

## Содержание

Введение	3
1 Назначение и цели разработки	5
2 Разработка технического проекта на основе анализа требований	6
2.1 Определение спецификаций программного обеспечения	6
2.2 Проектирование модели данных и диаграммы классов	7
2.3 Конструирование прототипа	13
3 Реализация	17
3.1 Обоснование выбора средств разработки	17
3.2 Разработка базы данных в среде СУБД	17
3.3 Описание программных модулей	17
4 Тестирование	28
4.1 Модульное тестирование	28
4.2 Интеграционное тестирование	29
5 Эксплуатационная документация	30
5.1 Руководство пользователя	30
5.1.1 Назначение программных модулей системы	30
5.1.2 Условия применения программных модулей системы	30
5.1.3 Сведения по подготовке программного обеспечения системы	30
к работе	30
5.1.4 Описание операций	31
5.1.5 Действия пользователя при аварийных ситуациях	32
Заключение	33
Список использованных источников	35
Приложение А Техническое задание. Требования к программным модулям	37

					<b>ККЭП 09.02.07 0139 ПЗ</b>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Молокотин			Разработка модулей для автоматизации процесса анализа движения пациентов в лечебных отделениях больницы			
Провер.		Головки Р.А.						
						Лит.	Лист	Листов
						КП	2	70
						Гр. 30-КД9-ЗИСП		

Приложение Б Программный код	45
Приложение В Скрипт базы данных	59
Приложение Г Тест-кейсы	69

## Введение

В текущее время подавляющее большинство сфер человеческой деятельности подвергается автоматизации при помощи компьютерных систем. Это способствует повышению эффективности работы предприятия, продуктивности сотрудников и снимает многие ограничения на развитие, связанные, прежде всего, с хранением и обработкой огромного объёма данных. Медицинская сфера не исключение, и от того, насколько будет автоматизирована система, могут зависеть жизнь и здоровье людей.

Целью курсового проекта является создание усовершенствованной подсистемы движения пациентов в лечебных отделениях больницы на основе существующей подсистемы краевой клинической больницы №1 города Краснодара. Для достижения поставленной цели определены следующие задачи:

- определить назначение и цели разработки;
- провести анализ предметной области;
- сформулировать требования к разрабатываемой системе;
- разработать технический проект на основе требований;
- разработать реализацию системы в соответствии с техническим проектом;
- провести тестирование реализации с целью выявления дефектов;
- составить эксплуатационную документацию для пользователя.

## 1 Назначение и цели разработки

Заданием предусмотрена разработка программных модулей подсистемы движений пациентов в лечебных отделениях больницы для внедрения в существующую компьютерную стационарную больницу.

Целью разработки является повышение эффективности работы больницы в результате внедрения разработанной подсистемы. Далее представлен список показателей для отражения эффективности внедрения:

- скорость загрузки данных медицинских карт;
- скорость загрузки данных о движениях пациента;
- скорость формирования отчётности.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

## 2 Разработка технического проекта на основе анализа требований

### 2.1 Определение спецификаций программного обеспечения

Рассмотрим определение вариантов использования (прецедентов).

В ходе анализа предметной области были выделены следующие актёры:

- врач;
- врач-статист;
- экономист;
- средний медперсонал (медсестра, медбрат);
- младший медперсонал (медрегистратор).

В таблице 2.1 показаны прецеденты (варианты использования) для разрабатываемой системы.

Таблица 2.1 – Прецеденты для разрабатываемой системы

Роль	Прецедент
Врач	- просмотреть движения - просмотреть карту - войти в систему
Врач-статист	- просмотреть движения - просмотреть карту - редактировать стационарную карту - войти в систему
Младший медперсонал	- создать стационарную карту - редактировать стационарную карту - создать движение - удалить движение - просмотреть движения - просмотреть карту - войти в систему
Средний медперсонал	- просмотреть движения - просмотреть карту - войти в систему
Экономист	- просмотреть карту - просмотреть отчёт - войти в систему

На рисунке 2.1 показана Use-Case диаграмма для разрабатываемой системы.

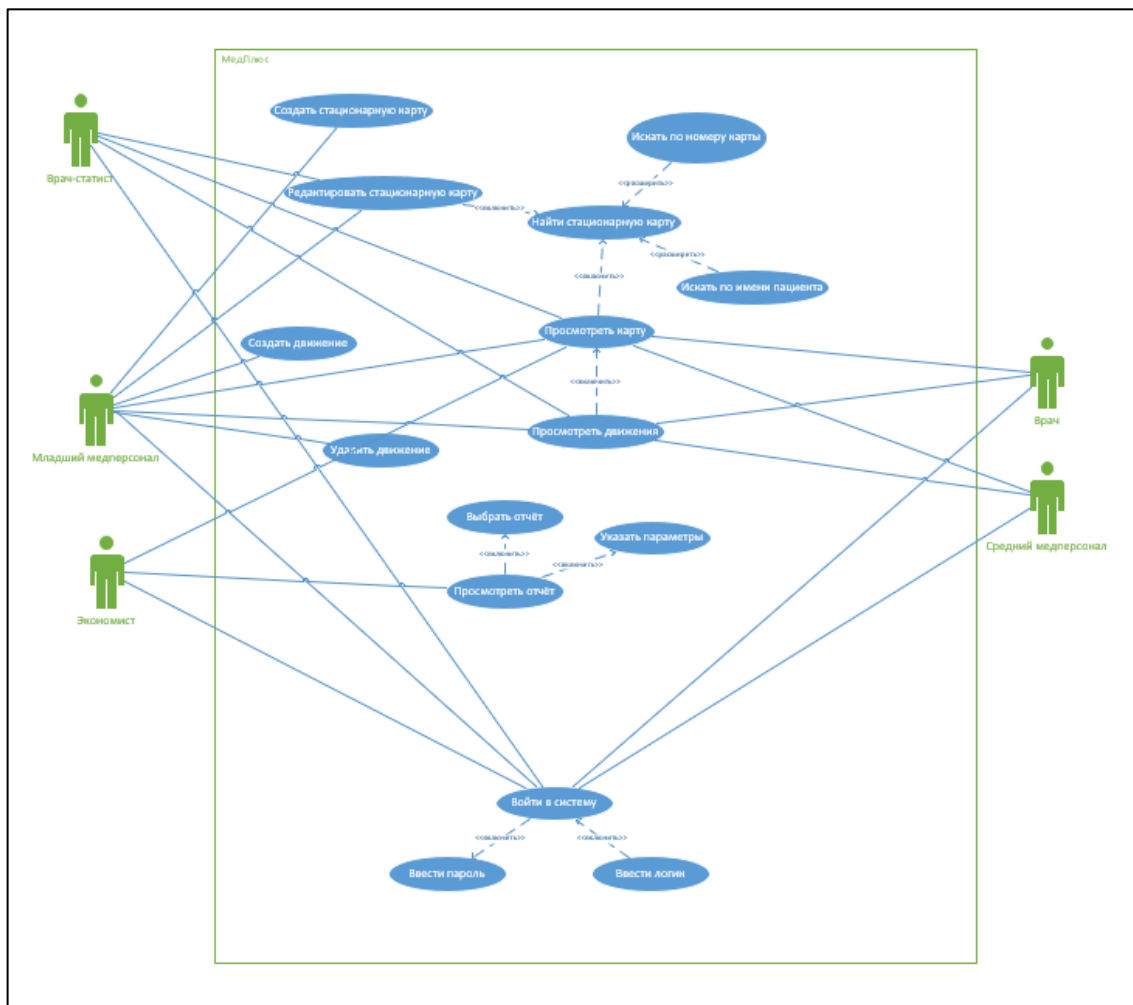


Рисунок 2.1 – Диаграмма вариантов использования программной системы  
«МедПлюс»

## 2.2 Проектирование модели данных и диаграммы классов

На основе анализа требований заказчика к функциям программной системы, требованиям к организации входных и выходных данных, с учетом спроектированных требований к реализации функций, описанных в диаграммах вариантов использования, деятельности и последовательности, была разработана модель данных системы и описана в виде ER-диаграммы, позволяющая четко описать требования к представлению логической структуры данных, на основе

которой в последующем будет разработана физическая структура данных для хранения во внешней памяти и программной обработки.

Для построения ER-диаграммы применялись средства среды MS Visio.

На рисунках 2.2 – 2.6 представлена ER – диаграмма для программной системы «МедПлюс».

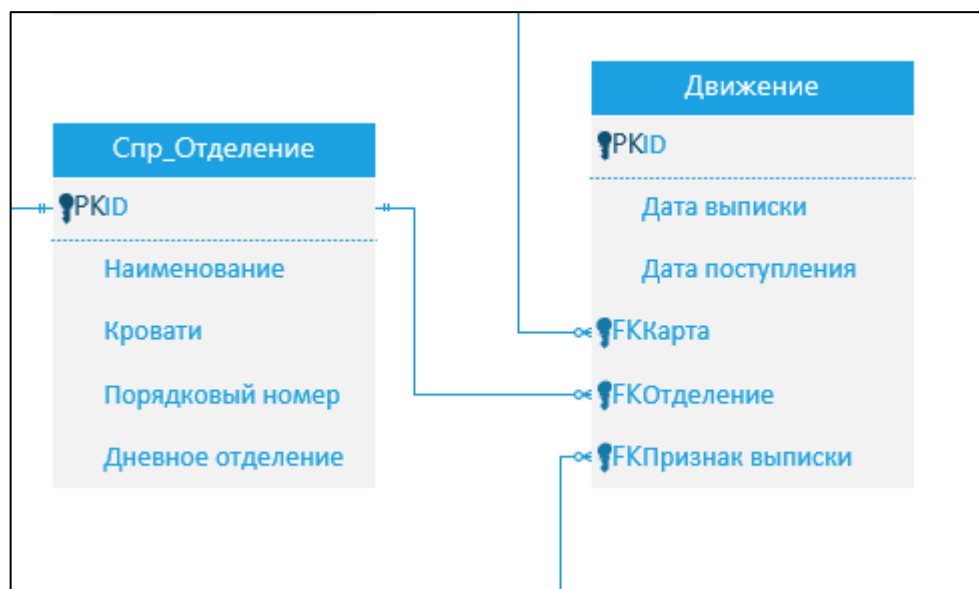


Рисунок 2.2 – ER-диаграмма для программной системы «Медплюс» (1)

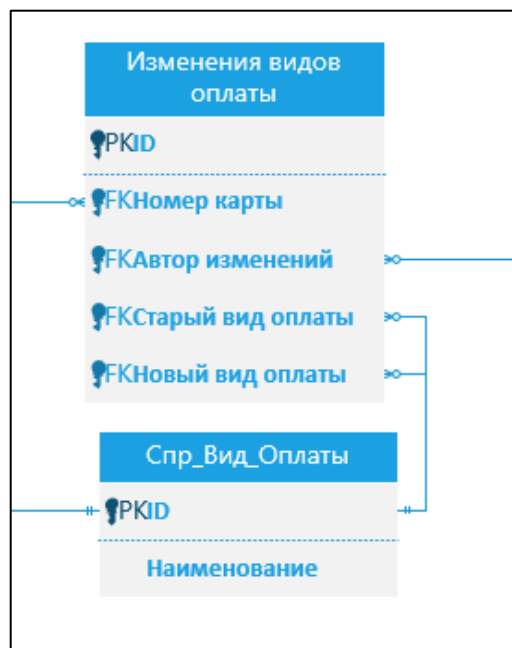


Рисунок 2.3 – ER-диаграмма для программной системы «Медплюс» (2)

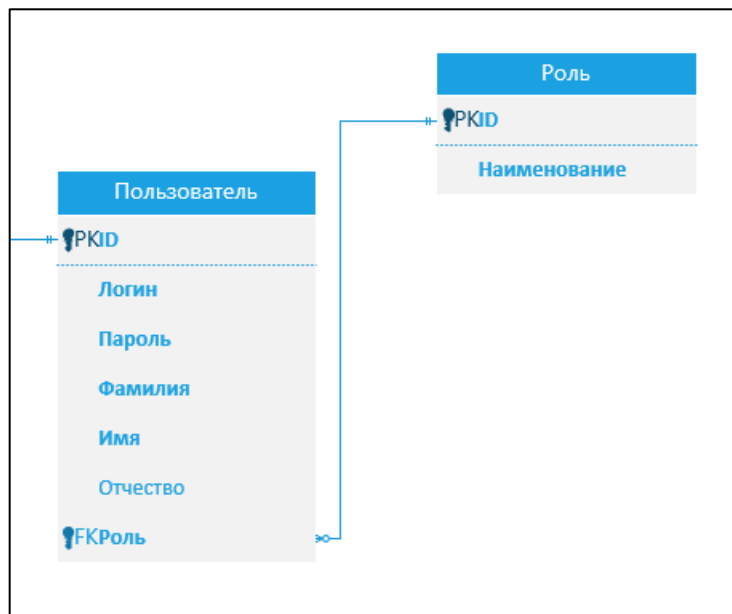


Рисунок 2.4 – ER-диаграмма для программной системы «МедПлюс» (3)



Рисунок 2.5 – ER-диаграмма для программной системы «МедПлюс» (3)



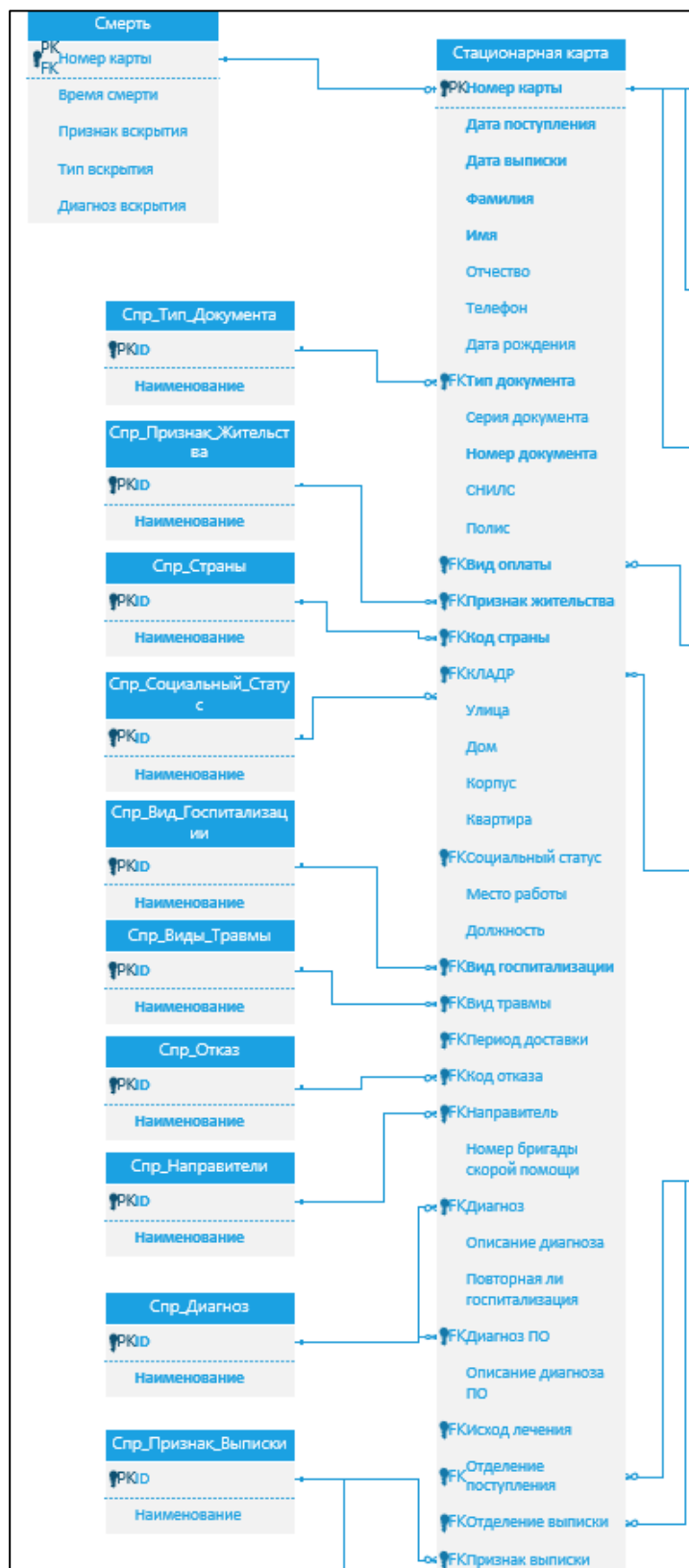


Рисунок 2.6 – ER-диаграмма для программной системы «МедПлюс» (4)

В таблице 2.2. представлены сведения об основных сущностях и связях между сущностями ER-диаграммы.

