Министерство образования Саратовской области

Государственное автономное профессиональное образовательное учреждение Саратовской области

«Новоузенский агротехнологический техникум»

**Специальность: 09.02.07 Информационные системы и программирование**

**Курсовая работа**

**по МДК 01.01. «Разработка программных модулей»**

**на тему:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Выполнил: студент(ка) группы \_\_\_\_\_\_

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

(фамилия, имя, отчество)

02.06.2023г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Руководитель: преподаватель

Бекбулатов Р.Ж.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество) (оценка)

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

(подпись, дата)

г. Новоузенск 2023г.

Содержание

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| … |  |  |

Введение

В современном мире активно развиваются и внедряются информационные технологии. На сегодняшний день уже во многих областях деятельности человека можно некоторую часть работы доверить компьютеру, так как они могут сделать эту работу не хуже человека, а в некоторых случаях лучше. Персональные компьютеры намного быстрее человека могут произвести математические вычисление, поиск информации, решение каких–либо задач. Но без программного обеспечения компьютеры бесполезны. Для каждой области деятельности человека необходимо свое программное обеспечение. Сегодня без этих помощников не обходиться ни одна компания, ни одно предприятие.

На сегодняшний день на рынке программного обеспечения отмечается устойчивый интерес к программным продуктам, способным обеспечить эффективное управление различной области компаний и предприятий. Большинство руководителей понимают, что для успешной организации производственного процесса необходимо применять современные информационные технологии.

В связи с глобальной информатизацией всех областей жизнедеятельности человека, а также с растущим уровнем потребности к программным продуктам, способным обеспечить эффективное управление производственным процессом, предоставляется возможность создания программного средства, решающего поставленную задачу, применяя современные информационные технологии.

Заданием на курсовое проектирование является разработка программного средства для автоматизации работы кассира в кинотеатре, которое должно обеспечивать ведение базы данных о фильмах и сеансах, билетах и брони на них, залах, персонале и пользователях, осуществлять формирование различных отчётов, выполнять продажу билетов на сеанс.

Автоматизация перечисленных процессов исключит возникновение ошибок при работе с данными, избавит от больших объемов различной документации и ускорит работу.

В результате работы над курсовым проектом будет разработано программное средство и составлена пояснительная записка, которая включает в себя шесть разделов, содержащих необходимую и достаточную информацию по использованию данного программного средства, подробное описание программы и вычислительной системы.

В заключении проанализируется созданное программное приложение, определится степень соответствия поставленной задачи и выполненной работы.

Также пояснительная записка будет содержать приложение и графическую часть, состоящую из диаграммы UML и классов.

1. Постановка задачи

1.1 Назначение разработки

Предметной областью решаемой задачи является работа кассира в кинотеатре. Такая работа связана с накоплением большого количества информации сеансах, билетах.

Традиционно информация храниться на бумажных носителях. При этом трудно осуществить быстрый поиск и исправление ошибочно указанных данных, а так же продажа билетов занимает много времени.

Таким образом, автоматизация работы кассира в кинотеатре, является актуальной и перспективной.

В функции кассира входят:

* Работа с сеансами;
* Продажа билетов на сеансы;
* Бронирование билетов на сеансы.

Функции администратора:

* Работа с сеансами;
* Продажа билетов на сеансы;
* Бронирование билетов на сеансы.
* Работа с фильмами;
* Работа с залами;
* Работа с персоналом;
* Работа с пользователями;
* Просмотр отчетов.

Из всех задач, которые будет решать разрабатываемое программное средство, можно выделить ряд основных:

* хранить данные о сеансах, билетах, фильмах, залах, персонале и пользователях;
* осуществлять продажу и бронирование билетов на сеанс;
* формировать различные отчеты;
* выполнять поиск, сортировку, фильтрацию данных.

Данное программное средство направлено на повышение эффективности, а также скорости работы кассира. Также программа должна упростить работу, связанную с поиском, редактированием и добавлением информации о сеансах.

В настоящее время существует множество аналогичных программ, предназначенные для составления и ведения баз данных. Однако разработка программы, направленной на решение задачи в определённой сфере, не только облегчает труд кассира, но и обеспечивает правильность и безопасность хранения и редактирования данных, обрабатываемых этой программой.

1.2 Входные данные

Ко входным данным относятся все данные, заносимые в базу данных при добавлении или редактировании записей о сеансах, билетах, залах, фильмах, пользователях или о персонале.

Структура входной информации должна соответствовать структуре данных, определенной на этапе проектирования базы данных, и соответствовать типам данных, требуемых в определении фильтра и ключа поиска.

Все эти данные в коде программы преобразуются и записываются в базу данных. Проверка корректности вводимых данных осуществляется в программе, поэтому пользователю не придется беспокоиться о неправильном вводе.

Также к входной информации можно отнести информацию, задаваемую пользователем при фильтрации и поиске данных.

1.3 Выходные данные

К выходным данным можно отнести просматриваемую пользователем информацию на экране на любой форме программного средства.

Так же выходными данными в приложении являются отчеты, которые выводится как пользователю на экран. Результаты вывода отчетов представлены на рисунках 4.1-4.3.

