# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет Комп’ютерних наук

Кафедра системотехніки

# **МІЖДИСЦИПЛІНАРНИЙ КУРСОВИЙ ПРОЕКТ**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

на тему: «Інтернет-магазин з продажу мобільних телефонів»

Керівник Міщеряков Ю.В.

Студент групи ІТКН-16-1 Троценка О.І.

Робота захищена с оцінкою «\_\_\_\_\_\_\_\_\_»

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Комісія:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, посада, прізвище, ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, посада, прізвище, ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, посада, прізвище, ініціали)

Харків 2018

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра системотехніки

Спеціальності підготовки 122 - Комп'ютерні науки

Курс 2 група ІТКН-16-1 семестр 4

ЗАВДАННЯ

на міждисциплінарний курсовий проект

студентові Троценко Олексію Ігоровичу

1. Тема роботи:Інформаційна система «Магазин з продажу мобільних телефонів»

2. Термін здачі студентом закінченої роботи 07.06.2018

3. Початкові дані до проекту:Розробити серверну і клієнтську частини інформаційної системи "Магазин з продажу мобільних телефонів". Серверна частина повинна забезпечувати взаємодію з базою даних, розроблену для платформи СУБД MySQL. Клієнтська частина повинна забезпечувати інтерфейс з користувачем. Функціональність бізнес процесів реалізується на сервері. Бізнес-функції системи для незареєстрованих користувачів: перегляд наявних мобільних телефонів; пошук телефонів по назві; перегляд інформації за обраним телефону; додавання в корзину; оформлення заказу; Реєстрація на сайті. Бізнес-функції системи для зареєстрованих користувачів: підбір мобільних телефонів в кошик; оформлення заказу; вхід в систему з визначенням статусу «user». Бізнес-функції системи для адміністраторів: вхід в систему з визначенням статусу «admin»; додавання телефонів; розподіл мобільних телефонів за категоріями; перегляд замовлень з фільтрацією по статусу ( «підтверджений», «не підтверджений», «у виконанні», «виконано»); зміна статусу обраного замовлення. Операційна система - Windows 7 або вище, програмне забезпечення: утиліта командного рядка MySQL Command Line Client; програмний пакет Workbench; програмний засіб Devart dbForge Studio for MySQL, CASE-засіб All Fusion Data Modeler (ERWin), ECLIPSE EE, BPwin.

4. Зміст пояснювальної записки (перелік питань, які підлягають розробці): Провести аналіз бізнес-процесів (бізнес-функцій) предметної області та виділити ті з них, які вимагають автоматизації; сформулювати і оформити вимоги до інформаційної системи; створити функціональну модель інформаційної системи ( «TO-BE»), використовуючи стандарт IDEF0; провести функціональне моделювання, визначити і уточнити вимоги до інформаційної системи; провести логічне моделювання БД з використанням стандарту IDEF1Х; обґрунтувати вибір платформи СУБД для реалізації БД; провести фізичне моделювання БД з використанням стандарту IDEF1Х; провести UML-моделювання проектованої клієнтської частини інформаційної системи, створивши діаграму прецедентів (Use Case Diagram), діаграму послідовності дій (Sequence Diagram), діаграму станів (Statechart Diagram), діаграму активності (Activity Diagram) і діаграму класів (Class Diagram); провести аналіз і виділити основні бізнес-процеси з поділом бізнес-функцій, що виконуються на стороні клієнтської і серверної частинах інформаційної системи; реалізувати фізичну модель БД для обраної платформи СУБД, створивши серверну частину інформаційної системи; реалізувати цілістність посилань даних, а також одну з функцій бізнес-процесу на стороні серверної частини інформаційної системи; реалізувати один або кілька бізнес-процесів на стороні клієнтської частини інформаційної системи, розробивши інтерфейс доступу до БД; розробити відповідно до ГОСТ 34.698-90 експлуатаційний документ «Посібник користувача»; підготувати відповідно ГОСТ 19.401-78 програмний документ «Текст програми».

5.Перелік графічного матеріалу (з точним визначенням обов'язкових креслень): діаграма IDEF0, IDEF1X, діаграма прецедентів (Use Case Diagram), діаграма класів (Class Diagram), діаграма послідовності дій (Sequence Diagram), діаграма станів (Statechart Diagram), діаграма активності (Activity Diagram), скріншоти роботи клієнт – серверної програми, скріншоти різних модулів системи.

6. Дата видачі завдання: 16.03.2018

Керівник роботи Міщеряков Ю.В.

(підпис)

Студент Троценко О. І.

(підпис)

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва етапів курсового проекту | Термін  виконання | Примітка |
| 1. | Аналіз предметної області | 10.03.18 – 11.03.18 | Виконано |
| 2. | Визначення основних бізнес-функцій інформаційної системи. | 12.04.18 – 12.04.18 | Виконано |
| 3. | Визначення функцій інтерфейсу клієнтської частини інформаційної системи. | 15.04.18 – 20.04.18 | Виконано |
| 4. | Розробка серверної частини інформаційної системи. | 20.04.18 – 24.04.18 | Виконано |
| 5. | Логічне та фізичне моделювання даних. | 24.04.18 – 25.04.18 | Виконано |
| 6. | Створення і заповнення баз даних. | 30.04.18 – 02.05.18 | Виконано |
| 7. | Розробка підтримки цілісності даних. | 02.05.18-10.05.18 | Виконано |
| 8. | Реалізація бізнес-функцій інформаційної системи на стороні сервера MySQL (процедур, функцій, тригерів). | 10.05.18 – 11.05.18 | Виконано |
| 9. | Розробка програмного забезпечення клієнтської частини інформаційної системи. | 25.05.18 – 27.06.18 | Виконано |
| 10. | Тестування розробленого програмного забезпечення | 28.05.18 – 29.05.18 | Виконано |
| 11. | Підготовка пояснювальної записки та її додатків. | 29.05.18-03.06.18 | Виконано |

Студент Троценко О. І.

(підпис)

Керівник проекту Мішеряков Ю. В.

(підпис)

«\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018г.

РЕФЕРАТ

Пояснювальна записка до міждисциплінарного курсового проекту: 29 с., 25 рис., 2 додатка, 12 джерел інформації.

БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, MYSQL, ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРФЕЙС ДОСТУПУ, СИСТЕМА ОБЛІКУ, ЕЛЕКТРОННА КОМЕРЦІЯ

Об'єктом досліджень міждисциплінарного курсового проекту є процес електронної комерції магазину з продажу мобільних телефонів, оформлення, облік і виконання замовлень, документування всіх облікових даних.

Предметом досліджень міждисциплінарного курсового проекту є інформаційні технології і програмні методи створення клієнтської і серверної частин інформаційної системи, що дозволяє автоматизувати облік продажів.

Мета досліджень: розробка клієнтської і серверної частин інформаційної системи інтернет-магазину з продажу мобільних телефонів.

Методи дослідження - системний підхід, методи структурного аналізу і моделювання реляційних баз даних, методи реляційної алгебри і реляційного числення, методи проектування клієнт-серверних додатків для Інтернету та Інтернет мереж, подійного об'єктно-орієнтованого програмування.

В роботі проведено аналіз предметної області, що відноситься до діяльності інтернет-магазину. Для визначення і уточнення вимог до розроблюваної інформаційної системи проведено функціональне моделювання (відповідно до стандарту IDEF0), логічне і фізичне моделювання даних (відповідно до методу IDEF1Х). Розроблено діаграми UML-моделі. Проведено проектування клієнтської і серверної частин інформаційної системи магазину з продажу мобільних телефонів. За результатами тестування проведено аналіз відповідності розробленого програмного забезпечення інформаційної системи висунутим вимогам.

Область застосування - підтримка електронної комерції магазину з продажу мобільних телефонів.

ЗМІСТ

|  |  |
| --- | --- |
| Перелік умовних позначень, символів, одиниць, скорочень і термінів ...................... | 7 |
| Вступ.................................................................................................................................... | 8 |
| 1 Аналіз предметної області ............................................................................................. | 9 |
| 1.1 Аналіз предметної області, яка визначає діяльність підприємства ................ | 9 |
| 1.2 Визначення основних бізнес-процесів які вимагають автоматизації. ........... | 9 |
| 1.3 Постановка завдання ................................................................................................ | 10 |
| 2 Розробка вимог до інформаційної системи ................................................................. | 11 |
| 2.1 Визначення функціональних вимог до інформаційної системи ..................... | 11 |
| 2.2 Логічне та фізичне моделювання бази даних інформаційної системи……… | 14 |
| 2.3 Діаграма прецедентів (Use Case Diagram) інформаційної системи ................ | 16 |
| 2.4 Діаграма класів (Class Diagram) інформаційної системи ................................. | 17 |
| 2.5 Діаграма послідовності дій (Sequence Diagram) ............................................... | 18 |
| 2.6 Діаграма активності (Activity Diagram)…………………………………………………………… | 19 |
| 2.7 Діаграма станів (Statechart Diagram)………………………………………………………………. | 20 |
| 2.8 Розробка вимог до функцій серверної частини системи .................................. | 21 |
| 2.9 Розробка вимог до функцій інтерфейсу клієнтської частини систем ............. | 21 |
| 3 Опис прийнятих проектних рішень ............................................................................... | 22 |
| 3.1 Обгрунтування вибору мови програмування ..................................................... | 22 |
| 3.2 Обгрунтування вибору СУБД .............................................................................. | 22 |
| 3.3 Створення бази даних для платформи «MySQL» ............................................... | 23 |
| 3.4 Розробка інтерфейсу клієнтської частини системи ............................................ . | 23 |
| 3.5 Тестування розробленого програмного забезпечення ....................................... | 24 |
| Висновки…………………………………………………………………………………. | 29 |
| Перелік посилань………………………………………………………………………… | 30 |
| Додаток А Текст програми ………………………………………………………………. | 31 |
| Додаток Б Посібник користувача ………………………………………………………. | 37 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

URL - uniform resource locator.

ІС - Інформаційна система

БД - База даних

СУБД - Система управління базами даних

IDEF0 - Function Modeling

IDEF1Х - Integration DEFinition for Information Modeling

UML - Unified Modeling Language

BPwin - AllFusion Process Modeler

Erwin - AllFusion ERwin Data Modeler

ПО - предметної області

ВСТУП

Інформаційна система (ІС) являє собою програмний комплекс, побудований на базі комп'ютерної техніки, призначений для зберігання, пошуку, обробки і передачі значних обсягів інформації, що має певну практичну сферу застосування.

Одним з основних компонентів інформаційної системи є база даних. База даних (БД) - це сукупність структурованої інформації про конкретні об'єкти реального світу певної предметної області, що зберігається у файлі. Доступ до цієї інформації здійснюється за допомогою інтерфейсу, а посередником виступає програмне забезпечення СУБД. Створення БД являє собою процес виробництва, який проводиться по етапах, що включають аналіз предметної області, логічне і фізичне моделювання даних.

Основою моделювання даних є визначення потреб користувачів, документування та візуалізація вимог (бізнес-вимог, вимог замовника, системних, функціональних і не функціональних вимог).

Актуальність роботи: набуття практичних навичок документування та візуалізації вимог замовника; моделювання даних і створення серверної частини ІС; проектування і розробки інтерфейсу доступу до БД (клієнтської частини ІС).

Мета міждисциплінарного курсового проекту– систематизація, засвоєння і поглиблення теоретичних знань з навчальних дисциплін, отримання практичних навичок моделювання бізнес-процесів і візуалізації вимог до програмного забезпечення; розробки БД, проектування і розробки програмного забезпечення інтерфейсу інформаційної системи.

