НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

***з дисципліни "Основи програмування"***

Виконала: Євтушенко Вікторія Павлівна

Група: КП-03

Допущено до захисту

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

2 семестр 2020/2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено  Керівник роботи    \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ | ЗАХИЩЕНА  "\_\_"\_\_\_\_\_\_\_\_\_2021р.  з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ |

**Програмний додаток електронного магазину товарів**

Виконавець роботи

Євтушенко Вікторія Павлівна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021р.

Зміст

[Вступ 4](#_Toc74996591)

[1. Аналіз предметної галузі 5](#_Toc74996592)

[1.1. Аналіз вимог до функціональності програмних засобів 5](#_Toc74996593)

[2. Аналіз мов програмування та технологій розроблення 6](#_Toc74996594)

[2.1. Мова програмування С# 6](#_Toc74996595)

[2.2. Фреймворк .NET 7](#_Toc74996596)

[2.3. База даних SQLite 7](#_Toc74996597)

[2.4. Бібліотека Terminal.Gui 7](#_Toc74996598)

[2.5. Бібліотека ScottPlott 7](#_Toc74996599)

[2.6. Бібліотека Microsoft.Data.Sqlite 7](#_Toc74996600)

[2.7. Бібліотека System.Collections.Generic 7](#_Toc74996601)

[2.8. Бібліотека System.IO 7](#_Toc74996602)

[2.9. Бібліотека System.Linq 8](#_Toc74996603)

[2.10. Бібліотека System.Xml 8](#_Toc74996604)

[2.11. Бібліотека System.Xml.Linq 8](#_Toc74996605)

[3. Опис розроблених програмних засобів 8](#_Toc74996606)

[3.1. Загальна структура системи 8](#_Toc74996607)

[3.2. Структура бази даних 9](#_Toc74996608)

[3.3. Модулі репозиторіїв даних 10](#_Toc74996609)

[3.4. Модуль аутентифікації 14](#_Toc74996610)

[3.5. Модуль експорту 15](#_Toc74996611)

[3.6. Модуль імпорту 15](#_Toc74996612)

[3.7. Модулі генерування звіту 15](#_Toc74996613)

[3.8. Бібліотека LibraryClass 17](#_Toc74996614)

[4. Аналіз розроблених програмних засобів 18](#_Toc74996615)

[4.1. Особливості реалізації 18](#_Toc74996616)

[4.2. Інтерфейс користувача 18](#_Toc74996617)

[Висновки 20](#_Toc74996618)

# Вступ

*(Приклад, змінити, доповнити)*

В процесі пошуку товарів у електронному форматі часто виникає необхідність у зручному ресурсі, магазині, який би дозволяв обирати товар, перевіряти своє замовлення та ін.

Проблемою зазвичай є незрозумілий інтерфейс користувача, невдалий вибір характеристик, які б описували товар, незручний формат пошуку. Виникає необхідність створення програмних додатків, які б задовольняли рішення цих проблем. Саме тому пропонується розробка програмних засобів, які б дозволяли користувачам виконувати замовлення, виконувати пошук потрібних їм товарів, тощо.

Створення системи, що забезпечує надійну базу даних та зручний формат, є актуальною задачею.

Даний проект присвячено розробленню програмного додатку, який призначено для використання та управління електронним магазином товарів.

# 1. Аналіз предметної галузі

## 1.1. Аналіз вимог до функціональності програмних засобів

В процесі аналізу вимог до системи було виділено такі функціональні вимоги:

1. Наявний консольний інтерфейс користувача. Команди використані відповідно до їхнього основного призначення.
   1. Наявно декілька станів інтерфейсу користувача.
   2. Дотримується єдиний стиль для всіх команд
2. Наявні стани для створення і редагування даних.
3. Можна видаляти дані (з підтвердженням цієї дії).
4. Наявний стан редагування даних має мати вже заповнені поля із попередніми значеннями. Є можливість відмінити оновлення сутності.
5. Використання списків та таблиць для показу колекцій даних.
6. Дані в таблиці не є лише відображенням однієї таблиці БД, а містять склеєні чи змінені дані.
7. Дані у списках та таблицях пагінуються, доступний пошук та фільтрація їх записів.
8. Всі вхідні дані від користувача та з файлів перевіряються, помилки обробляються, користувачу відображається інформація про помилки.
9. Програма має можливість завантаження зображень з файлової системи, їх збереження у сховище даних та їх показ у інтерфейсі користувача.
10. Програма має можливість експорту-імпорту даних у форматі даних XML. Є можливість зберігання даних у файлах, та передача даних до інших процесів через протокол TCP.
11. Програма взаємодіє з реляційною базою даних. SQL запити до бази даних не конкатенуються, а їх виконання винесено у спеціальний окремий модуль (вид репозиторія).
12. База даних містить мінімум 3 таблиці. Таблиці в БД приведені до 1-ї та 2-ї нормальних форм.
13. Між таблицями бази даних є one-to-many та/або many-to-many зв'язки
14. Наявна реєстрація користувачів у системі. Паролі користувачів не зберігаються у відкритому виді, а хешуються обраним алгоритмом хешування рядків.
15. Наявний стан аутентифікації. Пароль користувача при введені прихований. Цей стан має сповіщення про некоректність введених користувачем даних.
16. Програма авторизує доступу до даних по користувачах. У базі є дані, що належать конкретному користувачу (наприклад, авторство сутностей).
17. Система дозволяє генерувати зображення та файлові звіти на основі даних, що зберігаються у БД.
18. Частина проекту винесена у власні бібліотеки класів. Винесення саме цього коду обгрунтоване.

# 2. Аналіз мов програмування та технологій розроблення

## 2.1. Мова програмування С#

Були використані такі можливості мови програмування C# як наслідування класів, серіалізація даних, використання стандартних типів даних.

## 2.2. Фреймворк .NET

За допомогою фреймворку .NET використовувалися деякі бібліотеки (System, System.Collections.Generic, System.Text, System.IO, System.IO.Compression, System.Linq, System.Xml, System.Xml.Serialization, System.Xml.Linq, Terminal.Gui, ScottPlot, Microsoft.Data.Sqlite), призначення яких описано нижче.

## 2.3. База даних SQLite

Були використані наступні можливості бази даних: створення, редагування та видалення запису у таблиці, отримання записів таблиці, пошук по значенням.

## 2.4. Бібліотека Terminal.Gui

Використовувалася для створення подійно-орієнтованого інтерфейсу користувача. З її допомогою створювалися графічні об’єкти, такі як вікно, діалог, кнопка, тощо.

## 2.5. Бібліотека ScottPlott

Застосовувался для генерації графіку за варіантом (histogram графік розподілу цін на товари).

## 2.6. Бібліотека Microsoft.Data.Sqlite

Використовувалася для створення зв’язку та отримання доступу до бази даних.

## 2.7. Бібліотека System.Collections.Generic

Застосовувалася для роботи із такими структурами даних, як списки та словники.

