**ОГЛАВЛЕНИЕ**

**ВВЕДЕНИЕ**

**1 Анализ предметной области и технологий Разработки ИНФОРМАЦИОННОЙ СИСТЕМЫ дляя станции переливания крови…………………………………………………………2**

1.1 Анализ организационной структурыстанции переливания крови…2

1.2 Описание сущностей и определение ключевых полей, установление связей между таблицами и обеспечение целостности данных………………….5

**2 Описание процесса разработки…………………………….6**

2.1 Определение типа атрибутов…………………………………………..6

2.2 Заполнение таблиц………………………………………………………12

2.3 Схема данных……………………………………………………………13

2.4 Создание запросов………………………………………………………14

2.5 Создание отчетов………………………………………………………17

2.6 Создание и применение форм………………………………………….18

2.7 Тестирование программы………………………………………………29

**3 ИНФОМАЦИОННАЯ БЕЗОПАСНОСТЬ………………………….35**

3.1 Методы защиты информационной базы………………………………35

**4 Экономическое обоснование……………………………36**

4.1 Планирование разработки программного продукта «Станция переливания крови»…………………………………………………………….….36

4.2 Стоимость разработки………………………………………………….38

4.3 Расчет экономической эффективности программного продукта…….41

**ЗАКЛЮЧЕНИЕ…………………………………………………………….44**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ………………….45**

**ВВЕДЕНИЕ**

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется [системой управления базами данных (СУБД)](https://www.oracle.com/cis/database/what-is-database/#WhatIsDBMS). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL).

Существует множество различных типов баз данных. Выбор наилучшей базы данных для конкретной компании зависит от того, как она намеревается использовать данные.

Для базы данных обычно требуется комплексное программное обеспечение, которое называется системой управления базами данных (СУБД). СУБД служит интерфейсом между базой данных и пользователями или программами, предоставляя пользователям возможность получать и обновлять информацию, а также управлять ее упорядочением и оптимизацией. СУБД обеспечивает контроль и управление данными, позволяя выполнять различные административные операции, такие как мониторинг производительности, настройка, а также резервное копирование и восстановление.

Современные крупные корпоративные базы данных нередко поддерживают очень сложные запросы, и предполагается, что они должны предоставлять почти мгновенные ответы на них. В результате администраторы баз данных вынуждены применять самые разные методы для повышения производительности.

Объектом для дипломной работы является тема «Станция переливания крови». Целью создания данного программного продукта является автоматизированная система, для упрощения учета переливания крови.

Будущая база данных должна быть проста в использовании, чтобы её мог использовать любой пользователь с минимальными знаниями и умениями в компьютере.

**Актуальность:** тема дипломной работы актуальна, так как очень ценятся практические умения в данной среде.

**Цель:** Разработка базы данных Microsoft SQL Server по теме «Станция переливания крови».

**Задачи:**

1. Изучить теоретическую часть по базам данных;
2. Изучить программы Microsoft SQL Server и Visual Studio;
3. Изучить тему «Станция переливания крови»;
4. Создать структуру базы данной;
5. Создать и заполнить таблицы;
6. Создать логические запросы;
7. Создать формы;
8. Сделать программу с простым и понятным интерфейсом.

**1. Анализ предметной области и технологий Разработки ИНФОРМАЦИОННОЙ ПОДСИСТЕМЫ ДЛЯ СТАНЦИИ ПЕРЕЛИВАНИЯ КРОВИ**

**1.1 Анализ организационной структуры станции переливания крови**

Ста́нция перелива́ния кро́ви — базовое медицинское учреждение [Службы крови](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B1%D0%B0_%D0%BA%D1%80%D0%BE%D0%B2%D0%B8). По своему расположению бывают республиканские (РСПК), краевые (КСПК), областные (ОСПК), городские (ГСПК) станции.

Если [донора](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BD%D0%BE%D1%80%D1%81%D1%82%D0%B2%D0%BE_%D0%BA%D1%80%D0%BE%D0%B2%D0%B8) не допустили к процедуре сдаче крови по медицинским показаниям, ему выдаётся «Справка донору об обследовании». Этот документ после заполнения является основанием для освобождения донора от [работы](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%83%D0%B4) на время, затраченное им на обследование в учреждении [Службы крови](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B1%D0%B0_%D0%BA%D1%80%D0%BE%D0%B2%D0%B8), с сохранением средней [заработной платы](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D0%B0).

Если донора допустили к сдаче крови, плазмы и других компонентов крови, но по техническим причинам не взяли кровь, ему выдаётся такая же справка.

Если донора допустили к сдаче крови и процедура успешно проведена, то ему выдаётся «Справка донору об освобождении от работы в день кроводачи и предоставлении ему дополнительного дня отдыха». По желанию этот день отдыха может быть присоединён к ежегодному [отпуску](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BF%D1%83%D1%81%D0%BA) или использован в другое время в течение [года](https://ru.wikipedia.org/wiki/%D0%93%D0%BE%D0%B4) после дня сдачи крови или её компонентов.

В случае, если в день сдачи крови донор вышел на работу, согласовав это с руководителем, ему должны предоставить другой день отдыха. При сдаче крови за донором сохраняются средний заработок за все дни сдачи и предоставленные в связи с этим дни отдыха. В день сдачи крови донора обеспечивают бесплатным питанием, либо выдают деньги на питание, или продуктовый набор на эту же сумму.

Кроме того, каждому донору выдаётся [медицинская страховка](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B4%D0%B8%D1%86%D0%B8%D0%BD%D1%81%D0%BA%D0%BE%D0%B5_%D1%81%D1%82%D1%80%D0%B0%D1%85%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) за счёт средств Службы крови.