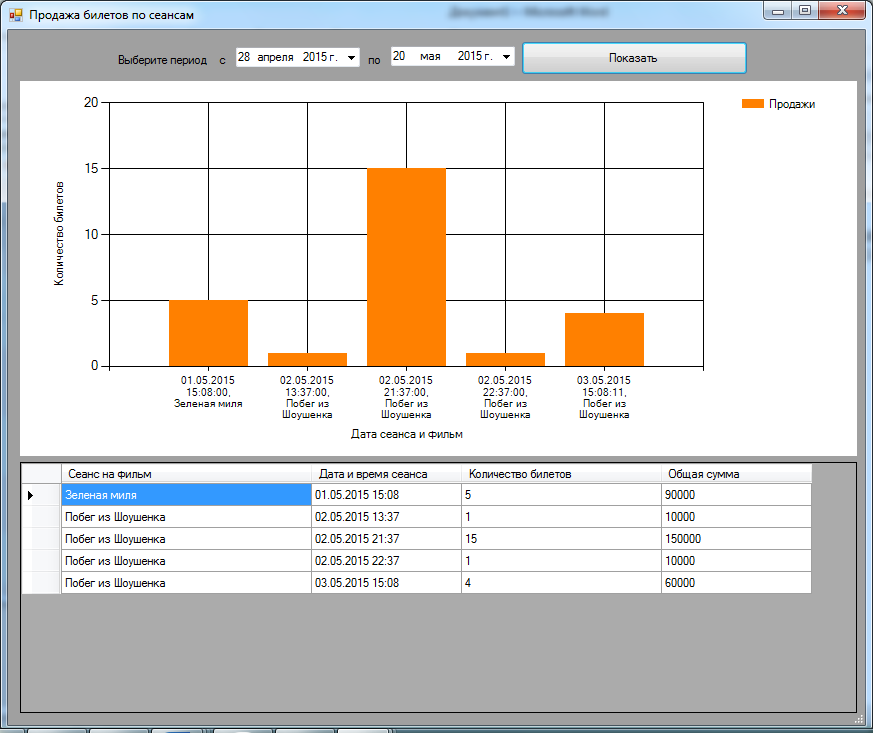


Рисунок 1.1 – Продажа билетов по сеансам

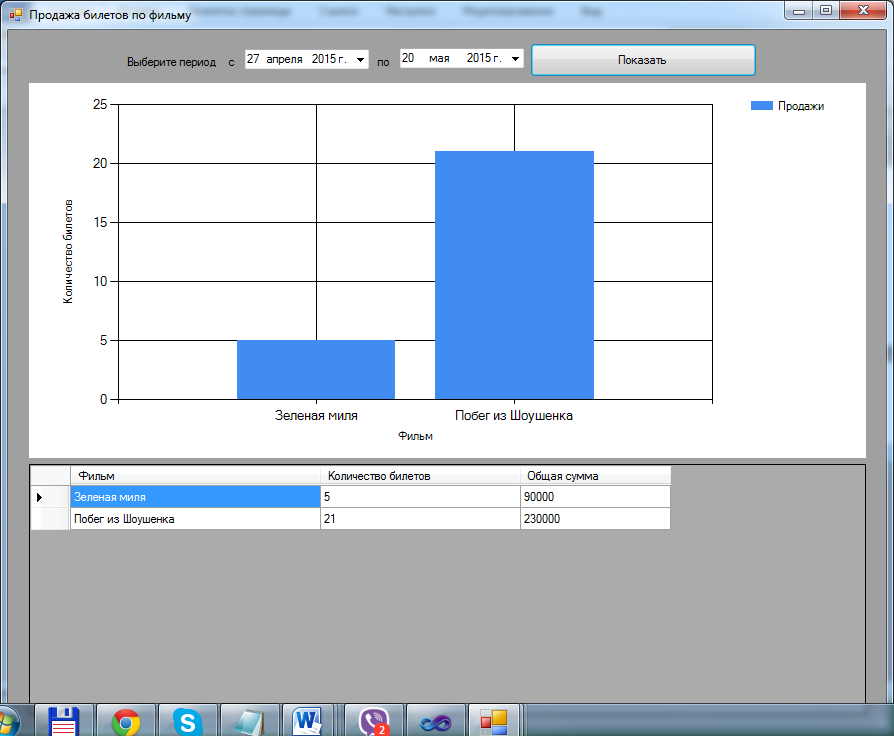


Рисунок 1.2 – Продажа билетов по фильмам

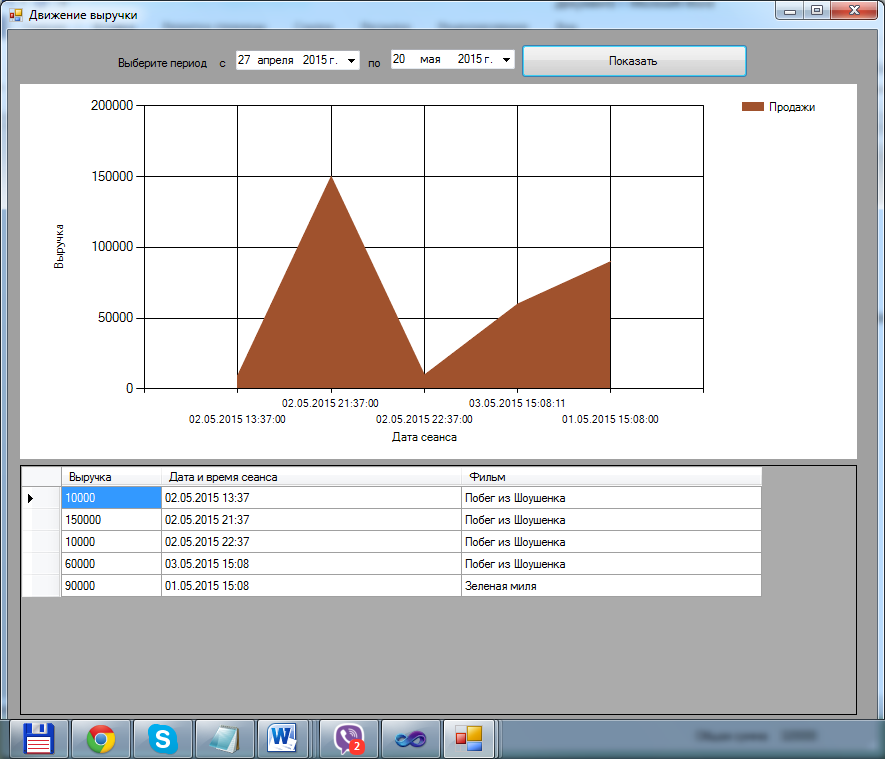


Рисунок 1.3 – Движение выручки за период

1.4 Среда эксплуатации

Данное программное средство разработано и предназначено для автоматизации работы кассира кинотеатра. Категория пользователя, на которого ориентирован данный программный продукт - кассир.

Основными задачами приложения являются ведение базы данных, продажа и бронирование билетов на сеансы, поиск и сортировка данных по различным критериям, формирование отчетов.

Данное приложение будет выполнять следующие функции:

– поиск и сортировка по различным критериям;

– ведение и редактирование баз данных;

– продажа и бронирование билетов на сеансы;

– автоматическое формирование отчетов в виде графиков.

В программе был организован интерфейс понятный на интуитивном уровне как пользователю, не обладающему глубокими знаниями компьютерных технологий и навыками работы с персональным компьютером, так и опытному пользователю. Необходимо отметить, что данный программный продукт обладает средствами, ограничивающими доступ к нему, а именно, при запуске приложения необходимо будет ввести пароль.

Необходимо отметить, что данное программное приложение не обладает средствами, препятствующими свободному распространения программы. Для работы с приложением не требуется установка дополнительных компонентов и специальных технических и программных ресурсов.

1.5 Обоснование выбора языка и сред разработки

Для реализации курсового проекта выбрана интегрированная среда разработки Microsoft Visual Studio. В целом среда имеет много достоинств:

* обеспечение интуитивно понятной, расширяемой, унифицированной среды для языка, конструкторов и инструментальных средств.
* предоставление разработчикам набора модернизированных взаимодействующих элементов, соответствующих имеющимся у разработчиков навыкам.
* предоставление высокопроизводительных инструментальных средств для всех этапов жизненного цикла разработки - от определения требований и планирования до последующего сопровождения продукта.

В качестве языка программирования был выбран C#.

Преимущества языка программирования C#:

* Подлинная объектная ориентированность (всякая языковая сущность претендует на то, чтобы быть объектом)
* Компонентно-ориентированное программирование
* Безопасный (по сравнению с языками C и C++) код
* Унифицированная система типизации
* Поддержка событийно-ориентированного программирования
* «Родной» язык для создания приложений в среде .NET
* . Объединение лучших идей современных языков программирования: Java, C++, Visual Basic и др.

C# — [язык программирования](http://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), сочетающий [объектно-ориентированные](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) и [аспектно-ориентированные](http://ru.wikipedia.org/wiki/%D0%90%D1%81%D0%BF%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) концепции. Разработан в [1998](http://ru.wikipedia.org/wiki/1998_%D0%B3%D0%BE%D0%B4)—[2001 годах](http://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4) группой инженеров под руководством [Андерса Хейлсберга](http://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%BB%D1%81%D0%B1%D0%B5%D1%80%D0%B3,_%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81) в компании [Microsoft](http://ru.wikipedia.org/wiki/Microsoft) как основной язык разработки приложений для платформы [Microsoft .NET](http://ru.wikipedia.org/wiki/.NET_Framework). [Компилятор](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80) с C# входит в стандартную установку самой .NET, поэтому программы на нём можно создавать и компилировать даже без инструментальных средств, вроде [Visual Studio](http://ru.wikipedia.org/wiki/Microsoft_Visual_Studio).

C# относится к семье языков с [C-подобным синтаксисом](http://ru.wikipedia.org/wiki/C-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81), из них его синтаксис наиболее близок к [С++](http://ru.wikipedia.org/wiki/C%2B%2B) и [Java](http://ru.wikipedia.org/wiki/Java). Язык имеет [строгую статическую типизацию](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](http://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_%D0%B2_%D1%8F%D0%B7%D1%8B%D0%BA%D0%B0%D1%85_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), [перегрузку операторов](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B9), указатели на функции-члены классов, атрибуты, [события](http://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B5), [свойства](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29), [исключения](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9), [комментарии](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29) в формате [XML](http://ru.wikipedia.org/wiki/XML). Переняв многое от своих предшественников — языков [С++](http://ru.wikipedia.org/wiki/C%2B%2B), [Java](http://ru.wikipedia.org/wiki/Java), [Delphi](http://ru.wikipedia.org/wiki/Delphi_%28%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%29), [Модула](http://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2_%28%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F%29) и [Smalltalk](http://ru.wikipedia.org/wiki/Smalltalk) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает [множественное наследование](http://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%28%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%29#.D0.9C.D0.BD.D0.BE.D0.B6.D0.B5.D1.81.D1.82.D0.B2.D0.B5.D0.BD.D0.BD.D0.BE.D0.B5_.D0.BD.D0.B0.D1.81.D0.BB.D0.B5.D0.B4.D0.BE.D0.B2.D0.B0.D0.BD.D0.B8.D0.B5) классов (в отличие от [C++](http://ru.wikipedia.org/wiki/C%2B%2B)).

C# разрабатывался как язык программирования прикладного уровня для [CLR](http://ru.wikipedia.org/wiki/Common_Language_Runtime) и, как таковой, зависит, прежде всего, от возможностей самой [CLR](http://ru.wikipedia.org/wiki/Common_Language_Runtime). Это касается, прежде всего, системы типов C#, которая отражает [FCL](http://ru.wikipedia.org/wiki/Base_Class_Library). Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем. (Однако эта закономерность была нарушена с выходом C# 3.0, представляющим собой расширения языка, не опирающиеся на расширения платформы .NET.) CLR предоставляет C#, как и всем другим [.NET](http://ru.wikipedia.org/wiki/.NET_Framework)-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, [сборка мусора](http://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0) не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на [VB.NET](http://ru.wikipedia.org/wiki/Visual_Basic_.NET), [J#](http://ru.wikipedia.org/wiki/Visual_J%E2%99%AF) .

1. Логическая и физическая модель

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы - атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация - это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

Структура базы данных разрабатываемого программного средства включает семь таблиц. Их структура, и краткое описание приводится в таблицах 2.1-2.7.

Таблица «Билет» предназначена для хранения информации о билетах.

Таблица 2.1 – Структура таблицы «Билет»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Код | int | Длинное целое | Код |
| Дата\_время | datetime | 64 | Дата и время покупки |
| Сеанс | int | Длинное целое | Код сеанса |
| Фильм | varchar | 300 | Название фильма |
| Номер\_зала | int | Длинное целое | Номер зала |
| Место | int | Длинное целое | Место |
| Ряд | int | Длинное целое | Ряд |
| Куплен | tinyint | 1 | Флаг купленного билета |
| Цена | bigint | Длинное целое | Цена билета |

Таблица «Бронь» предназначена для хранения информации о забронированных билетах.

Таблица 2.2 – Структура таблицы «Бронь»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Код | int | Длинное целое | Код |
| Билет | int | Длинное целое | Код билета |
| ФИО | varchar | 65535 | ФИО бронирующего |

Таблица «Зал» предназначена для хранения информации о залах.

