

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ
ЗАСОБАМИ FLUTTER ТА MONGODB»**

Виконала: студентка 2 курсу, групи 8.1210

спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення
(назва освітньої програми)

Г. Д. Крикунова

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії, доцент,
к.ф.-м.н., Кудін О. В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри фундаментальної та прикладної
математики, доцент, к.ф.-м.н., Клименко М. І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Математичний

Кафедра програмної інженерії

Рівень вищої освіти Магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент
Лісняк А.О.

(підпис)

« 09 » 06 2021 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Крикуновій Ганні Дмитрівні

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка інформаційної системи засобами Flutter та
MongoDB

керівник роботи (проекту) Кудін Олексій Володимирович, к.ф.-м.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 09 » червня 2021 року № 851-с

2. Строк подання студентом роботи 24.11.2021

3. Вихідні дані до роботи 1. Постановка задачі. 2. Перелік літератури

3. Види задач, які підлягають розгляду

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі, аналіз предметної області.

2. Специфікація вимог до інформаційної системи.

3. Моделювання та проектування інформаційної системи.

4. Реалізація та тестування інформаційної системи.

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

Презентація, функціональна та інформаційна моделі системи, керівництво користувача.

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| 1 | Кудін О.В., доцент | 08.06.2021 | 08.06.2021 |
| 2 | Кудін О.В., доцент | 15.09.2021 | 15.09.2021 |
| 3 | Кудін О.В., доцент | 16.10.2021 | 16.10.2021 |
| 4 | Кудін О.В., доцент | 15.11.2021 | 15.11.2021 |

7. Дата видачі завдання 08.06.2021

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-----|--|-------------------------------|----------|
| 1. | Розробка плану виконання кваліфікаційної роботи магістра. | 08.06.2021 | |
| 2. | Збір вихідних даних та аналіз предметної області. | 18.06.2021 | |
| 3. | Обробка методичних та теоретичних джерел. | 26.09.2021 | |
| 4. | Специфікація вимог до системи і об'єктів автоматизації. Робота над першим розділом. | 02.10.2021 | |
| 5. | Вибір архітектури побудови ІС. Розробка функціональної та інформаційної моделей системи. Робота над другим розділом. | 24.10.2021 | |
| 6. | Моделювання та проектування системи. Робота над третім розділом. | 26.10.2021 | |
| 7. | Реалізація та тестування системи. Робота над четвертим розділом. | 02.11.2021 | |
| 8. | Розробка керівництва користувача. Робота над додатками. | 05.11.2021 | |
| 9. | Оформлення та нормоконтроль кваліфікаційної роботи магістра. | 15.11.2021 | |
| 10. | Захист кваліфікаційної роботи магістра. | 17.12.2021 | |

Студентка

_____ (підпис)

Г.Д. Крикунова

_____ (ініціали та прізвище)

Керівник роботи

_____ (підпис)

О.В. Кудін

_____ (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер

_____ (підпис)

С.П. Швидка

_____ (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка інформаційної системи засобами Flutter та MongoDB»: 55 с., 38 рис., 20 табл., 14 джерел.

БД, ФРЕЙМВОРК, ANDROID, CRM, DART, ER, FLUTTER, IDEF, MONGOD, MVC, UML.

Об'єкт дослідження – діяльність магазину одягу.

Мета роботи: розробка та реалізація мобільного додатку для магазину одягу.

Метод дослідження – методи збору та аналізу вимог до програмного забезпечення, методи моделювання, проектування, конструювання та тестування програмних забезпечення.

На сьогоднішній день на ринку послуг існує досить велика кількість магазинів одягу. Багато власники бізнесів усвідомили, що присутність додатка на мобільних платформах є обов'язковою складовою результативною маркетингової стратегії розвитку бізнесу, тому розробка додатків для мобільних пристроїв актуальна і розвивається дуже активно.

При розробці системи було проведено аналіз предметної області, обрана архітектура і платформа реалізації, розроблена функціональна модель системи у нотаціях IDEF0, IDEF1 та IDEF3, спроектована архітектура система за допомогою UML у вигляді наборів діаграм використання, послідовностей, класів, діяльності, компонентів та розгортання, реалізована інформаційна система з використанням мови програмування Dart, здійснене тестування системи та оформлена супровідна документація.

SUMMARY

Master`s qualifying paper «Development of Information System Using Flutter and MongoDB»: 55 pages, 38 figures, 20 tables, 14 references.

ANDROID, BPMN, CRM, DART, ER, FLUTTER, FRAMEWORK, IDEF, MONGODB, MVC, UML.

The object of the study is activity of clothing store.

The aim of the study is development and implementation of a mobile application for a clothing store.

The methods of research are methods of collection and analysis of software requirements, modeling, design, construction and testing of software.

Today there are quite a number of clothing stores in the services market. Many business owners have realized that the presence of the application on mobile platforms is a necessary component of a successful marketing strategy for business development, so the development of applications for mobile devices is relevant and is developing very actively.

In developing an analysis of the subject area chosen architecture and platform implementation, developed a functional model system of notation IDEF0, IDEF1 and IDEF3, designed the system architecture using UML diagrams as sets use sequences, classes, activities, components, and deployment, information system implemented using programming language Dart, performed system testing and issued accompanying documents.

ЗМІСТ

| | |
|---|----|
| Завдання на кваліфікаційну роботу..... | 2 |
| Реферат | 4 |
| Summary | 5 |
| Вступ..... | 7 |
| 1 Технічне завдання | 9 |
| 1.1 Огляд останніх досліджень і публікацій | 9 |
| 1.2 Огляд готових рішень..... | 10 |
| 1.3 Вимоги до системи | 14 |
| 1.4 Огляд технологій | 15 |
| 2 Моделювання інформаційної системи..... | 17 |
| 2.1 Дослідження предметної області | 17 |
| 2.2 Функціональне проектування..... | 18 |
| 2.3 Концептуальне проектування інформаційної моделі | 22 |
| 2.4 Логічне проектування інформаційної моделі | 23 |
| 2.5 Діаграма варіантів використання..... | 27 |
| 2.6 Діаграма взаємодії та послідовності..... | 29 |
| 2.7 Діаграма потоків даних | 31 |
| 2.8 Діаграма класів..... | 32 |
| 2.9 Діаграма розгортання та компонентів | 33 |
| 2.10 Діаграма кооперації | 35 |
| 3 Розробка мобільного додатка..... | 36 |
| 3.1 Вибір бази даних..... | 36 |
| 3.2 Розробка інтерфейсу..... | 38 |
| 3.3 Тестування додатку | 48 |
| Висновки | 53 |
| Перелік посилань..... | 54 |

ВСТУП

У сучасному світі складно уявити людину, яка б не користувалася в повсякденному житті смартфоном. Кожен пристрій наповнений безліччю мобільних додатків. Записні книжки, фітнес-трекери, месенджери, онлайн-кінотеатри, графічні редактори. Цей список можна продовжувати до нескінченності. Серед усього різноманіття можна виділити цілий пласт мобільних додатків – інтернет-магазини.

З приходом в життя людини смартфонів - багато підприємців прагнуть зайняти свою нішу на ринку онлайн-покупок. На це у них є ряд досить вагомих причин:

1. Нині є безліч необхідних ресурсів для аналізу аудиторії, конкурентів, попиту на ринку. Таким чином, спрогнозувати, створити та розвинути успішний бізнес стає значно простіше.

2. Постійне зростання кваліфікованих кадрів і сервісів, які допоможуть підприємцю просувати бізнес, а також шукати нових клієнтів.

3. Можливість працювати над інтернет-магазином віддалено, що дозволяє шукати співробітників не тільки в своєму регіоні, а й по всьому світу.

4. Широкий спектр можливостей з роботою з доставкою товарів, куплених онлайн.

5. Відповідно до досліджень Data Insight – дослідному агентству, що спеціалізується на ринку електронної комерції - середнє зростання інтернет-торгівлі з 2011 по 2019 рік склав 28% [1].

Власники бізнесу, які знають про ці переваги, неодмінно замислюються про необхідність створення мобільного додатку для свого проекту і отримання додаткового прибутку з покупок їх товарів онлайн.

За допомогою мобільного додатку компанії розширюють свій вплив на ринку товарів і послуг, збільшують базу клієнтів і підвищують популярність свого бренду.

Метою даної роботи є створення інформаційної системи для магазину одягу та автоматизація основних бізнес-процесів. Виходячи з зазначеної мети дипломної роботи, її основними завданнями є:

- збір інформації для написання специфікацій вимог до системи (технічного завдання);
- розкрити теоретичні основи технології Flutter;
- проаналізувати діяльність магазину одягу;
- побудувати функціональну структуру для автоматизації основних бізнес процесів та потоків даних;
- побудувати моделі інформаційної системи;
- спроектувати інформаційну систему;
- реалізувати та протестувати інформаційну систему;
- оформити документацію до проекту.

Об'єктом дослідження даної роботи є інформаційний веб-простір магазину одягу. Предметом дослідження є інформаційне та програмне забезпечення бізнес-процесів магазину одягу.

Методом дослідження даної роботи є методи збору та аналізу вимог до програмного забезпечення, методи моделювання, проектування, конструювання та тестування програмного забезпечення.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновку, переліку посилань.

Перший розділ роботи присвячений формулюванню технічного завдання, специфікацій вимог до проекту та розкриттю теоретичних основ Flutter технології.

У другому розділі проектується функціональні та інформаційні складові системи за допомогою засобів IDEF0, ER-діаграми, UML-діаграми.

Третій розділ присвячений реалізації, тестуванню програмного продукту.

Також представлена документація до використання системи.

1 ТЕХНІЧНЕ ЗАВДАННЯ

1.1 Огляд останніх досліджень і публікацій

У сучасному суспільстві найбільше цінується час, люди дуже багато витрачають часу на покупки. Для того щоб економити час почали створюватися інтернет - магазини.

Інтернет-магазин - це прикладна система, яка використовує технології системи електронної торгівлі. Інтернет - магазин реалізує наступні функції: представлення товарів (послуг) покупцю, обробку замовлень, продаж і доставку товарів.

Переваги такого виду покупки:

- можливість вибору та придбання товару або послуги, не виходячи з будинку (економія часу);
- негайна доставка та супровід програм при покупці їх через Мережу;
- отримання додаткової інформації про необхідні товари;
- просто і зручно оформити замовлення.

У порівнянні зі звичайною формою транзакцій, інтернет-магазини:

- мають можливість встановлювати більш низькі ціни на весь асортимент завдяки меншим витратам (в першу чергу за рахунок оренди торгових площ, дорогого торгового обладнання);
- можуть організувати додаткові знижки та дисконтні програми завдяки врахуванню і ведення бази даних покупців;
- дають клієнтам консультації щодо вибору та застосування товарів в зручний час;
- пошук і доставка товарів під замовлення;
- кожен клієнт може бути персоналізований відповідно до історії замовника відвідування магазину та попередніх покупок.

Інформаційна система для продажу одягу ще не досить заповнена і тому конкурентоспроможність заходи щодо створення онлайн площадки з продажу саме цієї продукції досить висока. Крім того, дана продукція цікава як для жіночої, так і для чоловічої половини людства, що тільки підкреслює актуальність інтернет магазину.

1.2 Огляд готових рішень

Для аналізу було обрано декілька мобільних додатків зі схожою тематикою і призначенням.

Одним із таких додатків являється додаток магазину жіночого одягу Staff Clothes, представлений на рис. 1.1.

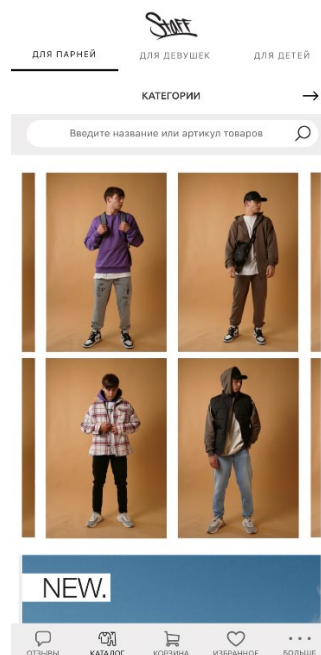


Рисунок 1.1 – Додаток магазину одягу «Staff Clothes»

Оскільки специфіка магазину – одяг, необхідно звернути увагу на такий функціонал, як «опції товару». Покупець повинен мати можливість обрати колір та розмір одягу.

Для підвищення продажів, важливими є маркетингові функції. «Зум» зображень – збільшення картинки товару важливе для прийняття рішення про покупку, також можливість побачити відео.

Додаток Zara має схожу тематику і призначення, представлена на рис.1.2.