Таблица 2.2 – Характеристики сущностей и связей между сущностями ER-диаграммы для программной системы «МедПлюс»

Сущность	Назначение сущности	Ключ	Характеристики связей
Стационарная карта	<p>Содержит данные о лечении пациента</p> <ul style="list-style-type: none"> <li>- номер карты;</li> <li>- дата поступления;</li> <li>- дата выписки;</li> <li>- отделение поступления;</li> <li>- отделение выписки;</li> <li>- тип документа, подтверждающего личность;</li> <li>- номер документа;</li> <li>- серия документа;</li> <li>- ФИО пациента;</li> <li>- пол;</li> <li>- дата рождения;</li> <li>- признак жительства;</li> <li>- страна;</li> <li>- адрес;</li> <li>- телефон;</li> <li>- социальный статус;</li> <li>- место работы;</li> <li>- должность;</li> <li>- тип оплаты;</li> <li>- серия полиса;</li> <li>- номер полиса;</li> <li>- СНИЛС;</li> <li>- направитель;</li> <li>- дата направления;</li> <li>- тип госпитализации;</li> <li>- тип доставки;</li> <li>- номер бригады скорой помощи</li> <li>- диагноз направителя;</li> <li>- диагноз приёмного отделения;</li> <li>- диагноз выписки;</li> <li>- признак повторной госпитализации;</li> <li>- вид травмы;</li> <li>- период доставки;</li> <li>- дата и время выписки;</li> </ul>	Номер карты (РК)	<p>Тип документа - Стационарная карта (n...1)</p> <p>Отделение - Стационарная карта (n...1)</p> <p>Признак жительства - Стационарная карта (n...1)</p> <p>Страна - Стационарная карта (n...1)</p> <p>Социальный статус - Стационарная карта (n...1)</p> <p>Тип оплаты - Стационарная карта (n...1)</p> <p>Направитель - Стационарная карта (n...1)</p> <p>Тип госпитализации - Стационарная карта (n...1)</p> <p>Тип доставки - Стационарная карта (n...1)</p> <p>Диагноз - Стационарная карта (n...1)</p> <p>Вид травмы - Стационарная карта (n...1)</p> <p>Период доставки - Стационарная карта (n...1)</p> <p>Признак выписки - Стационарная карта (n...1)</p> <p>Исход лечения - Стационарная карта (n...1)</p> <p>Отказ - Стационарная карта (n...1)</p>

	<ul style="list-style-type: none"> <li>- признак выписки;</li> <li>- исход лечения;</li> <li>- признак отказа;</li> </ul>		
Движение	Содержит данные о движении пациента: <ul style="list-style-type: none"> <li>- номер движения;</li> <li>- признак отказа;</li> <li>- отделение;</li> <li>- дата и время поступления;</li> <li>- дата и время выписки;</li> <li>- номер карты.</li> </ul>	Номер движения (РК)	Стационарная карта - Движение (n...1) Отделение - Движение (n...1) Признак выписки - Движение (n...1)
Смерть	<ul style="list-style-type: none"> <li>- номер карты;</li> <li>- время смерти;</li> <li>- признак вскрытия;</li> <li>- тип вскрытия;</li> <li>- диагноз вскрытия.</li> </ul>	Номер карты (РК)	Стационарная карта - Смерть (1...1) Тип вскрытия - Смерть (n...1)
Изменение оплаты	<ul style="list-style-type: none"> <li>- номер изменения;</li> <li>- номер карты;</li> <li>- старый вид оплаты;</li> <li>- новый вид оплаты;</li> <li>- пользователь;</li> <li>- дата и время изменения.</li> </ul>	Номер изменения (РК)	Стационарная карта - Изменение оплаты (n...1) Вид оплаты - Изменение оплаты (n...1) Пользователь - Изменение оплаты (n...1)

На следующем этапе была спроектирована диаграмма классов, которая описывает требования к внутренней организации проектируемого программного обеспечения на уровне интерфейсов и классов, их кооперации, атрибутов и методов классов, правил их взаимодействия.

## 2.3 Конструирование прототипа

Для построения прототипа первоначально были составлены требования к макету приложения, для которого разрабатываются программные модули.

Все компоненты должны иметь единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

- разметка и дизайн (предпочтение отдается масштабируемой компоновке; должно присутствовать ограничение на минимальный размер окна; должна присутствовать возможность изменения размеров окна, где это необходимо; увеличение размеров окна должно увеличивать размер контентной части, например, таблицы с данными из БД);
- группировка элементов (в логические категории);
- использование соответствующих элементов управления (например, выпадающих списков для отображения подстановочных значений из базы данных);
- расположение и выравнивание элементов (метки, поля для ввода и т.д.);
- последовательный переход фокуса по элементам интерфейса (по нажатию клавиши TAB);
- общая компоновка логична, понятна и проста в использовании;
- последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами или страницами в приложении (в том числе обратно, например, с помощью кнопки «Назад»);
- соответствующий заголовок на каждом окне приложения.

Основные требования руководства по стилю:

- все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом, представленном на рисунке 2.7. Цвет, размер и пропорции логотипа не должны изменяться при использовании программы. Так же для приложения должна быть установлена иконка, соответствующая логотипу;
- тип шрифта – Roboto;

– цветовая схема предусматривает использование в качестве основного фона – белый цвет #FFFFFF, в качестве дополнительного – синий цвет #05AEF6, для акцентирования внимания пользователя на целевое действие интерфейса – чёрный цвет #000000.



Рисунок 2.7 – Логотип приложения

С учетом требований к макету и руководству по стилю, для обеспечения требуемых функций, был разработан прототип программы.

На рисунке 2.8 представлен прототип главного окна программы и страницы регистрации пациента.

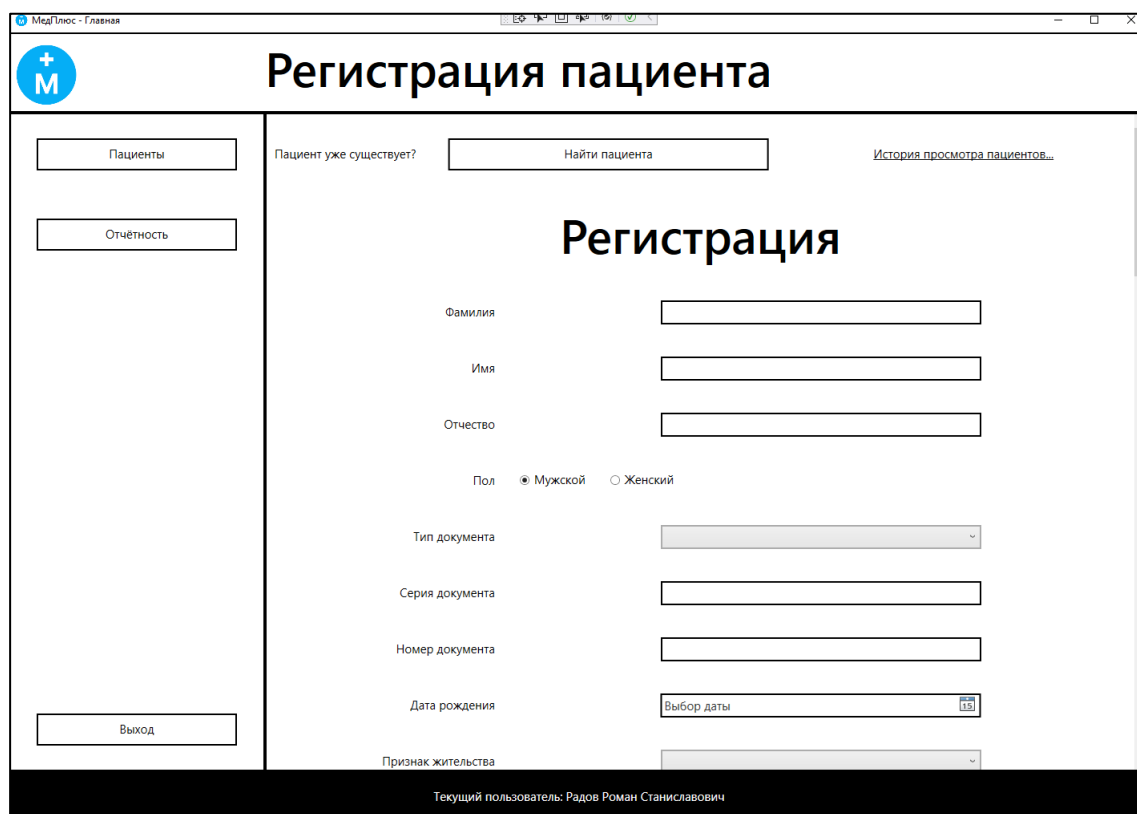


Рисунок 2.8 – Прототип главного окна программы и страницы регистрации пациента

На рисунке 2.9 представлен прототип страницы пациентов, предназначенной для поиска нужной карты пациента.

Поиск по имени:

Обновить

Иванов Иван Иванович

Пол:

Мужской

Номер карты:

3

Дата рождения:

7/10/1962 12:00:00 AM

Редактировать

Просмотреть

Иванов Иван Иванович

Пол:

Мужской

Номер карты:

4

Дата рождения:

5/23/1969 12:00:00 AM

Редактировать

Просмотреть

Следующие 200 записей

Рисунок 2.9 – Прототип страницы пациентов

На рисунке 2.10 представлен прототип страницы редактирования карты пациента.

Назад		Движения пациента	
Фамилия	<input type="text" value="Иванов"/>		
Имя	<input type="text" value="Иван"/>		
Отчество	<input type="text" value="Иванович"/>		
Пол	<input checked="" type="radio"/> Мужской <input type="radio"/> Женский		
Тип документа	<input type="text" value="Паспорт"/>		
Серия документа	<input type="text" value="1234"/>		
Номер документа	<input type="text" value="100000"/>		
Дата рождения	<input type="text" value="10.07.1962"/>		
Признак жительства	<input type="text" value="сельск.район города"/>		
Сохранить		Отмена	

Рисунок 2.10 – Прототип страницы редактирования карты пациента

На рисунке 2.11 представлен прототип страницы редактирования движения пациента.

Назад		Сохранить изменения	
Номер карты	3		
Отделение	<input type="text" value="ТО 1"/>		
Дата время перевода	1/1/2023 12:00:00 AM		
Дата время выписки	<input type="text" value="17 января 2023 г. 0:00:00"/>		
Признак выписки	<input type="text" value="Переведен в другой стационар"/>		
Добавить		Удалить последнее	

Рисунок 2.11 – Прототип страницы редактирования движения пациента

### 3 Реализация

#### 3.1 Обоснование выбора средств разработки

На таблице 3.1 представлен список средств разработки, с помощью которых возможна разработка программной системы.

Таблица 3.1 – Средства разработки

Среда разработки	Язык программиро	СУБД
Visual Studio	C#	SSMS
React Native	Java Script	MySQL
1С:Предприятие	1C	SSMS

Исходя из анализа требований к программной и информационной совместимости, описанных в подразделе 4.4 приложения А, было принято решение выбрать первый вариант, а именно технологию WPF, т.к. C#, WPF и SSMS изначально были заточены под ОС Windows.

#### 3.2 Разработка базы данных в среде СУБД

На основе ранее спроектированной ER-диаграммы в среде СУБД SSMS была разработана БД с именем MedPlusDB. На рисунке 3.1 представлена диаграмма базы данных, созданная средствами SSMS.



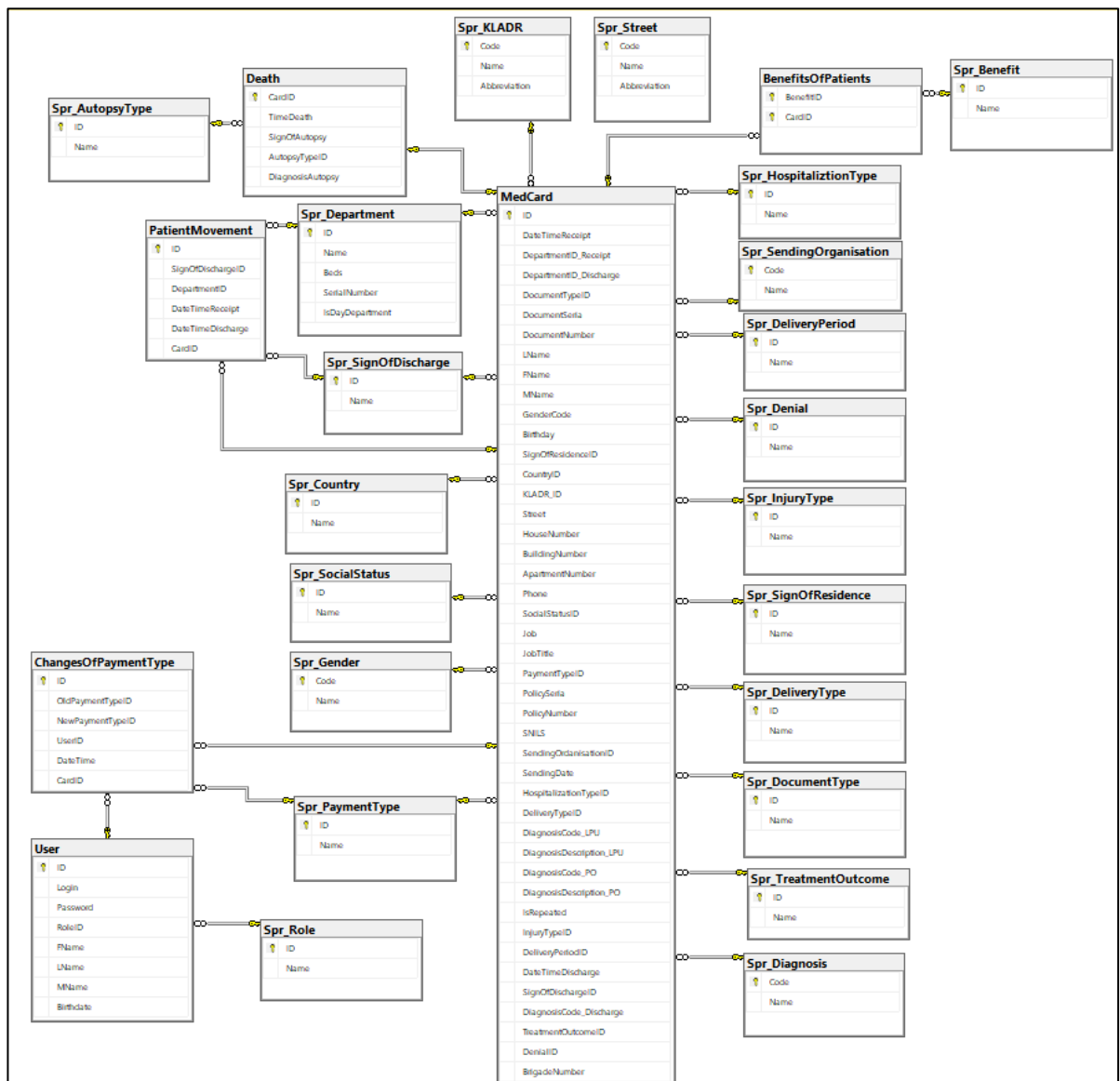


Рисунок 3.1 – Диаграмма БД в SSMS

На таблице 3.2 представлен перечень таблиц БД с кратким описанием.

Таблица 3.2 – Перечень таблиц БД

Имя таблицы	Описание
BenefitsOfPatients	Хранит льготы пациентов
ChangesOfPaymentType	В таблице содержатся все факты изменения вида оплаты у пациентов
Death	Хранит информацию о вскрытии умерших пациентов
MedCard	Хранит информацию о факте лечения пациента в больнице
PatientMovement	Хранит информацию о движениях пациентов
Spr_AutopsyType	Справочник типов вскрытия
Spr_Benefit	Справочник льгот
Spr_Country	Справочник стран
Spr_DeliveryPeriod	Справочник периодов доставки
Spr_DeliveryType	Справочник типов доставки
Spr_Denial	Справочник отказов
Spr_Department	Справочник отделений
Spr_Diagnosis	Справочник диагнозов
Spr_DocumentType	Справочник типов документов
Spr_Gender	Справочник полов
Spr_HospitalizationType	Справочник типов госпитализации
Spr_InjuryType	Справочник типов травм
Spr_KLADR	Справочник КЛАДР (адресов)
Spr_PaymentType	Справочник видов оплаты
Spr_Role	Справочник ролей
Spr_SendingOrganisation	Справочник организаций-направителей
Spr_SignOfDischarge	Справочник признаков выписки
Spr_SignOfResidence	Справочник признаков жительства
Spr_SocialStatus	Справочник социальных статусов
Spr_Street	Справочник улиц
Spr_TreatmentOutcome	Справочник исходов лечения
User	Хранит информацию о пользователях системы

### 3.3 Описание программных модулей

На рисунке 3.2 показана структура проекта приложения MedPlus.

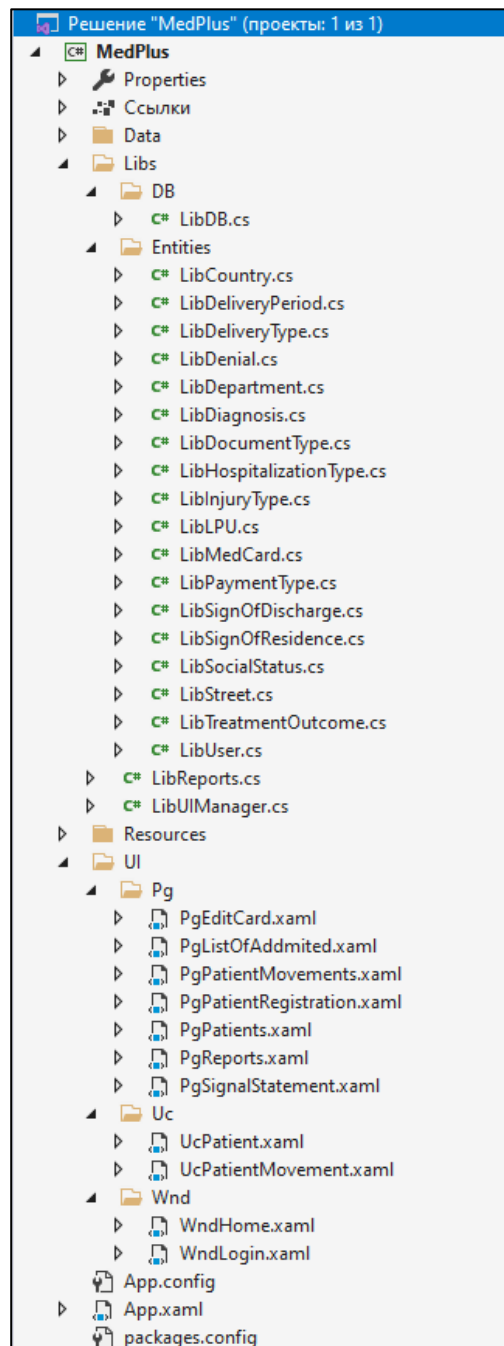


Рисунок 3.2 – Структура приложения MedPlus

На рисунке 3.3 представлено окно входа в систему. Код разметки окна представлен на рисунке Б.1. Код обработчика BtnLogin\_Click для кнопки «Войти» представлен на рисунке Б.2. Результат работы обработчика показан на рисунке 3.4.