Таблица 2.3 – Структура таблицы «Зал»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Номер | int | Длинное целое | Код зала |
| Название | varchar | 65535 | Название |
| Планограмма | varchar | 65535 | Планограмма |
| Мест | int | Длинное целое | Мест |
| Рядов | int | Длинное целое | Рядов |
| Всего\_мест | Высчитываемое | - | Всего мест |

Таблица «Персонал» предназначена для хранения информации о персонале.

Таблица 2.4 – Структура таблицы «Персонал»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Код | int | Длинное целое | Код |
| ФИО | varchar | 320 | ФИО работинка |
| Должность | varchar | 320 | Должность |
| Телефон | varchar | 320 | Телефон |

Таблица «Пользователь» предназначена для хранения информации о пользователях.

Таблица 2.5 – Структура таблицы «Пользователь»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Код | int | Длинное целое | Код |
| Роль | varchar | 120 | Роль |
| Пароль | varchar | 150 | Пароль |

Таблица «Сеанс» предназначена для хранения информации о сеансах.

Таблица 2.6 – Структура таблицы «Сеанс»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Код | int | Длинное целое | Код |
| Дата\_время | datetime | 64 | Дата и время сеанса |
| Номер\_зала | int | Длинное целое | Номер зала |
| Фильм | int | Длинное целое | Код фильма |
| Места | int | Длинное целое | Свободные места |
| Цена | bigint | Длинное целое | Цена билета на сеанс |

Таблица «Фильм» предназначена для хранения информации о фильмах.

Таблица 2.6 – Структура таблицы «Фильм»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля | Описание поля |
| Код | int | Длинное целое | Код |
| Название | varchar | 300 | Название |
| Жанр | varchar | 200 | Жанр |
| Продолжительность | int | Длинное целое | Продолжительность |
| Рейтинг | float | Длинное с плавающей точкой | Рейтинг фильма |
| Описание | varchar | 65535 | Описание |
| Актеры | varchar | 65535 | Актеры |
| Год | int | Длинное целое | Год выхода в прокат |

Структура базы данных представлена на схеме на рисунке 2.1.

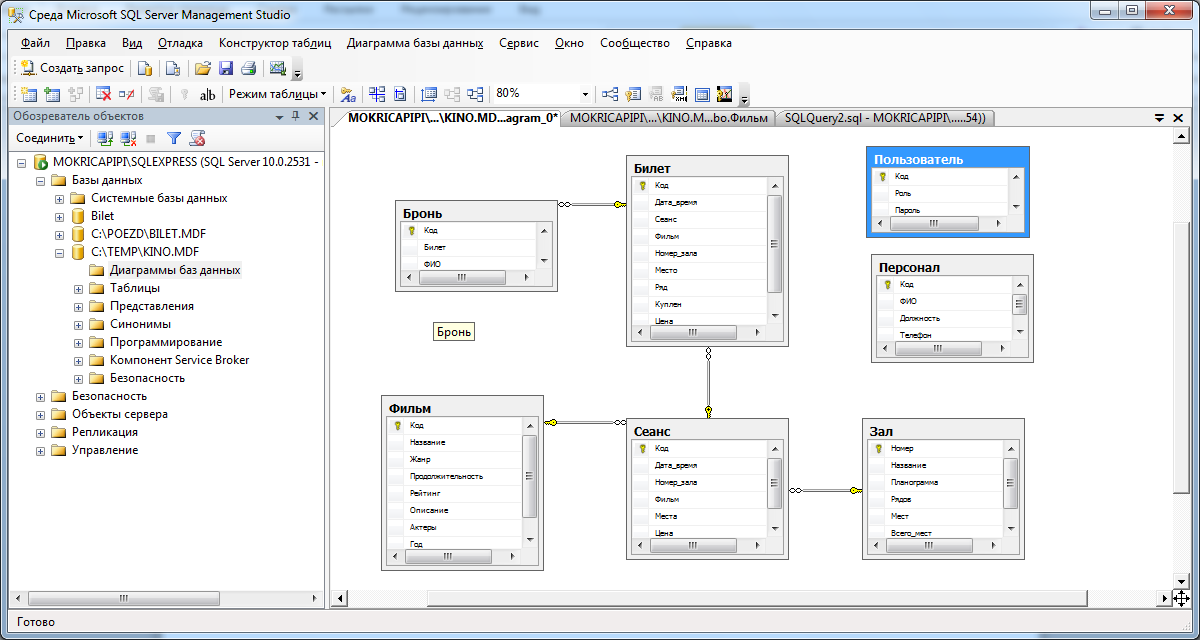


Рисунок 2.1 – Схема данных

1. Проектирование программного средства

Разрабатываемое приложение будет защищено от несанкционированного доступа паролем для того, чтобы другой пользователь не смог получить доступ к базе данных.

Доступ к структуре базы данных должен быть ограничен для неопытных пользователей в целях сохранения структуры базы данных.

Разрабатываемое программное средство должно иметь понятный и удобный в использовании интерфейс, чтобы взаимодействие между программой и пользователем было максимально упрощено. Для обучения пользователей необходимо разработать справочную систему, в которой должны быть раскрыты все аспекты работы с программой, возможные трудности, возникшие во время работы и пути их решения. Для удобного перемещения по экранным формам программного средства, должно присутствовать меню, которое давало бы возможность быстрого достижения пользователем поставленной цели.

Все записи протокола должны исключать возможность ввода некорректных данных.

Кроме этого, при разработке форм необходимо соблюдать определённые требования: формы в приложении должны иметь стандартные элементы управления данными, не должны быть перегружены информацией, понятны простому пользователю. При конструировании форм в необходимых случаях нужно предусмотреть возможность защиты данных от изменения, установить ограничения на корректировку записей, а также обработку на отсутствие данных при выполнении отбора записей в базе данных.

Из всех задач, которые будет решать разрабатываемое программное средство, можно выделить ряд основных требований:

* Продажа билетов;
* Бронирование билетов
* Работа с базой данных.

Исходя из исследования предметной области задачи принято решение об организации программного средства в виде базы данных MS SQL SERVER. В программе предполагается создание навигационного меню, состоящего из пунктов «Сеансы», «Билеты», «Фильмы», «Залы», «Персонал», «Отчеты», «Забронированные билеты», «Пользователи» и «Справка».

Пункт меню «Сеансы» будет вызывать форму с сеансами, где пользователь сможет добавить, редактировать сеансы, осуществить поиск и сортировку а так же продать билет на сеанс.

Пункт меню «Билеты» будет вызывать форму с билетами, где пользователь сможет осуществить просмотр и редактировать билеты, а так же осуществить возврат.

Пункт меню «Фильмы» будет вызывать форму с фильмами, где пользователь сможет осуществить просмотр, добавить и редактировать фильмы, а так же осуществить поиск и сортировку.

Пункт меню «Залы» будет вызывать форму с залами, где пользователь сможет осуществить редактирование залов.

Пункт меню «Персонал» будет вызывать форму с персоналом, где пользователь сможет осуществить просмотр, добавить и редактировать работников, а так же осуществить поиск и сортировку.

Пункт меню «Отчеты» будет вызывать форму с меню на котором находятся подпункты с различными видами отчетов.

Пункт меню «Забронированные билеты» будет вызывать форму с забронированными билетами, где пользователь сможет осуществить просмотр, а так же удаление билета с бронью и продажу забронированного билета.

Пункт меню «Пользователи» будет вызывать форму с пользователями программы, где пользователь сможет осуществить просмотр, добавить а так же сменить пароли у пользователей.

Пункт меню «Справка» будет вызывать справку по пользованию программой.

На формах будут располагаться различные элементы управления, предназначенные для определенных задач. Например, стандартный элемент управления Select, предназначенный для выбора нужных данных из списка программного приложения и будет осуществлять выбор пользователи. За компонентом Button будут закреплены разные действия для выполнения определенных функций, таких как: удаление, добавления и редактирования данных. Стандартный элемент управления Textbox будет использован для ввода и вывода разнообразных данных и для отображения текста страницах.

1. Создание программного средства

Программное средство имеет возможность фильтрации и редактирования данных, формирования отчетов, продажи билетов и их бронирование.

Данное программное средство было разработано при использовании персонального компьютера со следующей конфигурацией:

* процессор: Intel Core 2 Duo 2,5 гигагерц;
* материнская плата: Gigabyte GA-M720-US3;
* оперативная память: Corsair DDR2-800 мегагерц, 2048 мегабайт;
* видеокарта: GeForce 8600MGS;
* винчестер: SamsungSpinPointF1 250 мегабайт;
* клавиатура и мышь: Logitec.

Программа не требовательна к системным ресурсам, обладает развитым интерактивным интерфейсом, проста в использовании, однако требует специальных навыков при работе.

В целом программное средство делится на 2 части: Администраторская и часть для кассиров. Для входа с соответствующей учетной записью необходимо выбрать на первой форме пользователя и ввести пароль.

Для понятного разделения в программе имеется 2 вида главного меню:

* Меню для администратора;
* Меню для кассиров.

Меню администратора представлено на рисунке 4.1.

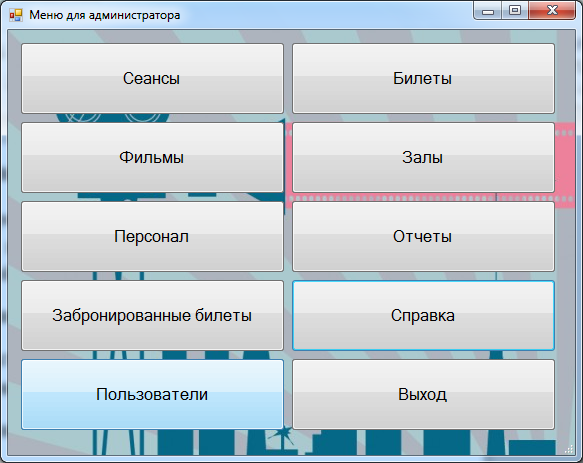


Рисунок 4.1 – меню для администратора

Меню кассиров представлено на рисунке 4.2.

