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 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 |
| 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 стационарной больницы.

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.

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



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

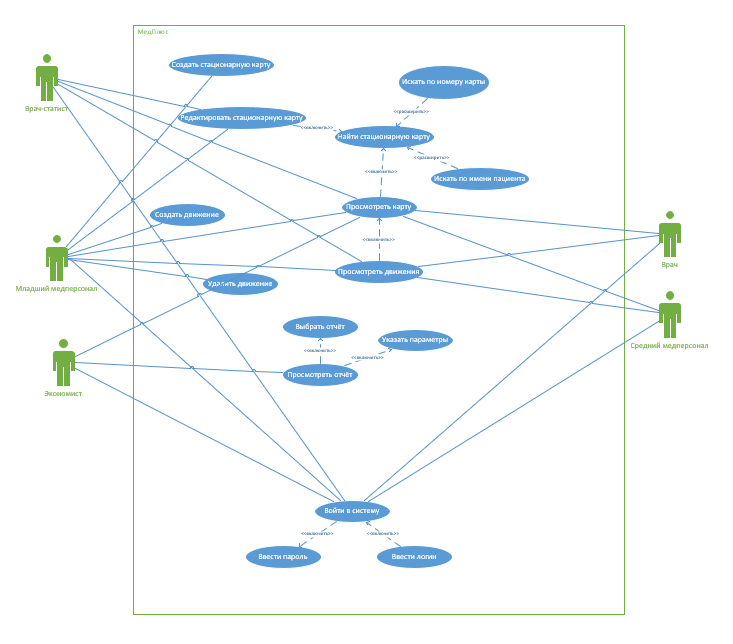


Рисунок 2.1 – Диаграмма вариантов использования программной системы «МедПлюс»

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».

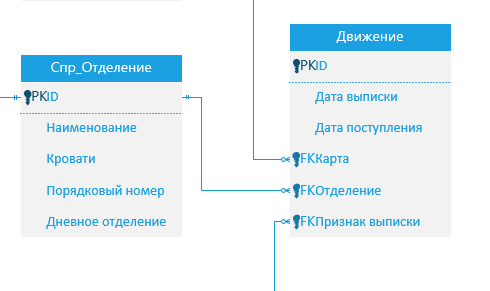


Рисунок 2.2 – ER-диаграмма для программной системы «Медплюс» (1)

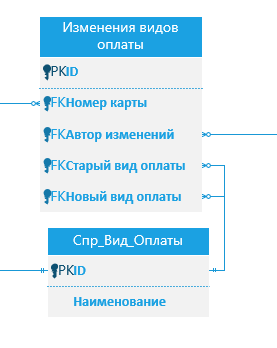


Рисунок 2.3 – ER-диаграмма для программной системы «Медплюс» (2)

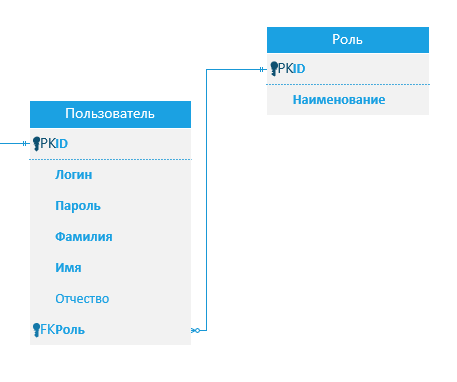


Рисунок 2.4 – ER-диаграмма для программной системы «МедПлюс» (3)

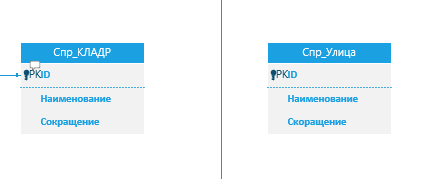


Рисунок 2.5 – ER-диаграмма для программной системы «МедПлюс» (3)