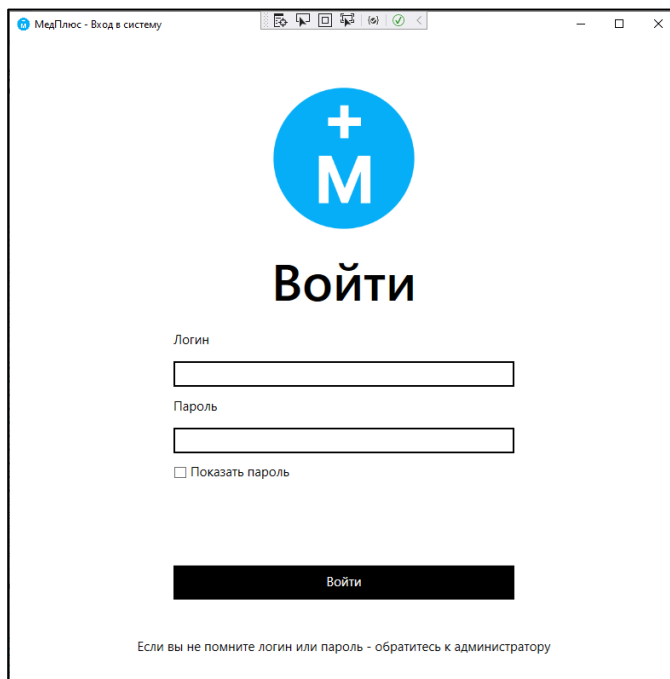


Рисунок 3.3 – Экранная форма окна «Вход в систему»

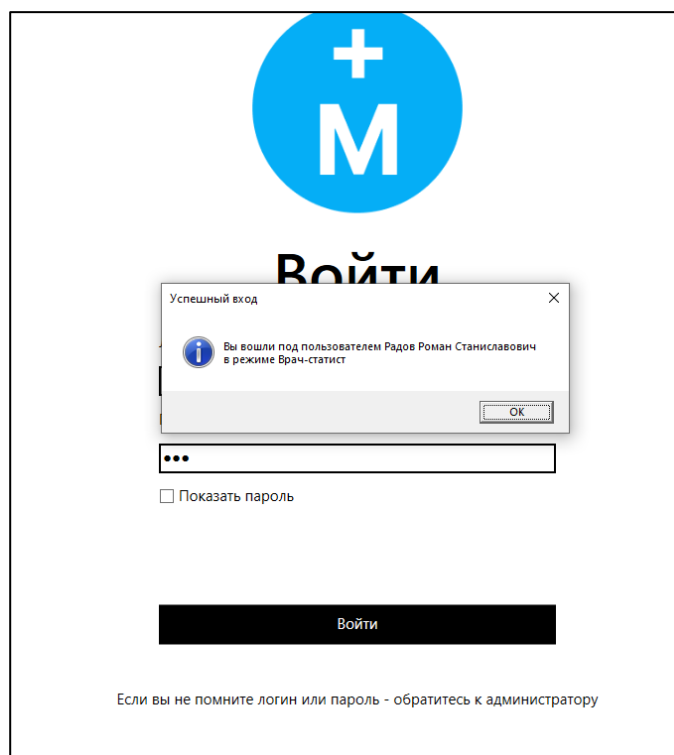


Рисунок 3.4 – Результат входа в окне «Вход в систему»

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0139 ПЗ

Лист

21

На рисунке 3.5 представлено рабочее окно пользователя. Код разметки окна представлен на рисунке Б.3, код обработчика BtnReports\_Click для кнопки «Отчёты» представлен на рисунке Б.4, код обработчика BtnPatients\_Click для кнопки «Пациенты» представлен на рисунке Б.5, код обработчика BtnExit\_Click для кнопки «Выход» представлен на рисунке Б.6.

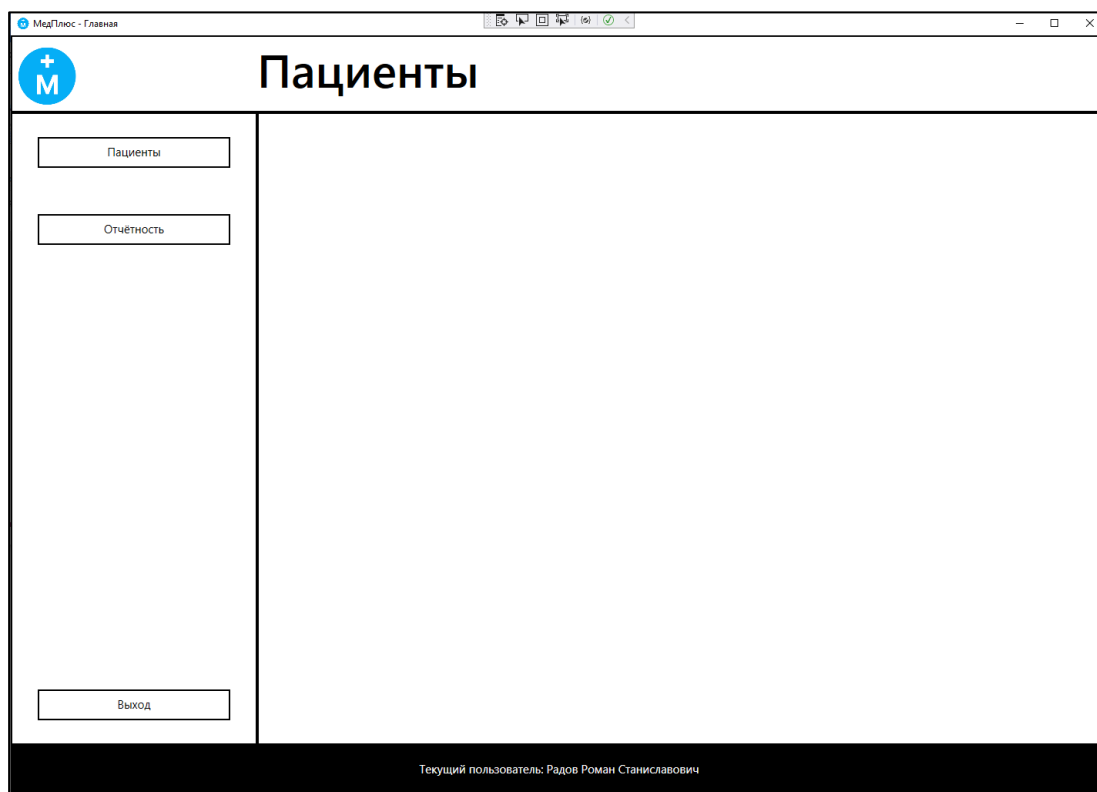


Рисунок 3.5 – Экранная форма окна «Главная»

На рисунке 3.6 показана страница «Регистрация пациента». Код разметки представлен на рисунке Б.7. Код обработчика BtnFindPatient\_Click кнопки «Найти пациента» показан на рисунке Б.8. Код обработчика BtnSave\_Click кнопки «Сохранить» показан на рисунке Б.9. Код обработчика BtnReset\_Click кнопки «Очистить поля» показан на рисунке Б.10.

Пациент уже существует?  [История просмотра пациентов...](#)

## Регистрация

Фамилия

Имя

Отчество

Пол ☒ Мужской ☐ Женский

Тип документа

Серия документа

Номер документа

Дата рождения

Признак жительства

Текущий пользователь: Радов Роман Станиславович

Рисунок 3.6 – Экранная форма страницы регистрации пациента

На рисунке 3.7 представлена страница списка пациентов. Код разметки представлен на рисунке Б.11. Код обработчика BtnUpdate\_Click кнопки «Обновить» показан на рисунке Б.12. Код обработчика TbSearch\_TextChanged текстового поля поиска показан на рисунке Б.13.

Поиск по имени:  Обновить

Иванов Иван Иванович

Пол: Мужской

Номер карты: 3

Дата рождения: 7/10/1962 12:00:00 AM

Редактировать Просмотреть

Иванов Иван Иванович

Пол: Мужской

Номер карты: 4

Дата рождения: 5/23/1969 12:00:00 AM

Редактировать Просмотреть

Следующие 200 записей

Рисунок 3.7 – Экранная форма страницы списка пациентов

На рисунке 3.8 показана страница редактирования карты пациента. Код разметки представлен на рисунке Б.14. Код обработчика BtnOpenMovements\_Click кнопки «Движения пациента» показан на рисунке Б.15. Код обработчика BtnSave\_Click кнопки «Сохранить» показан на рисунке Б.16.

Назад

Движения пациента

Фамилия: Иванов

Имя: Иван

Отчество: Иванович

Пол: ☒ Мужской ☐ Женский

Тип документа: Паспорт

Серия документа: 1234

Номер документа: 100000

Дата рождения: 10.07.1962

Признак жительства: сельск.район города

Сохранить

Отмена

Рисунок 3.8 – Экранная форма страницы «Редактирование карты пациента»



На рисунке 3.9 показана страница движений пациента. Код разметки представлен на рисунке Б.17 Код обработчика события BtnSaveChanges\_Click кнопки «Сохранить» представлен на рисунке Б.18. Код обработчика события BtnDeleteLast\_Click кнопки «Удалить последнее» показан на рисунке Б.19. Код обработчика события BtnAdd\_Click кнопки «Добавить» представлен на рисунке Б.20.

<input type="button" value="Назад"/>		<input type="button" value="Сохранить изменения"/>
Номер карты	3	
Отделение	ТО 1	
Дата время перевода	1/1/2023 12:00:00 AM	
Дата время выписки	17 января 2023 г. 0:00:00	
Признак выписки	Переведен в другой стационар	
<input type="button" value="Добавить"/>	<input type="button" value="Удалить последнее"/>	

Рисунок 3.9 – Экранная форма страницы «Движения пациента»

На рисунке 3.10 представлена страница отчёта сигнальной ведомости. Код разметки представлен на рисунке Б.21. Код обработчика BtnGenerate\_Click представлен на рисунке Б.22.

Назад

03.01.2023

15

Сформировать

Отделение	На начало	Поступило	Переведено из	Переведено в	Выписано	Умерло	На конец
КО-1	29	0	2	0	3	0	28
КО-2	12	0	7	0	0	0	19
ОНРС	15	0	0	0	0	0	15
КХО-1	8	0	0	0	0	0	8
КХО-2	37	0	0	1	2	0	34
ТХО-1	25	0	0	0	1	0	24
ТХО-2	6	0	0	0	0	0	6
Онкологическое	7	0	0	0	0	0	7
КО-3	25	0	3	0	5	0	23
Неврологическое отделение	4	0	1	0	0	0	5
Неврологическое отделение ОНМК	46	0	4	0	4	0	46
Отделение мед. реабилит пациентов с нарушением ЦНС	3	0	0	0	0	0	3
Пульмонология	12	0	0	0	0	1	11
ГЭТО	18	0	1	0	0	0	19
Нефрология-1	8	0	0	0	0	0	8
Нефрология-2	12	0	0	0	0	0	12
Эндокринология	1	0	0	0	0	0	1
Ревматология	6	0	1	0	0	0	7
НХО-1	17	0	0	0	2	0	15
НХО-2	39	0	1	0	1	0	39
НХО-3 (с поражен. спинного мозга)	21	0	3	0	1	0	23
Гинекология	18	0	3	0	5	0	16
Офтальмологическое	4	0	3	0	0	0	7
Колопроктология	25	0	1	0	0	0	26
ТО 1	36	0	7	0	5	0	38
ТО 2	27	0	1	0	4	0	24
ТО-3	26	0	0	0	9	0	17
Сосудистая хирургия-1	19	0	1	0	4	0	16
Сосудистая хирургия-2	19	0	3	0	0	0	22
Урология-1	9	0	0	0	1	0	8
Отделение онкоурологии	20	0	0	0	2	0	18
Урология-2	17	0	1	0	3	0	15
Хирургическое-1	51	0	5	0	1	1	54
Хирургическое-2	26	0	6	0	6	0	26
Ожоговое	26	0	3	0	0	0	29
Гнойной хирургии	25	0	5	0	1	0	29

Рисунок 3.10 – Экранная форма страницы «Сигнальная ведомость»

## 4 Тестирование программных модулей»

### 4.1 Модульное тестирование

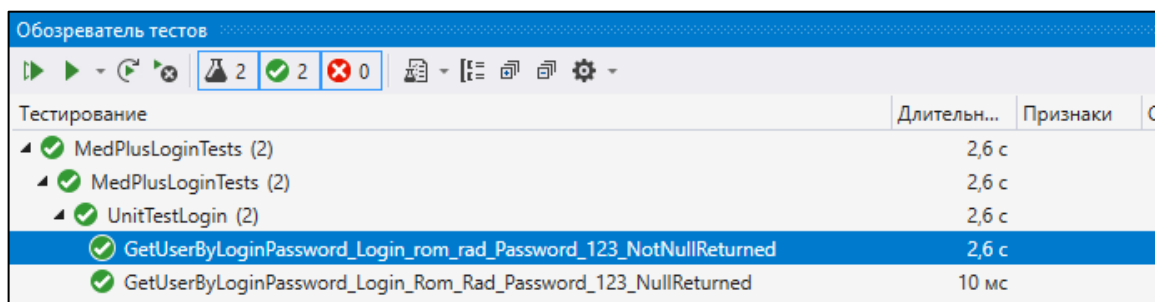
В рамках модульного тестирования были поставлены следующие задачи:

- выявить основные модули приложения;
- составить модульные тесты (в том числе с использованием средств автоматизации);
- сделать вывод исходя из результатов тестирования.

К основным модулям приложения можно отнести модули:

- входа в систему;
- модуль пациента.

Для модуля «Вход в систему» созданы два Unit-теста. Текст Unit-тестов показан на рисунке Б.23. Результат выполнения Unit-тестов показан на рисунке 4.1.



Тестирование	Длительн...	Признаки	С
MedPlusLoginTests (2)	2,6 с		
MedPlusLoginTests (2)	2,6 с		
UnitTestLogin (2)	2,6 с		
GetUserByLoginPassword_Login_rom_rad_Password_123_NullReturned	2,6 с		
GetUserByLoginPassword_Login_Rom_Rad_Password_123_NullReturned	10 мс		

Рисунок 4.1 – Результаты тестирования модуля входа в систему

Для модуля пациента были созданы два тест-кейса, показанные на рисунках Г.2 и Г.3.

## 4.2 Интеграционное тестирование

Задачей интеграционного тестирования является тестирование взаимодействия модулей пациента и движений пациента. Для этого были созданы два тест-кейса, показанные на рисунках Г.4 и Г.5.

В ходе тестирования были выявлены дефекты приложения, а именно:

- система не запрещает регистрацию пациента с незаполненными обязательными полями.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

## 5 Эксплуатационная документация

### 5.1 Руководство пользователя.

#### 5.1.1 Назначение программных модулей системы

Программные модули системы МедПлюс предназначены для автоматизации процесса анализа движений пациентов в лечебных отделениях больницы.

#### 5.1.2 Условия применения программных модулей системы

Программные модули системы применяются в компьютерной сети, состоящей из клиентских компьютеров и общего сервера. Клиентская программа выполняется на рабочих станциях пользователей. На сервере должен быть установлен Microsoft SQL Server Express 2019.

#### 5.1.3 Сведения по подготовке программного обеспечения системы к работе

Необходимо загрузить скрипт базы данных на удалённый сервер. Скрипт БД представлен на картинках В.1 – В.5. Затем на клиентские компьютеры необходимо скопировать папки с программой и в файле app.config поменять строку подключения к базе данных.

#### 5.1.4 Описание операций

- вход в систему. Для входа в систему необходимо ввести существующий логин и соответствующий пароль. Результатом операции является переход на главное окно в соответствии с ролью пользователя. Если логин или пароль некорректные, система выведет сообщение об ошибке входа.

- регистрация пациента. Регистрация пациента предусматривает заполнение полей регистрации. Все поля являются необязательными, кроме типа госпитализации. Номер карты создаётся автоматически и его нельзя поменять. После заполнения регистрационных данных нажмите кнопку «Сохранить» и, если обязательные данные заполнены, система сохранит карту в БД и уведомит об успешном сохранении.

- горячий неточный поиск пациента по имени. Для поиска пациента по имени введите ФИО искомого пациента в поле поиска.

- редактирование карты пациента. Для редактирования существующей карты пациента нажмите на кнопку «Редактировать» у нужной карты в списке медкарт пациентов. После этого откроется страница редактирования медкарты, в которой вы можете изменить значения большинства полей (кроме номера медкарты).

- редактирование движений пациента. Для редактирования движений пациента откройте страницу редактирования медкарты, по которой необходимо изменить движения. Затем нажмите на кнопку «Движения пациента». После этого откроется страница добавления/удаления/редактирования движений пациента. Для добавления движения нажмите кнопку «Создать», для удаления движения нажмите кнопку «Удалить последнее». Если у последнего движения не заполнена дата и время выписки, система откажет в создании нового движения и попросит указать дату и время выписки последнего движения. Для сохранения изменений в БД нажмите кнопку «Сохранить изменения». До тех пор изменения в БД не происходят.

- отчёты. Для открытия страницы со списком отчётов, на главном экране нажмите кнопку «Отчёты». В списке отчётов выберите нужный. Откроется страница отчёта. Как правило, большинство отчётов имеют параметры. Обязательные параметры выделены жирным шрифтом. После заполнения параметров отчёта нажмите кнопку «Сформировать», после чего программа отобразит печатную форму отчёта. Если обязательные параметры не заполнены, система сообщит вам о том, какие параметры надо заполнить, при этом отчёт сформирован не будет.

#### 5.1.5 Действия пользователя при аварийных ситуациях

В случае аварийной ситуации обратитесь к системному администратору.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

## Заключение

В результате проделанной работы в рамках курсового проектирования разработаны программные модули для автоматизации процесса анализа движения пациентов в лечебных отделениях больницы. Разработка программных модулей системы велась согласно требованиям ГОСТ 19.102-77 «Единая система программной документации (ЕСПД). Стадии разработки». В начале выполнения проекта были поставлены следующие задачи:

- сформулировать назначение и цели разработки;
- разработать технический проект на основе анализа требований;
- создать реализацию программных;
- составить план тестирования, провести модульное и интеграционное тестирование и составить тестовую документации;
- составить эксплуатационную документацию (руководство пользователя).

В ходе разработки технического проекта были выбраны технологии проектирования и построения моделей (Visio). При выборе технологии проектирования был выбран объектно-ориентированный подход.

Затем были определены спецификации программного обеспечения. Были выделены активные участники бизнес-процесса и соответствующие прецеденты. На основе этих данных построены диаграмма вариантов использования, диаграмма деятельности и диаграмма последовательностей.

Далее построены модели данных и диаграммы классов. В качестве модели базы данных выбрана реляционная модель.

После определения спецификаций программного обеспечения и построения модели данных, выполнено конструирование прототипа, в ходе которого перечислены требования к макету и составлено руководство по стилю.

Затем был выполнен этап реализации, в ходе которого был сделан и обоснован выбор средств разработки, разработана БД и клиентское приложение.



На этапе тестирования составлен план тестирования, по которому затем были созданы три тест-кейса и два unit-теста. В ходе модульного и интеграционного тестирования были выявлены дефекты приложения, возникшие в процессе разработки.