Практичне значення: знання і навички, отримані під час виконання міждисциплінарного курсового проекту знайдуть застосування в подальшій професійній діяльності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

* 1. Аналіз предметної області, яка визначає діяльність підприємства

Предметна область інформаційної системи - це матеріальна система або система, що характеризує елементи матеріального світу, інформація про яку зберігається і обробляється. Предметна область розглядається як деяка сукупність реальних об'єктів і зв'язків між ними.

Інтернет-магазин (англ. Online shop або e-shop) - сайт, який торгує товарами в інтернеті. Дозволяє користувачам сформувати замовлення на покупку і вказати доставку в мережі Інтернет. [1]

Вибравши необхідні товари, користувач має можливість перейти у вкладку кошик і ведучи персональні дані оформити свою покупку. Сукупність відібраних товарів, та інформація про доставку являють собою закінчене замовлення, який оформляється на сайті шляхом повідомлення мінімально необхідної інформації про покупця. Інформація про покупця зберігатися в базі даних магазину і розрахована на повторні покупки.

Адміністратор інтернет-магазину має можливість додавати і редагувати відповідні товари магазину, стежити за кількістю необхідного товару на складі, відслідковувати статуси замовлень користувачів і переводити їх в різні стани.

1.2 Визначення основних бізнес-процесів, що вимагають автоматизації

В процесі розробки інформаційної системи "Магазин з продажу мобільних телефонів" потрібно автоматизація наступних бізнес вимог:

Для незареєстрованих користувачів: перегляд наявних телефонів; пошук телефонів по назві; перегляд інформації за обраним телефоном; додавання товарів в корзину; оформлення заказу; Реєстрація на сайті.

Для зареєстрованих користувачів: підбір мобільних телефонів в кошик; оформлення заказу; вхід в систему з визначенням статусу «user».

Для адміністраторів: вхід в систему з визначенням статусу «admin»; додавання телефонів; розподіл мобільних телефонів за категоріями; перегляд замовлень з фільтрацією по статусу ( «підтверджений», «не підтверджений», «у виконанні», «виконано»); зміна статусу обраного замовлення.

* 1. Постановка задачі

Необхідно провести аналіз предметної області, що відноситься до діяльності інтернет магазину. Визначити і уточнити вимоги до розроблюваної інформаційної системи за допомогою функціонального моделювання (відповідно до стандарту IDEF0), логічного і фізичного моделювання даних (відповідно до стандарту IDEF1Х). Розробити діаграми UML-моделі. Провести тестування і проаналізувати відповідності розробленого програмного забезпечення інформаційної системи висунутим вимогам.

Розробити серверну і клієнтську частини інформаційної системи. Серверна частина повинна забезпечувати взаємодію з базою даних, розроблену для платформи СУБД MySQL. Клієнтська частина повинна забезпечувати інтерфейс з користувачем. Функціональність бізнес процесів реалізується на сервері.

Реалізувати функціонал в першій версії: Перегляд наявних мобільних телефонів, пошук телефонів по назві, перегляд інформації за обраним телефоном, додавання товарів в корзину, оформлення замовлення (для цього достатньо заповнити контактну інформацію: телефон та / або e-mail, адреса), перегляд замовлень з фільтрацією по статусу (новий, в виконанні, виконаний ...), зміна статусу обраного замовлення, вхід (тільки для адміністратора / менеджера).

1. РОЗРОБКА ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ
   1. Визначення функціональних вимог до інформаційної системи

Функціональні вимоги визначають функціональність ПЗ, яку розробники повинні побудувати, щоб користувачі могли виконати свої завдання в рамках бізнес-вимог.

Для проектування функціональної моделі To Be найбільш підходять два середовища функціонального моделювання - це Ramus і AllFusion Process Modeler 7 (BPwin). Ці середовища є одними з найвідоміших і легко освоюваних середовищ даного напрямку. Ramus і BPwin дуже схожі, і їх можна розглядати як альтернативи один одного. BPwin є більш універсальним і простим середовищем на відміну від Ramus. Для функціонального проектування обрано середовище AllFusion Process Modeler 7 (BPwin). [3]

Модель описує основну функцію інтернет магазину - функцію продажу телефонів через інтернет. Контекстна діаграма функціональної моделі зображена на рисунку 2.1.

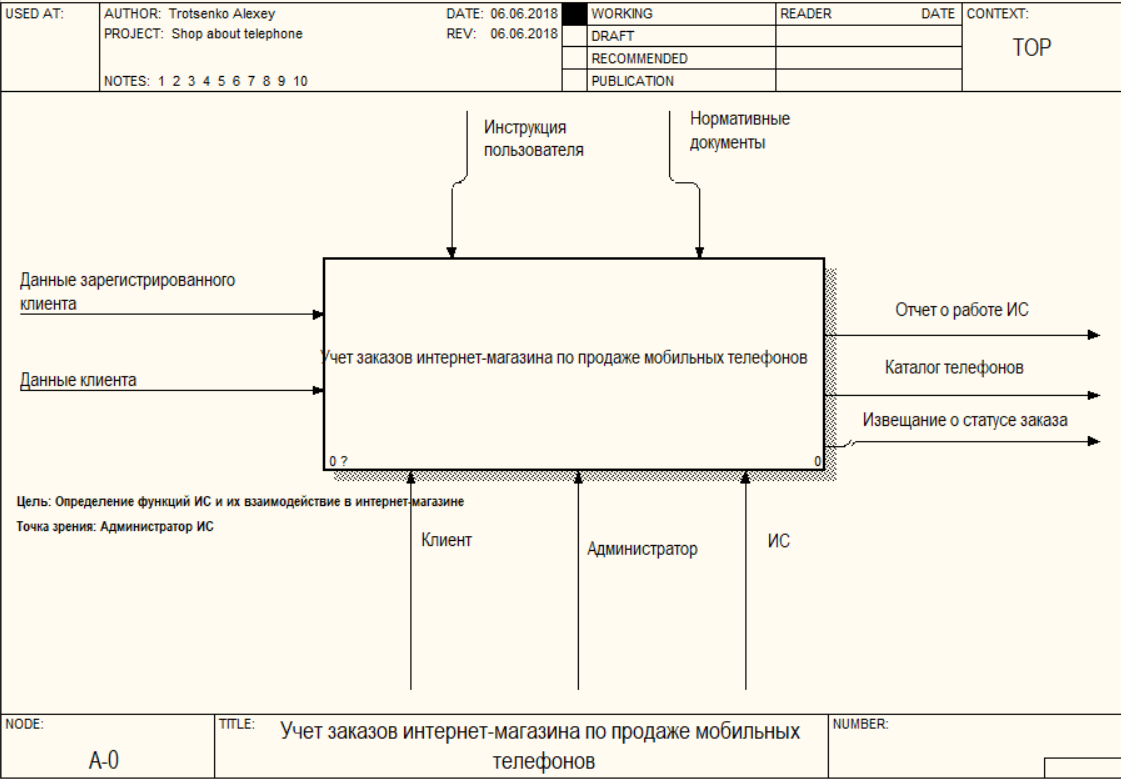


Рисунок 2.1 – Контекстна діаграма функціональної моделі

Другий рівень декомпозицій зображений на рисунку 2.2 розбиває контексну діаграму на чотори блоки які явно відображають, що каталог товарів породжується всередині системи виходячи з Адміністрювання і заходячи до інших блоків як управління.

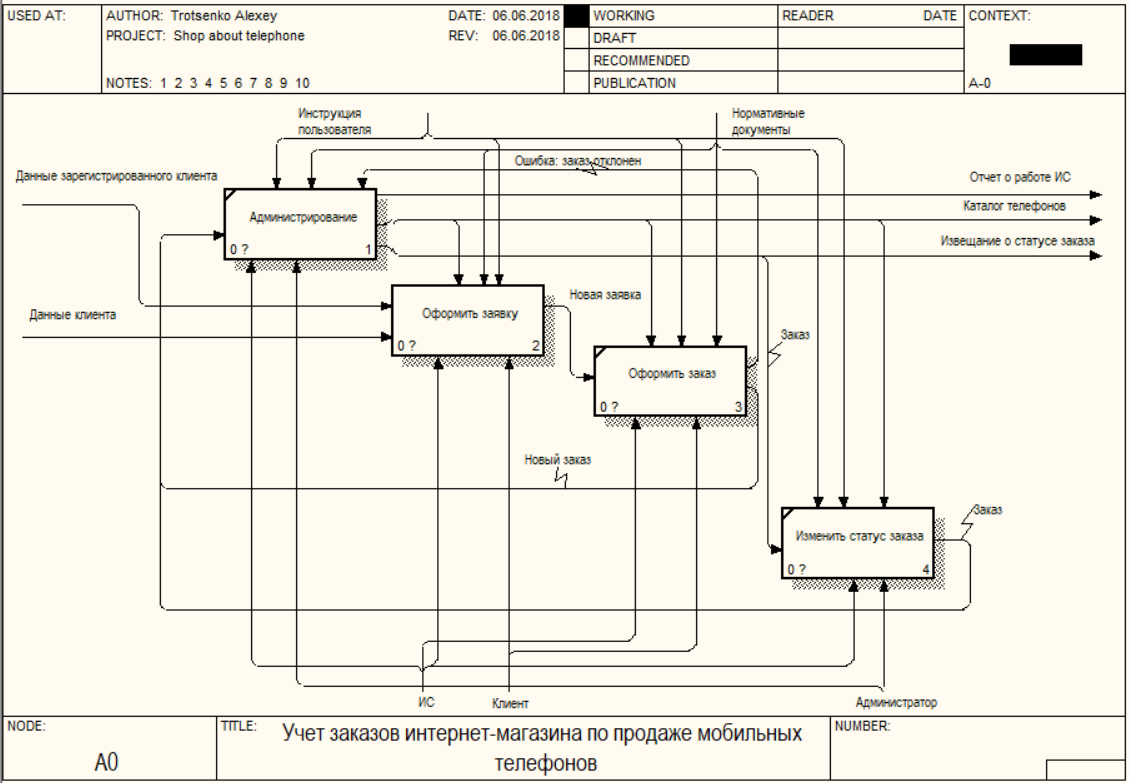


Рисунок 2.2 – Функціональна модель на другому рівні декомпозиції

Основними процесами оформлення заявки, зображеного на рисунку 2.3, є реєстрація, авторизація в системі та оформлення заявки.

