|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ |
|  | Руководитель Учебной практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженкова М. Л.  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  М.П. |

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

приложения «Информационная система для магазина военной амуниции»

|  |
| --- |
|  |
|  |

2024

**СОДЕРЖАНИЕ**

[**1.** **Описание проекта:** 2](#_Toc181998824)

[**2.** **Основные функциональные возможности:** 2](#_Toc181998825)

[**3.** **Уточнение структуры данных:** 2](#_Toc181998826)

[**3.1** **Выделение сущностей** 3](#_Toc181998827)

[**3.2** **Построение логической и физической ER-диаграмм** 5](#_Toc181998828)

[**4.** **Форма представления данных** 6](#_Toc181998829)

[**4.1** **Контекстные диаграммы** 6](#_Toc181998830)

[**4.2** **Пользовательские интерфейсы** 11](#_Toc181998831)

[**4.2.1** **Пользовательские интерфейсы администратора** 12](#_Toc181998832)

[**4.2.2** **Пользовательские интерфейсы сотрудника** 19](#_Toc181998833)

[**4.2.3** **Пользовательские интерфейсы покупателя** 21](#_Toc181998834)

[**5.** **Разработка алгоритма решения задачи:** 23](#_Toc181998835)

[**6.** **Определение языка, структуры программы и требований к техническим средствам:** 37](#_Toc181998836)

[**7.** **Требования к техническим средствам:** 38](#_Toc181998837)

# **Описание проекта:**

Название проекта: Информационная система для магазина по продаже военной амуниции.

Цель проекта: Разработка информационной системы для автоматизации управления магазином военной амуниции.

# **Основные функциональные возможности:**

Система продажи военной амуниции должна обеспечивать следующие возможности:

**Администратору:**

• Добавлять, редактировать и удалять категории амуниции, товары, поставщиков, учётные записи пользователей и заказы для поддержания актуальной информации.

• Добавлять новые категории, товары, поставщиков и учётные записи для внесения новой информации в систему.

• Удалять устаревшие или некорректные данные о категориях, товарах, поставщиках, учётных записях и заказах для поддержания точности и актуальности данных.

**Сотруднику:**

• Просматривать состав заказов, чтобы видеть детали заказанных товаров.

• Изменять статус заказа для отображения актуального состояния заказов.

**Покупателю:**

• Просматривать свои заказы и их состав, чтобы получать информацию о текущих и завершённых заказах.

• Создавать новые заказы для покупки амуниции.

• Редактировать свою учётную запись для актуализации личных данных.

Система будет работать с базой данных, содержащей информацию о товарах, категориях товаров, поставщиках, заказах и учётных записях пользователей.

# **Уточнение структуры данных:**

В ходе уточнения данных были выполнены следующие задачи, показанные в Таблице 2.

Таблица 2 – Задачи для уточнения структуры данных.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ задачи** | **Название** | **Содержание** | **Результат** |
| 1 | Выделение сущностей | Выделение основных сущностей программы, а также их описание | Готовая таблица с сущностями и их описанием |
| 2 | Построение логической ER-диаграммы | Схематическое представление модели базы данных в виде логической ER-диаграммы | Готовая логическая ER-диаграмма |
| 3 | Построение диаграммы потоков данных | Определить, как будут передвигаться данные в программе; построить DFD диаграмму | Готовая диаграмма потоков данных (DFD) |

# **Выделение сущностей**

Таблица 3 – Набор сущностей и их атрибутов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Сущность** | **Свойство** | **Тип** | **Уникальность** | **Обязательность заполнения** | **Огра**  **ниче**  **ния** |
| User | UsersID | Int | Да | Да | РК |
| RoleId | Int |  | Да | FK |
| Phone | Text |  | Да |  |
| FullName | Text |  | Да |  |
| Password | Text | Да | Да |  |
| Email | Text |  | Да |  |
| Role | RolesID | Int | Да | Да | РК |
| Tittle | Text |  | Да |  |
| Order | OrdersID | Int | Да | Да | РК |
| ListID | Int | Да | Да | FK |
| StuffID | Int |  | Да | FK |
| BuyerID | Int |  | Да | FK |
| Date | Date |  | Да |  |
| Status | Text |  | Да |  |
| Sum | Double |  | Да |  |
| Adress | Text |  | Да |  |
| Index | Text |  | Да |  |
| Provider | ProviderID | Int | Да | Да | РК |
| Tittle | Text |  | Да |  |
| Country | Text |  |  |  |
| Category | CategoryID | Int | Да | Да | РК |
| Tittle | Text |  | Да |  |
| Product | ProductID | Int | Да | Да | РК |
| Tittle | Text |  | Да |  |
| CategoryID | Int |  | Да | FK |
| Color | Text |  | Да |  |
| Weight | Double |  | Да |  |
| Size | Text |  | Да |  |
| CountryOfMade | Text |  | Да |  |
| ProciderID | Int |  | Да | FK |
| Cost | Double |  | Да |  |
| ProductList | ListID | Int | Да | Да | РК |
| OrderID | Int |  | Да | FK |
| ProductID | Int |  | Да | FK |
| Count | Int |  | Да |  |