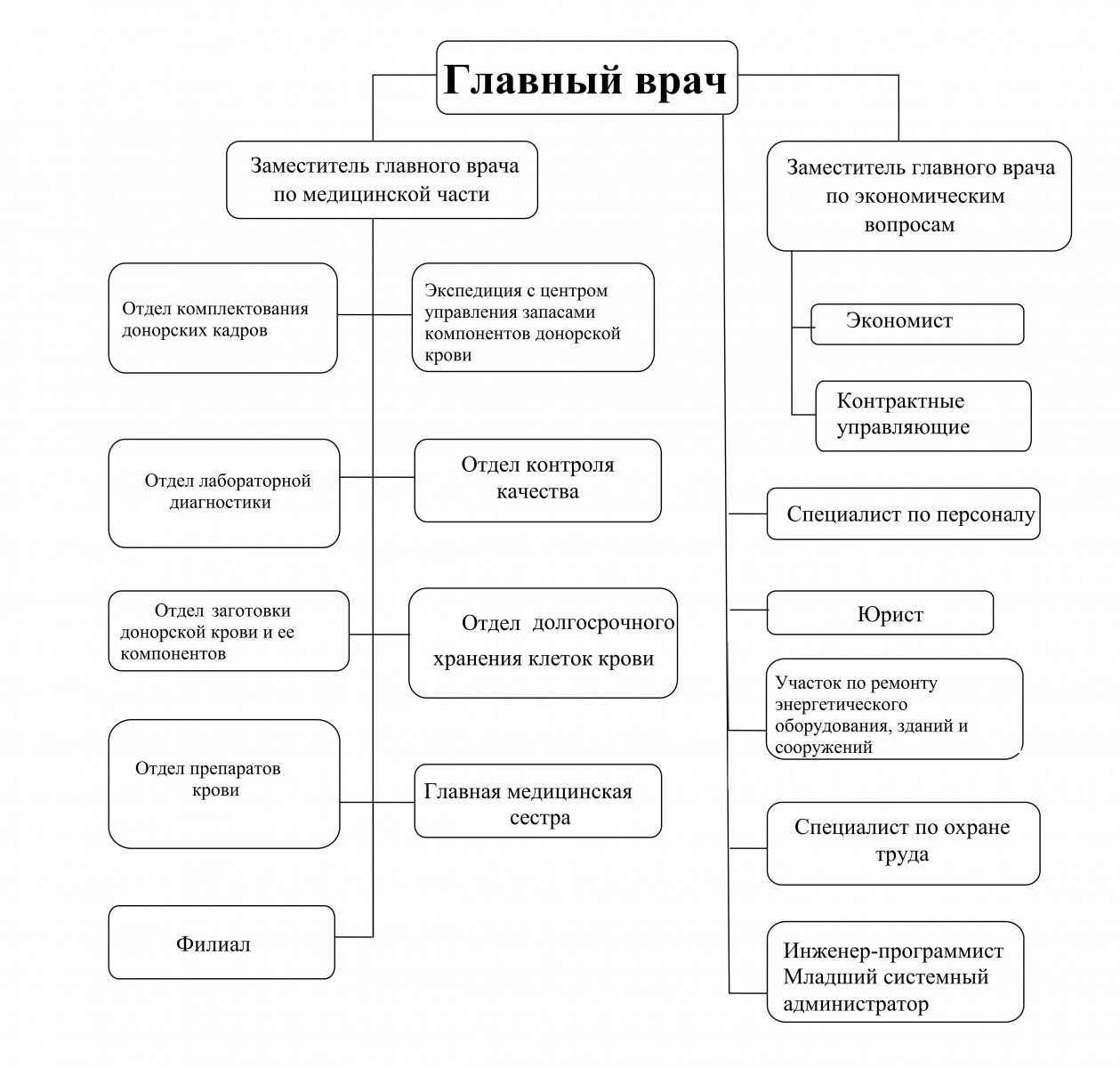


Рис. 1 Структура предприятия

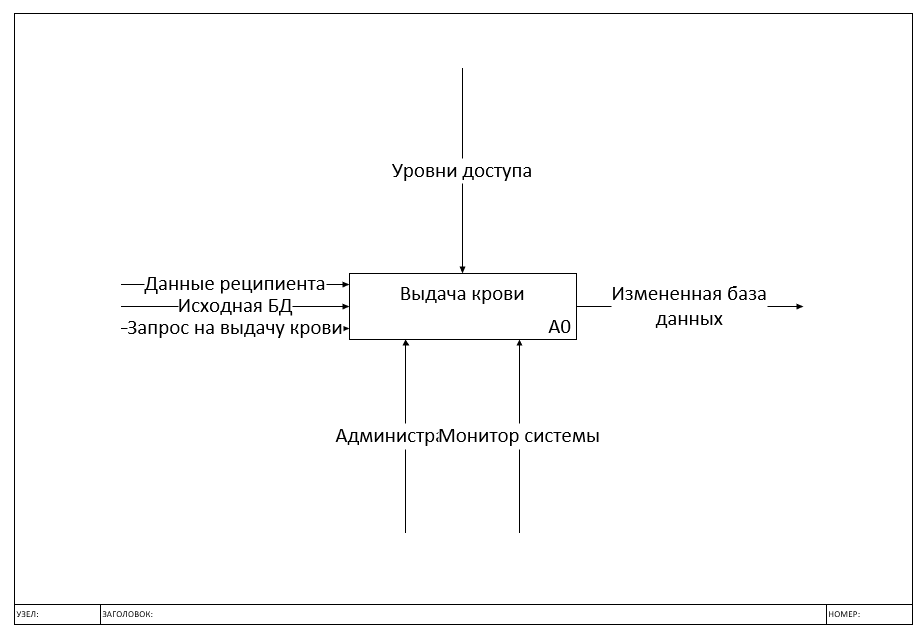


Рис. 2 Контекстная диаграмма предприятия

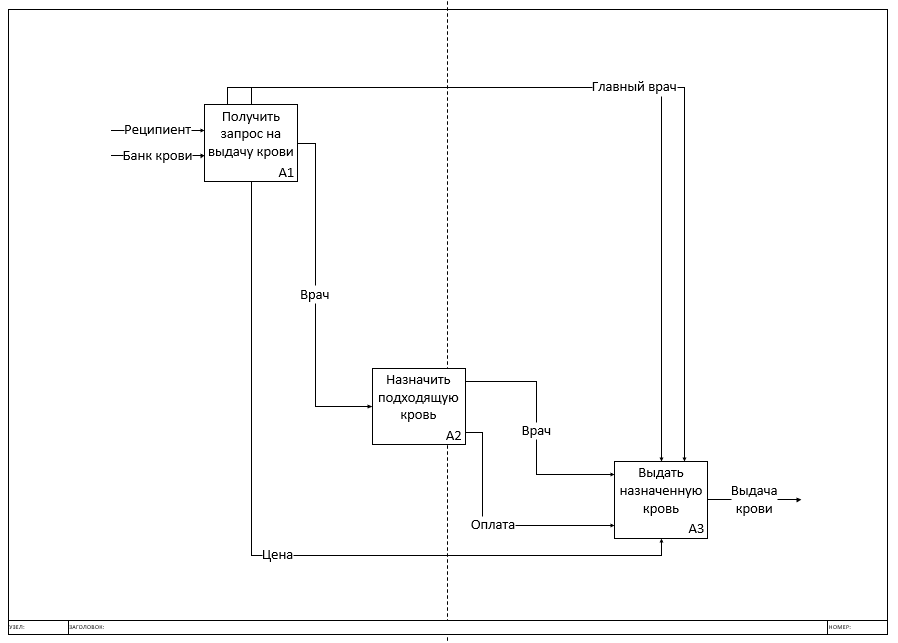


Рис. 3 Диаграмма декомпозиции уровень A0

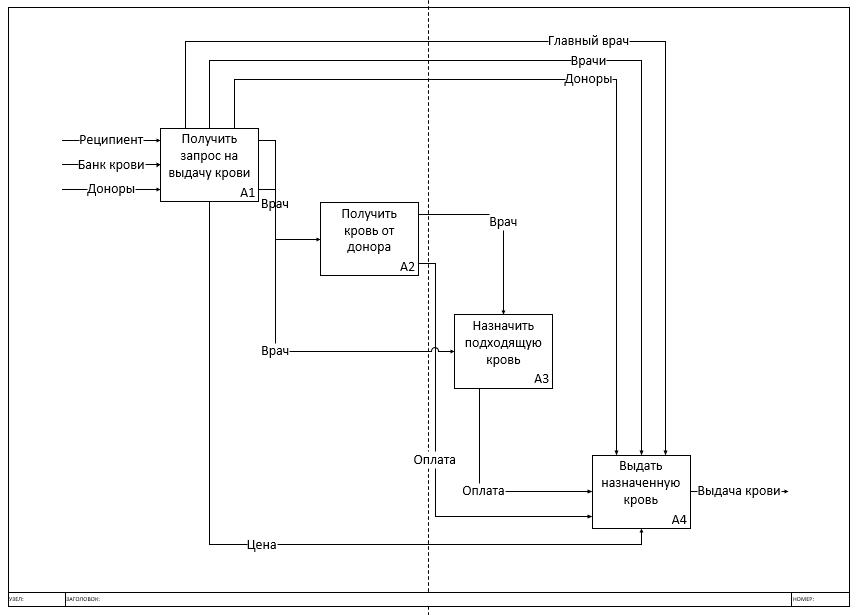


Рис. 4 Диаграмма декомпозиции уровень A2

**1.2** **Описание сущностей и определение ключевых полей, установление связей между таблицами и обеспечение целостности данных**

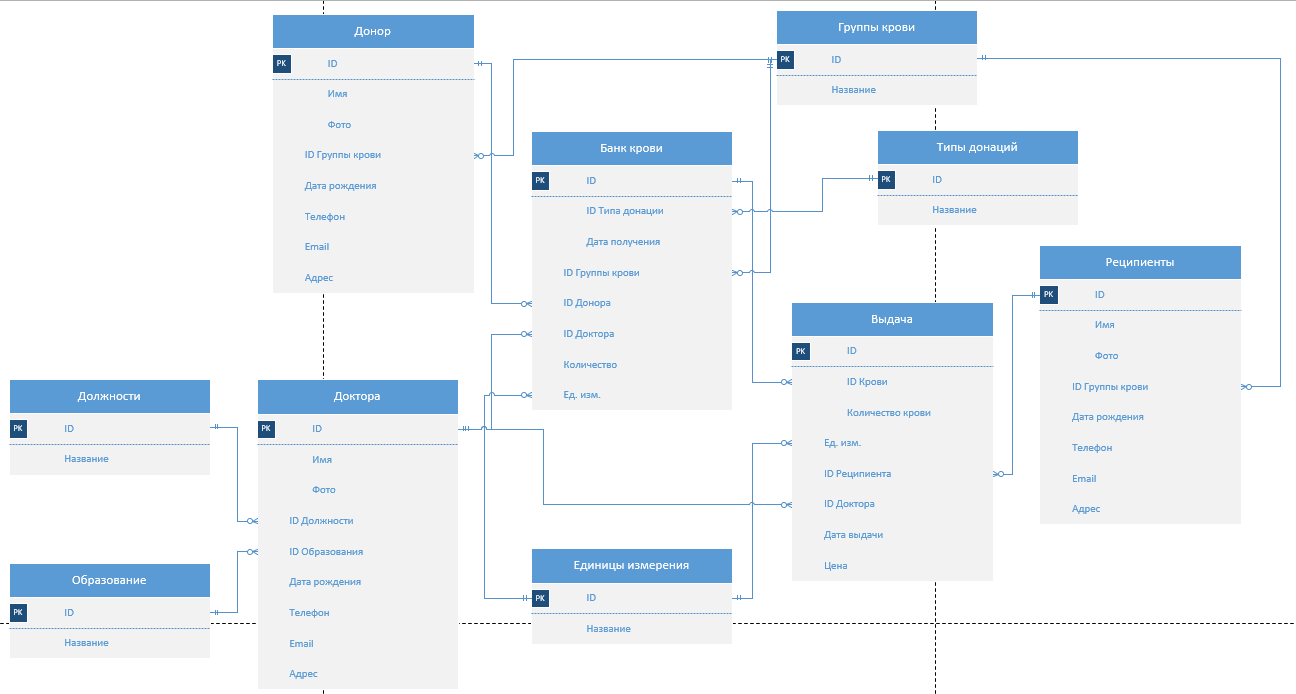


Рис. 5 ERD

**2. Описание процесса разработки**

**2.1 Определение типа атрибутов**

Таблица 1

Таблица «Банк крови»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| ID | int |
| Тип донации (ID) | int |
| Донор (ID) | int |
| Группа крови (ID) | int |
| Количество | float |
| Единица измерения | nvarchar |
| Дата получения | nvarchar |
| Врач (ID) | int |

Таблица 2

Таблица «Врачи»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| (ID) | int |
| Имя | nvarchar |
| Фото | nvarchar |
| Роль (ID) | int |
| Дата рождения | nvarchar |
| Телефон | nvarchar |
| Почта | nvarchar |
| Адрес | nvarchar |

Таблица 3

Таблица «Роли»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| (ID) | int |
| Название | nvarchar |

Таблица 4

Таблица «Доноры»

|  |  |
| --- | --- |
| ID | int |
| Имя | nvarchar |
| Фото | nvarchar |
| Группа крови | int |
| Дата рождения | nvarchar |
| Телефон | nvarchar |
| Почта | nvarchar |
| Адрес | nvarchar |

Таблица 5

Таблица «Пациенты»

|  |  |
| --- | --- |
| ID | int |
| Имя | nvarchar |
| Фото | nvarchar |
| Группа крови | int |
| Дата рождения | nvarchar |
| Телефон | nvarchar |
| Почта | nvarchar |
| Адрес | nvarchar |

Таблица 6

Таблица «Выдача»

|  |  |
| --- | --- |
| ID | int |
| Реципиент (ID) | int |
| Кровь (ID) | int |
| Количество крови | float |
| Единица измерения | nvarchar |
| Цена | money |
| Врач (ID) | int |
| Дата выдачи | nvarchar |

Таблица 7

Таблица «Типы донаций»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| (ID) | int |
| Название | nvarchar |

Таблица 8

Таблица «Группы крови»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| (ID) | int |
| Название | nvarchar |

Таблица 9

Таблица «Образование»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| (ID) | int |
| Название | nvarchar |

Таблица 10

Таблица «Единицы измерения»

|  |  |
| --- | --- |
| Название столбца | Тип данных |
| (ID) | int |
| Название | nvarchar |

**2.2 Заполнение таблиц**

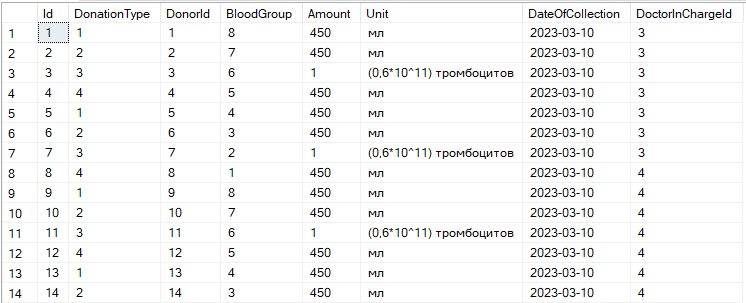


Рис. 6 Таблица «Банк крови»

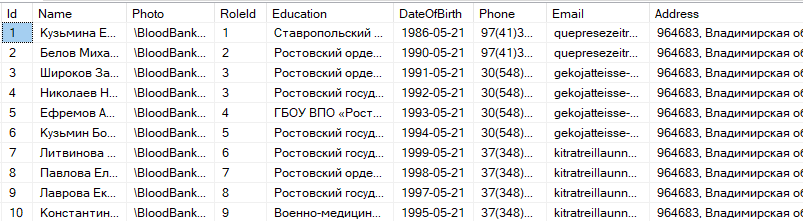


Рис. 7 Таблица «Врачи»

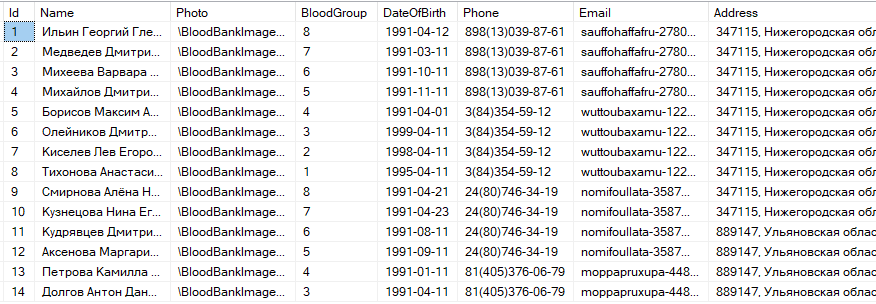


Рис. 8 Таблица «Доноры»

