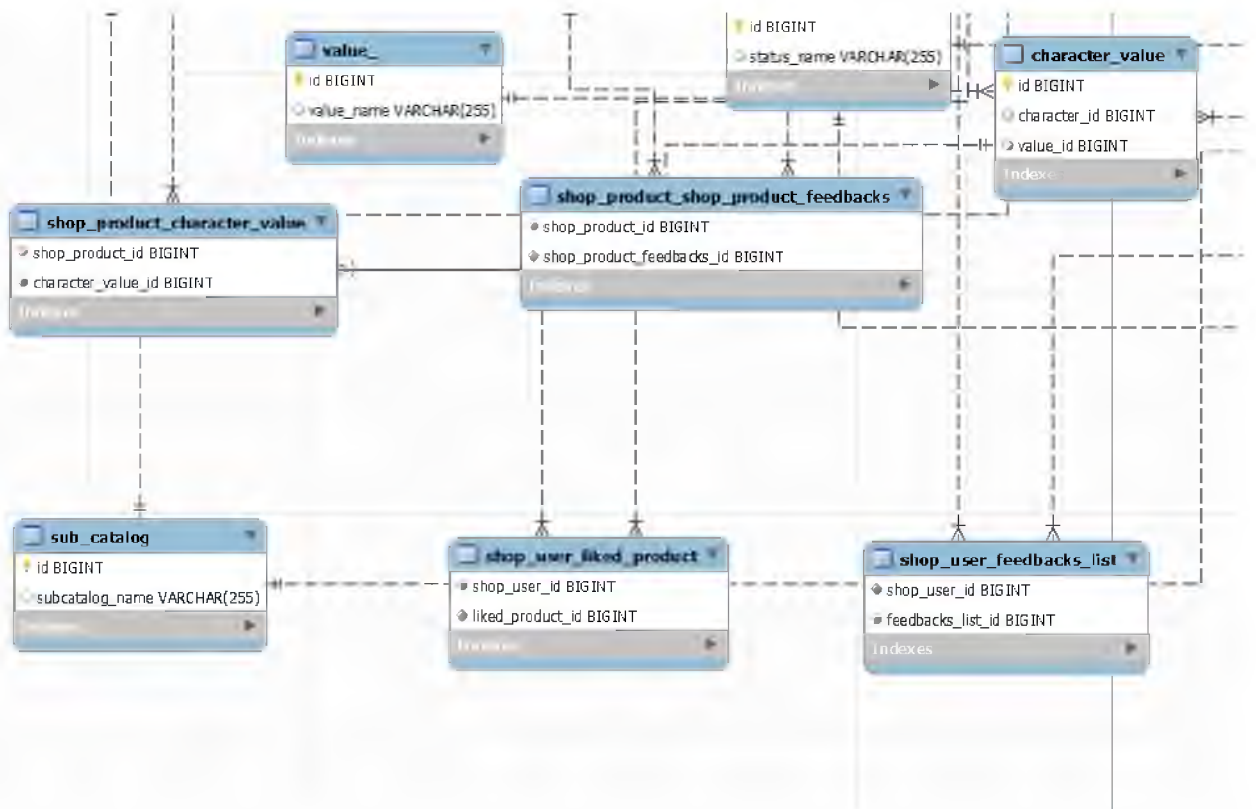
НУБІП України



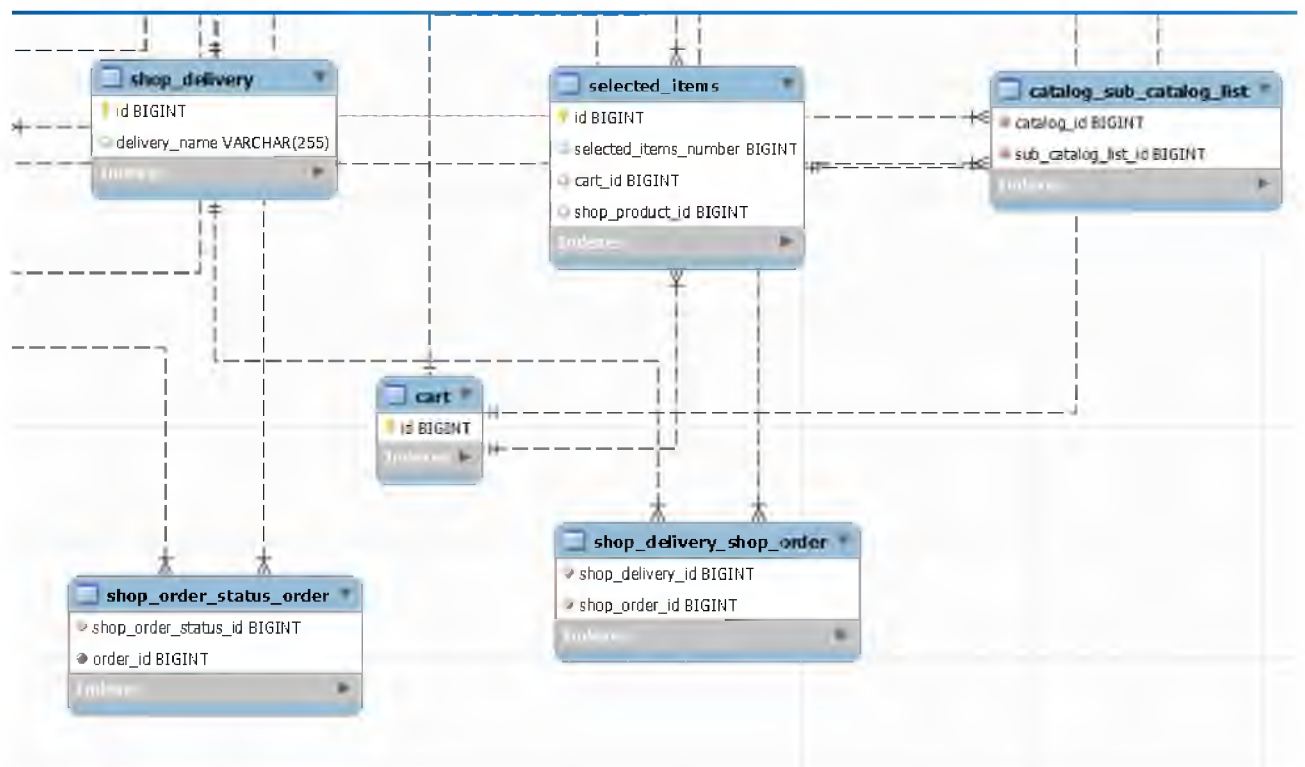
Лівий верхній кут



Правий верхній кут



Лівий нижній кут



Правий нижній кут



ГРАФІЧНІ МАТЕРІАЛИ