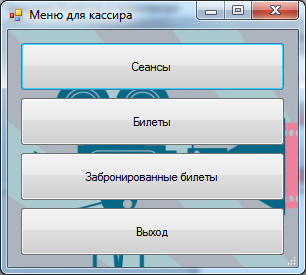


Рисунок 4.2 – меню для кассиров

Существуют общие требования, которые предъявляются к программным Windows-приложениям:

* соответствие стандартам организации интерфейса: использование многооконного подхода, реализация управления работой программного приложения с помощью элементов управления;
* требования к выбранной цветовой схеме рабочего экрана;
* выполнение одной функции с помощью разных элементов управления;
* предоставление справочной информации.

Согласно общим требованиям к графическому интерфейсу программное приложение должно:

* содержать привычные и понятные пользователю пункты меню, соответствующие функциям обработки;
* ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия;
* удовлетворять правилу «шести» – в одну линейку меню включать не более шести понятий, каждое из которых содержит не более шести опций;
* сохранять стандартизированное назначение и местоположение на экране графических объектов, работающих в среде Windows.

Современные программные приложения работают в диалоговом режиме.

В данном программном средстве данные требования учтены.

В работе приложения предусмотрены некоторые ситуации, которые должны предупреждать пользователя, чтобы он выполнял те или иные действия для продолжения работы программы. Для этого существуют системные сообщения, например, если пользователь начал добавлять данные о фильме, но не заполнил все обязательные поля, то в этом случае выводится на экран сообщение с предупреждением о невозможности продолжения работы, так как не все поля заполнены.

Разработанное приложение имеет понятный интерфейс, довольно просто в применении, и в тоже время позволяет уменьшить затраты времени и материальных ресурсов, используемых при автоматизации работы кассира кинотеатра.

Справочная система содержит в себе полную информацию обо всем разработанном приложении и его функциях. Состоит из двух разделов:

«О программе»;

«Пользование программой»

Внешний файл справочной системы «1.chm» будет создан в формате CHM.

Раздел «О программе» будет включать информацию о выполняемых функциях программным средством.

Раздел «Пользование программой» будет содержать полную информацию о пользовании программой.

Справочная система разработана с помощью программного средства «HTML help workshop».

Справочная система должна вызываться путем выбора пункта меню «Справка» главной формы мню.

1. Тестирование программного средства

В процессе написания программного средства необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Тестирование программного средства – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами при условии, что функции отрабатывали на различных значениях.

Тестирование программного средства будет производиться последовательно, переходя из одной части программного средства в другую. Во время теста будут проверяться все действия с программным средством, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

После входа в программу пользователю будет представлена авторизационная форма. Для того, что бы войти в программу необходимо выбрать учетную запись, ввести пароль и нажать на кнопку «Ок». Окно авторизации представлено на рисунке 5.1.

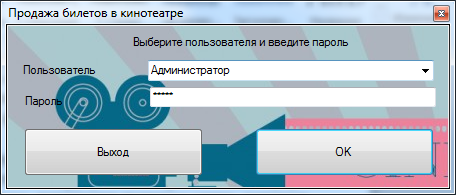


Рисунок 5.1 – Окно авторизации

При неправильном вводе пароля пользователю будет выведено соответствующее уведомление.

После выбора администратора и правильного ввода пароля будет открыта форма с меню. Внешний вид формы представлен на рисунке 5.2.

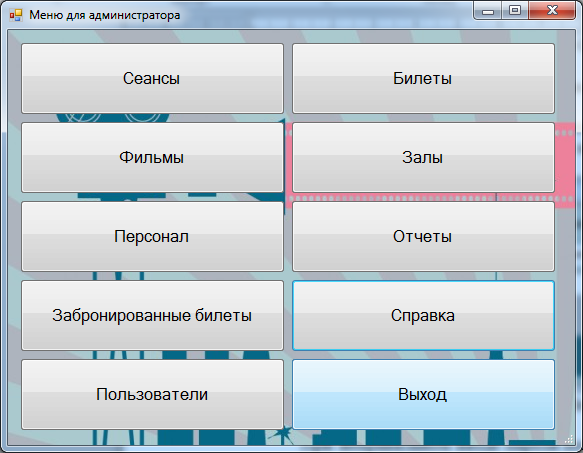


Рисунок 5.2 – Меню администратора

Для того, что бы посмотреть сеансы необходимо нажать на кнопку «Сеансы». Откроется форма, вид которой представлен на рисунке 5.3.

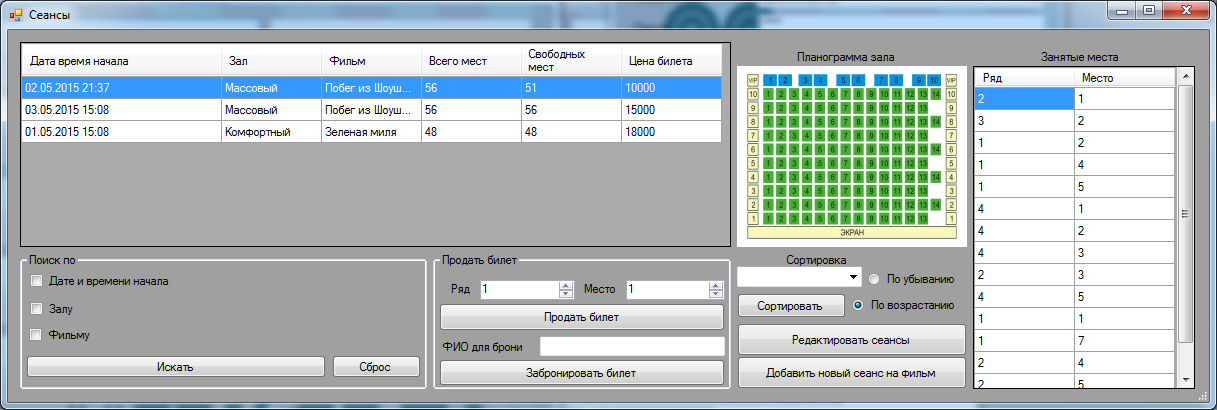


Рисунок 5.3 – Сеансы

Для того, что бы осуществить поиск необходимо выбрать критерии и ввести слова для поиска, после чего нажать на кнопку «Искать». Сбросить поиск можно нажав на кнопку «Сброс».

Для сортировки необходимо выбрать критерий сортировки, выбрать тип сортировки и нажать на кнопку «Сортировать».

Для продажи билета на сеанс необходимо ввести ряд и место, после чего нажать «Продать билет». В случае, если это место занято, пользователю будет выведено соответствующее уведомление. В случае, если ошибок нет, пользователю будет показан билет, вид которого представлен на рисунке 5.4.



Рисунок 5.4 – Билет

Для печати билета необходимо нажать на соответствующую кнопку на форме.

Для бронирования билета на сеанс необходимо ввести ряд, место и ФИО для брони, после чего нажать на кнопку «Забронировать билет».

Для редактирования сеансов необходимо нажать на кнопку «Редактирование сеансов». Будет открыта форма, на которой пользователь сможет изменить информацию о сеансах. Вид формы представлен на рисунке 5.5.

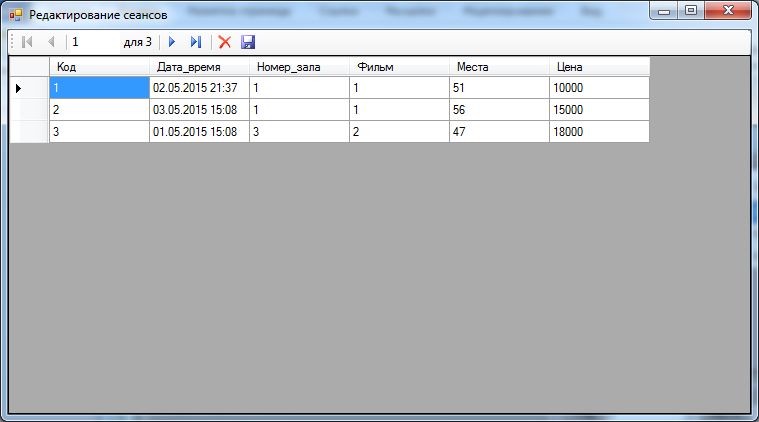


Рисунок 5.5 – Редактирование сеансов

Для добавления сеансов необходимо нажать на кнопку «Добавить новый сеанс на фильм». Будет открыта форма, на которой пользователь сможет внести информацию о сеансе, после чего необходимо нажать на кнопку «Добавить». Вид формы представлен на рисунке 5.6.

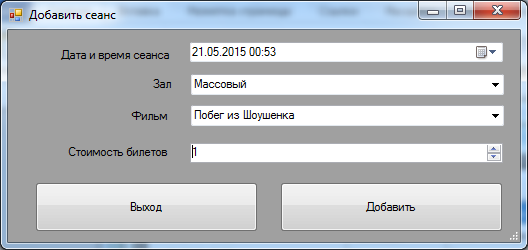


Рисунок 5.6 – Добавление сеанса

Для того, что бы посмотреть билеты необходимо выбрать на главном меню пункт «Билеты». Откроется форма, вид которой представлен на рисунке 5.7.

