**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ЗАВОЛЖСКИЙ АВТОМОТОРНЫЙ ТЕХНИКУМ»**

Специальность 09.02.07 «Информационные системы и программирование»

**Выпускная квалификационная работа**

**Пояснительная записка**

По дисциплине **Технология разработки и защиты баз данных**

Тема: Разработка автоматизированной информационной системы по учёту проданных товаров пекарни «Круглова» на базе платформы .NET.

Число листов курсового проекта (прописью)

Число страниц пояснительной записки

Студент Кожевников Сергей Романович Группа ИС-21Б

Руководитель Харчева Ольга Евгеньевна

Дата защиты Оценка

2024 - 2025 учебный год

Содержание

Введение 3

1. Постановка задачи 5
2. Проектирование 6
   1. Обоснование выбора среды разработки 6
   2. Выбор способа организации входных и выходных данных 9
   3. Проектирование моделей данных 14
   4. Алгоритм представления задачи 19
      1. Детальное проектирование пользовательского интерфейса 19
      2. Описание основных программных модулей 24
   5. Применяемые меры защиты информации 25
3. Тестирование
4. Экономическое обоснование

Заключение 37

Литература 38

**Введение**

Пекарня – небольшое предприятие по выпечке и реализации хлебобулочных и кондитерских изделий, как правило, также реализующее их на месте.

Продажи выпечки почти не зависят от сезона, продукция пользуется стабильным спросом круглый год.

Выпечка – это лучшее лекарство от стресса. Как показывают данные аналитических обзоров, в кризис прибавляют в продажах сладости и кондитерские изделия до 20–30%

Пекарня «Круглова» имеет более 10 филиалов, каждый из которых осуществляет продажу различных категорий выпечки, кондитерских изделий и других товаров. Для эффективной торговли требуется инструмент для оперативного оформления заказов клиентов, ведения информации об имеющихся товарах и анализа информации о продажах.

Автоматизированная информационная система - программно-аппаратный комплекс для авторизации деятельности организации, который обеспечивает хранение, передачу и обработку информации.

Автоматизированные информационные системы по ведению базы данных широко применяются в современной торговле. Такие системы упрощают работу сотрудников, а также повышают эффективность поиска и обработки информации.

Целью выпускной квалификационной работы является создание автоматизированной информационной системы по учёту проданных товаров пекарни «Круглова» на базе платформы .NET.

В процессе проектирования нужно решить следующие задачи:

* провести анализ существующих систем;
* сформулировать требования к программе;
* построить структуру базы данных;
* выбрать среду разработки;
* реализовать программный продукт;
* протестировать программный продукт;
* разработать техническую и эксплуатационную документацию

Программа может использоваться для учёта товаров и формирования заказов.

1. **Постановка задачи**

Разработать и программно реализовать автоматизированную информационную систему (АИС) по учету проданных товаров пекарни «Круглова». В программе должны соблюдаться принципы информационной закрытости, модульности и иерархической организации.

Предметная область: системы пекарни «Круглова» содержит необходимую информацию о продукции и расценках пекарни.

АИС по учету проданных товаров пекарни «Круглова» предназначена для ведения базы данных, оформления и получения заказов пекарни.

Автоматизированная информационная система по учету проданных товаров пекарни «Круглова» должна иметь форму авторизации (администратор, менеджер, сотрудник), в режиме администратора должна быть предусмотрена возможность создания нового пользователя.

Автоматизированная информационная система пекарни должна иметь возможность ведения и редактирования базы данных (добавление, удаление, редактирование, сохранение).

Программа позволяет оформить и получить заказ, принять оплату и распечатать чек об оплате (в MS Excel 2007), возможность получить скидку (на определенный товар), ведет учет остатков продуктов.

Пользователь должен иметь возможность отсортировать информацию о товарах (по возрастанию и убыванию) по названию или по стоимости. Возможность отфильтровать данные. Реализовать живой поиск информации.

Предусмотреть возможность составления отчетов (в MS Exсel 2007).

Автоматизированная информационная система пекарни должна иметь возможность импорта и экспорта данных базы.

Язык программирования С#. База данных составляется в MySQL.

1. **Проектирование**
   1. **Обоснование выбора среды разработки**

Для создания данного программного продукта можно использовать следующие среды разработки:

* Visual Studio 2022 Community - это интегрированная среда разработки (IDE), разработанная компанией Microsoft. Она используется для разработки компьютерных программ. Встроенные языки включают C, C++, C#. Visual Studio использует платформы разработки программного обеспечения Microsoft, включая Windows Forms и Windows Presentation Foundation (WPF). Visual Studio включает редактор кода, поддерживающий IntelliSense (компонент автоматического завершения кода), а также рефакторинг кода. Visual Studio включает отладчик, который работает как на уровне исходного кода, так и на уровне машинного кода. Он работает как с управляемым кодом, так и с собственным кодом и может использоваться для отладки приложений, написанных на любом языке, поддерживаемом Visual Studio. Кроме того, он может подключаться к запущенным процессам, отслеживать и отлаживать их. Отладчик Visual Studio позволяет устанавливать точки останова (которые позволяют временно приостановить выполнение в определённой точке). Visual Studio включает в себя множество визуальных конструкторов, помогающих в разработке приложений. К этим инструментам относятся конструктор Windows Forms и конструктор WPF;
* Rider– кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Rider поддерживает множество языков .NET-разработки, включая C#. Rider поддерживает модульные тесты в приложениях.NET Framework,.NET Core и Mono. Rider предоставляет интеграцию с Git. В специальном окне инструментов можно отслеживать локальные и внешние изменения в репозитории, управлять списками изменений, откладывать изменения, сравнивать файлы, делать коммиты и вливать локальные изменения в удаленный репозиторий. Встроенный инструмент визуального слияния позволяет с легкостью разрешать конфликты. Кроме того, Rider предупредит, если файлы были изменены после последней синхронизации;
* MonoDevelop - это интегрированная среда разработки с открытым исходным кодом, созданная для языков .NET. Она использует Mono и .NET Framework для разработки программного обеспечения. С помощью MonoDevelop можно эффективно создавать настольные приложения для различных операционных систем.