Рисунок 1.2 – Додаток магазину одягу «Zara»

Даний сайт також має недоліки, а саме не дуже зручний інтерфейс, відсутність зворотнього зв'язку, відсутність зміни кольору одягу, обмежена кількість доступних розмірів. Також кожен товар має невеликий опис товару, що для клієнта є важливим при покупці.

Для аналізу також був обраний сайт Mango основна тематика якого продаж одягу, представлений на рис. 1.3.

Перевагою даного сайту є можливість збільшення зображення товару, а також можливість зворотного зв'язку. Даний сайт також має недоліки, а саме відсутність можливості підібрати розмір та детального опису продукції.

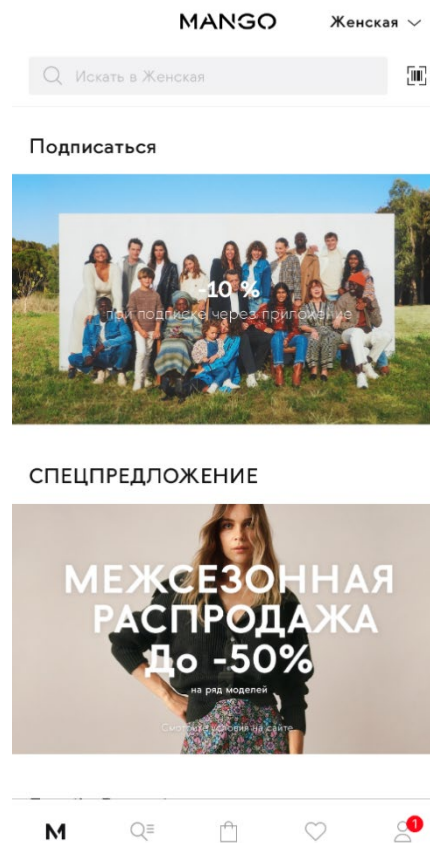


Рисунок 1.3 – Додаток магазину одягу «Mango»

А також відсутність достатньої кількості фото та відео не дає можливості в повному обсязі оцінити даний товар.

На сьогоднішній день створена велика кількість додатків подібного плану, але вони мають певні недоліки.

Проаналізувавши більшість вже створених додатків можна виділити такі недоліки:

- зручність використання. Потенційний покупець не може самостійно розібратися, як оформити замовлення, куди клікнути, щоб дізнатися умови доставки і куди поставити галочку, щоб обраний товар привезли додому;
- застарілий дизайн;
- відсутність детального опису товару та відео;
- відсутність можливості правильно підібрати розмір одягу, що призводить до повернення товару та негативній оцінці магазину.

Розглянемо детальніше інформацію про додатки та проведемо загальний аналіз (табл. 1.1).

Таблиця 1.1 – Аналіз розглянутих додатків для продажу одягу

| Критерії | Мобільний додаток Staff Clothes | Мобільний додаток Zara | Мобільний додаток Mango |
|-----------------------------------|--|-------------------------------|--------------------------------|
| Відсутність непотрібної реклами | + | + | + |
| Неактуальна інформація | - | + | + |
| Можливість зворотного зв'язку | + | - | + |
| Зручність використання | + | - | + |
| Некваплива обробка замовлень | + | + | + |
| Застарілий дизайн | - | + | - |
| Можливість збільшувати фотографії | + | - | + |
| Видима цінова політика | + | + | + |

Отже, проаналізувавши сайти зі схожою тематикою і призначенням було вирішено розробити додаток який не матиме недоліків виявлених в додатках, представлених в таблиці 1.1. Дана розробка матиме всі переваги даних додатків, а саме: актуальну інформацію, можливість зворотного зв'язку, відсутність непотрібної реклами, зручність використання, швидка обробка замовлень, новий дизайн, наявність фото та відео, які можна масштабувати, видима цінова політика, а також матиме унікальний дизайн та зручний інтерфейс.

1.3 Вимоги до системи

Мобільний додаток призначений для того, щоб користувач міг з легкістю обрати для себе одяг і замовити не виходячи з дому. Користувачу дається вибір способу доставки та оплати. Користувач має можливість оплатити товар карткою або готівкою при отриманні замовлення. Інтернет-магазин повинен бути реалізований як Android додаток, доступний в Google Play. Додаток повинен складатись з взаємозалежних частин, а функції кожної частини повинні бути чітко розділені.

Даний додаток повинен мати наступні можливості:

- оформлення замовлення;
- методи зворотного зв'язку;
- авторизація;
- адміністративна панель;
- пошук товару: по всім категоріям, по конкретній категорії;
- фільтрування товарів: за популярністю, ціна за зростанням, за зменшенням;
- кошик;
- трьохрівневий захист для адміністративної панелі.

Додаток повинен складатись з двох частин, а саме адміністративної і користувацької.

Адміністративна частина додатку надає можливість користувачу додавати, редагувати та видалити вміст розділів додатку.

Користувацький інтерфейс додатку повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації, швидко переходити до розділів.

Наявність навігації на всіх сторінках додатку. Правильна структура інформації дозволяє користувачам продовжувати дослідження додатку, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

1.4 Огляд технологій

Flutter – це розроблений Google фреймворк з відкритим програмним кодом, який дозволяє просто і швидко створювати мобільні додатки для iOS і Android. При цьому в роботі Flutter не використовує нативні компоненти зовсім. Замість цього всі UI-елементи у фреймворку створюються за допомогою власного графічного рушія [2].

Особливістю роботи з Flutter є те, що додатки під різні платформи можуть мати один і той самий код. Flutter використовує тільки одну мову програмування Dart. При побудові програми Flutter транслює код на Dart в нативний код, який можна запускати на Android або iOS або іншій платформі.

Переваги Flutter:

- однаковий інтерфейс, призначений для користувача і бізнес-логіка на всіх платформах;
- скорочення часу розробки коду;
- збільшення швидкості виведення на ринок;
- величезний потенціал налаштування призначеного для користувача інтерфейсу;
- проста реалізація логіки для конкретної платформи;
- потенційна можливість вийти за рамки мобільних пристроїв.

Отже, Flutter - це найшвидший спосіб створити добре працюючий кросплатформовий мобільний додаток.

Dart – мова програмування, яку розробляє компанія Google, позиціонуючи як мову структурованого програмування для Веб. Розробники вважали, що в довгостроковій перспективі Dart може стати прогресивною заміною JavaScript, котрий потерпає від наявних в даний час проблем з розширюваністю, продуктивністю і підтримкою розробки складних застосунків.

Dart може робити все те, що й звичайний JavaScript: веб-додатки та сервери (а в майбутньому й flutter).

Особливості мови Dart:

- звичний і простий для вивчення синтаксис, природний для програмістів на JavaScript, C і Java;
- забезпечення швидкого запуску і високої продуктивності для всіх сучасних веб-браузерів і різних типів оточень, від портативних пристроїв до потужних серверів;
- необов'язкове вказування типів, використовувати чи ні статичні типи вирішує розробник. Вказування типів дозволяє спростити виявлення помилок, робить код яснішим і читаним, спрощує його доопрацювання та аналіз сторонніми розробниками.
- підтримка використання бібліотек, що спрощують підтримку великих веб-проектів. Сторонні реалізації функцій можуть підключатися у вигляді поділюваних бібліотек. Застосунки можна розбити на частини і доручити розробку кожної з частин окремій команді програмістів;
- набір готових інструментів для підтримки розробки мовою Dart, включаючи реалізацію засобів динамічної розробки та виправлення коду на льоту ("edit-and-continue");
- можливість створювати однорідні системи, що охоплюють як клієнтську, так і серверну частину. Використання однієї мови та інструментарію для клієнтських і серверних компонентів спрощує процес кодування і позбавляє від постійної зміни контексту [3].

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Дослідження предметної області

Інтернет-магазин – сайт, який торгує товарами в інтернеті. Дозволяє користувачам сформувати замовлення на покупку, вибрати спосіб оплати і доставки замовлення в мережі Інтернет.

Обравши необхідні товари або послуги, користувач зазвичай має можливість на сайті вибрати спосіб оплати та доставки. Сукупність відібраних товарів, спосіб оплати і доставки є закінчене замовлення, яке оформлюється на сайті шляхом повідомлення мінімально необхідної інформації про покупця. Інформація про покупця може зберігатися в базі даних магазину якщо бізнес-модель магазину розрахована на повторні покупки, або ж відправлятися разово. В інтернет-магазинах, розрахованих на повторні покупки, також ведеться відстеження повернень відвідувача і історія покупок. Часто при оформленні замовлення передбачається можливість повідомити деякі додаткові побажання від покупця продавцю.

Існують два різновиди інтернет-магазинів, в залежності від виду торгівлі:

- Магазины, які продають товар зі свого складу. Такий магазин - варіант додаткового збуту товару, зазвичай дають більш низьку ціну, ніж навіть в своєму реальному магазині;
- Магазины, які продають товар інших магазинів / людей. Це може бути торгівля в середині країни, або міжнародна торгівля. У цьому випадку інтернет-магазин заробляє на комісії, яку платять продавці за виставляння товару. В даній ситуації інтернет-магазин виступає гарантом угоди між продавцем і покупцем. Такі магазини використовують систему "репутація" продавця. Крім того, покупець може поскаржитися адміністрації сайту на продавця і

отримати необхідну допомогу по поверненню грошей, в разі обману [4].

Цей мобільний додаток має виконувати наступні функції:

- реєстрація покупця;
- авторизація покупця та адміністратора;
- перегляд фото товару;
- можливість додавання товару у кошик у кількості наявній в магазині;
- можливість вибору розміру та кольору;
- наявність категорій одягу;
- зберігання замовлень в базі даних для швидкого і ефективного пошуку (історія покупок для покупців);
- перегляд історії та статусу замовлення;
- кошик;
- зручна навігація по додатку;
- адміністративна панель;
- фільтрування товарів.

2.2 Функціональне проектування

Опис системи за допомогою IDEF0 називається функціональною моделлю. Функціональна модель призначена для опису існуючих бізнес-процесів, в якому використовуються як природний, так і графічні мови. Для передачі інформації про конкретну систему джерелом графічного мови є сама методологія IDEF0.

Методологія IDEF0 наказує побудову ієрархічної системи діаграм - одиничних описів фрагментів системи. Спочатку проводиться опис системи в цілому і її взаємодії з навколишнім світом (контекстна діаграма), після чого проводиться функціональна декомпозиція – система розбивається на

підсистеми і кожна підсистема описується окремо (діаграми декомпозиції). Потім кожна підсистема розбивається на більш дрібні і так далі до досягнення потрібного ступеня деталізації.

IDEF0 – нотація графічного моделювання, яка використовується для створення функціональної моделі, що відображає структуру і функції системи, а також потоки інформації і матеріальних об'єктів, що зв'язують ці функції.

Контекстна діаграма – діаграма, на якій об'єкт моделювання представлений єдиним блоком з граничними стрілками. Ця діаграма називається А-0 (А мінус нуль). Стрілки на цій діаграмі відображають зв'язку об'єкта моделювання з навколишнім середовищем. Діаграма А-0 встановлює область моделювання та її кордон.

Після декомпозиції контекстної діаграми (тобто, отримання діаграми А 0) проводиться декомпозиція кожного блоку діаграми А 0 на більш дрібні фрагменти і так далі, до досягнення потрібного рівня деталізації опису. Після кожного сеансу декомпозиції проводяться сеанси експертизи - експерти предметної області (зазвичай це беруть інтерв'ю аналітиками співробітники підприємств) вказують на відповідність реальних бізнес-процесів створеним діаграм. Знайдені невідповідності виправляються, і тільки після проходження експертизи без зауважень можна приступати до наступного сеансу декомпозиції. Так досягається відповідність моделі реальних бізнес-процесів на будь-якому кожному рівні моделі. Синтаксис опису системи в цілому і кожного її фрагмента однаковий у всій моделі [5].

Функціональна модель є подальшою деталізацією (декомпозицією) контекстної діаграми. Спочатку функція системи в цілому (ціль, призначення, головне завдання) розбивається на декілька окремих функцій (завдань, робіт, цілей). Кожен функціональний блок (робота), яким виступає окрема функція (робота, ціль чи завдання), зображується прямокутником. Сторони прямокутників робіт (функціональних блоків) мають таке ж призначення, що й розглянуті вище сторони контекстної діаграми.

Нижче подано діаграму діяльності підприємства, для якого проектується ІС (рис. 2.1).