## 2.8. Бібліотека System.IO

Використовувалася для взаємодії із файловою системою, зчитування та запису інформації.

## 2.9. Бібліотека System.Linq

Застосовувалася для роботи з колекціями та об’єктами.

## 2.10. Бібліотека System.Xml

Використовувалася для серіалізації/десеріалізації даних.

## 2.11. Бібліотека System.Xml.Linq

Застосовувалася для модифікації документів XML, зчитуванню та заміни даних у них.

# 3. Опис розроблених програмних засобів

## 3.1. Загальна структура системи

Програмні засоби реалізовані у вигляді десктопного додатку з графічним інтерфейсом користувача. Структурна схема системи зображена на рис.

UI client

SQLite DB

Рис. Структурна схема системи

Проект розбитий на такі модулі:

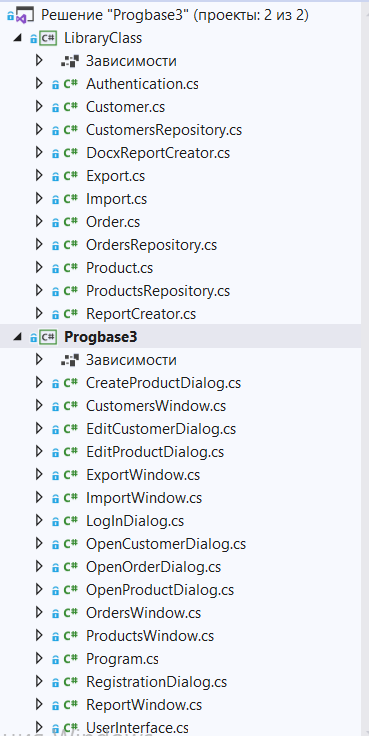


Рис. Модулі проекту

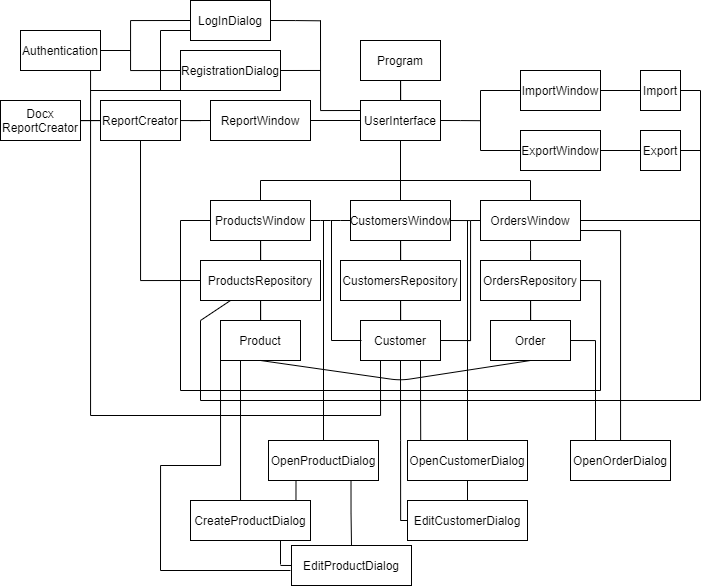
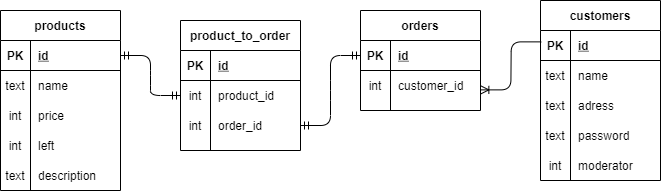
Діаграма залежностей модулів IDE:

Рис. Залежність модулів IDE

## 3.2. Структура бази даних

На рис. представлена ER-діаграма структури бази даних системи.

Рис. Структура бази даних

Таблиця products описує сутність «товар» і містить наступні колонки: id (INTEGER NOT NULL UNIQUE), name (TEXT NOT NULL), price (INTEGER NOT NULL), left (INTEGER NOT NULL), description (TEXT NOT NULL), де id – primary key.

Таблиця customers описує користувача і містить наступні колонки: id (INTEGER NOT NULL UNIQUE), name (TEXT NOT NULL), address (TEXT NOT NULL), password (TEXT NOT NULL), moderator (INTEGER NOT NULL), де id – primary key. При зчитуванні даних у сутність, значення поля moderator конвертується у boolean.

Таблиця orders описує сутність «замовлення» і містить наступні колонки: id (INTEGER NOT NULL UNIQUE), customer\_id (INTEGER NOT NULL), де id – primary key, customer\_id – foreign key до таблиці customers. Між користувачем та сутністю «замовлення» встановлений зв’язок one-to-many.

Таблиця product\_to\_order створена для опису зв’язку many-to-many між продуктом та замовленням і містить наступні колонки: product\_id (INTEGER NOT NULL), customer\_id (INTEGER NOT NULL), id (INTEGER NOT NULL), де id – primary key, product\_id та customer\_id – foreign keys.

## 3.3. Модулі репозиторіїв даних

Інтерфейс модулів:

|  |
| --- |
| **ProductsRepository.сs** |
| using System;  using System.Collections.Generic;  using Microsoft.Data.Sqlite;  namespace LibraryClass  {  public class ProductsRepository  {  private SqliteConnection connection;  public ProductsRepository(SqliteConnection connection)  {  this.connection = connection;  }  public static Product GetProduct(SqliteDataReader reader)  {  }  public long Insert(Product p)  {  }  public bool Delete(long id)  {  }  public List<Product> GetExport(string valueX)  {  }  public List<Product> GetProducts()  {  }  public List<Order> GetOrdersOfProduct(long customer\_id)  {  }  public bool Update(long id, Product p)  {  }  private long GetCount()  {  }  public int GetTotalPages(int pageSize)  {  }  public List<Product> GetPage(int pageNumber, int pageSize)  {  }  public int GetSearchPagesCount(int pageSize, string value)  {  }  public List<Product> GetSearchPage(string value, int pageNumber, int pageSize)  {  }  public Dictionary<long, List<Product>> GetProductsInOrders()  {  }  private static Product GetProductForOrder(SqliteDataReader reader)  {  }  }  } |
| **OrdersRepository.cs** |
| using System;  using System.Collections.Generic;  using Microsoft.Data.Sqlite;  namespace LibraryClass  {  public class OrdersRepository  {  private SqliteConnection connection;  public OrdersRepository(SqliteConnection connection)  {  this.connection = connection;  }  public static Order GetOrder(SqliteDataReader reader)  {  }  public Order GetById(int id)  {  }  public long Insert(Order o)  {  }  public bool Delete(long id)  {  }  public List<Order> GetCustomersOrders(long customer\_id)  {  }  public List<Order> GetOrdersOfProduct(long productId)  {  }  private long GetCount(long customer\_id)  {  }  public int GetTotalPages(int pageSize, long customer\_id)  {  }  public List<Order> GetPage(int pageNumber, int pageSize, long customer\_id)  {  }  public List<Product> GetProductsInOrder(long orderId)  {  }  }  } |
| **CustomersRepository.cs** |
| using System;  using System.Collections.Generic;  using Microsoft.Data.Sqlite;  namespace LibraryClass  {  public class CustomersRepository  {  private SqliteConnection connection;  public CustomersRepository(SqliteConnection connection)  {  this.connection = connection;  }  static Customer GetCustomer(SqliteDataReader reader)  {  }  public Customer GetById(int id)  {  }  public long Insert(Customer c)  {  }  public bool Delete(long id)  {  }  public Customer GetToLogin(string name, string password)  {  }  public bool GetToRegister(string name, string adress)  {  }  public bool Update(long id, Customer c)  {  }  private long GetCount()  {  }  public int GetTotalPages(int pageSize)  {  }  public List<Customer> GetPage(int pageNumber, int pageSize)  {  }  public int GetSearchPagesCount(int pageSize, string value)  {  }  public List<Customer> GetSearchPage(string value, int pageNumber, int pageSize)  {  }  }  } |