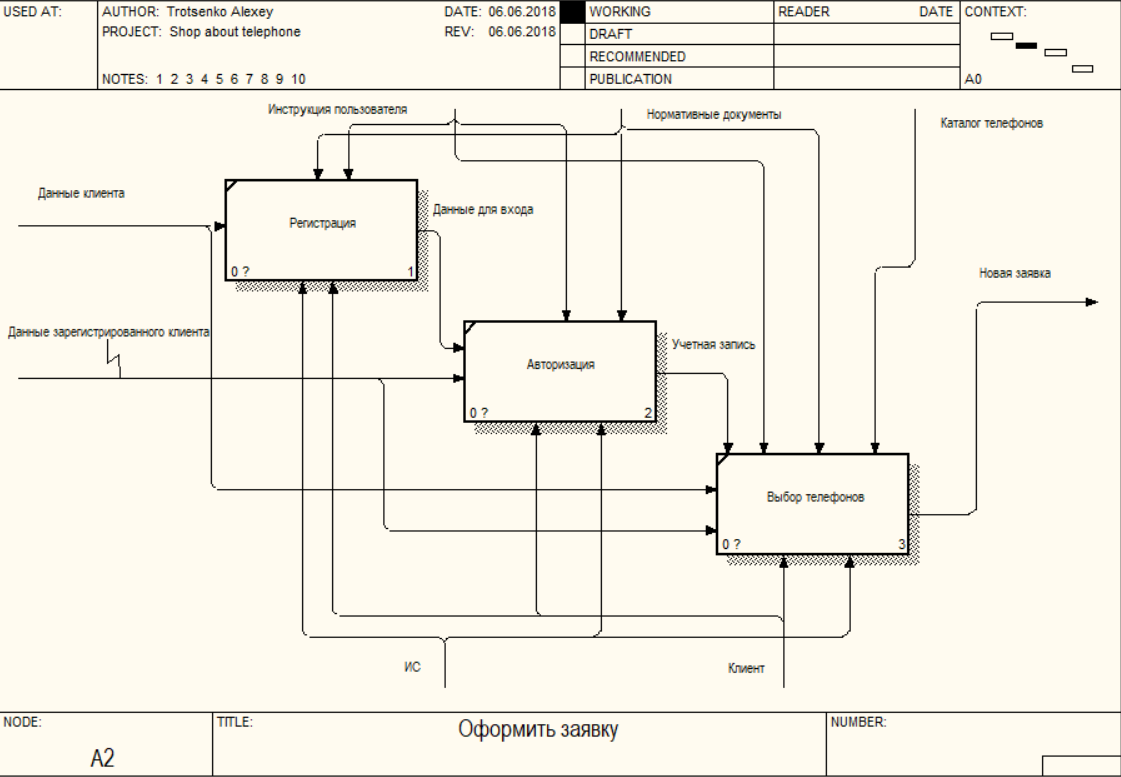


Рисунок 2.3 – Декомпозиція процесу «Оформлення заявки»

На виході всієї системи ми отримуємо повідомлення про статус замовлення, каталог телефонів, звіт про работу ІС.

* 1. Логічне та фізичне моделювання бази даних інформаційної системи

Традиційно процедуру проектування бази даних розбивають на три етапи, кожен з яких завершується створенням відповідної інформаційної моделі [1, 21].

Етап 1-й. Концептуальне проектування - створення схеми БД, що включає визначення найважливіших сутностей (таблиць) і зв'язків між ними, але не залежить від моделі БД (ієрархічної, мережевої, реляційної і т. Д.) та фізичної реалізації (цільової СУБД).

Етап 2-й. Логічне проектування - розвиток концептуальної схеми БД з урахуванням прийнятої моделі (ієрархічної, мережевої, реляційної і т.д.).

Етап 3-й. Фізичне проектування - розвиток логічного схеми БД з урахуванням обраної цільової СУБД.[4]

Концептуальне та логічне проектування разом називають також инфологической або семантичним проектуванням.

При використанні будь-якого CASE-засобу спочатку будується логічна схема БД у вигляді діаграми із зазначенням сутностей і зв'язків між ними.

Логічною схемою називається універсальне опис структури даних, незалежне від кінцевої реалізації бази даних і апаратної платформи. На підставі отриманої логічної схеми переходять до фізичної схемою даних. Фізична схема являє собою діаграму, що містить всю необхідну інформацію для генерації БД для конкретної СУБД або навіть конкретної версії СУБД. Якщо в логічній схемі не має значення, які ідентифікатори носять таблиці і атрибути, тип даних атрибутів і т. д., то у фізичній схемі повиннен бути повний опис БД відповідно до прийнятого в ній синтаксисом, із зазначенням типів атрибутів, тригерів, збережених процедур і т.д. За однією і тією ж логічною схемою можна створити кілька фізичних. [5]

Виходячи з предметної області інтернет-магазину, можна виділити 6 основних сутності:

Користувач [User] - представляє інформацію про кожного користувача.

Телефони [Telephone] - представляє інформацію про телефони, наявних.

Замовлення [Order] - представляє інформацію про всі замовлення користувачів.

Доставка (delivery) - являє інформацію про доставку

Категорії (сategory) - являє інформацію про категоріях

Виробник (Manufacture) - являє інформацію про виробника

Для кожної сутності визначені атрибути.

Атрибут role - показує якими правами володіє користувач (Адміністратор, звичайний покупець)

Користувач може обрати кілька товарів для замовлення і звортне - один товар може бути обраний декільками користувачами. Тому зв'язок від користувача до товару - М: М (багато до багатьох).

Між сутностями телефон, замовлення утворюється зв'язок багато до багатьох, в результаті чого ми реалізуємо її через додаткову таблицю. Зв'язок між категорією, телефоном і виробником, телефоном є один до багатьох, зв'язок між користувачем і доставкою є один до багатьох, так само як і доставка та замовлення.

ER-модель, складена по ER-діаграмі, зображена на рисунку 2.4.

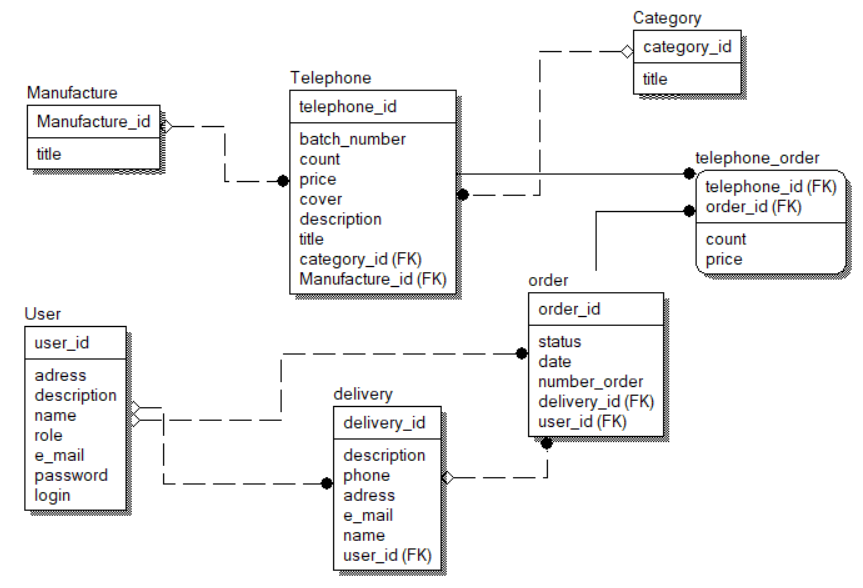


Рисунок 2.4 – Логічна модель даних

Таким чином, логічна модель даних складається з 6 сутностей.

З логічної моделі даних засобами середовища згенерована фізична модель зображеная на рисунку 2.5. Зараз клієнт, телефон, категорія, замовлення, доставка і виробник є таблицями, а не сутностями.

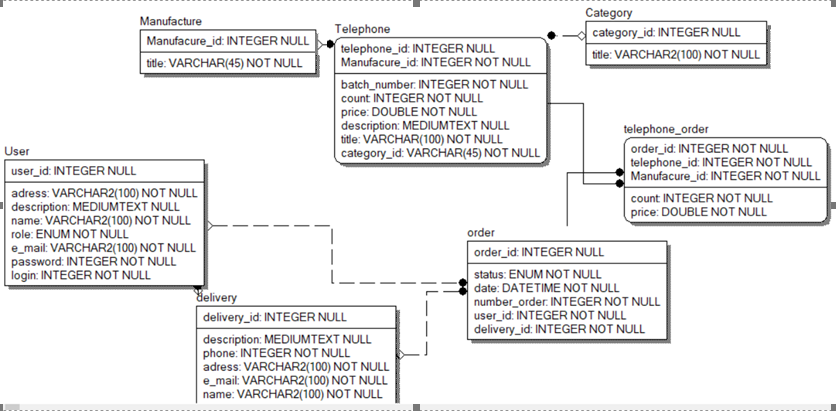


Рисунок 2.5 – Фізична модель даних

Всі атрибути сутностей стали полями таблиці. Кожне з полів має певний тип.

* 1. Діаграма прецедентів (Use Case Diagram) інформаційної системи

Варіанти використання призначені в першу чергу для визначення функціональних вимог до системи і керують усім процесом розробки. Всі основні види діяльності такі як аналіз, проектування, тестування виконуються на основі варіантів використання. На рисунку 2.6 представлена діаграма прецедентів, що відображає як результати, які хоче отримати користувач впливають на архітектуру системи і як повинні поводитися компоненти системи, для того щоб реалізувати потрібну для користувача функціональність.

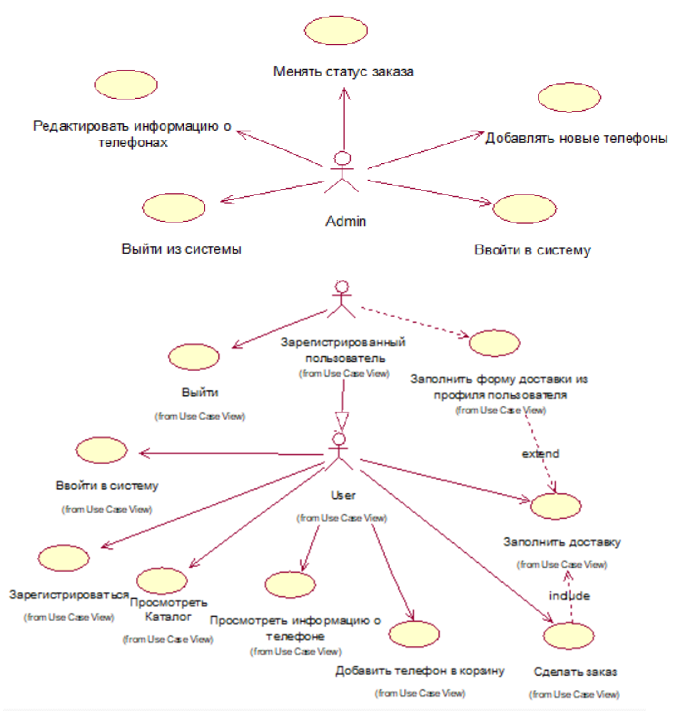


Рисунок 2.6 – Діаграма прецедентів (Use Case Diagram)

* 1. Діаграма класів (Class Diagram) інформаційної системи

Діаграма класів використовуються при моделюванні ПС найбільш часто. Вони є однією з форм статичного опису системи з точки зору її проектування, показуючи її структуру. Діаграма класів не відображає динамічну поведінку об'єктів зображених на ній класів.

Мова моделювання UML має набір відносин для побудови моделі класів, але навіть такий розвиненою ООП мова, як Java має тільки дві явні конструкції для відображення зв'язків: extends (розширення) і interface / implements (реалізація). З цієї позиції і построіна діаграма класів яка зображена на рисунку 2.7.