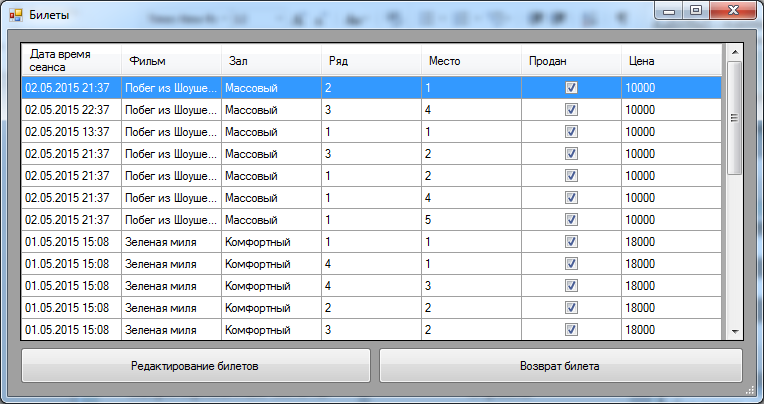


Рисунок 5.7 – Билеты

Для того, что бы осуществить возврат билета необходимо выбрать билет и нажать на кнопку «Возврат билета». Пользователю будет выведено уведомление о возврате.

Для редактирования билетов необходимо нажать на кнопку «Редактирование билетов». Будет открыта форма, на которой пользователь сможет изменить информацию о билетах. Вид формы представлен на рисунке 5.8.

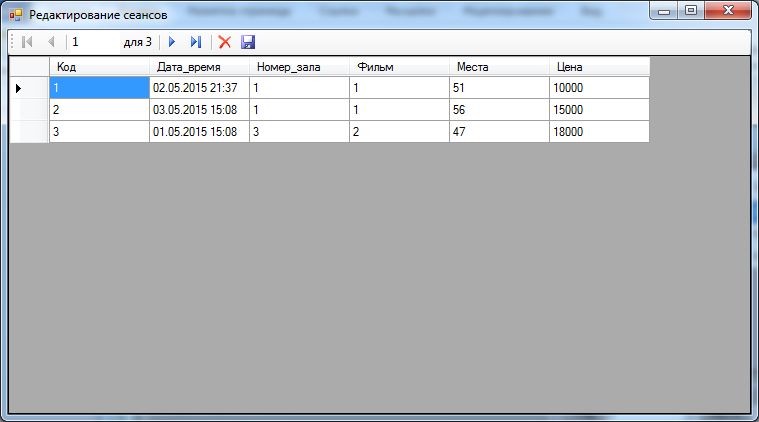


Рисунок 5.8 – Редактирование билетов

Для того, что бы посмотреть фильмы необходимо выбрать на главном меню пункт «Фильмы». Откроется форма, вид которой представлен на рисунке 5.9.

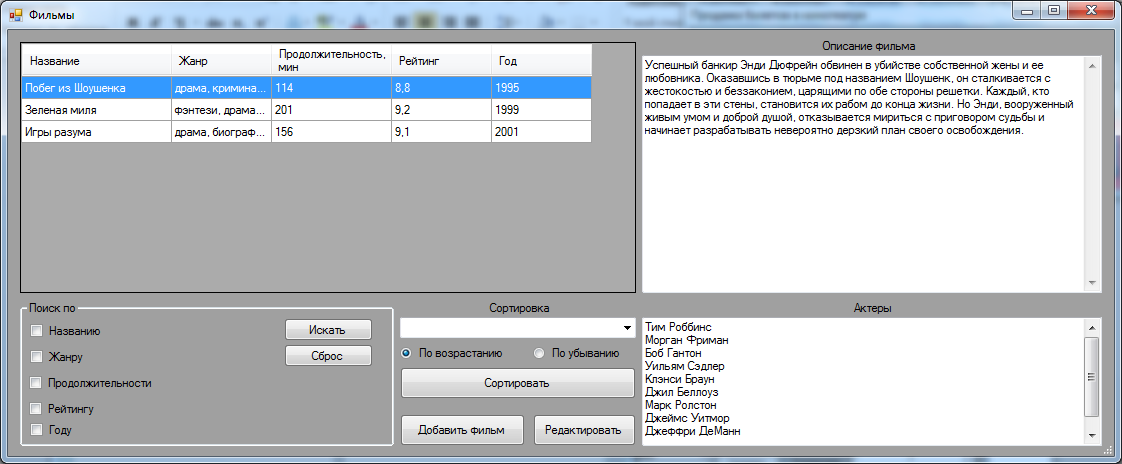


Рисунок 5.9 – Фильмы

Для того, что бы осуществить поиск необходимо выбрать критерии и ввести слова для поиска, после чего нажать на кнопку «Искать». Сбросить поиск можно нажав на кнопку «Сброс».

Для сортировки необходимо выбрать критерий сортировки, выбрать тип сортировки и нажать на кнопку «Сортировать».

Для редактирования фильмов необходимо нажать на кнопку «Редактировать». Будет открыта форма, на которой пользователь сможет изменить информацию о фильмах. Вид формы представлен на рисунке 5.10.

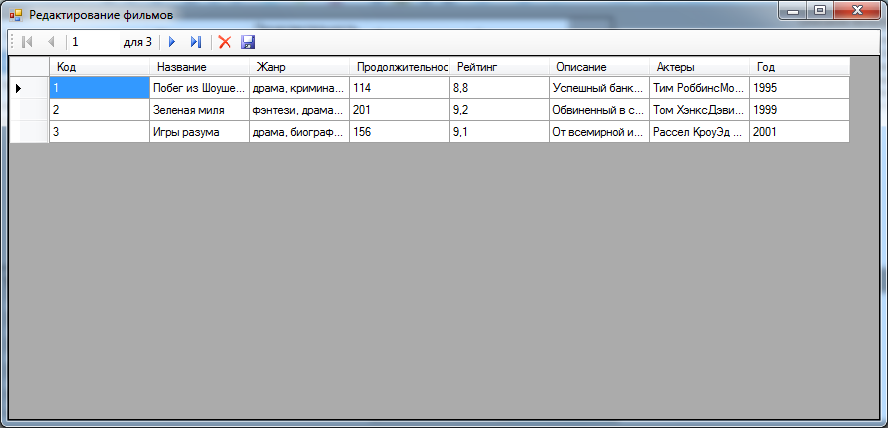


Рисунок 5.10 – Редактирование фильмов

Для добавления фильма необходимо нажать на кнопку «Добавить фильм». Будет открыта форма, на которой пользователь сможет внести информацию о фильме, после чего необходимо нажать на кнопку «Добавить». Вид формы представлен на рисунке 5.11.

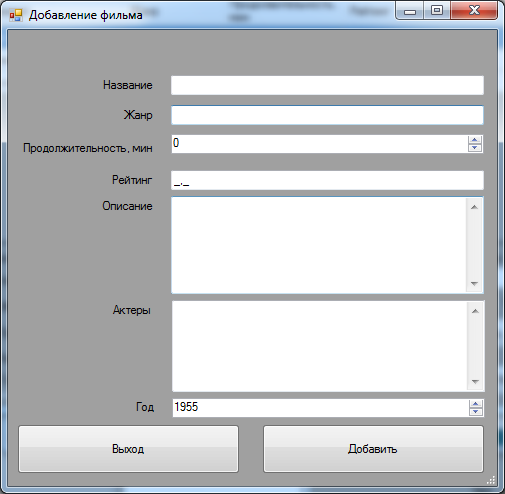


Рисунок 5.11 – Добавление фильма

Для того, что бы посмотреть залы необходимо выбрать на главном меню пункт «Залы». Откроется форма, вид которой представлен на рисунке 5.12.

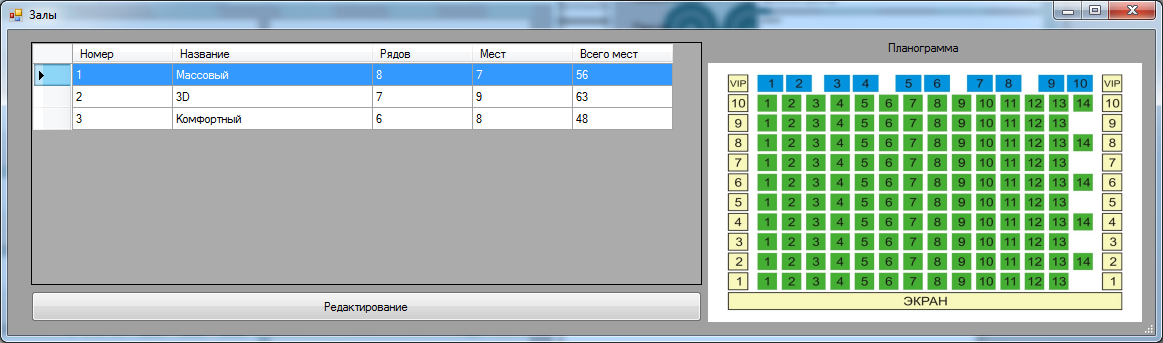


Рисунок 5.12 – Залы

Для редактирования залов необходимо нажать на кнопку «Редактирование». Будет открыта форма, на которой пользователь сможет изменить информацию о залах. Вид формы представлен на рисунке 5.13.