Связи между сущностями:

«Клиент» – «Заказ»: один ко многим, у каждого заказа определен конкретный клиент, но один клиент может оформлять несколько заказов.

«Роль» – «Клиент»: один ко многим, каждый пользователь имеет одну роль, но одна роль, может быть, у нескольких пользователей.

«Категория» – «Продукт»: один ко многим, каждый продукт относится к одной категории, но одна категория может включать несколько продуктов.

«Поставщик» – «Продукт»: один ко многим, каждый продукт поставляется одним поставщиком, но один поставщик может поставлять несколько продуктов.

«Продукт» – «Список продуктов»: один ко многим, каждый элемент списка продуктов относится к одному продукту, но один продукт может находиться в нескольких списках.

«Заказ» – «Список продуктов»: один ко многим, каждый элемент списка продуктов относится к одному заказу, но один заказ может содержать несколько продуктов.

# **Построение логической и физической ER-диаграмм**

Логическая ER-диаграмма показана на рисунке 2.

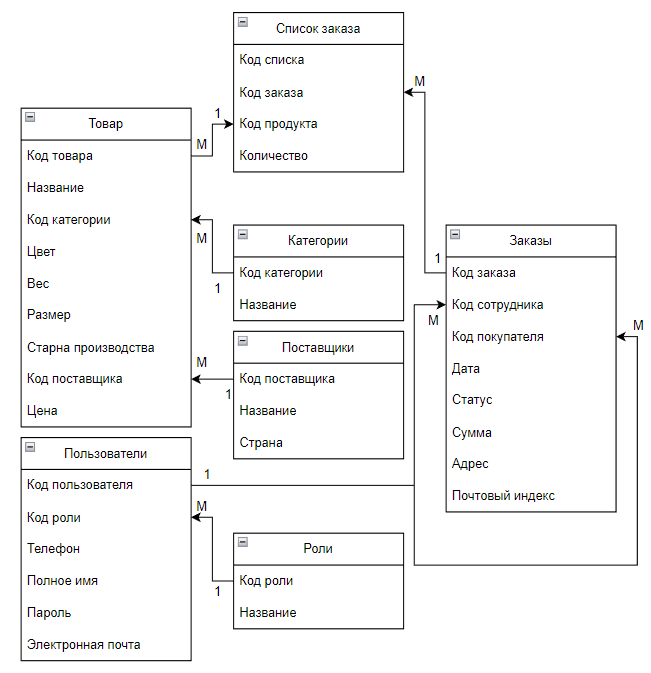


Рисунок 2 – Логическая модель базы данных

Физическая ER-диаграмма показана на рисунке 3.