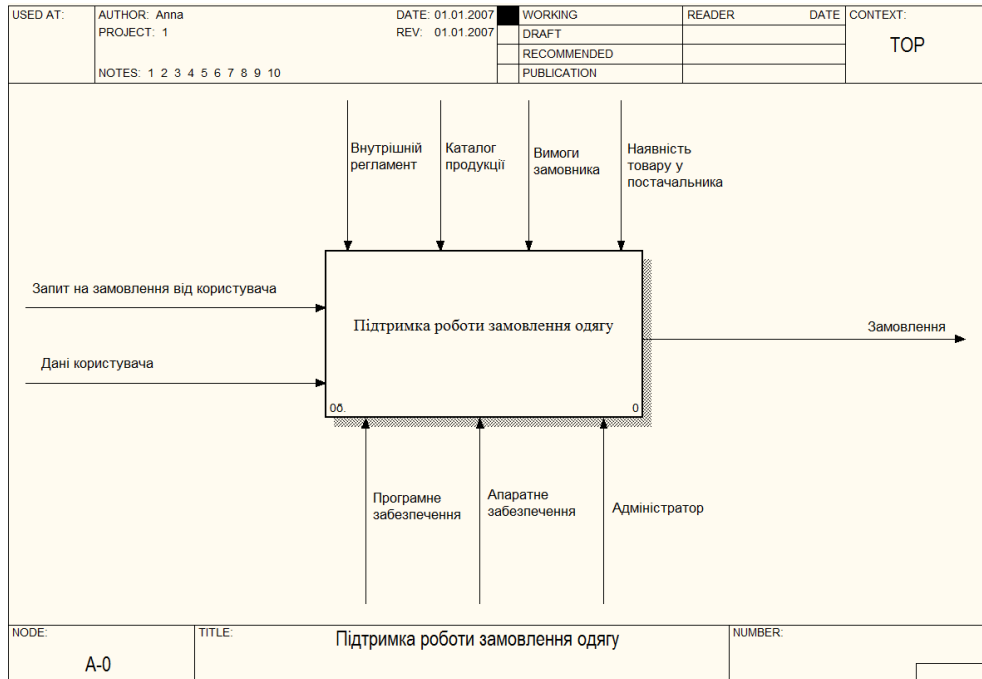


Рисунок 2.1 – Контекстна діаграма функціонування магазину одягу

Діяльність підприємства розкривається діаграмою декомпозиції контекстної діаграми (рис. 2.2).

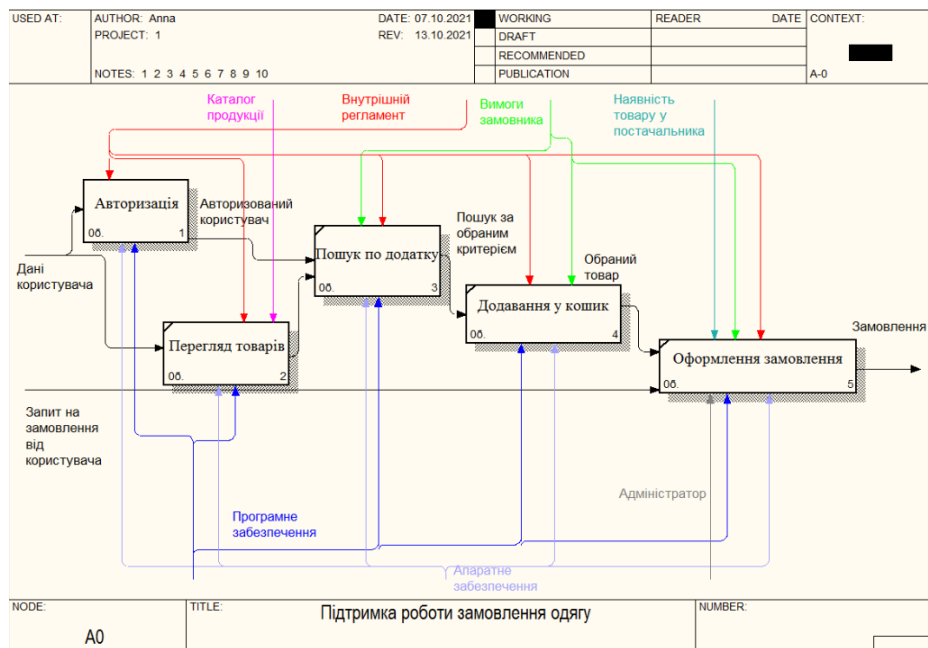


Рисунок 2.2 – Діаграма декомпозиції процесу "Діяльність магазину одягу"

Основні бізнес-процеси конкретизуються діаграмою декомпозиції бізнес-процесів (рис. 2.3 – 2.4).

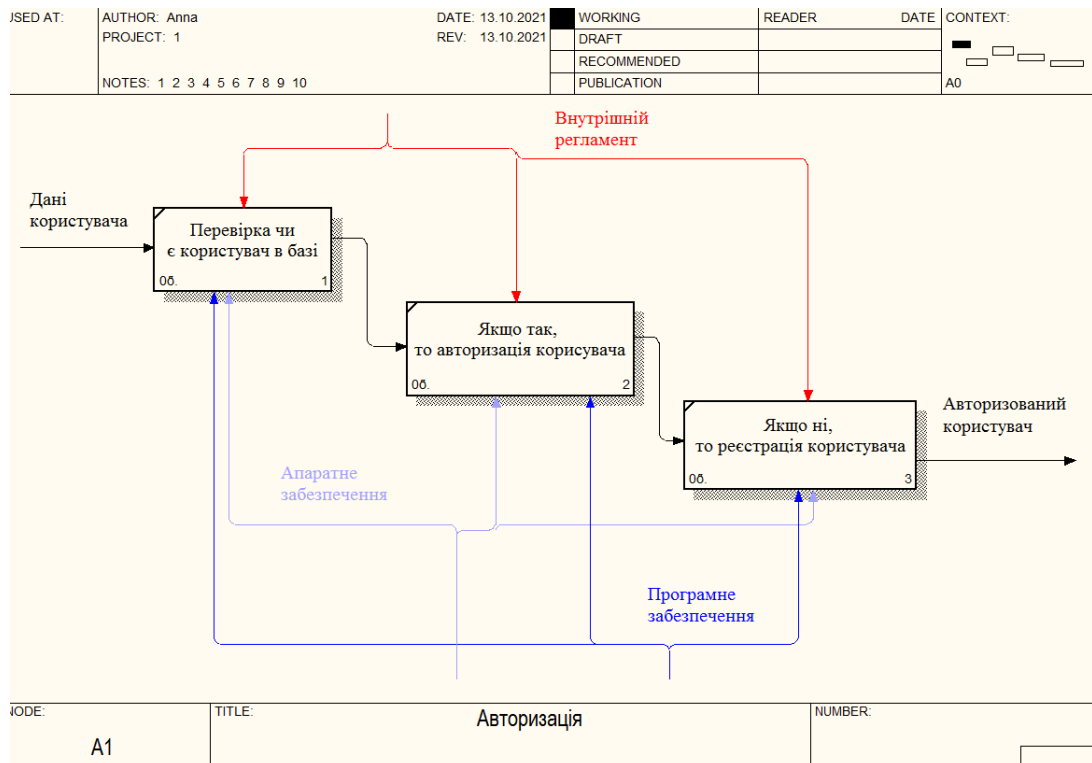


Рисунок 2.3 – Діаграма декомпозиції процесу "Авторизація користувачів"

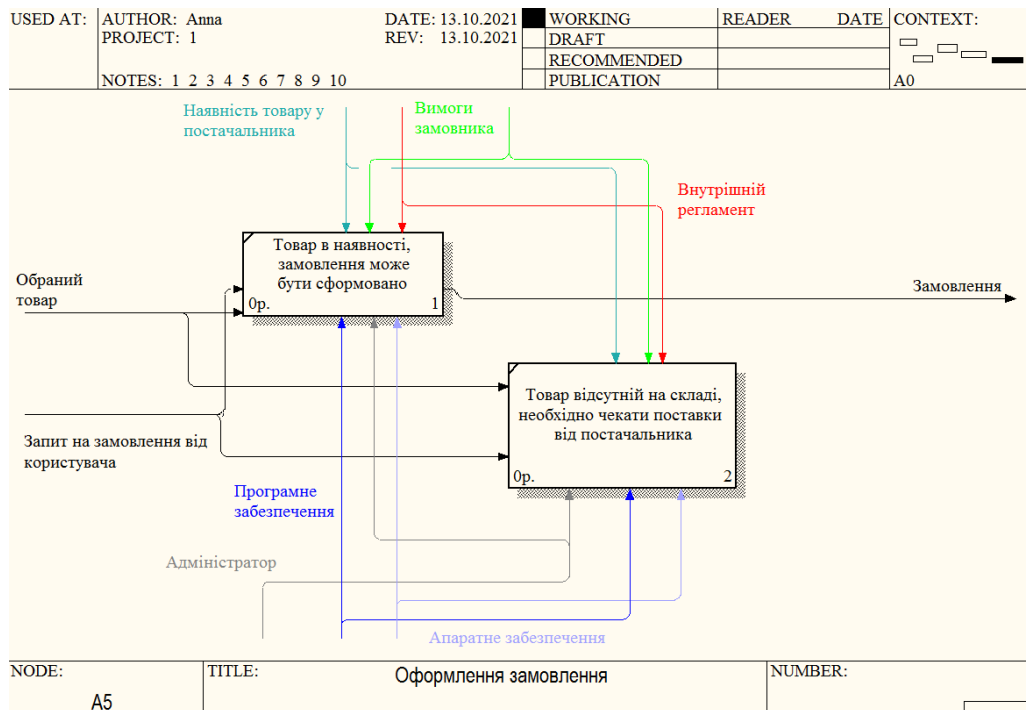


Рисунок 2.4 – Діаграма декомпозиції процесу "Оформлення замовлення"

Таким чином, були спроектовані основні бізнес – процеси підприємства, необхідні для наступних етапів проектування інформаційної системи.

На рисунку 2.5 діаграма «Дерево вузлів» показує ієрархію робіт в моделі. Діаграма «Дерево вузлів» дозволяє переглянути всю модель.

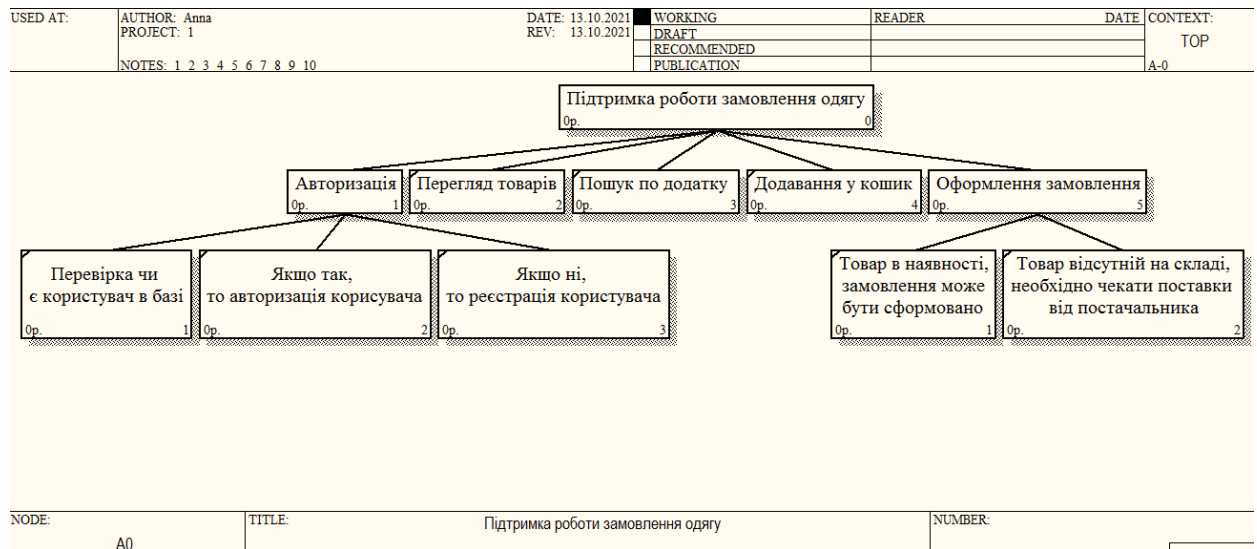


Рисунок 2.5 – Діаграма дерева вузлів "Діяльність магазину одягу"

2.3 Концептуальне проектування інформаційної моделі

Схема бази даних – її структура, описана на формальній мові, що підтримується системою управління базами даних.

Концептуальна модель – відображає інформаційні об'єкти, їх властивості і зв'язки між ними без вказівки способів фізичного зберігання інформації (модель предметної області, іноді її також називають інформаційно-логічною чи інфологічною моделлю) [6].

Інформаційними об'єктами зазвичай є сутності – відокремлені об'єкти або події, інформацію про які необхідно зберігати, які мають певні набори властивостей – атрибутів. Схема бази даних була розроблена в нотації IDEF1X. Характеризується високою швидкістю, стійкістю і простотою

використання. Схема бази даних приведена на рис. 2.6, вона відображає таблиці і зв'язки між ними.