Исходя из вышеперечисленных характеристик сред разработки выбираем Visual Studio 2022 Community, т.к. данная интегрированная среда разработки имеет удобный отладчик и визуальный конструктор для разработки дизайна приложения.

В состав выбранной интегрированной среды разработки входят следующие языки программирования:

* C# - высокоуровневый объектно-ориентированный язык программирования общего назначения. Разработан как язык разработки приложений для платформы.NET Framework и .NET Core. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, исключения, комментарии в формате XML.C# предоставляет удобные классы для построения графических интерфейсов по технологии Windows Forms;
* C++ - компилируемый, статически типизированный язык программирования общего назначения. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков.
* C - компилируемый, статически типизированный язык программирования общего назначения. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C не является объектно-ориентированным языком программирования. Основные особенности С:
* простая языковая база, из которой в стандартную библиотеку вынесены многие существенные возможности, вроде математических функций или функций работы с файлами;
* ориентация на процедурное программирование;
* система типов, предохраняющая от бессмысленных операций;
* использование препроцессора для абстрагирования однотипных операций;
* доступ к памяти через использование указателей;
* небольшое число ключевых слов;
* наличие указателей на функции и статические переменные;
* области видимости имён;
* структуры и объединения - определяемые пользователем собирательные типы данных, которыми можно манипулировать как одним целым.

В то же время в C отсутствуют:

* вложенные функции;
* прямое возвращение нескольких значений из функций;
* средства автоматического управления памятью;
* встроенные средства объектно-ориентированного программирования;
* средства функционального программирования.

Исходя из вышеперечисленных характеристик языков программирования выбираем C#, т.к. данный язык программирования изначально разработан как язык разработки приложений для платформы .NET, а также предоставляет удобные классы для построения графических интерфейсов по технологии Windows Forms.

* 1. **Выбор способа организации входных и выходных данных**

Для организации входных и выходных данных необходима база данных

* MySql - свободная реляционная СУБД. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты. MySQL имеет API и коннекторы для языков C, C++, библиотеки для языков платформы.NET;
* SQLite - компактная встраиваемая СУБД. Представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа;
* MongoDB - документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Система поддерживает ad-hoc-запросы: они могут возвращать конкретные поля документов и пользовательские JavaScript-функции. Поддерживается поиск по регулярным выражениям. Также можно настроить запрос на возвращение случайного набора результатов. Имеется поддержка индексов.

Исходя из вышеперечисленных характеристик баз данных выбираем MySql, т.к. данная СУБД является реляционной и работает по принципу «Клиент-сервер», что позволяет организовать централизованное хранилище данных.

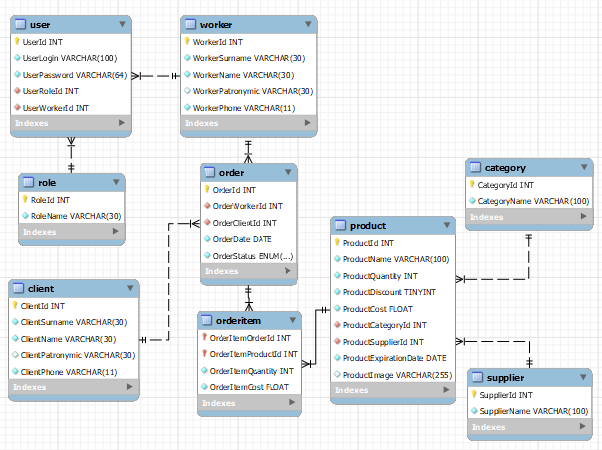


Рисунок 2.1 – ER-диаграмма

База данных имеет 9 таблиц, каждая имеет связи в соответствии с рисунком 2.1

Структура таблиц базы данных представлена таблицами 2.1-2.9

Таблица 2.1 - Структура таблицы продуктов

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор продукта | ProductId | INT | Обязательное, уникальное |
| Название продукта | ProductName | VARCHAR | Обязательное, до 100 символов |
| Количество продукта | ProductQuantity | INT | Обязательное |
| Скидка на продукт | ProductDiscount | TINYINT | Обязательное |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Стоимость продукта | ProductCost | FLOAT | Обязательное, точность 2 |
| Идентификатор категории продукта | ProductCategoryId | INT | Обязательное, внешний ключ поля CategoryId |
| Идентификатор поставщика продукта | ProductSupplierId | INT | Обязательное, внешний ключ поля SupplierId |
| Срок годности продукта | ProductExpirationDate | DATE | Обязательное |
| Название изображения продукта | ProductImage | VARCHAR | До 100 символов |

Таблица 2.2 - Структура таблицы пользователей

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор пользователя | UserId | INT | Обязательное, уникальное |
| Логин пользователя | UserLogin | VARCHAR | Обязательное, до 100 символов |
| Hash пароля пользователя | UserPassword | VARCHAR | Обязательное, до 32 символов |
| Идентификатор роли пользователя | UserRoleId | INT | Обязательное, внешний ключ поля RoleId |
| Идентификатор работника, относящегося к пользователю | UserWorkerId | INT | Обязательное, внешний ключ поля WorkerId |