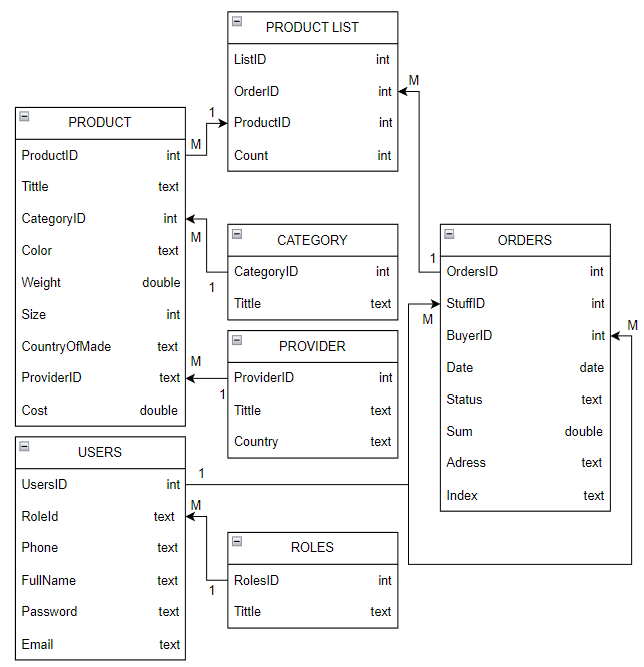


Рисунок 3 – Физическая модель базы данных

# **Форма представления данных**

# **Контекстные диаграммы**

**ЗДЕСЬ БУДУТ IDEF0, ИЗВИНИТЕ НЕ УСПЕЛИ (**

# **Пользовательские интерфейсы**

На рисунках ниже предоставлены прототипы ИС для магазина военной амуниции.

При входе в приложение пользователя встречает окно авторизации, прототип (см. рис. 4) показывает, что в окне должны быть поля ввода логина и пароля, также должна быть информация о полях. Внизу окна должна быть кнопка «Войти».

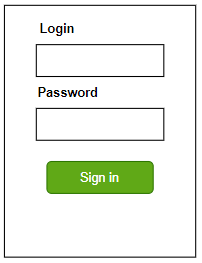


Рисунок 3 – Прототип окна авторизации

После входа пользователя встречает окно той роли, какая у его учётной записи, то есть после окна авторизации программа должна сразу понимать, что за пользователь перед ней, это либо администратор, сотрудник или покупатель.

# **Пользовательские интерфейсы администратора**

В главном меню администратора должны быть кнопки просмотра «Продуктов», «Категорий», «Поставщиков», «Пользователей» и «Заказов». При нажатии каждой из кнопок должно открываться окно для просмотра соответствующей информации. В самом низу окна должна быть кнопка выхода из учётной записи (см. рис. 5). Вверху окна должна быть текстовая надпись, в которой указано то, что это администратор и его имя.

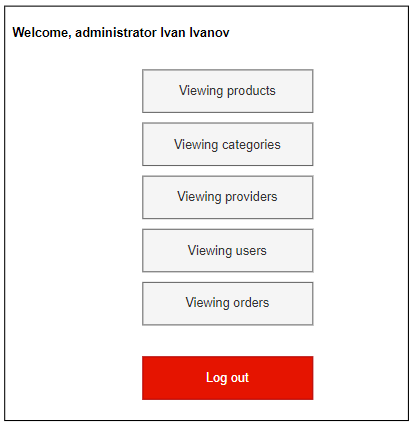


Рисунок 4 – Прототип главного окна администратора

При нажатии кнопки «Viewing products» должно открыться окно, в котором храниться табличная информация по всем продуктам, а также с возможностями добавления, редактирования, обновления и удаления данной информации (см. рис. 5). На прототипе представлены 5 кнопок, при нажатии на «Add» должно открыться окно добавления товара (см. рис. 6), при нажатии на «Edit» должно открываться окно редактирования товара (см. рис. 7), при нажатии на «Delete» должно появляться окно подтверждения, после утвердительного ответа по поводу удаления записи, запись должна быть удалена. При нажатии на кнопку «Refresh» должны обновляться данные. Ниже всех кнопок должна располагаться кнопка выхода из данного окна.