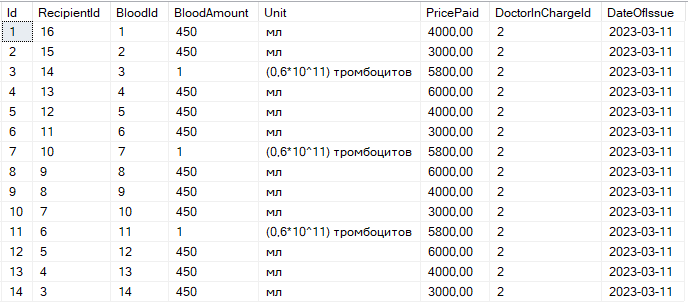


Рис. 9 Таблица «Выдача»

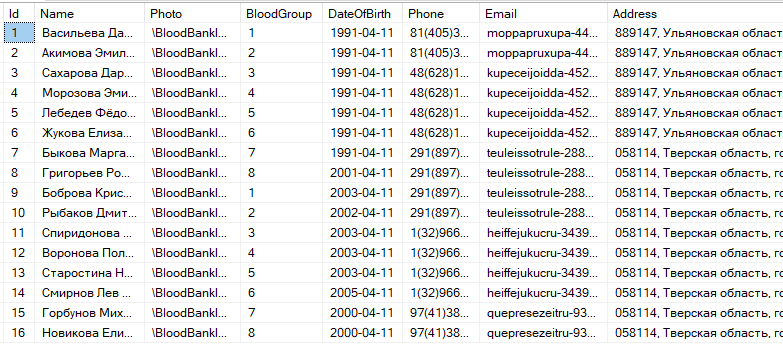


Рис. 10 Таблица «Пациенты»

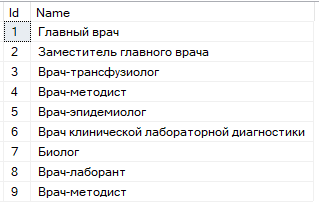


Рис. 11 Таблица «Роли»

**2.3 Схема данных**

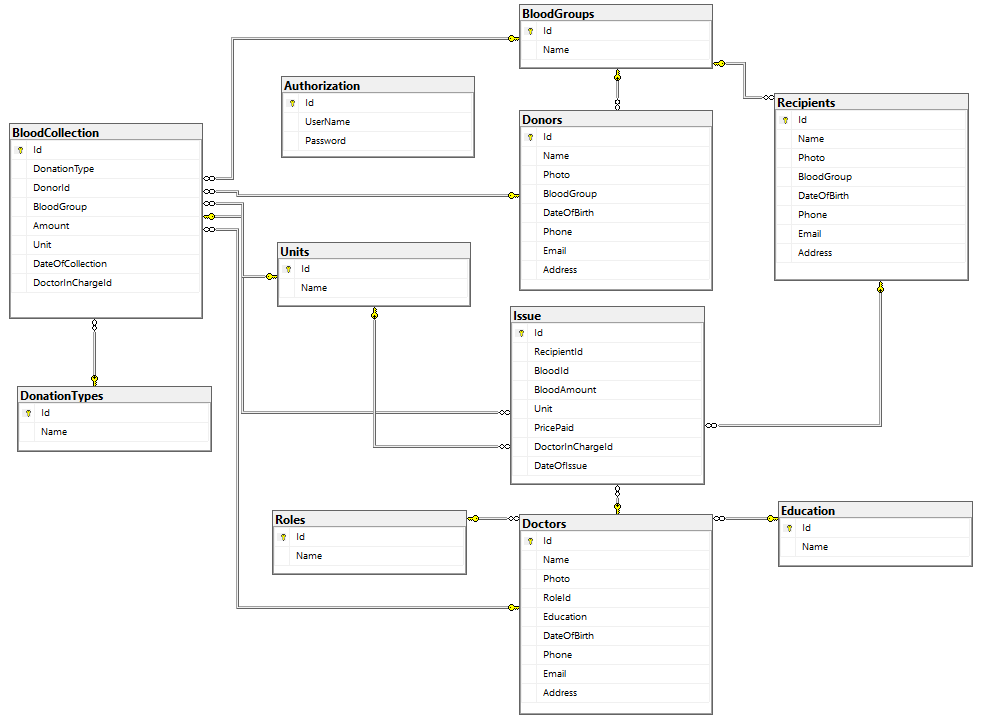


Рис. 12 Схема данных

**2.4 Создание запросов**

Запрос 1. Удаление из BloodCollection

ALTER PROCEDURE [dbo].[spBloodCollection\_DeleteById]

@Id int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

delete from dbo.BloodCollection where Id = @Id;

END

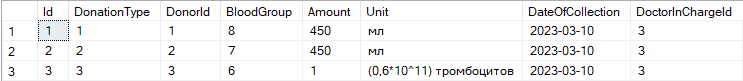


Рис. 13 Удаление из BloodCollection

Запрос 2. Выборка из BloodCollection

ALTER PROCEDURE [dbo].[spBloodCollection\_GetByIssueId]

@IssueId int

AS

BEGIN

SET NOCOUNT ON;

select dbo.BloodCollection.Id, dbo.BloodCollection.DonationType, dbo.BloodCollection.DonorId, dbo.BloodCollection.BloodGroup, dbo.BloodCollection.Amount, dbo.BloodCollection.Unit, dbo.BloodCollection.DateOfCollection, dbo.BloodCollection.DoctorInChargeId

from dbo.BloodCollection join dbo.Issue on dbo.BloodCollection.Id = dbo.Issue.BloodId

where dbo.Issue.Id = @IssueId;

END

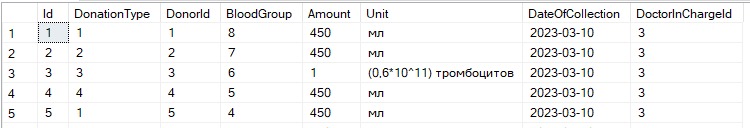


Рис. 14 Выборка из BloodCollection

Запрос 3. Вставка в BloodCollection

ALTER PROCEDURE [dbo].[spBloodCollection\_Insert]

@DonationType int,

@DonorId int,

@BloodGroup int,

@Amount float,

@Unit nvarchar(max),

@DateOfCollection nvarchar(max),

@DoctorInChargeId int,

@Id int = 0 output

AS

BEGIN

SET NOCOUNT ON;

insert into dbo.BloodCollection (DonationType, DonorId, BloodGroup, Amount, Unit, DateOfCollection, DoctorInChargeId)

values (@DonationType, @DonorId, @BloodGroup, @Amount, @Unit, @DateOfCollection, @DoctorInChargeId);

select @Id = SCOPE\_IDENTITY();

END



Рис. 15 Вставка в BloodCollection

Запрос 4. Обновление BloodCollection

ALTER PROCEDURE [dbo].[spBloodCollection\_UpdateAmountById]

@Amount float,

@Id int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

update dbo.BloodCollection

set Amount = @Amount

where Id = @Id;

END

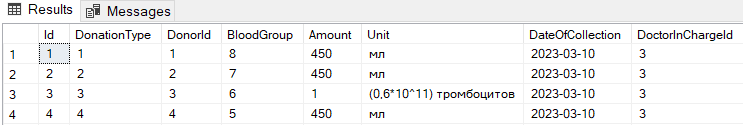


Рис. 16 Обновление BloodCollection

Запрос 5. Выборка из Doctors

ALTER PROCEDURE [dbo].[spDoctors\_GetAll]

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

select \* from dbo.Doctors

END

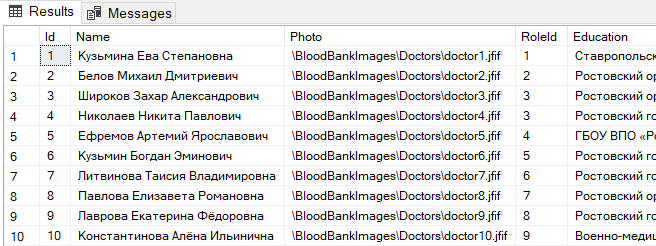


Рис. 17 Выборка из Doctors

Запрос 6. Выборка из Doctors по ID крови

ALTER PROCEDURE [dbo].[spDoctors\_GetByBloodId]

@BloodId int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

select dbo.Doctors.Id, dbo.Doctors.[Name], dbo.Doctors.Photo, dbo.Doctors.RoleId, dbo.Doctors.Education, dbo.Doctors.DateOfBirth , dbo.Doctors.Phone, dbo.Doctors.Email, dbo.Doctors.[Address]

from dbo.Doctors join dbo.BloodCollection on dbo.Doctors.Id = dbo.BloodCollection.DoctorInChargeId

where dbo.BloodCollection.Id = @BloodId;

END

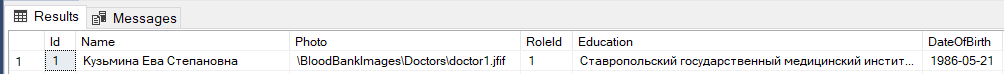


Рис. 18 Выборка из Doctors по ID крови

Запрос 7. Вставка в Doctors

ALTER PROCEDURE [dbo].[spDoctors\_Insert]

@Name nvarchar(max),

@Photo nvarchar(max),

@RoleId int,

@Education nvarchar(max),

@DateOfBirth nvarchar(max),

@Phone nvarchar(20),

@Email nvarchar(max),

@Address nvarchar(max),

@Id int = 0 output

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

insert into dbo.Doctors ([Name], Photo, RoleId, Education, DateOfBirth, Phone, Email, [Address])

values (@Name, @Photo, @RoleId, @Education, @DateOfBirth, @Phone, @Email, @Address);

select @Id = SCOPE\_IDENTITY();

END

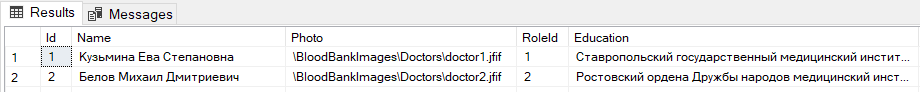


Рис. 19 Вставка в Doctors

Запрос 8. Обновление Doctors

ALTER PROCEDURE [dbo].[spDoctors\_UpdateById]

@Name nvarchar(max),

@Photo nvarchar(max),

@RoleId int,

@Education nvarchar(max),

@DateOfBirth nvarchar(max),

@Phone nvarchar(20),

@Email nvarchar(max),

@Address nvarchar(max),

@Id int

AS

BEGIN

SET NOCOUNT ON;

update dbo.Doctors

set [Name] = @Name,

Photo = @Photo,

RoleId = @RoleId,

Education = @Education,

DateOfBirth = @DateOfBirth,

Phone = @Phone,

Email = @Email,

[Address] = @Address

where Id = @Id;

END

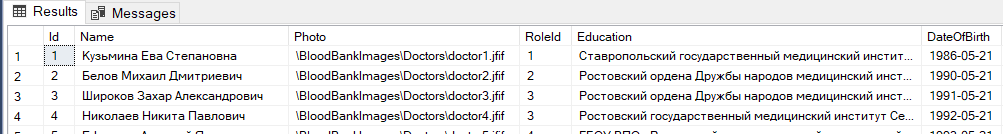


Рис. 20 Обновление Doctors

Запрос 9. Выборка из Donors

ALTER PROCEDURE [dbo].[spDonors\_GetAll]

AS

BEGIN

SET NOCOUNT ON;

select \* from dbo.Donors;