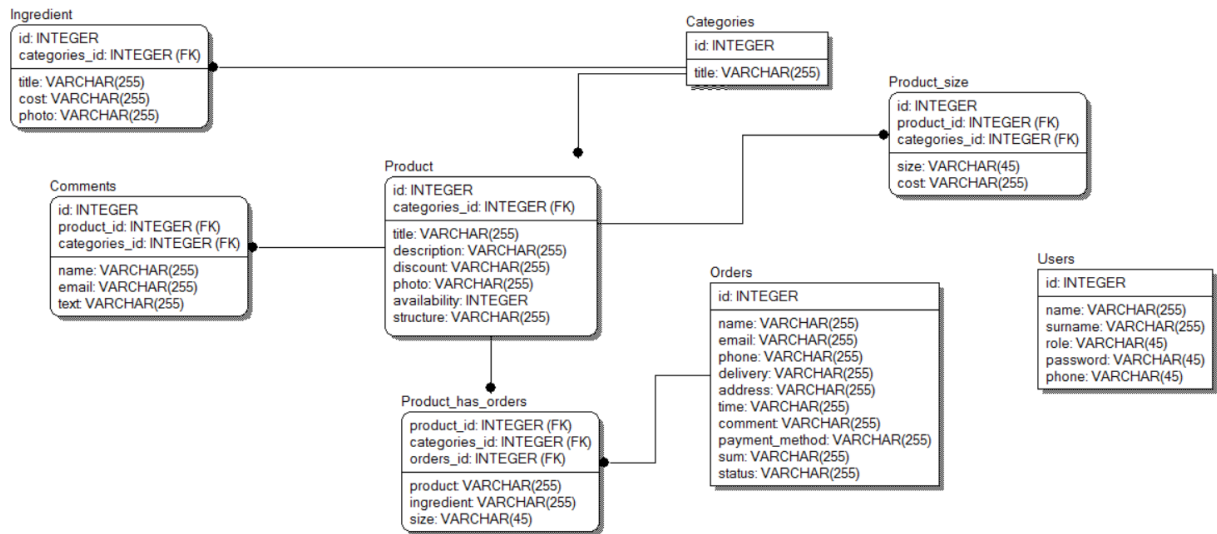


Рисунок 2.6 – ER – діаграма

2.4 Логічне проектування інформаційної моделі

Логічне проектування – перетворення концептуального представлення в логічну структуру бази даних, включаючи проектування зв'язків [7].

Основним завданням логічного проектування є розробка логічної схеми, орієнтованої на вибрану СУБД. Оскільки переважна більшість сучасних СУБД - реляційні, то і концептуальну модель БД слід відображати на реляційну модель.

У основі реляційної моделі використовується поняття “відносини”, яке використовується для уявлення набору екземплярів об'єкту (сутність) та відносин (зв'язків) між об'єктами [8].

Відношення представляється як певним чином організована таблиця.

Таблиця 2.1 «Users» зберігає інформацію про клієнтів у додатку. Опис таблиці представлено нижче.

Таблиця 2.1 – Структура таблиці «Users»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|------------|--------------|-----------------------|---------|--|
| id | INT(11) | PK, Auto_increment | | Id |
| name | Varchar(255) | | | Зберігає ім'я користувача |
| surname | Varchar(255) | | | Зберігає прізвище користувача |
| role | Varchar(45) | | | Зберігає роль користувача |
| phone | Varchar(45) | | | Зберігає мобільний телефон користувача |

Таблиця 2.2 «Product» зберігає інформацію про товари у додатку. Опис таблиці представлено нижче.

Таблиця 2.2 – Структура таблиці «Product»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|--------------|--------------|-----------------------|------------------------|-----------------------------|
| id | INT(11) | PK, Auto_increment | | Id |
| categoriesId | INT(11) | FK | Від: Categories(id) | Id |
| title | Varchar(255) | | | Зберігає назву товару |
| description | Varchar(255) | | | Зберігає опис товару |
| discount | Varchar(255) | | | Зберігає знижку товару |
| photo | Varchar(255) | | | Зберігає фото товару |
| availability | INT(11) | | | Зберігає доступність товару |
| structure | Varchar(255) | | | Зберігає склад товару |

Таблиця 2.3 «Ingredient» зберігає інформацію про елемент товару. Структура таблиці представлена нижче.

Таблиця 2.3 – Структура таблиці «Ingredient»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|--------------|--------------|-----------------------|-------------------------|--------------------------------|
| categoriesId | INT(11) | FK | Від: Categories (id) | Id |
| id | INT(11) | PK, Auto increment | | Id |
| title | Varchar(100) | | | Зберігає назву елемента товару |
| cost | Varchar(100) | | | Зберігає ціну елемента товару |
| photo | INT(5) | | | Зберігає фото елемента товару |

Таблиця 2.4 «Orders» є необхідною таблицею, бо у ній зберігається уся інформація, щодо заказів.

Таблиця 2.4 – Структура таблиці «Orders»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|----------------|--------------|-----------------------|---------|---------------------------------|
| id | INT(11) | PK, Auto_increment | | Id |
| name | Varchar(255) | | | Зберігає ім'я покупця |
| email | Varchar(255) | | | Зберігає пошту покупця |
| delivery | Varchar(255) | | | Зберігає метод доставки |
| phone | Varchar(255) | | | Зберігає телефон покупця |
| address | Varchar(255) | | | Зберігає адресу покупця |
| time | Varchar(255) | | | Зберігає час замовлення |
| comment | Varchar(255) | | | Зберігає коментар до замовлення |
| payment_method | Varchar(255) | | | Зберігає метод оплати |
| status | Varchar(255) | | | Зберігає статус замовлення |
| sum | Varchar(255) | | | Зберігає суму замовлення |

Таблиця 2.5 «Product size» є необхідною таблицею, бо у ній зберігається уся інформація, щодо розміру замовлення. Структуру таблиці наведено нижче.

Таблиця 2.5 – Структура таблиці «Product size»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|--------------|--------------|-----------------------|-------------------------|------------------------------|
| Id | INT(11) | PK, Auto_increment | | Id |
| productId | INT(11) | FK | Від: Product (id) | Id |
| categoriesId | INT(11) | FK | Від: Categories (id) | Id |
| Size | Varchar(45) | | | Зберігає розмір замовлення |
| Cost | Varchar(255) | | | Зберігає вартість замовлення |

Таблиця 2.6 «Categories» зберігає інформацію про категорію товару. Структура таблиці представлена нижче.

Таблиця 2.6 – Структура таблиці «Categories»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|------------|--------------|-----------------------|---------|---------------------------------|
| Id | INT(11) | PK, Auto_increment | | Id |
| Title | Varchar(255) | | | Зберігає назву категорії товару |

Таблиця 2.7 «Comments» зберігає інформацію про коментар до товару.

Таблиця 2.7 – Структура таблиці «Comments»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|------------|-----------|-----------------------|---------|------|
| 1 | 2 | 3 | 4 | 5 |
| Id | INT(11) | PK, Auto_increment | | Id |

Продовження таблиці 2.7

| 1 | 2 | 3 | 4 | 5 |
|-----------|--------------|----|----------------------|--------------------------|
| productId | INT(11) | FK | Від: Product (id) | Id |
| Name | Varchar(255) | | | Зберігає назву коментаря |
| Email | Varchar(255) | | | Зберігає пошту |
| text | Varchar(255) | | | Зберігає текст коментаря |

Таблиця 2.8 «Product_has_orders» зберігає інформацію про товар, який є наявний у замовленні. Структура таблиці представлена нижче.

Таблиця 2.8 – Структура таблиці «Product_has_orders»

| Назва поля | Тип даних | Властивості | Зв'язок | Опис |
|--------------|--------------|-----------------------|-------------------------|----------------------------|
| orderId | INT(11) | PK, Auto_increment | | Id |
| productId | INT(11) | FK | Від: Product (id) | Id |
| categoriesId | INT(11) | FK | Від: Categories (id) | Id |
| product | Varchar(255) | | | Зберігає назву товару |
| ingredient | Varchar(255) | | | Зберігає елемент товару |
| size | Varchar(255) | | | Зберігає розмір замовлення |

2.5 Діаграма варіантів використання

Use case (Варіанти використання) – це функціональна вимога. Даний вид діаграм в основному використовується для опису функціональних вимог до системи, для опису предметної області з метою кращого розуміння функціонування системи.

Суть діаграми полягає в наступному: розроблена система представлена у вигляді сутностей або суб'єктів, які взаємодіють із системою, використовуючи так звані випадки використання. У цьому випадку учасником є будь-яка організація, яка взаємодіє із системою ззовні. Це може бути людина,

технічний пристрій, програма або будь-яка інша система і може виступати джерелом впливу на систему моделювання, визначену розробником. У свою чергу, випадки використання використовуються для опису послуг, які система надає учасникам [9]. Іншими словами, кожен випадок використання визначає певний набір операцій, які система виконує після розмови з учасниками.

Приклад діаграми інтернет магазину представлено на рис. 2.7. Розглянемо інформацію про акторів (табл. 2.9) та опис варіантів використання (табл. 2.10).

Таблиця 2.9 – Структура таблиці «Опис акторів»

| Назва | Опис |
|---------------|---|
| Користувач | Користувач інтернет магазину |
| Адміністратор | Користувач, що використовує функції «адміністратора». |
| База даних | База даних, що зберігає інформацію. |

Таблиця 2.10 – Структура таблиці «Опис варіантів використання»

| Назва | Опис |
|------------------------|--|
| Авторизація | Функція входу в систему. |
| Перегляд товару | Користувач має можливість переглянути товар у додатку. |
| Пошук товару | Можливість виконувати пошук бажаного товару у додатку. |
| Додання товару в кошик | Функція для додання обраного товару в кошик для оформлення замовлення. |
| Редагування даних | Адміністратор може редагувати дані у додатку. |
| Додання товару | Адміністратор може додати новий товар у додатку. |
| Перегляд замовлень | Адміністратор може переглядати замовлення користувачів. |

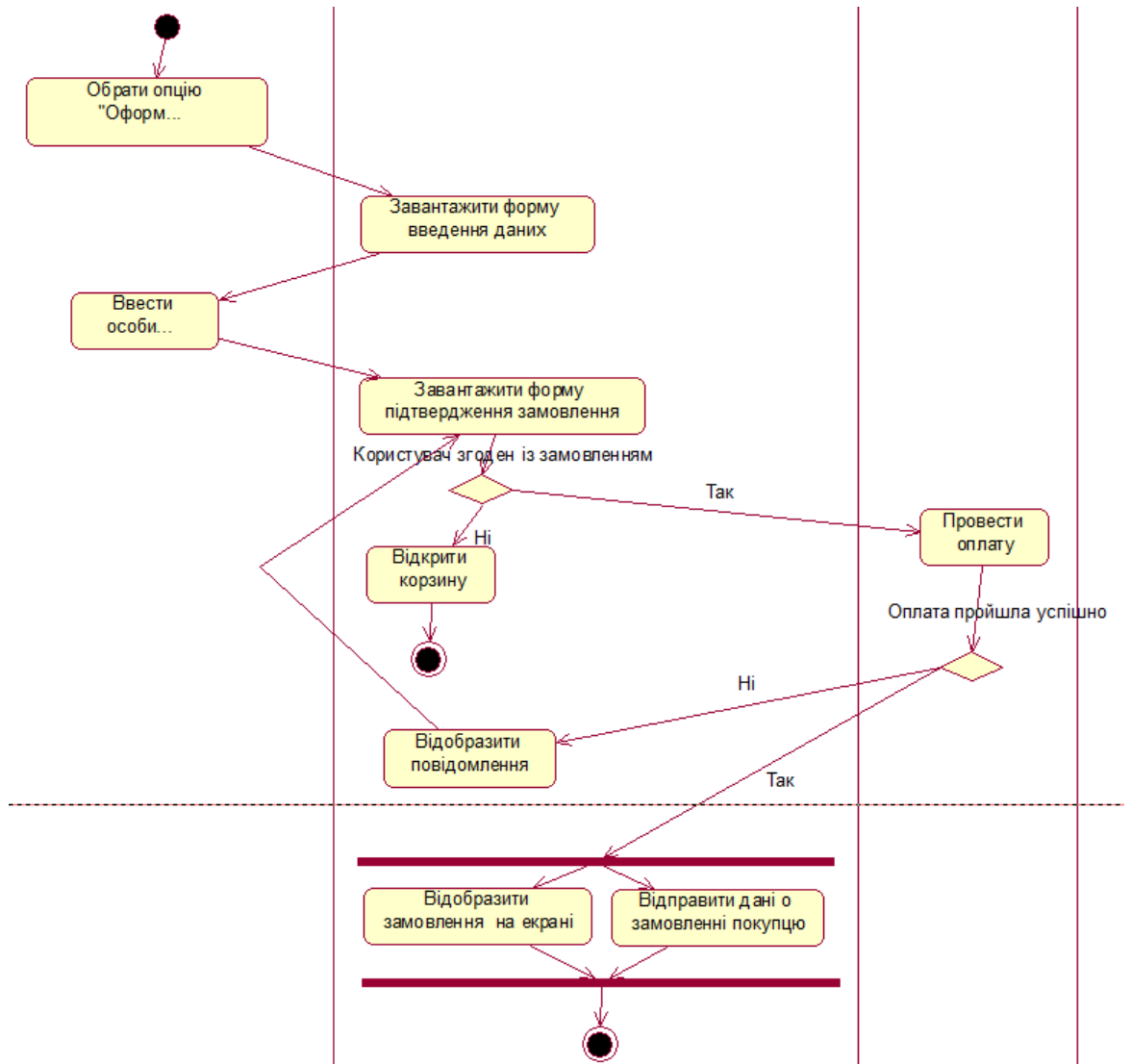


Рисунок 2.8 – Діаграма взаємодії бізнес- процесу "Оформлення замовлення"

Діаграма послідовності – відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень [10].