Таблица 2.3 - Структура таблицы работников

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| 1 | 2 | 3 | 4 |
| Идентификатор работника | WorkerId | INT | Обязательное, уникальное |
| Фамилия работника | WorkerSurname | VARCHAR | Обязательное, до 30 символов |

Продолжение таблицы 2.3

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Имя работника | WorkerName | VARCHAR | Обязательное, до 30 символов |
| Отчество работника | WorkerPatronymic | VARCHAR | До 30 символов |
| Телефон работника | WorkerPhone | VARCHAR | Обязательное, 11 символов |

Таблица 2.4 - Структура таблицы позиций заказов

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор заказа | OrderItemOrderId | INT | Обязательное, уникальное при не уникальном OrderItemProductId, внешний ключ поля OrderId |
| Идентификатор продукта | OrderItemProductId | INT | Обязательное, уникальное при не уникальном OrderItemOrderId, внешний ключ поля ProductId |
| Количество продукта | OrderItemQuantity | INT | Обязательное |
| Стоимость продукта | OrderItemCost | FLOAT | Обязательное |

Таблица 2.5 - Структура таблицы категорий

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор категории | CategoryId | INT | Обязательное, уникальное |
| Наименование категории | CategoryName | VARCHAR | Обязательное, до 100 символов |

Таблица 2.6 - Структура таблицы заказов

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор заказа | OrderId | INT | Обязательное, уникальное |
| Идентификатор пользователя сформировавшего заказ | OrderUserId | INT | Обязательное, внешний ключ поля UserId |
| Дата формирования заказа | OrderDate | DATE | Обязательное |
| Статус заказа | OrderStatus | ENUM(«Новый», «Проведён») | Обязательное |

Таблица 2.7 - Структура таблицы ролей

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор роли | RoleId | INT | Обязательное, уникальное |
| Наименование роли | RoleName | VARCHAR | Обязательное, до 30 символов |

Таблица 2.8 - Структура таблицы поставщиков

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор поставщика | SupplierId | INT | Обязательное, уникальное |
| Наименование поставщика | SupplierName | VARCHAR | Обязательное, до 100 символов |

Таблица 2.9 - Структура таблицы клиентов

|  |  |  |  |
| --- | --- | --- | --- |
| Название столбца | Название столбца в БД | Тип данных | Ограничения |
| Идентификатор клиента | WorkerId | INT | Обязательное, уникальное |
| Фамилия клиента | WorkerSurname | VARCHAR | Обязательное, до 30 символов |
| Имя клиента | WorkerName | VARCHAR | Обязательное, до 30 символов |
| Отчество клиента | WorkerPatronymic | VARCHAR | До 30 символов |
| Телефон клиента | WorkerPhone | VARCHAR | Обязательное, 11 символов |

* 1. **Проектирование моделей данных**

При проектировании моделей данных были построены диаграмма вариантов использования и диаграмма деятельности.

Диаграмма вариантов использования - диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей

Диаграмма вариантов использования представлена в соответствии с рисунком 2.2.

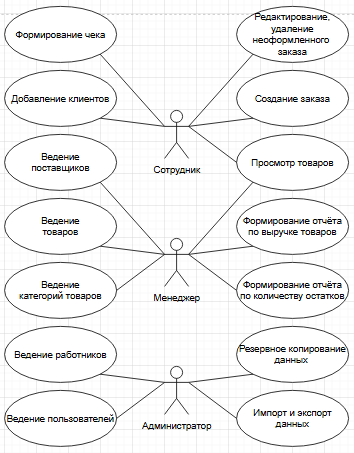


Рисунок 2.2 – Диаграмма вариантов использования

Сценарий диаграммы вариантов использования - специально написанный текст, который описывает поведение моделируемой системы в форме последовательности выполняемых действий актеров и самой системы.

Сценарий диаграммы вариантов использования представлен в соответствии с таблицей 2.10.

Таблица 2.10 - Варианты использования

|  |  |
| --- | --- |
| Варианты использования | Создание заказа |
| Актёры | Сотрудник, менеджер, администратор |
| Краткое описание | Сотрудник просматривает товары, сотрудник формирует заказы, сотрудник ведёт информацию о клиентах менеджер ведёт информацию о товарах, поставщиках и категориях, менеджер формирует отчёты по количеству остатков, менеджер формирует отчёты по выручке товаров, администратор ведёт информацию о работниках и пользователях приложения, администратор выполняет резервное копирование данных, администратор выполняет импорт и экспорт данных |
| Цель | Продажа товаров |
| Тип | Базовый |
| Ссылки на другие варианты использования | Просмотр товаров, ведение информации о товарах, ведение информации о поставщиках, ведение информации о категориях, ведение информации о клиентах, ведение информации о работниках, ведение информации о пользователях, формирование отчёта по количеству остатков, формирование отчёта по выручке товаров, резервное копирование данных, импорт и экспорт данных |

Последовательность действий - это список действий, которые выполняются одно за другим. Вся последовательность считается одним атомарным действием, поэтому ее нельзя прервать извне, но можно присоединять к переходам или шагам при взаимодействии.

Последовательность действий представлена в соответствии с таблицей 2.11.