END

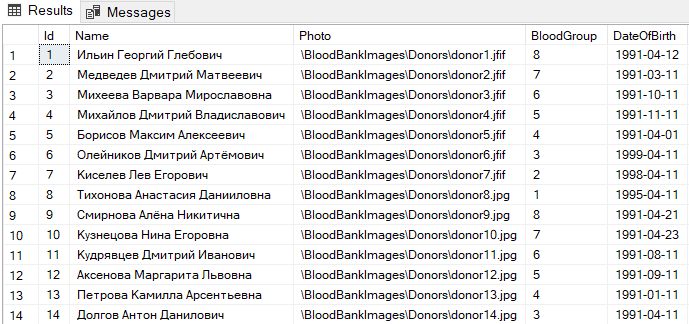


Рис. 21 Выборка из Donors

Запрос 10. Вставка в Issue

ALTER PROCEDURE [dbo].[spIssue\_Insert]

@RecipientId int,

@BloodId int,

@BloodAmount float,

@Unit nvarchar(max),

@PricePaid money,

@DoctorInChargeId int,

@DateOfIssue nvarchar(max),

@Id int = 0 output

AS

BEGIN

SET NOCOUNT ON;

insert into dbo.Issue (RecipientId, BloodId, BloodAmount, Unit, PricePaid, DoctorInChargeId, DateOfIssue)

values (@RecipientId, @BloodId, @BloodAmount, @Unit, @PricePaid, @DoctorInChargeId, @DateOfIssue);

select @Id = SCOPE\_IDENTITY();

END

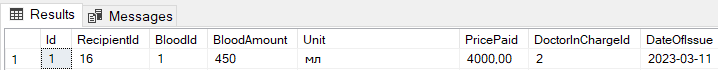


Рис. 22 Вставка в Issue

**2.5 Создание отчетов**

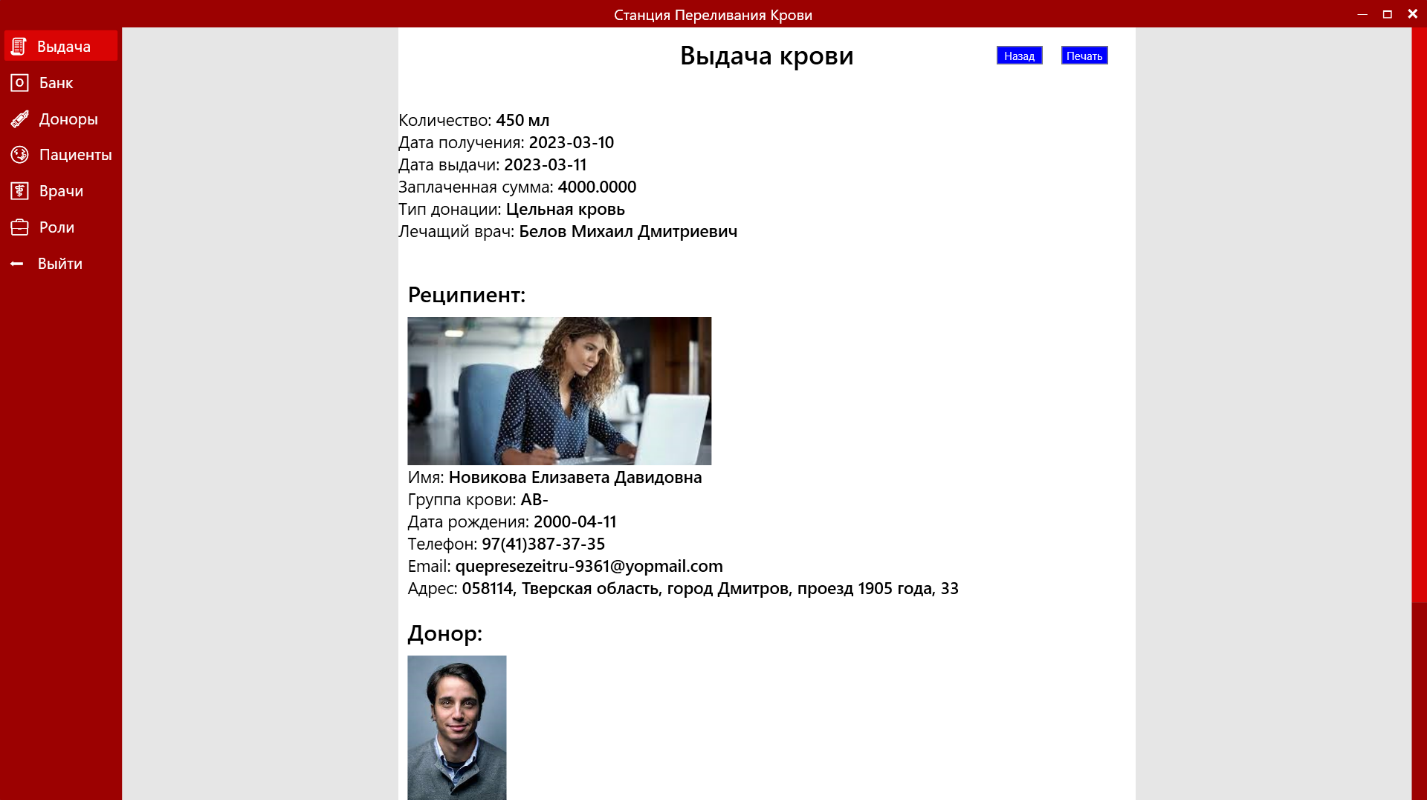


Рис. 23 Окно печати записи о выдаче крови

**2.6 Создание и применение форм**

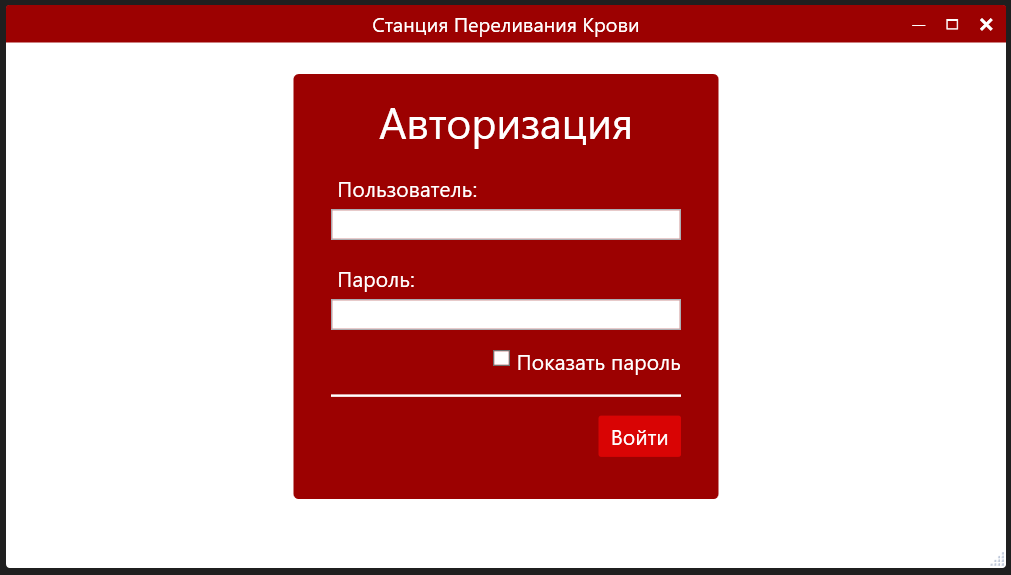


Рис. 24 Окно авторизации

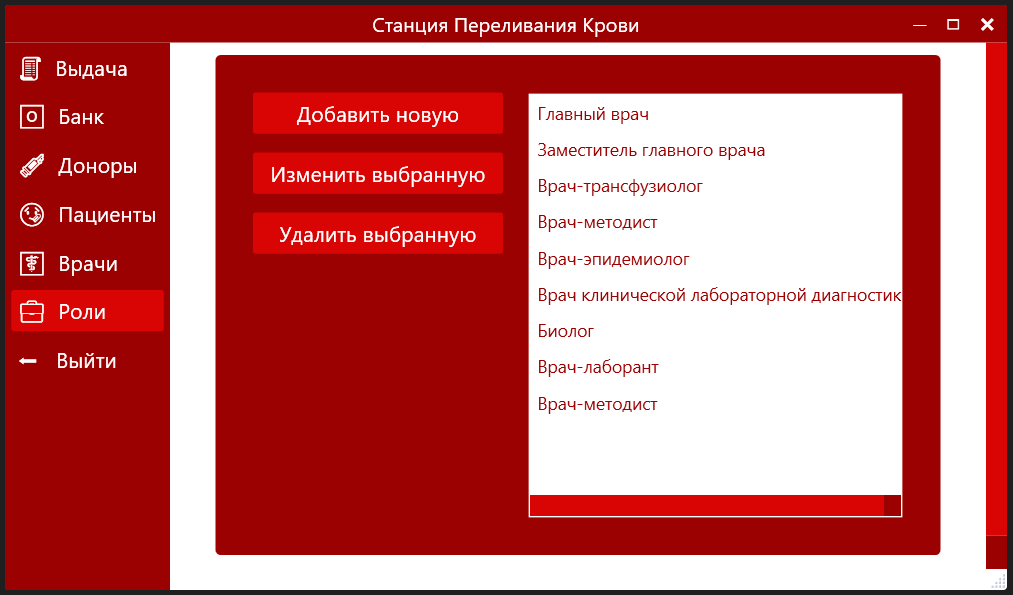


Рис. 25 Окно «Роли»

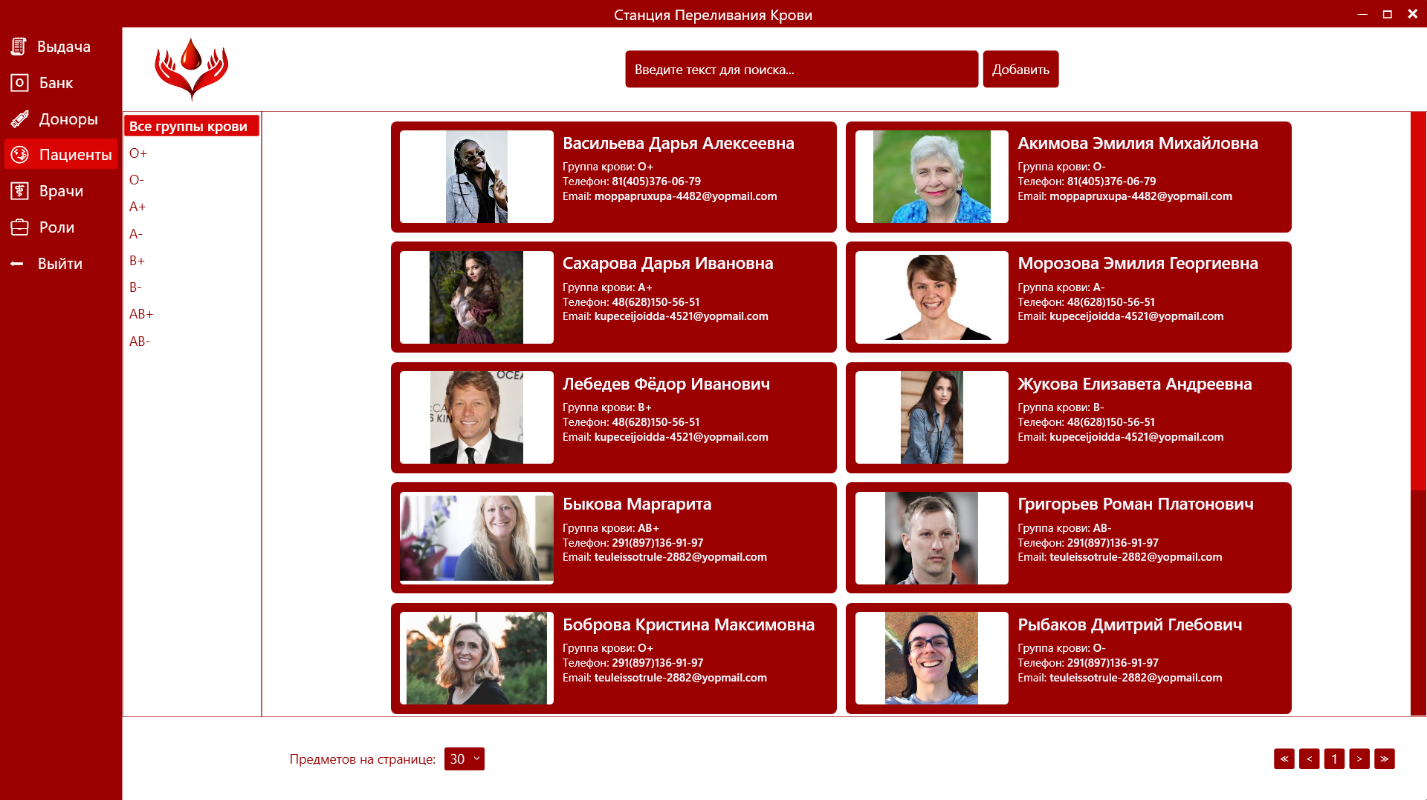


Рис. 26 Окно «Пациенты»

