

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРОКАТУ
ДИТЯЧОГО ОДЯГУ
SOFTWARE DEVELOPMENT FOR CHILDREN'S CLOTHING RENTAL

Виконав: студент(ка) 4 курсу, групи КНТ-127
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Янчара А.М.

(прізвище та ініціали)

Керівник Федорончак Т.В.

(прізвище та ініціали)

Рецензент Казурова А.Є.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Янчари Анастасії Миколаївни

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення прокату дитячого одягу

Software Development for Children's Clothing Rental

керівник проєкту (роботи) Федорончак Тетяна Василівна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 26.05.2021

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми. 3. Розробка програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Федорончак Т.В., доцент		
Нормоконтролер	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “18” березня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка архітектури програми	4 тиждень	Розділ 2
4	Розробка програми	5-6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент(ка)

(підпис)

Янчара А.М.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Федорончак Т.В.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 80 с., 26 рис., 16 табл., 3 дод., 14 джерел.

ДИТЯЧИЙ ОДЯГ, СЕРВІС ПРОКАТУ, ЗАМОВЛЕННЯ, ВЕБЗАСТОСУНОК, PHP, MYSQL.

Об'єкт дослідження – процес прокату дитячого одягу. Предмет дослідження – програмне забезпечення прокату дитячого одягу.

Мета роботи – розроблення програмного забезпечення для підтримки роботи з клієнтами прокату дитячого одягу.

Матеріали, методи та технічні засоби: мова програмування PHP, система керування базами даних MySQL.

Результати. Виконано аналіз предметної області, що включив аналіз і порівняння програмних засобів, які можуть вирішувати проблему. Визначено функціональні вимоги до програми, обрано мову програмування PHP і систему керування базами даних MySQL для реалізації програмного забезпечення, виконано проєктування бази даних. Реалізовано вебзастосунок для підтримки прокату дитячого одягу. Описано алгоритм функціонування програмного забезпечення з приведенням відповідної схеми функціонування, описано структуру створеного програмного забезпечення. Представлено основні розроблені в процесі реалізації класи.

Висновки. Розроблене програмне забезпечення дозволяє перевіряти доступність дитячого одягу для прокату на задані дати, створювати власні замовлення, прокатникам керувати створеними користувачами замовленнями, створювати каталог дитячого одягу, вносити дані про наявність дитячого одягу у відповідних відділеннях прокату.

Галузь використання. Сервіси прокату дитячого одягу з одним або декількома відділеннями.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 80 pages, 26 figures, 16 tables, 3 appendixes, 14 sources.

CHILDREN'S CLOTHES, RENTAL SERVICE, ORDERS, WEB APPLICATION, PHP, MYSQL.

The object of research is the process of renting children's clothes. The subject of the research is children's clothing rental software.

The purpose of the work is to develop software to support work with customers of children's clothing rental.

Materials, methods and technical means: PHP programming language, MySQL database management system.

Results. The analysis of the subject area, which included the analysis and comparison of software for solving the problem, was performed. The functional requirements for the software are defined, the PHP programming language and the MySQL database management system for software implementation are selected, the database is designed. Web application was implemented to support the rental of children's clothing. The algorithm of software operation with the corresponding scheme of functioning is described, the structure of the created software is described. The main classes developed in the process of implementation are presented.

Conclusions. The developed software allows users to check the availability of children's clothing for rent on specified dates, create own orders, allows renters to manage user-created orders, create a catalog of children's clothing, enter data on the availability of children's clothing in the relevant rental offices.

Field of usage. Children's clothing rental services with one or more branches.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	9
1.1 Програмні засоби прокату дитячого одягу.....	9
1.2 Порівняння існуючих програмних засобів.....	14
1.3 Висновки за розділом 1	15
2 Розробка архітектури програми.....	17
2.1 Функціональні вимоги до програмного забезпечення	17
2.2 Вибір мови програмування	19
2.3 Проектування бази даних	22
2.4 Висновки за розділом 2	31
3 Розробка програми	32
3.1 Алгоритм функціонування програмного забезпечення	32
3.2 Структура розробленого програмного забезпечення	34
3.3 Опис прийнятих під час реалізації рішень	36
3.4 Висновки за розділом 3	41
4 Експлуатація, тестування та експериментальне дослідження програми	42
4.1 Призначення програми	42
4.2 Умови виконання програми	42
4.3 Виконання програми.....	42
Висновки	50
Перелік джерел посилання	51
Додаток А Технічне завдання	53
Додаток Б Текст програми	57
Додаток В Слайди презентації.....	74

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД – база даних;

СКБД – система керування базами даних.

ВСТУП

Прокат дитячого одягу є однією з тих сфер, які за рахунок розвитку інформаційних технологій отримали додатковий поштовх до розвитку.

За статистикою орієнтовно 350 000 тонн одягу (вартістю 140 мільйонів фунтів стерлінгів) щороку збирається на сміттєзвалища у Великобританії, тому разом з революцією у споживанні відносно одягу загалом, здійснюється революція для дітей, відносно якої можна відзначити наступне:

- орендований одяг для дітей є зручним та доступним рішенням для екологічно обізнаних батьків, які хочуть до того ж мінімізувати безлад у своїх будинках, адже діти, як правило, переростають одяг протягом декількох місяців;
- за дослідженнями у випадку продовження терміну експлуатації одягу на дев'ять місяців активного використання викиди вуглецю та відходів можуть зменшитися на цілих 30 відсотків за одиницю;
- предмети дитячого одягу можна здавати в оренду декілька разів, відремонтувати за необхідності [1]-[2].

Усе це створює постійну потребу в сервісах прокату дитячого одягу. Разом з тим актуалізується і проблема розроблення програмного забезпечення прокату дитячого одягу, яку і потрібно розв'язати в дипломній кваліфікаційній роботі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмні засоби прокату дитячого одягу

Засобів прокату дитячого одягу в Україні загалом достатньо велика кількість, але всі вони мають певні обмеження. Наприклад, популярна дошка оголошень olx.ua [3] (рис. 1.1) надає для оголошень щодо прокату дитячого одягу ту саму структуру роботи, що надається і для будь-якого іншого оголошення, тобто специфіка не враховується.

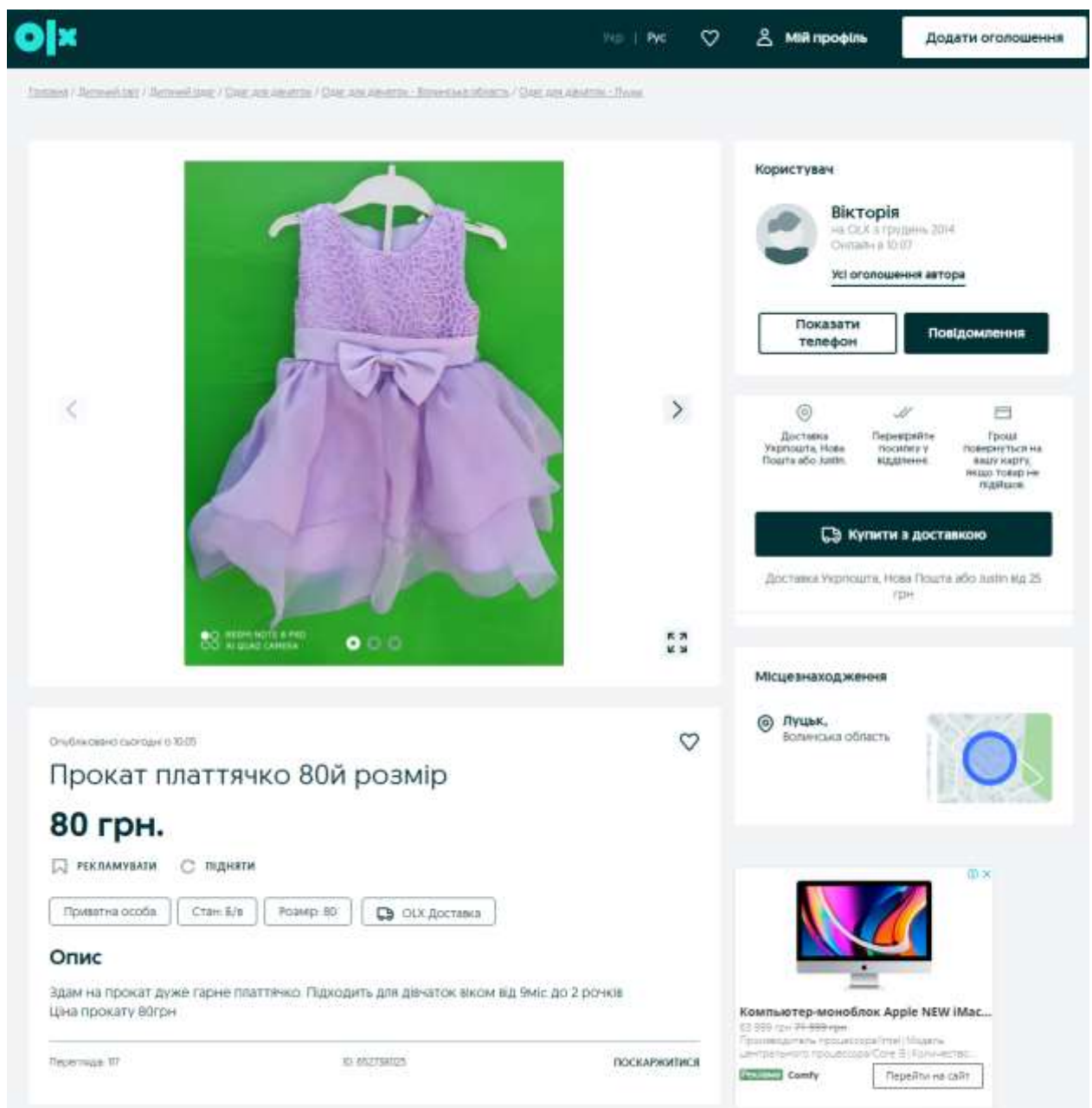


Рисунок 1.1 – Вебсервіс olx.ua

В olx можна отримати інформацію про самий дитячий одяг, місцезнаходження продавця, його розміри і стан, але будь-яка інтерактивність відсутня, тобто не можна перевірити наявність певного розміру в поточний момент, окрім того усі оголошення фрагментовані, тобто весь дитячий одяг відноситься до загальної категорії дитячого одягу. Немає ніяких засобів додаткової категоризації. Відповідно пошук здійснюється за назвою, а для його успішного виконання потрібно вгадати ту назву або опис, які встановив автор оголошення. Тобто роботу зі знаходження потрібних товарів можна вважати ускладненою.

Вебсервіс zakupka.com [4] (рис. 1.2) окремо виділяє прокат дитячого одягу, має певні засоби виділення особливих характеристик одягу (наприклад, визначення статі дитини, категоризації, зокрема визначення вигляду (наприклад, святковий), так званого коду, який фактично є категорією (наприклад, куртка). При цьому використовувати останні засоби достатньо незручно, тому що тільки за статтю дитини можна відобразити весь відповідний одяг, а відобразити одяг за виглядом або за категорією неможливо.

Варто також вказати на відсутність засобів перевірки наявності, а також зручного визначення доступних розмірів, адже вони можуть бути тільки вказані в назві або описі речі, тобто переходити до відповідних речей за розміром або виконувати пошук за розміром неможливо, що значно звужує використання тих засобів, що існують.

Виконувати пошук можна, наприклад, за вартістю прокату, але розширені засоби відсутні. При цьому olx, не зважаючи на обмеженість засобів, виконувати такий пошук дозволяє.

Оформлення замовлення відбувається у вигляді відправлення повідомлення, при цьому не зазначаються дати прокату, тобто це інформування продавця про бажання зв'язатися, а не замовлення прокату як таке.

Zakupka.com RU Каталог

Пошук по 10 500 000 товарів Zaku... **Знайти**

Запоріжжя Вибране Кошик Вхід Реєстрація

Запоріжжя Товари Одяг Одяг дитячий Костюми дитячі Стильний костюм зі спідницею 152

Стильний бузковий брючний КОСТЮМ

У обране Замовити дзвінок Задати питання

400 ₴ **В наявності**

Купити +380 (97) 43... показати

Доставка
Можливість доставки післяплатою уточнюйте
Нова Пошта
Укрпошта
Самовивіз з Хмельницької безпосередньо
[Детальніше про доставку](#)

Оплата
Накладною платіж
Оплата на карту «ПриватБанк»

Продавець
Інтернет-магазин "SiTN family shop"
Всі товари продавця
Хмельницький
Додатковий
Умови повернення та обміну
Рекомендації по безпеці покупок
Повідомити про недостоірну інформацію

опис
Перевірений продавець «Інтернет-магазин "SiTN family shop"» (з м. Хмельницький) працює на порталі Zakupka.com.
Пропонує купити Стильний костюм зі спідницею 152

Трендовий костюм зі спідницею
Кофта трикотаж кукурузка
спідниця -Мікро вельвет

Характеристики

Країна виробництва	Україна
Стать дитини	для дівчаток
Розмір одягу для дітей (по зростанню), см	152
сезон	осінь, весна, літо, демісезон
стиль	святковий

Рисунок 1.2 – Сторінка дитячої речі на вебсервісі zakupka.com

Вебсервіс barvu.net [5] (рис. 1.3) на відміну від попередніх аналогів є особливим тим, що він надає засоби замовлення прокату від одного й того самого прокатника. Тобто це вебсайт компанії з прокату дитячого одягу. На попередніх вебсайтах прокатниками могли виступати різні компанії.

У даному рішенні відсутній пошук як такий, замість цього використовуються засоби категоризації речей. Наприклад, речі відображаються за групами («Пори року і природа», «Квіти та ягоди», «Птахи та комахи» тощо).

Негативною особливістю є те, що не можна виконати замовлення через вебсайт. Є відповідна кнопка, але вона призводить до переходу до контактних даних сервісу прокату. Відповідно і перевірити, чи наявні потрібні розміри речей на потрібні дати неможливо. Вебсайт можна

використовувати тільки як каталог дитячого одягу, що дещо звужує використання самого вебсайту.