После тестирования и устранения дефектов была составлена эксплуатационная документация в виде руководства пользователя.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

## Список использованных источников

1. ГОСТ 2.105 – 2019. Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам (Издание с Изменением N 1) = Unified system for design documentation. General requirements for textual documents: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 29 апреля 2019 г. № 175-ст: введен впервые: дата введения 2021-02-01 / Разработан Федеральным государственным унитарным предприятием «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ»). – Москва.: Стандартинформ, 2021. – 35 с. –Текст непосредственный.

2. ГОСТ 7.0.100 – 2018. Система стандартов по информации, библиотечному и издательскому делу (СИБИД). Библиографическая запись. Библиографическое описание. Общие требования и правила составления (с Поправкой) = System of standards on information, librarianship and publishing. Bibliographic record. Bibliographic description. General requirements and rules: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 года № 1050-ст: введен впервые: дата введения 2019-07-01 / Разработан Федеральным государственным унитарным предприятием "Информационное телеграфное агентство России (ИТАР-ТАСС)", филиал "Российская книжная палата", Федеральным государственным бюджетным учреждением "Российская государственная библиотека", Федеральным государственным бюджетным учреждением "Российская национальная библиотека". – Москва: Стандартинформ, 2018. – 128 с. – Текст непосредственный.

3. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 0139 ПЗ

Лист

35

N 1) = Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3351: введен впервые: дата введения 1980-01-01. – Москва: Стандартиформ, 2010. – 4 с. – Текст непосредственный.

4. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: руководящий документ по стандартизации: издание официальное: утверждены и введены в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 27 декабря 1990 г. № 3380: дата введения 1992-01-01 / Разработан Министерством электротехнической промышленности и приборостроения СССР. – Москва.: ИПК Издательство стандартов, 2002 г. – 27 с. –Текст непосредственный.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

## Приложение А

### Техническое задание. Требования к программным модулям

#### 1 Введение

##### 1.1 Наименование программы

Наименование программной системы «МедПлюс»

##### 1.2 Краткая характеристика области применения

Программная система «МедПлюс» предназначена для автоматизации процесса анализа движения пациентов в лечебных отделениях больницы.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

## 2 Основания для разработки

Основанием для разработки является Договор №3 от 28.04.2023. Договор утвержден Директором краевой клинической больницы №3 Белых Сергеем Викторовичем, именуемым в дальнейшем Заказчиком, и Молокотиним Вячеславом Сергеевичем (самозанятый), именуемым в дальнейшем исполнителем.

Согласно Договору, Исполнитель обязан разработать и установить программную систему «МедПлюс» на оборудовании Заказчика не позднее 10.07.2023, предоставить исходные коды и документацию к разработанной системе не позднее 25.08.2023.

Наименование темы разработки – «Разработка программных модулей тестовой системы «МедПлюс».

Условное обозначение темы разработки (шифр темы) – «МедПлюс\_01».

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

### 3 Назначение разработки

#### 3.1 Функциональное назначение

Программная система будет использоваться следующими категориями пользователей: врачи, средний медперсонал (медсестра, медбрат), младший медперсонал (медрегистраторы), врачи-статисты, экономисты.

Для каждой категории пользователей существует соответствующая роль. Далее приведён перечень функциональных возможностей каждой роли.

Врачи:

- просмотр данных пациентов;
- просмотр данных движений пациентов.

Средний медперсонал:

- просмотр данных пациентов;
- просмотр данных движений пациентов.

Младший медперсонал:

- просмотр данных пациентов;
- просмотр данных движений пациентов;
- создание, редактирование и удаление движений пациента (до следующего дня после выписки);
- создание и редактирование движений пациента.

Врачи-статисты:

- просмотр данных пациентов;
- просмотр данных движений пациентов;
- просмотр отчётности;
- создание и редактирование стационарной медицинской карты пациента;
- создание, редактирование и удаление движений пациента.

Экономисты:

- просмотр данных пациентов;
- просмотр данных движений пациентов;
- просмотр отчётности.

Программная система должна обеспечить возможность регистрировать поступающих пациентов, т.е. составлять стационарную карту на время лечения, фиксировать движения пациентов по стационарным отделениям больницы и вносить дополнительную информацию об исходе лечения, а также просматривать отчётность больницы, связанную с движениями и исходом лечения пациентов. Удаление стационарных медицинских карт пациентов должно быть запрещено.

### 3.1 Эксплуатационное назначение

Программная система будет использоваться на рабочих местах пользователей, представляющих собой персональные компьютеры, имеющие доступ к сети внутренней рабочей сети.

## 4 Требования к программе

### 4.1 Требования к функциональным характеристикам

#### 4.1.1 Требования к составу выполняемых функций

В разрабатываемой системе должен быть реализован следующий функционал:

- регистрация поступления пациента в больницу (регистрация медицинской карты). Требования к данным медицинской карты описаны в подразделе 4.1.2. Информация о карте вводится пользователем вручную на отдельной странице приложения;

- редактирование созданной карты. Редактирование осуществляется в соответствии с полномочиями ролей пользователей, описанными в подразделе 3.1. Изменение карты данных карты для младшего персонала возможно лишь в течение дня после выписки пациента. Врач-статист может редактировать медицинскую карту в любой момент времени;

- создание движений пациентов. Первое движение создаётся только после составления медицинской карты пациента. Последующие движения идут последовательно друг за другом и могут быть изменены и только удалены (в соответствии с полномочиями ролей пользователей). Удаление может происходить только последовательно – от самой новой записи к самой старой. Должна присутствовать возможность удалять только последнее на данный момент движение;

- просмотр отчётной документации. В зависимости от типа отчёта, пользователь может задавать соответствующие параметры перед формированием. Если обязательные параметры не были заполнены, программа должна вывести пользователю сообщение об ошибке, при этом отчёт не должен быть сформирован;



– вход в систему. Для входа в систему после запуска программы пользователь должен ввести логин и пароль, который закреплён за ним.

#### 4.1.2 Требования к организации входных и выходных данных

Требования к входным и выходным данным описаны на основе данных собеседования с заказчиком.

Стационарная медицинская карта пациента:

– входные и выходные данные: ФИО пациента, день рождения, пол, даты и время поступления и выписки, отделения поступления и выписки (справочник), тип подтверждающего документа (справочник), серия документа, номер документа, дата рождения, признак жительства (справочник), страна (справочник), адрес, телефон, социальный статус (справочник), место работы, должность, тип оплаты (справочник), серия и номер полиса, номер СНИЛС, организация-направитель (справочник), дата направления, тип госпитализации (справочник), тип доставки (справочник), номер бригады скорой помощи, диагноз направителя (справочник), диагноз приёмного отделения (справочник), диагноз выписки (справочник), признак повторной госпитализации, вид травмы (справочник), период доставки с момента начала заболевания (справочник), признак выписки (справочник), исход лечения (справочник), код отказа (справочник).

Движение пациента:

– входные и выходные данные: признак выписки, отделение, дата и время поступления и выписки, номер медицинской карты.

Сигнальная ведомость:

– выходные данные: отделение, количество пациентов на начало дня, количество поступивших, количество переведённых из отделения, количество переведённых в отделение, количество выписанных, количество умерших, количество пациентов на конец периода;

– входные данные: дата отчёта.

Отчёт по поступившим:

– выходные данные: номер медкарты, дата и время поступления, отделение, ФИО пациента;

– входные данные: период начала, период окончания.

#### 4.2 Требования к надежности и безопасности

Так как система содержит конфиденциальные данные пациентов, к системе должны быть предъявлены повышенные требования к безопасности системы. С целью исключения несанкционированного доступа к данным, за каждым пользователем системы должен быть закреплён свой индивидуальный аккаунт, состоящий из логина и пароля, зашифрованного алгоритмом шифрования MD5.

#### 4.3 Требования к составу и параметрам технических средств

Состав технических средств предусматривает наличие компьютеров-рабочих станций и компьютеров-серверов.

Компьютер-рабочая станция включает:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- видеокарту, монитор, мышь, клавиатура.

Два компьютера-сервера (для функционирования СУБД) (основной и резервный), включают в себя:

- процессор x86 с тактовой частотой, не менее 3 ГГц;
- оперативную память объемом, не менее 16 Гб;
- видеокарту, монитор, мышь.

#### 4.4 Требования к информационной и программной совместимости

На серверах должны быть установлены ОС Windows Server 2019, сервер БД Microsoft SQL Server Express 2019, SQL Server Management Studio 18.8.

На компьютерах-рабочих станциях должна быть установлена операционная система ОС Microsoft Windows 10 Pro.

					<i>ККЭП 09.02.07 0139 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

## Приложение Б

### Программный код

Программный код представлен в контексте текстового редактора MS Visual Studio 2019.

```
1 <Window x:Class="MedPlus.UI.Wnd.WndLogin"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:local="clr-namespace:MedPlus.UI.Wnd"
7     mc:Ignorable="d"
8     Title="МедПлюс - Вход в систему"
9     Icon="/Resources/Assets/Icon.png"
10    Height="1024" Width="1440"
11    MinHeight="800" MinWidth="800"
12    SizeChanged="Window_SizeChanged"
13    WindowStartupLocation="CenterScreen"
14    Loaded="Window_Loaded">
15    <StackPanel Orientation="Vertical" HorizontalAlignment="Center" VerticalAlignment="Center">
16        <StackPanel Orientation="Vertical" HorizontalAlignment="Center">
17            <Image x:Name="ImgLogo" Source="/Resources/Assets/Icon.png" Height="300"></Image>
18            <Label>Войти</Label>
19
20            <TextBlock HorizontalAlignment="Left" Style="{StaticResource TblBase}">Логин</TextBlock>
21            <TextBox x:Name="TbLogin"></TextBox>
22            <TextBlock HorizontalAlignment="Left" Style="{StaticResource TblBase}">Пароль</TextBlock>
23            <PasswordBox x:Name="PwbxPassword"></PasswordBox>
24            <TextBox x:Name="TbPassword" Visibility="Collapsed"></TextBox>
25            <CheckBox x:Name="ChbxPassword" HorizontalAlignment="Left" Click="ChbxPassword_Click">Показать пароль</CheckBox>
26
27            <Button Margin="0,100,0,20" x:Name="BtnLogin" Style="{StaticResource BtnBlack}" Click="BtnLogin_Click">Войти</Button>
28            <TextBlock FontSize="20"></TextBlock>
29        </StackPanel>
30
31        <StackPanel Orientation="Vertical" HorizontalAlignment="Center">
32            <TextBlock Style="{StaticResource TblBase}">Если вы не помните логин или пароль - обратитесь к администратору</TextBlock>
33        </StackPanel>
34    </StackPanel>
35</Window>
```

Рисунок Б.1 – Код разметки окна «Вход в систему»

```

43  /// <summary>
44  /// Обработчик нажатия кнопки "Войти". Иницирует процесс аутентификации и авторизации пользователя пользователя.
45  /// </summary>
46  /// <param name="sender"></param>
47  /// <param name="e"></param>
48  ссылка: 1
49  private void BtnLogin_Click(object sender, RoutedEventArgs e)
50  {
51      login();
52  }
53  /// <summary>
54  /// Метод. Иницирует процесс входа в систему.
55  /// </summary>
56  ссылка: 1
57  private void login()
58  {
59      string login = TbLogin.Text;
60      string password = (bool)ChbxPassword.IsChecked ? PwbxPassword.Password : PwbxPassword.Password;
61      User user = LibUser.GetUserByLoginPassword(login, password);
62
63      if (user != null)
64      {
65          LibUser.CurrentUser = user;
66
67          MessageBox.Show($"Вы вошли под пользователем {LibUser.GetFullName(user)} в режиме {user.Spr_Role.Name}",
68              "Успешный вход",
69              MessageBoxButton.OK,
70              MessageBoxImage.Information);
71
72          WndHome wnd = new WndHome
73          {
74              Owner = this
75          };
76          wnd.Show();
77          Hide();
78      }
79      else
80      {
81          MessageBox.Show($"Неправильный логин или пароль!",
82              "Ошибка при входе в систему",
83              MessageBoxButton.OK,
84              MessageBoxImage.Error);
85      }
86  }

```

Рисунок Б.2 – Код обработчика для кнопки «Войти» на окне «Вход в систему»

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3  xmlns:s="http://schemas.microsoft.com/winfx/2006/xaml"
4  xmlns:mc="http://schemas.microsoft.com/expression/blend/2008"
5  xmlns:local="clr-namespace:MedPlus.UI.Wnd"
6  mc:Ignorable="d"
7  Title="MedPlus - Главная"
8  Icon="/Resources/Assets/Icon.png"
9  Height="1024" Width="1440"
10 MinHeight="800" MinWidth="800"
11 Closing="Window_Closing"
12 WindowStartupLocation="CenterScreen"
13 </Window>
14 <Grid>
15     <Grid.ColumnDefinitions>
16         <ColumnDefinition Width="320"/>
17         <ColumnDefinition/>
18     </Grid.ColumnDefinitions>
19     <Grid.RowDefinitions>
20         <RowDefinition Height="100"/>
21         <RowDefinition/>
22         <RowDefinition Height="64"/>
23     </Grid.RowDefinitions>
24
25     <Border BorderThickness="2" BorderBrush="{StaticResource AccentColor}" Grid.ColumnSpan="2">
26         <Image Source="/Resources/Assets/Icon.png" Margin="10" HorizontalAlignment="Left"/>
27         <Label x:Name="LabHeader" Grid.Column="1" HorizontalAlignment="Left" Margin="0" Padding="0">Пациенты</Label>
28     </Border>
29
30     <Border Grid.Row="1" BorderThickness="2" BorderBrush="{StaticResource AccentColor}">
31         <DockPanel>
32             <StackPanel Orientation="Vertical" DockPanel.Dock="Top">
33                 <Button Style="{StaticResource BtnWhite}" Width="250" x:Name="BtnPatients" Click="BtnPatients_Click">Пациенты</Button>
34                 <Button Style="{StaticResource BtnWhite}" Width="250" x:Name="BtnReports" Click="BtnReports_Click">Отчётность</Button>
35             </StackPanel>
36             <Grid>
37                 <Button x:Name="BtnExit" Width="250" VerticalAlignment="Bottom" Style="{StaticResource BtnWhite}" Click="BtnExit_Click">Выход</Button>
38             </Grid>
39         </DockPanel>
40     </Border>
41
42     <Border Grid.Row="1" Grid.Column="1" BorderThickness="2" BorderBrush="{StaticResource AccentColor}">
43         <Frame x:Name="FrMain" NavigationUIVisibility="Hidden" Navigated="FrMain_Navigated"></Frame>
44     </Border>
45
46     <Grid Background="{StaticResource AccentColor}" Grid.Row="2" Grid.ColumnSpan="2">
47         <TextBlock x:Name="Tb1CurrentUser" Foreground="White" Style="{StaticResource Tb1Base}" Text="Текущий пользователь: "></TextBlock>
48     </Grid>
49 </Grid>

```

Рисунок Б.3 – Код разметки окна «Главная»

```

73
74
75
76
77
78
79
80
81
82
    /// <summary>
    /// Обработчик события кнопки "Отчёты". Открывает страницу со списком отчётов.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    ссылка: 1
    private void BtnReports_Click(object sender, RoutedEventArgs e)
    {
        FrMain.Navigate(new PgReports());
    }

```

Рисунок Б.4 – Код обработчика нажатия кнопки «Отчёты» на окне «Главная»

```

64
65
66
67
68
69
70
71
72
73
    /// <summary>
    /// Обработчик события кнопки "Выход". Открывает окно регистрации пациента.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    ссылка: 1
    private void BtnPatients_Click(object sender, RoutedEventArgs e)
    {
        FrMain.Navigate(new PgPatientRegistration());
    }

```

Рисунок Б.5 – Код обработчика нажатия кнопки «Пациенты» на окне «Главная»

```

52
53
54
55
56
57
58
59
60
61
    /// <summary>
    /// Обработчик события кнопки "Выход". Иницирует выход из программы.
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    ссылка: 1
    private void BtnExit_Click(object sender, RoutedEventArgs e)
    {
        Close();
    }

```

Рисунок Б.6 – Код обработчика нажатия кнопки «Выход» на окне «Главная»

```

1 <Page x:Class="MedPlus.UI.Pg.PgPatientRegistration"
2 xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3 xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4 xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5 xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6 xmlns:scx="http://schemas.xceed.com/wpf/xaml/toolkit"
7 xmlns:local="clr-namespace:MedPlus.UI.Pg"
8 mc:Ignorable="d"
9 d:DesignWidth="500" d:DesignHeight="1120"
10 Title="PgPatientRegistration">
11
12 <ScrollViewer>
13 <StackPanel Orientation="Vertical">
14 <DockPanel>
15 <TextBlock Style="{StaticResource TblBase}" DockPanel.Dock="Left" Margin="10">Пациент уже существует?</TextBlock>
16 <Button x:Name="BtnFindPatient" DockPanel.Dock="Left" Style="{StaticResource BtnWhite}" Click="BtnFindPatient_Click" Background="White">Найти пациента</Button>
17 <TextBlock x:Name="TblHistoryOfPatients" Style="{StaticResource TblBase}" Margin="10" DockPanel.Dock="Right" TextDecorations="Underline" MouseLeftButtonDown="TextBlock_MouseLeftButtonDown">История просмотра пациентов...</TextBlock>
18 </DockPanel>
19
20 <Label Content="Перепрочитай"></Label>
21
22 <Grid>
23 <Grid.RowDefinitions>
24 <RowDefinition Height="70"/>
25 <RowDefinition Height="70"/>
26 <RowDefinition Height="70"/>
27 <RowDefinition Height="70"/>
28 <RowDefinition Height="70"/>
29 <RowDefinition Height="70"/>
30 <RowDefinition Height="70"/>
31 <RowDefinition Height="70"/>
32 <RowDefinition Height="70"/>
33 <RowDefinition Height="70"/>
34 <RowDefinition Height="70"/>
35 <RowDefinition Height="70"/>
36 <RowDefinition Height="70"/>
37 <RowDefinition Height="70"/>
38 <RowDefinition Height="70"/>
39 <RowDefinition Height="70"/>
40 <RowDefinition Height="70"/>
41 <RowDefinition Height="70"/>
42 <RowDefinition Height="70"/>
43 <RowDefinition Height="70"/>
44 <RowDefinition Height="70"/>
45 <RowDefinition Height="70"/>
46 <RowDefinition Height="70"/>
47 <RowDefinition Height="70"/>
48 <RowDefinition Height="70"/>
49 <RowDefinition Height="70"/>
50 <RowDefinition Height="70"/>
51 <RowDefinition Height="70"/>
52 <RowDefinition Height="70"/>
53 <RowDefinition Height="70"/>
54 <RowDefinition Height="70"/>
55 <RowDefinition Height="70"/>
56 <RowDefinition Height="70"/>
57 <RowDefinition Height="70"/>
58 <RowDefinition Height="70"/>
59 <RowDefinition Height="70"/>
60 <RowDefinition Height="70"/>

```

Рисунок Б.7 – Код разметки страницы «Регистрация пациента» (1)