Використовувалися для доступу до бази, зчитування даних, їх пошуку, редагування, видалення.

## 3.4. Модуль аутентифікації

Інтерфейс модуля:

|  |
| --- |
| **Authentication.cs** |
| using System;  using System.Collections.Generic;  using System.Text;  using System.Security.Cryptography;  namespace LibraryClass  {  public class Authentication  {  private SHA256 sha256Hash = SHA256.Create();  public Customer Register(CustomersRepository customersRepository, string userName, string password, string adress)  {  }  public Customer LogIn(CustomersRepository customersRepository, string possibleName, string possiblePassword)  {  }  private static string GetHash(HashAlgorithm hashAlgorithm, string input)  {  }  private static bool VerifyHash(HashAlgorithm hashAlgorithm, string input, string hash)  {  }  }  } |

Використовувався для авторизації користувача, хешування його пароля, збереження даних користувача у базі.

## 3.5. Модуль експорту

Інтерфейс модуля:

|  |
| --- |
| **Export.cs** |
| using System.Collections.Generic;  using System.IO;  using System.Xml.Serialization;  using System.IO.Compression;  namespace LibraryClass  {  public class Export  {  public void ExportProducts(ProductsRepository rep, string value, string xmlFilePath, string sourceFolder, string zipFile)  {  }  }  } |

Використовувався для експорту даних (всі товари, що містять заданий підрядок) з бази у документ XML та архівував утворені файли.

## 3.6. Модуль імпорту

Інтерфейс модуля:

|  |
| --- |
| **Import.cs** |
| using System.Collections.Generic;  using System.IO;  using System.IO.Compression;  using System.Xml.Serialization;  namespace LibraryClass  {  public class Import  {  public void ImportProduct(ProductsRepository rep, string targetFolder, string zipFile, string xmlFilePath)  {  }  }  } |

Використовувався для імпорту усіх експортованих даних з файлу XML у базу.

## 3.7. Модулі генерування звіту

Інтерфейс модулів:

|  |
| --- |
| **DocxReportCreator.cs** |
| using System.IO;  using System.IO.Compression;  using System.Xml;  using System.Xml.Linq;  namespace LibraryClass  {  class DocxReportCreator  {  private string name;  private string destination;  public DocxReportCreator(string fileName)  {  }  public XDocument GetXmlTemplate()  {  }  public void SaveDocx(XDocument document)  {  }  public string GetImageFileName()  {  }  private void CopyDirectory(DirectoryInfo dir, string destination)  {  }  private void DeleteDirectory(string path)  {  }  private XDocument ReadXml(string filePath)  {  }  }  } |
| **ReportCreator.cs** |
| using System.Collections.Generic;  using ScottPlot;  using System.Linq;  using System.Xml;  using System.Xml.Linq;  namespace LibraryClass  {  public class ReportCreator  {  private ProductsRepository productsRepository;  public ReportCreator(ProductsRepository productsRepository)  {  this.productsRepository = productsRepository;  }  public void SaveReport(string filePath)  {  }  private void ReplaceImage(string filePath)  {  }  public void FindAndReplace(XElement node)  {  }  private string GetMostPopularProduct()  {  }  private string GetMaxPriceProduct()  {  }  private string GetMinPriceProduct()  {  }  private string GetMaxRevenueProduct()  {  }  private string GetAvgPrice()  {  }  private string GetTotalCount()  {  }  public void PlotImage(ProductsRepository rep, string filePath)  {  }  }  } |

Використовувалися для генерації звіту формату DOCX, що містив інформацію про кількість товарів, середню ціну товарів, інформацію про найдорожчий та найдешевший товар, товар, який замовили найбільшу кількість раз, товар, який приніс найбільшу виручку, згенерований графік.

## 3.8. Бібліотека LibraryClass

Бібліотека LibraryClass була створена для доступу до бази даних, опису всіх сутностей, експорту, імпорту та генерації звіту.

# 4. Аналіз розроблених програмних засобів

## 4.1. Особливості реалізації

1. Особливості авторизації користувачів:
2. реєстрація перевіряє унікальність імені та адреси користувача і створює новий запис у таблиці користувачів;
3. функція логіну перевіряє надані ім’я та пароль користувача і повертає об’єкт користувача на основі запису з БД.
4. Паролі користувачів хешуються за допомогою алгоритму SHA256 та зберігаються у базі даних у захешованному вигляді.
5. Роль користувача зазначається у базі даних як ціле число, яке потім програмно конвертується у значення bool.
6. Сутність Product є Serializable.

## 4.2. Інтерфейс користувача

Для показу даних система має інтерфейс користувача, що складається з:

1. Вікна аутентифікації;
2. Головного вікна;
3. Вікна, що містить список товарів та поле їх пошуку;
4. Вікна, що містить список замовлень користувача, що авторизований у програмі;
5. Вікна, що містить список користувачів, видимий лише для користувача ролі «модератор» та кнопку, що дає доступ до власної персональної інформації.