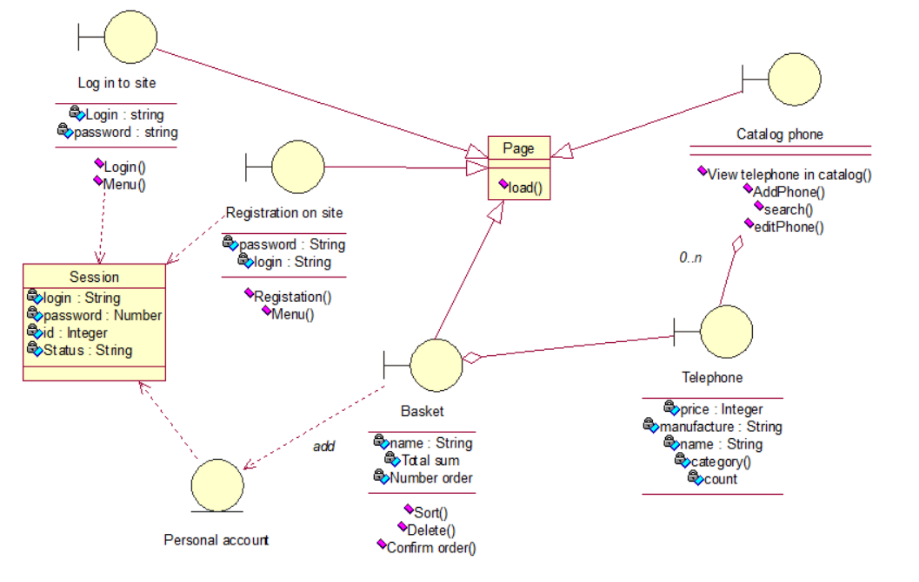


Рисунок 2.7 – Взаємозв'язок класів яка забезпечує взаємодію з користувачем

* 1. Діаграма послідовності дій

Діаграми послідовності (sequence diagram) є видом діаграм взаємодії мови UML, які описують відносини об'єктів в різних умовах. Умови взаємодії задаються сценарієм, отриманим на етапі розробки діаграм варіантів використання [8]. Спроектовані дві діаграми послідовності дій які відображають основний бізнес процес, а саме на рисунку 2.8 зображена діаграма для формування заявки та на рисунку 2.9 зображена діаграма для зміни статусу замовлення.

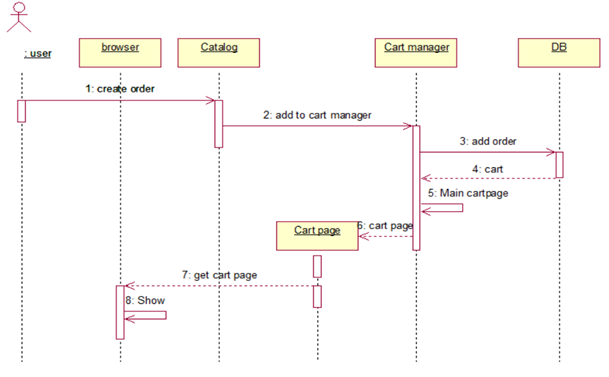
****

Рисунок 2.8 – Діаграма послідовності для формування заявки

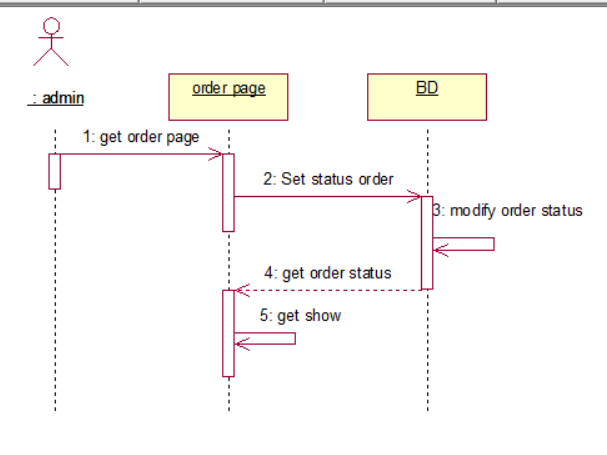


Рисунок 2.9 – Діаграма послідовності для зміни статусу замовлення

* 1. Діаграма активності

Діаграма діяльності (Activity diagram) показує потік переходів від однієї діяльності до іншої. Діяльність (Activity) - це триває в часі неатомарний крок обчислень в автоматі. Діяльності, в кінцевому рахунку, приводять до виконання якоїсь дії (Action), складеного з виконуваних атомарних обчислень, кожне з яких або змінює стан системи, або повертає якесь значення. Графічно діаграма діяльності представляється у вигляді графа, що має вершини і ребра.

На рисунку 2.10 зображена спроектована діаграма діяльності, що реалізують процес продажу телефонів.

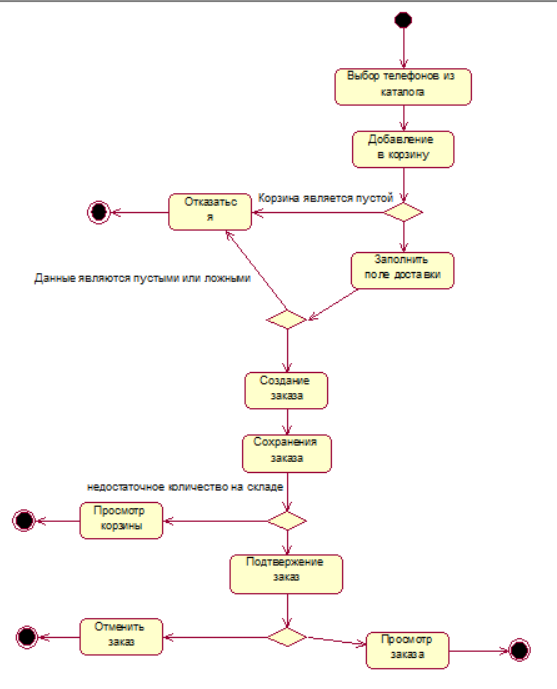


Рисунок 2.10 – Діаграма діяльності для продажу

* 1. Діаграма стану

Діаграми станів є добре відомим засобом описання поведінки систем. Вони визначають всі можливі стани, в котрих може перебувати конкретний об'єкт, а також процес зміни станів об'єкта в результаті впливу деяких подій.[6]

На рисунку 2.11 зображена діаграма станів для замовлення.

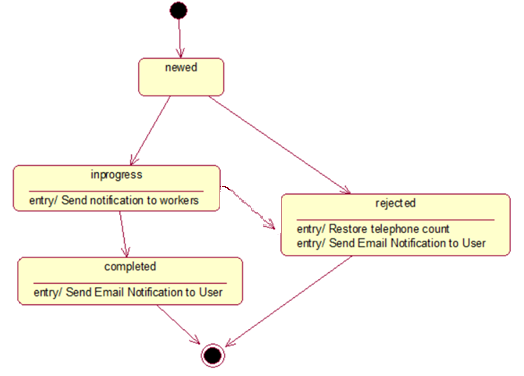


Рисунок 2.11 – Діаграма станів для замовлення

Важливо те що за умовою предметної області замовлення може бути відхилене адміністратором під час виконання.

* 1. Розробка вимог до функцій серверної частини системи

Серверна частина забезпечує взаємодію з базою даних, розроблену для платформи СУБД MySQL. Поступив запит обробляється сервером додатків, який в свою чергу зв'язується з базою даних. СУБД - здійснює зберігання і обробку даних про товари, клієнтів, рахунках і т.д. Підключення до баз даних реалізується за допомогою інтерфейсу JDBC DriverManager Interface.

При проектуванні інформаційної системи виявляються деякі прошарки, які відповідають за взаємодію різних модулів системи. З'єднання з базою даних є однією з найважливішою складовою програми. Завжди виділяється частина коду, модуль, відповідающий за передачу запитів в БД і обробку отриманих від неї відповідей. У загальному випадку, визначення Data Access Object описує його як прошарок між БД і системою. DAO абстрагує сутності системи і робить їх відображення на БД, визначає загальні методи використання з'єднання, його отримання, закриття та (або) повернення в Connection Pool.[7]

Структура комплексу управління Інтернет-магазином реалізується у вигляді архітектури клієнт / сервер. Така архітектура розбиває процес обробки даних між [14]:

1. клієнтом;
2. сервером доданків;
3. базою даних.
   1. Розробка вимог до функцій інтерфейсу клієнтської частини системи

Клієнтська частина забезпечує інтерфейс з користувачем.

При розробки інтерфейсу інтернет-магазину з продажу мобільних телефонів використати мову програмування java та вільний набір інструментів Bootstrap для створення сайтів і веб-додатків. Bootstrap включає в себе HTML і CSS-шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення.

**До основних переваг Bootstrap можна віднести:**

a) скорочення часу, необхідного для створення макета сайту. Фреймворк пропонує достатню кількість шаблонів і готових рішень;

b) кросбраузерність і адаптивність. Сайти, створені за допомогою Bootstrap, ідентично відображаються на різних пристроях і в сучасних браузерах;

c) простота використання. Для роботи з фреймворком будуть потрібні елементарні навички верстки;

d) власні стилі і шрифти.

Так само використовується кроссплатформенная бібліотека JQuery. Бібліотека спрощує розробку веб-додатків і має величезних кількістю додаткових модулів (плагінів).

1. ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ
   1. Обгрунтування вибору мови програмування

В Java взаємодія з базою даних передбачено за допомогою стандарту JDBC (Java Database Connectivity). Цей стандарт багато в чому спрощує роботу з SQL базами даних. Для розробки інтернет-магазину була вибрана мова Java, тому що ця мова є кросплатформленою і найпопулярнішою мовою в світі в наслідок чого можна досить легко знайти необхідний матеріал по застосуванню різних конструкцій мови.

Найбільш популярними середовищами програмування на мові Java є Eclipse SDK і NetBeans IDE. NetBeans IDE заснована на Eclipse, але вона має розширений і багато в чому прегруженний функціонал. Для розробки сайту обрано середовище Eclipse EE, через простоту і зручність.

* 1. Обгрунтування вибору СУБД

Для створення і супроводу баз даних системи найбільш популярні дві СУБД: MySQL і Microsoft SQL Server. Обидві системи працюють з базами даних на основі мови SQL. Обидві системи приблизно в рівній мірі підтримують захист інформації. Microsoft SQL Server більша система з великим числом можливостей по роботі з базами даних. MySQL простіша і має менший функціонал. Так як інформаційна система інтернет-магазину з продажу телефонів буде виконувати більшу частину роботи з базою даних, то великий функціонал не потрібно. В цьому плані СУБД MySQL більше підходить для розроблюваної системи. Основними перевагами баз даних MySQL є: многопоточность, підтримка декількох одночасних запитів, гнучка система привілеїв і паролів, легкість управління таблицею, включаючи додавання та видалення. Також MySQL - найпопулярніша і поширена СУБД і є найбільш пристосованою для застосування в середовищі web.[8]

* 1. Створення бази даних для платформи «MySQL»

Для розробки бази даних для інтернет-магазину потрібно було створити таблиці: telephone, order, manufacture, category, delivery і user. Дані в таблиці category і telephone заносяться тільки адміністратором. Таблиця order містить перелік товарів, замовлених користувачем, а також інформацію про користувача (прізвище, ім'я, номер телефону, адреса і т.д.). Дані в таблицю order додаються користувачем як і при реєстрації так і без неї. Таблиця user містить дані про всі зареєстровані користувачів системи. Щоб створити нову базу даних використовується команда CREATE DATABASE. Синтаксис: CREATE DATABASE dbname; mysql> CREATE DATABASE category; Тут category є ім'ям створюваної бази даних. Перш ніж працювати з базою даних, необхідно встановити з нею з'єднання з мережею, а також провести авторизацію користувача. Для цього служить функція mysql\_connect (). Функція mysql\_connect () встановлює мережеве з'єднання з базою даних. До того як послати перший запит серверу MySQL, необхідно вказати, з якою базою даних ми збираємося працювати. Для цього і призначена описувана функція mysql\_select\_db. \_select\_db ( «category») or die ( «Немає з'єднання з базою даних»); Вона повідомляє, що в подальших операціях використовуватиметься база даних category. В іншому випадку буде виведено повідомлення про помилку «Немає з'єднання з базою даних». Тепер необхідно формувати і посилати запити до бази даних. Для цього існує функція - mysql\_query (). Вона повертає ідентифікатор результуючого набору даних. mysql\_query (string $ query [, int $ link\_identifier]) Ця функція в своєму роді універсальна: вона посилає MySQL-серверу запит $ query і повертає ідентифікатор відповіді, або результат. Параметр $ query є рядком, складений за правилами мови SQL.