```

61 <RowDefinition Height="70"/>
62 <RowDefinition Height="70"/>
63 <RowDefinition Height="70"/>
64 <RowDefinition Height="70"/>
65 <RowDefinition Height="70"/>
66 <RowDefinition Height="70"/>
67 <RowDefinition Height="70"/>
68 <RowDefinition Height="70"/>
69 </Grid.RowDefinitions>
70 <Grid.ColumnDefinitions>
71 <ColumnDefinition Width="300"/>
72 <ColumnDefinition Width="300"/>
73 </Grid.ColumnDefinitions>
74
75 <TextBlock Style="{StaticResource TblBase}" Grid.Row="0" HorizontalAlignment="Right">Фамилия</TextBlock>
76 <TextBox x:Name="TblName" Grid.Column="1" Grid.Row="0" Text="{Binding LName}"></TextBlock>
77
78 <TextBlock Style="{StaticResource TblBase}" Grid.Row="1" HorizontalAlignment="Right">Имя</TextBlock>
79 <TextBox x:Name="TblFName" Grid.Column="1" Grid.Row="1" Text="{Binding FName}"></TextBlock>
80
81 <TextBlock Style="{StaticResource TblBase}" Grid.Row="2" HorizontalAlignment="Right">Отчество</TextBlock>
82 <TextBox x:Name="TblMName" Grid.Column="1" Grid.Row="2" Text="{Binding MName}"></TextBlock>
83
84 <TextBlock Style="{StaticResource TblBase}" Grid.Row="3" HorizontalAlignment="Right">Пол</TextBlock>
85 <StackPanel Orientation="Horizontal" Grid.Column="1" Grid.Row="3">
86 <RadioButton x:Name="RbtnMale" IsChecked="True">Мужской</RadioButton>
87 <RadioButton x:Name="RbtnFemale">Женский</RadioButton>
88 </StackPanel>
89
90 <TextBlock Style="{StaticResource TblBase}" Grid.Row="4" HorizontalAlignment="Right">Тип документа</TextBlock>
91 <ComboBox x:Name="CmbxDocumentType" Grid.Column="1" Grid.Row="4" SelectedItem="{Binding Spr_DocumentType}" DisplayMemberPath="Name"></ComboBox>
92
93 <TextBlock Style="{StaticResource TblBase}" Grid.Row="5" HorizontalAlignment="Right">Серия документа</TextBlock>
94 <TextBox x:Name="TblDocumentSeria" Grid.Column="1" Grid.Row="5" Text="{Binding DocumentSeria}"></TextBlock>
95
96 <TextBlock Style="{StaticResource TblBase}" Grid.Row="6" HorizontalAlignment="Right">Номер документа</TextBlock>
97 <TextBox x:Name="TblDocumentNumber" Grid.Column="1" Grid.Row="6" Text="{Binding DocumentNumber}"></TextBlock>
98
99 <TextBlock Style="{StaticResource TblBase}" Grid.Row="7" HorizontalAlignment="Right">Дата рождения</TextBlock>
100 <DatePicker x:Name="DtpkBirthday" Grid.Column="1" Grid.Row="7" SelectedDate="{Binding Birthday}"></DatePicker>
101
102 <TextBlock Style="{StaticResource TblBase}" Grid.Row="8" HorizontalAlignment="Right">Признак жительства</TextBlock>
103 <ComboBox x:Name="CmbxSignOfResidence" Grid.Column="1" Grid.Row="8" SelectedItem="{Binding Spr_SignOfResidence}" DisplayMemberPath="Name"></ComboBox>
104
105 <TextBlock Style="{StaticResource TblBase}" Grid.Row="9" HorizontalAlignment="Right">Страна</TextBlock>
106 <ComboBox x:Name="CmbxCountry" Grid.Column="1" Grid.Row="9" SelectedItem="{Binding Spr_Country}" IsEditable="True" DisplayMemberPath="Name"></ComboBox>
107
108 <TextBlock Style="{StaticResource TblBase}" Grid.Row="10" HorizontalAlignment="Right">Регион</TextBlock>
109 <TextBox x:Name="TblRegion" Grid.Column="1" Grid.Row="10"></TextBlock>
110
111 <TextBlock Style="{StaticResource TblBase}" Grid.Row="11" HorizontalAlignment="Right">Район</TextBlock>
112 <TextBox x:Name="TblArea" Grid.Column="1" Grid.Row="11"></TextBlock>
113
114 <TextBlock Style="{StaticResource TblBase}" Grid.Row="12" HorizontalAlignment="Right">Город</TextBlock>
115 <TextBox x:Name="TblCity" Grid.Column="1" Grid.Row="12"></TextBlock>
116
117 <TextBlock Style="{StaticResource TblBase}" Grid.Row="13" HorizontalAlignment="Right">Населённый пункт</TextBlock>
118 <TextBox x:Name="TblLocality" Grid.Column="1" Grid.Row="13"></TextBlock>
119
120 <TextBlock Style="{StaticResource TblBase}" Grid.Row="14" HorizontalAlignment="Right">Улица</TextBlock>

```

Рисунок Б.8 – Код разметки страницы «Регистрация пациента» (2)

```

121 <TextBox x:Name="TbStreet" Grid.Column="1" Grid.Row="14" Text="{Binding Street}"></TextBox>
122
123 <TextBlock Style="{StaticResource TblBase}" Grid.Row="15" HorizontalAlignment="Right">Номер дома</TextBlock>
124 <TextBox x:Name="TbHouseNumber" Grid.Column="1" Grid.Row="15" Text="{Binding HouseNumber}"></TextBox>
125
126 <TextBlock Style="{StaticResource TblBase}" Grid.Row="16" HorizontalAlignment="Right">Копыс</TextBlock>
127 <TextBox x:Name="TbBuildingNumber" Grid.Column="1" Grid.Row="16" Text="{Binding BuildingNumber}"></TextBox>
128
129 <TextBlock Style="{StaticResource TblBase}" Grid.Row="17" HorizontalAlignment="Right">Квартира</TextBlock>
130 <TextBox x:Name="TbApartmentNumber" Grid.Column="1" Grid.Row="17" Text="{Binding ApartmentNumber}"></TextBox>
131
132 <TextBlock Style="{StaticResource TblBase}" Grid.Row="18" HorizontalAlignment="Right">Телефон</TextBlock>
133 <TextBox x:Name="TbPhone" Grid.Column="1" Grid.Row="18" Text="{Binding Phone}" MaxLength="14"></TextBox>
134
135 <TextBlock Style="{StaticResource TblBase}" Grid.Row="19" HorizontalAlignment="Right">Социальный статус</TextBlock>
136 <ComboBox x:Name="CmbxSocialStatus" Grid.Column="1" Grid.Row="19" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_SocialStatus}"></ComboBox>
137
138 <TextBlock Style="{StaticResource TblBase}" Grid.Row="20" HorizontalAlignment="Right">Место работы</TextBlock>
139 <TextBox x:Name="TbJob" Grid.Column="1" Grid.Row="20" Text="{Binding Job}"></TextBox>
140
141 <TextBlock Style="{StaticResource TblBase}" Grid.Row="21" HorizontalAlignment="Right">Должность</TextBlock>
142 <TextBox x:Name="TbJobTitle" Grid.Column="1" Grid.Row="21" Text="{Binding JobTitle}"></TextBox>
143
144 <TextBlock Style="{StaticResource TblBase}" Grid.Row="22" HorizontalAlignment="Right">Вид госпитализации</TextBlock>
145 <ComboBox x:Name="CmbxHospitalizationType" Grid.Column="1" Grid.Row="22" SelectedItem="{Binding Spr_HospitalizationType}" IsEditable="True" DisplayMemberPath="Name"></ComboBox>
146
147 <TextBlock Style="{StaticResource TblBase}" Grid.Row="23" HorizontalAlignment="Right">Вид оплаты</TextBlock>
148 <ComboBox x:Name="CmbxPaymentType" Grid.Column="1" Grid.Row="23" SelectedItem="{Binding Spr_PaymentType}" IsEditable="True" DisplayMemberPath="Name"></ComboBox>
149
150 <TextBlock Style="{StaticResource TblBase}" Grid.Row="24" HorizontalAlignment="Right">Серия полиса</TextBlock>
151 <TextBox x:Name="TbPolicySeria" Grid.Column="1" Grid.Row="24" Text="{Binding PolicySeria}" MaxLength="5"></TextBox>
152
153 <TextBlock Style="{StaticResource TblBase}" Grid.Row="25" HorizontalAlignment="Right">Номер полиса</TextBlock>
154 <TextBox x:Name="TbPolicyNumber" Grid.Column="1" Grid.Row="25" Text="{Binding PolicyNumber}" MaxLength="16"></TextBox>
155
156 <TextBlock Style="{StaticResource TblBase}" Grid.Row="26" HorizontalAlignment="Right">СНИЛС</TextBlock>
157 <TextBox x:Name="TbSNILS" Grid.Column="1" Grid.Row="26" Text="{Binding SNILS}" MaxLength="11"></TextBox>
158
159 <TextBlock Style="{StaticResource TblBase}" Grid.Row="27" HorizontalAlignment="Right">Направитель</TextBlock>
160 <ComboBox x:Name="CmbxPU" Grid.Column="1" Grid.Row="27" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_SendingOrganisation}"></ComboBox>
161
162 <TextBlock Style="{StaticResource TblBase}" Grid.Row="28" HorizontalAlignment="Right">Дата направления</TextBlock>
163 <DatePicker x:Name="DtptkSendingDate" Grid.Column="1" Grid.Row="28" SelectedDate="{Binding SendingDate}"></DatePicker>
164
165 <TextBlock Style="{StaticResource TblBase}" Grid.Row="29" HorizontalAlignment="Right">Кем доставлен</TextBlock>
166 <ComboBox x:Name="CmbxDeliveryType" Grid.Column="1" Grid.Row="29" IsEditable="True" SelectedItem="{Binding Spr_DeliveryType}" DisplayMemberPath="Name"></ComboBox>
167
168 <TextBlock Style="{StaticResource TblBase}" Grid.Row="30" HorizontalAlignment="Right">Номер бригады</TextBlock>
169 <TextBox x:Name="TbBrigadeNumber" Grid.Column="1" Grid.Row="30" Text="{Binding BrigadeNumber}"></TextBox>
170
171 <TextBlock Style="{StaticResource TblBase}" Grid.Row="31" HorizontalAlignment="Right">Диагноз направителя</TextBlock>
172 <ComboBox x:Name="CmbxDiagnosisPU" Grid.Column="1" Grid.Row="31" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_Diagnosis2}"></ComboBox>
173
174 <TextBlock Style="{StaticResource TblBase}" Grid.Row="32" HorizontalAlignment="Right">Описание диагноза направителя</TextBlock>
175 <TextBox x:Name="TbDiagnosisDescriptionPU" Grid.Column="1" Grid.Row="32" Text="{Binding DiagnosisDescription_LPU}"></TextBox>
176
177 <TextBlock Style="{StaticResource TblBase}" Grid.Row="33" HorizontalAlignment="Right">Диагноз приёмного отделения</TextBlock>
178 <ComboBox x:Name="CmbxDiagnosisPO" Grid.Column="1" Grid.Row="33" IsEditable="True" SelectedItem="{Binding Spr_Diagnosis1}" DisplayMemberPath="Name"></ComboBox>
179
180 <TextBlock Style="{StaticResource TblBase}" Grid.Row="34" HorizontalAlignment="Right">Описание диагноза приёмного отделения</TextBlock>

```

Рисунок Б.9 – Код разметки страницы «Регистрация пациента» (2)

```

181 <TextBox x:Name="TbDiagnosisDescriptionPO" Grid.Column="1" Grid.Row="34" Text="{Binding DiagnosisDescription_PO}"></TextBox>
182
183 <TextBlock Style="{StaticResource TblBase}" Grid.Row="35" HorizontalAlignment="Right">Повторная госпитализация</TextBlock>
184 <CheckBox x:Name="ChbxIsRepeated" Grid.Column="1" Grid.Row="35"></CheckBox>
185
186 <TextBlock Style="{StaticResource TblBase}" Grid.Row="36" HorizontalAlignment="Right">Вид травмы</TextBlock>
187 <ComboBox x:Name="CmbxInjuryType" Grid.Column="1" Grid.Row="36" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_InjuryType}"></ComboBox>
188
189 <TextBlock Style="{StaticResource TblBase}" Grid.Row="37" HorizontalAlignment="Right">Период доставки</TextBlock>
190 <ComboBox x:Name="CmbxDeliveryPeriod" Grid.Column="1" Grid.Row="37" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_DeliveryPeriod}"></ComboBox>
191
192 <TextBlock Style="{StaticResource TblBase}" Grid.Row="38" HorizontalAlignment="Right">Дата и время выписки</TextBlock>
193 <xctk:DateTimePicker x:Name="DtptkDateTimeDischarge" Grid.Column="1" Grid.Row="38" Value="{Binding DateTimeDischarge}"></xctk:DateTimePicker>
194
195 <TextBlock Style="{StaticResource TblBase}" Grid.Row="39" HorizontalAlignment="Right">Признак выписки</TextBlock>
196 <ComboBox x:Name="CmbxSignOfDischarge" Grid.Column="1" Grid.Row="39" DisplayMemberPath="Name" SelectedItem="{Binding Spr_SignOfDischarge}"></ComboBox>
197
198 <TextBlock Style="{StaticResource TblBase}" Grid.Row="40" HorizontalAlignment="Right">Диагноз выписки</TextBlock>
199 <ComboBox x:Name="CmbxDiagnosisDischarge" Grid.Column="1" Grid.Row="40" DisplayMemberPath="Name" IsEditable="True" SelectedItem="{Binding Spr_Diagnosis1}"></ComboBox>
200
201 <TextBlock Style="{StaticResource TblBase}" Grid.Row="41" HorizontalAlignment="Right">Исход лечения</TextBlock>
202 <ComboBox x:Name="CmbxTreatmentOutcome" Grid.Column="1" Grid.Row="41" DisplayMemberPath="Name" SelectedItem="{Binding Spr_TreatmentOutcome}"></ComboBox>
203
204 <TextBlock Style="{StaticResource TblBase}" Grid.Row="42" HorizontalAlignment="Right">Причина отказа</TextBlock>
205 <ComboBox x:Name="CmbxDenial" Grid.Column="1" Grid.Row="42" DisplayMemberPath="Name" SelectedItem="{Binding Spr_Denial}"></ComboBox>
206
207 <TextBlock Style="{StaticResource TblBase}" Grid.Row="43" HorizontalAlignment="Right">Отделение поступления</TextBlock>
208 <ComboBox x:Name="CmbxDepartmentRecient" Grid.Column="1" Grid.Row="43" DisplayMemberPath="Name" SelectedItem="{Binding Spr_Department1}"></ComboBox>
209
210 <TextBlock Style="{StaticResource TblBase}" Grid.Row="44" HorizontalAlignment="Right">Отделение выписки</TextBlock>
211 <ComboBox x:Name="CmbxDepartmentDischarge" Grid.Column="1" Grid.Row="44" DisplayMemberPath="Name" SelectedItem="{Binding Spr_Department}"></ComboBox>
212
213
214 </Grid>
215
216 <Grid>
217 <Button x:Name="BtnSave" Style="{StaticResource BtnBlack}" Click="BtnSave_Click" HorizontalAlignment="Left">Сохранить</Button>
218 <Button x:Name="BtnReset" Style="{StaticResource BtnWhite}" Click="BtnReset_Click" HorizontalAlignment="Right">Очистить поля</Button>
219 </Grid>
220
221 </StackPanel>
222 </ScrollViewer>
223 </Page>

```

Рисунок Б.10 – Код разметки страницы «Регистрация пациента» (3)



```

55      /// <summary>
56      /// Обработчик нажатия кнопки "Найти пациента". Открывает страницу со списком пациентов.
57      /// </summary>
58      /// <param name="sender"></param>
59      /// <param name="e"></param>
60      //Ссылка: 1
61      private void BtnFindPatient_Click(object sender, RoutedEventArgs e)
62      {
63          LibUIManager.FrMain.Navigate(new PgPatients());
64      }

```

Рисунок Б.11 – Код обработчика нажатия кнопки «Найти пациента» на странице «Регистрация пациента»

```

70      private void BtnSave_Click(object sender, RoutedEventArgs e)
71      {
72          if (ChbxIsRepeated.IsChecked == true)
73              card.IsRepeated = 1;
74          else
75              card.IsRepeated = 0;
76          if (RbtnMale.IsChecked == true)
77              card.GenderCode = "М";
78          else
79              card.GenderCode = "Ж";
80          card.DateTimeReceipt = DateTime.Now;
81
82          if (LibMedCard.AddCard(card))
83              System.Windows.MessageBox.Show("Медкарта добавлена!", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);
84          else
85              System.Windows.MessageBox.Show("Не удалось добавить медкарту!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
86      }
87

```

Рисунок Б.12 – Код обработчика нажатия кнопки «Сохранить» на странице «Регистрация пациента»