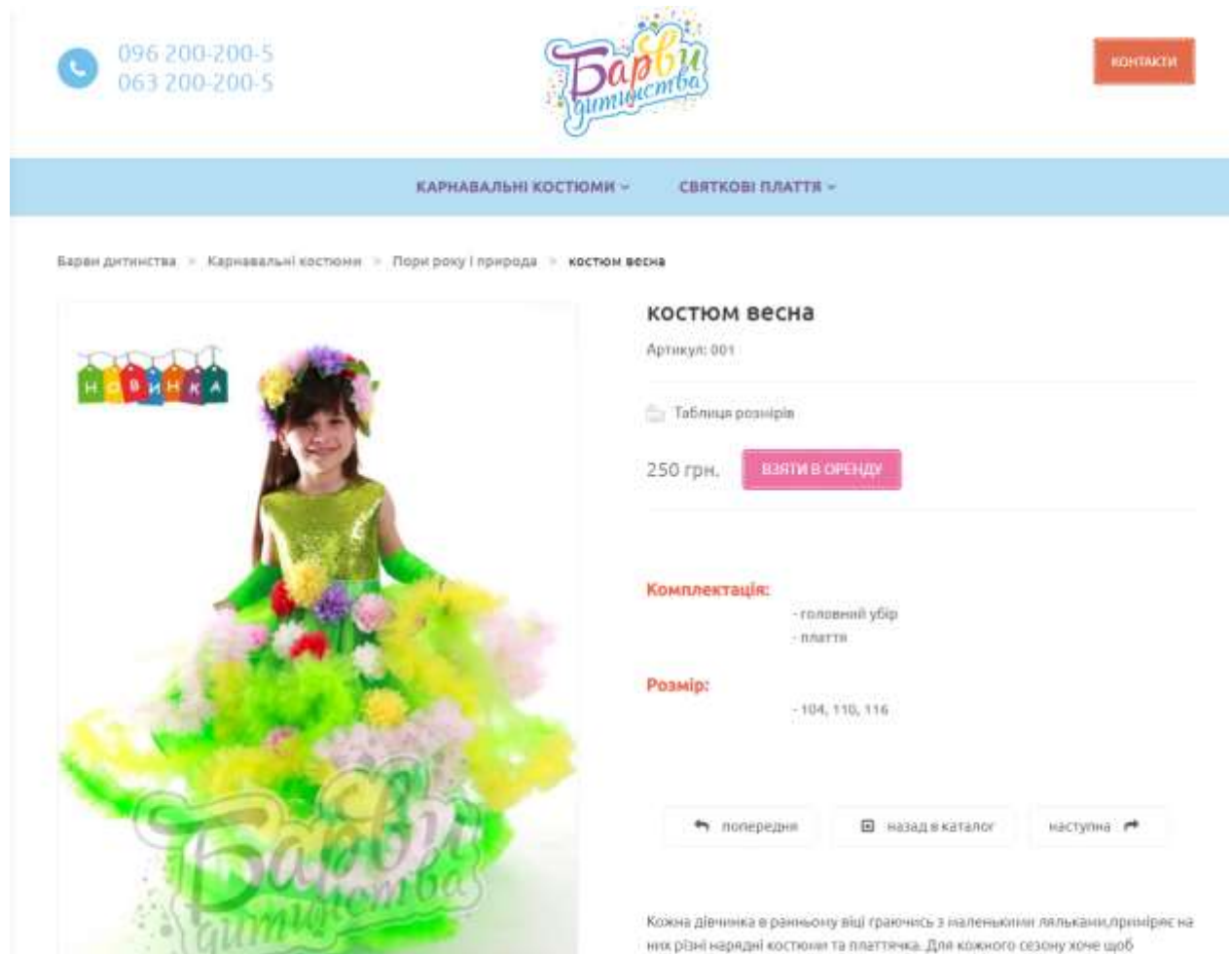


Рисунок 1.3 – Сторінка дитячої речі на вебсервісі barvu.net

За кожною річчю доступна інформація про її розміри, опис, фотографії, вартість.

Вебсервіс gir.ua [6] (рис. 1.4) направлений на оренду та прокат різних речей, не тільки дитячого одягу, але така категорія як дитячий одяг в ньому присутня.

Особливістю сервісу є те, що враховуючи його направленість саме на прокат, на сторінках одягу присутня форма перевірки доступності дитячого одягу на задані дати, при цьому одразу ж виконується розрахунок вартості такого прокату і надається засіб створення замовлення на потрібні дати. Це є

головною особливістю даного програмного рішення порівняно з іншими аналогами.

При цьому щодо роботи саме з дитячим одягом засобів небагато. Можна переглянути умови оренди, є можливість визначення різних тарифів прокату, але категоризація загалом відсутня. Не можна також і під час пошуку задати особливі характеристики (наприклад, розмір), тобто пошук відбувається лише за назвою.

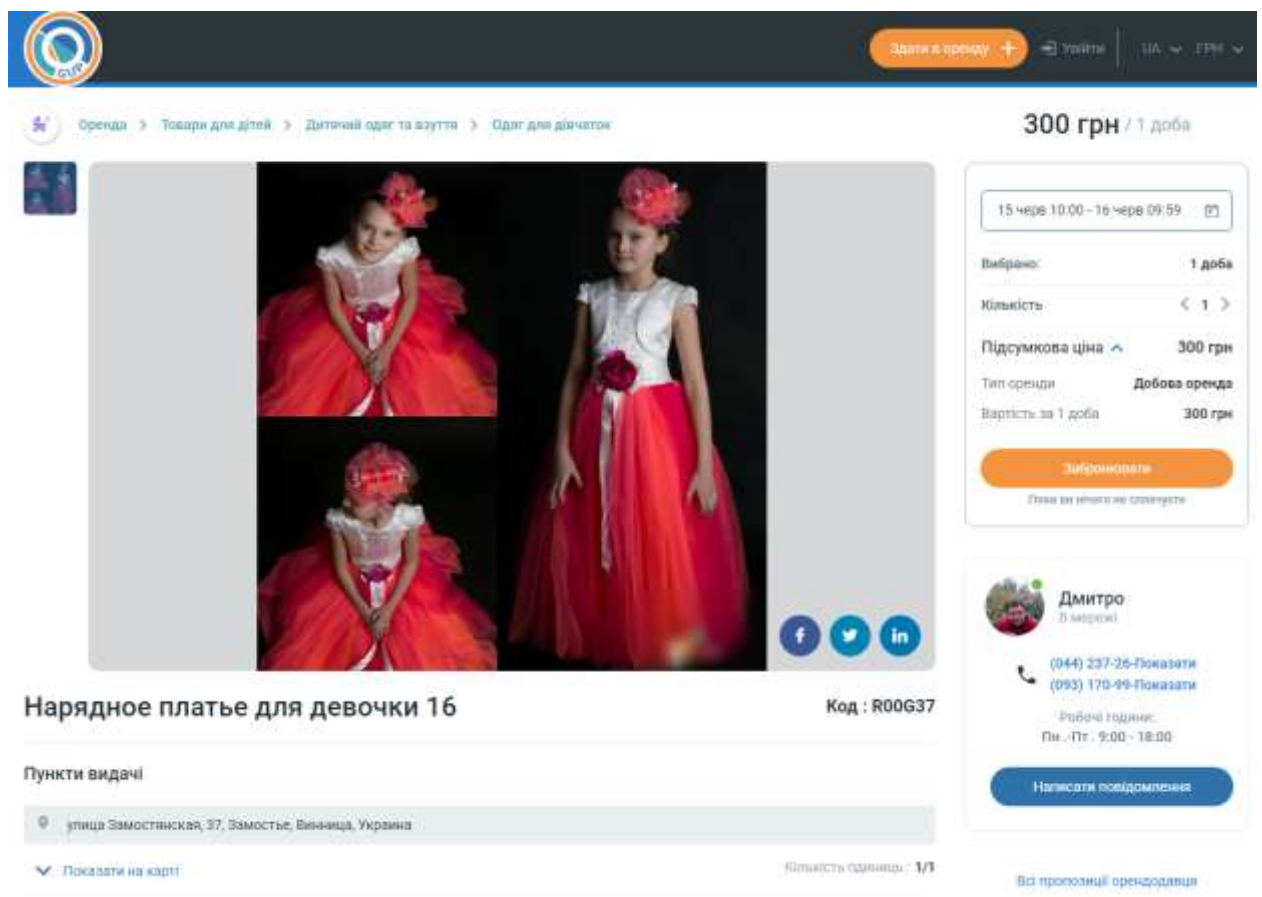


Рисунок 1.4 – Сторінка дитячої речі на вебсервісі gur.ua

Окрім того можна згадати про існування вебсайтів, які загалом створюють певний каталог дитячого одягу, а підтримка роботи над прокатом реалізується через особисті контакти, або сайти-візитівки відповідних сервісів.

1.2 Порівняння існуючих програмних засобів

За результатами виконаного аналізу програмних аналогів було виконано їх порівняння (табл. 1.1). До основних критеріїв порівняння віднесено:

- пошук одягу: базова функція, яка має надаватися будь-яким сервісом, надається всіма винесеними в остаточне порівняння аналогами, але не надається сайтами-каталогами, наприклад, barvu.net;
- пошук за розміром: дана функція надається тими засобами, які враховують специфіку дитячого одягу, відсутня в zakupka.com;
- категоризація одягу: є зручним інструментом загального перегляду одягу, який має бути присутній безвідносно наявності пошуку;
- відображення одягу за категоріями: при наявності категорій як таких, важливо, щоб була можливість переходити за ними, так у zakupka.com є категорії в конкретному оголошенні, але перейти за ними неможливо, даний засіб присутній у barvu.net;
- відображення одягу за статтю: є функцією відносно однієї з найпростіших категорій;
- замовлення прокату: важлива відмінність повноцінних засобів прокату від каталогів;
- замовлення прокату на задані дати: має бути реалізовано повноцінну функцію замовлення прокату, а не просто відправлення повідомлення прокатнику;
- відгуки про одяг: відгуки інших користувачів про одяг, який був взятий ними на прокат, є важливою функцією для будь-кого, хто намагається знайти для себе можливі варіанти для прокату;
- перевірка наявності одягу на задані дати: є зручним засобом, який дозволяє зрозуміти, які альтернативні варіанти слід розглядати.

Таблиця 1.1 – Порівняння програмних аналогів

	zakupka.com	olx.ua	gur.ua	Розробка
Пошук одягу	+	+	+	+
Пошук за розміром	—	+	+	+
Категоризація одягу	+	—	—	+
Відображення одягу за категоріями	—	—	—	+
Відображення одягу за статтю	+	+	+	+
Замовлення прокату	+	+	+	+
Замовлення прокату на задані дати	—	—	+	+
Відгуки про одяг	+	—	—	+
Перевірка наявності одягу на задані дати	—	—	+	+

1.3 Висновки за розділом 1

Для досягнення мети дипломної кваліфікаційної роботи було визначено наступні завдання роботи:

- аналіз предметної області;
- проєктування програмного забезпечення;
- реалізація програмного забезпечення.

У першому розділі виконано аналіз предметної області, що включив аналіз програмних засобів, які можуть вирішувати проблему, що розглядається, включаючи olx.ua, zakupka.com, gur.ua, barvu.net, а також виконано порівняння аналогів і виділено основні функції, які в якості переваг над альтернативами має виконувати розробка.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

2.1 Функціональні вимоги до програмного забезпечення

На основі проведеного порівняння програмних аналогів, виділення особливостей програмного забезпечення, що розробляється, а також необхідності підтримки даних функцій, було сформульовано набір функціональних вимог, які має забезпечити програма:

- внесення наявного дитячого одягу;
- редагування інформації про дитячий одяг;
- перегляд наявного загалом і у відділенні дитячого одягу;
- перегляд замовлень на прокат за відділенням;
- оформлення замовлення на прокат речі;
- пошук потрібного дитячого одягу;
- перегляд всіх речей за заданим розміром;
- перегляд дитячого одягу за категорією;
- перегляд дитячого одягу за групами;
- перегляд дитячого одягу за статтю;
- перегляд дитячого одягу за відділенням;
- перегляд власних замовлень на прокат;
- скасування замовлення на прокат;
- підтвердження замовлення на прокат;
- перегляд доступних речей на заданий період;
- реєстрація облікового запису клієнта;
- авторизація облікового запису;
- створення відгуку про одяг;
- створення відгуку про клієнта;
- перегляд відгуків про клієнта;
- перегляд відгуків про одяг;
- перегляд інформації про одяг;
- уподобання одягу;

– перегляд власних уподобань одягу.

На основі даного переліку функціональних вимог було виконано моделювання роботи користувачів з системою.

Прокатник може вносити дані одягу, тобто створювати каталог дитячого одягу, визначати розташування одягу, а також може використовувати загальні засоби роботи з каталогом одягу. Відповідна діаграма use case приведена на рис. 2.1.

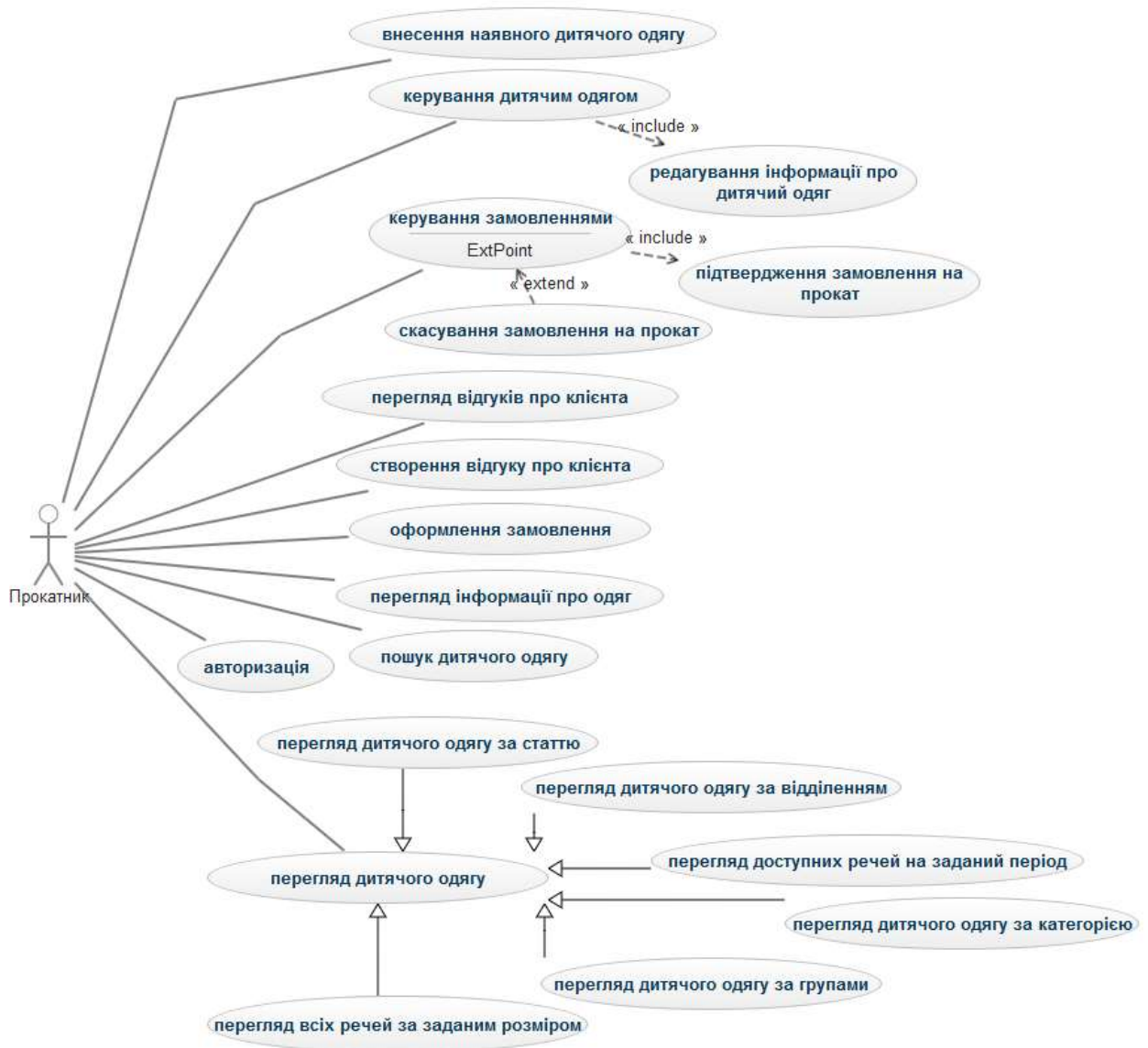


Рисунок 2.1 – Діаграма use case прокатника

Клієнт може працювати з наборами дитячого одягу також власними даними, що стосуються замовлень, відгуків. Відповідна діаграма use case

приведена на рис. 2.2.