* 1. Розробка інтерфейсу клієнтської частини системи

Інтернет-магазин представляє із себе каталог мобільних телефонів з можливістю оформленням замовлення в мережі інтернет.

Вибравши необхідні товари, користувач має можливість перейти у вкладку кошик, вказавши дані про доставку оформити свою покупку. Сукупність відібраних товарів і доставки являють собою закінчене замовлення, яке оформляється на сайті шляхом повідомлення мінімально необхідної інформації про покупця.

Адміністратор інтернет-магазину має можливість додавати, видаляти і редагувати відповідні товари магазину, стежити за кількістю необхідного товару на складі, відслідковувати статуси замовлень користувачів і переводити їх в різні стани.

Дизайн сайту реалізований за допомогою вільних інструментів BootStrap, для елементів управління сайтом використовується іконки Glyphicons halflings.

Стартова сторінка сайту являє собою список телефонів оформлений у вигляді таблиці з можливістю переходу на вкладку кошик, форму входу в систему, форму реєстрації на ресурсі. Сайт реалізує всі заявлені вимоги до системи. Приклад сторінки каталогу зображений на рисунку 3.1

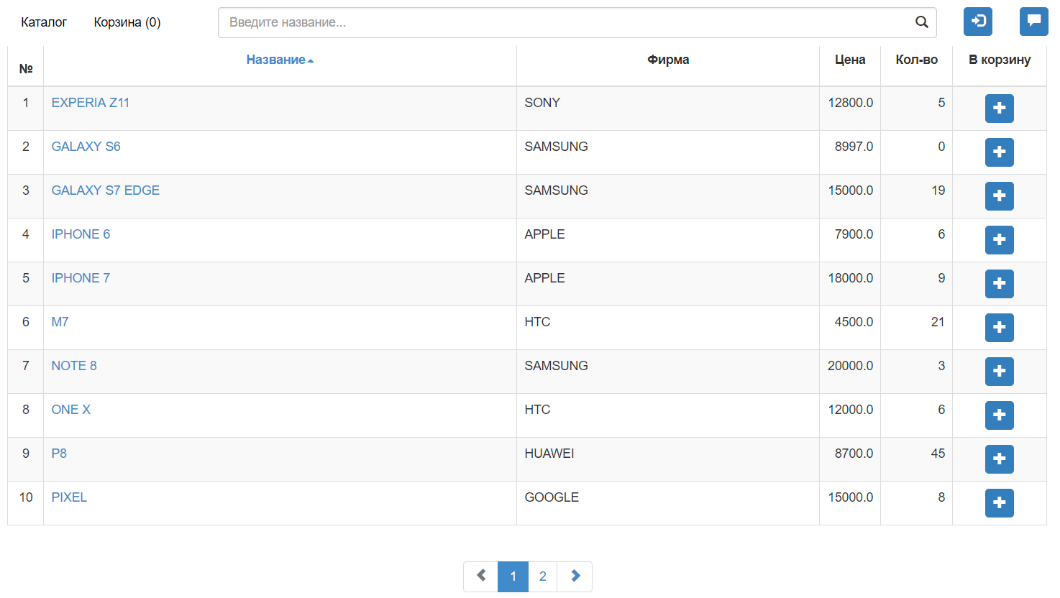


Рисунок 3.1 – Сторінка каталогу

* 1. Тестування розробленого програмного забезпечення

На головній сторінці інтернет-магазину показаному на рисунку 3.2 користувач бачить перед собою каталог товарів і елементи управління з ним.

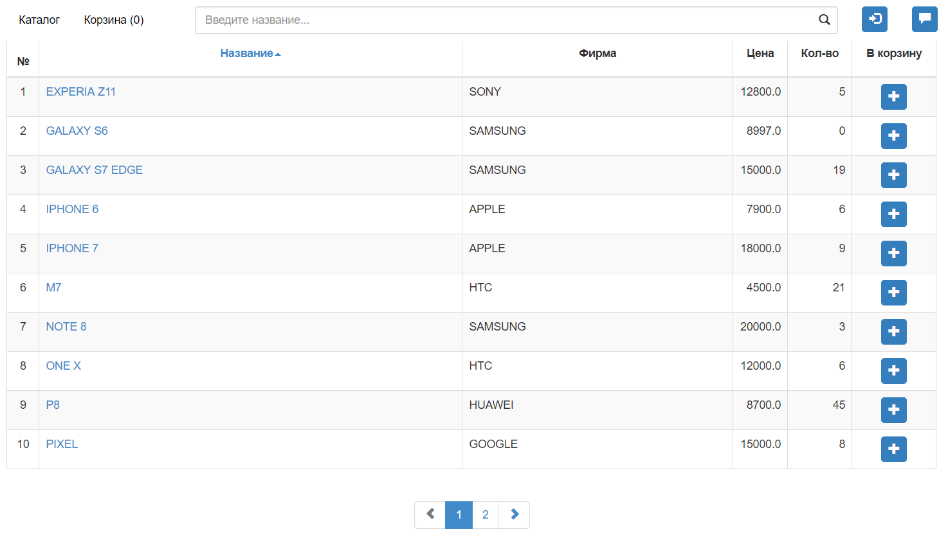
****

Рисунок 3.2 – Головна сторінка з каталогом товарів

Для знаходження необхідного телефон користувачеві необхідно скористатися рядком пошуку, відображених на рисунку 3.3, розташованай у верхній частині сайту після чого сайт відобразить всі відповідні результати.

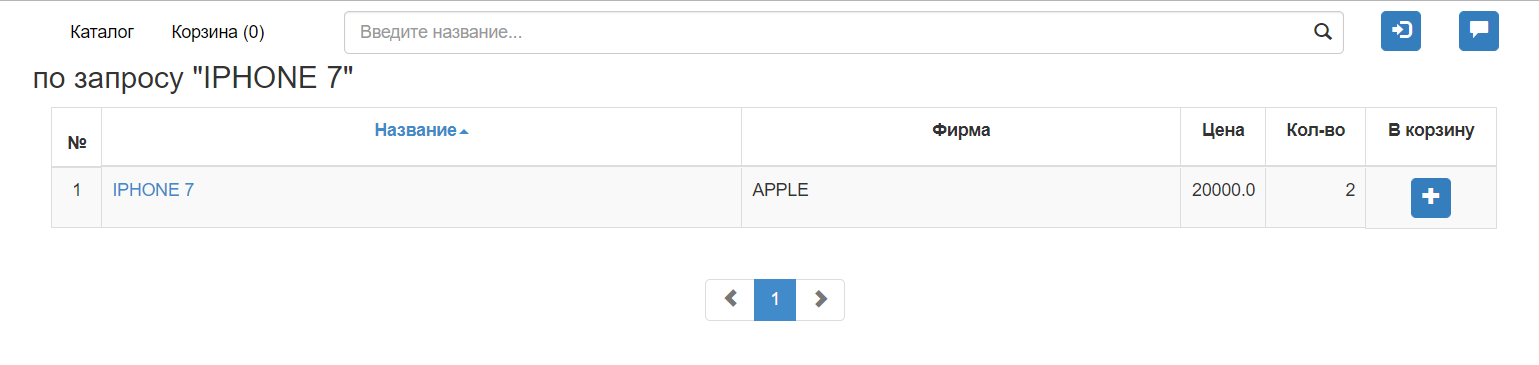


Рисунок 3.3 – Результати пошуку

Для перегляду інформації по обраному телефону досить натиснути на назву моделі телефону. Приклад отриманого результату на рисунку 3.4

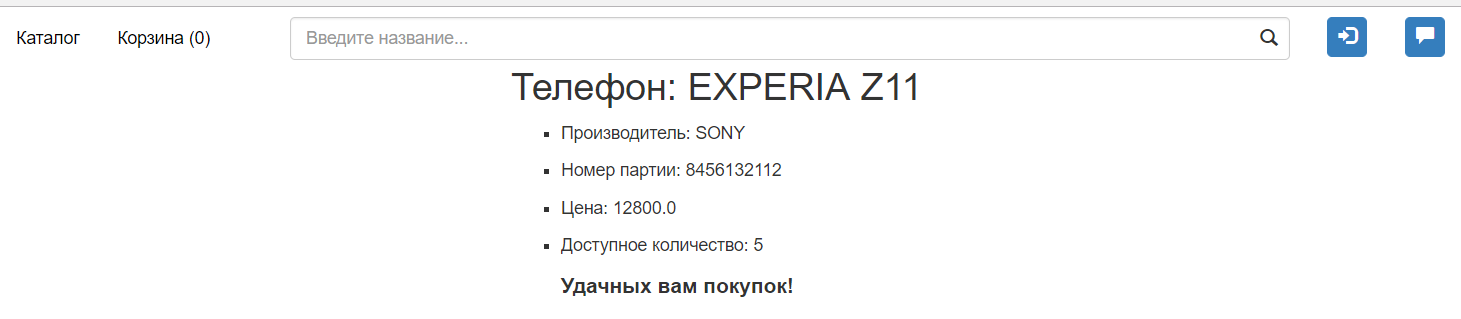


Рисунок 3.4 – Перегляд інформації про телефон

Користувач може пройти необов'язкову форму реєстрації, відображену на рисунку 3.5 в результаті чого інформація про нового користувача буде додана в базу даних.

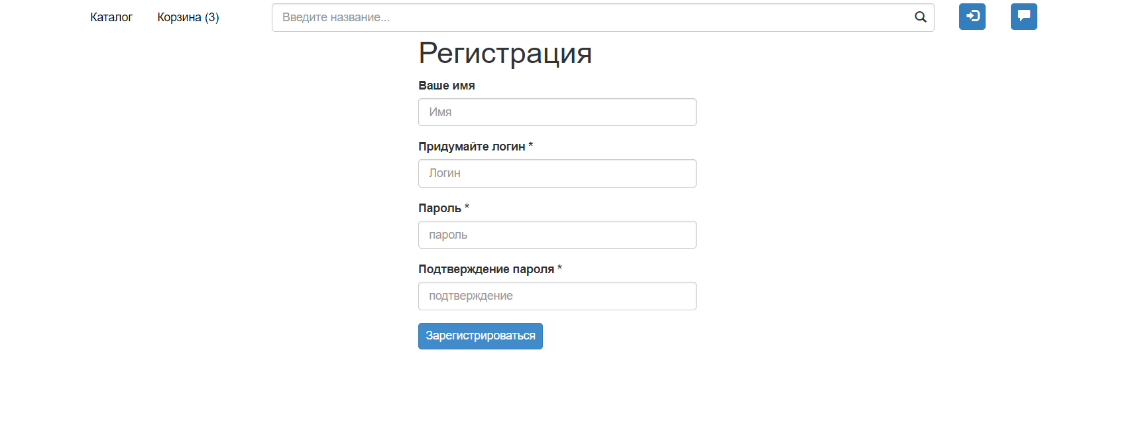


Рисунок 3.5 – Форма регістрації

Зареєстрований користувач може перейти в форму входу в систему, показаного на рисунку 3.6, в результаті чого буде визначено статус зареєстрованого користувача.