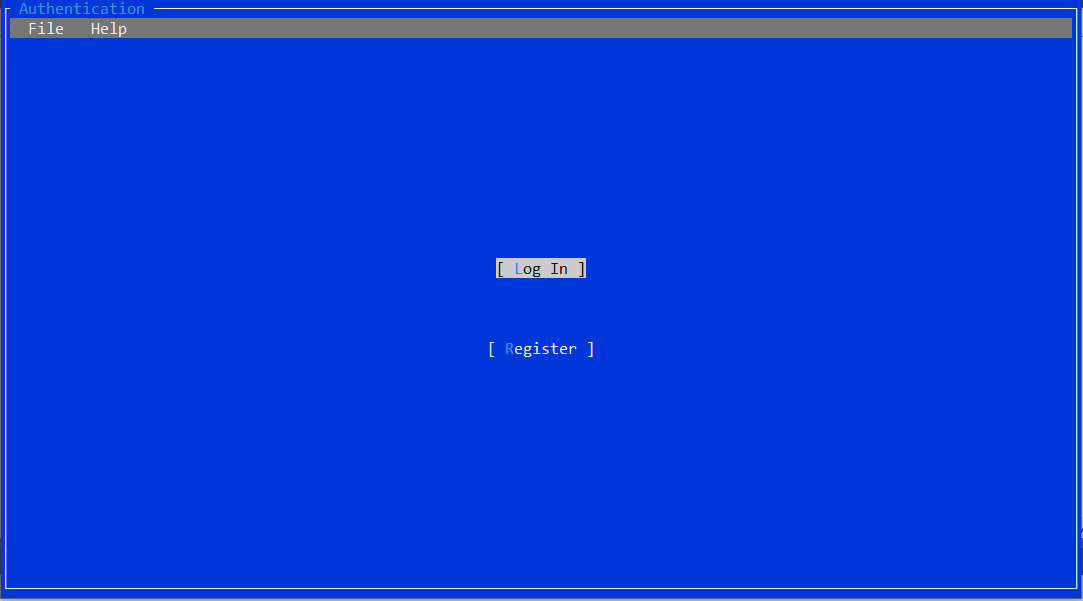
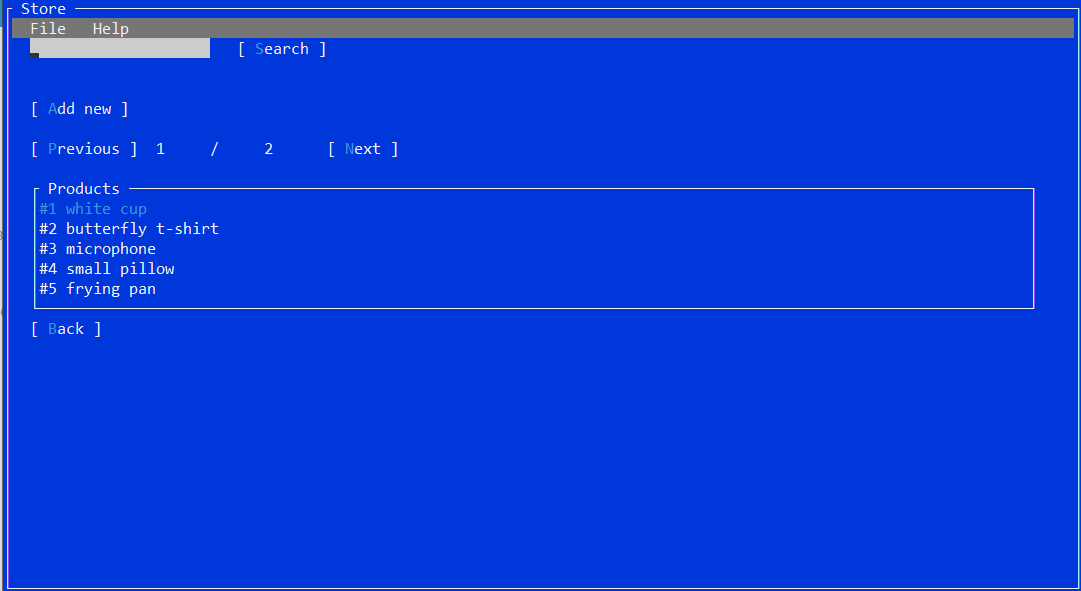
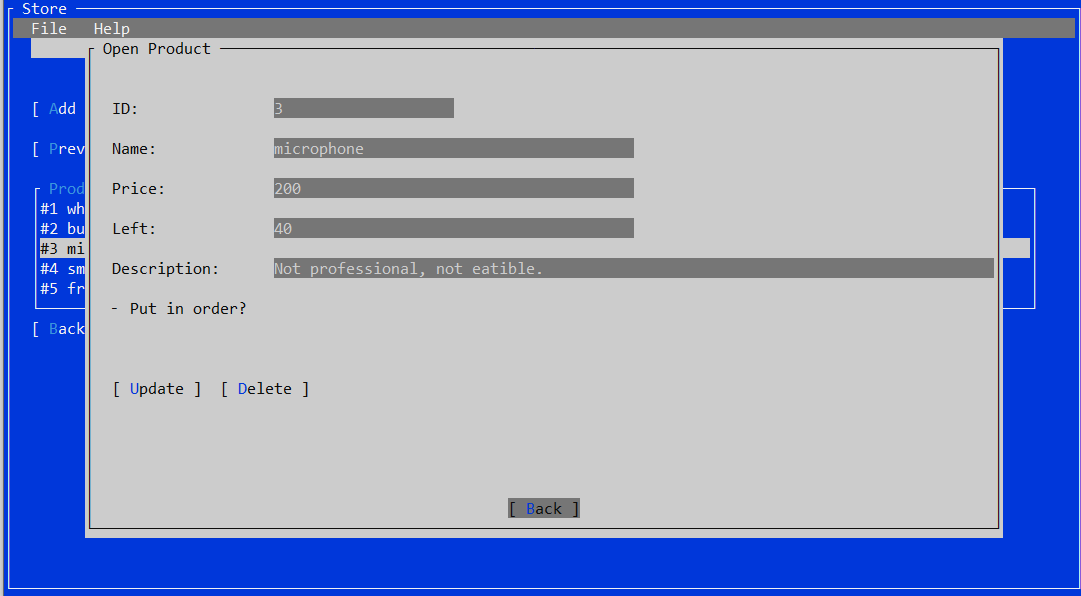
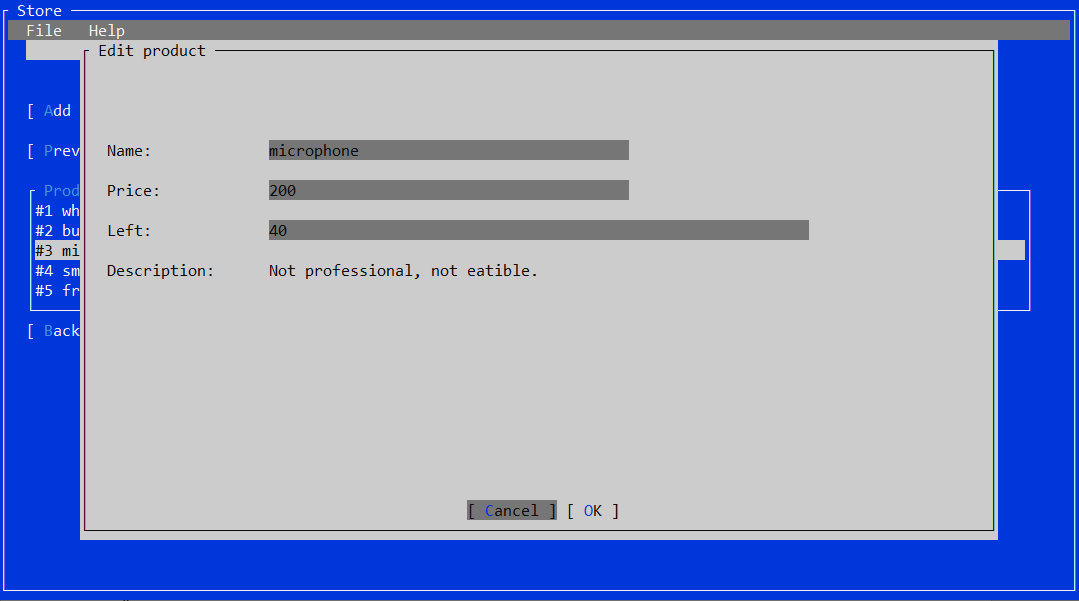
Рис. Вікно аутентифікації