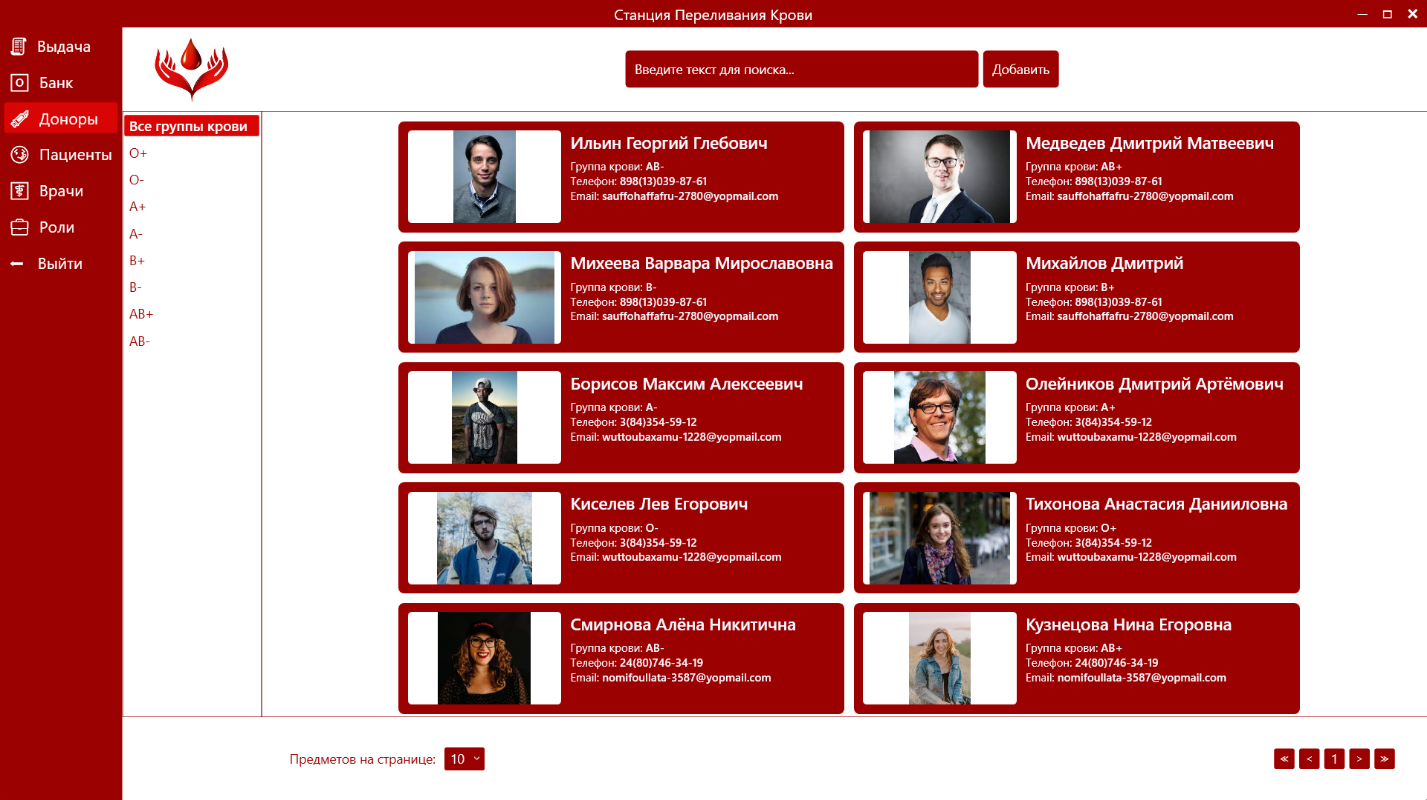


Рис. 27 Окно «Доноры»

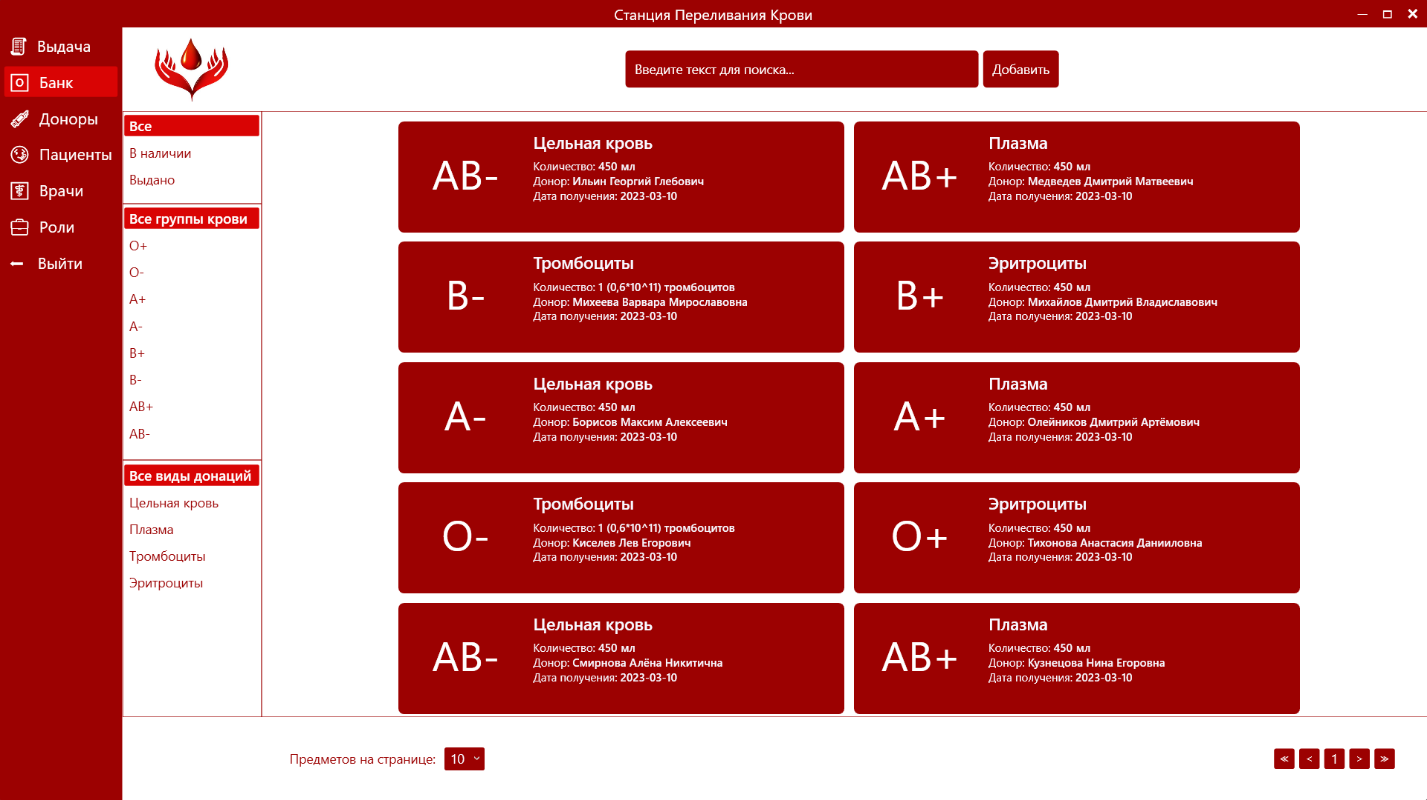


Рис. 28 Окно «Банк крови»

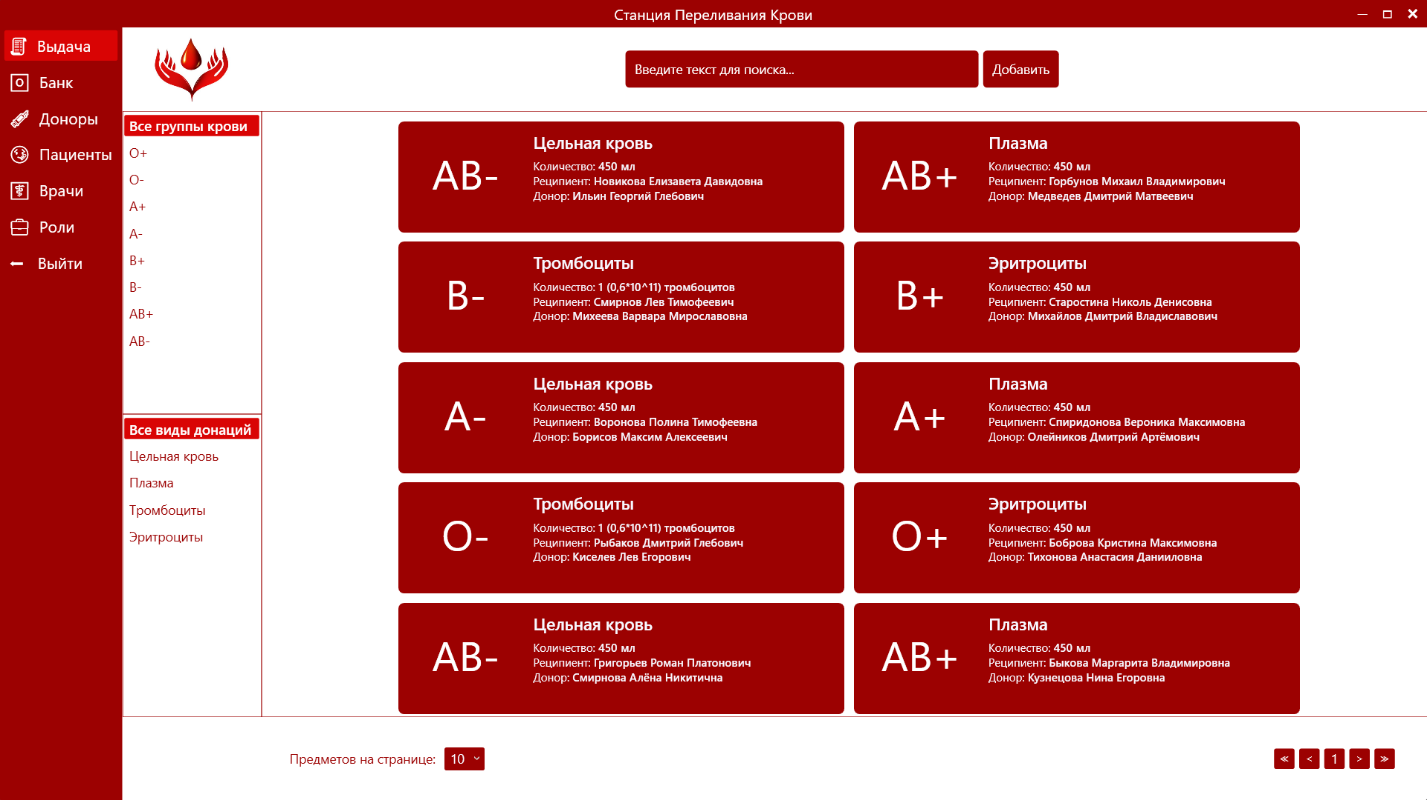


Рис. 29 Окно «Выдача крови»