Таблица 2.11 – Последовательность действий

|  |  |
| --- | --- |
| Действия актёров | Отклик системы |
| 1 | 2 |
| 1. Сотрудник авторизуется в системе  Искл. 1. Введён неверный логин или пароль | 2. Открывается главное меню в соответствии с ролью сотрудника  3. Очистка полей «Логин» и «Пароль», вывод сообщения об ошибке ввода, отображение captcha |
| 4. Сотрудник открывает список товаров | 5. Отображается информация о товарах |
| 6. Сотрудник выбирает товар из списка и добавляет к заказу | 7. Создаётся заказ, отображается кнопка для перехода к заказу |
| 8. Сотрудник открывает заказ | 9. Отображаются товары, добавленные в заказ |
| 10. Сотрудник формирует заказ | 11. Данные о заказе сохраняются в базе данных, отображается сообщение о создании заказа |
| 12. Сотрудник открывает список клиентов | 13. Отображается информация о клиентах |
| 14. Сотрудник добавляет клиента | 15. Данные о клиенте сохраняются в базе данных, отображается сообщение об успешном добавлении |
| 16. Сотрудник открывает список заказов | 17. Отображается информация о заказах |
| 18. Сотрудник открывает заказ | 19. Отображаются товары в заказе и информация о заказе |
| 20. Сотрудник редактирует заказ | 21. Изменения о заказе сохраняются в базе данных, отображается сообщение об успешном редактировании |
| 22. Менеджер открывает список категорий товаров | 23. Отображается информация о категориях товаров |
| 24. Менеджер открывает список поставщиков | 25. Отображается информация о поставщиках |
| 26. Менеджер открывает список товаров | 27. Отображается информация о товарах |

Продолжение таблицы 2.11

|  |  |
| --- | --- |
| 1 | 2 |
| 28. Менеджер формирует отчёт по выручке товаров | 29. Открывается отчёт по выручке товаров в Excel |
| 30. Менеджер формирует отчёт по количеству остатков | 31. Открывается отчёт по количеству остатков в Excel |
| 32. Администратор создаёт резервную копию БД | 33. Формируется резервная копия данных БД |
| 34. Администратор импортирует данные в таблицу | 35. Данные добавляются в базу данных, отображается сообщение об успешном импорте |
| 36. Администратор экспортирует таблицу | 37. Создаётся .csv файл с данными из выбранной таблицы, разделёнными «;» |
| 38. Администратор открывает список работников | 39. Отображается информация о работниках |
| 40. Администратор открывает список пользователей | 41. Отображается информация о пользователях |
| 42. Администратор удаляет пользователя | 43. Отображается сообщение о подтверждении удаления |
| 44. Администратор подтверждает удаление записи | 45. Запись удаляется из базы данных, отображается сообщение об успешном удалении |

Диаграмма деятельности – это блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, при этом внимание фиксируется на результате деятельности.

Диаграммы деятельности сотрудника, менеджера и администратора предоставлены в соответствии с рисунками 2.3-2.5.

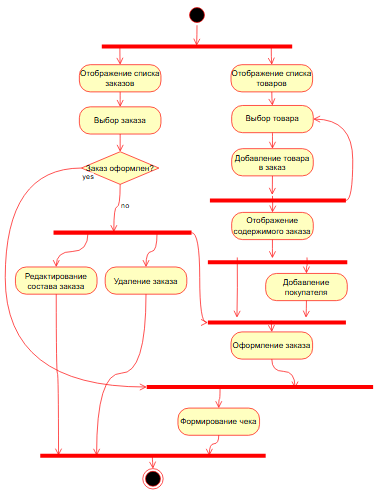


Рисунок 2.3 – Диаграмма деятельности сотрудника

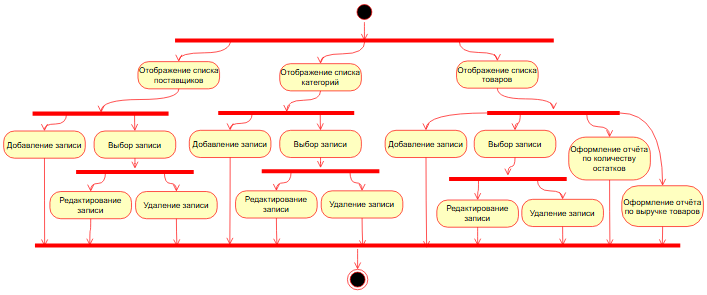


Рисунок 2.4 – Диаграмма деятельности менеджера

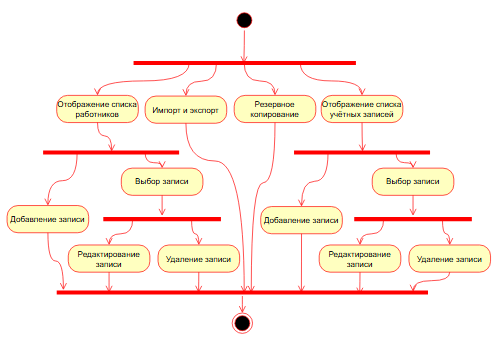


Рисунок 2.5 – Диаграмма деятельности администратора

* 1. **Алгоритм представления задачи**
     1. **Детальное проектирование пользовательского интерфейса**

Детальное проектирование ПО – это подробное описание компонентов программного обеспечения и интерфейсов между ними

Прототипирование программного обеспечения - этап разработки программного обеспечения, процесс создания прототипа программы - макета (черновой, пробной версии) программы, обычно - с целью проверки пригодности предлагаемых для применения концепций, архитектурных или технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки.