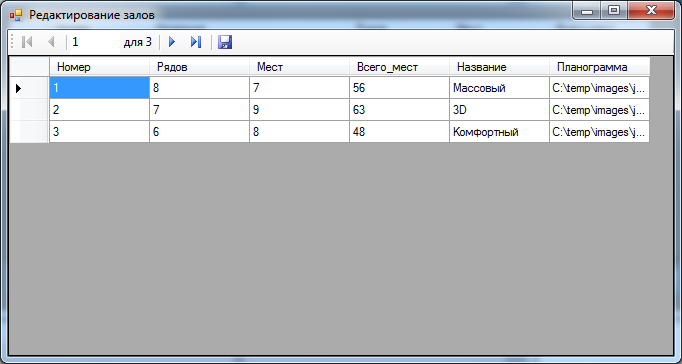


Рисунок 5.13 – Редактирование залов

Для того, что бы посмотреть персонал необходимо выбрать на главном меню пункт «Персонал». Откроется форма, вид которой представлен на рисунке 5.14.

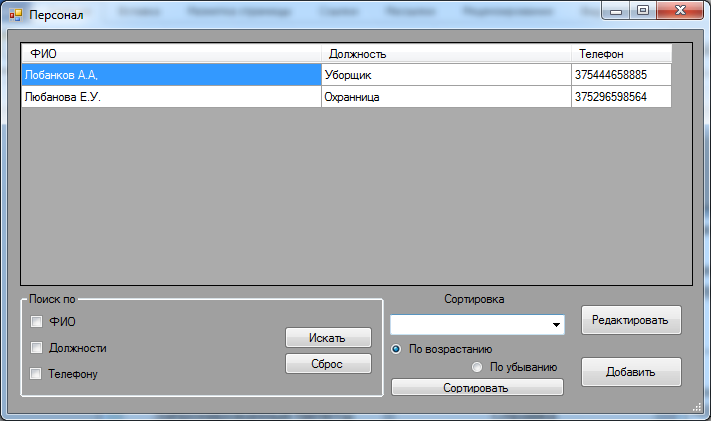


Рисунок 5.14 – Персонал

Для того, что бы осуществить поиск необходимо выбрать критерии и ввести слова для поиска, после чего нажать на кнопку «Искать». Сбросить поиск можно нажав на кнопку «Сброс».

Для сортировки необходимо выбрать критерий сортировки, выбрать тип сортировки и нажать на кнопку «Сортировать».

Для редактирования персонала необходимо нажать на кнопку «Редактировать». Будет открыта форма, на которой пользователь сможет изменить информацию о работниках. Вид формы представлен на рисунке 5.15.

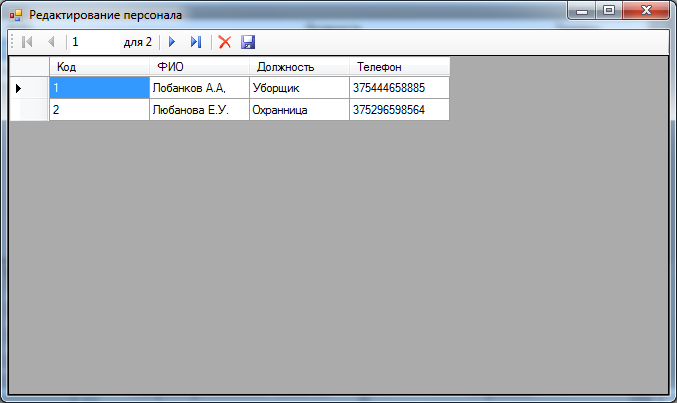


Рисунок 5.15 – Редактирование персонала

Для добавления работника необходимо нажать на кнопку «Добавить». Будет открыта форма, на которой пользователь сможет внести информацию о работнике, после чего необходимо нажать на кнопку «Добавить». Вид формы представлен на рисунке 5.16.

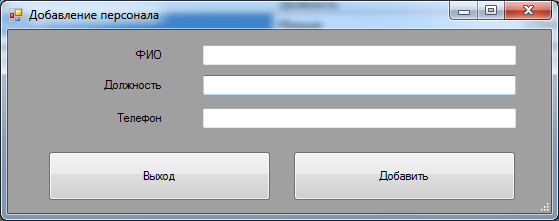


Рисунок 5.16 – Добавление работника

Для того, что бы посмотреть пользователей необходимо выбрать на главном меню пункт «Пользователи». Откроется форма, вид которой представлен на рисунке 5.17.

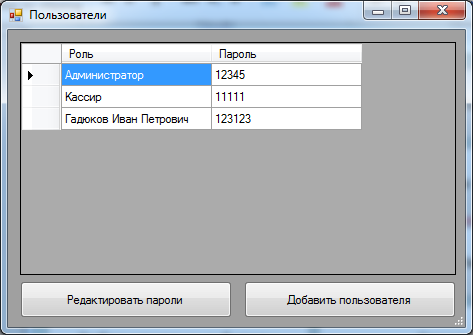


Рисунок 5.17 – Пользователи

Для редактирования паролей у пользователей необходимо нажать на кнопку «Редактировать пароли». Будет открыта форма, на которой пользователь сможет изменить пароли пользователей. Вид формы представлен на рисунке 5.18.

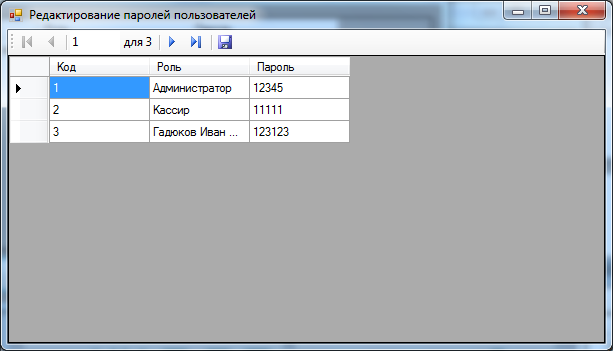


Рисунок 5.18 – Редактирование пользователей

Для добавления пользователей необходимо нажать на кнопку «Добавить». Будет открыта форма, на которой пользователь сможет внести информацию о пользователе, после чего необходимо нажать на кнопку «Добавить». Вид формы представлен на рисунке 5.19.

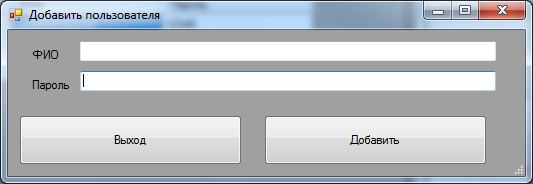


Рисунок 5.19 – Добавление пользователя

Для того, что бы посмотреть забронированные билеты необходимо выбрать на главном меню пункт «Билеты». Откроется форма, вид которой представлен на рисунке 5.20.

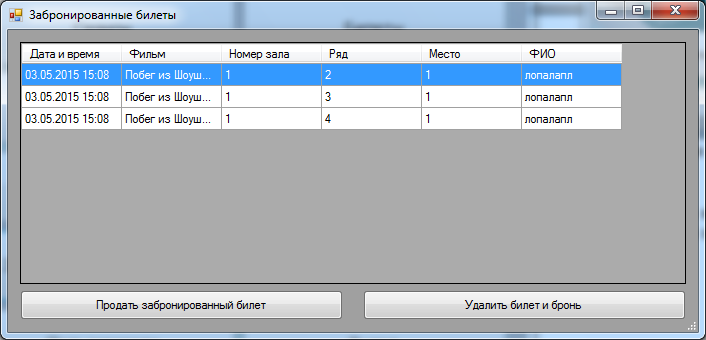


Рисунок 5.20 – Забронированные билеты

Для того, что бы продать забронированный билет необходимо выбрать билет и нажать на кнопку «Продать забронированный билет». После продажи будет открыта уже описанная ранее форма с билетом для печати.

Для удаления билета и брони на него необходимо нажать на кнопку «Удалить билет и бронь». После удаления пользователю будет выведено соответствующее уведомление.

Для того, что бы посмотреть отчеты необходимо выбрать на главном меню пункт «Отчеты». Откроется форма, вид которой представлен на рисунке 5.21.

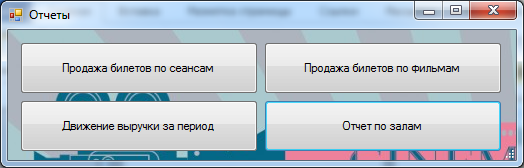


Рисунок 5.21 – Отчеты

Для просмотра отчета с продажами билетов по сеансам необходимо нажать на соответствующую кнопку. Для просмотра необходимо выбрать промежуток времени и нажать на кнопку «Показать». Вид формы представлен на рисунке 5.22.