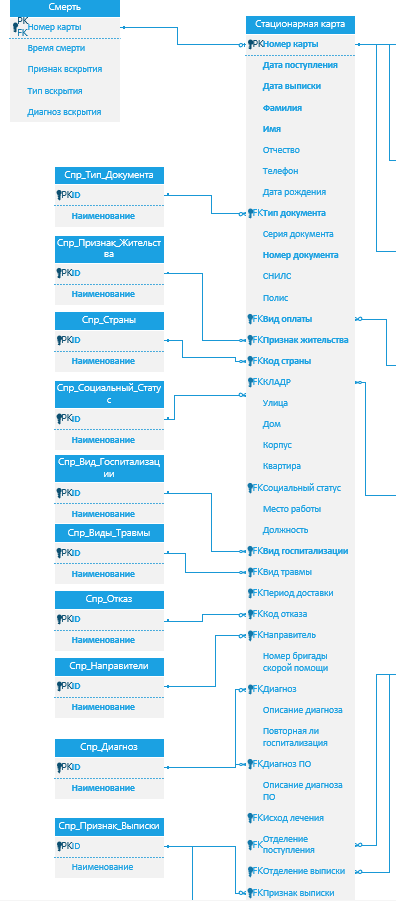


Рисунок 2.6 – ER-диаграмма для программной системы «МедПлюс» (4)

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 | Содержит данные о лечении пациента - номер карты; - дата поступления; - дата выписки; - отделение поступления; - отделение выписки; - тип документа, подтверждающего личность; - номер документа; - серия документа; - ФИО пациента; - пол; - дата рождения; - признак жительства; - страна; - адрес; - телефон; - социальный статус; - место работы; - должность; - тип оплаты; - серия полиса; - номер полиса; - СНИЛС; - направитель; - дата направления; - тип госпитализации; - тип доставки; - номер бригады скорой помощи - диагноз направителя; - диагноз приёмного отделения; - диагноз выписки; - признак повторной госпитализации; - вид травмы; - период доставки; - дата и время выписки; - признак выписки; - исход лечения; - признак отказа; | Номер карты (PK) | Тип документа - Стационарная карта (n…1) Отделение - Стационарная карта (n…1) Признак жительства - Стационарная карта (n…1) Страна - Стационарная карта (n…1) Социальный статус - Стационарная карта (n…1) Тип оплаты - Стационарная карта (n…1) Направитель - Стационарная карта (n…1) Тип госпитализации - Стационарная карта (n…1) Тип доставки - Стационарная карта (n…1) Диагноз - Стационарная карта (n…1) Вид травмы - Стационарная карта (n…1) Период доставки - Стационарная карта (n…1) Признак выписки - Стационарная карта (n...1) Исход лечения - Стационарная карта (n...1) Отказ - Стационарная карта (n...1) |
| Движение | Содержит данные о движении пациента: - номер движения; - признак отказа; - отделение; - дата и время поступления; - дата и время выписки; - номер карты. | Номер движения (PK) | Стационарная карта - Движение (n…1) Отделение - Движение (n…1) Признак выписки - Движение (n…1) |
| Смерть | - номер карты; - время смерти; - признак вскрытия; - тип вскрытия; - диагноз вскрытия. | Номер карты (PK) | Стационарная карта - Смерть (1...1) Тип вскрытия - Смерть (n…1) |
| Изменение оплаты | - номер изменения; - номер карты; - старый вид оплаты; - новый вид оплаты; - пользователь; - дата и время изменения. | Номер изменения (PK) | 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.