```

88 private void BtnReset_Click(object sender, RoutedEventArgs e)
89 {
90     resetFields();
91 }
92
93 private void resetFields()
94 {
95     TblName.Text = string.Empty;
96     TblFName.Text = string.Empty;
97     TblMName.Text = string.Empty;
98     TblDocumentSeria.Text = string.Empty;
99     TblDocumentNumber.Text = string.Empty;
100     TblRegion.Text = string.Empty;
101     TblArea.Text = string.Empty;
102     TblCity.Text = string.Empty;
103     TblLocality.Text = string.Empty;
104     TblStreet.Text = string.Empty;
105     TblHouseNumber.Text = string.Empty;
106     TblBuildingNumber.Text = string.Empty;
107     TblApartmentNumber.Text = string.Empty;
108     TblPhone.Text = string.Empty;
109     TblJob.Text = string.Empty;
110     TblJobTitle.Text = string.Empty;
111     TblPolicySeria.Text = string.Empty;
112     TblPolicyNumber.Text = string.Empty;
113     TblSNILS.Text = string.Empty;
114     TblBrigadeNumber.Text = string.Empty;
115     TblDiagnosisDescriptionLPU.Text = string.Empty;
116     TblDiagnosisDescriptionPO.Text = string.Empty;
117     TblBrigadeNumber.Text = string.Empty;
118
119     DtpkBirthday.SelectedDate = null;
120     DtpkDateTimeDischarge.Text = string.Empty;
121
122     CmbxDocumentType.SelectedItem = null;
123     CmbxSignOfResidence.SelectedItem = null;
124     CmbxCountry.SelectedItem = null;
125     CmbxSocialStatus.SelectedItem = null;
126     CmbxHospitalizationType.SelectedItem = null;
127     CmbxPaymentType.SelectedItem = null;
128     CmbxLPU.SelectedItem = null;
129     CmbxDeliveryType.SelectedItem = null;
130     CmbxDiagnosisLPU.SelectedItem = null;
131     CmbxDiagnosisPO.SelectedItem = null;
132     CmbxInjuryType.SelectedItem = null;
133     CmbxDeliveryPeriod.SelectedItem = null;
134     CmbxSignOfDischarge.SelectedItem = null;
135     CmbxDiagnosisDischarge.SelectedItem = null;
136     CmbxTreatmentOutcome.SelectedItem = null;
137     CmbxDenial.SelectedItem = null;
138 }

```

Рисунок Б.13 – Код обработчика нажатия кнопки «Очистить поля» на странице «Регистрация пациента»

```

1 <Page x:Class="MedPlus.UI.Pg.PgPatients">
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:ix="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6     xmlns:local="clr-namespace:MedPlus.UI.Pg"
7     mc:Ignorable="d"
8     d:DesignHeight="450" d:DesignWidth="800"
9     Title="PgPatients"
10     Loaded="Page_Loaded">
11
12
13     <Grid>
14         <Grid.RowDefinitions>
15             <RowDefinition Height="100"></RowDefinition>
16             <RowDefinition/>
17             <RowDefinition Height="100"></RowDefinition>
18         </Grid.RowDefinitions>
19         <StackPanel Orientation="Horizontal">
20             <TextBlock Style="{StaticResource TblBase}">Поиск по имени:</TextBlock>
21             <TextBox x:Name="TblSearch" TextChanged="TblSearch_TextChanged"/></TextBlock>
22         </StackPanel>
23         <Button x:Name="BtnUpdate" Style="{StaticResource BtnBlack}" Width="200" Click="BtnUpdate_Click" HorizontalAlignment="Right" Visibility="Visible" Margin="10">Обновить</Button>
24         <ScrollViewer Grid.Row="1">
25             <StackPanel Orientation="Vertical" x:Name="SpContainer" ScrollViewer.VerticalScrollBarVisibility="Visible">
26
27             </StackPanel>
28         </ScrollViewer>
29         <Button x:Name="BtnNextRecords" Grid.Row="2" Style="{StaticResource BtnWhite}" Width="200" HorizontalAlignment="Right" Click="BtnNextRecords_Click">Следующие 200 записей</Button>
30         <Button x:Name="BtnPreviousRecords" Grid.Row="2" Style="{StaticResource BtnWhite}" Width="200" HorizontalAlignment="Left" Click="BtnPreviousRecords_Click" Visibility="Collapsed">Предыдущие 200 записей</Button>
31     </Grid>
32 </Page>
33

```

Рисунок Б.14 – Код разметки страницы «Список пациентов»

```

53  /// <summary>
54  /// Обработчик нажатия кнопки "Обновить". Иницирует процесс обновления записей.
55  /// </summary>
56  /// <param name="sender"></param>
57  /// <param name="e"></param>
58  ссылка: 1
59  private void BtnUpdate_Click(object sender, RoutedEventArgs e)
60  {
61      currentPage = 0;
62      BtnPreviousRecords.Visibility = Visibility.Collapsed;
63      BtnNextRecords.Visibility = Visibility.Visible;
64      updateCards();
65      downloadCards();
66  }
67  /// <summary>
68  /// Метод. Обновляет список пациентов.
69  /// </summary>
70  ссылка: 5
71  private void updateCards()
72  {
73      cards = LibMedCard.GetMedCards(pageSize * currentPage, TbSearch.Text);
74      int rowCount = LibMedCard.GetCountOfCards(TbSearch.Text);
75      if (rowCount % pageSize == 0)
76          pageCount = rowCount / pageSize;
77      else
78          pageCount = rowCount / pageSize + 1;
79  }

```

Рисунок Б.15 – Код обработчика кнопки «Обновить» на странице «Список пациентов»

```

80  /// <summary>
81  /// Обработчик изменения текста в поле поиска.
82  /// </summary>
83  /// <param name="sender"></param>
84  /// <param name="e"></param>
85  ссылка: 1
86  private void TbSearch_TextChanged(object sender, TextChangedEventArgs e)
87  {
88      currentPage = 0;
89      BtnPreviousRecords.Visibility = Visibility.Collapsed;
90      if (pageCount > 1)
91          BtnNextRecords.Visibility = Visibility.Visible;
92      else
93          BtnNextRecords.Visibility = Visibility.Visible;
94      updateCards();
95      downloadCards();
96  }

```

Рисунок Б.16 – Код обработчика текстового поля поиска на странице «Список пациентов»

```

1  <Page x:Class="MedPlus.UI.Pg.EditCard"
2  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6  xmlns:xc="http://schemas.xceed.com/wpf/xaml/toolkit"
7  xmlns:local="clr-namespace:MedPlus.UI.Pg"
8  mc:Ignorable="d"
9  d:DesignHeight="450" d:DesignWidth="800"
10 Title="PgEditCard">
11
12     <Grid>
13         <Grid.RowDefinitions>
14             <RowDefinition Height="100"/>
15             <RowDefinition Height="100"/>
16             <RowDefinition Height="100"/>
17         </Grid.RowDefinitions>
18
19         <Grid>
20             <Button x:Name="BtnBack" Style="{StaticResource BtnWhite}" Width="300" HorizontalAlignment="Left" Click="BtnCancel_Click">Назад</Button>
21             <Button x:Name="BtnOpenMovements" Style="{StaticResource BtnWhite}" Width="300" HorizontalAlignment="Right" Click="BtnOpenMovements_Click">Движения пациента</Button>
22         </Grid>
23
24     <ScrollViewer Grid.Row="1">
25         <Grid>
26             <Grid.RowDefinitions>
27                 <RowDefinition Height="70"/>
28                 <RowDefinition Height="70"/>
29                 <RowDefinition Height="70"/>
30                 <RowDefinition Height="70"/>
31                 <RowDefinition Height="70"/>
32                 <RowDefinition Height="70"/>
33                 <RowDefinition Height="70"/>
34                 <RowDefinition Height="70"/>
35                 <RowDefinition Height="70"/>
36                 <RowDefinition Height="70"/>
37                 <RowDefinition Height="70"/>
38                 <RowDefinition Height="70"/>
39                 <RowDefinition Height="70"/>
40                 <RowDefinition Height="70"/>
41                 <RowDefinition Height="70"/>
42                 <RowDefinition Height="70"/>
43                 <RowDefinition Height="70"/>
44                 <RowDefinition Height="70"/>
45                 <RowDefinition Height="70"/>
46                 <RowDefinition Height="70"/>
47                 <RowDefinition Height="70"/>
48                 <RowDefinition Height="70"/>
49                 <RowDefinition Height="70"/>
50                 <RowDefinition Height="70"/>
51                 <RowDefinition Height="70"/>
52                 <RowDefinition Height="70"/>
53                 <RowDefinition Height="70"/>
54                 <RowDefinition Height="70"/>
55                 <RowDefinition Height="70"/>
56                 <RowDefinition Height="70"/>
57                 <RowDefinition Height="70"/>
58                 <RowDefinition Height="70"/>
59                 <RowDefinition Height="70"/>
60                 <RowDefinition Height="70"/>

```

Рисунок Б.17 – Код разметки страницы «Редактирование карты пациента» (1)

```

61      <RowDefinition Height="70"/>
62      <RowDefinition Height="70"/>
63      <RowDefinition Height="70"/>
64      <RowDefinition Height="70"/>
65      <RowDefinition Height="70"/>
66      <RowDefinition Height="70"/>
67      <RowDefinition Height="70"/>
68      <RowDefinition Height="70"/>
69      <RowDefinition Height="70"/>
70      <RowDefinition Height="70"/>
71      <RowDefinition Height="70"/>
72    </Grid.RowDefinitions>
73    <Grid.ColumnDefinitions>
74      <ColumnDefinition Width="350"/></ColumnDefinition>
75      <ColumnDefinition/></ColumnDefinition>
76    </Grid.ColumnDefinitions>
77
78    <TextBlock Style="{StaticResource TblBase}" Grid.Row="0" HorizontalAlignment="Right">Фамилия</TextBlock>
79    <TextBox x:Name="TbName" Grid.Column="1" Grid.Row="0" Text="{Binding LName}" IsEnabled="False"/></TextBox>
80
81    <TextBlock Style="{StaticResource TblBase}" Grid.Row="1" HorizontalAlignment="Right">Имя</TextBlock>
82    <TextBox x:Name="TbFName" Grid.Column="1" Grid.Row="1" Text="{Binding FName}" IsEnabled="False"/></TextBox>
83
84    <TextBlock Style="{StaticResource TblBase}" Grid.Row="2" HorizontalAlignment="Right">Отчество</TextBlock>
85    <TextBox x:Name="TbMName" Grid.Column="1" Grid.Row="2" Text="{Binding MName}" IsEnabled="False"/></TextBox>
86
87    <TextBlock Style="{StaticResource TblBase}" Grid.Row="3" HorizontalAlignment="Right">Пол</TextBlock>
88    <StackPanel Orientation="Horizontal" Grid.Column="1" Grid.Row="3">
89      <RadioButton x:Name="RbtnMale" IsChecked="True" IsEnabled="False">Мужской</RadioButton>
90      <RadioButton x:Name="RbtnFemale" IsEnabled="False">Женский</RadioButton>
91    </StackPanel>
92
93    <TextBlock Style="{StaticResource TblBase}" Grid.Row="4" HorizontalAlignment="Right">Тип документа</TextBlock>
94    <ComboBox x:Name="CmbxDocumentType" Grid.Column="1" Grid.Row="4" IsEnabled="False" SelectedItem="{Binding Spr_DocumentType}" DisplayMemberPath="Name"/></ComboBox>
95
96    <TextBlock Style="{StaticResource TblBase}" Grid.Row="5" HorizontalAlignment="Right">Серия документа</TextBlock>
97    <TextBox x:Name="TbDocumentSerial" Grid.Column="1" Grid.Row="5" IsEnabled="False" Text="{Binding DocumentSerial}"></TextBox>
98
99    <TextBlock Style="{StaticResource TblBase}" Grid.Row="6" HorizontalAlignment="Right">Номер документа</TextBlock>
100    <ComboBox x:Name="TbDocumentNumber" Grid.Column="1" Grid.Row="6" IsEnabled="False" Text="{Binding DocumentNumber}"></ComboBox>
101
102    <TextBlock Style="{StaticResource TblBase}" Grid.Row="7" HorizontalAlignment="Right">Дата рождения</TextBlock>
103    <DatePicker x:Name="DtpkBirthday" Grid.Column="1" Grid.Row="7" IsEnabled="False" SelectedDate="{Binding Birthday}"></DatePicker>
104
105    <TextBlock Style="{StaticResource TblBase}" Grid.Row="8" HorizontalAlignment="Right">Признак жительства</TextBlock>
106    <ComboBox x:Name="CmbxSignOfResidence" Grid.Column="1" Grid.Row="8" IsEnabled="False" SelectedItem="{Binding spr_SignOfResidence}" DisplayMemberPath="Name"/></ComboBox>
107
108    <TextBlock Style="{StaticResource TblBase}" Grid.Row="9" HorizontalAlignment="Right">Страна</TextBlock>
109    <ComboBox x:Name="CmbxCountry" Grid.Column="1" Grid.Row="9" IsEnabled="False" SelectedItem="{Binding Spr_Country}" IsEditable="True" DisplayMemberPath="Name"/></ComboBox>
110
111    <TextBlock Style="{StaticResource TblBase}" Grid.Row="10" HorizontalAlignment="Right">Период</TextBlock>
112    <TextBlock x:Name="TbRegion" Grid.Column="1" Grid.Row="10" IsEnabled="False"/></TextBlock>
113
114    <TextBlock Style="{StaticResource TblBase}" Grid.Row="11" HorizontalAlignment="Right">Район</TextBlock>
115    <TextBlock x:Name="TbArea" Grid.Column="1" Grid.Row="11" IsEnabled="False"/></TextBlock>
116
117    <TextBlock Style="{StaticResource TblBase}" Grid.Row="12" HorizontalAlignment="Right">Город</TextBlock>
118    <TextBlock x:Name="TbCity" Grid.Column="1" Grid.Row="12" IsEnabled="False"/></TextBlock>
119
120    <TextBlock Style="{StaticResource TblBase}" Grid.Row="13" HorizontalAlignment="Right">Населённый пункт</TextBlock>

```

Рисунок Б.18 – Код разметки страницы «Редактирование карты пациента» (2)

```

121    <TextBlock x:Name="TbLocality" Grid.Column="1" Grid.Row="13" IsEnabled="False"/></TextBlock>
122
123    <TextBlock Style="{StaticResource TblBase}" Grid.Row="14" HorizontalAlignment="Right">Улица</TextBlock>
124    <TextBlock x:Name="TbStreet" Grid.Column="1" Grid.Row="14" Text="{Binding Street}" IsEnabled="False"/></TextBlock>
125
126    <TextBlock Style="{StaticResource TblBase}" Grid.Row="15" HorizontalAlignment="Right">Номер дома</TextBlock>
127    <TextBlock x:Name="TbHouseNumber" Grid.Column="1" Grid.Row="15" Text="{Binding HouseNumber}" IsEnabled="False"/></TextBlock>
128
129    <TextBlock Style="{StaticResource TblBase}" Grid.Row="16" HorizontalAlignment="Right">Копыс</TextBlock>
130    <TextBlock x:Name="TbBuildingNumber" Grid.Column="1" Grid.Row="16" Text="{Binding BuildingNumber}" IsEnabled="False"/></TextBlock>
131
132    <TextBlock Style="{StaticResource TblBase}" Grid.Row="17" HorizontalAlignment="Right">Квартира</TextBlock>
133    <TextBlock x:Name="TbApartmentNumber" Grid.Column="1" Grid.Row="17" Text="{Binding ApartmentNumber}" IsEnabled="False"/></TextBlock>
134
135    <TextBlock Style="{StaticResource TblBase}" Grid.Row="18" HorizontalAlignment="Right">Телефон</TextBlock>
136    <TextBlock x:Name="TbPhone" Grid.Column="1" Grid.Row="18" Text="{Binding Phone}" MaxLength="14" IsEnabled="False"/></TextBlock>
137
138    <TextBlock Style="{StaticResource TblBase}" Grid.Row="19" HorizontalAlignment="Right">Социальный статус</TextBlock>
139    <ComboBox x:Name="CmbxSocialStatus" Grid.Column="1" Grid.Row="19" IsEnabled="False" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_SocialStatus}"></ComboBox>
140
141    <TextBlock Style="{StaticResource TblBase}" Grid.Row="20" HorizontalAlignment="Right">Место работы</TextBlock>
142    <TextBlock x:Name="TbJob" Grid.Column="1" Grid.Row="20" Text="{Binding Job}" IsEnabled="False"/></TextBlock>
143
144    <TextBlock Style="{StaticResource TblBase}" Grid.Row="21" HorizontalAlignment="Right">Должность</TextBlock>
145    <TextBlock x:Name="TbJobTitle" Grid.Column="1" Grid.Row="21" Text="{Binding JobTitle}" IsEnabled="False"/></TextBlock>
146
147    <TextBlock Style="{StaticResource TblBase}" Grid.Row="22" HorizontalAlignment="Right">Вид госпитализации</TextBlock>
148    <ComboBox x:Name="CmbxHospitalizationType" Grid.Column="1" Grid.Row="22" IsEnabled="False" SelectedItem="{Binding Spr_HospitalizationType}" DisplayMemberPath="Name"/></ComboBox>
149
150    <TextBlock Style="{StaticResource TblBase}" Grid.Row="23" HorizontalAlignment="Right">Вид оплаты</TextBlock>
151    <ComboBox x:Name="CmbxPaymentType" Grid.Column="1" Grid.Row="23" IsEnabled="False" SelectedItem="{Binding Spr_PaymentType}" IsEditable="True" DisplayMemberPath="Name"/></ComboBox>
152
153    <TextBlock Style="{StaticResource TblBase}" Grid.Row="24" HorizontalAlignment="Right">Серия полиса</TextBlock>
154    <TextBlock x:Name="TbPolicySerial" Grid.Column="1" Grid.Row="24" IsEnabled="False" Text="{Binding PolicySerial}" MaxLength="5"/></TextBlock>
155
156    <TextBlock Style="{StaticResource TblBase}" Grid.Row="25" HorizontalAlignment="Right">Номер полиса</TextBlock>
157    <TextBlock x:Name="TbPolicyNumber" Grid.Column="1" Grid.Row="25" IsEnabled="False" Text="{Binding PolicyNumber}" MaxLength="16"/></TextBlock>
158
159    <TextBlock Style="{StaticResource TblBase}" Grid.Row="26" HorizontalAlignment="Right">ОЧИС</TextBlock>
160    <TextBlock x:Name="TbSNILS" Grid.Column="1" Grid.Row="26" IsEnabled="False" Text="{Binding SNILS}" MaxLength="11"/></TextBlock>
161
162    <TextBlock Style="{StaticResource TblBase}" Grid.Row="27" HorizontalAlignment="Right">Направитель</TextBlock>
163    <ComboBox x:Name="CmbxPU" Grid.Column="1" Grid.Row="27" IsEnabled="False" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_SendingOrganisation}"></ComboBox>
164
165    <TextBlock Style="{StaticResource TblBase}" Grid.Row="28" HorizontalAlignment="Right">Дата направления</TextBlock>
166    <DatePicker x:Name="DtpkSendingDate" Grid.Column="1" Grid.Row="28" SelectedDate="{Binding SendingDate}"></DatePicker>
167
168    <TextBlock Style="{StaticResource TblBase}" Grid.Row="29" HorizontalAlignment="Right">Как доставлен</TextBlock>
169    <ComboBox x:Name="CmbxDeliveryType" Grid.Column="1" Grid.Row="29" IsEnabled="False" IsEditable="True" SelectedItem="{Binding Spr_DeliveryType}" DisplayMemberPath="Name"/></ComboBox>
170
171    <TextBlock Style="{StaticResource TblBase}" Grid.Row="30" HorizontalAlignment="Right">Номер бригады</TextBlock>
172    <TextBlock x:Name="TbBrigadeNumber" Grid.Column="1" Grid.Row="30" IsEnabled="False" Text="{Binding BrigadeNumber}"></TextBlock>
173
174    <TextBlock Style="{StaticResource TblBase}" Grid.Row="31" HorizontalAlignment="Right">Диагноз направителя</TextBlock>
175    <ComboBox x:Name="CmbxDiagnosisPU" Grid.Column="1" Grid.Row="31" IsEnabled="False" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_Diagnosis2}"></ComboBox>
176
177    <TextBlock Style="{StaticResource TblBase}" Grid.Row="32" HorizontalAlignment="Right">Описание диагноза направителя</TextBlock>
178    <TextBlock x:Name="TbDiagnosisDescriptionPU" Grid.Column="1" Grid.Row="32" IsEnabled="False" Text="{Binding DiagnosisDescription_LPU}"></TextBlock>
179
180    <TextBlock Style="{StaticResource TblBase}" Grid.Row="33" HorizontalAlignment="Right">Диагноз приёмного отделения</TextBlock>

```

Рисунок Б.19 – Код разметки страницы «Редактирование карты пациента» (3)