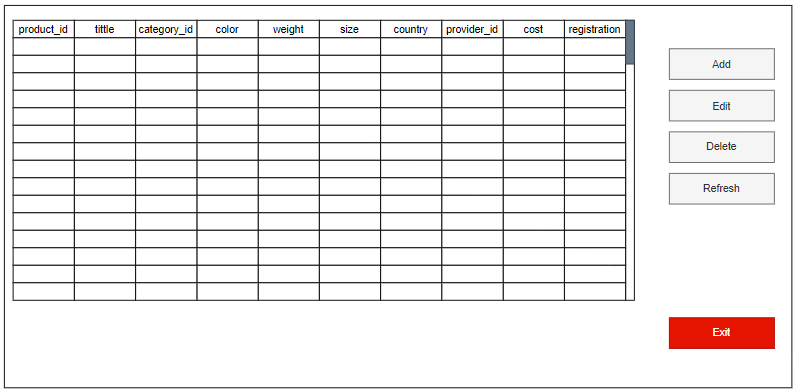


Рисунок 5 – Прототип окна просмотра товаров

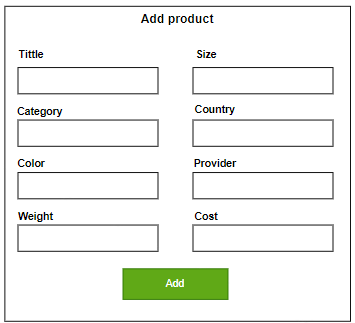


Рисунок 6 – Прототип окно добавления товара

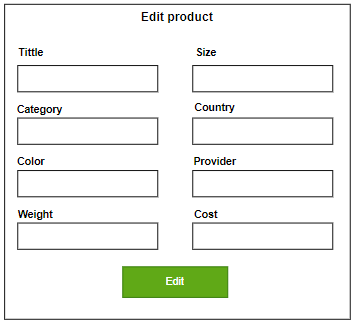


Рисунок 7 – Прототип окна редактирования товара

При нажатии кнопки «Viewing categories» должно открыться окно, в котором храниться табличная информация по всем категориям, а также с возможностями добавления, редактирования, обновления и удаления данной информации (см. рис. 8). На прототипе представлены 5 кнопок, при нажатии на «Add» должно открыться окно добавления категории (см. рис. 9), при нажатии на «Edit» должно открываться окно редактирования категории (см. рис. 10), при нажатии на «Delete» должно появляться окно подтверждения, после утвердительного ответа по поводу удаления записи, она должна быть удалена. При нажатии на кнопку «Refresh» должны обновляться данные. Ниже всех кнопок должна располагаться кнопка выхода из данного окна.

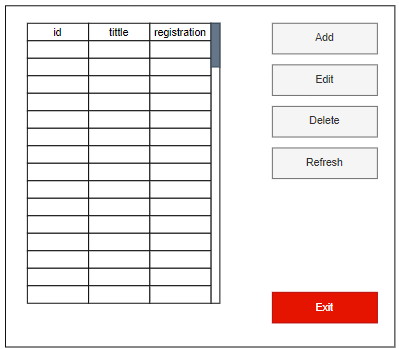


Рисунок 8 – Прототип окна просмотра данных о категориях



Рисунок 9 – Прототип окна добавления категории

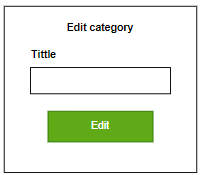


Рисунок 10 – Прототип окна редактирования категории

При нажатии кнопки «Viewing providers» должно открыться окно, в котором храниться табличная информация по всем поставщикам, а также с возможностями добавления, редактирования, обновления и удаления данной информации (см. рис. 11). На прототипе представлены 5 кнопок, при нажатии на «Add» должно открыться окно добавления поставщика (см. рис. 12), при нажатии на «Edit» должно открываться окно редактирования поставщика (см. рис. 13), при нажатии на «Delete» должно появляться окно подтверждения, после утвердительного ответа по поводу удаления записи, она должна быть удалена. При нажатии на кнопку «Refresh» должны обновляться данные. Ниже всех кнопок должна располагаться кнопка выхода из данного окна.