Іншими словами, діаграма послідовностей відображає часові особливості передачі і прийому повідомлень об'єктами.

Діаграми послідовностей можна використовувати для уточнення діаграм прецедентів. На діаграмі послідовності відображаються тільки ті об'єкти, які безпосередньо беруть участь у взаємодії.

На рисунку 2.9 представлена діаграма послідовності дій бізнес - процесу "Надання послуг".

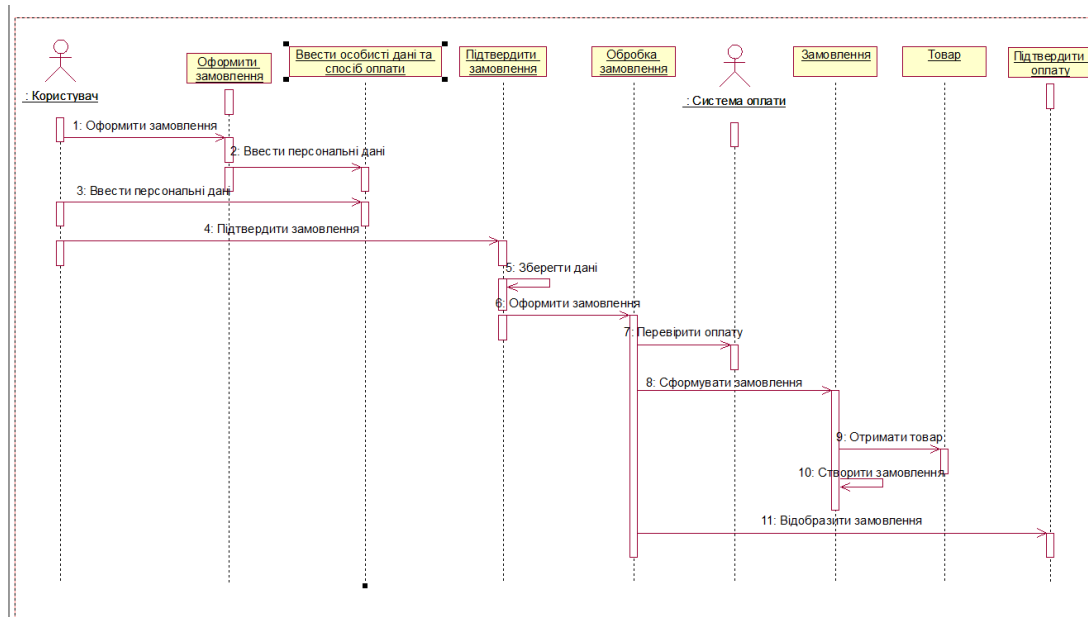


Рисунок 2.9 – Діаграма послідовності бізнес- процесу "Надання послуг"

2.7 Діаграма потоків даних

На моделі інформаційних потоків організації (DFD) безпосередньо показані джерела та одержувачі інформації, потоки даних між процесами, а так само як зберігаються дані. На концептуальній діаграмі (рис. 2.10) демонструється зв'язок системи з її оточенням.

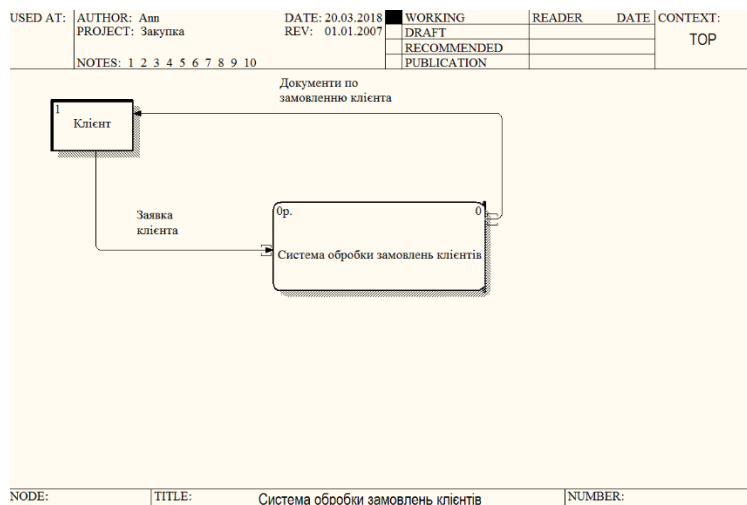


Рисунок 2.10 – Концептуальна діаграма

На рисунку 2.11 демонструється модель потоків даних для функції «Обробка замовлень клієнтів». На діаграмі є процеси, зовнішні сутності і зв'язку між ними.

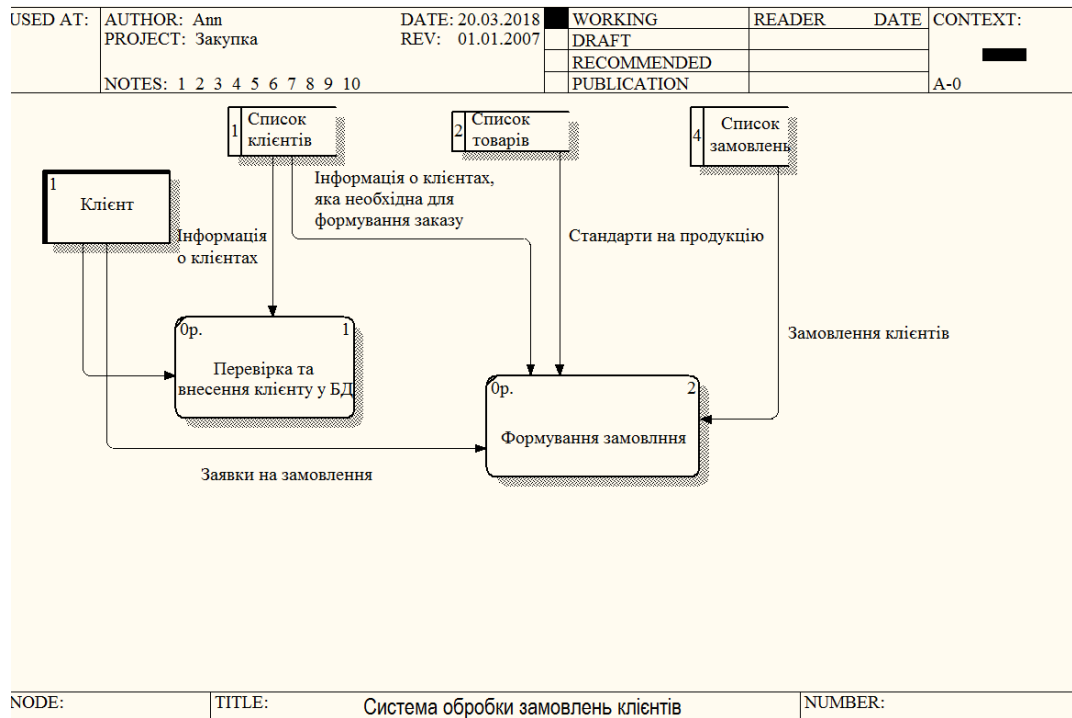


Рисунок 2.11 – Модель потоків даних для функції "Обробка замовлень клієнтів"

2.8 Діаграма класів

Діаграма класів – це основа будь-якої документації програмного проекту. Модель класів лежить в основі об'єктно - орієнтованого програмування. Вона описує загальний стан всієї програмної системи, так і її поведінку. Кожний клас з цієї моделі інкапсулює свій стан (це поля класу) та пропонує сервіси для керування цим станом (на основі методів). Для ІС діаграма класів виглядає наступним чином (див. рис. 2.12).

Діаграма класів дозволяє формалізувати логічну модель вашої програми, на рівні структурних елементів системи, тобто класів. Вона надає статичне

представлення про структуру програми – з яких класів вона складається, які зв'язки між цими класами, з чого складається кожний клас [11].

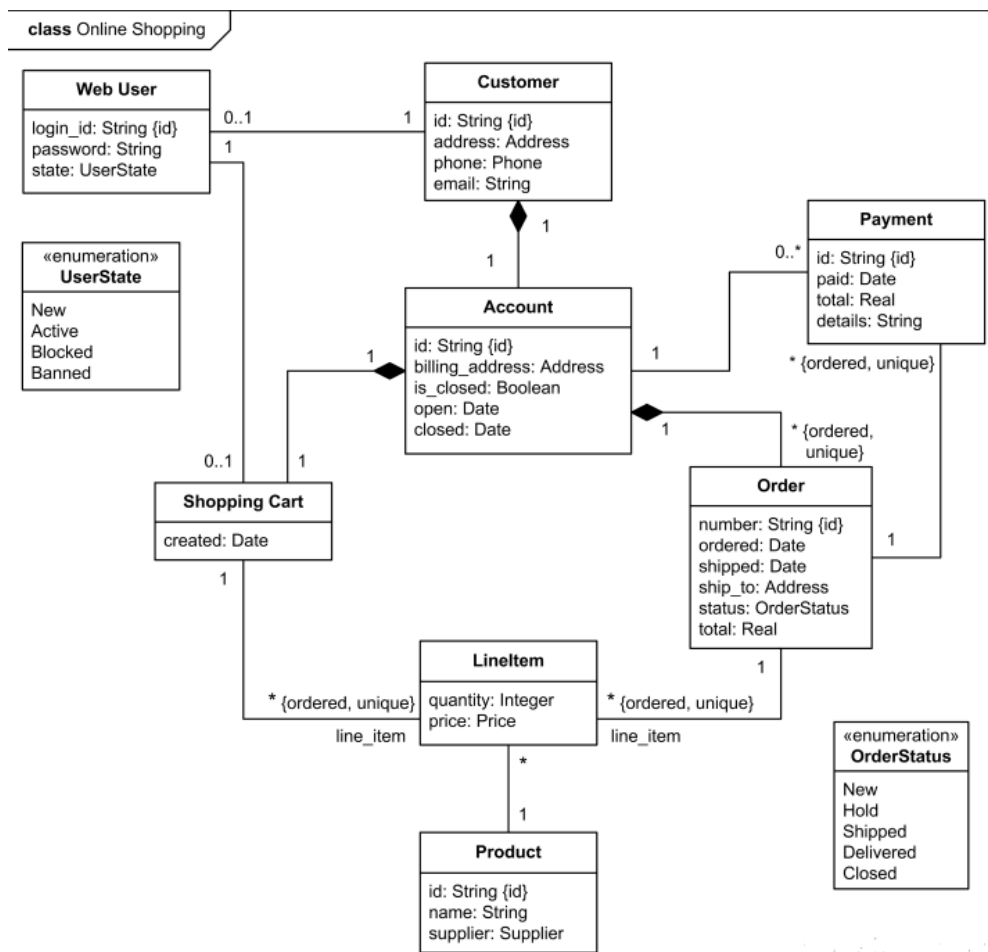


Рисунок 2.12 – Діаграма класів

2.9 Діаграма розгортутання та компонентів

Діаграма розгортутання відображає фізичні взаємозв'язку між програмними і апаратними компонентами системи. Вона є хорошим засобом для того, щоб показати маршрути переміщення об'єктів і компонентів в розподіленій системі. Кожен вузол на діаграмі розміщення являє собою певний тип обчислювального пристрою – в більшості випадків, частина апаратури. Діаграму розгортутання наведено на рисунку 2.13.