Прототип программы представлен в соответствии с рисунками 2.6-2.17

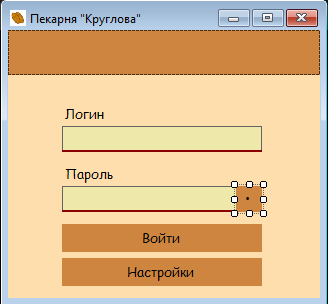


Рисунок 2.6 – Прототип формы авторизации без captcha

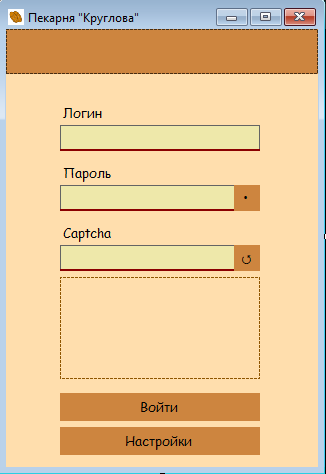


Рисунок 2.7 – Прототип формы авторизации с captcha

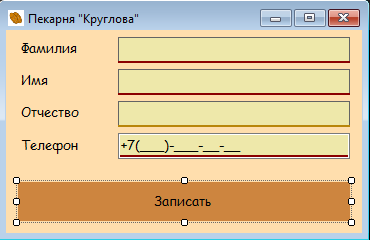


Рисунок 2.8 – Прототип формы клиента

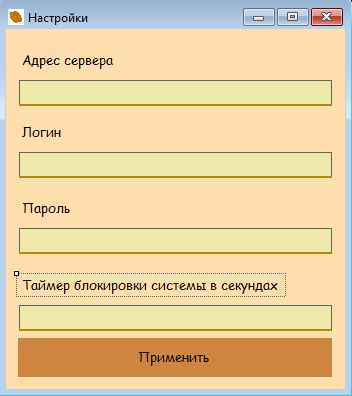


Рисунок 2.9 – Прототип формы настроек

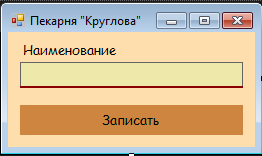


Рисунок 2.10 – Прототип формы элемента справочника

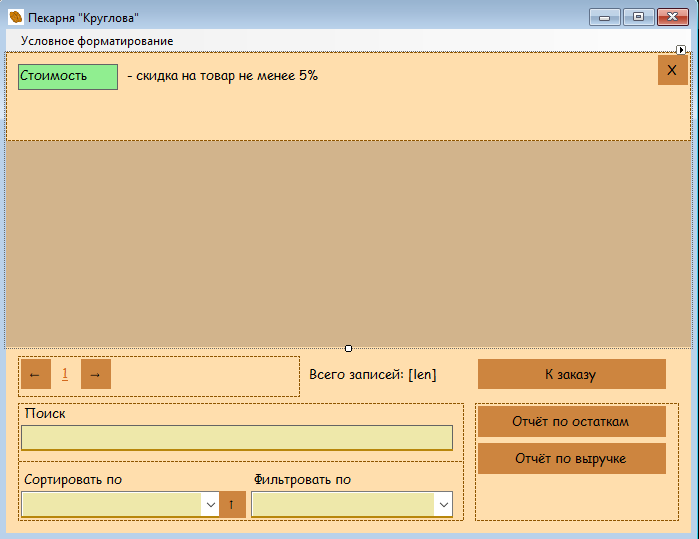


Рисунок 2.11 – Прототип формы списка

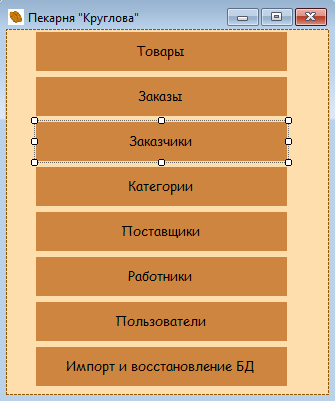


Рисунок 2.12 – Прототип формы меню

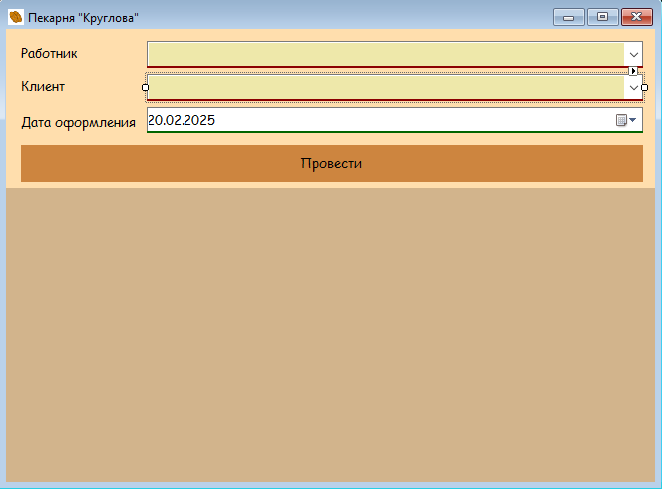


Рисунок 2.13 – Прототип формы заказа

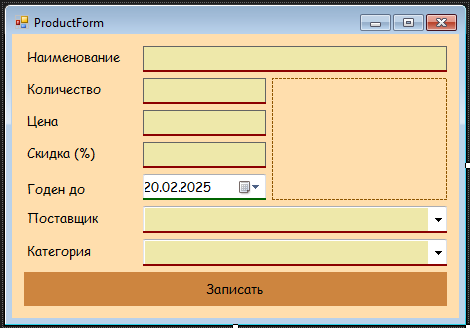


Рисунок 2.14 – Прототип формы товара

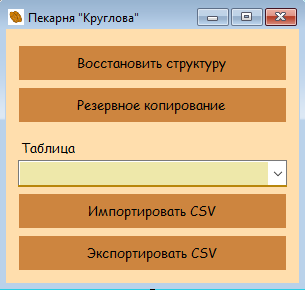


Рисунок 2.15 – Прототип формы администрирования

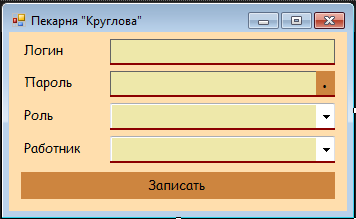


Рисунок 2.16 – Прототип формы пользователя

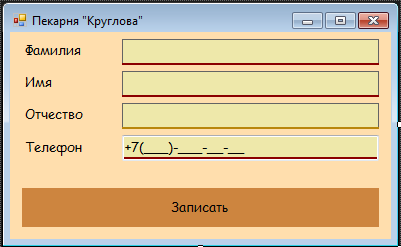


Рисунок 2.17 – Прототип формы работника

* + 1. **Описание основных программных модулей**

AuthForm - модуль формы авторизации. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

Connection – модуль для работы с базой данных: получения, добавления, редактирования и удаления данных.