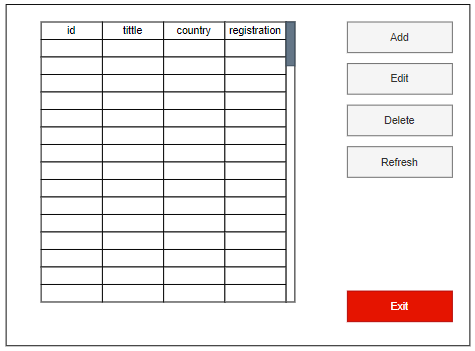


Рисунок 11 – Прототип окна просмотра данных о поставщиках

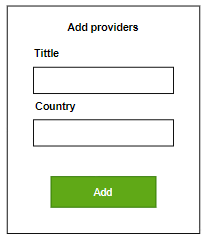


Рисунок 12 – Прототип окна добавления поставщика

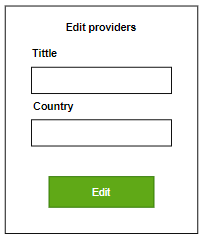


Рисунок 13 – Прототип окна редактирования поставщика

При нажатии кнопки «Viewing users» должно открыться окно, в котором храниться табличная информация по всем пользователям, а также с возможностями добавления, редактирования, обновления и удаления данной информации (см. рис. 14). На прототипе представлены 5 кнопок, при нажатии на «Add» должно открыться окно добавления пользователя (см. рис. 15), при нажатии на «Edit» должно открываться окно редактирования пользователя (см. рис. 16), при нажатии на «Delete» должно появляться окно подтверждения, после утвердительного ответа по поводу удаления записи, она должна быть удалена. При нажатии на кнопку «Refresh» должны обновляться данные. Ниже всех кнопок должна располагаться кнопка выхода из данного окна.

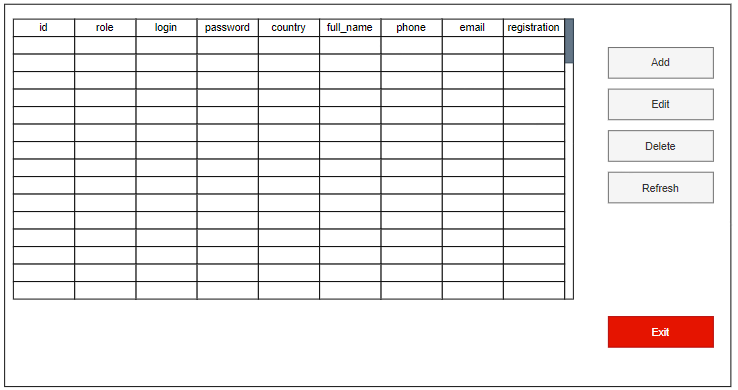


Рисунок 14 – Прототип окна просмотра данных о пользователях

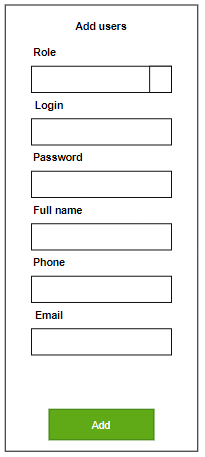


Рисунок 15 – Прототип окна добавления пользователя

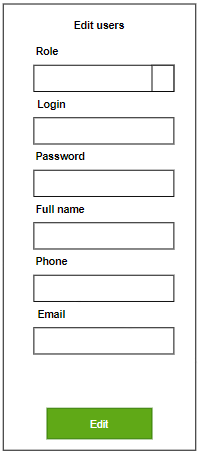


Рисунок 16 – Прототип окна редактирования пользователя

При нажатии кнопки «Viewing orders» должно открыться окно, в котором храниться табличная информация по всем заказам, а также с возможностями редактирования, обновления и удаления данной информации (см. рис. 17). На прототипе представлены 4 кнопки, при нажатии на «Edit» должно открываться окно редактирования заказа, но только общей суммы, адреса и почтового индекса (см. рис. 18), при нажатии на «Delete» должно появляться окно подтверждения, после утвердительного ответа по поводу удаления записи, она должна быть удалена. При нажатии на кнопку «Refresh» должны обновляться данные. Ниже всех кнопок должна располагаться кнопка выхода из данного окна.