```

181 <ComboBox x:Name="cmbDiagnosisPO" Grid.Column="1" Grid.Row="33" IsEnabled="False" IsEditable="True" SelectedItem="{Binding Spr_Diagnosis}" DisplayMemberPath="Name"></ComboBox>
182
183 <TextBlock Style="{StaticResource TbBase}" Grid.Row="34" HorizontalAlignment="Right">Описание диагноза приёмного отделения</TextBlock>
184 <TextBox x:Name="TbDiagnosisDescriptionPO" Grid.Column="1" Grid.Row="34" IsEnabled="False" Text="{Binding DiagnosisDescription_PO}"></TextBox>
185
186 <TextBlock Style="{StaticResource TbBase}" Grid.Row="35" HorizontalAlignment="Right">Повторная госпитализация</TextBlock>
187 <CheckBox x:Name="ChbxIsRepeated" Grid.Column="1" Grid.Row="35" IsEnabled="False"></CheckBox>
188
189 <TextBlock Style="{StaticResource TbBase}" Grid.Row="36" HorizontalAlignment="Right">Вид травмы</TextBlock>
190 <ComboBox x:Name="cmbInjuryType" Grid.Column="1" Grid.Row="36" IsEnabled="False" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_InjuryType}"></ComboBox>
191
192 <TextBlock Style="{StaticResource TbBase}" Grid.Row="37" HorizontalAlignment="Right">Период доставки</TextBlock>
193 <ComboBox x:Name="cmbDeliveryPeriod" Grid.Column="1" Grid.Row="37" IsEnabled="False" IsEditable="True" DisplayMemberPath="Name" SelectedItem="{Binding Spr_DeliveryPeriod}"></ComboBox>
194
195 <TextBlock Style="{StaticResource TbBase}" Grid.Row="38" HorizontalAlignment="Right">Дата и время выписки</TextBlock>
196 <xctk:DateTimePicker x:Name="DtpkDateTimeDischarge" Grid.Column="1" Grid.Row="38" IsEnabled="False" Grid.Row="38" Text="{Binding DateTimeDischarge}"></xctk:DateTimePicker>
197
198 <TextBlock Style="{StaticResource TbBase}" Grid.Row="39" HorizontalAlignment="Right">Признак выписки</TextBlock>
199 <ComboBox x:Name="cmbSignOfDischarge" Grid.Column="1" Grid.Row="39" IsEnabled="False" DisplayMemberPath="Name" SelectedItem="{Binding Spr_SignOfDischarge}"></ComboBox>
200
201 <TextBlock Style="{StaticResource TbBase}" Grid.Row="40" HorizontalAlignment="Right">Диагноз выписки</TextBlock>
202 <ComboBox x:Name="cmbDiagnosisDischarge" Grid.Column="1" Grid.Row="40" IsEnabled="False" DisplayMemberPath="Name" IsEditable="True" SelectedItem="{Binding Spr_DiagnosisI}"></ComboBox>
203
204 <TextBlock Style="{StaticResource TbBase}" Grid.Row="41" HorizontalAlignment="Right">Исход лечения</TextBlock>
205 <ComboBox x:Name="cmbTreatmentOutcome" Grid.Column="1" Grid.Row="41" IsEnabled="False" DisplayMemberPath="Name" SelectedItem="{Binding Spr_TreatmentOutcome}"></ComboBox>
206
207 <TextBlock Style="{StaticResource TbBase}" Grid.Row="42" HorizontalAlignment="Right">Причина отказа</TextBlock>
208 <ComboBox x:Name="cmbDenial" Grid.Column="1" Grid.Row="42" DisplayMemberPath="Name" IsEnabled="False" SelectedItem="{Binding Spr_Denial}"></ComboBox>
209
210 <TextBlock Style="{StaticResource TbBase}" Grid.Row="43" HorizontalAlignment="Right">Отделение поступления</TextBlock>
211 <ComboBox x:Name="cmbDepartmentRecient" Grid.Column="1" Grid.Row="43" DisplayMemberPath="Name" SelectedItem="{Binding Spr_DepartmentI}"></ComboBox>
212
213 <TextBlock Style="{StaticResource TbBase}" Grid.Row="44" HorizontalAlignment="Right">Отделение выписки</TextBlock>
214 <ComboBox x:Name="cmbDepartmentDischarge" Grid.Column="1" Grid.Row="44" DisplayMemberPath="Name" SelectedItem="{Binding Spr_Department}"></ComboBox>
215
216 </Grid>
217 </ScrollViewer>
218
219 <Grid Grid.Row="2">
220 <Button x:Name="BtnSave" Style="{StaticResource BtnBlack}" Click="BtnSave_Click" HorizontalAlignment="Left">Сохранить</Button>
221 <Button x:Name="BtnCancel" Style="{StaticResource BtnWhite}" Click="BtnCancel_Click" HorizontalAlignment="Right">Отмена</Button>
222 </Grid>
223 </Grid>
224 </Page>

```

Рисунок Б.20 – Код разметки страницы «Редактирование карты пациента» (4)

```

175 /// <summary>
176 /// Обработчик нажатия кнопки "Движения пациента". Открывает страницу движений текущего пациента.
177 /// </summary>
178 /// <param name="sender"></param>
179 /// <param name="e"></param>
180 ссылка: 1
181 private void BtnOpenMovements_Click(object sender, RoutedEventArgs e)
182 {
183     LibUIManager.FrMain.Navigate(new PgPatientMovements(card));
184 }
185 }

```

Рисунок Б.21 – Код обработчика нажатия кнопки «Движения пациента» на странице «Редактирование карты пациента»

```

67     /// <summary>
68     /// Обработчик события. Иницирует процесс сохранения изменений в редактируемой карте.
69     /// </summary>
70     /// <param name="sender"></param>
71     /// <param name="e"></param>
72     private void BtnSave_Click(object sender, RoutedEventArgs e)
73     {
74         card.GenderCode = (bool)rbtnFemale.IsChecked ? "Ж"; "М";
75         card.IsRepeated = (bool)chbxIsRepeated.IsChecked ? 1; 0;
76         if (LibMedCard.UpdateMedCard(card))
77         {
78             MessageBox.Show("Данные успешно сохранены!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);
79             if (oldPaymentType != null)
80             {
81                 if (oldPaymentType.ID != card.PaymentTypeID)
82                 {
83                     ChangesOfPaymentType changesOfPaymentType = new ChangesOfPaymentType
84                     {
85                         oldPaymentTypeID = oldPaymentType.ID,
86                         NewPaymentTypeID = (int)card.PaymentTypeID,
87                         DateTime = DateTime.Now,
88                         MedCard = card,
89                         User = LibUser.CurrentUser
90                     };
91
92                     LibDB.GetContext().ChangesOfPaymentTypes.Add(changesOfPaymentType);
93                     LibDB.GetContext().SaveChanges();
94                     MessageBox.Show("Тип оплаты был изменён!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
95                 }
96             }
97             LibUIManager.FrmMain.GoBack();
98         }
99         else
100             MessageBox.Show("Данные не были сохранены!", "Ошибка при обновлении", MessageBoxButton.OK, MessageBoxImage.Error);
101     }
102 }

```

Рисунок Б.22 – Код обработчика нажатия кнопки «Сохранить» на странице «Редактирование карты пациента»

```

1  <Page x:Class="MedPlus.UI.Pg.PatientMovements"
2      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
5      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6      xmlns:local="clr-namespace:MedPlus.UI.Pg"
7      mc:Ignorable="d"
8      d:DesignHeight="450" d:DesignWidth="800"
9      Title="PgPatientMovements">
10
11     <Grid>
12         <Grid.RowDefinitions>
13             <RowDefinition Height="100"/>
14             <RowDefinition/>
15             <RowDefinition Height="100"/>
16         </Grid.RowDefinitions>
17
18         <Button x:Name="BtnBack" Style="{StaticResource BtnWhite}" Width="200" HorizontalAlignment="Left" Click="BtnBack_Click">Назад</Button>
19         <Button x:Name="BtnSaveChanges" Style="{StaticResource BtnBlack}" Width="200" HorizontalAlignment="Right" Click="BtnSaveChanges_Click">Сохранить изменения</Button>
20
21         <Button x:Name="BtnAdd" Style="{StaticResource BtnWhite}" Grid.Row="2" Width="200" HorizontalAlignment="Left" Click="BtnAdd_Click">Добавить</Button>
22         <Button x:Name="BtnDeleteLast" Style="{StaticResource BtnWhite}" Grid.Row="2" Width="200" HorizontalAlignment="Right" Click="BtnDeleteLast_Click">Удалить последнее</Button>
23     </Grid>
24 </Page>
25
26

```

Рисунок Б.23 – Код разметки страницы «Движения пациента»

```

42     /// <summary>
43     /// Обработчик нажатия кнопки "Сохранить изменения". Иницирует процесс сохранения изменений в БД.
44     /// </summary>
45     /// <param name="sender"></param>
46     /// <param name="e"></param>
47     private void BtnSaveChanges_Click(object sender, RoutedEventArgs e)
48     {
49         LibDB.GetContext().PatientMovements.AddRange(added);
50         LibDB.GetContext().PatientMovements.RemoveRange(deleted);
51
52         LibDB.GetContext().SaveChanges();
53
54         deleted.Clear();
55         added.Clear();
56     }

```

Рисунок Б.24 – Код обработчика нажатия кнопки «Сохранить изменения» на странице «Движения пациента»



Рисунок Б.25 – Код обработчика нажатия кнопки «Удалить последнее» на странице «Движения пациента»

Рисунок Б.26 – Код обработчика нажатия кнопки «Добавить» на странице «Движения пациента»

Рисунок Б.27 – Код разметки страницы «Сигнальная ведомость»



```

45  /// <summary>
46  /// Обработчик нажатия кнопки "Сформировать". Получает сигнальную ведомость из базы данных.
47  /// </summary>
48  /// <param name="sender"></param>
49  /// <param name="e"></param>
    ссылка: 1
50  private void BtnGenerate_Click(object sender, RoutedEventArgs e)
51  {
52      DateTime day = (DateTime)DtpkDay.SelectedDate;
53      List<Sp_SignalStatement_Result> report = LibReports.GetSignalStatement(day);
54      DgContainer.ItemsSource = report;
55  }

```

Рисунок Б.28 – Код обработчика нажатия кнопки «Сформировать» на странице «Сигнальная ведомость»

## Приложение В

### Скрипт базы данных

```
USE [MedPlusDB]
GO
/***** Object: Table [dbo].[BenefitsOfPatients]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[BenefitsOfPatients](
    [BenefitID] [int] NOT NULL,
    [CardID] [int] NOT NULL,
    CONSTRAINT [PK_BenefitsOfPatients] PRIMARY KEY CLUSTERED
(
    [BenefitID] ASC,
    [CardID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[ChangesOfPaymentType]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ChangesOfPaymentType](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [OldPaymentTypeID] [int] NOT NULL,
    [NewPaymentTypeID] [int] NOT NULL,
    [UserID] [int] NOT NULL,
    [DateTime] [date] NOT NULL,
    [CardID] [int] NOT NULL,
    CONSTRAINT [PK_ChangesOfPaymentType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Death]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Death](
    [CardID] [int] NOT NULL,
    [TimeDeath] [time](7) NULL,
    [SignOfAutopsy] [bit] NULL,
    [AutopsyTypeID] [int] NULL,
    [DiagnosisAutopsy] [int] NULL,
    CONSTRAINT [PK_Death] PRIMARY KEY CLUSTERED
(
    [CardID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

Рисунок В.1

```

/***** Object: Table [dbo].[MedCard]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MedCard](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [DateTimeReceipt] [datetime] NULL,
    [DepartmentID_Receipt] [int] NULL,
    [DepartmentID_Discharge] [int] NULL,
    [DocumentTypeID] [int] NULL,
    [DocumentSerial] [nvarchar](100) NULL,
    [DocumentNumber] [nvarchar](100) NULL,
    [LName] [nvarchar](500) NULL,
    [FName] [nvarchar](500) NULL,
    [MName] [nvarchar](500) NULL,
    [GenderCode] [nchar](1) NULL,
    [Birthday] [date] NULL,
    [SignOfResidenceID] [int] NULL,
    [CountryID] [int] NULL,
    [KLADR_ID] [nchar](11) NULL,
    [Street] [nvarchar](500) NULL,
    [HouseNumber] [nvarchar](50) NULL,
    [BuildingNumber] [nvarchar](50) NULL,
    [ApartmentNumber] [nvarchar](50) NULL,
    [Phone] [nchar](14) NULL,
    [SocialStatusID] [int] NULL,
    [Job] [nvarchar](1000) NULL,
    [JobTitle] [nvarchar](1000) NULL,
    [PaymentTypeID] [int] NULL,
    [PolicySerial] [nvarchar](5) NULL,
    [PolicyNumber] [nvarchar](16) NULL,
    [SNILS] [nchar](11) NULL,
    [SendingOrganisationID] [nvarchar](6) NULL,
    [SendingDate] [date] NULL,
    [HospitalizationTypeID] [int] NOT NULL,
    [DeliveryTypeID] [int] NULL,
    [DiagnosisCode_LPU] [nvarchar](6) NULL,
    [DiagnosisDescription_LPU] [nvarchar](1000) NULL,
    [DiagnosisCode_PO] [nvarchar](6) NULL,
    [DiagnosisDescription_PO] [nvarchar](1000) NULL,
    [IsRepeated] [int] NULL,
    [InjuryTypeID] [int] NULL,
    [DeliveryPeriodID] [int] NULL,
    [DateTimeDischarge] [datetime] NULL,
    [SignOfDischargeID] [int] NULL,
    [DiagnosisCode_Discharge] [nvarchar](6) NULL,
    [TreatmentOutcomeID] [int] NULL,
    [DenialID] [int] NULL,
    [BrigadeNumber] [nvarchar](100) NULL,
    CONSTRAINT [PK_MedCard] PRIMARY KEY CLUSTERED
(

```

Рисунок В.2

```

[ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[PatientMovement]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PatientMovement](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [SignOfDischargeID] [int] NOT NULL,
    [DepartmentID] [int] NOT NULL,
    [DateTimeReceipt] [datetime] NOT NULL,
    [DateTimeDischarge] [datetime] NULL,
    [CardID] [int] NOT NULL,
    CONSTRAINT [PK_PatientMovement] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_AutopsyType]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_AutopsyType](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](400) NOT NULL,
    CONSTRAINT [PK_AutopsyType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Benefit]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Benefit](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_Benefit] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Country]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON

```

Рисунок В.3

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Country](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_Country] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_DeliveryPeriod]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_DeliveryPeriod](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](150) NOT NULL,
    CONSTRAINT [PK_Spr_DeliveryPeriod] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_DeliveryType]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_DeliveryType](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](400) NOT NULL,
    CONSTRAINT [PK_Spr_DeliveryType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Denial]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Denial](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_Denial] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

Рисунок В.4

```

GO
ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Department]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Department](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](500) NOT NULL,
    [Beds] [int] NULL,
    [SerialNumber] [int] NULL,
    [IsDayDepartment] [int] NULL,
    CONSTRAINT [PK_Department] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Diagnosis]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Diagnosis](
    [Code] [nvarchar](6) NOT NULL,
    [Name] [nvarchar](1000) NOT NULL,
    CONSTRAINT [PK_Spr_Diagnosis] PRIMARY KEY CLUSTERED
(
    [Code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_DocumentType]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_DocumentType](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_DocumentType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Gender]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

Рисунок В.5

```

CREATE TABLE [dbo].[Spr_Gender](
    [Code] [nchar](1) NOT NULL,
    [Name] [nvarchar](10) NOT NULL,
    CONSTRAINT [PK_Spr_Gender_1] PRIMARY KEY CLUSTERED
(
    [Code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_HospitalizationType]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_HospitalizationType](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_HospitalizationType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_InjuryType]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_InjuryType](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_InjuryType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_KLADR]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_KLADR](
    [Code] [nchar](11) NOT NULL,
    [Name] [nvarchar](200) NOT NULL,
    [Abbreviation] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_KLADR_1] PRIMARY KEY CLUSTERED
(
    [Code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

```

Рисунок В.6

```

/***** Object: Table [dbo].[Spr_PaymentType]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_PaymentType](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_PaymentType] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Role]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Role](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Role] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_SendingOrganisation]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_SendingOrganisation](
    [Code] [nvarchar](6) NOT NULL,
    [Name] [nvarchar](1000) NOT NULL,
    CONSTRAINT [PK_Spr_SendingOrganisation] PRIMARY KEY CLUSTERED
(
    [Code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_SignOfDischarge]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_SignOfDischarge](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](150) NOT NULL,
    CONSTRAINT [PK_Spr_TreatmentOutcome] PRIMARY KEY CLUSTERED
(

```

Рисунок В.7