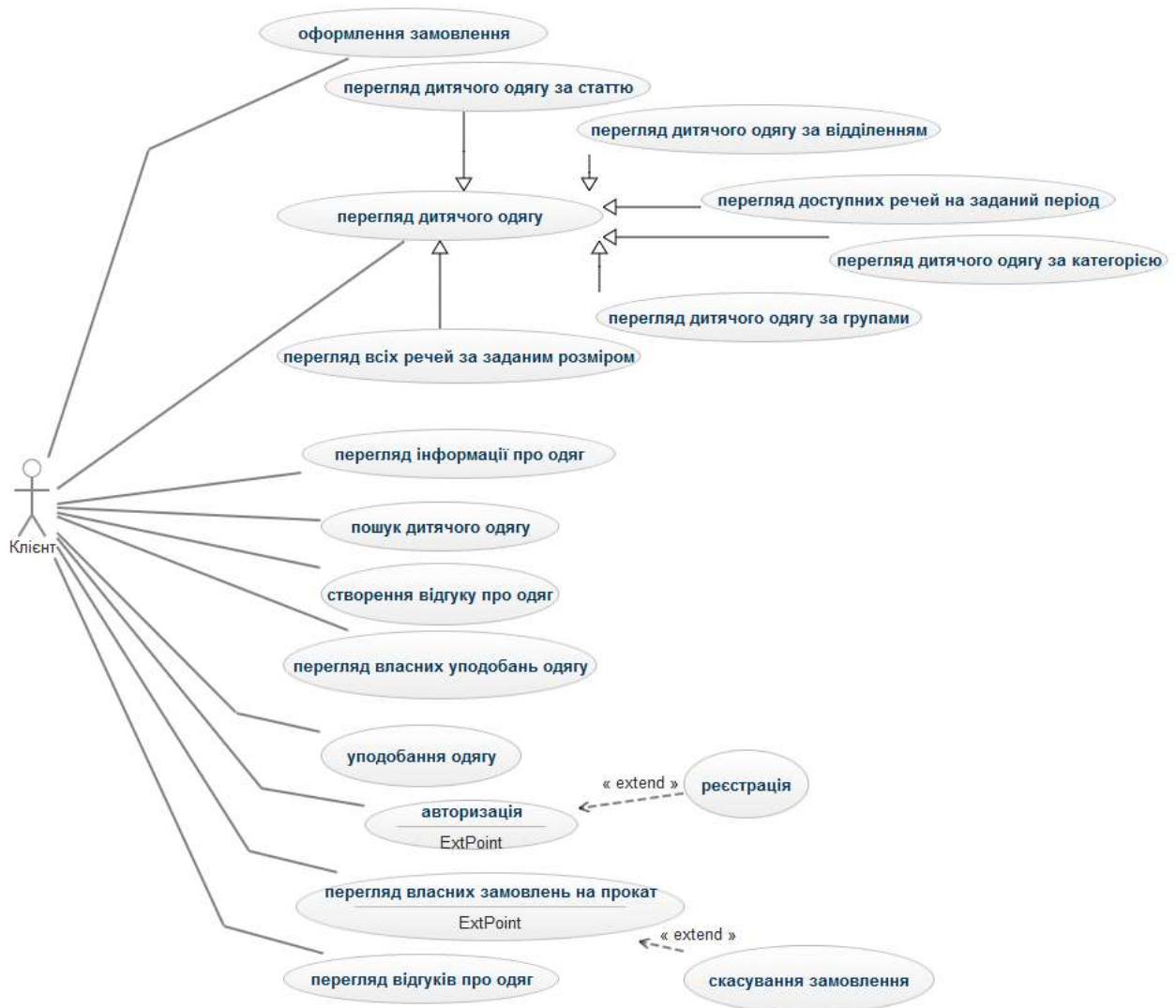


Рисунок 2.2 – Діаграма use case клієнта

2.2 Вибір мови програмування

Як було проаналізовано в розділі 1, подібні рішення створюються у вигляді вебзастосунків, що дозволяє звертатися до програми користувачам без будь-яких обмежень. Враховуючи даний факт, мови програмування PHP [7]-[8] та Python було розглянуто як можливий вибір для розробки.

Порівняння було виконано за наступними критеріями:

- спеціалізація мови: спеціалізованість мови програмування призводить до того, що відповідне рішення простіше створити;

- швидкість виконання: важливо для будь-якого програмного забезпечення, що буде потенційно використовуватися в реальних умовах під навантаженням;
- безкоштовність: безкоштовне, але якісне рішення є основним напрямком в умовах дипломної кваліфікаційної роботи;
- інтеграція з базою даних (БД): в основі предметної області лежить БД, що забезпечує збереження даних, а основні функції, описані вище, організовані на основі звернень до цієї БД.

У результаті порівняння (табл. 2.1) мовою програмування було обрано PHP, оскільки вона більше орієнтована на веброзробку, має вищу швидкість виконання програм, до того ж дещо кращі можливості роботи з БД, що важливо в умовах обмеженого часу розробки [9]-[10].

Таблиця 2.1 – Вибір мови програмування

	Python	PHP
Спеціалізація мови	Мова програмування загального призначення	Спеціалізована для веброзробки мова
Швидкість виконання	Відносно висока	Майже в 3 більша порівняно з PHP, починаючи з версії 7.x
Безкоштовне і відкрите рішення	+	+
Інтеграція з базами даних	+	++

Загалом мова програмування PHP відзначається наступними особливостями:

- простота: мова дуже проста і легка у використанні, загалом широко використовується;

- інтерпретована мова, тобто немає необхідності в компіляції;
- швидкість: швидша ніж інші мови сценаріїв, наприклад, ASP, JSP, Python;
- відкритий код: не потрібно платити за використання PHP, можна безкоштовно завантажити та використовувати;
- незалежність платформи: PHP-код може бути запущений на будь-якій платформі, включаючи Linux, Unix, Mac OS X, Windows;
- повідомлення про помилки: PHP має заздалегідь визначені константи звітування про помилки для створення попередження або повідомлення про помилку;
- моніторинг доступу в режимі реального часу: PHP забезпечує ведення журналу доступу шляхом створення короткого опису нещодавніх звернень для користувача;
- PHP підтримує використання змінних без декларування типу даних: тип буде взятий під час виконання на основі типу даних, якому відповідає значення [11].

Системою керування БД (СКБД) буде використовуватися MySQL, що має наступні позитивні риси:

- безкоштовна з відкритим кодом: за рахунок даного критерію MySQL стає першим вибором для проєктів, які мають ресурсні обмеження, але при цьому MySQL надає майже всі функції, необхідні на будь-якому сервері БД, тому узгодженість та продуктивність програми не порушуються;
- велика громада: хоча MySQL має відкритий код, це не означає, що розробник залишається наодинці з проблемами, величезна інтернет-громада завжди у розпорядженні для пошуку рішень або пошуку найкращого способу зробити щось;
- висока продуктивність, масштабованість, гнучкість: MySQL вважається однією з найшвидших доступних БД, крім того, система також забезпечує багатопоточність для досягнення більш оптимізованої продуктивності, завдяки підтримці вбудованих програм MySQL підходить

для різних випадків використання;

– незалежність від платформи: MySQL не залежить від платформи, тому якщо програма працює на Linux Server, Windows Server або будь-якій іншій операційній системі, це не проблема для MySQL [12]-[13].

2.3 Проєктування бази даних

Враховуючи результати аналізу предметної області, було виділено наступні сутності предметної області.

Сутність User представляє обліковий запис користувача.

Сутність Review представляє відгук про користувача.

Сутність Department представляє відділення прокату.

Сутність City представляє місто, в якому працює прокат.

Сутність Order представляє замовлення на прокат речі.

Сутність Price визначає вартість прокату дитячого одягу.

Сутність Group представляє групу дитячого одягу, що необхідна для категоризації, а таким чином упорядкування дитячого одягу в системі.

Сутність Clothes представляє дитячий одяг (узагальнено, без конкретизації розмірів).

Сутність ClothesElement представляє річ (дитячий одяг з розміром).

Оскільки користувач замовляє конкретні екземпляри речей, які знаходяться в конкретних відділеннях прокату, то було введено таблицю DepartmentClothes для збереження цих даних з одночасним зв'язуванням таблиць Department і ClothesElement.

Поля таблиці User (табл. 2.2):

- id – ідентифікаційний номер облікового запису користувача;
- name – ім'я користувача;
- phone – номер телефону користувача;
- status – стан облікового запису користувача (визначає відключення доступу облікового запису);

- score – середня оцінка облікового запису користувача (на основі відгуків);
- staff – поле, що визначає, чи є користувач звичайним користувачем, чи він є обслуговуючим працівником;
- login – логін користувача;
- password – пароль користувача.

Таблиця 2.2 – Структура таблиці User

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	name	varchar(50)	–	–
3	phone	varchar(13)	–	–
4	status	bool	–	–
5	score	decimal(1,1)	–	–
6	staff	bool	–	–
7	login	varchar(20)	–	–
8	password	varchar(20)	–	–

Поля таблиці Group (табл. 2.3):

- id – номер групи дитячого одягу;
- title – назва групи.

Поля таблиці Category (табл. 2.4):

- id – номер групи дитячого одягу;
- title – назва категорії.

Поля таблиці Clothes (табл. 2.5):

- id – ідентифікаційний номер одягу;
- title – назва одягу;
- group_id – група, до якої належить одяг;
- sex – стать дитини, для якої призначено одяг;

- description – опис одягу;
- photo – шлях до папки з фотографіями;
- likes – сумарна кількість уподобань даного одягу;
- reviews – сумарна кількість відгуків.

Таблиця 2.3 – Структура таблиці Group

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	title	varchar(30)	–	–

Таблиця 2.4 – Структура таблиці Category

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	title	varchar(30)	–	–

Таблиця 2.5 – Структура таблиці Clothes

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	title	varchar(30)	–	–
3	group_id	int	–	+
4	sex	bool	–	–
5	description	text	–	–
6	photo	varchar(255)	–	–
7	likes	int	–	–
8	reviews	int	–	–
9	category_id	int	–	+

Поля таблиці ClothesElement (табл. 2.6):

- id – ідентифікаційний номер речі;
- clothes_id – ідентифікаційний номер дитячого одягу, до якого належить дана річ;
- size – розмір речі;
- quantity – загальна кількість цих речей у прокаті.

Таблиця 2.6 – Структура таблиці ClothesElement

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	clothes_id	varchar(30)	–	+
3	size	int	–	–
4	quantity	bool	–	–

Поля таблиці Department (табл. 2.7):

- id – ідентифікаційний номер відділення;
- title – назва відділення;
- city_id – місто, в якому розташовано відділення.

Таблиця 2.7 – Структура таблиці Department

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	title	varchar(50)	–	–
3	city_id	int	–	+

Поля таблиці City (табл. 2.8):

- id – ідентифікаційний номер міста;
- title – назва міста.

Таблиця 2.8 – Структура таблиці City

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	—
2	title	varchar(30)	—	—

Поля таблиці DepartmentClothes (табл. 2.9):

- id – ідентифікаційний номер екземпляру речі;
- department_id – номер відділення, де знаходиться даний екземпляр;
- clotheselement_id – річ;
- state – стан речі;
- status – доступність речі (необхідно для тих випадків, коли річ знаходиться в ремонті, втрачена або життєвий цикл речі завершився);
- quantity – кількість екземплярів речі.

Таблиця 2.9 – Структура таблиці DepartmentClothes

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	—
2	department_id	int	—	+
3	clotheselement_id	int	—	+
4	state	tinyint	—	—
5	status	bool	—	—
6	quantity	int	—	—

Поля таблиці Order (табл. 2.10):

- id – ідентифікаційний номер замовлення;
- fromdate – дата початку прокату;
- todate – дата завершення прокату;
- user_id – користувач, який оформив замовлення;

- price – сумарна вартість замовлення;
- paid – сплачена сума за замовлення;
- status – стан замовлення;
- creationdate – дата створення замовлення.

Таблиця 2.10 – Структура таблиці Order

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	fromdate	datetime	–	–
3	todate	datetime	–	–
4	user_id	int	–	+
5	price	decimal(1,1)	–	–
6	paid	decimal(1,1)	–	–
7	status	tinyint	–	–
8	creationdate	datetime	–	–

Поля таблиці OrderClothes (табл. 2.11):

- id – ідентифікаційний номер запису;
- departmentclothes_id – екземпляр речі;
- order_id – замовлення;
- quantity – кількість екземплярів речі.

Поля таблиці Price (табл. 2.12):

- id – ідентифікаційний номер тарифу прокату дитячого одягу;
- clothes_id – одяг;
- fromdate – дата початку дії тарифу;
- todate – дата завершення дії тарифу;
- charge – вартість тарифу;
- quantity – період, за який здійснюється прокат.

Таблиця 2.11 – Структура таблиці OrderClothes

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	—
2	departmentclothes_id	int	—	+
3	order_id	int	—	+
4	quantity	int	—	—

Таблиця 2.12 – Структура таблиці Price

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	—
2	clothes_id	int	—	+
3	fromdate	date	—	—
4	todate	date	—	—
5	charge	decimal(1,1)	—	—
6	quantity	int	—	—

Поля таблиці Review (табл. 2.13):

- id – ідентифікаційний номер відгуку про користувача;
- user_id – користувач, на якого написано відгук;
- creationdate – дата створення відгуку;
- score – оцінка відгуку;
- rev – текст відгуку.

Поля таблиці Like (табл. 2.14):

- id – ідентифікаційний номер уподобання;
- user_id – користувач, який поставив уподобання;
- clothes_id – дитячий одяг, який уподобано.

Поля таблиці ClothesReview (табл. 2.15):

- id – ідентифікаційний номер відгуку;

- user_id – користувач, який написав відгук;
- clothes_id – дитячий одяг, на який написано відгук;
- rev – текст відгуку;
- creationdate – дата створення відгуку.

Таблиця 2.13 – Структура таблиці Review

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	user_id	int	–	+
3	creationdate	datetime	–	–
4	score	int	–	–
5	rev	text	–	–

Таблиця 2.14 – Структура таблиці Like

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	user_id	int	–	+
3	clothes_id	int	–	+

Таблиця 2.15 – Структура таблиці ClothesReview

Номер поля	Назва поля	Тип поля	Первинний ключ	Зовнішній ключ
1	id	int	+	–
2	user_id	int	–	+
3	clothes_id	int	–	+
4	rev	text	–	–
5	creationdate	datetime	–	–

Схему БД приведено на рис. 2.3.