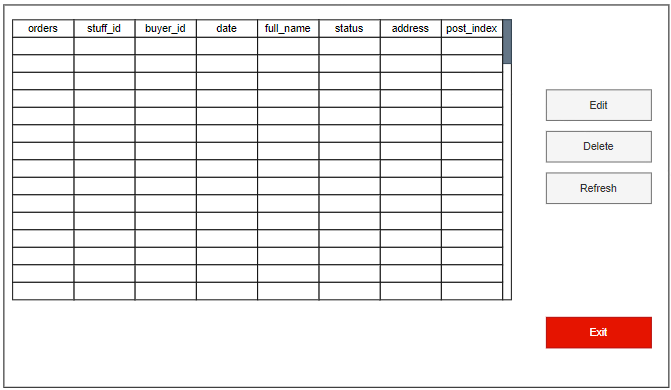


Рисунок 17 – Прототип окна просмотра данных о заказах

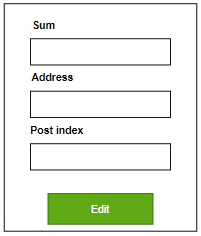


Рисунок 18 – Прототип окна редактирования данных заказа

# **Пользовательские интерфейсы сотрудника**

В главном меню сотрудника должна быть таблица с заказами, в ней должны быть названия столбцов (см. рис. 19). Также в этой таблице должна быть возможность выбирать строку с заказом, ещё нужно, чтобы была возможность просматривать все данные таблицы, можно использовать слайдер, как указанно на рисунке. В окне должны присутствовать кнопки для смены статуса заказа «Взять в обработку», «Собран», «Отправлен». Также кнопки «Обновить», «Подробнее» и «Выйти из учётной записи». Также должна присутствовать информативная надпись о пользователе, о том, что его роль – сотрудник и его имя.

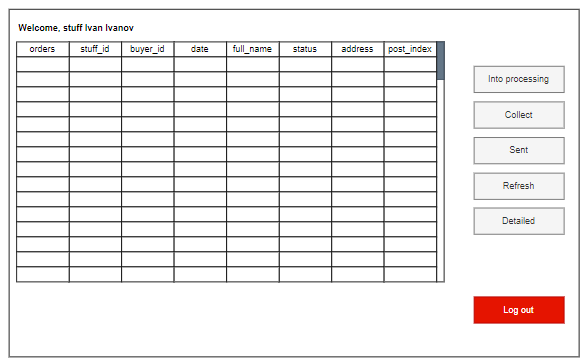


Рисунок 19 – Прототип главного окна сотрудника

При нажатии кнопки «Detailed» должно открыться окно, в котором храниться табличная информация по составу выбранного заказа (см. рис. 20). На прототипе представлена лишь одна кнопка и она позволяет закрыть окно и вернуться к главному окну сотрудника.

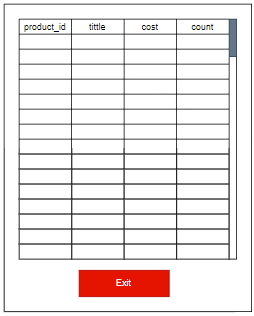


Рисунок 20 – Прототип окна просмотра состава заказа

# **Пользовательские интерфейсы покупателя**

В главном меню покупателя должны быть кнопки «Просмотр заказов», «Создать заказ», «Редактировать учётную запись» и «Выйти из учётной записи» (см. рис. 7). Вверху окна должна быть текстовая надпись, в которой указано то, что это покупатель и его имя.

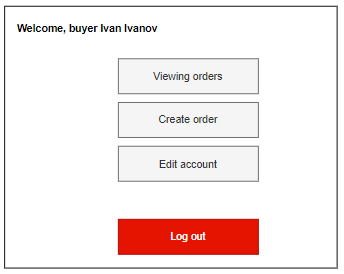


Рисунок 21 – Прототип главного окна покупателя

При нажатии кнопки «Viewing orders» должно открыться окно, в котором храниться табличная информация по всем заказам пользователя (см. рис. 22). На прототипе представлены лишь две кнопки первая «Detailed» позволяет посмотреть подробную информацию по выбранному заказу (см. рис. 23), вторая должна закрывать окно и возвращаться на главное меню покупателя.

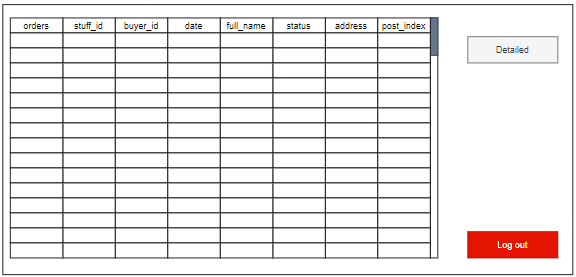


Рисунок 22 – Прототип окна просмотра данных о заказах пользователя

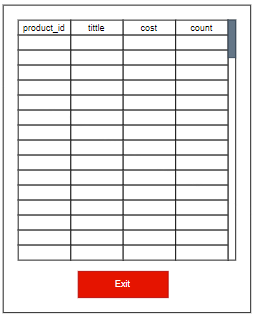


Рисунок 23 – Прототип окна просмотра состава заказа

При нажатии кнопки «Create order» должно открыться окно, в котором пользователь может создать новый заказ (см. рис. 24). В окне должна быть табличная информация по всем заказам, поле (корзина) в которое пользователь добавляет товары, также должны присутствовать поля для ввода адреса и почтового индекса. Ещё 5 кнопок «Add in order» позволяет добавить выбранный товар в корзину, «Delete» позволяет удалить выбранный товар из корзины, «Refresh» позволяет обновить информацию по списку товаров, «Complete» позволяет создать заказ и пустить его в обработку.