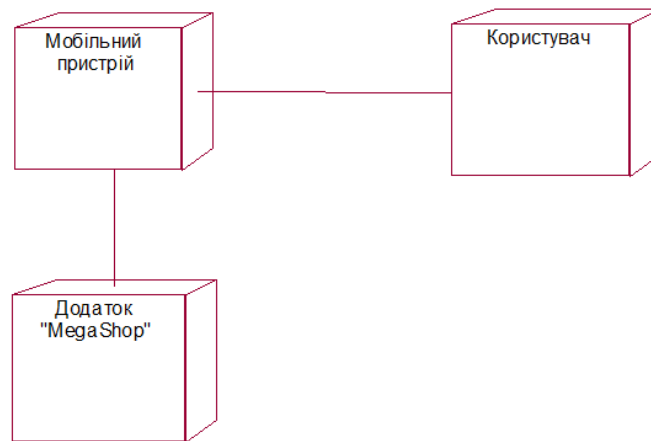


Рисунок 2.13 – Діаграма розгортвання

Діаграма компонентів відображає організацію компонентів і залежності між ними. Компонент – це фізичний елемент реалізації (компоненти вихідного коду, бінарного коду, що виконують компоненти) з чітко визначеним інтерфейсом, призначений для використання в якості заміної частини системи [12]. Діаграма компонентів включає в себе 7 компонентів. Дану діаграму наведено на рисунку 2.14.

Компонент Головний модуль – є головним, служить для видачі необхідної інформації клієнту.

Компонент Обробка запиту клієнта – служить для обробки запиту клієнта, отриманого від головного модуля, через нього надходить інформація в базу даних категорій товару, базу даних клієнтів, базу даних постачальників і на екран виводиться інформація, що цікавить клієнта інформація.

Компонент БД клієнтів – містить в собі всю інформацію про зареєстрованих клієнтів магазину.

Компонент БД постачальників – містить в собі всю інформацію про постачальників магазину.

Компонент БД категорій товару – містить в собі всю інформацію про товари, колір, наявність, розмір, кількість.

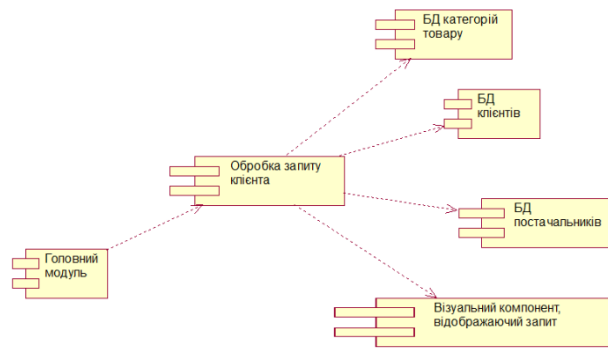


Рисунок 2.14 – Діаграма компонентів

2.10 Діаграма кооперації

Головна особливість діаграми кооперації полягає в можливості графічно представити не тільки послідовність взаємодії, але і всі структурні відносини між об'єктами, що беруть участь в цій взаємодії. Дану діаграму зображено на рисунку 2.15.

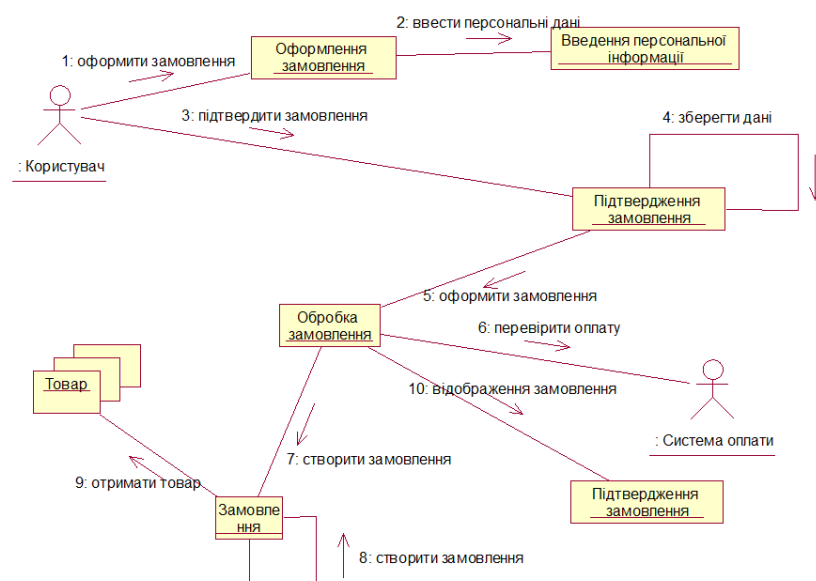


Рисунок 2.15 – Діаграма кооперації

3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКА

3.1 Вибір бази даних

Для зберігання даних про товари в магазині, клієнтів було обрано базу даних – MongoDB. Це Open Source система керування базами даних, в якій дані зберігаються як колекції (групи) документів. Документи представлені в людино-зрозумілому формат JSON (JavaScript Object Notation).

MongoDB реалізує новий підхід до побудови баз даних, де немає таблиць, схем, запитів SQL, зовнішніх ключів та багатьох інших речей, які притаманні об'єктно-реляційним базам даних. На відміну від реляційних баз даних MongoDB пропонує документо-орієнтовану модель даних, завдяки чому MongoDB працює швидше, має кращу масштабованість, її легше використовувати [13].

Якщо реляційні бази даних зберігають рядки, MongoDB зберігає документи. На відміну від рядків, документи можуть зберігати складну за структурою інформацію. Документ можна подати як сховище ключів та значень. Ключ представляє просту мітку, з якою асоційовано певний шматок даних.

Однак за всіх відмінностей є одна особливість, яка зближує MongoDB та реляційні бази даних. У реляційних СУБД зустрічається таке поняття як первинний ключ. Це поняття описує стовпець, який має унікальні значення. У MongoDB для кожного документа є унікальний ідентифікатор, який називається `_id`. І якщо явно не вказати його значення, то MongoDB автоматично згенерує для нього значення [14].

Кожному ключу порівнюється певне значення. Але тут також треба враховувати одну особливість: якщо в реляційних базах є чітко окреслена структура, де є поля, і якщо якесь поле не має значення, то йому (залежно від налаштувань конкретної бд) можна надати значення NULL. У MongoDB все

інакше. Якщо якомусь ключу не зіставлено значення, цей ключ легко опускається у документі і використовується.

Якщо в традиційному світі SQL є таблиці, то світі MongoDB є колекції. І якщо в реляційних БД таблиці зберігають однотипні жорстко структуровані об'єкти, то в колекції можуть містити різні об'єкти, що мають різну структуру і різний набір властивостей.

Переваги MongoDB:

- Схема менша – MongoDB – це база даних документів, в якій одна колекція містить різні документи. Кількість полів, зміст та розмір документа можуть відрізнитися від одного документа до іншого.
- Глибока ймовірність запити. MongoDB підтримує динамічні запити до документів, використовуючи засновану на документі мову запитів, майже таку ж потужну, як SQL.
- Простота масштабування - MongoDB легко масштабується.

В результаті було розроблено такі колекції як Banner, Category, Colour, Fabric, Feature_Img, Pattern, Product_Image, Product, Product_Colour, Product_pattern, Product_Size, Size, Sub_category, User (див. рис. 3.1).

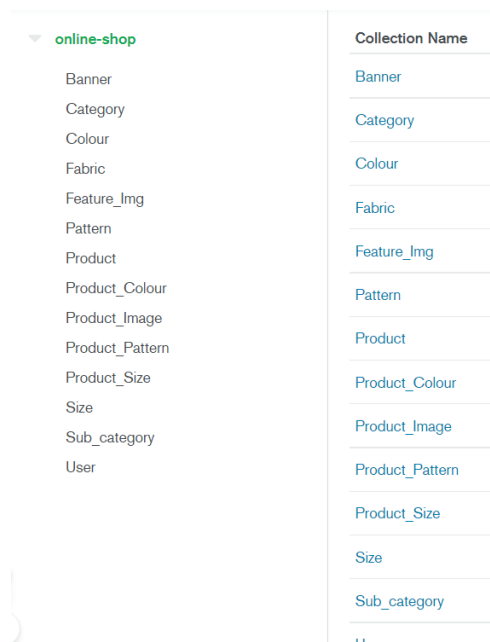


Рисунок 3.1 – База даних додатку «MegaShop»

3.2 Розробка інтерфейсу

Графічний інтерфейс в операційній системі Android представляє собою ієрархією об'єктів `android.view.View` і `android.view.ViewGroup`. Кожен об'єкт `ViewGroup` представляє контейнер, який містить і впорядковує дочірні об'єкти `View`. Об'єкти `View` є елементами управління та інші віджети, наприклад, кнопки, текстові поля і т.д., через які користувач взаємодіє з програмою (див. рис. 3.4).

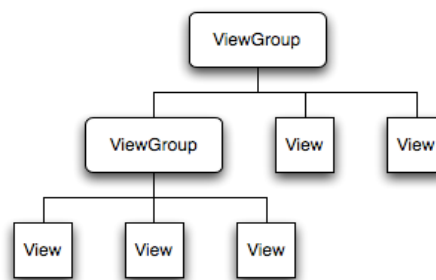


Рисунок 3.4 – Ієрархія об'єктів `android.view.View`

Основу обміну інформацією користувача із системою складають діалоги. Інтерфейс програми має бути зручним та інтуїтивно зрозумілим, не викликати роздратованості.

При запуску додатку відбувається відображення форми для авторизації клієнта (див. рис. 3.5).

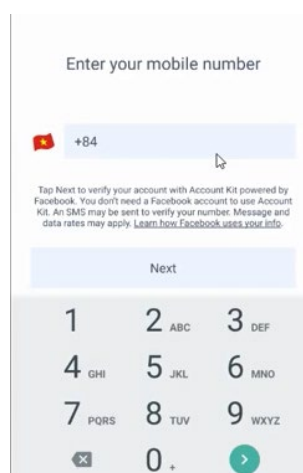


Рисунок 3.5 – Форма авторизації

Користувачу пропонується авторизуватися використовуючи свій номер телефону. Після введення мобільного телефону, необхідно натиснути кнопку «Next». Після цього з'явиться форма вводу коду, який прийшов на введений номер телефону.

На головній сторінці можна подивитись акції та категорії товарів, які представлені в магазині (див. рис. 3.6). Натиснувши «гамбургер меню», користувач може побачити навігаційну панель категорій та під категорій продукції магазину (див. рис. 3.7). Якщо користувач натисне на категорію товару, з'явиться картки з товаром, які містять в собі зображення, назву, ціну (див. рис. 3.8).