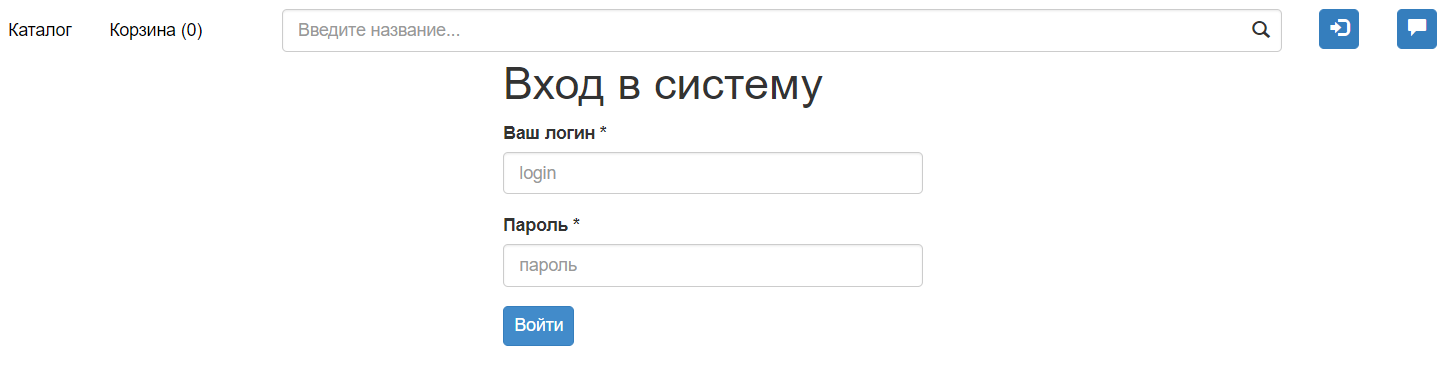


Рисунок 3.6 – Форма входу в систему

Статус зареєстрованого користувача може бути двох видів:

a) User

b) Admin

Вхід з систему під статусом користувач дозволяє ідентифікувати його замовлення, під статусом адміністратор дозволяє отримати доступ до таких елементів сайту як: редагування товару, додавання нового товару показаного на рисунку 3.10, підтверджувати або скасовувати замовлення, працювати з станами замовлення, переглядати інформацію про покупців відображений на рисунку 3.8.

Відображення замовлень і елементів роботи з ними під статусом адміністратор показана на рисунку 3.7 і на рисунку 3.9 відповідно.

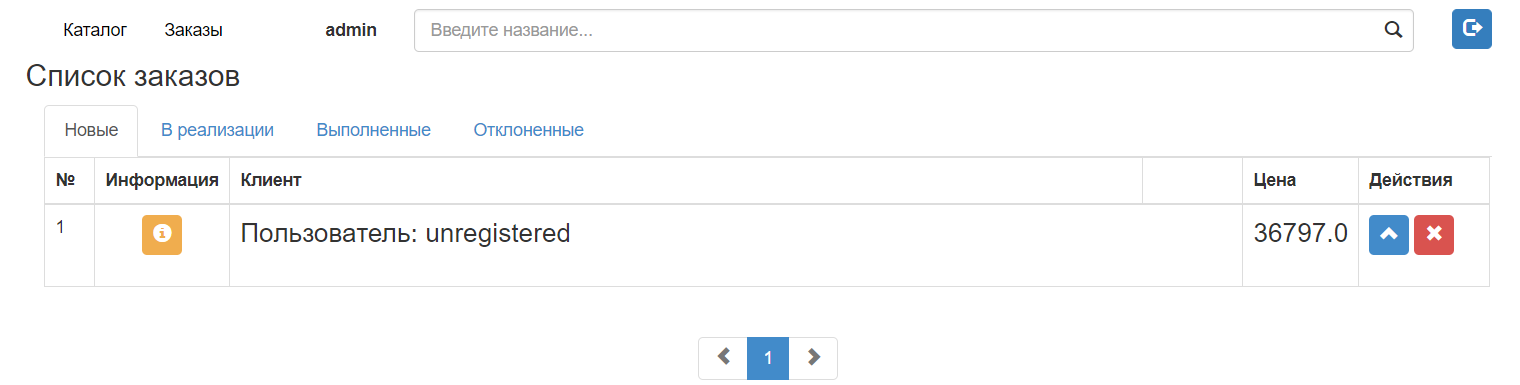


Рисунок 3.7 – Відображення замовлень і елементів роботи з ними

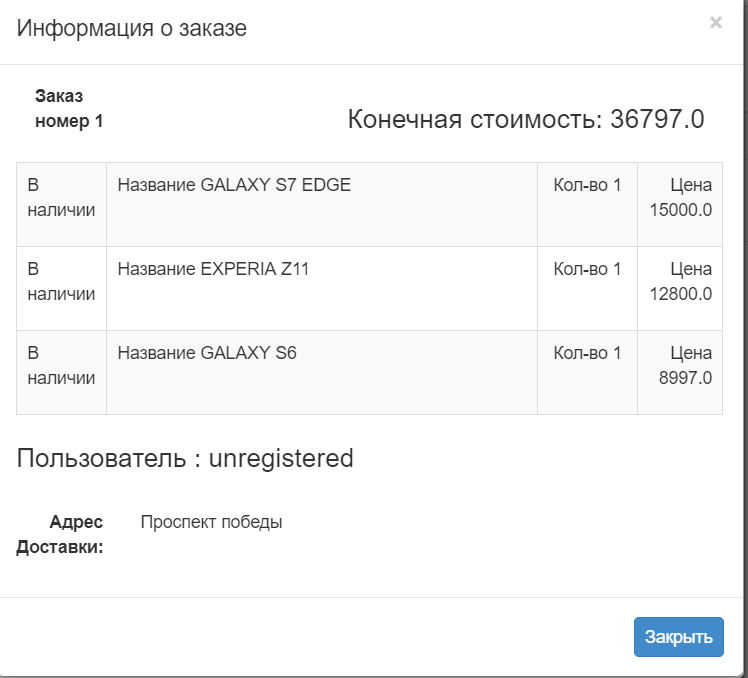


Рисунок 3.8 – Відображення інформації про покупця

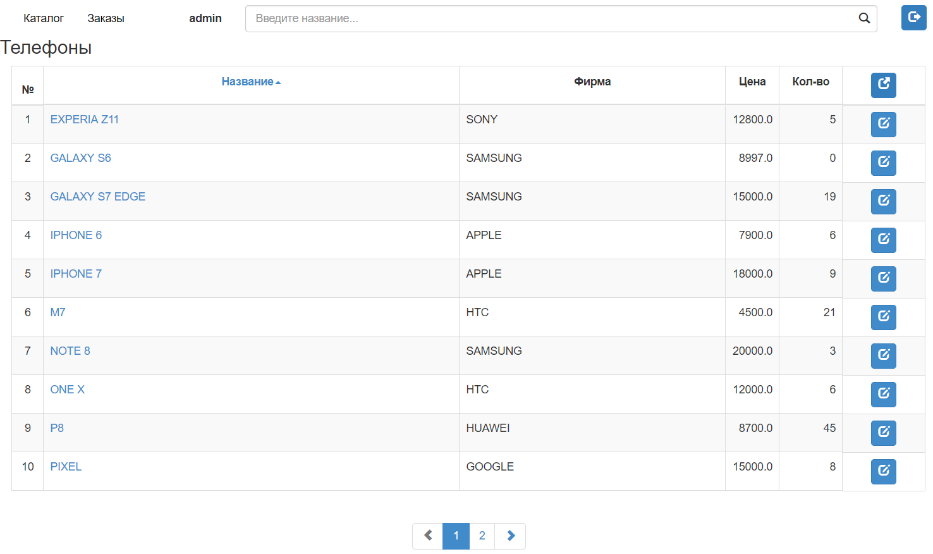


Рисунок 3.9 – Відображення елементів під статусом Admin

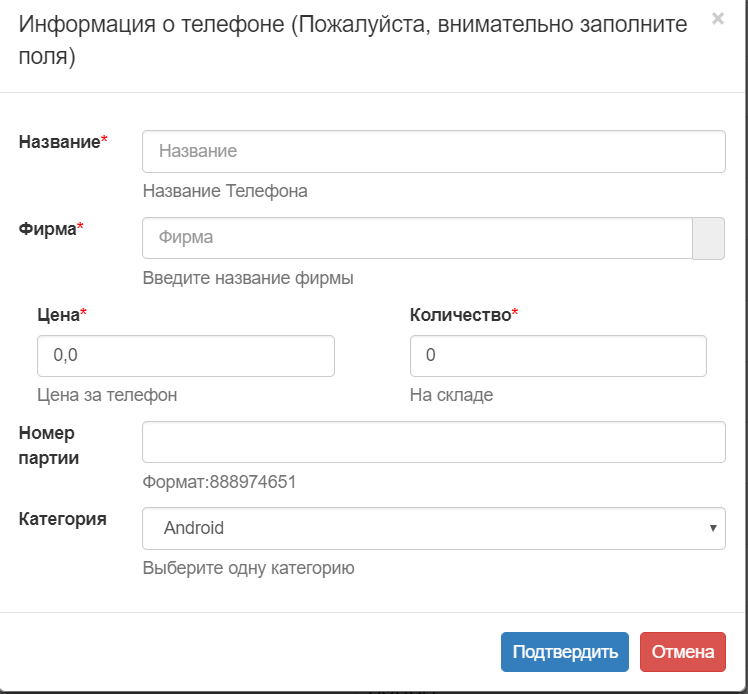


Рисунок 3.10 – Вікно додавання нового товару

Як і зареєстрований так і незареєстрований користувач може оформити замовлення.

У випадки вибору потрібних товарів користувач переходить у вкладку кошик зображеного на рисунку 3.11, де він може оформити відповідне замовлення.

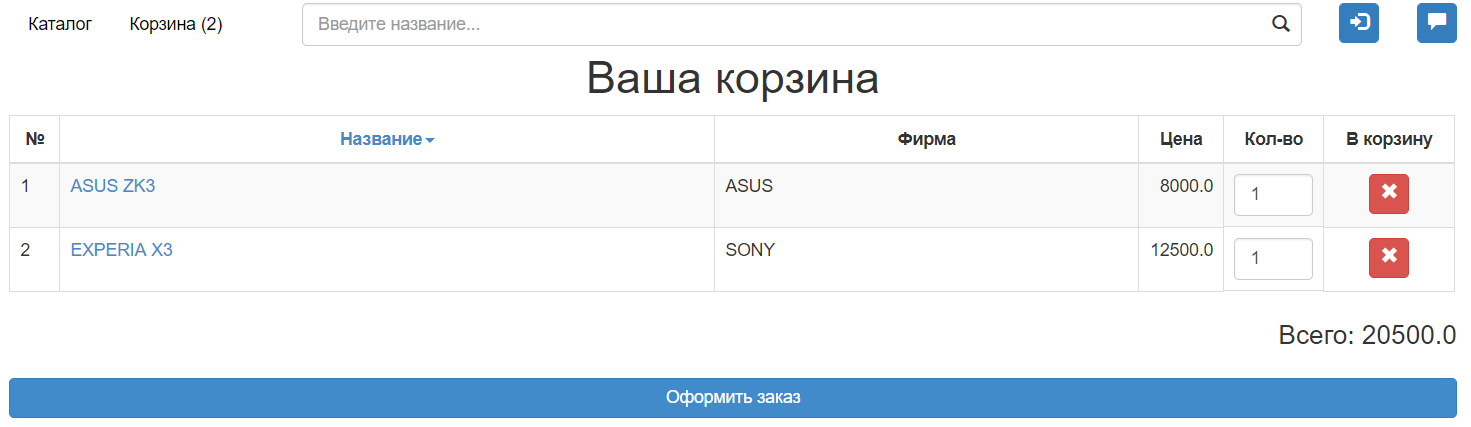


Рисунок 3.11 – Кошик

При додаванні в кошик товарів, які відсутні на складі поле даного товару позначається червоним, приклад показано на рисунку 3.12, і в даному випадку оформлення замовлення неможливо.