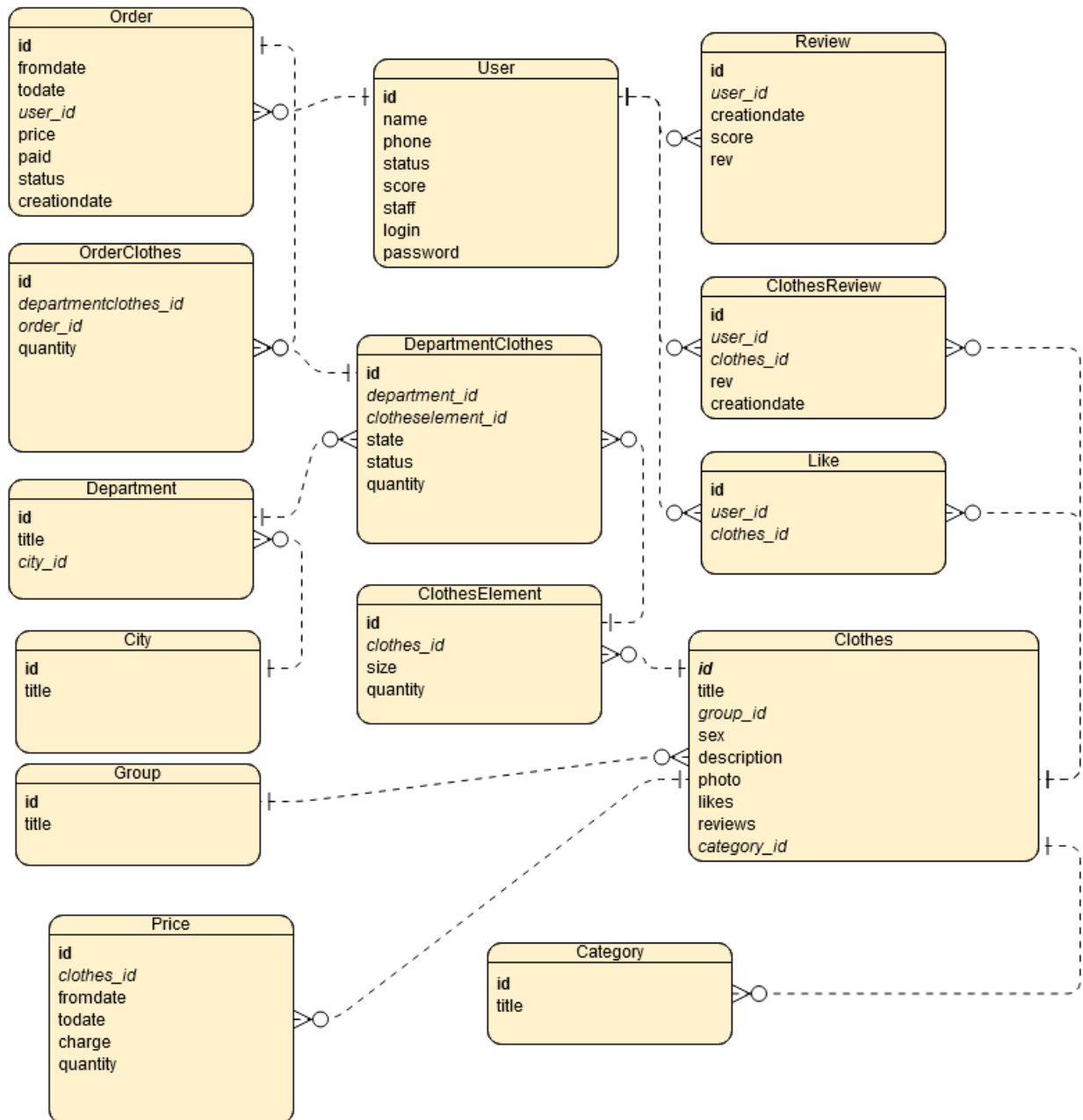


Рисунок 2.3 – Схема БД

Таблиці Group і Clothes, Clothes і ClothesElement, City і Department, Department і DepartmentClothes, ClothesElement і DepartmentClothes, User і Order, OrderClothes і DepartmentClothes, Order і OrderClothes, Clothes і Price, User і Review, User і Like, Clothes і Like зв'язані зв'язком один-до-багатьох.

Таблиці OrderClothes і Like реалізують зв'язки багато-до-багатьох між таблицями Order і DepartmentClothes, User і Clothes.

2.4 Висновки за розділом 2

У другому розділі визначено функціональні вимоги до програми, обрано мову програмування PHP і СКБД MySQL для реалізації програмного забезпечення, виконано проєктування БД.

3 РОЗРОБКА ПРОГРАМИ

3.1 Алгоритм функціонування програмного забезпечення

Робота програми розпочинається з авторизації облікового запису. На цьому етапі користувач повинен задати власні логін і пароль, які у випадку співпадіння за даними БД визначають його роль у системі: клієнт або прокатник.

Якщо в системі авторизувався прокатник, то йому надається меню, з якого він може обрати один з варіантів роботи, що включають:

- керування записами про дитячий одяг, де відображається весь дитячий одяг і надається можливість змінювати існуючий одяг, додавати новий або переходити до конкретних речей та редагувати їх, тобто у цій процедурі реалізується загальне (без врахування відділення) керування дитячим одягом;

- керування замовленнями, де відображається перелік створених замовлень і можна підтвердити, скасувати або редагувати замовлення;

- визначення дитячого одягу у відділенні, де можна внести дані про наявність одягу відповідних розмірів у відділенні;

- пошук дитячого одягу, де за набором параметрів можна виконати пошук дитячого одягу;

- обрати один з параметрів перегляду, що включають стать, відділення, категорію, групу, розмір, і побачити набір речей, що відповідають заданому значенню параметру перегляду.

Користувач може в будь-який момент завершити роботу з програмою шляхом закриття браузера або виходом з облікового запису. Після виходу з облікового запису користувач має заново авторизуватися, тобто це призводить до повторення дій, які були зазначені в даному алгоритмі.

На рис. 3.1 представлена схема функціонування програми, яку було описано вище.

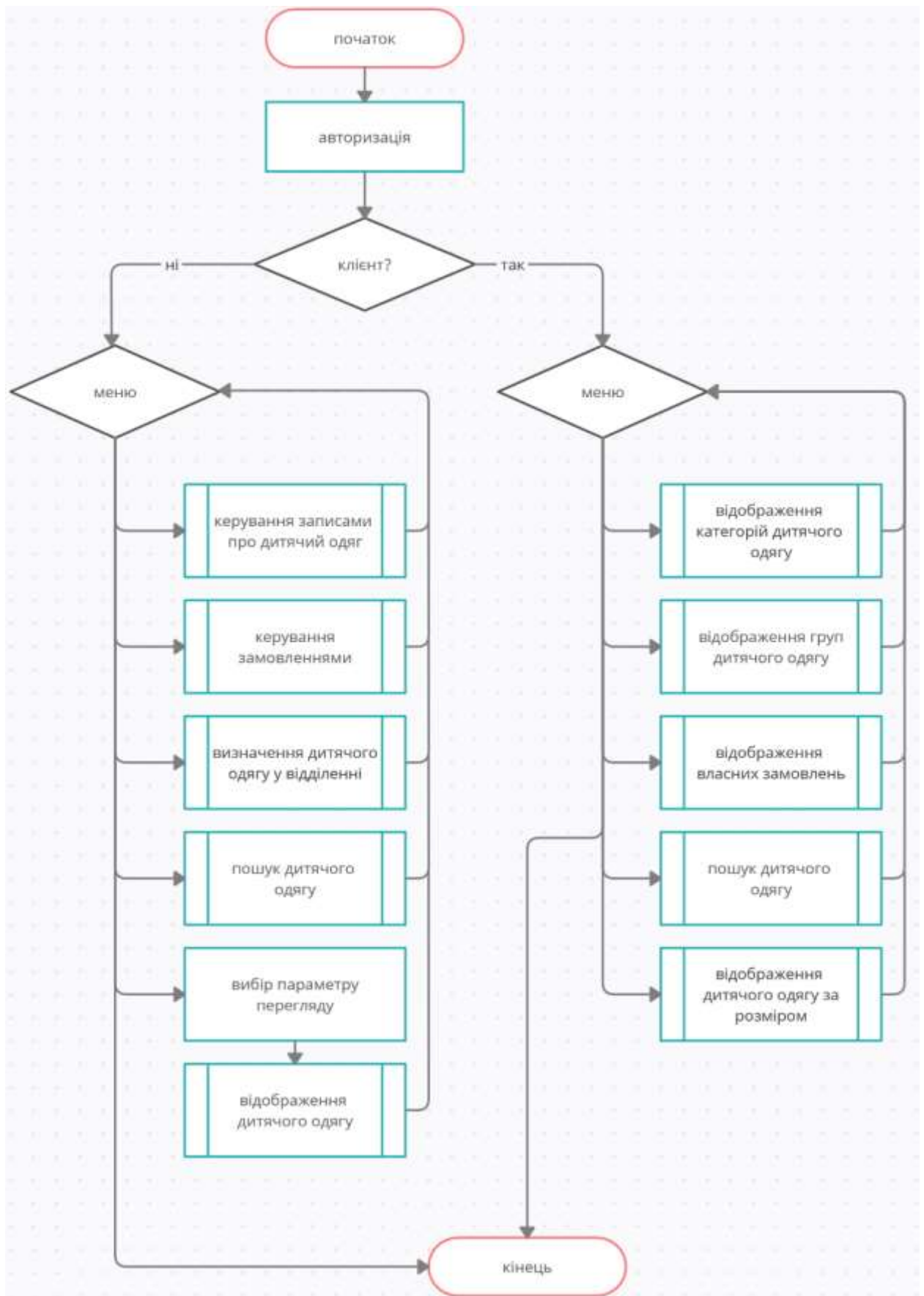


Рисунок 3.1 – Схема функціонування програми

Якщо в системі авторизувався клієнт, то йому надається меню, з якого він може обрати один з варіантів роботи, що включають:

- відображення категорій дитячого одягу, звідки він може обрати потрібну категорію, а за обраною категорією йому відображається перелік всього одягу, що підпадає під даний критерій;
- відображення груп дитячого одягу, звідки він може обрати потрібну групу, а за обраною групою йому відображається перелік всього одягу, що підпадає під даний критерій;
- відображення власних замовлень, звідки клієнт може побачити всі замовлення, які були ним створені, а також скасувати власне замовлення до моменту його підтвердження;
- пошук дитячого одягу, звідки клієнт може реалізувати розширений пошук потрібного дитячого одягу;
- відображення дитячого одягу за розміром, звідки клієнт може задати потрібний розмір та відобразити весь одяг, який представлений даним розміром.

3.2 Структура розробленого програмного забезпечення

Для забезпечення реалізації описаного алгоритму та визначених функціональних вимог в структурі програми було виділено наступні моделі:

- RenterModel – клас моделі підтримки роботи прокатника;
- BaseModel – базовий клас моделі, який реалізує формування даних для засобів, спільних для прокатника і користувача;
- UserModel – клас моделі звичайного користувача (клієнта);
- RequesterModel – клас для роботи з даними щодо виконання перевірки наявності потрібної речі, визначення тарифів;
- AuthModel – клас моделі авторизації облікового запису;
- AuthOutModel – клас моделі виходу з облікового запису;
- RegModel – клас моделі реєстрації облікового запису.

Представлення реалізовані за допомогою шаблону [14] і включають наступні:

- newclothes.phtml – представлення внесення нового дитячого одягу;
- newclotheselement.phtml – представлення внесення нової дитячої речі;
- newreview.phtml – представлення створення нового відгуку на користувача;
- newdepclothes.phtml – представлення внесення нового запису про наявність дитячого одягу у відділенні;
- newprice.phtml – представлення створення нового тарифу на прокат;
- editprice.phtml – представлення редагування тарифу на прокат;
- user.phtml – представлення перегляду даних користувача прокатником;
- reviews.phtml – представлення перегляду відгуків;
- orders.phtml – представлення перегляду замовлень;
- editclothes.phtml – представлення редагування дитячого одягу;
- auth.phtml – представлення авторизації облікового запису;
- order.phtml – представлення перегляду даних замовлення;
- categclothes.phtml – представлення перегляду речей за категорією;
- prices.phtml – представлення перегляду тарифів на прокат;
- clothes.phtml – представлення перегляду інформації про дитячий одяг;
- groups.phtml – представлення перегляду дитячого одягу за групою;
- departments.phtml – представлення перегляду дитячого одягу за відділенням;
- search.phtml – представлення пошуку дитячого одягу;
- sizes.phtml – представлення перегляду дитячого одягу за розміром;
- newclothreview.phtml – представлення створення нового відгуку про дитячий одяг.

Структурна схема програми представлена на рис. 3.2.

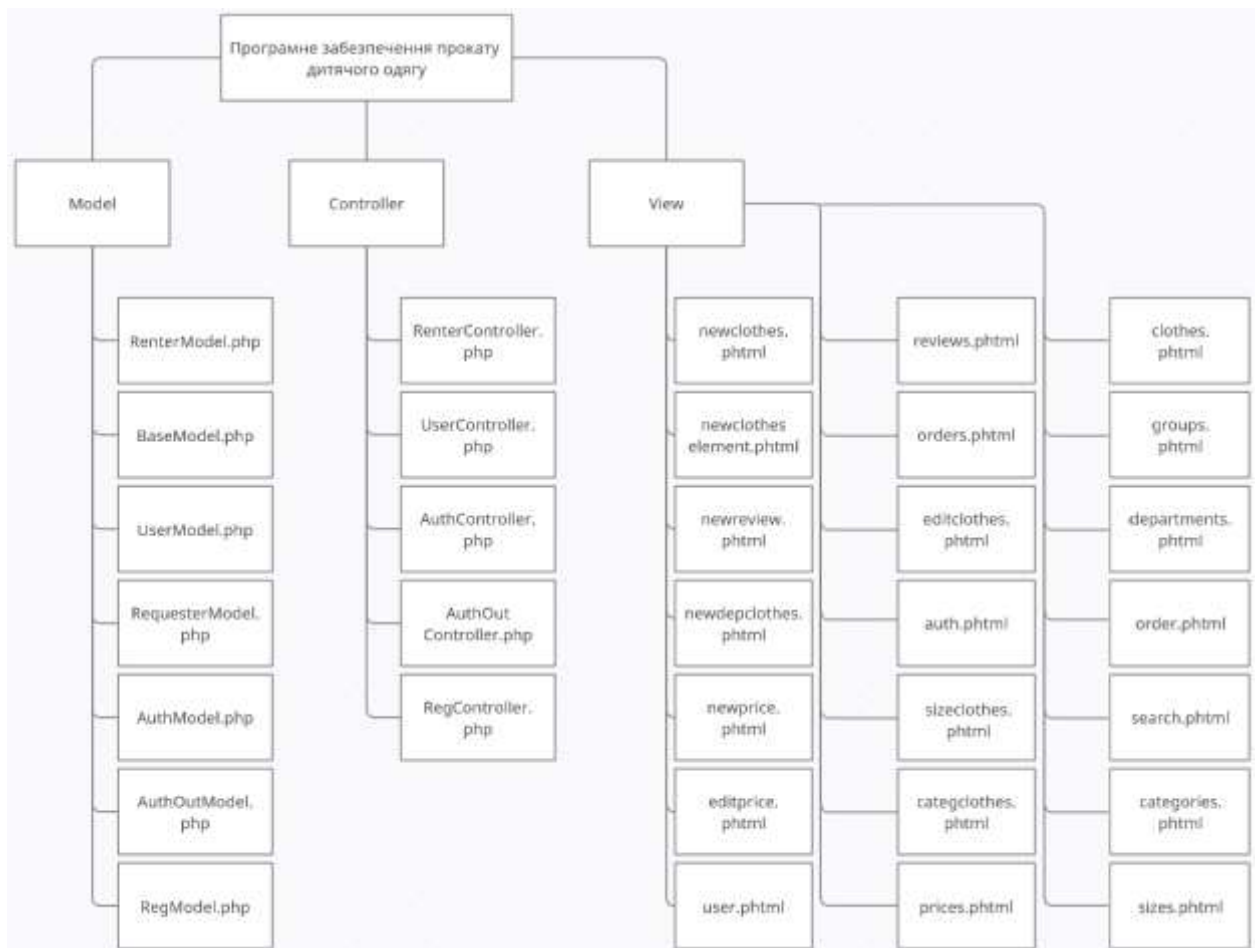


Рисунок 3.2 – Структурна схема програми

3.3 Опис прийнятих під час реалізації рішень

У складі класу моделі прокатника RenterModel було реалізовано наступні методи.