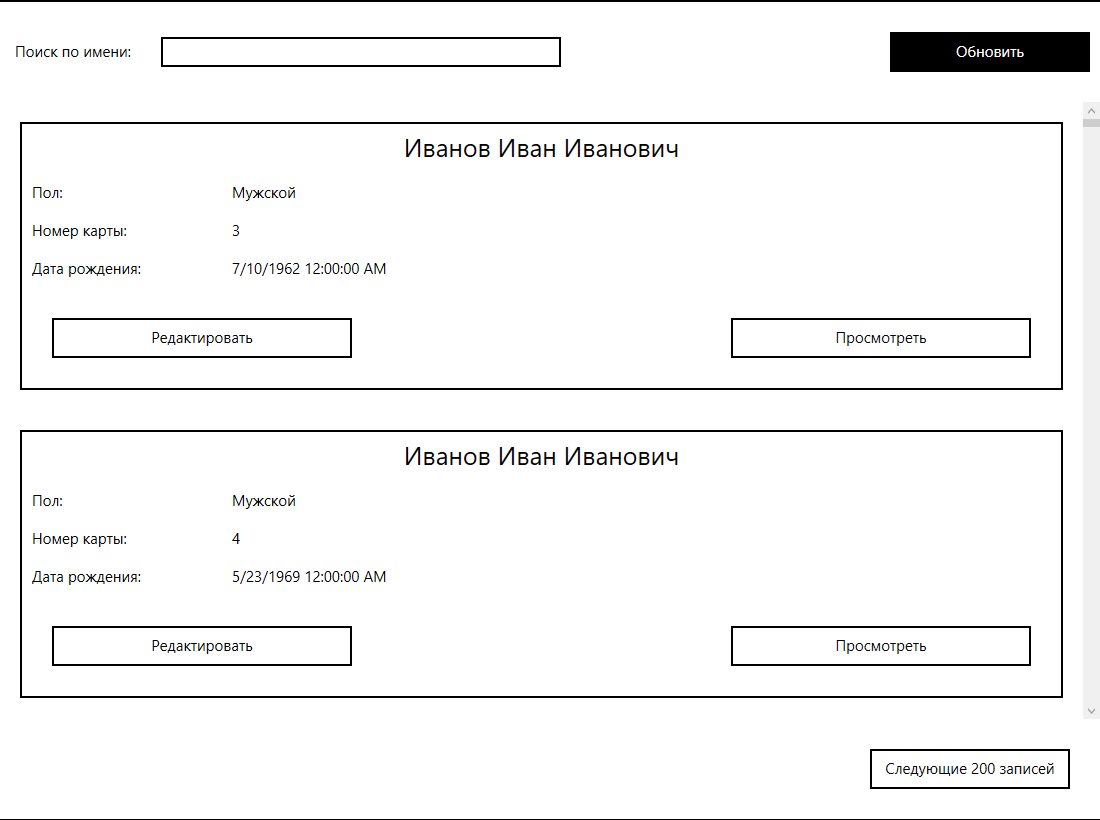


Рисунок 2.9 – Прототип страницы пациентов

На рисунке 2.10 представлен прототип страницы редактирования карты пациента.