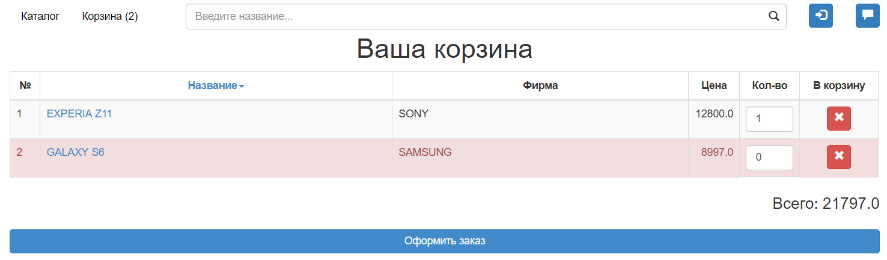


Рисунок 3.12 – Повідомлення відсутності товару на складі

Для завершення оформлення замовлення користувачеві необхідно повідомити мінімальну кількість інформації про себе і підтвердити замовлення, приклад форми показан на рисунку 3.13.

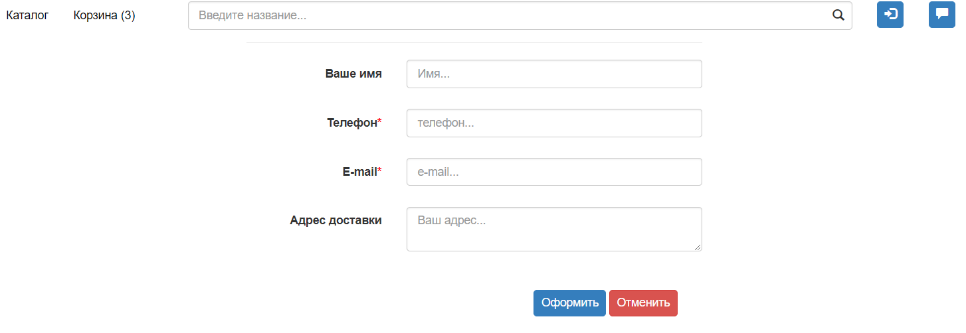
****

Рисунок 3.13 – Форма доставки

На останньому етапі користувачеві відображається сторінка інформації про оформленому замовленні, приклад інформації про замовлення відображений на рисунку 3.14

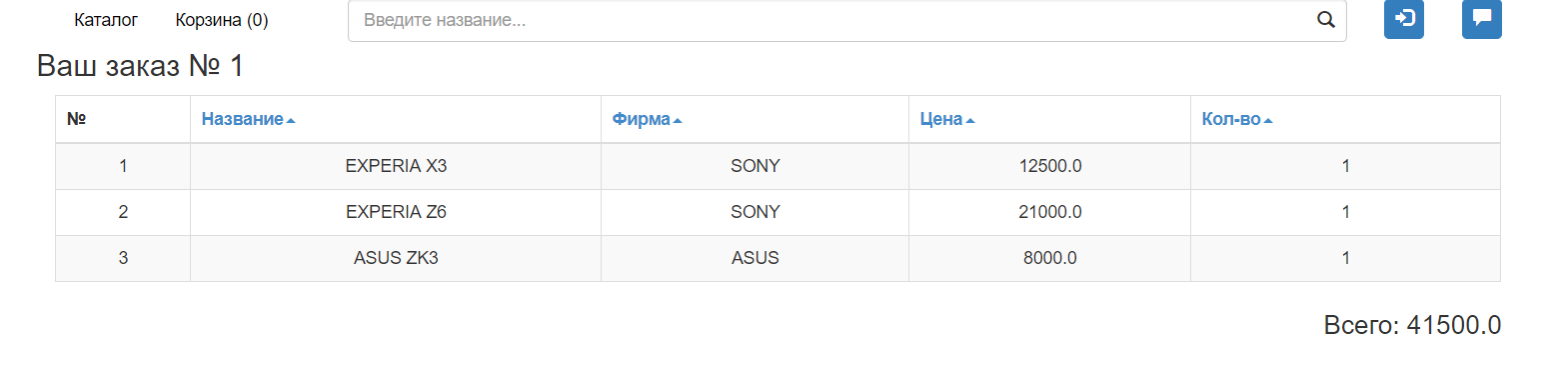


Рисунок 3.14 – Інформація про замовлення

В результаті тестування було виявлено що інтерфейс по роботі з інтернет-магазином відповідає висунутим вимогам.

ВИСНОВКИ

В процесі виконання міждисциплінарного курсового проекту розглядався процес електронної комерції магазину з продажу мобільних телефонів, оформлення, облік і виконання замовлень, документування всіх облікових даних. Інформаційні технології та програмні методи створення клієнтської і серверної частин інформаційної системи, дозволили автоматизувати облік продажів. Серверна частина забезпечує взаємодію з базою даних, розроблену для платформи СУБД MySQL. Клієнтська частина забезпечує інтерфейс з користувачем. В роботі проведено аналіз предметної області, що відноситься до діяльності інтернет магазину. Для визначення і уточнення вимог до розроблюваної інформаційної системи проведено функціональне моделювання (відповідно до стандарту IDEF0), логічне і фізичне моделювання даних (відповідно до стандарту IDEF1Х). Розроблено діаграми UML-моделі. Проведено проектування клієнтської і серверної частин інформаційної системи магазину з продажу мобільних телефонів. За результатами тестування проведено аналіз відповідності розробленого програмного забезпечення інформаційної системи висунутим вимогам.

Для побудови реляційної моделі бази даних була використана СУБД MySQL. Відмінною рисою даної СУБД є простота її використання в поєднанні з широкими можливостями по розробці додатків.

Отримана модель предметної області дозволяє відобразити всі аспекти функціонування електронного магазину.

ПЕРЕЧЕНЬ ССЫЛОК

1. ДСТУ 3008-2015. Документація. Звіти в сфері науки і техніки. Структура і правила оформлювання. Чинний від 2017-07-01. – Київ : ДП «УкрНДНЦ», 2016. – 26 с.

2. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 2016–07–01. – Київ : ДП «УкрНДНЦ», 2016. – 16 с.

3. Маклаков, С.В. BPwin и ERwin [Текст] : CASE-средства разработки информационных систем / С.В. Маклаков – М.: Диалог-МИФИ, 1999. - 256 с.

4. Роб, П., Коронел, К. Системы баз данных [Текст] : Проектирование, реализация и управление : Пер. с англ. / П. Роб, К. Коронел - Спб.: БХВ-Петербург, 2004. – 1040 с.

5. Коннолли, Т., Бегг, К., Страчан, А. Базы данных. Проектирование, реализация и сопровождение [Текст] : Теория и практика : Пер. с англ. / Т. Коннолли, К. Бегг, А. Страчан – М.: Издательский дом «Вильямс», 2003. – 1436 с.

6. Вигерс, К. Разработка требований к программному обеспечению [Текст] : пер. с англ. / К. Вигерс – М.: Издательско-торговый дом «Русская Редакция», 2004. – 576 с.

7. Дейт, К. Введение в системы баз даннях [Текст] : пер. с англ. / К. Дейт – М.: Издательский дом «Вильямс», 2001. – 1072 с.

8. Пасічник, В.В. Організація баз даних та знань [Текст] / В.В. Пасічник, В.А. Резніченко. – К.: Видавнича група BHV, 2006. – 384 с.

9. Мацяшек, Л. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML [Текст] : пер. с англ. / Л. Мацяшек – М.: Издательский дом «Вильямс», 2002. – 432 с.

10. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. [Текст] / Г. Буч, Д. Рамбо – М.: ДМК Пресс, 2001. – 432 с.

11. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов [Текст] / Д. Розенберг, К. Скотт. – М.: ДМК-издательство, 2002. – 169 с.

12. Синицын, С. В., Налютин, Н. Ю. Верификация программного обеспечения [Текст] // С. В. Синицын, Н. Ю. Налютин – М.:Бином, 2008. – 368 c.

Міністерство освіти і науки України

ЗАТВЕРДЖУЮ

Керівник міждисциплінарного

курсового проекту,

доц. кафедри системотехніки

\_\_\_\_\_\_\_\_\_\_\_\_ Ю.В. Міщеряков

(підпис, дата)

ІНФОРМАЦІЙНА СИСТЕМА «МАГАЗИН З ПРОДАЖУ МОБІЛЬНИХ ТЕЛЕФОНІВ»

Текст програми

Аркуш затвердження

ГЮИК.506900.014 - 01 12 01 - ЛУ

Студент групи ІТКН-16-1

(назва групи)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, дата, прізвище, ім’я по батькові)

2018

Міністерство освіти і науки України

ЗАТВЕРДЖЕНО

ГЮИК. 506900.014 - 01 12 01 - ЛУ

ІНФОРМАЦІЙНА СИСТЕМА «МАГАЗИН З ПРОДАЖУ МОБІЛЬНИХ ТЕЛЕФОНІВ»

Текст програми

ГЮИК. 506900.014 - 01 12 01

Аркушів 4

2018

ЗМІСТ

Скрипт создания базы данных в MySQL:

-- -----------------------------------------------------

-- Schema telephones

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `telephones` DEFAULT CHARACTER SET utf8 ;

USE `telephones` ;

-- -----------------------------------------------------

-- Table `telephones`.`manufacture`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`manufacture` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

INDEX `manufacture\_title\_idx` (`title` ASC))

ENGINE = InnoDB

AUTO\_INCREMENT = 7

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `telephones`.`category`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`category` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `title\_UNIQUE` (`title` ASC))

ENGINE = InnoDB

AUTO\_INCREMENT = 5

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `telephones`.`telephone`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`telephone` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(45) NOT NULL,

`batch\_number` VARCHAR(45) NULL DEFAULT NULL,

`price` DOUBLE UNSIGNED NOT NULL DEFAULT '0',

`manufacture\_id` INT(11) NOT NULL,

`count` INT(11) UNSIGNED NOT NULL DEFAULT '0',

`category\_id` INT(11) NOT NULL,

`description` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`id`),

INDEX `telephone\_title\_idx` (`title` ASC),

INDEX `fk\_telephone\_category1\_idx` (`category\_id` ASC),

INDEX `fk\_telephone\_manufacture1\_idx` (`manufacture\_id` ASC),

CONSTRAINT `fk\_telephone\_manufacture1\_idx`

FOREIGN KEY (`manufacture\_id`)

REFERENCES `telephones`.`manufacture` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_telephone\_category1`

FOREIGN KEY (`category\_id`)

REFERENCES `telephones`.`category` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

AUTO\_INCREMENT = 8

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `telephones`.`user`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`user` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`login` VARCHAR(45) NOT NULL,

`password` VARCHAR(65) NOT NULL,

`role` ENUM('client','admin') NOT NULL DEFAULT 'client',

`e-mail` VARCHAR(45) NULL DEFAULT NULL,

`phone` VARCHAR(15) NULL DEFAULT NULL,

`name` VARCHAR(45) NULL DEFAULT NULL,

`address` VARCHAR(1000) NULL DEFAULT NULL,

`description` MEDIUMTEXT NULL DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE INDEX `user\_login\_idx` (`login` ASC))

ENGINE = InnoDB

AUTO\_INCREMENT = 3

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `telephones`.`delivery`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`delivery` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(45) NULL DEFAULT NULL,

`phone` VARCHAR(15) NULL DEFAULT NULL,

`email` VARCHAR(45) NULL DEFAULT NULL,

`address` VARCHAR(1000) NULL DEFAULT NULL,

`description` VARCHAR(1000) NULL DEFAULT NULL,

`user\_id` INT(11) NULL DEFAULT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_delivery\_user1\_idx` (`user\_id` ASC),