```

    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_SignOfResidence]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_SignOfResidence](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_SignOfResidence] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_SocialStatus]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_SocialStatus](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_SocialStatus] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_Street]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Spr_Street](
    [Code] [nvarchar](17) NOT NULL,
    [Name] [nvarchar](400) NOT NULL,
    [Abbreviation] [nvarchar](100) NOT NULL,
    CONSTRAINT [PK_Spr_Street] PRIMARY KEY CLUSTERED
(
    [Code] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Spr_TreatmentOutcome]  Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

Рисунок В.8

```

CREATE TABLE [dbo].[Spr_TreatmentOutcome](
    [ID] [int] NOT NULL,
    [Name] [nvarchar](400) NOT NULL,
    CONSTRAINT [PK_Spr_TreatmentOutcome_1] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[User]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[User](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Login] [nvarchar](50) NOT NULL,
    [Password] [nchar](32) NOT NULL,
    [RoleID] [int] NOT NULL,
    [FName] [nvarchar](100) NOT NULL,
    [UName] [nvarchar](100) NOT NULL,
    [MName] [nvarchar](100) NOT NULL,
    [Birthdate] [date] NOT NULL,
    CONSTRAINT [PK_User] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY],
    CONSTRAINT [UK_User_Login] UNIQUE NONCLUSTERED
(
    [Login] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[BenefitsOfPatients] WITH CHECK ADD CONSTRAINT [FK_BenefitsOfPatients_MedCard] FOREIGN KEY([CardID])
REFERENCES [dbo].[MedCard] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[BenefitsOfPatients] CHECK CONSTRAINT [FK_BenefitsOfPatients_MedCard]
GO
ALTER TABLE [dbo].[BenefitsOfPatients] WITH CHECK ADD CONSTRAINT [FK_BenefitsOfPatients_Spr_Benefit] FOREIGN KEY([BenefitID])
REFERENCES [dbo].[Spr_Benefit] ([ID])
GO
ALTER TABLE [dbo].[BenefitsOfPatients] CHECK CONSTRAINT [FK_BenefitsOfPatients_Spr_Benefit]
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] WITH CHECK ADD CONSTRAINT [FK_ChangesOfPaymentType_MedCard] FOREIGN KEY([CardID])
REFERENCES [dbo].[MedCard] ([ID])
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] CHECK CONSTRAINT [FK_ChangesOfPaymentType_MedCard]
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] WITH CHECK ADD CONSTRAINT [FK_ChangesOfPaymentType_Spr_NewPaymentType] FOREIGN KEY([NewPaymentTypeID])
REFERENCES [dbo].[Spr_PaymentType] ([ID])
GO

```

Рисунок В.9

```

ALTER TABLE [dbo].[ChangesOfPaymentType] CHECK CONSTRAINT [FK_ChangesOfPaymentType_Spr_NewPaymentType]
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] WITH CHECK ADD CONSTRAINT [FK_ChangesOfPaymentType_Spr_OldPaymentType] FOREIGN KEY([OldPaymentTypeID])
REFERENCES [dbo].[Spr_PaymentType] ([ID])
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] CHECK CONSTRAINT [FK_ChangesOfPaymentType_Spr_OldPaymentType]
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] WITH CHECK ADD CONSTRAINT [FK_ChangesOfPaymentType_User] FOREIGN KEY([UserID])
REFERENCES [dbo].[User] ([ID])
GO
ALTER TABLE [dbo].[ChangesOfPaymentType] CHECK CONSTRAINT [FK_ChangesOfPaymentType_User]
GO
ALTER TABLE [dbo].[Death] WITH CHECK ADD CONSTRAINT [FK_Death_AutopsyType] FOREIGN KEY([AutopsyTypeID])
REFERENCES [dbo].[Spr_AutopsyType] ([ID])
GO
ALTER TABLE [dbo].[Death] CHECK CONSTRAINT [FK_Death_AutopsyType]
GO
ALTER TABLE [dbo].[Death] WITH CHECK ADD CONSTRAINT [FK_Death_MedCard] FOREIGN KEY([CardID])
REFERENCES [dbo].[MedCard] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Death] CHECK CONSTRAINT [FK_Death_MedCard]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_Country] FOREIGN KEY([CountryID])
REFERENCES [dbo].[Spr_Country] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_Country]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DeliveryPeriod] FOREIGN KEY([DeliveryPeriodID])
REFERENCES [dbo].[Spr_DeliveryPeriod] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DeliveryPeriod]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DeliveryType] FOREIGN KEY([DeliveryTypeID])
REFERENCES [dbo].[Spr_DeliveryType] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DeliveryType]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_Denial] FOREIGN KEY([DenialID])
REFERENCES [dbo].[Spr_Denial] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_Denial]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DepartmentDischarge] FOREIGN KEY([DepartmentID_Discharge])
REFERENCES [dbo].[Spr_Department] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DepartmentDischarge]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DepartmentReceipt] FOREIGN KEY([DepartmentID_Receipt])
REFERENCES [dbo].[Spr_Department] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DepartmentReceipt]

```

Рисунок В.10



```

GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DiagnosisDischarge] FOREIGN KEY([DiagnosisCode_Discharge])
REFERENCES [dbo].[Spr_Diagnosis] ([Code])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DiagnosisDischarge]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DiagnosisLPU] FOREIGN KEY([DiagnosisCode_LPU])
REFERENCES [dbo].[Spr_Diagnosis] ([Code])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DiagnosisLPU]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DiagnosisPO] FOREIGN KEY([DiagnosisCode_PO])
REFERENCES [dbo].[Spr_Diagnosis] ([Code])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DiagnosisPO]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_DocumentType] FOREIGN KEY([DocumentTypeID])
REFERENCES [dbo].[Spr_DocumentType] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_DocumentType]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_Gender] FOREIGN KEY([GenderCode])
REFERENCES [dbo].[Spr_Gender] ([Code])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_Gender]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_HospitalizationType] FOREIGN KEY([HospitalizationTypeID])
REFERENCES [dbo].[Spr_HospitalizationType] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_HospitalizationType]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_InjuryType] FOREIGN KEY([InjuryTypeID])
REFERENCES [dbo].[Spr_InjuryType] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_InjuryType]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_KLADR] FOREIGN KEY([KLADR_ID])
REFERENCES [dbo].[Spr_KLADR] ([Code])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_KLADR]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_PaymentType] FOREIGN KEY([PaymentTypeID])
REFERENCES [dbo].[Spr_PaymentType] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_PaymentType]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_SendingOrganisation] FOREIGN KEY([SendingOrganisationID])
REFERENCES [dbo].[Spr_SendingOrganisation] ([Code])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_SendingOrganisation]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_SignOfDischarge] FOREIGN KEY([SignOfDischargeID])

```

Рисунок В.11

```

REFERENCES [dbo].[Spr_SignOfResidence] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_SignOfResidence]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_SocialStatus] FOREIGN KEY([SocialStatusID])
REFERENCES [dbo].[Spr_SocialStatus] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_SocialStatus]
GO
ALTER TABLE [dbo].[MedCard] WITH CHECK ADD CONSTRAINT [FK_MedCard_Spr_TreatmentOutcome] FOREIGN KEY([TreatmentOutcomeID])
REFERENCES [dbo].[Spr_TreatmentOutcome] ([ID])
GO
ALTER TABLE [dbo].[MedCard] CHECK CONSTRAINT [FK_MedCard_Spr_TreatmentOutcome]
GO
ALTER TABLE [dbo].[PatientMovement] WITH CHECK ADD CONSTRAINT [FK_PatientMovement_MedCard] FOREIGN KEY([CardID])
REFERENCES [dbo].[MedCard] ([ID])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[PatientMovement] CHECK CONSTRAINT [FK_PatientMovement_MedCard]
GO
ALTER TABLE [dbo].[PatientMovement] WITH CHECK ADD CONSTRAINT [FK_PatientMovement_Spr_Department] FOREIGN KEY([DepartmentID])
REFERENCES [dbo].[Spr_Department] ([ID])
GO
ALTER TABLE [dbo].[PatientMovement] CHECK CONSTRAINT [FK_PatientMovement_Spr_Department]
GO
ALTER TABLE [dbo].[PatientMovement] WITH CHECK ADD CONSTRAINT [FK_PatientMovement_Spr_SignOfDischarge] FOREIGN KEY([SignOfDischargeID])
REFERENCES [dbo].[Spr_SignOfDischarge] ([ID])
GO
ALTER TABLE [dbo].[PatientMovement] CHECK CONSTRAINT [FK_PatientMovement_Spr_SignOfDischarge]
GO
ALTER TABLE [dbo].[User] WITH CHECK ADD CONSTRAINT [FK_User_Role] FOREIGN KEY([RoleID])
REFERENCES [dbo].[Spr_Role] ([ID])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[User] CHECK CONSTRAINT [FK_User_Role]
GO
/***** Object: StoredProcedure [dbo].[Sp_ListOfAdmitted]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Sp_ListOfAdmitted]
    @dateBegin DATETIME,
    @dateEnd DATETIME
AS
SELECT MedCard.ID AS CardID, MedCard.DateTimeReceipt AS DateTimeReceipt, Spr_Department.[Name] as DepartmentName, MedCard.LName AS LName, MedCard.FName AS FName, MedCard.PName AS MName
FROM MedCard INNER JOIN Spr_Department ON MedCard.DepartmentID=Receipt-Spr_Department.ID
WHERE ((DateTimeReceipt >= convert(datetime, DATEADD(hh, 7, @dateBegin),104)
AND DateTimeReceipt<=convert(datetime, DATEADD(ss, 1, DATEADD(hh, 7, @dateEnd)),104))
)

```

Рисунок В.12

```

ORDER BY DepartmentName, LName
GO
/***** Object: StoredProcedure [dbo].[Sp_SignalStatement]    Script Date: 26.06.2023 11:36:05 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Sp_SignalStatement]
    @date DATETIME
AS
BEGIN
SELECT Spr_Department.ID AS DepartmentID, Spr_Department.[Name] AS DepartmentName,
Sum(case when alias_1.na_nach is not null THEN alias_1.na_nach ELSE 0 END) AS OnStart,
Sum(CASE WHEN alias_1.post is not null THEN alias_1.post ELSE 0 END) AS AdmittedPatientsCount,
Sum(CASE WHEN alias_1.perevod_from is not null THEN alias_1.perevod_from ELSE 0 END) AS TransferredFrom,
Sum(CASE WHEN alias_1.perevod_in is not null THEN alias_1.perevod_in ELSE 0 END) AS TransferredIn,
Sum(CASE WHEN alias_1.vipisan is not null THEN alias_1.vipisan ELSE 0 END) AS DischargedCount,
Sum(CASE WHEN alias_1.ymer is not null THEN alias_1.ymer ELSE 0 END) AS DiesCount,
Sum(CASE WHEN alias_1.na_konec is not null THEN alias_1.na_konec ELSE 0 END) AS OnEnd
FROM ((SELECT PatientMovement.DepartmentID AS Dotd,
CASE WHEN (PatientMovement.DateTimeReceipt < convert(datetime,@date,104)
AND (PatientMovement.DateTimeDischarge>=convert(datetime,@date,104) OR PatientMovement.DateTimeDischarge IS NULL)
OR (PatientMovement.DateTimeReceipt-MedCard.DateTimeReceipt AND MedCard.DateTimeReceipt=convert(datetime,@date,104)
AND cast(MedCard.DateTimeReceipt as time) < ('7:00:00') and MedCard.DepartmentID_Receipt=PatientMovement.DepartmentID))
AND MedCard.DenialID IS NULL THEN 1 ELSE 0 END AS na_nach,
CASE WHEN (((MedCard.DateTimeReceipt > convert(datetime,@date,104)
AND MedCard.DateTimeReceipt<=convert(datetime,@date,104))
AND PatientMovement.DateTimeReceipt-MedCard.DateTimeReceipt AND PatientMovement.DepartmentID=MedCard.DepartmentID_Receipt)
OR (MedCard.DateTimeReceipt=convert(datetime,@date,104)
AND CAST(MedCard.DateTimeReceipt as time)>=('7:00:00'))
AND PatientMovement.DateTimeReceipt-MedCard.DateTimeReceipt AND PatientMovement.DepartmentID=MedCard.DepartmentID_Receipt)
OR (MedCard.DateTimeReceipt>dateadd(d,1,convert(datetime,@date,104))
AND CAST(MedCard.DateTimeReceipt as time) < ('7:00:00') AND PatientMovement.DateTimeReceipt-MedCard.DateTimeReceipt
AND PatientMovement.DepartmentID=MedCard.DepartmentID_Receipt) AND DenialID IS NULL) THEN 1 ELSE 0 END AS post,
CASE WHEN (PatientMovement.DateTimeReceipt<=convert(datetime,@date,104)
AND PatientMovement.DateTimeReceipt<>MedCard.DateTimeReceipt) THEN 1 ELSE 0 END AS perevod_from,
CASE WHEN (PatientMovement.DateTimeDischarge<=convert(datetime,@date,104)
AND PatientMovement.DateTimeDischarge<=convert(datetime,@date,104) AND PatientMovement.SignOfDischargeID=4 THEN 1 ELSE 0 END AS perevod_in,
CASE WHEN (PatientMovement.DateTimeDischarge<=convert(datetime,@date,104)
AND PatientMovement.DateTimeDischarge<=convert(datetime,@date,104)
AND (PatientMovement.SignOfDischargeID=1 or PatientMovement.SignOfDischargeID=2 or PatientMovement.SignOfDischargeID=5 or PatientMovement.SignOfDischargeID=6)
AND MedCard.DenialID IS NULL THEN 1 ELSE 0 END AS vipisan,
CASE WHEN (PatientMovement.DateTimeDischarge<=convert(datetime,@date,104)
AND PatientMovement.DateTimeDischarge<=convert(datetime,@date,104)) AND PatientMovement.SignOfDischargeID=3
AND MedCard.DenialID IS NULL THEN 1 ELSE 0 END AS ymer,
CASE WHEN (PatientMovement.DateTimeReceipt<=convert(datetime,@date,104) OR (PatientMovement.DateTimeReceipt-MedCard.DateTimeReceipt
AND MedCard.DateTimeReceipt>dateadd(d,1,convert(datetime,@date,104))
AND cast(MedCard.DateTimeReceipt as time) < ('7:00:00') AND PatientMovement.DepartmentID=MedCard.DepartmentID_Receipt ))
AND (PatientMovement.DateTimeDischarge > convert(datetime,@date,104) OR PatientMovement.DateTimeDischarge IS NULL)
AND MedCard.DenialID IS NULL THEN 1 ELSE 0 END AS na_konec
FROM MedCard INNER JOIN PatientMovement ON MedCard.ID=PatientMovement.CardID) AS alias 1

```

Рисунок В.13

```

RIGHT OUTER JOIN Spr_Department ON alias_1.Dotd = Spr_Department.ID)
where Spr_Department.IsDayDepartment is null and Spr_Department.Beds>0
GROUP BY Spr_Department.ID, Spr_Department.[Name], alias_1.Dotd, Spr_Department.SerialNumber
ORDER BY Spr_Department.SerialNumber
END;

```

Рисунок В.14

## Приложение Г

### Тест-кейсы

Таблица Г.1 – Общая информация о тестировании

Название проекта	MedPlus
Номер версии	1.0
Имя тестера	Молокотин В.С.
Даты тестирования	20.05.2023

Таблица Г.2

Test Case #	TC_Module_Patient_2
Приоритет теста	Средний
Название тестирования/Имя	Попытка регистрации пациента с незаполненными обязательными полями
Резюме испытания	Необходимо определить реакцию системы на попытку регистрации пациента с незаполненными обязательными полями, а именно «Вид госпитализации»
Шаги тестирования	1. Запустить программу 2. Войти в систему (ввести тестовые логин и пароль) 3. Открыть страницу регистрации пациента 4. Нажать кнопку сохранить, не заполняя поля
Данные тестирования	Логин: rom_rad Пароль: 123
Ожидаемый результат	После попытки регистрации пациента с незаполненными обязательными полями система выводит сообщение об ошибке с перечнем полей, которые необходимо заполнить.
Фактический результат	Система сообщила об успешном сохранении карты пациента
Предпосылки	
Постусловия	Удалить созданную карту пациента в случае провала теста
Статус (Pass/Fail)	Fail
Комментарии	

Таблица Г.3

<b>Test Case #</b>	TC_Module_Patient_1
<b>Приоритет теста</b>	Высокий
<b>Название тестирования/Имя</b>	Попытка регистрации пациента с заполненными обязательными полями
<b>Резюме испытания</b>	Необходимо определить реакцию системы на попытку регистрации пациента с заполненными обязательными полями, а именно «Вид госпитализации»
<b>Шаги тестирования</b>	1. Запустить программу 2. Войти в систему (ввести тестовые логин и пароль) 3. Открыть страницу регистрации пациента 4. Заполнить обязательные поля 5. Нажать кнопку сохранить
<b>Данные тестирования</b>	Логин: tom_rad Пароль: 123
<b>Ожидаемый результат</b>	После попытки регистрации пациента с заполненными обязательными полями система сохранит карту в БД и выведет сообщение об успешной регистрации пациента
<b>Фактический результат</b>	Система сохранит карту в БД и выведет сообщение об успешной регистрации пациента
<b>Предпосылки</b>	
<b>Постусловия</b>	
<b>Статус (Pass/Fail)</b>	Pass
<b>Комментарии</b>	

Таблица Г.4

<b>Test Case #</b>	TC_Integration_Patient_PatientMovements_1
<b>Приоритет теста</b>	Средний
<b>Название тестирования/Имя</b>	Проверка взаимодействия модулей пациента и движений
<b>Резюме испытания</b>	Необходимо проверить корректность данных при открытии движений пациента. Должны быть доступны только движения выбранного пациента.
<b>Шаги тестирования</b>	1. Запустить программу 2. Войти в систему (ввести тестовые логин и пароль) 3. Открыть страницу списка пациентов 4. Открыть медкарту с номером 3 5. Открыть движения пациента
<b>Данные тестирования</b>	Логин: tom_gad Пароль: 123 Номер медкарты: 3
<b>Ожидаемый результат</b>	В списке движений все движения имеют значение 3 для поля «Номер карты»
<b>Фактический результат</b>	В списке движений все движения имеют значение 3 для поля «Номер карты»
<b>Предпосылки</b>	
<b>Постусловия</b>	
<b>Статус (Pass/Fail)</b>	Pass
<b>Комментарии</b>	