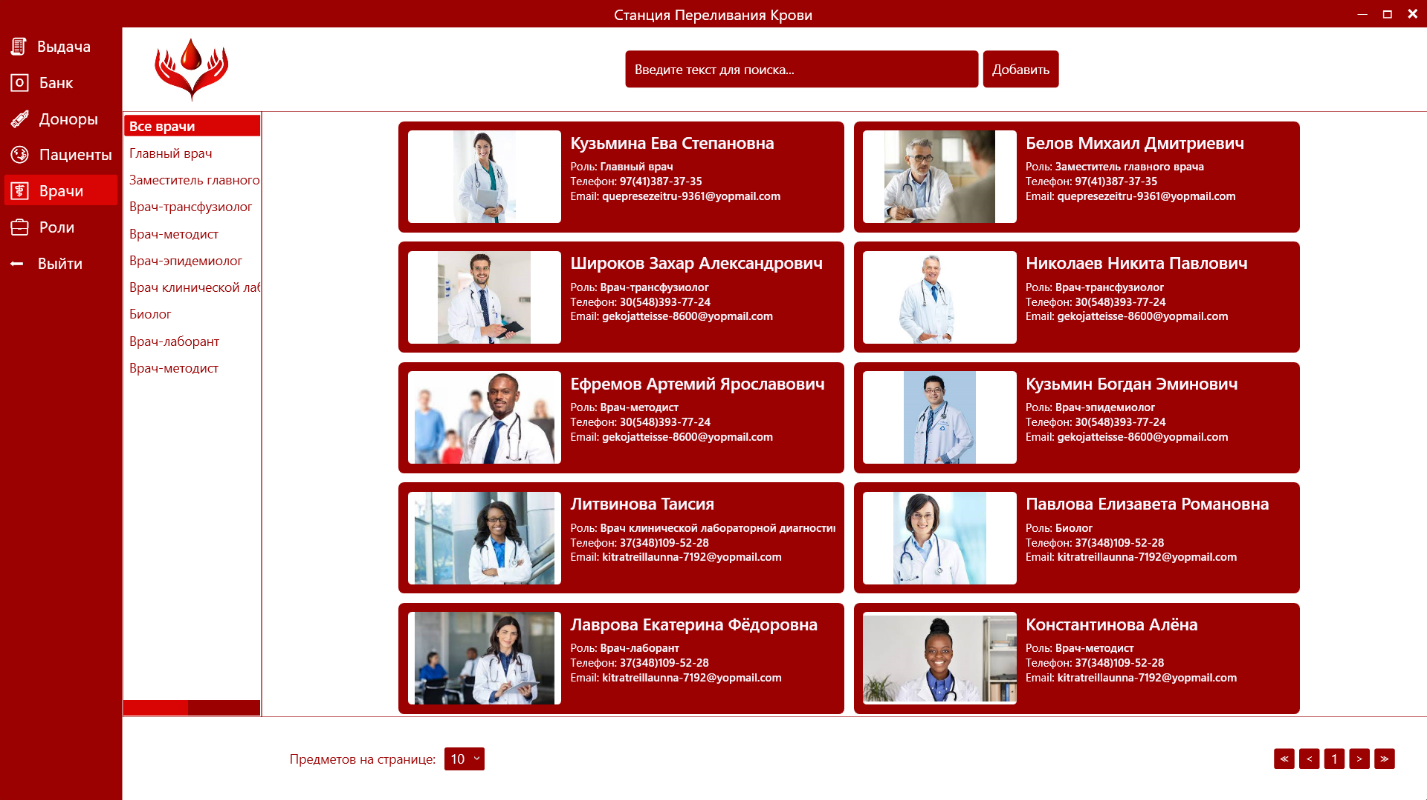


Рис. 30 Окно «Врачи»

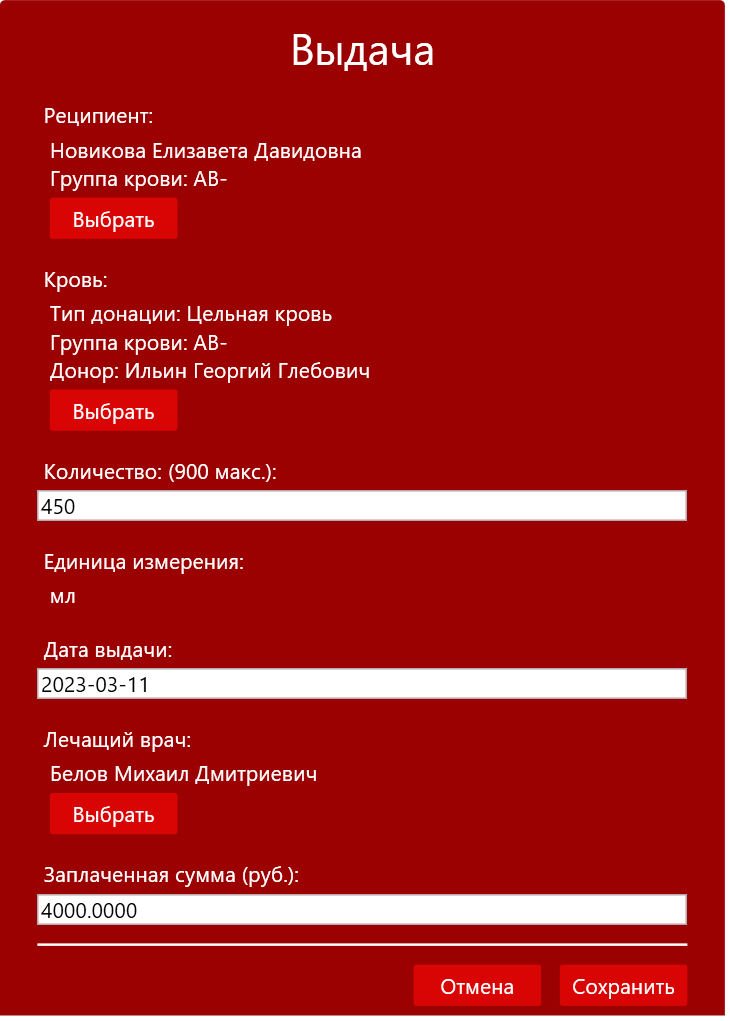


Рис. 31 Окно создания и редактирования записи о выдаче крови

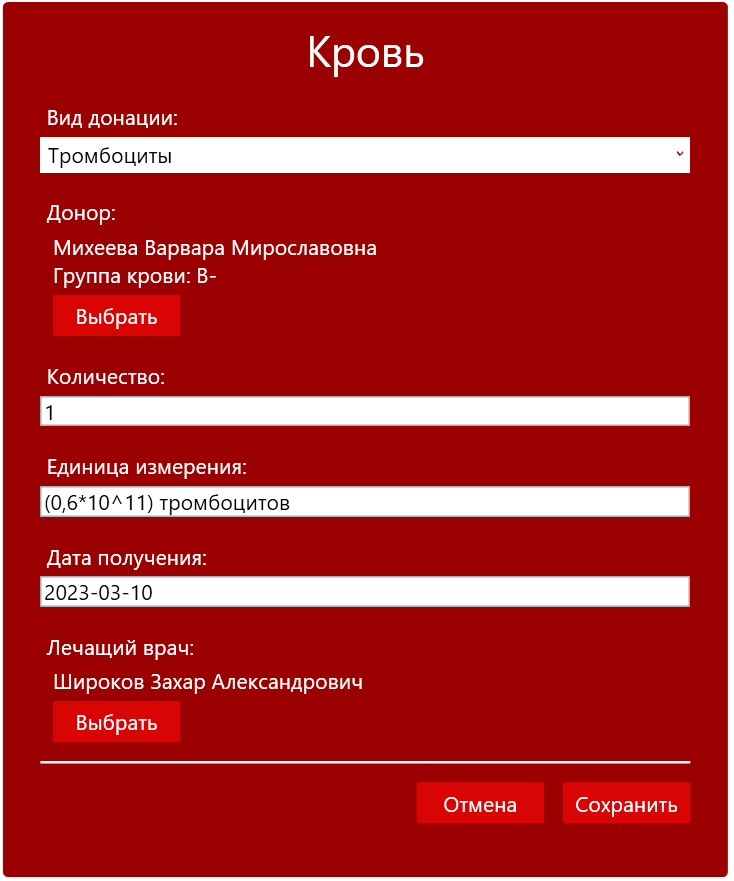


Рис. 32 Окно создания и редактирования записи о крови в банке крови



Рис. 33 Окно создания и редактирования записи о доноре



Рис. 34 Окно создания и редактирования записи о пациенте



Рис. 35 Окно создания и редактирования записи о враче

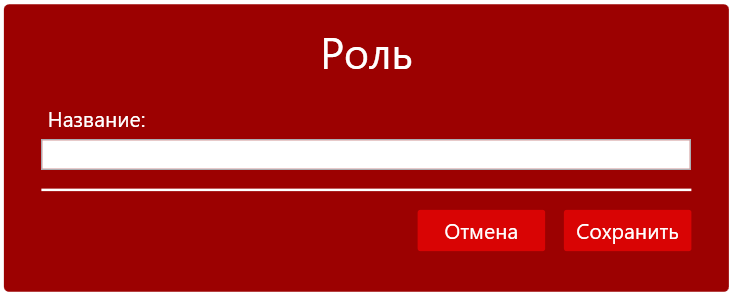


Рис. 36 Окно создания и редактирования записи о роли (должности)

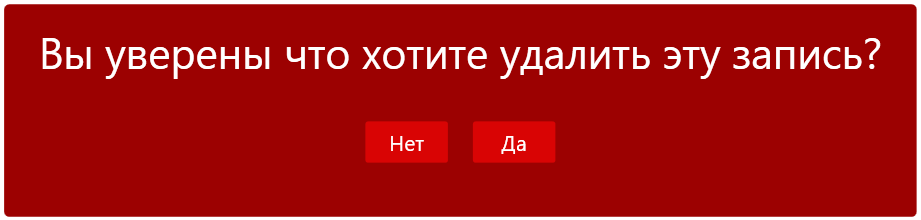


Рис. 37 Окно удаления записи

**2.7 Тестирование программы**

Таблица 11

Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | BloodBank |
| Номер версии | 1.00 |
| Имя тестера | Кузнецов Е.С. |
| Даты тестирования | 28.05.23 – 30.05.23 |