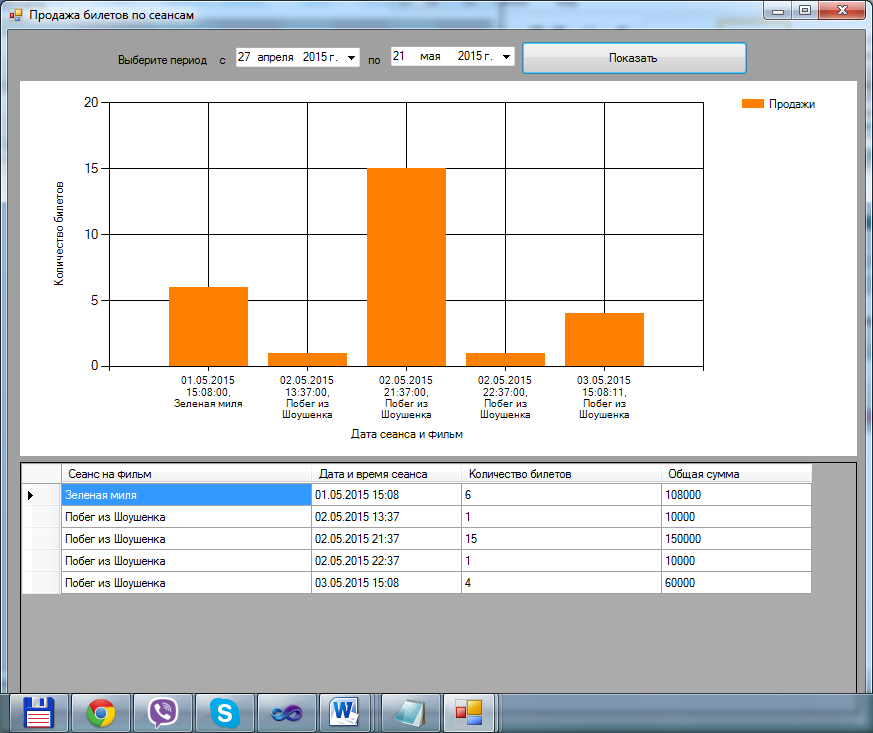


Рисунок 5.22 – Продажа билетов по сеансам

Для просмотра отчета с продажами билетов по фильмам необходимо нажать на соответствующую кнопку. Для просмотра необходимо выбрать промежуток времени и нажать на кнопку «Показать». Вид формы представлен на рисунке 5.23.

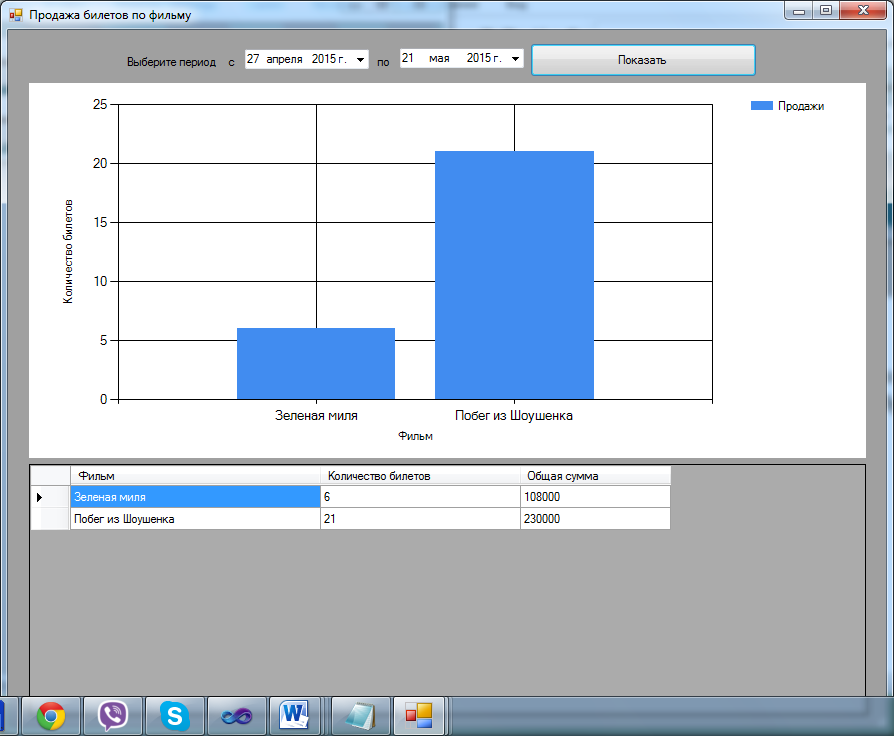


Рисунок 5.23 – Продажа билетов по фильмам

Для просмотра отчета с движением выручки необходимо нажать на соответствующую кнопку. Для просмотра необходимо выбрать промежуток времени и нажать на кнопку «Показать». Вид формы представлен на рисунке 5.24.

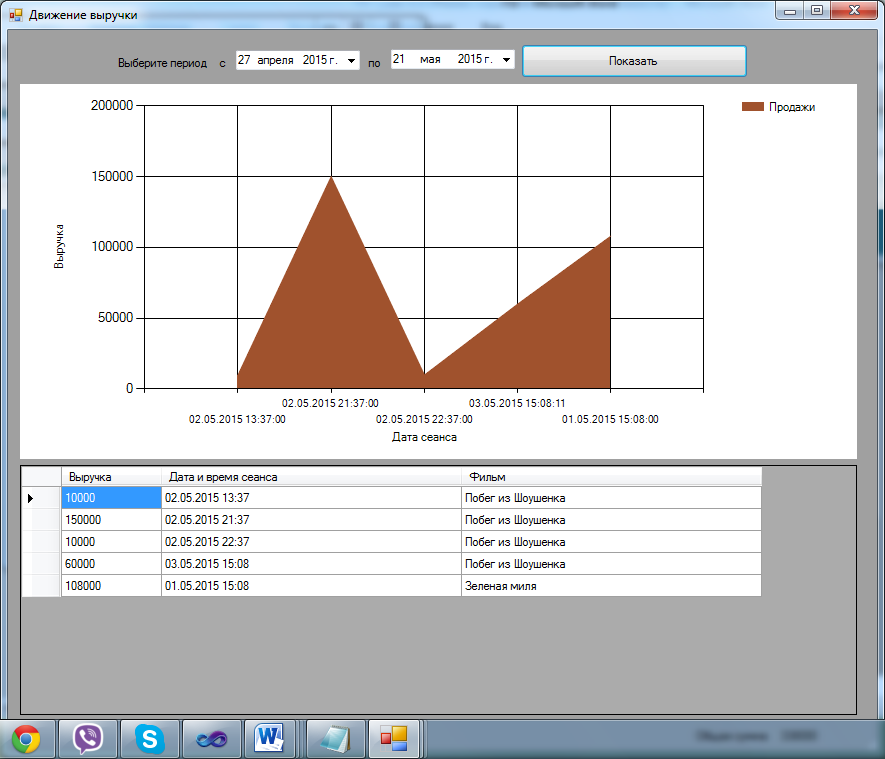


Рисунок 5.24 – Продажа билетов по фильмам

Для просмотра отчета по залам необходимо нажать на соответствующую кнопку. Для просмотра необходимо выбрать промежуток времени и нажать на кнопку «Показать». Вид формы представлен на рисунке 5.25.

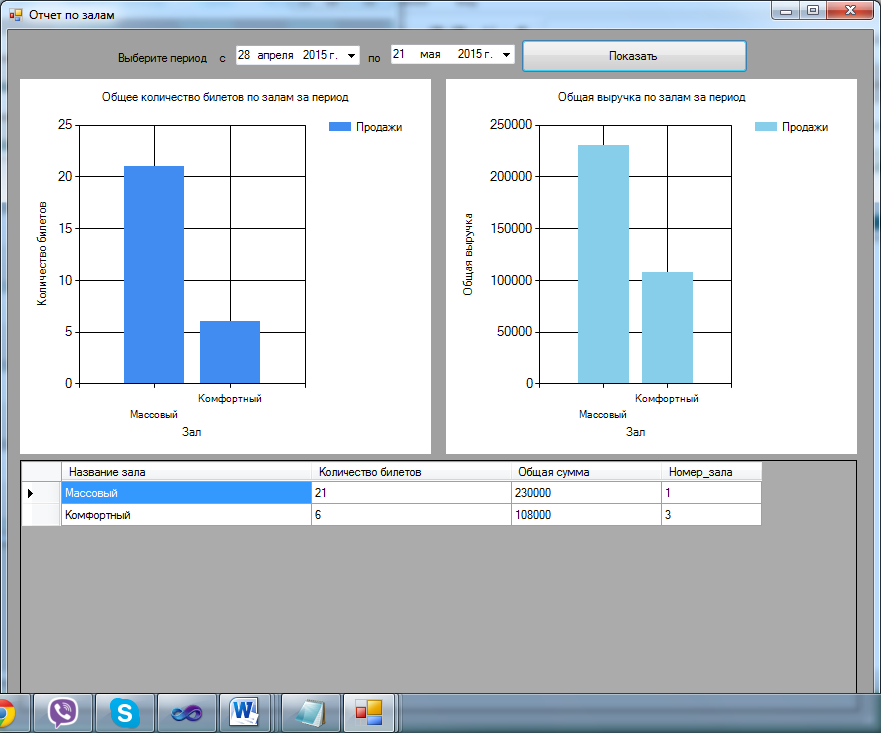


Рисунок 5.25 – Отчет по залам

После выбора режима кассира и правильного ввода пароля будет открыта форма с меню для кассира. Внешний вид формы представлен на рисунке 5.26.

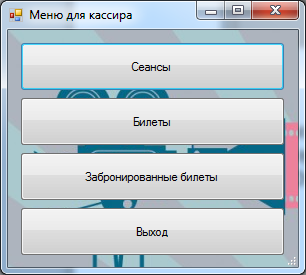


Рисунок 5.26 – Меню для кассира

Все функции кассира уже протестированы ранее.