CONSTRAINT `fk\_delivery\_user1`

FOREIGN KEY (`user\_id`)

REFERENCES `telephones`.`user` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `telephones`.`order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`order` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`number` INT(11) NULL DEFAULT NULL,

`user\_id` INT(11) NULL DEFAULT NULL,

`date` DATETIME NULL DEFAULT CURRENT\_TIMESTAMP,

`status` ENUM('newed','inprogress','completed','rejected') NULL DEFAULT 'newed',

`delivery\_id` INT(11) NOT NULL,

PRIMARY KEY (`id`),

INDEX `fk\_order\_user1\_idx` (`user\_id` ASC),

INDEX `fk\_order\_delivery1\_idx` (`delivery\_id` ASC),

INDEX `order\_date\_idx` (`date` ASC),

INDEX `order\_status\_idx` (`date` ASC),

CONSTRAINT `fk\_order\_delivery1`

FOREIGN KEY (`delivery\_id`)

REFERENCES `telephones`.`delivery` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_order\_user1`

FOREIGN KEY (`user\_id`)

REFERENCES `telephones`.`user` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

-- -----------------------------------------------------

-- Table `telephones`.`telephone\_has\_order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`telephone\_has\_order` (

`telephone\_id` INT(11) NOT NULL,

`order\_id` INT(11) NOT NULL,

`count` INT(11) UNSIGNED NOT NULL,

`price` DOUBLE UNSIGNED NOT NULL DEFAULT '0',

PRIMARY KEY (`telephone\_id`, `order\_id`),

INDEX `fk\_telephone\_has\_order\_order1\_idx` (`order\_id` ASC),

INDEX `fk\_telephone\_has\_order\_telephone1\_idx` (`telephone\_id` ASC),

CONSTRAINT `fk\_telephone\_has\_order\_telephone1`

FOREIGN KEY (`telephone\_id`)

REFERENCES `telephones`.`telephone` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_telephone\_has\_order\_order1`

FOREIGN KEY (`order\_id`)

REFERENCES `telephones`.`order` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET = utf8;

USE `telephones` ;

-- -----------------------------------------------------

-- Placeholder table for view `telephones`.`telephones`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`telephones` (`id` INT, `title` INT, `manufactures` INT, `batch\_number` INT, `price` INT, `count` INT, `category` INT);

-- -----------------------------------------------------

-- Placeholder table for view `telephones`.`orders`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`orders` (`user\_id` INT, `login` INT, `order\_id` INT, `status` INT, `telephone\_id` INT, `title` INT, `count` INT, `price` INT, `osum` INT, `delivery\_id` INT);

-- -----------------------------------------------------

-- Placeholder table for view `telephones`.`sum\_by\_order`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `telephones`.`sum\_by\_order` (`oid` INT, `osum` INT);

-- -----------------------------------------------------

-- View `telephones`.`telephones`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `telephones`.`telephones`;

USE `telephones`;

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `telephones`.`telephones` AS select `telephones`.`telephone`.`id` AS `id`,`telephones`.`telephone`.`title` AS `title`,`telephones`.`telephone`.`batch\_number` AS `batch\_number`,`telephones`.`telephone`.`price` AS `price`,`telephones`.`telephone`.`count` AS `count`,`telephones`.`category`.`title` AS `category` from (`telephones`.`telephone` join `telephones`.`category`) where (`telephones`.`category`.`id` = `telephones`.`telephone`.`category\_id`);

-- -----------------------------------------------------

-- View `telephones`.`orders`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `telephones`.`orders`;

USE `telephones`;

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `telephones`.`orders` AS select `telephones`.`order`.`user\_id` AS `user\_id`,(select `telephones`.`user`.`login` from `telephones`.`user` where (`telephones`.`user`.`id` = `telephones`.`order`.`user\_id`)) AS `login`,`telephones`.`order`.`id` AS `order\_id`,`telephones`.`order`.`status` AS `status`,`telephones`.`telephone`.`id` AS `telephone\_id`,`telephones`.`telephone`.`title` AS `title`,`telephones`.`telephone\_has\_order`.`count` AS `count`,`telephones`.`telephone\_has\_order`.`price` AS `price`,`sum\_by\_order`.`osum` AS `osum`,`telephones`.`order`.`delivery\_id` AS `delivery\_id` from ((((`telephones`.`order` join `telephones`.`telephone`) join `telephones`.`telephone\_has\_order`) join `telephones`.`sum\_by\_order`) join `telephones`.`delivery`) where ((`telephones`.`order`.`id` = `telephones`.`telephone\_has\_order`.`order\_id`) and (`telephones`.`telephone\_has\_order`.`telephone\_id` = `telephones`.`telephone`.`id`) and (`telephones`.`order`.`id` = `sum\_by\_order`.`oid`)) order by `telephones`.`order`.`user\_id`,`telephones`.`order`.`id`,`telephones`.`telephone`.`id`;

-- -----------------------------------------------------

-- View `telephones`.`sum\_by\_order`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `telephones`.`sum\_by\_order`;

USE `telephones`;

CREATE OR REPLACE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `telephones`.`sum\_by\_order` AS select `telephones`.`telephone\_has\_order`.`order\_id` AS `oid`,sum((`telephones`.`telephone\_has\_order`.`price` \* `telephones`.`telephone\_has\_order`.`count`)) AS `osum` from `telephones`.`telephone\_has\_order` group by `telephones`.`telephone\_has\_order`.`order\_id`;

USE `telephones`;

DELIMITER $$

USE `telephones`$$

CREATE

DEFINER=`root`@`localhost`

TRIGGER `telephones`.`telephone\_has\_order\_BEFORE\_INSERT`

BEFORE INSERT ON `telephones`.`telephone\_has\_order`

FOR EACH ROW

begin

SET @price = (select price from telephone where id = NEW.`telephone\_id`);

SET @newcount = (select count from telephone where id = NEW.`telephone\_id`) - NEW.count;

SET NEW.price = @price;

UPDATE telephone SET telephone.count = @newcount where id = NEW.`telephone\_id`;

end$$

DELIMITER ;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

Міністерство освіти і науки України

уТВЕРЖДАЮ

ГЮИК. 506900.014 ИЗ – ЛУ

ІНФОРМАЦІЙНА СИСТЕМА «МАГАЗИН З ПРОДАЖУ МОБІЛЬНИХ ТЕЛЕФОНІВ»

Посібник користувача

ГЮИК. 506900.014 ИЗ

Аркушів 3

2018

Міністерство освіти і науки України

Затверджую

Керівник міждисциплінарного

курсового проекту,

доц. кафедри системотехніки

\_\_\_\_\_\_\_\_\_\_\_\_ Ю.В. Міщеряков

(підпис, дата)

ІНФОРМАЦІЙНА СИСТЕМА «МАГАЗИН З ПРОДАЖУ МОБІЛЬНИХ ТЕЛЕФОНІВ»

Посібник користувача

Аркуш затвердження

ГЮИК. 506900.014 - 34 01 ИЗ - ЛУ

Студент групи ІТКН-16-1

(назва групи)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис, дата, прізвище, ім’я по батькові)

2018

ВСТУП

Інтернет-магазин (англ. Online shop або e-shop) - сайт, який торгує товарами в інтернеті. Дозволяє користувачам сформувати замовлення на покупку і вказати доставку в мережі Інтернет. [1]

Вибравши необхідні товари, користувач ма можливість перейти у вкладку кошик і ведучи персональні дані оформити свою покупку. Сукупність відібраних товарів, спосіб оплати і доставки являють собою закінчене замовлення, яке оформляється на сайті шляхом повідомлення мінімально необхідної інформації про покупця. Інформація про покупця зберігатися в базі даних магазину і розрахована на повторні покупки.

Адміністратор інтернет-магазину має можливість додавати і редагувати відповідні товари магазину, стежити за кількістю необхідного товару на складі, відслідковувати статуси замовлень користувачів і переводити їх в різні стани.

Користувачі системи повинні мати досвід роботи з персональним комп'ютером на рівні користувача, володіти знаннями по роботі з MySQL WorkBench і вміти здійснювати базові операції в мережі Інтернет.

Призначення і умови застосування

Розглядається електронна комерція магазину з продажу мобільних телефонів, оформлення, обліку і виконання замовлень, документування всіх облікових даних. Основною метою системи автоматизувати облік продажів. Серверна частина забезпечує взаємодію з базою даних, розроблену для платформи СУБД MySQL. Клієнтська частина забезпечує інтерфейс з користувачем.

Відмінною рисою даної СУБД є простота її використання в поєднанні з широкими можливостями по розробці додатків. Щоб користуватися програмою необхідний набір інструментів, а саме один з популярних браузерів для операційної система - Windows 7 або вище, програмний пакет Workbench, Eclips EE для запуску програми.

Підготовка до роботи

Запустити безпосередньо базу даних MySQL і сам проект в програмному середовищі Eclipse EE. Для перевірки працездатності системи досить виконати базові операції з нею які описані в розділі нижче. При першому використані системи потрібно мати данні для заповненя відповідної таблиці товарів.

Опис операцій

Для всіх перерахованих операцій нижче необхідно мати запущений сайт.

а) Для входу в систему необхідно перейти на відповідні вкладку і вести, попередньо зареєстрованого, ім'я користувача в поле login, кількість символів не повинно перевищувати 21 символ і також необхідно вести пароль користувача в поле password, кількість символів якого також не повинно буте більше 21 символу.

б) Для реєстрації в система необхідно перейти на відповідну вкладку в системі і заповнити поле ім'я користувача і пароль кількість символів, яких не повинно перевищувати 21 символ, в подальші дії ці поля буде використовуватися для входу в систему.

в) Для додавання товару необхідно увійти в систему під обліковим записом admin, перейти на сторінку каталог і натиснути на відповідну кнопку додати товар, надалі ви побачите вікно з полями для заповнення, для підтвердження або скасування слід використовувати кнопки розташовані в самому низу вікна. Кількість символів в назві товару не повинна перевищувати 15, кількість на складі не повинно перевищувати 10 символів, категорія може бути змінена, кількість символів в ціні товару може бути дробовим числом і має перевищувати 21 символ.

г) Для зміни товару необхідно увійти в систему під обліковим записом admin, перейти на сторінку каталог і натиснути на соответсвующію кнопку змінити товар, надалі ви побачите вікно з інформацією про товар, всю інформацію можна змінювати, для підтвердження або скасування слід використовувати кнопки распаложенние в самому низу вікна. Кількість символів в назві товару не повинна перевищувати 15, кількість на складі не повинно перевищувати 10 символі, категорія може бути змінена, кількість символів в ціні товару може бути дробовим числом і має перевищувати 21 символ.

д) Для оформлення замовлення користувачеві необхідно вибрати товари і сформувати замовлення за допомогою кнопки додати в кошик, після перейти на сторінку кошика і оформити доставку. Кількість товарів не обмежена, оформляти товар може як зареєстрований користувач так і незареєстрований.

Аварійні ситуації

При виникненні помилки у заповнені даних слід увійти в систему під обліковим записом адмін і перейшовши на сторінку каталогу змінити дані.

При виникненні проблем у виконання замовлення слід увійти в систему під обліковим записом адмін і перейшовши на сторінку замовлення, скасувати відповідне замовлення.

Якщо виникають проблеми з додатку слід зберегти помилку і звернутися до розробника програмного забезпечення.

Рекомендації з освоєння

Виконувати вимоги, описані в підготовці до роботи.