Таблица 12

Test case #1:

|  |  |
| --- | --- |
| Test Case # | 1 |
| Приоритет теста | Высокий |
| Название | Авторизация в качестве обычного пользователя |
| Резюме испытания | Необходимо авторизоваться в качестве обычного пользователя |
| Шаги | Ввод логина и пароля. Нажатие кнопки «Войти» |
| Данные | Логин: user, Пароль: 123 |
| Ожидаемый результат | Открытие формы обычного пользователя |
| Фактический результат | Открытие формы обычного пользователя |
| Предпосылки | Необходимо запустить приложение и открыть форму авторизации |
| Постусловия | Форма обычного пользователя открыта |
| Статус | Pass |
| Комментарии | Комментарии отсутствуют |

Таблица 13

Test case #2:

|  |  |
| --- | --- |
| Test Case # | 2 |
| Приоритет теста | Высокий |
| Название | Авторизация в качестве администратора |
| Резюме испытания | Необходимо достигнуть успешной авторизации в качестве администратора |
| Шаги | Ввод логина и пароля. Нажатие кнопки «Войти». |
| Данные | Логин: admin, Пароль: 12345 |
| Ожидаемый результат | Открытие формы администратора |
| Фактический результат | Открытие формы администратора |
| Предпосылки | Необходимо запустить приложение и открыть на форму авторизации |
| Постусловия | Форма администратора открыта |
| Статус | Pass |
| Комментарии | Комментарии отсутствуют |

Таблица 14

Test case #3:

|  |  |
| --- | --- |
| Test Case # | 3 |
| Приоритет теста | Средний |
| Название | Поиск записи в таблице Врачи с помощью строки поиска. |
| Резюме испытания | Необходимо выполнить поиск записей в таблице Врачи, воспользовавшись строкой поиска |
| Шаги | Ввод текста в строку для поиска. Нажатие кнопки «Enter» |
| Данные | Данные для поиска |
| Ожидаемый результат | Показ результатов поиска |
| Фактический результат | Показ результатов поиска |
| Предпосылки | Необходимо запустить приложение, авторизоваться и открыть окно Врачи |
| Постусловия | Открыто окно Врачи с результатом выполненного поиска |
| Статус | Pass |
| Комментарии | Комментарии отсутствуют |

Таблица 15

Test case #4:

|  |  |
| --- | --- |
| Test Case # | 4 |
| Приоритет теста | Средний |
| Название | Фильтрация записей в таблице Выдача |
| Резюме испытания | Необходимо выполнить успешную фильтрацию записей по группам крови в окне Выдача |
| Шаги | Выбор различных групп крови в списке слева |
| Данные | Нет данных |
| Ожидаемый результат | Успешнаяфильтрация записей в таблице |
| Фактический результат | Успешнаяфильтрация записей в таблице |
| Предпосылки | Необходимо запустить приложение, авторизоваться и открыть окно Выдача |
| Постусловия | Открыто окно Выдача с результатом выполненной сортировки |
| Статус | Pass |
| Комментарии | Комментарии отсутствуют |

Таблица 16

Test case #5:

|  |  |
| --- | --- |
| Test Case # | 5 |
| Приоритет теста | Средний |
| Название | Поиск записей в таблице Доноры |
| Резюме испытания | Необходимо выполнить поиск записей в таблице Доноры. |
| Шаги | Ввод текста в строку для поиска. Нажатие кнопки «Enter» |
| Данные | Данные для поиска |
| Ожидаемый результат | Успешный показ результатов поиска в таблице |
| Фактический результат | Успешный показ результатов поиска в таблице |
| Предпосылки | Необходимо запустить приложение, авторизоваться и открыть окно Доноры |
| Постусловия | Открыто окно Доноры с результатом поиска |
| Статус | Pass |
| Комментарии | Комментарии отсутствуют |

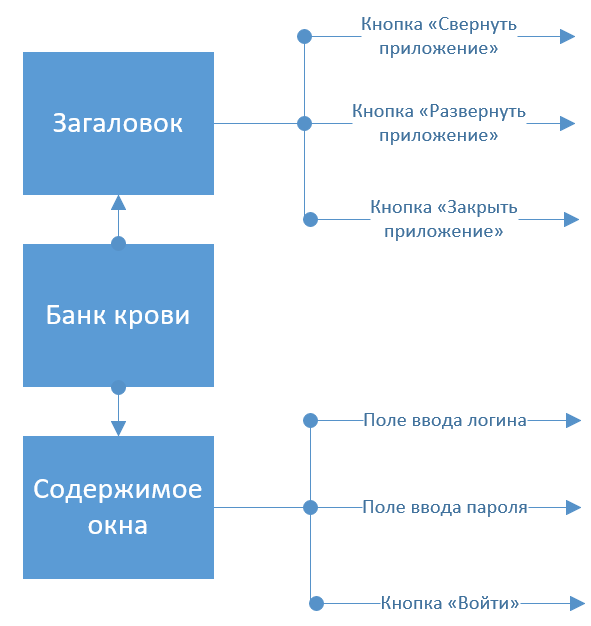


Рис. 38 Карта приложения

Таблица 17

Чек-лист

|  |
| --- |
| Проверка |
| Заголовок |
| Проверка работы кнопки «Свернуть приложение» |
| Проверка работы кнопки «Развернуть приложение» |
| Проверка работы кнопки «Закрыть приложение» |
| Содержимое окна |
| Проверка ввода текста в поле ввода логина (числа, буквы, спецсимволы) |
| Проверка ввода текста в поле ввода пароля (числа, буквы, спецсимволы) |
| Проверка работы кнопки "Войти" |

**3. ИНФОМАЦИОННАЯ БЕЗОПАСНОСТЬ**

**3.1 Методы защиты информационной базы**

Методом защиты информационной базы является авторизация пользователя.

Авториза́ция — предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки данных прав при попытке выполнения этих действий. Авторизация — процесс предоставления пользователю или группе пользователей определенных разрешений, прав доступа и привилегий в компьютерной системе.

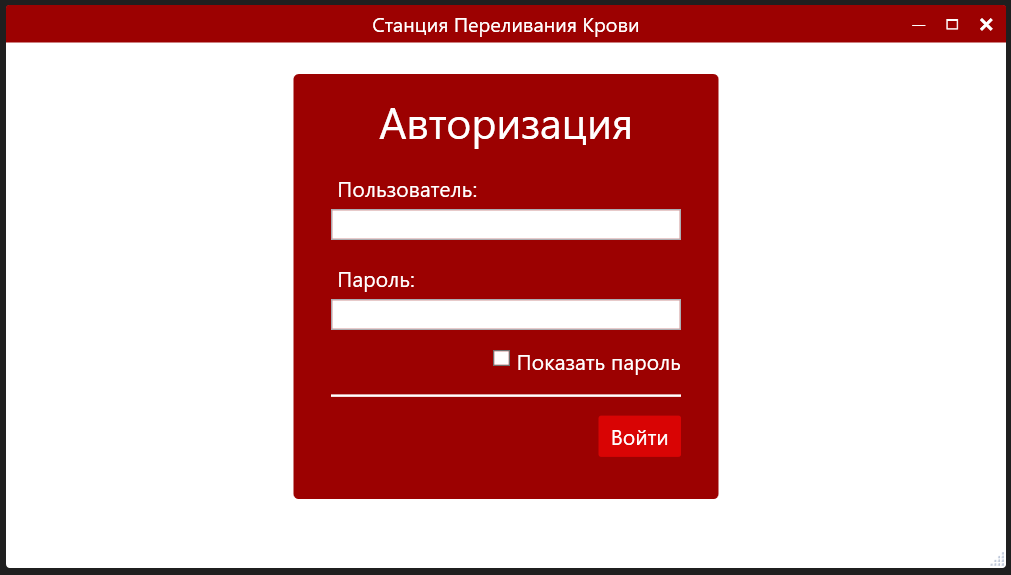


Рис. 39 Окно авторизации

Код окна авторизации

internal class AuthorizationViewModel : ViewModelBase

{

private string \_userName;

public string UserName

{

get { return \_userName; }

set

{

\_userName = value;

OnPropertyChanged();

}

}

private string \_password;

public string Password

{

get { return \_password; }

set { \_password = value;

OnPropertyChanged();

}

}

public DelegateCommand OpenMainMenuCommand { get; set; }

public AuthorizationViewModel()

{

OpenMainMenuCommand = new DelegateCommand(x =>

{

if (UserName != null && Password != null)

{

SqlConnection sqlcon = new SqlConnection(Connector.GetConnectionString("BloodBank"));

string query = "select \* from [Authorization] where UserName = '"

+ UserName.Trim() + "' and Password = '" + Password.Trim() + "'";

SqlDataAdapter sqlda = new SqlDataAdapter(query, sqlcon);

DataTable dTable1 = new DataTable();

sqlda.Fill(dTable1);

if (dTable1.Rows.Count > 0)

{

if (UserName == "admin")

{

MainWindowViewModel.Instance.CurrentUserIsAdmin = true;

MainWindowViewModel.Instance.CurrentViewModelOne = MainWindowViewModel.Instance.MainMenuViewModel;

}

else

{

MainWindowViewModel.Instance.CurrentUserIsAdmin = false;

MainWindowViewModel.Instance.CurrentViewModelOne = MainWindowViewModel.Instance.MainMenuViewModel;

}

}

else

{

MessageBox.Show("Неверно введено имя пользователя или пароль","Неверный пароль",MessageBoxButton.OK,MessageBoxImage.Warning);

}

}

else

{

MessageBox.Show("Неверно введено имя пользователя или пароль", "Неверный пароль", MessageBoxButton.OK, MessageBoxImage.Warning);

}

});

}

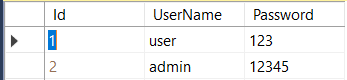


Рис. 40 Логины и пароли

**4. Экономическое обоснование**

Разрабатываемая информационная система по учету получения крови от доноров и выдачи крови реципиентам для Станции переливания крови, основные функции: хранение данных о донорах, реципиентах, полученной и выданной крови, выплатах заработной платы, вывод на печать основных документов. Система является автоматизированной, что облегчает работу в ней. Назначение информационной системы реализация продажи услуг, расширенная информационная база и простой интерфейс для управления системой. Учтены все области работы станции переливания крови. Реализована возможность добавлять и редактировать уже существующую информацию. Цена составляет мини система: 340р в месяц, сеть система: 2765р в месяц, популярная бизнес система: 1480р в месяц.

**4.1 Планирование разработки программного продукта «Станция переливания крови».**

Разработка программного продукта подразделяется на следующие этапы: анализ требований, проектирование, кодирование (программирование), тестирование и отладка, эксплуатация и составление программной документации. Каждый этап имеет свою трудоемкость, показатели трудоемкости данных этапов приведены в таблице.

Трудоемкость человека в днях определяется по формуле: Ч=Х\*Т, где

Ч – человеко-дни;

Х – количество работников на каждом этапе;

Т – фактическое количество дней, потраченных на выполнение каждого этапа в отдельности.