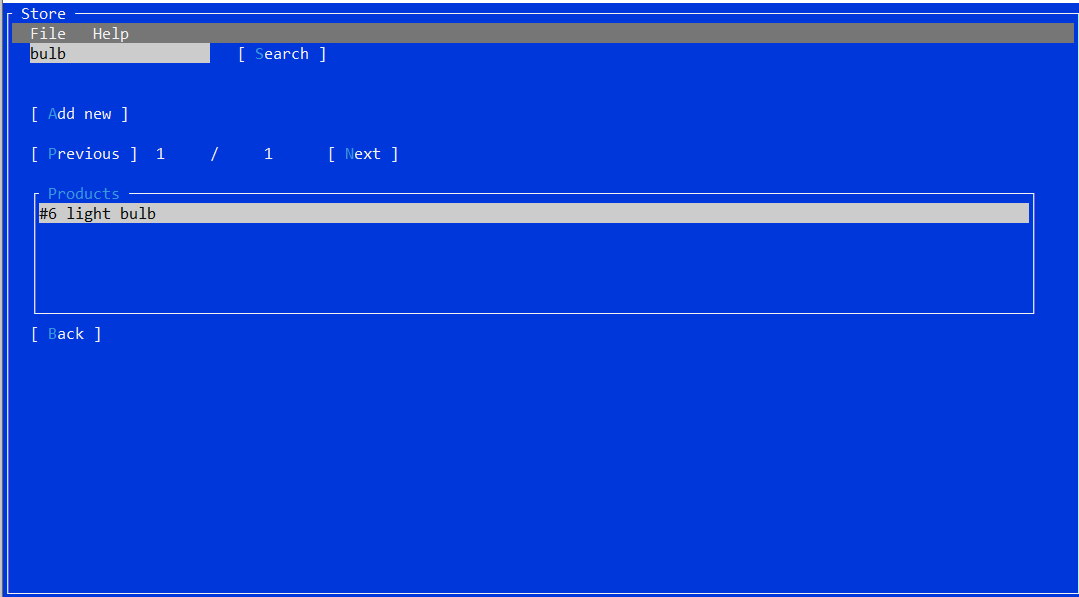
Рис. Головне вікно

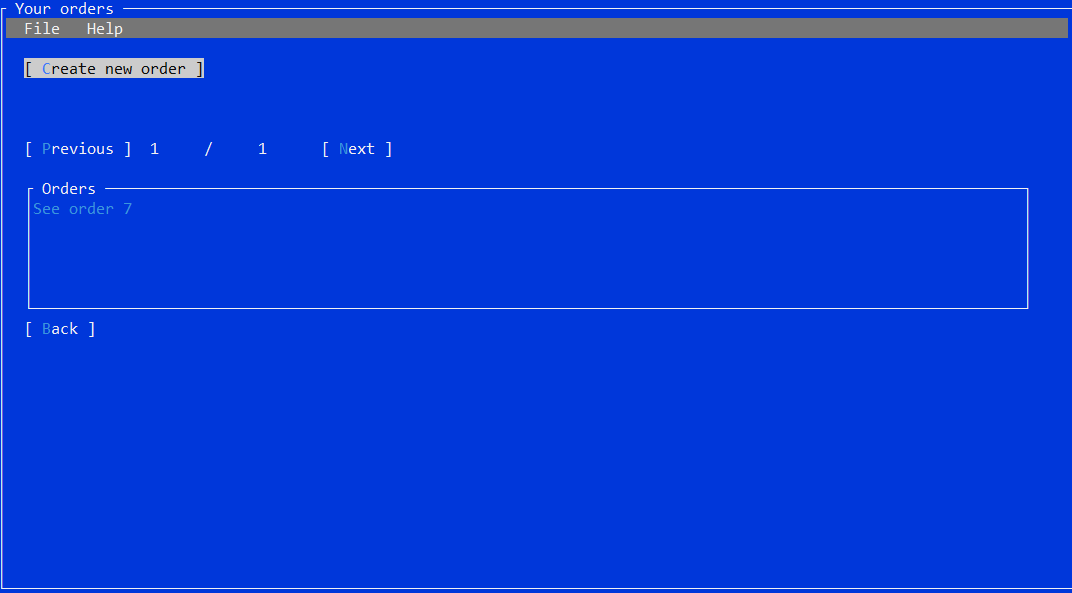
Рис. Вікно зі списком товарів

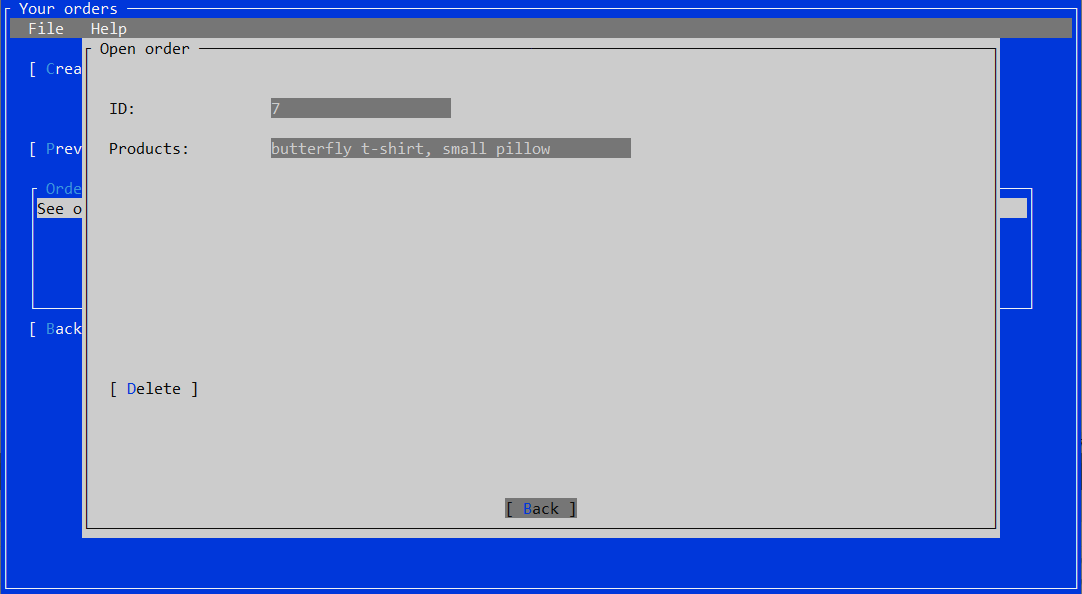
Рис. Діалог перегляду товару

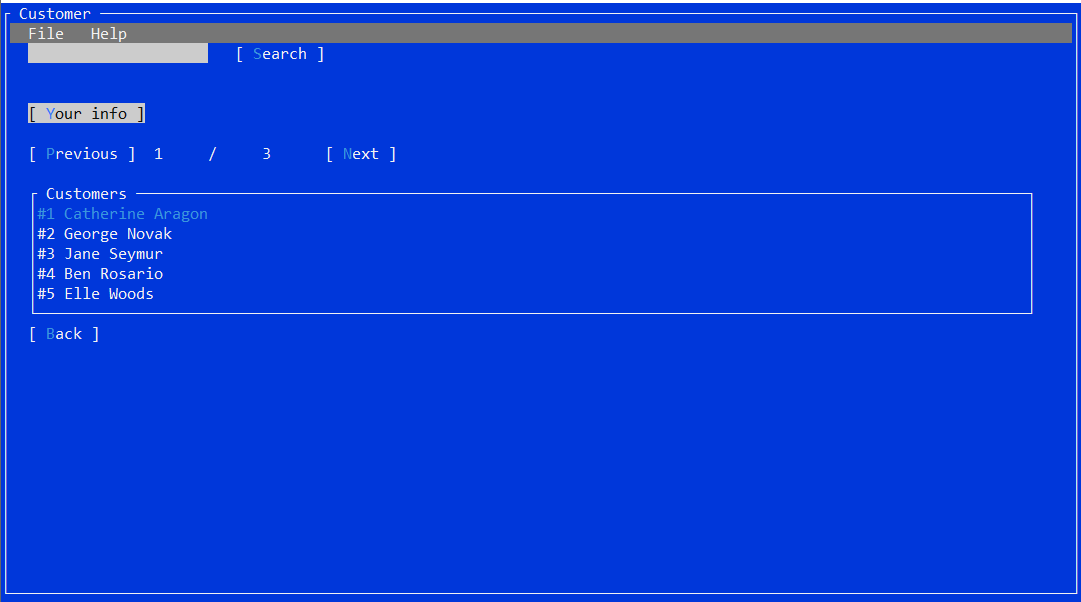
Діалог редагування товару

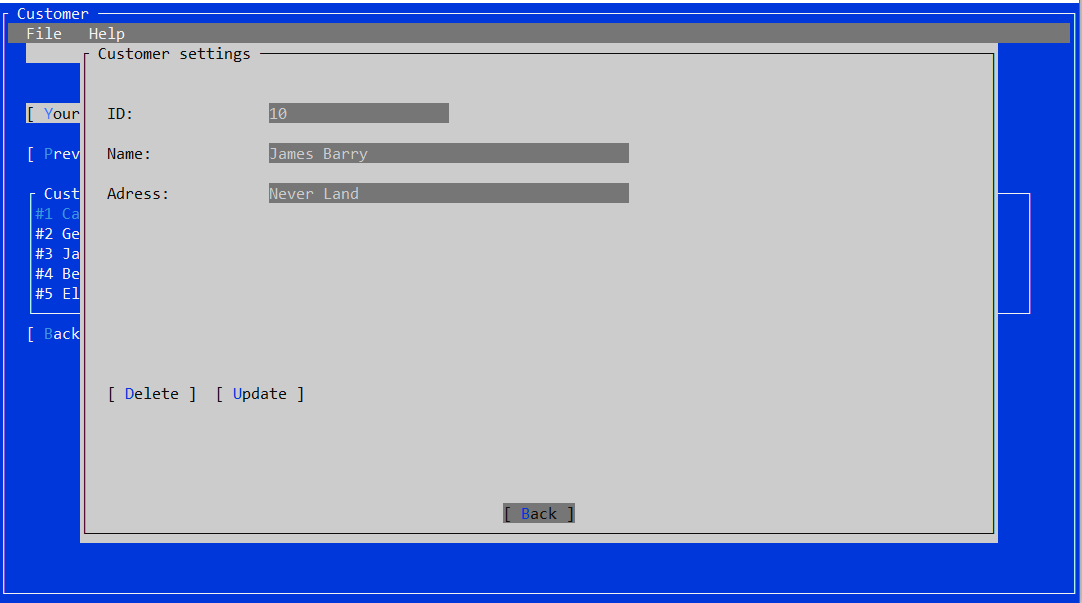
Діалог створення товару

Рис. Вікно після успішного пошуку

Рис. Вікно зі списком замовлень

Рис. Діалог перегляду замовлення

Рис. Вікно зі списком користувачів (авторизований користувач – модератор)