1. Руководство по установке и использованию

Для применения данного программного средства необходимы следующие технические требования:

* процессор Pentium 800 или выше;
* минимальный объем оперативной памяти 128 Мб;
* операционная система Windows XP и выше;
* пакет обновлений Framework v4.0;
* рекомендуется монитор типа VGA или с лучшей разрешающей способностью;
* клавиатура;
* мышь.

Компьютер должен работать под управлением операционной системы, начиная с Windows XP и выше. Наиболее удобной операционной системой для проведения испытаний является Windows 7, так как она ориентирована на максимальное использование всех возможностей ПК, сетевых ресурсов и обеспечение комфортных условий работы.

Программа имеет небольшой размер, оптимизирована для быстрого запуска и минимальной загрузки системы. Доступный дружественный интерфейс обеспечивает наглядность простоты в управлении программы и не содержит сложных настроек.

Для начала работы с программным средством необходимо запустить файл «Kino.exe», размер которого составляет 549КБ. Для функционирования программного средства необходимо установить платформу .NET Framework 4.0. Программное средство не нуждается в установке.

Заключение

В рамках курсового проектирования было разработано «Программное средство для автоматизации работы кассира в кинотеатре».

Программное средство имеет ряд достоинств: простой и понятный интерфейс, небольшой объем памяти, занимаемый приложением на различных носителях информации, удобно для быстрой обработки информации.

В программе реализованы такие задачи, как: редактирование данных, фильтрация данных, формирование отчетов, продажа и бронирование билетов на сеансы.

Программа реализована в полном объеме и в соответствии с заданными требованиями. Полностью отлажена и протестирована. Поставленные задачи выполнены.

В программном средстве, благодаря среде разработки Microsoft Visual Studio 2010, был разработан удобный, понятный и простой в использовании интерфейс.

Программа реализована полностью в соответствии с поставленной задачей курсового проекта. Приложение было протестировано и отлажено.

Проект был реализован с помощью среды разработки Microsoft Visual Studio 2010, с использованием языка C# и системы управления базами данных Microsoft SQL Server 2008.

Программное средство готово к практическому использованию. Данное приложение может быть дополнено и модернизировано.

Список литературы

1 Багласова Т.Г. Методические указания по выполнению дипломного проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий».– Мн.: ТБП, 2003

2 Багласова Т.Г. Методические указания по оформлению курсовых и дипломных проектов.- Мн.: ТБП, 2006

3 Кунец И.Е. Объектно – ориентированный анализ и проектирование. Работа в среде Rational Rose. Методическое пособие для учащихся ССУЗ по специальности 2-40 01 01 «Программное обеспечение информационных технологий».– Мн.: ТБП, 2006

4 МаккиА.И. VisualStudio 2010 для профессионалов. –СПб.: Питер, 2010

5 КарповР.Н. Microsoft Office 2007 – Мн.: Бином, 2007

6 Томров А.К. RationalRose для начинающих. – Мн.: Бином, 2005

7 Хернандес М. Вьесках. Дж. SQL. – СПб.: Питер, 2001

8 Петцольд Ч. Программирование для MicrosoftWindows на C# – М.: Издательско-торговый дом «Русская редакция», 2002

9 Холецкий В.И. Windows7операционная система. – СПб.: Питер, 2011

10 ГОСТ 19.301-2000 ЕСПД. Программа и методика испытаний

11 ГОСТ 19.401-2000 ЕСПД. Текст программы

12 ГОСТ 19.402-2000 ЕСПД. Описание программы

13 ГОСТ2.105-95 ЕСКД. Общие требования к текстовым документам

Листинг

Перечень основных функций

На данном этапе были выделены и реализованы функции, позволяющие решить задачи, обеспечивающие достижение цели курсового проекта.

Функция продажи билета на сеанс осуществляется в момент нажатия на кнопку «Продать билет» формы с сеансами и вызывается методом «AddBilet()». При выполнении данной функции будут формироваться запросы, добавляться в базу информация о новом билете, уменьшение свободных мест в сеансе. Так же эта функция служит для бронирования билетов. Текст данной функции представлен ниже.

private void addBilet(bool bronirovanie)

{

if (билетDataGridView.RowCount > 0)

{

for (int i = 0; i < билетDataGridView.RowCount; i++)

{

if (Convert.ToInt32(билетDataGridView.Rows[i].Cells["dataGridViewTextBoxColumn15"].Value) == Convert.ToInt32(numericUpDown2.Value) &&

Convert.ToInt32(билетDataGridView.Rows[i].Cells["dataGridViewTextBoxColumn12"].Value) == Convert.ToInt32(numericUpDown1.Value))

{

MessageBox.Show("Это место уже занято");

return;

}

}

}

byte kuplen = 1;

if (bronirovanie)

{

kuplen = 0;

}

билетTableAdapter.Insert(Convert.ToDateTime(сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn2"].Value.ToString()), Convert.ToInt32(сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn1"].Value.ToString()), Convert.ToString((сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn4"] as DataGridViewComboBoxCell).FormattedValue.ToString()), Convert.ToInt32(сеансDataGridView.CurrentRow.Cells["Номер\_зала"].Value.ToString()), Convert.ToInt32(numericUpDown1.Value), kuplen, Convert.ToInt32(сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn6"].Value.ToString()), Convert.ToInt32(numericUpDown2.Value));

string new\_id = "";

string sql = "SELECT top 1 \* from Билет order by Код DESC";

using (SqlConnection c = new SqlConnection(Connection))

{

c.Open();

using (SqlDataAdapter a = new SqlDataAdapter(sql, c))

{

DataTable t = new DataTable();

a.Fill(t);

new\_id = t.Rows[0]["Код"].ToString();

}

}

try

{

using (SqlConnection con = new SqlConnection(Connection))

{

con.Open();

using (SqlCommand command = new SqlCommand("Update Сеанс set Места = Места-1 where Код=" + сеансDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn1"].Value.ToString(), con))

{

command.ExecuteNonQuery();

}

this.сеансTableAdapter.Fill(this.kinoDataSet.Сеанс);

con.Close();

}

}

catch (SystemException ex)

{

MessageBox.Show(string.Format("An error occurred: {0}", ex.Message));

}

if (bronirovanie)

{

броньTableAdapter.Insert(Convert.ToInt32(new\_id), textBox1.Text);

MessageBox.Show("Билет забронирован");

}

else {

MessageBox.Show("Билет продан");

printBilet(Convert.ToInt32(new\_id));

}

this.билетTableAdapter.Fill(this.kinoDataSet.Билет);

}

Функция поиска по таблице сеансов обрабатывается по нажатию кнопки «Искать» после выбора критериев поиска и ввода слов для поиска. Данная функция вызывается методом «button4\_Click», закрепленным за событием «Click». Текст данной функции представлен ниже.

private void button4\_Click(object sender, EventArgs e)

{

string filter = "Код = Код "; //задаем строку для поиска

bool error = false;

if (checkBox1.Checked || checkBox2.Checked || checkBox3.Checked) // если хотя бы один вид выбрано

{

if (checkBox1.Checked)

{

filter += String.Format("and Дата\_время >= '{0:yyyy-MM-dd}' AND Дата\_время < '{1:yyyy-MM-dd}'", dateTimePicker1.Value.ToShortDateString(), dateTimePicker2.Value.AddDays(1).ToShortDateString());

}

if (checkBox3.Checked)

{

filter += " and Номер\_зала = " + comboBox2.SelectedValue + " ";

}

if (checkBox2.Checked)

{

filter += " and Фильм = " + comboBox3.SelectedValue + " ";

}

if (error)

{

MessageBox.Show("Заполните поле поиска для выбранного критерия!");

return;

}

else

{

this.сеансBindingSource.Filter = filter; //фильтрация

}

}

else

MessageBox.Show("Выберите хотя бы один критерий поиска!");

}

Функция возврата билета на сеанс осуществляется в момент нажатия на кнопку «Вернуть билет» на форме с билетами. Данная функция вызывается методом «button2\_Click», закрепленным за событием «Click». Текст данной функции представлен ниже.

private void button2\_Click(object sender, EventArgs e)

{

int id = Convert.ToInt32(билетDataGridView.CurrentRow.Cells["dataGridViewTextBoxColumn1"].Value);

try

{

using (SqlConnection con = new SqlConnection(Connection))

{

con.Open();

using (SqlCommand command = new SqlCommand("Delete from Билет where Код=" + id, con))

{

command.ExecuteNonQuery();

MessageBox.Show("Билет возвращен");

this.билетTableAdapter.Fill(this.kinoDataSet.Билет);

}

con.Close();

}

}

catch (SystemException ex)

{

MessageBox.Show(string.Format("An error occurred: {0}", ex.Message));

}

}

Функция печати билета на сеанс осуществляется в момент нажатия на кнопку «Печать» на форме с печатью билета после его продажи. Данная функция вызывается методом «button2\_Click», закрепленным за событием «Click». Текст данной функции представлен ниже.

private void CaptureScreen()

{

Graphics myGraphics = this.CreateGraphics();

Size s = this.Size;

memoryImage = new Bitmap(s.Width, s.Height-20, myGraphics);

Graphics memoryGraphics = Graphics.FromImage(memoryImage);

memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, s);

}

Диаграмма классов