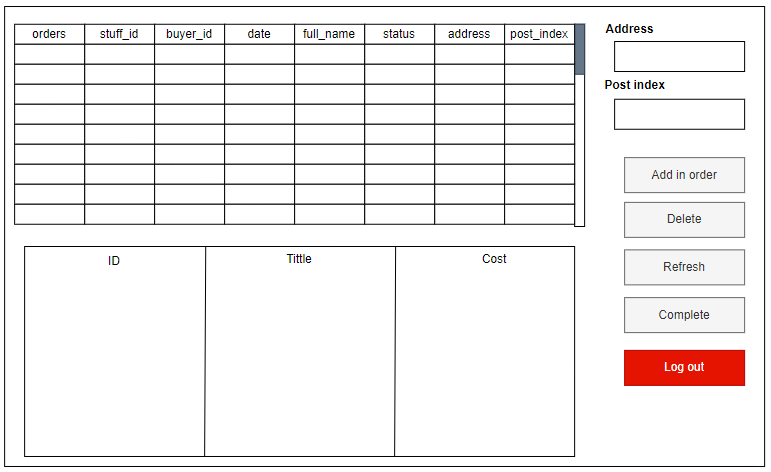


Рисунок 24 – Прототип окна создания заказа

При нажатии кнопки «Edit account» должно открыться окно, в котором пользователь сможет изменить данные своего аккаунта. Он может изменить логин, пароль, имя, телефон и электронную почту (см. рис. 25).

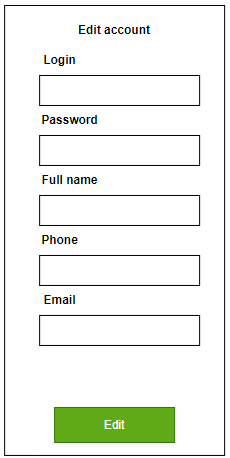


Рисунок 25 – Прототип окна редактирования аккаунта

# **Разработка алгоритма решения задачи:**

**ЗДЕСЬ БУДУТ БЛОК-СХЕМЫ АЛГОРИТМОВ, ИЗВИНИТЕ НЕ УСПЕЛИ (**

Пользовательские сценарии представлены в таблице 2.