Рис. Діалог перегляду персональної інформації

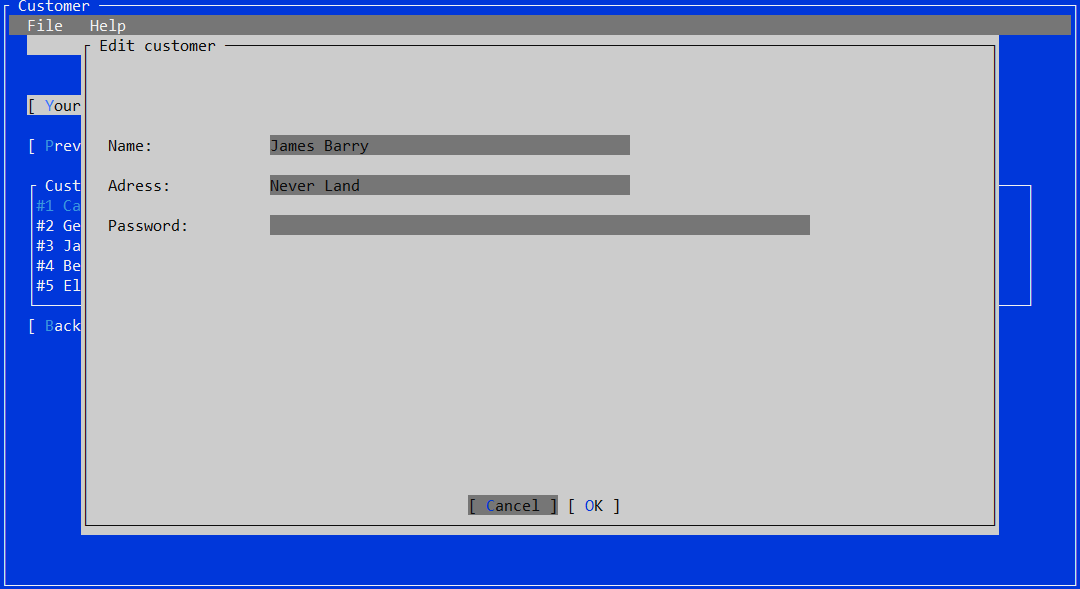


Рис. Діалог редагування персональної інформації

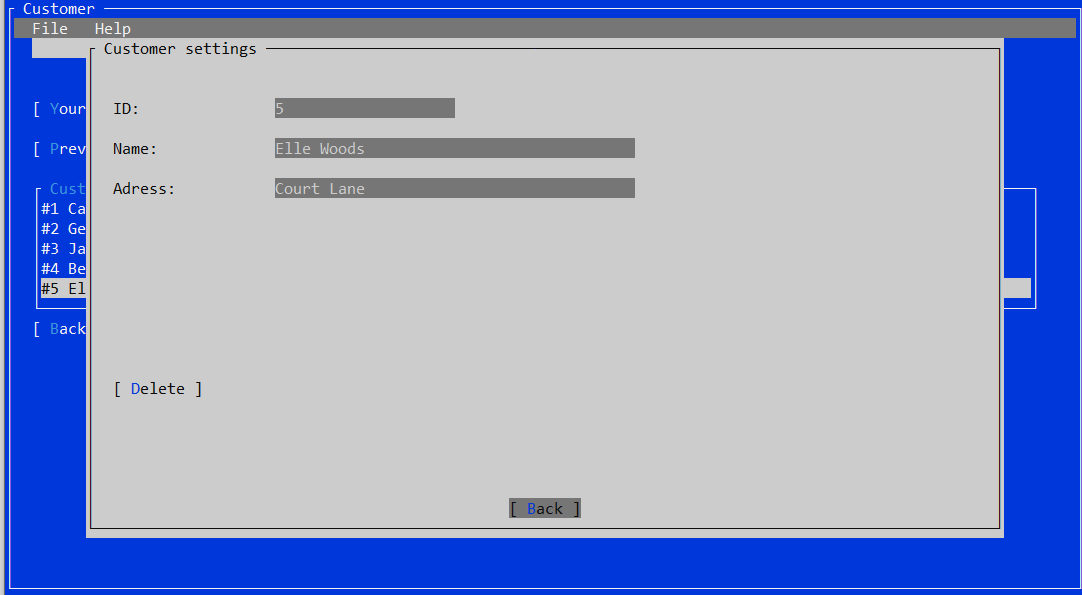