Таблица 18

Характеристика работ по составлению программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название работы | Чел. дни | % к итогу | Кол-во человек | Исполнители | Кол-во дней |
| Анализ требований | 15 | 15 | 1 | Программист | 15 |
| Проектирование | 30 | 30 | 1 | Программист | 30 |
| Кодирование | 32 | 32 | 1 | Программист | 32 |
| Тестирование | 23 | 23 | 1 | Программист | 23 |
| ИТОГО | 100 | 100 |  |  | 100 |

Далее определяется общее количество дней, потраченных на проектирование и разработку всей системы. Это рассчитывается по формуле:

𝑇ц = Q/R , где

Тц – длительность цикла, дней;

Q – трудоемкость человеко-дней;

R – количество исполнителей.

Таким образом можно сказать, что программист работал:

𝑇ц = 100/1 = 100 дней

Проект будет начат 02.05.2023 + 45 дней и закончен 16.06.2023

**4.2 Стоимость разработки**

Разработка приложения на базе WPF является не самой дорогой, но чаще всего может лучше удовлетворить требования заказчика в связи с тем, что, имея полный доступ к изменению системы управления, может более гибко подойти к выполнению заказа.

Любая программа не может работать без операционной системы. В настоящее время наиболее распространенной операционной системой является Windows 10 Pro. Данная операционная система наиболее оптимизирована для работы с прикладными программами, имеет удобный интерфейс. Стоимость Windows 10 Pro составляет 23900 руб..

Таблица 19

Системные требования Visual Studio 2022

|  |  |  |
| --- | --- | --- |
| Оборудование | Оптимально | Минимально |
| Процессор | 64-разрядный процессор с тактовой частотой не ниже 2,6 ГГц. | 64-разрядный процессор с тактовой частотой не ниже 1,8 ГГц |
| HD | доступное дисковое пространство 5,3 Гб. | доступное дисковое пространство 2,7 Гб. |
| CD-ROM | Да | Да |
| Дисплей | 1024x768, 16-бит видеорежим (32-бита рекомендовано), 256 цветов | Super VGA (800Ѕ600) и выше, 256 цветов |
| ОС | [Windows 11 Домашняя в режиме S](https://support.microsoft.com/en-us/windows/windows-10-and-windows-11-in-s-mode-faq-851057d6-1ee9-b9e5-c30b-93baebeebc85#WindowsVersion=Windows_11), [Windows IoT Корпоративная](https://learn.microsoft.com/ru-ru/windows/iot/product-family/windows-iot) | [Windows 10 IoT Базовая](https://learn.microsoft.com/ru-ru/windows/iot-core/windows-iot), [Windows 10 Корпоративная, выпуск LTSC](https://learn.microsoft.com/ru-ru/windows/deployment/update/waas-overview), [Windows 10 S](https://www.microsoft.com/windows/windows-10-s) и [Windows 10 Team Edition](https://support.microsoft.com/surface-hub) |
| Периферия | Мышь Microsoft | Мышь Microsoft |
| Периферия | Клавиатура Microsoft | Клавиатура Microsoft |
| RAM | 8 Гб | 4 Гб |

Таблица 20

Необходимые требования и расходы

|  |  |  |
| --- | --- | --- |
| Товар | Описание | Цена |
| Процессор | AMD A6 2C+3G 2.6ГГц | 23000 |
| Материнская плата | ASUS P5B SE (LGA775, P965, PCI-E, 4xDDR-II(800),1xIDE,5xSATA-II, 3xPCI, ATX) | 12000 |
| RAM | 8,00 Гб | 7000 |
| Видеоадаптер | AMD Radeon (TM) R$ Graphics | 8000 |
| Жесткий диск | SAMSUNG MZ7DKB233LJHFKK-000L2 | 12000 |
| CD-ROM | CD-ROM LG 52х (Black) | 2000 |
| Корпус | ATX Midle Tower ASUS TA-891, 350W | 3000 |
| Клавиатура | PS/2 A-4 Tech KB-28G Game Master | 700 |
| Мышь | A-7 Tech X9710 OPTICAL PS/2 1000 dpi | 500 |
| Монитор | 17" SAMSUNG 794 MB | 10000 |
| Итого: | | 83600 руб. |

**4.3 Расчет экономической эффективности программного продукта**

Программа позволяет сократить время и трудоёмкость процесса принятия решений путем построения иерархии критериев с последующей оценкой возможных альтернатив по каждому из выбранных критериев. Найденные в результате работы программы решения будут математически обоснованными, что может гарантировать получение наиболее оптимального результата. Экономическая эффективность программы достигается за счет сокращения времени расчётов на ЭВМ с 900 до 350 часов в год и автоматизации дополнительной работы по обслуживанию аналитика, что в сумме приводит к экономии заработной платы 208845,19 рублей в месяц, 12 месяцев в году.

Месячная экономия по заработной плате составляет:

Эзп.мес. = ∆ 𝑇 ∗ 𝐶 1ч/и, где

∆ 𝑇 – экономия времени, час;

∆ 𝑇 = Тв – Тн, где

Тв – годовая трудоемкость решения задачи (базовый вариант);

Тн – годовая трудоемкость решения задачи новый вариант;

Таблица 21

Нормы времени выполнения работ

|  |  |  |  |
| --- | --- | --- | --- |
| Показатель | Единицы измерения | Базовый вариант (вручную) | Новый вариант (Audit Expert) |
| Годовая трудоемкость операции по вводу и корректировке данных | Человеко-час | 125 | 20 |
| Годовая трудоемкость операции по оформлению и выводу данных | Человеко-час | 51 | 5 |
| ИТОГО |  | 264 | 44 |

Тв = 22 ∗ 8 = 176 ч

Тн = 25 ч

∆ 𝑇 = 176 – 25 = 151 ч

𝐶 1ч/и – стоимость 1 часа работы исполнителя (администратора);

𝐶 1ч/и = (ЗП\*L)/(22\*8), где

ЗП – заработная плата исполнителя;

L – коэффициент социальной выплаты 1,22%;

22 – количество рабочих дней в месяце;

8 – количество рабочих часов в день;

𝐶 1ч/и = (30000\*1,22)/(22\*8) = 207,95 руб.

Таким образом можно рассчитать месячную экономию по заработной плате:

Эзп.мес. = 151 ∗ 207,95 = 31400,45 руб.

Годовая экономия по заработной плате с учетом начислений составляет:

Эзп = Э зп.мес. ∗ (1 + Котч) ∗ 𝑛, где

Эзп - годовая экономия по заработной плате с отчислениями на социальные мероприятия, руб.;

Эзп.мес.- месячная экономия заработной платы при внедрении программы, руб.;

Котч - коэффициент, учитывающий отчисления на социальное страхование и в другие фонды, 0,299;

𝑛 - количество месяцев.

Эзп = 31400,45 ∗ (1 + 0,299) ∗ 12 = 489470,21 руб.

Годовая экономия определяется по формуле:

,где Тм1 , Тм2 - машинное время, необходимое для решения поставленных задач, соответственно в старом и в новом вариантах, машинно-ч/год;

Рэкс.ктс - эксплуатационные расходы, приходящиеся на 1 ч машинного времени ЭВМ руб./машино-час;

- нормативный коэффициент эффективности капитальных вложений в средства автоматизации (В целом по макроэкономике данный коэффициент брался в размере 0,14);

Фб.ктс- балансовая стоимость ЭВМ (КТС), руб.;

Фэф.ктс- годовой эффективный фонд времени работы ЭВМ, эф. ктс. машинно-ч;

Годовой эффективный фонд времени работы ЭВМ составляет 1800 часов.

𝑆рп- суммарные затраты на разработку программы, руб.;

Тс- срок службы внедряемой программы до морального износа, лет.

Срок службы внедряемой программы до морального износа составит 4 года.

Тм1 = 176 ∗ 12 = 2112 часов/год

Тм2 = 25 ∗ 12 = 300 часов/год

Рэкс.ктс = (23900 + 83600)/176 = 610,79 руб.

Фб.ктс = 23900 + 83600 = 107500 руб.

Э = (2112 – 300) \* 610,79 + 0,14 \* (107500\*(2112-300))/1800 + 489470,21 = 1196753,21 руб.

Коэффициент эффективности капитальных вложений Е и срок их окупаемости Т рассчитываются по формулам:

Е = Э/Sрп

Ток = Sрп/Э

Таким образом:

Е = 1196753,21/107500 = 11,13

Ток = 107500/1196753,21 = 0,09

Таблица 22

Технико-экономические показатели

|  |  |  |
| --- | --- | --- |
| Показатели | Единица измерения | Значение показателей |
| Затраты на разработку программы | руб. | 107500 |
| Годовая экономия | руб. | 489470,21 |
| Срок окупаемости программы | лет | 0,09 |

Затраты на разработку программы равны 107500 руб. Срок окупаемости программы составляет 0,09 года. Коэффициент эффективности капитальных вложений равняется 11,46. Годовая экономия при использовании программы составляет 489470,21 руб. Таким образом, внедрение программного продукта является экономически целесообразным.

**ЗАКЛЮЧЕНИЕ**

На современном этапе развития производительных сил и производственных отношений информация стала товаром со всеми присущими ей свойствами. Сегодня существуют информационная промышленность, национальные информационные ресурсы, происходит переход от индустриальной экономики к экономике, основанной на информации.

Служба крови – это важнейшая отрасль здравоохранения, целью которой является обеспечение качественной трансфузионной терапии.

Здравоохранение располагает не только самой мощной в области лечебной базой, но и наиболее высоким уровнем оказания специализированной медицинской помощи.

Мобильный пункт заготовки крови – это специально оборудованный трейлер, в котором есть все необходимое для проведения выездных донорских акций. Полуприцеп имеет шесть отдельных помещений, изолированных друг от друга стационарными стенами и дверьми. Каждое помещение предназначено для выполнения определенных функций: комната отдыха и ожидания доноров, кабинет регистратуры, кабинет медицинского осмотра доноров, процедурный кабинет для забора крови с 5 донорскими креслами и медицинская лаборатория с оборудованием для хранения крови.

Масштабы деятельности отрасли, рассосредоточенность её объектов, динамический характер производственного процесса, воздействие большого числа случайных факторов обусловливают необходимость сбора, передачи, хранения информации и доведения её до пользователя, для чего и предназначена данная информационная система.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. 1. Агуров, Павел C#. Сборник рецептов / Павел Агуров. - М.: "БХВ-Петербург", 2021. - 432 c.
2. Албахари, Джозеф C# Справочник / Джозеф Албахари , Бен Албахари. - М.: БХВ-Петербург, 2022. - 944 c.
3. Албахари, Джозеф C# Справочник / Джозеф Албахари , Бен Албахари. - М.: БХВ-Петербург, 2021. - 940 c.
4. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2021. - 266 c.
5. Бишоп, Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2022. - 472 c.
6. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2021. - 320 c.
7. Зиборов, В.В. Visual C# 2022 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2022. - 480 c.
8. Зиборов, Виктор Visual C# 2021 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2021. - 432 c.
9. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2021. - 496 c.
10. Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами / А.И. Касаткин. - М.: Высшая школа, 2021. - 432 c.
11. Лотка, Рокфорд C# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. - М.: Вильямс, 2022. - 816 c.
12. Мак-Дональд, Мэтью Silverlight 5 с примерами на C# для профессионалов / Мэтью Мак-Дональд. - М.: Вильямс, 2013. - 848 c.
13. Марченко, А. Л. Основы программирования на С# / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2021.-552c.