Таблица 2 – Пользовательские сценарии

|  |  |  |
| --- | --- | --- |
| **№** | **Пользователь** | **Система** |
| **1** | Администратор заполнил поля для ввода новой категории корректными данными и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Сохраняет данные в БД.  3. Переходит на страницу «Категории», на которой отображены все ранние занесённые категории. |
| **2** | Администратор заполнил поля для ввода нового товара корректными данными и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Сохраняет данные в БД.  3. Переходит на страницу «Товары», на которой отображены все ранние занесённые товары. |
| **3** | Администратор заполнил поля для ввода нового поставщика корректными данными и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Сохраняет данные в БД.  3. Переходит на страницу «Поставщики», на которой отображены все ранние занесённые поставщики. |
| **4** | Администратор заполнил поля для ввода новой учетной записи пользователя корректными данными и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Сохраняет данные в БД.  3. Переходит на страницу «Пользователи», на которой отображены все ранние занесённые пользователи. |
| **5** | Администратор заполнил поля для редактирования категории и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Изменяет данные в БД.  3. Переходит на страницу «Категории», на которой отображены все ранние занесённые категории. |
| **6** | Администратор заполнил поля для редактирования товара и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Изменяет данные в БД.  3. Переходит на страницу «Товары», на которой отображены все ранние занесённые товары. |
| **7** | Администратор заполнил поля для редактирования поставщика и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Изменяет данные в БД.  3. Переходит на страницу «Поставщики», на которой отображены все ранние занесённые поставщики. |
| **8** | Администратор заполнил поля для редактирования учетной записи пользователя и нажал на кнопку «Сохранить» | 1. Система получает введённые данные.  2. Изменяет данные в БД.  3. Переходит на страницу «Пользователи», на которой отображены все ранние занесённые пользователи. |
| **9** | Администратор нажимает на кнопку «Удалить категорию» | 1. Система отправляет подтверждающее сообщение.  1.1. Если ответ «Да»:     А) Получает ID удаляемой категории.     Б) Находит по полученному ID запись в БД.     В) Удаляет найденную запись.  1.2. Если ответ «Нет»:     А) См. пункт 2.  2. Переходит на страницу «Категории». |
| **10** | Администратор нажимает на кнопку «Удалить товар» | 1. Система отправляет подтверждающее сообщение.  1.1. Если ответ «Да»:     А) Получает ID удаляемого товара.     Б) Находит по полученному ID запись в БД.     В) Удаляет найденную запись.  1.2. Если ответ «Нет»:     А) См. пункт 2.  2. Переходит на страницу «Товары». |
| **11** | Администратор нажимает на кнопку «Удалить поставщика» | 1. Система отправляет подтверждающее сообщение.  1.1. Если ответ «Да»:     А) Получает ID удаляемого поставщика.     Б) Находит по полученному ID запись в БД.     В) Удаляет найденную запись.  1.2. Если ответ «Нет»:     А) См. пункт 2.  2. Переходит на страницу «Поставщики». |
| **12** | Администратор нажимает на кнопку «Удалить пользователя» | 1. Система отправляет подтверждающее сообщение.  1.1. Если ответ «Да»:     А) Получает ID удаляемого пользователя.     Б) Находит по полученному ID запись в БД.     В) Удаляет найденную запись.  1.2. Если ответ «Нет»:     А) См. пункт 2.  2. Переходит на страницу «Пользователи». |
| **13** | Сотрудник выбирает конкретный заказ для просмотра его состава | 1. Система получает запрос на просмотр заказа.  2. Извлекает данные о составе заказа из БД.  3. Отображает информацию о составе заказа на экране. |
| **14** | Сотрудник выбирает заказ и изменяет его статус на нужный | 1. Система получает запрос на изменение статуса заказа.  2. Изменяет статус заказа в БД на нужный.  3. Отображает обновлённый статус на странице заказов. |
| **15** | Покупатель заходит на страницу «Мои заказы» и выбирает конкретный заказ для просмотра | 1. Система получает запрос на просмотр заказов.  2. Извлекает данные о заказах из БД.  3. Отображает список заказов покупателя и их состав на экране. |
| **16** | Покупатель заполнил все необходимые поля для создания заказа и нажал на кнопку «Создать заказ» | 1. Система получает введённые данные.  2. Автоматически генерирует уникальный ID для нового заказа.  3. Сохраняет данные заказа в БД.  4. Отображает уведомление об успешном создании заказа. |
| **17** | Покупатель заходит на страницу своей учётной записи и вносит изменения, после чего нажимает кнопку «Сохранить» | 1. Система получает введённые данные.  2. Изменяет данные пользователя в БД.  3. Отображает уведомление об успешном обновлении информации. |

# **Определение языка, структуры программы и требований к техническим средствам:**

**Сервер:** Python используется для создания сервера, так как он обеспечивает гибкость, обширную экосистему библиотек и легкость работы с API, что позволяет эффективно обрабатывать данные и взаимодействовать с базой данных.

**Фреймворк и графическая библиотека**: основное приложение разрабатывается на языке C# с использованием Windows Forms (WinForms) для построения интуитивно понятного и удобного графического интерфейса, подходящего для настольных систем.

**Хранение данных**: для хранения данных используется SQLite, поскольку она является легковесной и легко интегрируется с Python и C#. SQLite позволяет надёжно хранить данные и обеспечивает высокую производительность при работе с умеренными объемами данных, что идеально подходит для целей данной информационной системы.

# **Требования к техническим средствам:**

Минимальные требования для информационной системы Military Store:

* Операционная система: Windows 7 и выше;
* Оперативная память (ОЗУ): 4 ГБ;
* Жесткий диск: 5 ГБ;
* Процессор: Intel Core i3 или аналогичный;
* Разрешение экрана: минимум 1024x768 пикселей для удобного отображения интерфейса;
* Сетевое подключение: стабильное соединение с высокой скоростью передачи данных для корректного подключения и работы базы данных.