DateSelectForm - модуль формы выбора даты. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

DictionaryForm - модуль формы записи справочника. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

ListForm - модуль формы списка. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

MenuForm - модуль формы меню. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

OrderForm - модуль формы заказа. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

ProductForm - модуль формы записи товара. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

Program - модуль, инициализирующий работу приложения.

ReportBuilder - модуль компоновки отчётов.

User - объект, содержащий данные о текущем пользователе в системе.

UserForm - модуль формы записи пользователя. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

WorkerForm - модуль формы записи работника. Устанавливает, форматирует, отображает элементы управления, устанавливает обработчики событий на форму.

* 1. **Применяемые меры защиты информации**

Защита данных – это предупреждение несанкционированного доступа к данным, их изменения или разрушения со стороны пользователей; предупреждения изменения или разрушения данных при сбоях аппаратных или программных средств и ошибках в работе сотрудников группы эксплуатации.

Функции:

* обеспечение безопасности данных:
* обеспечение секретности данных:

В процессе разработки информационной системы были реализованы следующие меры защиты информации:

* проверка на заполнение обязательных полей;
* блокировка ввода букв английского алфавита во всех полях, кроме логина, пароля и наименования поставщика;
* при создании продажи автоматически проставляются системная дата, информация об организации;
* подтверждение действия при добавлении, изменении и удалении записи из БД;
* резервное копирование;
* скрытие части персональных данных пользователей;
* при отсутствии активности работы в системе автоматический перевод на форму авторизации;
* скрытие пароля;
* шифрование пароля;
* разграничение прав доступа на следующие роли: администратор, менеджер, сотрудник.

1. **Тестирование**

Тестирование – ряд мероприятий, связанных с различного рода испытаниями объекта тестирования с целью установления соответствия или несоответствия его характеристик определенным требованиям и выявления дефектов. Дефектами, в свою очередь, могут быть как ошибки в работе, так и неприемлемое качество функционирования в определенных условиях эксплуатации.

Результаты тестирования - это список выявленных несоответствий и дефектов, но, как правило, без указания их причин.

Тестирование программного продукта проводят как на этапах его создания, так и на этапах технического сопровождения и дальнейшего развития. Оно является одним из наиболее устоявшихся способов обеспечения качества разработки программного обеспечения и входит в набор эффективных средств современной системы обеспечения качества программного продукта.

Таблица 3.1 Тестирование программы

|  |  |  |
| --- | --- | --- |
| Название элемента формы | Ожидаемый результат | Полученный результат |
| 1 | 2 | 3 |
| **Форма авторизации** | | |
| Кнопка «Войти» | При правильно введённых логине и пароле – переход в меню | Корректное выполнение команды |
| Кнопка закрытия формы | Отображение сообщения о подтверждении действия | Корректное выполнение команды |
| Кнопка «Настройки» | Переход на форму настроек | Корректное выполнение команды |
| Кнопка с точкой | Переключение видимости ввода в поле «Пароль» | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Кнопка «Войти» | При неправильно введённых логине и пароле – очистка полей «Логин» и «Пароль», отображение captcha, поля для её ввода и кнопки для её замены | Корректное выполнение команды |
| **Форма восстановления** | | |
| Кнопка «Восстановить структуру» | Создание структуры БД | Корректное выполнение команды |
| Кнопка «Резервное копирование» | Отображение диалога выбора директории, при выборе - создание резервной копии данных БД | Корректное выполнение команды |
| Кнопка «Импортировать CSV» | Отображение диалога выбора .csv файла, при выборе – добавление записей в выбранную таблицу БД | Корректное выполнение команды |
| Кнопка «Экспортировать CSV» | Отображение диалога сохранения .csv файла, при выборе – создание .csv файла с данными из выбранной таблицы БД | Корректное выполнение команды |
| **Форма меню администратора** | | |
| Кнопка «Работники» | Переход на форму списка работников | Корректное выполнение команды |
| Кнопка «Пользователи» | Переход на форму списка пользователей | Корректное выполнение команды |
| Кнопка «Импорт и восстановление БД» | Переход на форму восстановления | Корректное выполнение команды |
| **Форма меню менеджера** | | |
| Кнопка «Товары» | Переход на форму списка товаров | Корректное выполнение команды |
| Кнопка «Категории» | Переход на форму списка категорий товаров | Корректное выполнение команды |
| Кнопка «Поставщики» | Переход на форму списка поставщиков | Корректное выполнение команды |

Продолжение таблицы 3.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| **Форма меню сотрудника** | | |
| Кнопка «Товары» | Переход на форму списка товаров | Корректное выполнение команды |
| Кнопка «Заказы» | Переход на форму списка заказов | Корректное выполнение команды |
| Кнопка «Клиенты» | Переход на форму списка клиентов | Корректное выполнение команды |

1. **Экономическое обоснование**

Программа разработана для конкретных категорий пользователей. Цель разработки - создание программного продукта для ведения, учёта и анализа данных.

В процессе исследования предметной области был проанализирован программный продукт для ведения, учёта и анализа данных 1С.