Метод newClothes(\$clothes) реалізує створення нового запису про дитячий одяг, що відповідає набору характеристик clothes.

Метод newClothesElement(\$clothes) вносить дані про прибуття нових речей, що відповідають набору характеристик clothes.

Метод newReview(\$review) реалізує створення нового відгуку на користувача review.

Для реалізації цих дій спочатку вносяться за допомогою INSERT-запиту дані самого відгуку:

```

$db = DB::getConnection();
$req = $db->prepare('INSERT INTO Review (score,
    rev,
    user_id,
    creationdate);
VALUES (:score,
    :rev,
    :user_id,
    :creationdate)');
$req->execute($review);

```

На наступному кроці відбувається обчислення середньої оцінки користувача на основі всіх відгуків про нього:

```

$req = $db->prepare('SELECT AVG(score)
    FROM Review
    WHERE user_id = :user_id');
$req->bindParam('user_id', $review['user_id']);
$req->execute();
$req = $req->fetchAll(PDO::FETCH_ASSOC);

```

Після цього відбувається оновлення оцінки користувача:

```

$res = $db->prepare('UPDATE User SET score = :score');

```

Таким чином, переобчислення оцінки користувача реалізується автоматично після створення кожного нового відгуку на користувача.

Метод `newDepartmentClothes($clothes)` реалізує внесення даних про наявність речей `clothes` у відділенні.

Метод `newPrice($price)` реалізує внесення даних про новий тариф прокату речі `price`.

Метод `editPrice($price)` забезпечує редагування існуючого тарифу прокату за значеннями `price`.

Метод `check($clothes_id, $fromdate, $todate, $department, $quantity)` виконує перевірку доступності речі з номером `clothes_id` на період з дати `fromdate` до дати `todate` у відділенні `department` у потрібній кількості `quantity` екземплярів.

Метод `getUser($id)` повертає інформацію про користувача з номером `id`, що необхідно для роботи прокатника і ознайомлення з клієнтом. До результатів включаються дані профіля користувача та набір відгуків.

Метод `getOrders()` повертає перелік всіх замовлень на прокат, відсортованих, починаючи з найближчої до поточного моменту дати створення замовлення.

Метод `getDepartmentClothes($id)` повертає весь одяг, який є в наявності відділення з номером `id`.

Метод `editClothes($clothes_id, $clothes)` виконує редагування дитячого одягу з номером `clothes_id`, вносячи інформацію з `clothes`.

Метод `cancelOrder($order_id)` скасовує замовлення з номером `order_id`.

Метод `payOrder($order_id, $paid)` реалізує сплату за замовлення з номером `order_id`, вносячи дані про сплату в сумі `paid`.

Метод `submitOrder($order_id)` реалізує підтвердження замовлення з номером `order_id`.

У складі класу `UserModel` моделі звичайного користувача (клієнта) реалізовано наступні методи.

Метод `newClothesReview($review)` реалізує створення нового відгуку `review` про дитячий одяг.

Метод `getMyOrders()` повертає перелік власних замовлень на прокат, відсортованих, починаючи з найближчої до поточного моменту дати створення замовлення.

Метод `cancelOrder($order_id)` скасовує замовлення з номером `order_id` у випадку, якщо воно знаходиться не в підтвердженому стані. Якщо замовлення знаходиться в підтвердженому стані, то клієнт вже не може вплинути на замовлення.

Метод `Like($clothes_id)` реалізує уподобання авторизованим у даний момент користувачем дитячого одягу з номером `clothes_id`.

Метод `Unlike($clothes_id)` реалізує скасування уподобання авторизованим у даний момент користувачем дитячого одягу з номером `clothes_id`.

І модель клієнта, і модель прокатника базуються на класі `BaseModel`.

У складі класу моделі `BaseModel` було реалізовано наступні методи. Це методи, які використовуються обома класами і включають засоби перегляду дитячого одягу, пошуку його.

Метод `newOrder($order, $orderclothes, $department)` забезпечує створення нового замовлення на прокат дитячих речей `orderclothes` у вигляді замовлення `order` у відділенні `department`. Тобто даний метод працює одразу з двома таблицями, що визначають саме замовлення і склад даного замовлення.

Обчислення вартості замовлення відбувається шляхом перебору речей зі складу замовлення `orderclothes` з одночасною перевіркою доступності цих речей на вказану дату, при цьому якщо хоча б одна річ буде недоступна, то значення змінної `check` буде дорівнювати `false` і в результаті замовлення не буде створено, а шляхом оброблення повернутого методом значення `false` у контролері буде виведено повідомлення про помилку створення замовлення:

```
$check = true;
foreach ($orderclothes as $value)
{
    $price = $this->getClothesPrice($value['clothes_id'], $order['fromdate'],
    $order['todate']);
```

```

$sum = $sum + $price * $value['quantity'];
$check = $check and this->checkAvailability($value['clothes_id'],
$order['fromdate'], $order['todate'], $department, $value['quantity']);
}
$order['price'] = $sum;

```

Метод getCategoryClothes(\$category) повертає набір речей, що відповідають категорії з номером category.

Метод getPrices(\$id) повертає тарифи на прокат речі з номером id, які відповідають поточній даті.

Метод getClothes(\$id) повертає значення набору характеристик, що визначають дитячий одяг з номером id. Набір характеристик включає дані розмірів цієї речі, опис самої речі.

Метод getGroups() повертає набір всіх доступних груп дитячого одягу.

Метод getDepartments() повертає перелік всіх відділень сервісу.

Метод getOrder(\$id) повертає повні дані замовлення з номером id.

Метод searchClothes(\$department_id, \$sex, \$group, \$category, \$size, \$title) виконує пошук речей з визначенням відділення department_id, статі дитини, для якої виконується пошук речі, sex, групи речей group, категорії речей category, розміру size та назви речі title.

Метод getSexClothes(\$sex) повертає перелік всього дитячого одягу за статтю дитини sex.

Метод getSizeClothes(\$size) повертає перелік всього дитячого одягу з розміром size.

Метод getGroupClothes(\$group) повертає перелік всього дитячого одягу за групою group.

Метод getCategories() повертає перелік всіх наявних категорій.

3.4 Висновки за розділом 3

Реалізовано вебзастосунок для підтримки прокату дитячого одягу. Описано алгоритм функціонування програмного забезпечення з приведенням відповідної схеми функціонування, описано структуру створеного програмного забезпечення. Представлено основні розроблені в процесі реалізації класи.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення програми

Програма забезпечує підтримку замовлення прокату дитячого одягу. Для цього вона дозволяє перевіряти доступність дитячого одягу для прокату на задані дати, створювати власні замовлення, прокатникам керувати створеними користувачами замовленнями, створювати каталог дитячого одягу, вносити дані про наявність дитячого одягу у відповідних відділеннях прокату.

4.2 Умови виконання програми

Серверна частина програмного забезпечення має відповідати наступним мінімальним вимогам:

- вільний простір на жорсткому диску – 10 Гб, що обумовлено вимогами до організації супутнього програмного забезпечення та необхідністю завантаження фотографій для демонстрації дитячого одягу;
- оперативна пам'ять – 4 Гб;
- процесор: 4 ядра, 2,8 ГГц частота;
- підключення до мережі інтернет.

4.3 Виконання програми

Для виконання програми необхідно запустити вебсервер та СКБД. Якщо вебсервер та СКБД запущено, то можна виконувати запити за адресою вебсервера через браузер. Одразу після звернення до вебсервера надається початкова сторінка вебзастосунку, основою якої є форма авторизації.

Після авторизації клієнта йому відображається меню в лівій панелі.

Меню складається з наступних пунктів меню:

- «Пошук» для пошуку дитячого одягу;
- «Категорії» для перегляду речей за заданою категорією;
- «Групи» для перегляду речей за заданою групою;
- «Замовлення» для перегляду власних замовлень;
- «Розміри» для перегляду речей за заданим розміром;
- «Вийти» для виходу з облікового запису.

За допомогою меню можна відкрити пункт меню «Категорії», тоді клієнту буде показано вебсторінку категорій дитячого одягу (рис. 4.1). На ній відображається карусель, що містить по 3 категорії за кожного положення. Якщо навести на зображення категорії, то з'явиться посилання «Переглянути», яке дозволяє перейти до дитячого одягу відповідної категорії. До категорій в застосунку належать куртки, сукні, взуття, святкові костюми, спортивний одяг тощо.

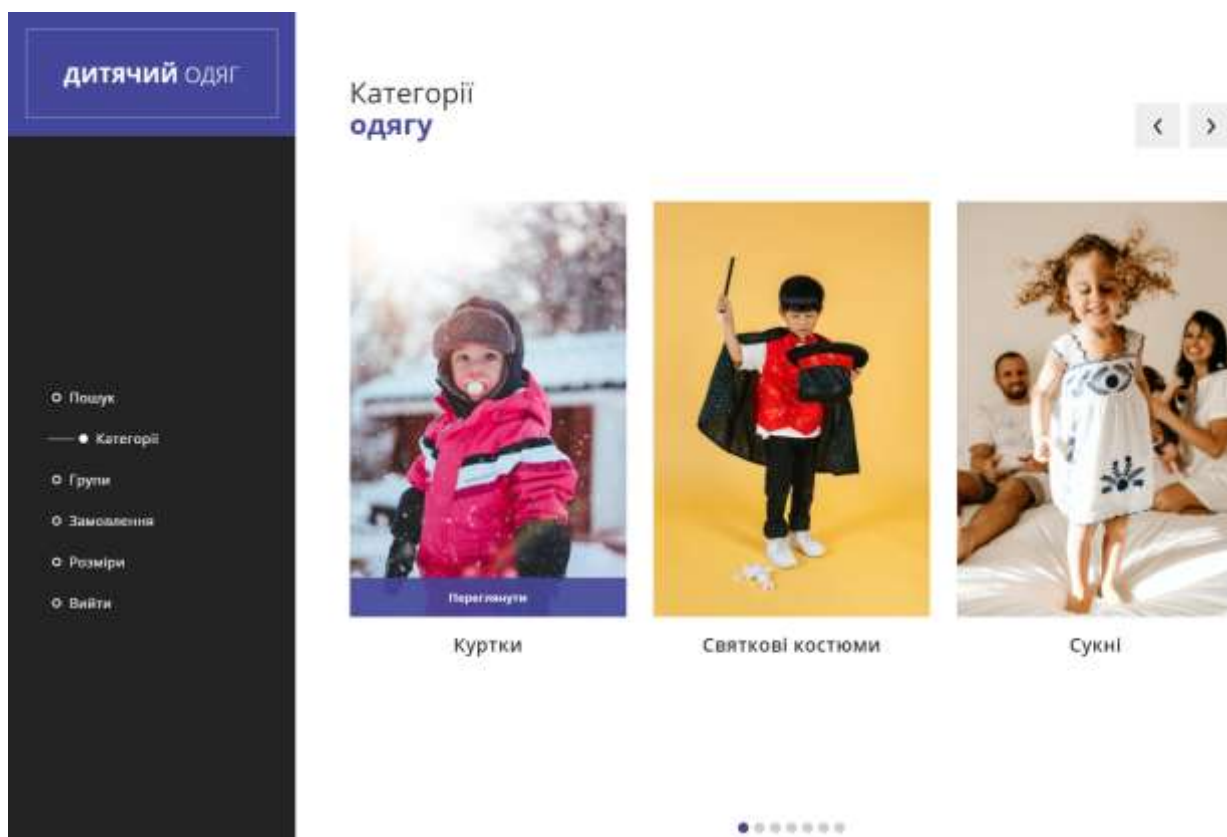


Рисунок 4.1 – Перегляд категорій дитячого одягу

Якщо з будь-якого переліку речей обрати одну з речей, то відкриється

вебсторінка відповідної речі (рис. 4.2).

На відповідній вебсторінці відображається:

- назва речі;
- опис речі;
- категорія речі;
- група речі (за наявності);
- кількість уподобань відповідного дитячого одягу;
- можливість уподобати або скасувати уподобання (якщо вона вже була уподобана даним користувачем) речі;
- кількість відгуків, що була створена для даної речі, та можливість перейти до читання відгуків за натискання на відповідну кнопку «Відгуки»;
- вартість прокату за мінімальну тривалість з усіх наявних тарифів у поточний момент часу;
- можливість за допомогою кнопки «Тарифи» перейти до переліку всіх наявних тарифів за цією річчю;
- доступні розміри для даної речі;
- можливість перевірити наявність речі.

Наявність вказаної речі можна перевірити на заданий період часу (обирається шляхом вибору з календаря дати початку прокату і дати завершення прокату) у заданому відділенні (потрібно обрати необхідне відділення з випадаючого переліку) та у заданому розмірі (обирається шляхом натискання кнопки відповідного з доступних за даною річчю розміру).

Після того, як натиснути на кнопку «Наявність», з'явиться нижче інформація про наявність речі за цими параметрами та вартість такого прокату. Якщо річ недоступна за такими параметрами, то відобразиться повідомлення про відсутність речі.

Якщо клієнт бажає замовити річ, то йому також потрібно задати ці параметри, а після натиснути на кнопку «Замовити».

Клієнту також доступна можливість перегляду власних замовлень у вигляді таблиці, що дозволяє за необхідності скасувати замовлення, яке ще не було підтверджено, а також впорядкувати власні замовлення.