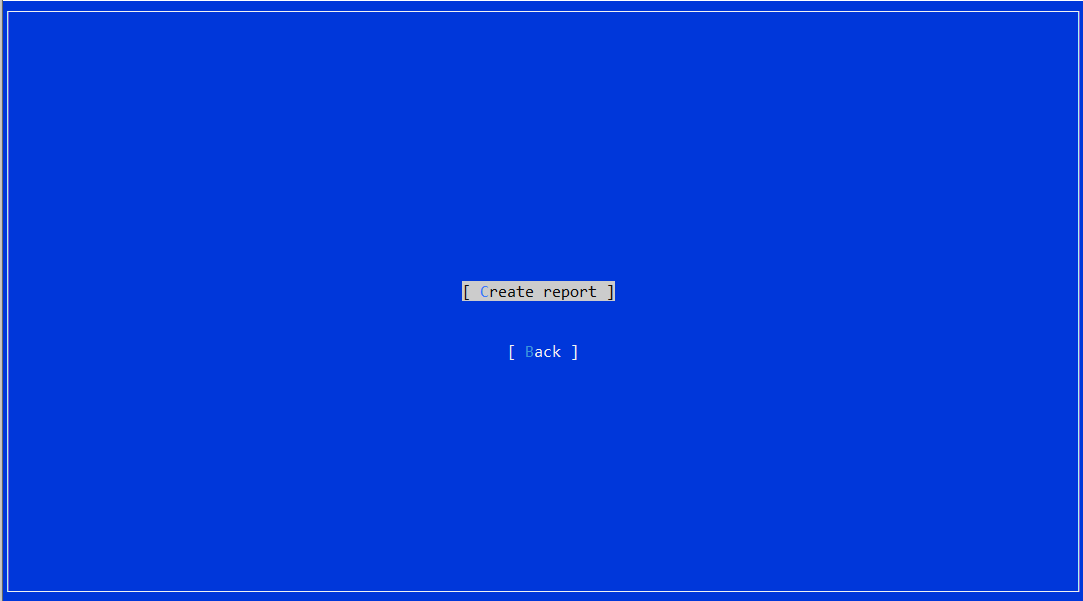
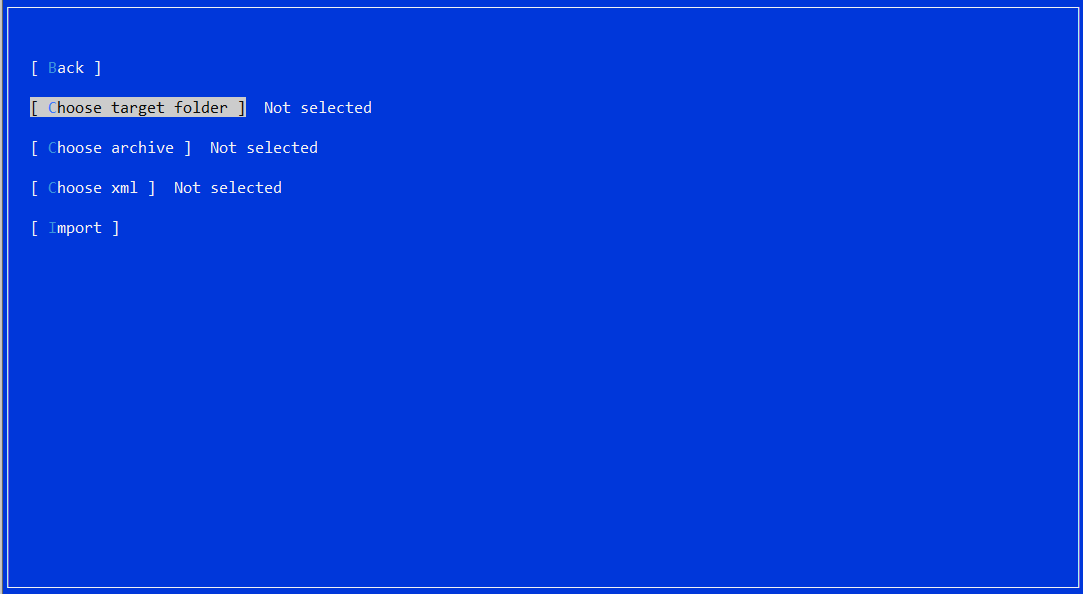
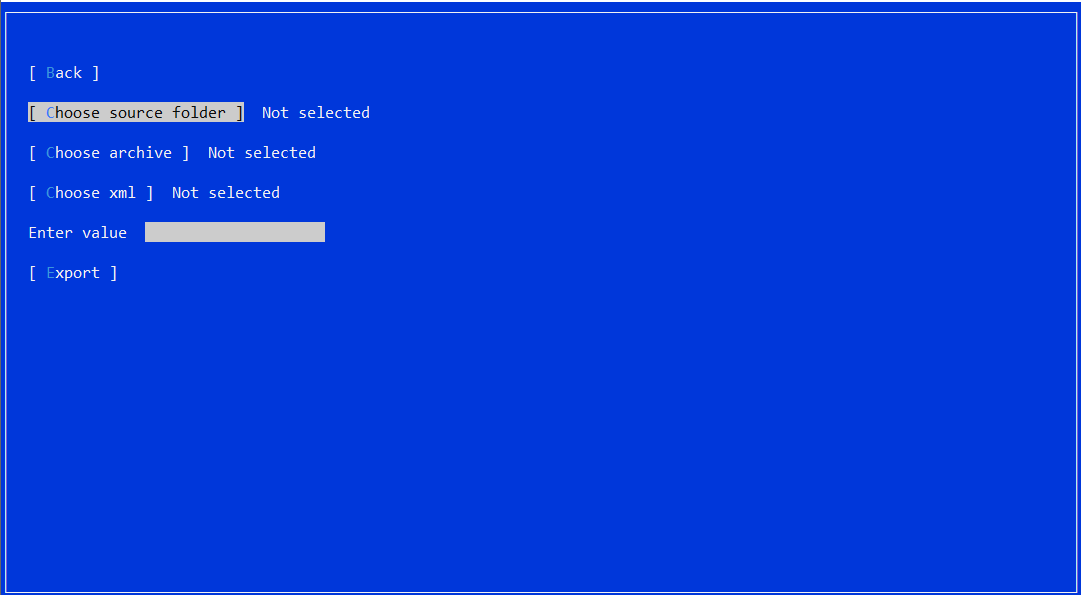
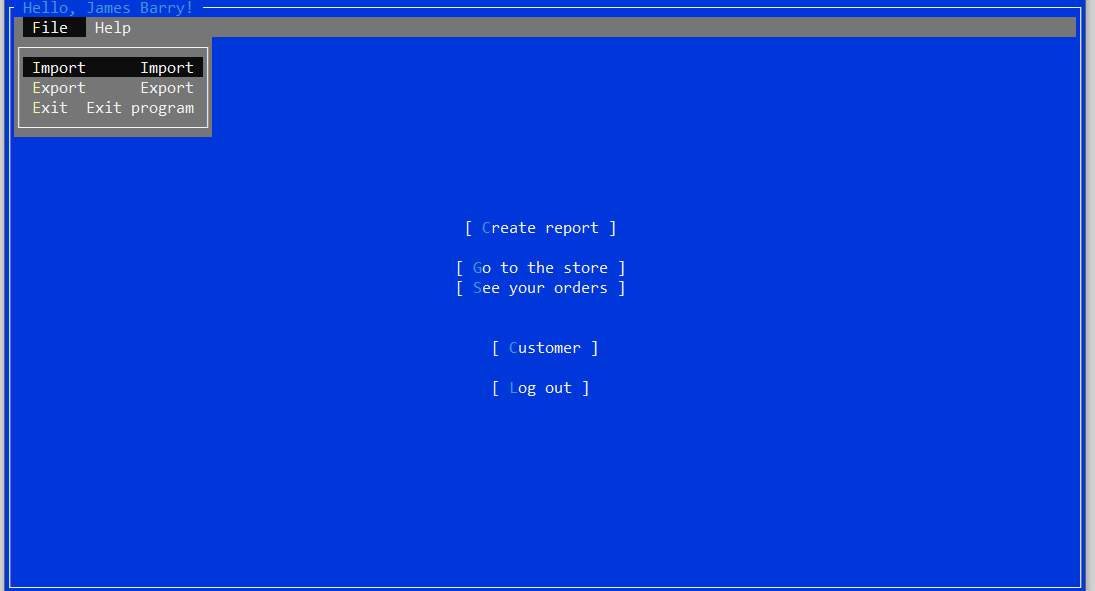