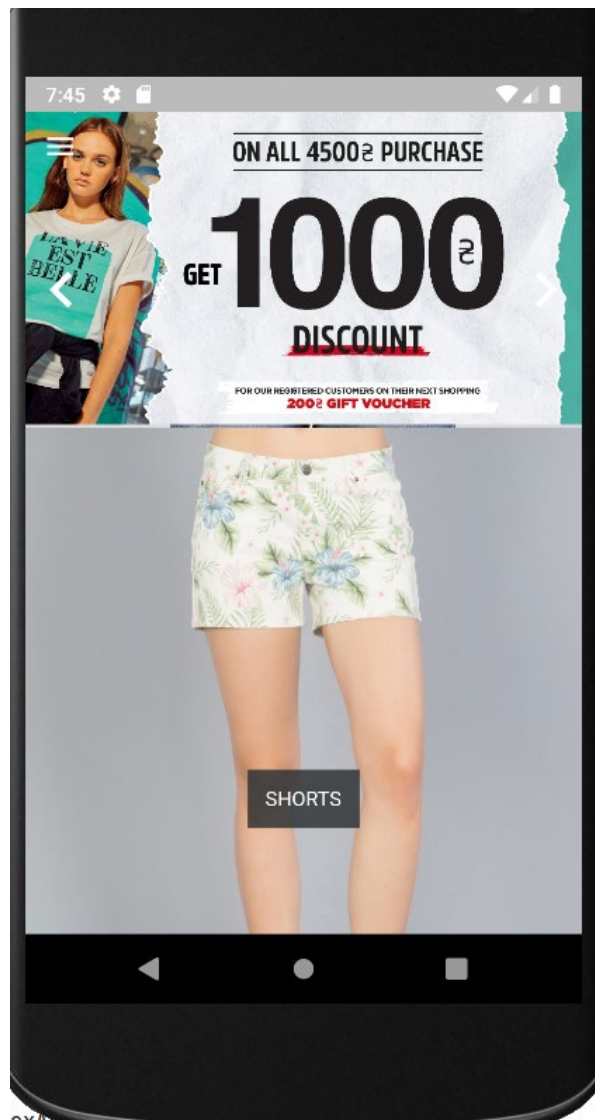


Рисунок 3.6 – Головна сторінка додатку

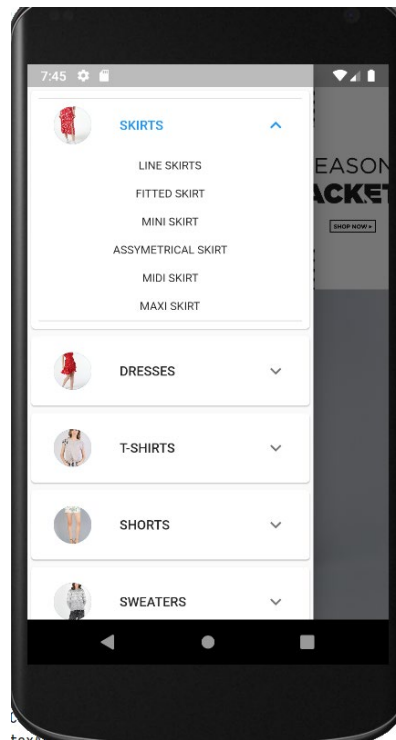


Рисунок 3.7 – Навігаційна панель зі списком категорій та підкатегорій магазину

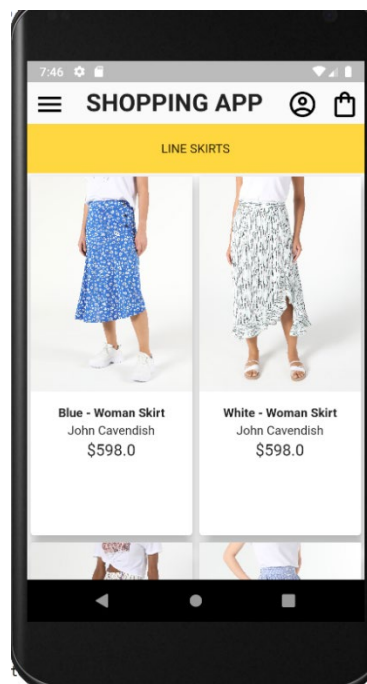


Рисунок 3.8 – Товари обраної категорії

На сторінці товарів користувач має можливість перейти до сторінки профілю, а також до кошику та подивитись, які товари були обрані.

Натиснувши на картку товару відбувається перехід на сторінку з детальної інформацією про товар (див. рис. 3.9 – 3.11).

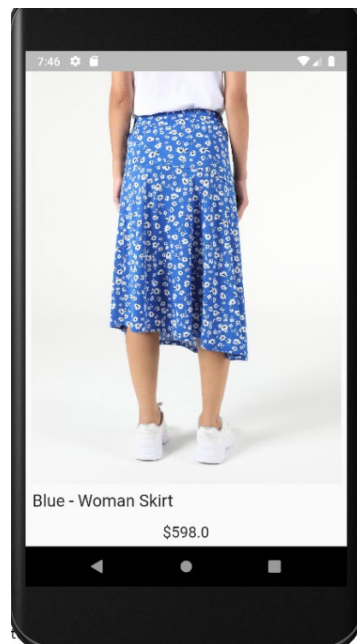


Рисунок 3.9 – Сторінка з обраним товаром

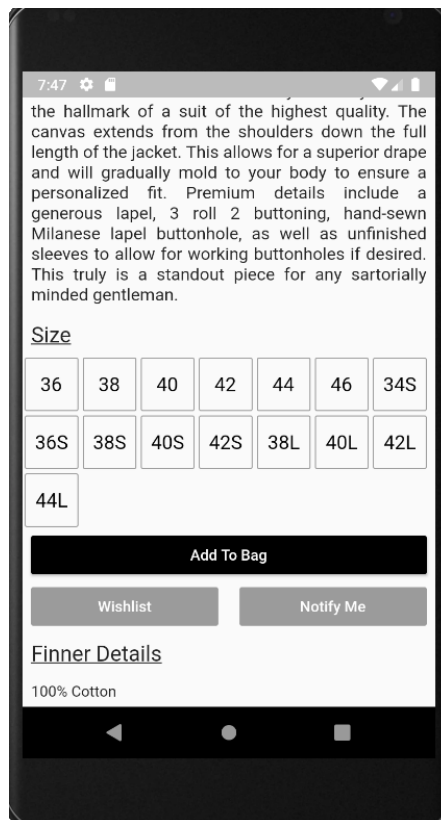


Рисунок 3.10 – Сторінка з обраним товаром

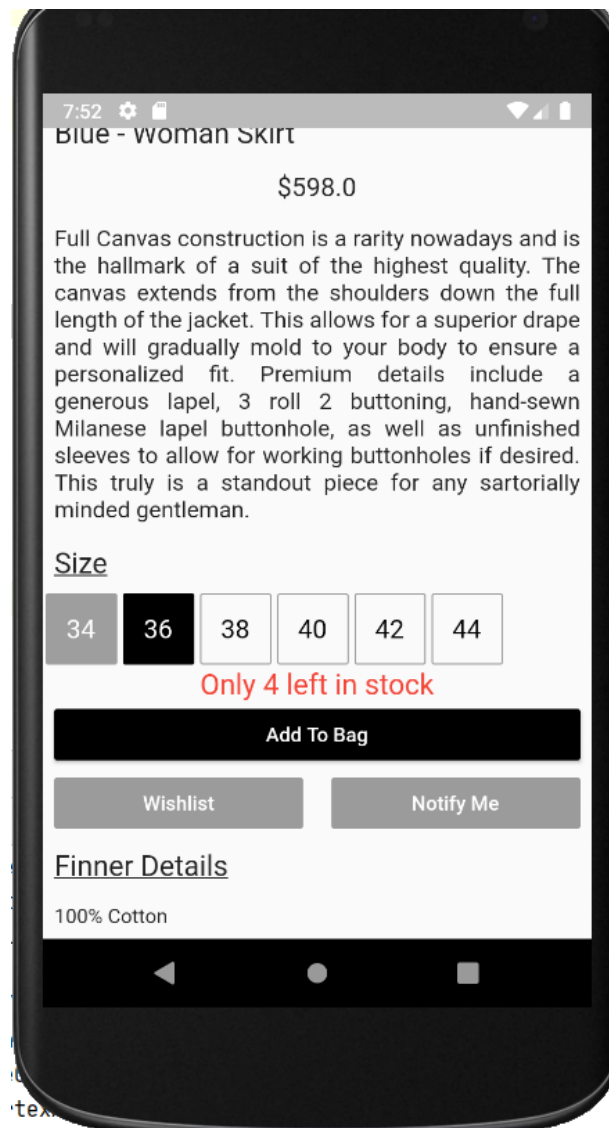


Рисунок 3.11 – Сторінка з обраним товаром

На даній сторінці можна обрати розмір і колір одягу, задати кількість обраного товару для замовлення, а також прочитати опис товару, з якого матеріалу виготовлений та подивитись які розміри, доступні в наявності. При натисканні на кнопку «Add to Bag» товар з вказаними розміром, кольором і кількістю потрапить в кошик. Також користувач має можливість додати товар до списку бажань або, обравши розмір, додати товар у кошик (див. рис. 3.12). Якщо товар у обраного розміру немає в наявності, натиснувши на кнопку «Notify me» користувач отримає повідомлення, коли товар з'явиться. Якщо мало товару залишилося в наявності (<5), тоді повідомлення буде показано користувачу.



Рисунок 3.12 – Кошик користувача

Сторінка кошику містить список обраного товару який за бажанням можна видалити, змінити кількість. Під списком товару знаходиться блок з загальною ціною замовлення та інформацією про доставку.

Сторінка для оформлення замовлення складається з форми для заповнення контактних даних про замовника. Також є загальна сума замовлення і кількість обраних товарів. Користувач має можливість змінити свою адресу, ім'я та номер телефону (див. рис. 3.13 – 3.14).

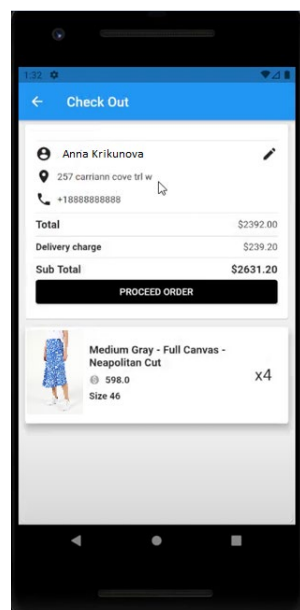


Рисунок 3.13 – Оформлення замовлення

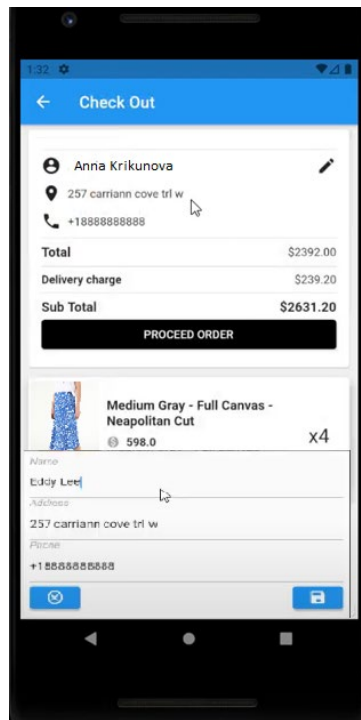


Рисунок 3.14 – Оформлення замовлення

Після заповнення всіх необхідних полів потрібно натиснути кнопку «Proceed order» після чого з'явиться повідомлення про успішне замовлення і детальну інформацію про замовлення. Користувач також має можливість подивитись історію замовлень (див. рис. 3.15).

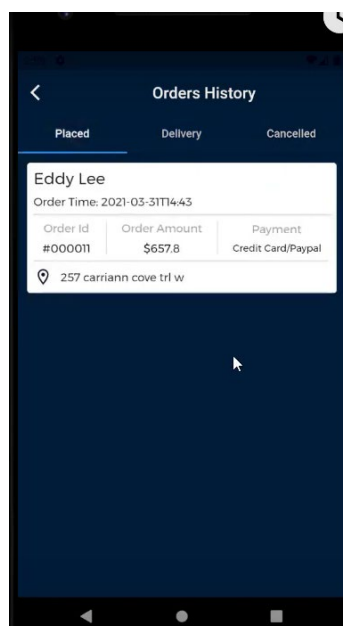


Рисунок 3.15 – Історія замовлень

Для управління сайтом, додання товару і обробки замовлень додаток має адміністративну панель. Щоб до неї потрапити необхідно авторизуватись. Для авторизації потрібно ввести номер телефону, в базі даних у кожного користувача є поле “Role”, за допомогою якого визначається це користувач адміністратор або звичайний покупець (див. рис. 3.16).

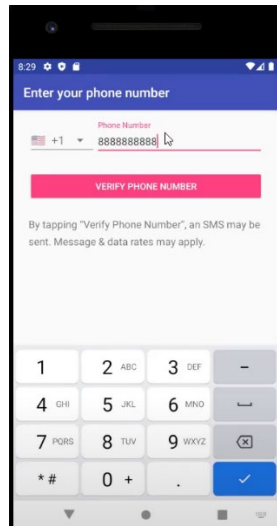


Рисунок 3.16 – Форма авторизації в адміністративній панелі

Після авторизації в якості адміністратора користувач потрапляє на сторінку адміністративної панелі. На якій знаходяться вкладки із замовленнями, інформація про доставку та скасовані замовлення (див. рис. 3.17).

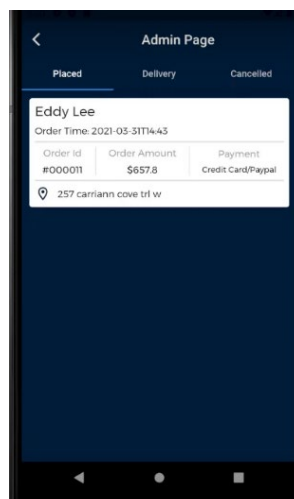


Рисунок 3.17 – Адміністративна панель

На картці із замовленням адміністратор має можливість змінити статус замовлення (див. рис. 3.18). Всього є 3 статуси: Placed (замовлення створено), Shipped (замовлення доставлено), Cancelled (замовлення скасовано).

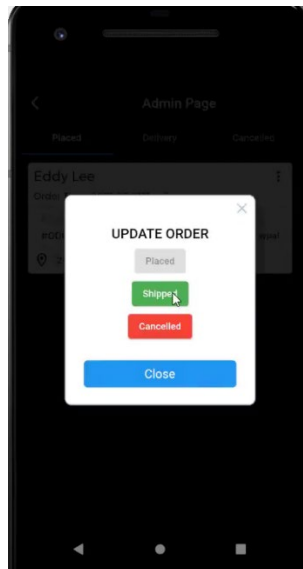


Рисунок 3.18 – Форма зміни статусу замовлення

Якщо адміністратор змінить статус замовлення на Shipped, замовлення з’явиться у вкладці “Delivery” (див. рис. 3.19). Адміністратор також має можливість скасувати замовлення, в такому випадку замовлення з’явиться у вкладці “Cancelled” (див. рис. 3.20).