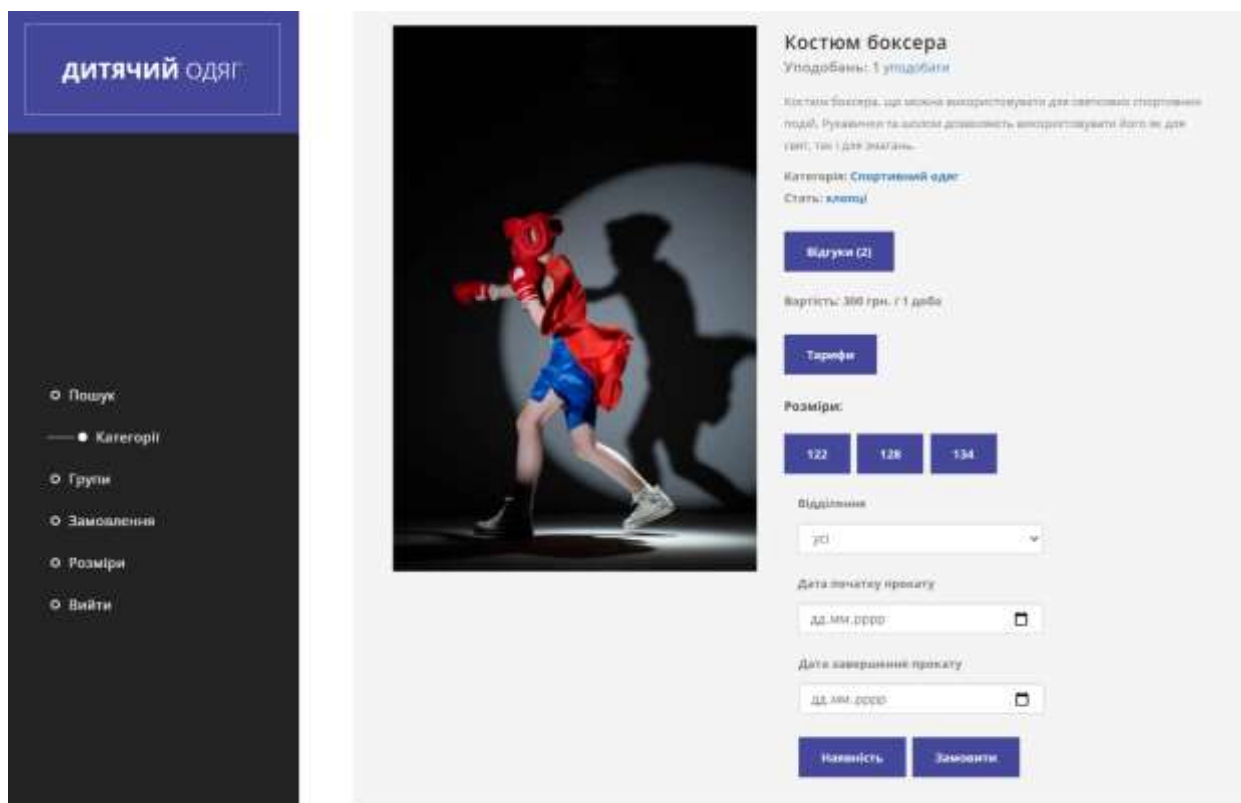


Рисунок 4.2 – Перегляд інформації про річ

Якщо в програмі авторизувався прокатник, то йому доступне дещо видозмінене меню. Прокатнику доступний набір варіантів перегляду дитячого одягу, які він може подібно до клієнта використовуватися для обслуговування клієнтів:

- «Пошук» для пошуку дитячого одягу;
- «Відділення» для керування речами, наявними у відділенні;
- «Одяг» для керування дитячим одягом, який обслуговується в прокаті;
- «Замовлення» для керування замовленнями клієнтів, що дозволяє керувати вже створеними замовленнями;
- «Перегляд» для перегляду всього дитячого одягу;

– «Вийти» для виходу з облікового запису.

Усі варіанти перегляду дитячого одягу для прокатника об'єднані в пункт меню «Перегляд». При відкритті даного пункту відображається вебсторінка, на якій доступний набір закладок, що включає:

- «Групи» (рис. 4.3) для перегляду груп дитячого одягу та вибору потрібної групи;
- «Категорії» для перегляду груп дитячого одягу та вибору потрібної категорії;
- «Стать» для перегляду груп дитячого одягу та вибору потрібної статі;
- «Розмір» для перегляду груп дитячого одягу та вибору потрібного розміру.

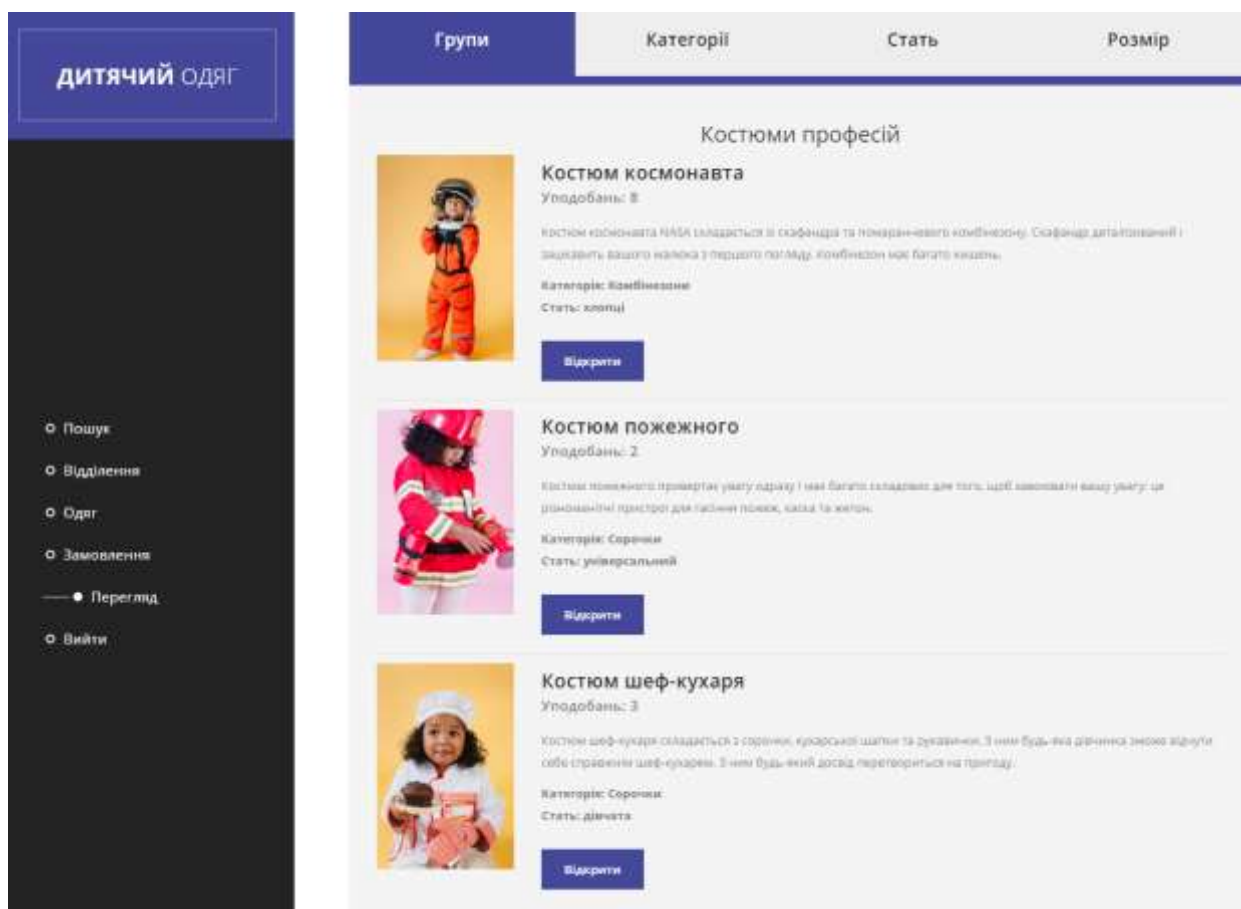


Рисунок 4.3 – Перегляд дитячого одягу за обраною групою одягу

Аналогічно розглянутому варіанту з категоріями у клієнта, якщо

обрати пункт меню «Групи», то відкриється сторінка з переліком всіх груп дитячого одягу. Після вибору потрібної групи буде відображено весь дитячий одяг за обраною групою. Назва групи відображається вгорі вкладки (рис. 4.3).

Усі відповідні позиції дитячого одягу відображаються таблицею нижче. За кожною позицією відображається фотографія речі, її назва, кількість уподобань, опис речі, назва категорії, статі. Для того щоб уподобати річ або перейти до всіх речей відповідної категорії або за статтю, то потрібно спочатку відкрити сторінку перегляду речі, для чого натиснути на кнопку «Відкрити».

Пункт меню «Пошук» дозволяє відкрити вебсторінку пошуку дитячого одягу (рис. 4.4).

Рисунок 4.4 – Пошук дитячого одягу

На даній сторінці потрібно ввести назву речі, задати статі (хлопець,

дівчина або універсальний), групу, категорію, розмір, відділення. Після цього потрібно натиснути на кнопку «Шукати», щоб внизу було відображено перелік всіх знайдених речей. У випадку пошуку для кожної речі також відображається група, до якої вона належить.

При відкритті пункту меню «Замовлення» відкривається вебсторінка керування замовленнями прокатником (рис. 4.5). На цій сторінці відображається перелік всіх замовлень, які були виконані клієнтами. Усі вони відсортовані за зменшенням ідентифікаційного номеру замовлення.

За допомогою кнопки «Створити замовлення» прокатник може створити нове замовлення, якщо клієнт оформлює його одразу через прокатника, без власної участі.

Для кожного замовлення відображається:

- номер, який містить посилання і дозволяє перейти до сторінки відображення повних даних замовлення, зокрема звідси доступна можливість створення замовлення про клієнта прокатником;
- логін користувача, який замовив прокат, що містить посилання на вебсторінку користувача, де міститься оцінка клієнта та всі створені відгуки про нього;
- період виконання замовлення, включаючи дати початку прокату та завершення прокату;
- кнопки керування замовленнями.

Кнопки керування замовленнями в залежності від стану замовлення змінюються.

Якщо замовлення тільки створене, при чому створене самим клієнтом, то відображається кнопка «Підтвердити», що дозволяє підтвердити замовлення речі клієнтом, після цього дана дитяча річ буде вважатися зайнятою на вказані дати. Також доступна кнопка «Скасувати», яка дозволяє скасувати замовлення без підтвердження, а також скасувати замовлення в будь-який наступний момент за необхідності. Дана кнопка відсутня тоді, коли замовлення виконано.

Для завершення роботи над замовленням потрібно після того, як воно підтверджене було, натиснути на кнопку «Завершити».

Кнопка «Сплатити» дозволяє перейти до сторінки внесення сплати за замовлення, де можна ввести суму, яку ввів користувач. Дану суму можна змінювати до моменту завершення замовлення.

Замовлення на прокат одягу									
Створити Замовлення									
N	Користувач	Річ	Період	Вартість	Сплачено				
24	client22	Стильована куртка	20.05.2021-21.05.2021	200 грн.	0 грн.	Підтвердити	Скасувати	Сплатити	
23	client11	Костюм пожежного	20.06.2021-30.06.2021	2000 грн.	0 грн.	Підтвердити	Скасувати	Сплатити	
22	client11	Костюм лікаря	23.05.2021-25.05.2021	150 грн.	0 грн.	Підтвердити	Скасувати	Сплатити	
21	user2	Костюм супермена	18.05.2021-28.05.2021	1800 грн.	200 грн.	Завершити	Скасувати	Сплатити	
20	user	Спортивний костюм Nike	18.05.2021-20.05.2021	300 грн.	300 грн.	Завершити	Скасувати	Сплатити	
19	client11	Льняна сорочка	17.05.2021-18.05.2021	150 грн.	150 грн.	Завершити	Скасувати	Сплатити	
18	user	Костюм шеф-кухаря	17.05.2021-18.05.2021	200 грн.	200 грн.	Завершити	Скасувати	Сплатити	
17	client22	Сукня чарівниці	15.05.2021-17.05.2021	600 грн.	600 грн.				
16	user2	Костюм космонавта	15.05.2021-16.05.2021	400 грн.	400 грн.				
15	client11	Червоний попелашки	14.05.2021-17.05.2021	1200 грн.	1200 грн.				

Рисунок 4.5 – Керування замовленнями

ВИСНОВКИ

У дипломній кваліфікаційній роботі виконано розроблення програмного забезпечення для підтримки роботи з клієнтами прокату дитячого одягу.

Виконано аналіз предметної області, що включив аналіз програмних засобів, які можуть вирішувати проблему, зокрема вебсайтів olx.ua, zakupka.com, gup.ua, barvu.net, а також виконано порівняння аналогів і виділено основні функції, які в якості переваг над альтернативи має виконувати розробка.

Визначено функціональні вимоги до програми, на основі чого побудовано діаграми use case для роботи клієнта та прокатника з програмою. Обґрунтовано вибір мови програмування PHP і СКБД MySQL для реалізації програмного забезпечення. Виконано проєктування БД, в результаті описано структуру створеної БД та представлено схему БД.

Реалізовано вебзастосунок для підтримки прокату дитячого одягу. Описано алгоритм функціонування програмного забезпечення з приведенням відповідної схеми функціонування, описано структуру створеного програмного забезпечення. Представлено основні розроблені в процесі реалізації класи.

Розроблене програмне забезпечення дозволяє перевіряти доступність дитячого одягу для прокату на задані дати, створювати власні замовлення, прокатникам керувати створеними користувачами замовленнями, створювати каталог дитячого одягу, вносити дані про наявність дитячого одягу у відповідних відділеннях прокату.

Завдання дипломної кваліфікаційної роботи в результаті повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Barton, J. 6 best kids' clothing rental services for shopping more sustainably [Electronic resource] / J. Barton. – Access mode : <https://www.independent.co.uk/extras/indybest/kids/clothing-footwear/rent-kids-clothes-baby-uk-b1807190.html>.
2. Jackson, L. The Rental Revolution : Would You Rent Your Child's Clothes? [Electronic resource] / L. Jackson. – Access mode : <https://www.fashionroundtable.co.uk/news/2020/9/15/the-rental-revolution-would-you-rent-your-kids-clothes>.
3. Оголошення OLX.ua : сервіс оголошень України – купівля/продаж нових та бу товарів, різноманітні послуги на сайті OLX.ua [Електрон. ресурс]. – Режим доступу : <https://www.olx.ua/>.
4. Zakupka.com – каталог інтернет-магазинів, товарів, услуг и цен Украины [Электрон. ресурс]. – Режим доступа : <https://zakupka.com/>.
5. Прокат карнавальних костюмів в м. Львів "Барви дитинства" [Електрон. ресурс]. – Режим доступу : <https://barvu.net/>.
6. Оренда і прокат будь-яких товарів на GUP : розмістити Вашу пропозицію [Електрон. ресурс]. – Режим доступу : <https://gur.ua/>.
7. Nixon, R. Learning PHP, MySQL & JavaScript [Text] : With jQuery, CSS & HTML5 / R. Nixon. – Sebastopol : O'Reilly, 2018. – 832 p.
8. Tatroe, K. Programming PHP [Text] : Creating Dynamic Web Pages / KL. Tatroe, P. MacIntyre. – Sebastopol : O'Reilly, 2020. – 544 p.
9. Goel, A. Python vs PHP in 2021 [Electronic resource] / A. Goel. – Access mode : <https://hackr.io/blog/python-vs-php>.
10. Python Vs PHP : What's the Difference? [Electronic resource]. – Access mode : <https://www.guru99.com/python-vs-php.html>.
11. Features of PHP – PHP Tutorial [Electronic resource]. – Access mode : <https://www.sitesbay.com/php/php-features-of-php>.
12. MySQL Advantages and Disadvantages [Electronic resource]. –

Access mode : <https://blueclawdb.com/mysql/advantages-disadvantages-mysql/>.

13. Васвани, В. MySQL : использование и администрирование [Текст] / В. Васвани. – М. : Питер, 2011. – 368 с.

14. Free 518 Template Sentra [Electronic resource]. – Access mode : <https://templatemo.com/tm-518-sentra>.

ДОДАТОК А
Технічне завдання

Вступ

У програмі має бути забезпечено підтримку прокату дитячого одягу в сервісі прокату, який може мати відділення у різних містах або декілька відділень у одному місті. Користувачами програми повинні бути клієнти та працівники сервісу.

A.1 Підстави для розробки

Підставою для розробки було завдання на дипломну кваліфікаційну роботу за темою «Розробка програмного забезпечення прокату дитячого одягу» на основі наказу № 103 від 30 березня 2021 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Розроблювана програма має забезпечити підтримку замовлення прокату дитячого одягу. Для цього має виконуватися заповнення даними дитячого одягу (створення каталогу) та оформлення замовлень.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Розроблювана програма повинна забезпечувати наступні функціональні характеристики:

- внесення наявного дитячого одягу;
- редагування інформації про дитячий одяг;
- перегляд наявного загалом і у відділенні дитячого одягу;
- перегляд замовлень на прокат за відділенням;
- оформлення замовлення на прокат речі;

- пошук потрібного дитячого одягу;
- перегляд всіх речей за заданим розміром;
- перегляд дитячого одягу за категорією;
- перегляд дитячого одягу за групами;
- перегляд дитячого одягу за статтю;
- перегляд дитячого одягу за відділенням;
- перегляд власних замовлень на прокат;
- скасування замовлення на прокат;
- підтвердження замовлення на прокат;
- перегляд доступних речей на заданий період;
- реєстрація облікового запису клієнта;
- авторизація облікового запису;
- створення відгуку про одяг;
- створення відгуку про клієнта;
- перегляд відгуків про клієнта;
- перегляд відгуків про одяг;
- перегляд інформації про одяг;
- уподобання одягу;
- перегляд власних уподобань одягу.