По сравнению с аналогом программа имеет некоторые преимущества, среди которых:

* простая настройка системы;
* разработанная программа является готовым решением, в то время как аналог требует разработки рабочей среды специалистом.

В таблице 4.1 приведены основные этапы разработки программного продукта.

Таблица 4.1 – Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название этапа** | **Содержание** | **Количество дней** |
| 1 | 2 | 3 | 4 |
| 1 | Анализ | Анализ предметной области | 2 |
| 2 | Проектирование | Составление ТЗ | 4 |
| 3 | Разработка | Разработка структуры базы данных | 25 |
| Разработка форм программного продукта |
| Разработка программного кода |
| 4 | Тестирование | Проверка работоспособности всех функций программного продукта | 3 |

Продолжение таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | Документирование | Составление руководства пользователя | 2 |
| Составление листинга | 1 |
| Составление пояснительной записки | 3 |
| Итого | | | 40 |

Приведенные в данном разделе выпускной квалификационной работы результирующие таблицы, перечни данных, позволяют сопоставить результаты разработки и затраты на нее, чтобы сделать вывод об эффективности проекта. Исходные данные для расчета экономических показателей приведены в таблице 4.2.

|  |  |  |  |
| --- | --- | --- | --- |
| Обозначение | Наименование показателя | Единицы измерения | Значение показателя |
| СЭВМ | Стоимость ЭВМ | руб. | 30000 |
| ДМ | Среднее количество рабочих дней в месяце | дни | 24 |
| Счас | Стоимость час работы программиста | руб. | 100 |
| ЦЭЛ | Тариф за 1 кВт/час | руб. | 4,4 |
| Тчас | Количество часов работы над разработкой в день | час | 5 |
| P | Мощность, потребляемая ЭВМ | кВт | 0,07 |
| Тинтер | Тариф за услугу Интернет, месяц | руб. | 500 |

* 1. **Расчет себестоимости программного продукта**

Себестоимость разработки программного продукта рассчитываются по следующей формуле:

|  |  |
| --- | --- |
| Спп = ЗП + Зэл + А + Зпр | (1) |

где ЗП – зарплата программиста;

Зэл – затраты на электроэнергию;

А – амортизация оборудования;

Зпр – прочие расходы.

Зарплата программиста рассчитывается по формуле:

|  |  |
| --- | --- |
| ЗП **=** Сч **×** Тчас **×** Д | (2) |

где Сч– стоимость час работы программиста (руб.);

Тчас– количество часов работы над разработкой в день;

Д– количество дней, затраченных на разработку программного продукта.

Согласно таблице 4.1 на разработку программного продукта ушло 40 дней. Среднее количество часов работы в день составило 5 часов. Так как навыков разработки программных продуктов мало, то часовая ставка начинающего программиста составит 100 рублей, что соответствует данным таблицы 4.2

|  |  |
| --- | --- |
| ЗП **=** 100 **×** 5**×** 40 = 20000 руб. |  |

Затраты на электроэнергию рассчитывается по формуле:

|  |  |
| --- | --- |
| Зэл **=** Чотраб **×** Ц1кВт **×** Р | (3) |

где Чотраб – количество часов работы оборудования (ПК), освещения;

Ц1кВт – стоимость 1кВт электроэнергии, (руб.);

Р – мощность, потребляемая ЭВМ, (кВт).

Количество часов работы оборудования (ПК), освещения рассчитывается по формуле:

|  |  |
| --- | --- |
| Чотраб **=** Тчас **×** Д | (4) |

где Тчас – количество часов работы над разработкой в день;

Д– количество дней, затраченных на разработку программного продукта.

Количество дней на разработку составило 40, что соответствует данным таблицы 4.1. Количество часов работы в день 5, в соответствии с таблицей 4.2

|  |  |
| --- | --- |
| Чотраб **=** 5 **×** 40 = 200 часов |  |

В соответствии с данными таблицы 4.2 стоимости 1кВт электроэнергии равна 4,4 рубля, а потребляемая мощность ПК составляет 0,07 кВт в час.

|  |  |
| --- | --- |
| Зэл **=** 200 **×** 4,4 **×** 0,07 = 61,6 руб. |  |

Амортизационные отчисления рассчитываются с учетом нормы амортизации (Нам **=** 5-10%):

|  |  |
| --- | --- |
| А **=** Нам **×** СЭВМ | (5) |

где СЭВМ – стоимость оборудования, согласно таблице 4.2.

|  |  |
| --- | --- |
| А **=** 30000 **×** 0,05=1500руб. |  |

Для разработки программного продукта необходимо использовать Интернет, следовательно, определим затраты на Интернет, как прочие.

|  |  |
| --- | --- |
| Зпр **=** Тинтер = Цинт(1ч) **×** Тчас **×** Д | (6) |

Цена одного часа работы в Интернет рассчитывается по формуле:

|  |  |
| --- | --- |
| Цинт(1ч) = Цтар : Дмес : 24 | (7) |

Согласно таблице 4.2 стоимость услуги Интернет в месяц составила 500 рублей. Исходя из стоимости тарифа определим стоимость часа работ в Интернет.

|  |  |
| --- | --- |
| Цинт(1ч) = 500 : 24 : 24 = 0,86 руб.  Зпр **=** 0,86 **×** 5 **×** 40 = 172 руб. |  |

Выполнив все расчеты определим себестоимость программного продукта.

|  |  |
| --- | --- |
| Спп= 20000 + 61,6 +1500+ 172 = 21733,6 руб. |  |

* 1. **Стратегия продвижения программного продукта на рынок**

Сегодня самым эффективным способом продажи программного продукта является Интернет.