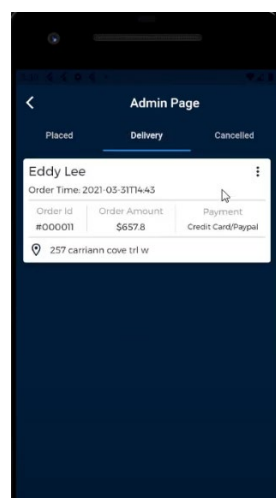


Рисунок 3.19 – Вкладка із доставленими замовленнями

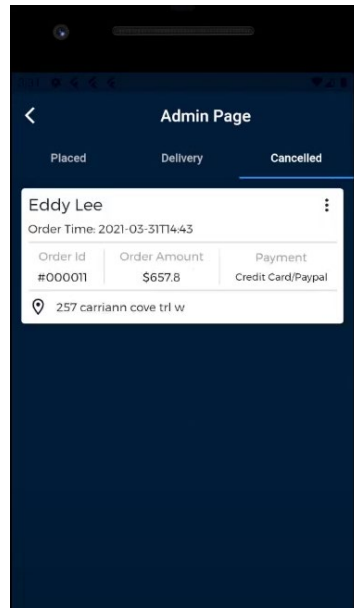


Рисунок 3.20 – Вкладка із скасованими замовленнями

У додатку адміністратор має можливість побачити статистику по доставленим замовленням. На діаграмі можна подивитися статистику за сьогодні, 7 днів, 30 днів, 45 днів (див. рис. 3.21).

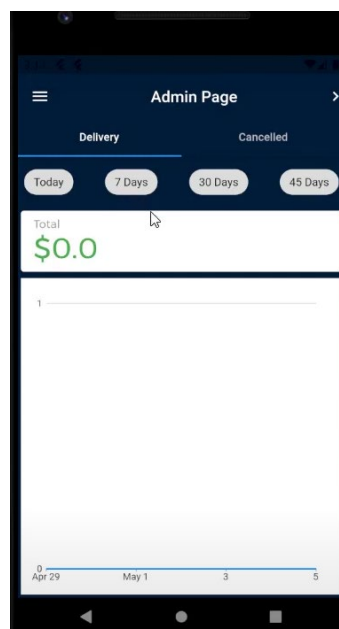


Рисунок 3.21 – Статистика доставлених замовлень

Код проекту можна знайти за посиланням в репозиторії на Github:
https://github.com/AnnaKrikunova/flutter_shopping_app.

3.3 Тестування додатку

Тестовий випадок (testcase) – це сукупність кроків, конкретних умов та параметрів, необхідних для перевірки реалізації тестованої функції або її частини. Тестовий випадок - це найменша одиниця тестування, яку можна самостійно виконати від початку до кінця.

Тестові випадки поділяються за очікуваним результатом на позитивні та негативні. Позитивний тест-кейс використовує тільки коректні дані та перевіряє правильне виконання додатком викликаної функції, що викликається. Негативний тест-кейс оперує як коректними, так й некоректними даними (мінімум 1 некоректний параметр) та ставить за мету перевірку виняткових ситуацій (спрацьовування валідаторів). У таблиці 3.1 представлено тест кейс для прецеденту «Вхід у систему».

Таблиця 3.1 – «Вхід у систему»

| | | |
|-------------------|---|---|
| 1 | | |
| Функція | Вхід у систему | |
| Передумова | Відкрита стартова сторінка. | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрити додаток | 1. Відображається назва додатка . 2. На формі 1 поле - "Номер телефону". 3. Після введення номеру телефону з'являється форма з полем "Код". 4. Кнопка "Next" доступна. |
| 2 | Введення номеру телефону та коду, який прийшов на телефон, потім натиснення кнопки «Next» | 1.Перехід на головну сторінку додатку |

У таблиці 3.2 представлено тест кейс для прецеденту «Перегляд товарів в обраній категорії».

Таблиця 3.2 – Перегляд товарів в обраній категорії

| | | |
|-------------------|---|---|
| 2 | | |
| Функція | Перегляд товарів в обраній категорії | |
| Передумова | Відкрито головне вікно, обрана категорія. | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрити вікно перегляду товарів | 1. Відображається список товарів в обраній категорії. |

У таблиці 3.3 представлено тест кейс для прецеденту «Перегляд історії замовлень».

Таблиця 3.3 – Перегляд історії замовлень

| | | |
|-------------------|---|--|
| 3 | | |
| Функція | Перегляд історії замовлень | |
| Передумова | Відкрито головне вікно, натиснута кнопка “Orders History” . | |
| # | Кроки | Очікуваний результат |
| 51 | Відкрити вікно перегляду історії замовлень | 1. Відображається список замовлень клієнта. 2. Відображається такі параметри як: дата та час замовлень, id замовлення, ціна замовлення, адреса доставки, спосіб оплати. |

У таблиці 3.4 представлено тест кейс для прецеденту «Перегляд товарів у кошику».

Таблиця 3.4 – Перегляд товарів у кошику

| | | |
|-------------------|---|--|
| 4 | | |
| Функція | Перегляд товарів у кошику | |
| Передумова | Відкрито вікно товарів, натиснута іконка “Cart” . | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрити вікно перегляду товарів у кошику | 1. Відображається список товарів, яких клієнт додав у кошик. 2. Внизу відображається загальна сума товарів. |

У таблиці 3.5 представлено тест кейс для прецеденту «Оформлення замовлення».

Таблиця 3.5 – Оформлення замовлення

| | | |
|-------------------|--|--|
| 5 | | |
| Функція | Оформлення замовлення | |
| Передумова | Відкрито вікно товарів, натиснута іконка “Cart”, заповнено поля ім’я, адреса доставки, номер телефону, склад замовлення. | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрити вікно оформлення замовлення | 1. Відображається заголовок " Check out ". 2. Відображається інформація щодо замовлення: ім’я та номер телефону покупця, адреса доставки, склад замовлення(кількість товарів, фото товарів, сума), загальна сума замовлення. 4. Кнопка "Proceed order" доступна. |

У таблиці 3.6 представлений тест кейс «Неможливість додавання у кошик товар, який недоступний у магазині».

Таблиця 3.6 – Неможливість додавання у кошик товар, який недоступний у магазині

| | | |
|-------------------|---|--|
| 6 | | |
| Функція | Неможливість додавання у кошик товар, який недоступний у магазині | |
| Передумова | Відкрито вікно перегляду товару | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрито вікно перегляду товару | 1. Відображається інформація про товар, розміри, колір, ціна, опис товару. |
| 2 | Обрати недоступний розмір(товар обраного розміру недоступний на складі) | 1.Відображення повідомлення, що товар недоступний |

У процесі проходження тестування були розроблені також і негативні тест кейси. У таблиці 3.7 представлено тест кейс для прецеденту «Вхід у систему із незареєстрованим номером телефону користувача».

Таблиця 3.7 – Вхід у систему із незареєстрованим номером телефону користувача

| | | |
|-------------------|---|---|
| 7 | | |
| Функція | Вхід у систему із незареєстрованим номером телефону користувача | |
| Передумова | Відкрита стартова сторінка. | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрити додаток | <ol style="list-style-type: none"> 1. Відображається назва додатка . 2. На формі 1 поле - "Номер телефону". 3. Після введення номеру телефону з'являється форма з полем "Код". 4. Кнопка "Next" доступна. |
| 2 | Введення некоректного номеру телефона та коду авторизації, натиснення кнопки "Next" | 1. Не було зареєстровано такий номер телефону або код авторизації не вірний. |

У таблиці 3.8 представлено тест кейс для прецеденту «Видалення товару при оформленні замовлення».

Таблиця 3.8 – Видалення товару при оформленні замовлення

| | | |
|-------------------|--|---|
| 8 | | |
| Функція | Видалення товару при оформленні замовлення | |
| Передумова | Відкрито вікно товарів, натиснута іконка "Cart", заповнено поля ім'я, адреса доставки, номер телефону, склад замовлення. | |
| # | Кроки | Очікуваний результат |
| 1 | Відкрити вікно оформлення замовлення | <ol style="list-style-type: none"> 1. Відображається заголовок "Check out". 2. Відображається інформація щодо замовлення: ім'я та номер телефону покупця, адреса доставки, склад замовлення(кількість товарів, фото товарів, сума), загальна сума замовлення. |

На основі створених тест-кейсів було створено чек-лист та проведено тестування додатку (див. табл. 3.9).

Таблиця 3.9 – Чек лист

| Функція | ОС ANDROID 8.1 | ОС ANDROID 7.1 |
|---|-----------------------|-----------------------|
| Вхід у систему | Пройдено | Пройдено |
| Перегляд товарів в обраній категорії | Пройдено | Пройдено |
| Перегляд історії замовлень | Пройдено | Пройдено |
| Перегляд товарів у кошику | Пройдено | Пройдено |
| Оформлення замовлення | Пройдено | Пройдено |
| Неможливість додавання у кошик товар, який недоступний у магазині | Пройдено | Пройдено |
| Вхід у систему із незареєстрованим номером телефону | Пройдено | Пройдено |
| Видалення товару при оформленні замовлення | Пройдено | Пройдено |

ВИСНОВКИ

В результаті виконання роботи було розроблена багатокористувацька інформаційна система по автоматизація основних бізнес-процесів функціонування інтернет магазину.

У відповідності з поставленими завданнями технічного завдання були виконані наступні етапи створення технічного проекту:

- виконано аналіз предметної області;
- обрана архітектура побудови і платформа реалізації системи;
- спроектована концептуальна модель системи "MegaShop";
- спроектована логічна модель системи "MegaShop" на основі концептуальної моделі;
- розроблена функціональна структура до складу якої увійшли: діаграми організаційної структури салонів краси, контекстні діаграми, діаграми декомпозиції, діаграми дерева вузлів, діаграми потоків даних, CASE – інструментом AllFusion Process Modeler;
- розроблені діаграми моделей даних інструментом ERwin Data Modeler (IDEF1X);
- спроектована інформаційна система інструментом Rational Rose UML 2.0 в склад якої входять діаграми: взаємодії, кооперації, прецедентів, послідовностей, діяльності, класів, компонентів, розгортання;
- здійснене тестування системи та оформлена документації до проекту.

Отже, проект буде надавати повний функціонал для продажу одягу в мережі інтернет. Опираючись на всю інформацію та знання був розроблений Android додаток для підтримки роботи магазину одягу. Всі цілі і завдання були досягнуті і виконані.

ПЕРЕЛІК ПОСИЛАНЬ

1. Казакова К., Пушилин Д. Перспективы развития интернет-торговли. URL : <https://www.fundamental-research.ru/ru/article/view?id=36472> (дата звернення: 08.09.2021).
2. Алеев А. Быстрый старт Flutter разработчика. Москва : ООО «Омега-Л», 2017. 180 с.
3. Бакетт К. Dart в действии. Москва : ДМК Пресс, 2013. 52 с.
4. Интернет-торгівля. URL : <https://retail-loyalty.org/knowledgebase-/glossary/internet-torgovlya/> (дата звернення: 15.09.2021).
5. Эрх Г. Приемы объектно-ориентированного проектирования. Паттерны проектирования. Москва : Microsoft, 2016. 366 с.
6. Грекул В., Денищенко Г., Коровкин Н. Проектирование информационных систем. Москва : Интернет-университет информационных технологий, 2005. 304 с.
7. Крэг Л. Применение UML 2.0 и шаблон проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. Москва : Вильямс, 2013. 736 с.
8. Молино П. Технологии CRM. Экспресс-курс. Москва : «Фаир-Пресс», 2004. 272 с.
9. Маглинец Ю. А. Анализ требований к автоматизированным информационным системам. Москва : Интернет-Университет Информационных Технологий; Бином, 2008. 200 с.
10. Черемных С., Семенов И., Ручкин В. Моделирование и анализ систем. IDEF-технологии: практикум. Москва : Финансы и статистика, 2006. 192 с.
11. Хансен Г. Базы данных: разработка и управление. Москва : ЗАО Издательство "БИНОМ", 1999. 704 с.

12. Борисенко В., Левыкин В., Пономарев Ю., Борисенко Т. Объектно-ориентированный анализ и проектирование ИКС на основе UML. Харьков : "ХНУРЭ", 2004. 80 с.

13. Брэдшоу Ш., Брэзил Й., Ходоров К. Mongo DB. Полное руководство. Москва : ДМК Пресс, 2020. 540 с.

14. Саймон Х. Стек MEAN. Mongo, Express, Angular, Node. Санкт-Петербург : Питер Пресс, 2017. 496 с.