A.3.2 Вимоги до надійності

Надійність роботи програми має забезпечуватися надійною роботою вебсервера, СКБД та обробленням даних, отриманих від користувачів для уникнення некоректностей та спотворення даних.

A.3.3 Умови експлуатації

Програма має експлуатуватися шляхом розміщення вебзастосунку на сервері. Для підтримки роботи програми та системи керування базами даних

має бути виділено системного адміністратора.

A.3.4 Вимоги до складу та параметрів технічних засобів

Серверна частина має відповідати наступним мінімальним вимогам:

- вільний простір на жорсткому диску – 10 Гб, що обумовлено вимогами до організації супутнього програмного забезпечення та необхідністю завантаження фотографій для демонстрації дитячого одягу;
- оперативна пам'ять – 4 Гб;
- процесор: 4 ядра, 2,8 ГГц частота;
- підключення до мережі інтернет.

A.4 Порядок контролю та приймання

Контроль за виконанням дипломної кваліфікаційної роботи покладається на керівника. Остаточне рішення приймається екзаменаційною комісією під час захисту роботи.

ДОДАТОК Б
Текст програми

Б.1 Текст файла RenterModel.php

```

<?php

class RenterModel extends BaseModel
{
    public function newClothes($clothes)
    {
        $clothes['likes'] = 0;
        $clothes['reviews'] = 0;
        unset($clothes['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('INSERT INTO Clothes (title,
            group_id,
            sex,
            description,
            photo,
            likes,
            reviews);
            VALUES (:title,
                :group_id,
                :sex,
                :description,
                :photo,
                :likes,
                :reviews)');
        return $req->execute($clothes);
    }

    public function newClothesElement($clothes)
    {
        unset($clothes['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('INSERT INTO ClothesElement
(clothes_id,
            size,
            quantity);
            VALUES (:clothes_id,
                :size,
                :quantity)');
        return $req->execute($clothes);
    }
}

```

```

    }

    public function newReview($review)
    {
        $review ['creationdate'] = date('Y.m.d H:i:s');
        unset($review['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('INSERT INTO Review (score,
            rev,
            user_id,
            creationdate);
            VALUES (:score,
            :rev,
            :user_id,
            :creationdate)');
        $req->execute($review);

        $req = $db->prepare('SELECT AVG(score)
            FROM Review
            WHERE user_id = :user_id');
        $req->bindParam('user_id', $review['user_id']);
        $req->execute();
        $req = $req->fetch(PDO::FETCH_ASSOC);
        $res = $db->prepare('UPDATE User SET score = :score');
        return $res->execute($req[0]);
    }

    public function newDepartmentClothes($clothes)
    {
        $clothes['status'] = 0;
        unset($clothes['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('INSERT INTO DepartmentClothes
(department_id,
            clotheselement_id,
            state,
            quantity);
            VALUES (:department_id,
            :clotheselement_id,
            :state,
            :quantity)');
        $req->execute($clothes);
        $req = $this->updateClothesElement($clothes);
    }

```

```

        return true;
    }

    public function newPrice($price)
    {
        unset($price['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('INSERT INTO Price (clothes_id,
            fromdate,
            todate,
            charge,
            quantity);
            VALUES (:clothes_id,
            :fromdate,
            :todate,
            :charge,
            :quantity)');
        return $req->execute($price);
    }

    public function editPrice($price)
    {
        unset($price['price']);

        $db = DB::getConnection();
        $req = $db->prepare('UPDATE Price SET fromdate =
:fromdate, todate = :todate, charge = :charge, quantity = :quantity
WHERE id = :id');
        $res = $req->execute($price);
        return $res;
    }

    public function check($clothes_id, $fromdate, $todate,
    $department, $quantity)
    {
        return this->checkAvailability($clothes_id, $fromdate,
    $todate, $department, $quantity);
    }

    public function getUser($id)
    {
        $db = DB::getConnection();
        $req = $db->prepare('SELECT User.id,
            User.name,

```

```

        User.login,
        User.score
    FROM User
    WHERE User.id = :user_id');
$req->bindParam('user_id', $id);
$req->execute();
$req = $req->fetchAll(PDO::FETCH_ASSOC);
$res['user'] = $req;
$req = $db->prepare('SELECT Review.id,
    Review.creationdate,
    Review.score,
    Review.rev
    FROM Review
    WHERE Review.user_id = :user_id');
$req->bindParam('user_id', $id);
$req->execute();
$req = $req->fetchAll(PDO::FETCH_ASSOC);
$res['reviews'] = $req;
return $res;
}

public function getOrders()
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT Order.id,
        Order.fromdate,
        Order.todate,
        Order.price,
        Order.paid,
        Order.status,
        Order.creationdate
    FROM Order
    ORDER BY Order.id desc');
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}

public function getDepartmentClothes($id)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT DepartmentClothes.state,
        DepartmentClothes.status,
        DepartmentClothes.quantity,
        ClothesElement.size,

```

```

        Clothes.title,
        Clothes.sex,
        Clothes.description,
        Clothes.photo,
        Clothes.likes,
        Clothes.reviews
    FROM DepartmentClothes
    WHERE      DepartmentClothes.department_id      =
:department_id AND
        ClothesElement.id = DepartmentClothes.id AND
        ClothesElement.clothes_id = Clothes.id');
$req->bindParam('department_id', $id);
$req->execute();
$req = $req->fetchAll(PDO::FETCH_ASSOC);
return $req;
}

public function editClothes($clothes_id, $clothes)
{
    $clothes['id'] = $clothes_id;
    unset($clothes['submit']);

    $db = DB::getConnection();
    $req = $db->prepare('UPDATE Clothes SET title = :title,
group_id = :group_id, sex = :sex, description = :description, photo =
:photo WHERE id = :id');
    $res = $req->execute($clothes);
    return $res;
}

public function cancelOrder($order_id)
{
    $order['id'] = $order_id;
    $order['status'] = 2;
    unset($order['submit']);

    $db = DB::getConnection();
    $req = $db->prepare('UPDATE Order SET status = :status
WHERE id = :id');
    $res = $req->execute($order);
    return $res;
}

public function endOrder($order_id)
{

```

```

        $order['id'] = $order_id;
        $order['status'] = 3;
        unset($order['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('UPDATE Order SET status = :status
WHERE id = :id');
        $res = $req->execute($order);
        return $res;
    }

    public function payOrder($order_id, $paid)
    {
        $order['id'] = $order_id;
        $order['paid'] = $paid;
        unset($order['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('UPDATE Order SET paid = :paid WHERE
id = :id');
        $res = $req->execute($order);
        return $res;
    }

    public function submitOrder($order_id)
    {
        $order['id'] = $order_id;
        $order['status'] = 1;
        unset($order['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('UPDATE Order SET status = :status
WHERE id = :id');
        $res = $req->execute($order);
        return $res;
    }
}

```

Б.2 Текст файла AuthController.php

```
<?php
```

```

class AuthController
{
    public $authModel;
    public $check;
    public function __construct()
    {
        $this->authModel = new AuthModel();
        $this->check = new Check();
    }

    public function actionIndex()
    {
        if(isset($_POST['submit'])) {
            $authData = $_POST;
            if ($this->check->checkEmptyData($authData)) {
                $error = 'Усі поля мають бути заповнені';
            } else {
                $result = $this->authModel->authorize($authData);
                if($result) {
                    header($result);
                } else {
                    $error = 'Невірні дані аутентифікації.';
                }
            }
        }

        require_once(ROOT . '/view/auth.phtml');
        return true;
    }
}

```

Б.3 Текст файла AuthOutController.php

```
<?php
```

```

class AuthOutController
{
    public $authOutModel;

    public function __construct()
    {

```

```

        $this->authOutModel = new AuthOutModel();
    }

    public function actionIndex($login) {
        $this->authOutModel->authOutUser($login);

        header("Location: /auth");
        exit();
    }
}

```

Б.4 Текст файла RegController.php

```

<?php

class RegController
{
    public $regModel;
    public $check;

    public function __construct()
    {
        $this->regModel = new RegModel();
        $this->check = new Check();
    }

    public function actionIndex()
    {
        if(isset($_POST['submit'])) {
            $regData = $_POST;
            if ($this->check->checkEmptyData($regData)) {
                $error = 'Будь ласка, заповніть усі поля';
            } else {
                if(!$this->check->isLoginExist($regData['email'])) {
                    $redirect = $this->regModel->saveUser($regData);
                    header($redirect);
                } else {
                    $error = 'Даний логін вже використано';
                }
            }
        }
    }
}

```

```

    }

    require_once(ROOT . '/view/reg.phtml');
    return true;
}
}

```

Б.5 Текст файла AuthModel.php

```

<?php
class AuthModel
{
    public function authorize ($authData)
    {
        $info = $this->getUserId($authData);
        $staff = $info[0]['staff'];
        $userId = $info[0]['id'];
        if($staff) {
            return $this->detect($staff, $userId);
        } else {
            return false;
        }
    }

    public function getUserId($authData)
    {
        unset($authData['submit']);

        $db = DB::getConnection();
        $req = $db->prepare('SELECT id,
            staff,
            password
            FROM User
            WHERE login = :login');
        $req->execute(['login' => $authData['login']]);
        $info = $req->fetchAll();
        if(isset($info[0]['password']) && $info->correctPassword($authData['password'], $info[0]['password'])) {
            return $info[0];
        }

        return false;
    }
}

```

```

public function correctPassword($password, $passwordHash)
{
    if(password_verify($password, $passwordHash)) {
        return true;
    }

    return false;
}

public function detect ($staff, $userId)
{
    session_start();
    if ($staff) {
        $_SESSION['userId'] = $userId;
        return 'Location: /client';
    } else {
        $_SESSION['userId'] = $userId;
        return 'Location: /renter';
    }

    return 'Location: /reg';
}
}

```

Б.6 Текст файла BaseModel.php

```

<?php

class RenterModel extends RequesterModel
{
    public function newOrder($order, $orderclothes, $department)
    {
        $order['status'] = 0;
        $order['user_id'] = $_SESSION['user'];
        $order['creationdate'] = date('Y.m.d H:i:s');
        $sum = 0;
        $check = true;
        foreach ($orderclothes as $value) {
            $price = $this->getClothesPrice($value['clothes_id'], $order['fromdate'],
            $order['todate']);
            $sum = $sum + $price * $value['quantity'];
        }
    }
}

```

```

        $check = $check and this-
>checkAvailability($value['clothes_id'], $order['fromdate'],
$order['todate'], $department, $value['quantity']);

    }
    $order['price'] = $sum;

    unset($order['submit']);

    if ($check) {
        $db = DB::getConnection();
        $req = $db->prepare('INSERT INTO Order (fromdate,
            todate,
            user_id,
            price,
            paid,
            status,
            creationdate);
        VALUES (:fromdate,
            :todate,
            :user_id,
            :price,
            :paid,
            :status,
            :creationdate)');
        $req->execute($order);
        $order_id = $db->lastInsertId();

        foreach ($orderclothes as $value) {
            $ordercloth['order_id'] = $order_id;
            $ordercloth['departmentclothes_id'] =
getClothesDepartment($department, $value['clothes_id']);
            $ordercloth['quantity'] = $value['quantity'];
            $req = $db->prepare('INSERT INTO OrderClothes
(departmentclothes_id,
            order_id,
            quantity);
        VALUES (:departmentclothes_id,
            :order_id,
            :quantity)');
            $req->execute($ordercloth);
        }
        return true;
    }
    else {

```

```

        return false;
    }

}

public function getCategoryClothes($category)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT id,
        title,
        sex,
        description,
        photo,
        likes,
        reviews
    FROM Clothes
    WHERE category_id = :category
    ORDER BY Clothes.likes desc');
    $req->bindParam('group_id', $category);
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}

public function getPrices($id)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT id,
        charge,
        quantity
    FROM Price
    WHERE clothes_id = :id AND
        fromdate <= :today AND
        todate >= :today');
    $today = date('Y.m.d H:i:s');
    $req->bindParam('clothes_id', $id);
    $req->bindParam('today', $today);
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}

public function getClothes($id)
{
    $db = DB::getConnection();

```

```

        $req = $db->prepare('SELECT ClothesElement.id,
                                ClothesElement.size,
                                Clothes.id,
                                Clothes.title,
                                Clothes.sex,
                                Clothes.description,
                                Clothes.photo,
                                Clothes.likes,
                                Clothes.reviews
                                FROM ClothesElement
                                INNER JOIN Clothes on ClothesElement.clothes_id =
Clothes.id
                                WHERE ClothesElement.clothes_id = :clothes_id
                                ORDER BY ClothesElement.size');
        $req->bindParam('clothes_id', $id);
        $req->execute();
        $req = $req->fetchAll(PDO::FETCH_ASSOC);
        return $req;
    }

    public function getGroups()
    {
        $db = DB::getConnection();
        $req = $db->prepare('SELECT id,
                                title
                                FROM Group
                                ORDER BY Group.title');
        $req->execute();
        $req = $req->fetchAll(PDO::FETCH_ASSOC);
        return $req;
    }

    public function getDepartments()
    {
        $db = DB::getConnection();
        $req = $db->prepare('SELECT Department.id,
                                Department.title,
                                City.title
                                FROM Department
                                INNER JOIN City on City.id = Department.city_id
                                ORDER BY Department.city_id');
        $req->execute();
        $req = $req->fetchAll(PDO::FETCH_ASSOC);
        return $req;
    }

```

```

public function getOrder($id)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT Clothes.id,
        Clothes.title,
        Clothes.photo,
        Clothes.likes,
        Clothes.reviews,
        OrderClothes.quantity
    FROM OrderClothes
    WHERE Order.id = :order_id AND OrderClothes.order_id
= Order.id AND
        OrderClothes.departmentclothes_id =
DepartmentClothes.clotheselement_id AND
        ClothesElement.clothes_id = Clothes.id
    ORDER BY Order.creationdate desc');
    $req->bindParam('order_id', $id);
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}

public function searchClothes($department_id, $sex, $group,
$category, $size, $title)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT DepartmentClothes.state,
        DepartmentClothes.status,
        DepartmentClothes.quantity,
        ClothesElement.size,
        Clothes.title,
        Clothes.sex,
        Clothes.description,
        Clothes.photo,
        Clothes.likes,
        Clothes.reviews
    FROM DepartmentClothes
    WHERE
        DepartmentClothes.department_id =
:department_id AND
        Clothes.sex = :sex AND
        Clothes.group_id = :group AND
        Clothes.category_id = :category AND
        Clothes.size = :size AND
        Clothes.title LIKE ? AND
        ClothesElement.id = DepartmentClothes.id AND