Одним из наиболее эффективных способов привлечения потенциальных покупателей является реклама. Существует несколько видов рекламы:

* баннеры - изображения с текстом, размещаемые на сайте. В среднем стоимость рекламы такого плана предполагает оплату в 350 рублей за 1000 показов обычного баннера, и вдвое больше – анимированного;
* видеореклама - короткие ролики, которые размещают на популярных видеоплатформах и в стриминговых сервисах. Они отображаются до, после или во время основного видео;
* контекстная реклама - когда пользователь ищет в сети какой-то запрос, ему демонстрируются не только результаты поисковой выдачи, но ещё и рекламные объявления близкой тематики. Для большинства тематик можно найти тарифы по 10–15 рублей за один переход на сайт.

Для продвижения программного продукта выбрана реклама на баннерах на 10000 показов.

Суммарные затраты на продвижение программы определяются по следующей формуле:

|  |  |
| --- | --- |
| Зпродв = Трекл × Кпоказ | (8) |

где Зпродв – суммарные затраты на продвижение программы (руб.);

Трекл – тариф на показ рекламы (руб.);

Кпоказ – количество показов рекламы;

|  |  |
| --- | --- |
| Зпродв = (350 : 1000) × 10000 = 3500 руб. |  |

* 1. **Расчёт цены разработанной программы**

Цена разработанной программы определяется по следующей формуле:

|  |  |
| --- | --- |
| Цпп = Спп + Зпродв | (9) |

где Спп – себестоимость программного продукта (руб.);

Зпродв – суммарные затраты на продвижение программы (руб.);

|  |  |
| --- | --- |
| Цпп = 21733,6 + 3500 = 25233,6 |  |

Выручка от продаж при условии Nп – количество пользователей, желающих прибрести программу, составит:

|  |  |
| --- | --- |
| В = Цпп × Nп | (10) |

Срок окупаемости инвестиционного проекта (Ток) – это период времени, который потребуется для возмещения инвестиций. Ток – определяется с учетом дисконтирования, путем суммирования ежегодных поступлений до определенного периода, в котором они превзойдут первоначальные расходы денежных средств. Расчёт срока окупаемости представлен в таблице 4.3

Таблица 4.3 – Расчет срока окупаемости разработанной программы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Период | Цена ПП,  руб. | Кол-во загрузок  шт. | Выручка,  руб. | Накапливаемая выручка,  руб. |
| 1 | 2 | 3 | 4 | 5 |
| Июнь | 200 | 13 | 2600 | 2600 |
| Июль | 200 | 10 | 2000 | 4600 |

Продолжение таблицы 4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Август | 200 | 15 | 3000 | 7600 |
| Сентябрь | 200 | 9 | 1800 | 9400 |
| Октябрь | 200 | 12 | 2400 | 11800 |
| Ноябрь | 200 | 10 | 2000 | 13800 |
| Декабрь | 200 | 12 | 2400 | 16200 |
| Январь | 200 | 9 | 1800 | 18000 |
| Февраль | 200 | 9 | 1800 | 19800 |
| Март | 200 | 11 | 2200 | 22000 |
| Апрель | 200 | 10 | 2000 | 24000 |
| Май | 200 | 8 | 1600 | 25600 |
| Итого | 2400 | 128 | 25600 |  |

* 1. **Технико-экономические показатели разработки программы**

Обобщенные технико-экономические показатели разработки программы сведены в таблицу 4.4.

Таблица 4.4 – Технико-экономические показатели разработки программы

|  |  |
| --- | --- |
| Показатель | Значение |
| Себестоимость программного продукта (руб.) | 21733,6 |
| Цена разработки ПП (руб.) | 25233,6 |
| Отпускная цена (руб.) | 200 |
| Срок окупаемости (мес.) | 12 |

Данные произведенные расчеты показывают, что разработанный программный продукт является эффективным с экономической точки зрения.

**Заключение**

В ходе выполнения выпускной квалификационной работы была создана автоматизированная информационная система по учёту проданных товаров пекарни «Круглова» на базе платформы .NET.

В процессе создания использовался язык C# и СУБД MySQL

Программный продукт обеспечивает выполнение следующих функций:

* ведение данных о товарах, клиентах, заказах;
* формирование отчётов.

В программном приложении реализованы функции:

* автоматизация ввода исходных данных;
* проверка на корректность ввода данных;
* импорт и экспорт данных;
* резервное копирование данных;
* создание отчётов по количеству остатков и выручке товаров;
* формирование чека для заказа;
* пагинация, поиск, сортировка и фильтрация товаров;
* авторизация и разграничение прав доступа к объектам БД;
* генерация captcha при неудачной авторизации;

Программа требует минимальные ресурсы компьютера. Программный продукт можно переносить на различных источниках, при этом программа будет работать на любом компьютере.

**Литература**

1 Бен Форта, Освой самостоятельно SQL за 10 минут. М., Вильямс, 2021

2 Мацяшек Л.А., Лионг Б.Л. Практическая программная инженерия на основе учебного примера - М.:БИНОМ, Лаборатория знаний – ЭБК BOOK.RU 2022.

3 Могрунов Е.П. Основы языка SQL, учебное пособие. – СПб.: БХВ-Петербург, 2022.

4 Новиков Б.А., Основы технологий баз данных, Postgres Professional. - М., 2021.

5 Прайс Марк Дж. C# 7 и .NET Core Кросс-платформенная разработка для профессионалов. 3-е изд. - СПб.: Питер, 2022.

6 Эндрю Троелсен Язык программирования C# 6.0 и платформа .NET 4.6, 7-е изд.: пер. с англ. - М.: ООО “И.Д. Вильямс”, 2021.