Рис. Діалог перегляду чужої інформації

Рис. Вікно генерації звіту

Рис. Вікно імпорту

Рис. Вікно експорту



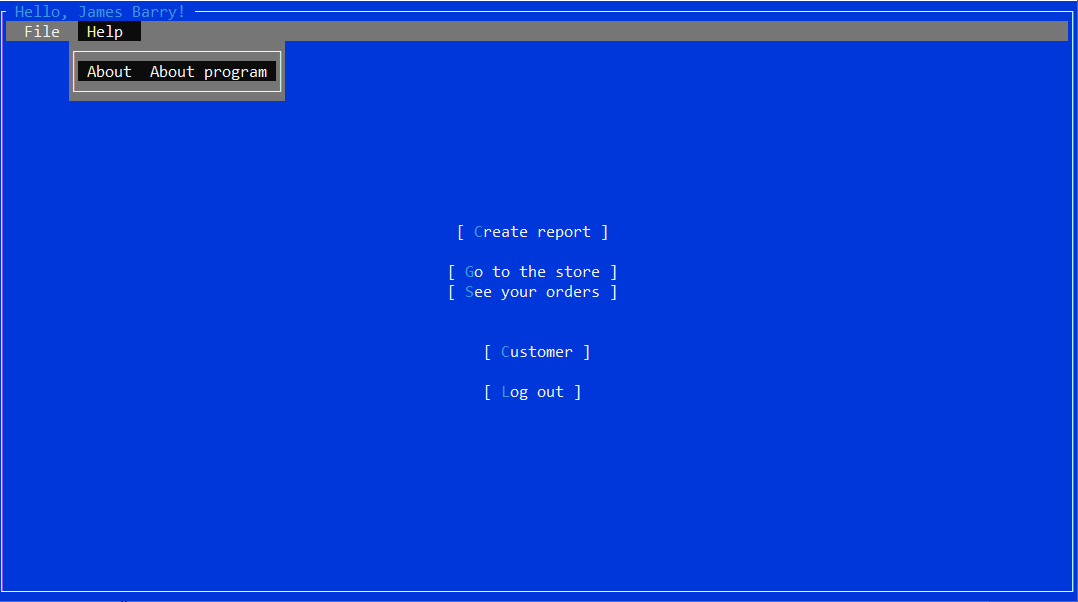


Рис. Меню.

# Висновки

Метою даного курсового проекту було розроблення десктопного додатку з подійно-орієнтованим інтерфейсом користувача для використання та керування електронним магазином товарів.

Розроблена система:

* забезпечує авторизований доступ до перегляду списку власних замовлень, списку товарів та редагування персональної інформації;
* дозволяє створювати замовлення з товарів, що є у наявності;
* має меню, що дає можливість експорту/імпорту, вийти з програми, отримати інформацію про неї.
* дає додаткові права користувачам ролі «модератор» створювати, редагувати товар, видаляти користувачів, генерувати звіт.

Особливу увагу під час розроблення даного програмного продукту було приділено роботі з базами даних та створенню подійно-орієнтованого інтерфейсу користувача.

Розробка виконана у повному обсязі, всі вимоги враховані, продукт протестовано.

Використання розробленого додатку дозволяє користувачу переглядати списки товарів, створювати замовлення та змінювати персональну інформацію.