```

```

        ClothesElement.clothes_id = Clothes.id');
$req->bindParam('sex', $sex);
$req->bindParam('department_id', $department_id);
$req->bindParam('group', $group);
$req->bindParam('category', $category);
$req->bindParam('size', $size);
$req->bindParam('s', $title);
$req->execute();
$req = $req->fetchAll(PDO::FETCH_ASSOC);
return $req;
}

public function getSexClothes($sex)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT id,
        Clothes.title,
        Group.title,
        Clothes.description,
        Clothes.photo,
        Clothes.likes,
        Clothes.reviews
    FROM Clothes
    INNER JOIN Group on Group.id = Clothes.group_id
    WHERE Clothes.sex = :sex
    ORDER BY Clothes.likes desc');
    $req->bindParam('sex', $sex);
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}

public function getSizeClothes($size)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT id,
        Clothes.title,
        Group.title,
        Clothes.description,
        Clothes.photo,
        Clothes.likes,
        Clothes.reviews
    FROM Clothes
    INNER JOIN Group on Group.id = Clothes.group_id

```

```

        INNER JOIN ClothesElement
ClothesElement.clothes_id = Clothes.id
        WHERE ClothesElement.size = :size
        ORDER BY Clothes.likes desc');
$req->bindParam('size', $size);
$req->execute();
$req = $req->fetchAll(PDO::FETCH_ASSOC);
return $req;
}

public function getGroupClothes($group)
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT id,
        title,
        sex,
        description,
        photo,
        likes,
        reviews,
    FROM Clothes
    WHERE group_id = :group_id
    ORDER BY Clothes.likes desc');
    $req->bindParam('group_id', $group);
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}

public function getCategories()
{
    $db = DB::getConnection();
    $req = $db->prepare('SELECT id,
        title
    FROM Category
    ORDER BY Category.title');
    $req->execute();
    $req = $req->fetchAll(PDO::FETCH_ASSOC);
    return $req;
}
}

```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Слайд № 1



Рисунок В.2 – Слайд № 2

Порівняння програмних аналогів				
	zakupka.com	olx.ua	gipr.ua	Розробка
Пошук одягу	+	+	+	+
Пошук за розміром	-	+	+	+
Категоризація одягу	+	-	-	+
Відображення одягу за категоріями	-	-	-	+
Відображення одягу за статтю	+	+	+	+
Замовлення прокату	+	+	+	+
Замовлення прокату на задані дати	-	-	+	+
Відгуки про одяг	+	-	-	+
Перевірка наявності одягу на задані дати	-	-	+	+

3

Рисунок В.3 – Слайд № 3

Вибір мови програмування		
	Python	PHP
Спеціалізація мови	Мова програмування загального призначення	Спеціалізована для веброзробки мова
Швидкість виконання	Відносно висока	Майже в 3 більша порівняно з PHP, починаючи з версії 7.x
Безкоштовне і відкрите рішення	+	+
Інтеграція з базами даних	+	++

4

Рисунок В.4 – Слайд № 4

Діаграма use case прокатника

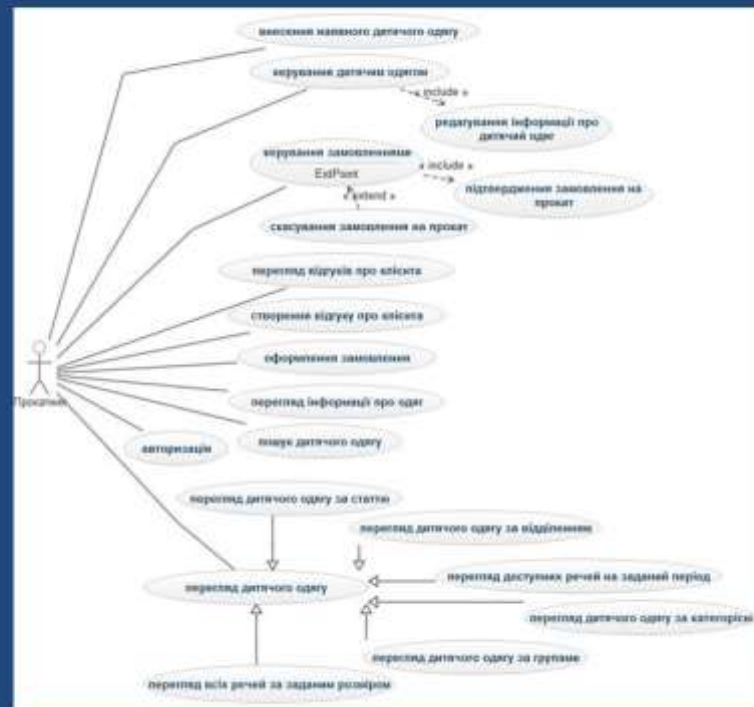


Рисунок В.5 – Слайд № 5

Діаграма use case клієнта

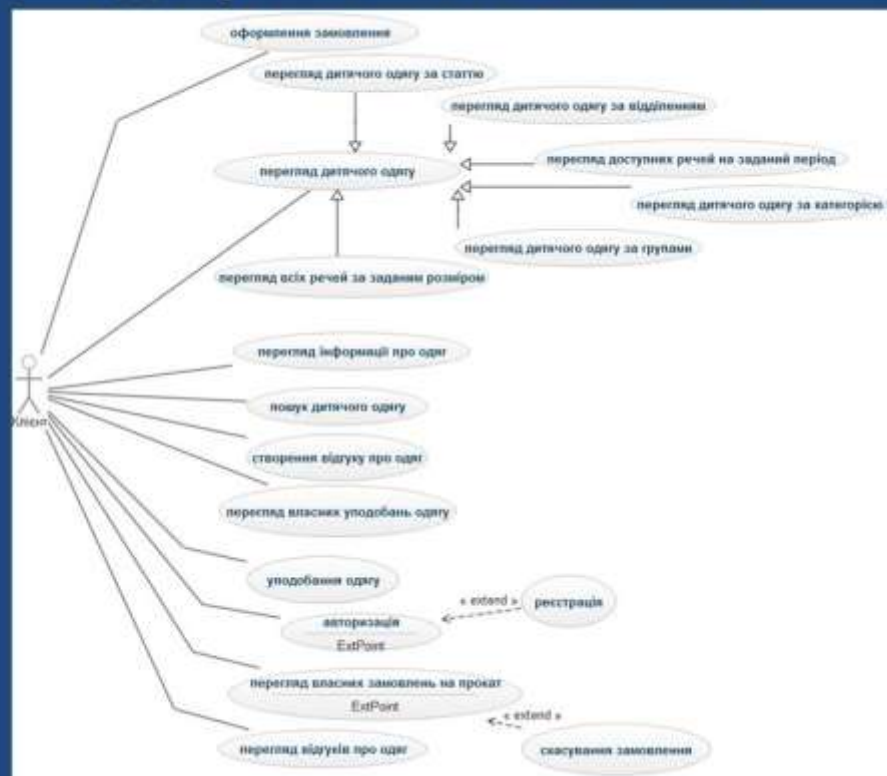


Рисунок В.6 – Слайд № 6

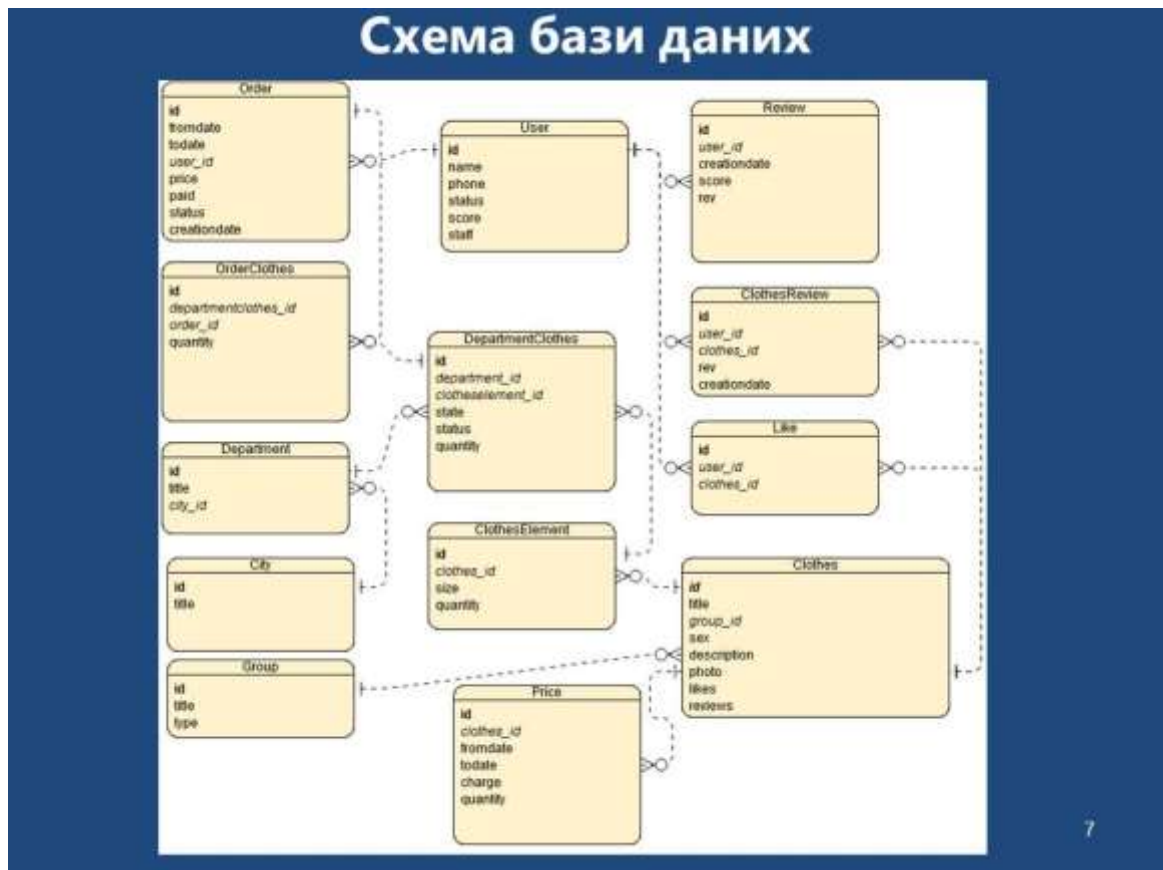


Рисунок В.7 – Слайд № 7



Рисунок В.8 – Слайд № 8



Рисунок В.9 – Слайд № 9

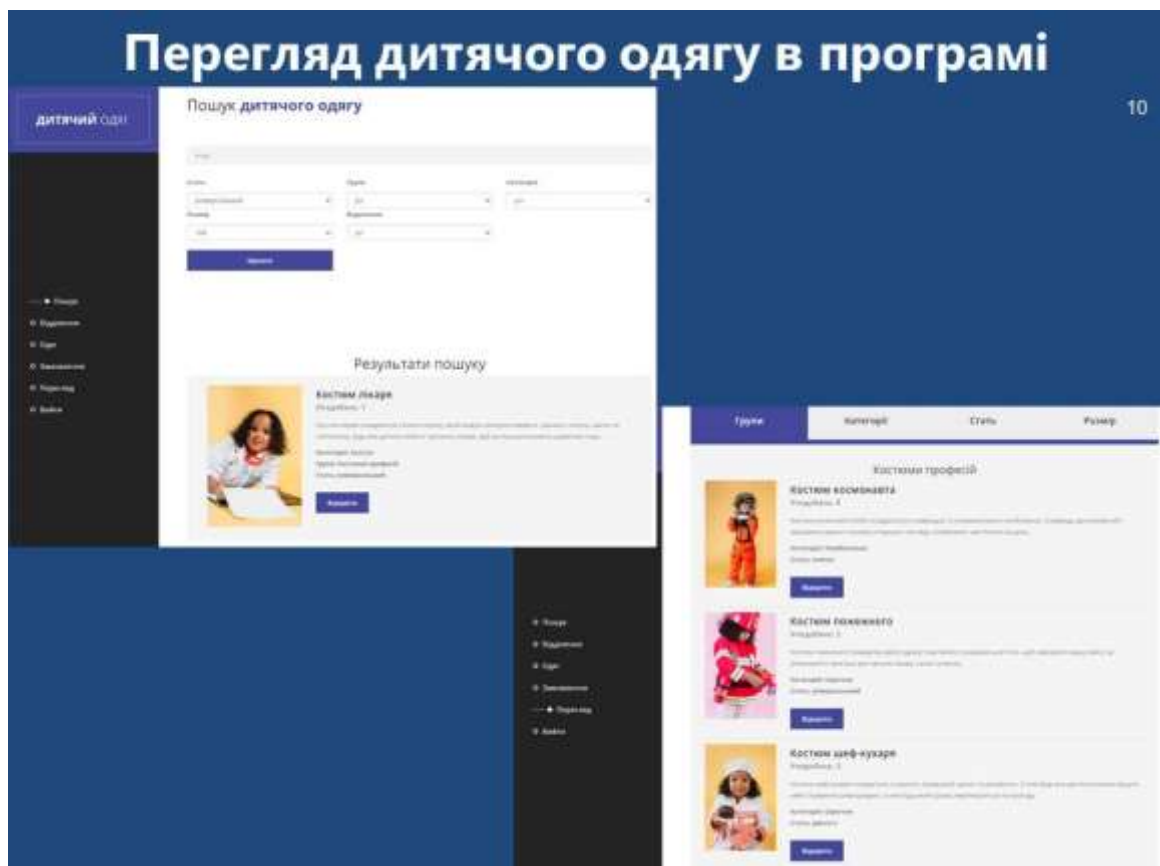


Рисунок В.10 – Слайд № 10

Керування замовленнями в програмі

дитячий одяг

- После
- Контроль
- Список
- Заказчики
- Платежи
- Вывод

КОСТЮМ БОКСЕРА
Модель № 7 (детский)

Описание: Костюм боксера, состоящий из куртки и штанов, выполненный из высококачественных материалов. Подходит для тренировок и выступлений.

Детали: Спинка/Плечи: 100 см, Рукава: 100 см, Штаны: 100 см, Пояс: 100 см, Длина: 100 см.

Варианты:

Избранное:

Категория:

Форматирование:

Детали:

11

Замовлення на прокат одягу

№	№	№	Период	Вартість	Статус
10	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
20	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
30	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
40	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
50	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
60	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
70	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
80	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
90	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано
100	1000000	1000000	10.05.2021 - 10.05.2021	100 грн	Виконано

После

Контроль

Список

Заказчики

Платежи

Вывод

Рисунок В.11 – Слайд № 11

Висновки

У дипломній кваліфікаційній роботі виконано розроблення програмного забезпечення для підтримки роботи з клієнтами прокату дитячого одягу.

Виконано аналіз предметної області, що включив аналіз програмних засобів, які можуть вирішувати проблему, зокрема вебсайтів olx.ua, zakupka.com, dip.ua, barvi.net, а також виконано порівняння аналогів і виділено основні функції, які в якості переваг над альтернативи має виконувати розробка.

Визначено функціональні вимоги до програми, на основі чого побудовано діаграми use case для роботи клієнта та прокатника з програмою. Обгрунтовано вибір мови програмування PHP і системи керування БД MySQL для реалізації програмного забезпечення. Виконано проєктування БД, в результаті описано структуру створеної БД та представлено схему БД.

Реалізовано вебзастосунок для підтримки прокату дитячого одягу. Описано алгоритм функціонування програмного забезпечення з приведенням відповідної схеми функціонування, описано структуру створеного програмного забезпечення. Представлено основні розроблені в процесі реалізації класи.

Завдання дипломної кваліфікаційної роботи в результаті повністю виконано.

Рисунок В.12 – Слайд № 12