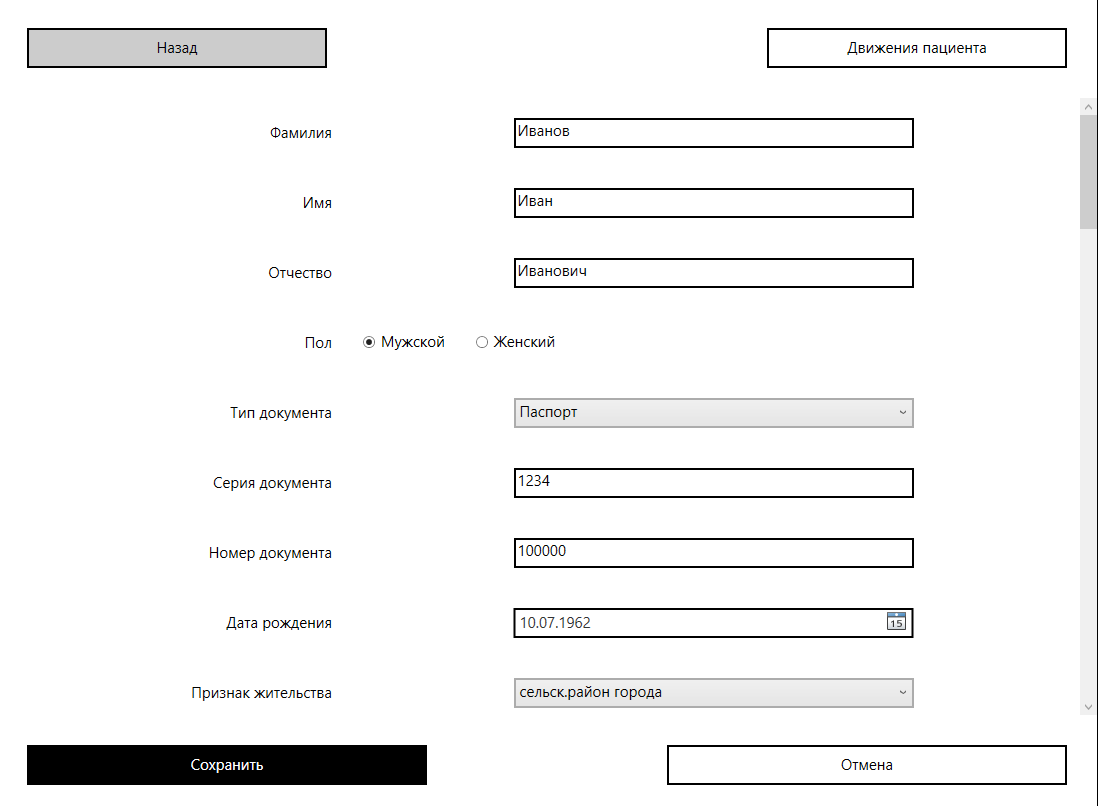


Рисунок 2.10 – Прототип страницы редактирования карты пациента

На рисунке 2.11 представлен прототип страницы редактирования движения пациента.

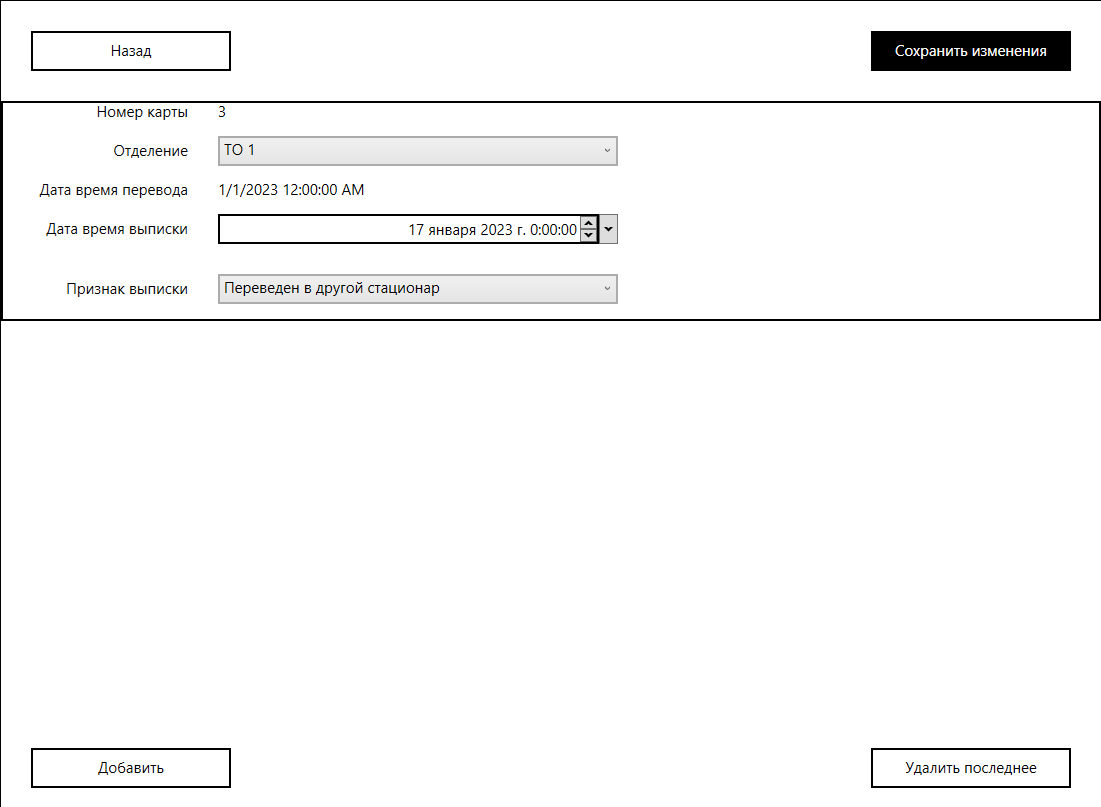


Рисунок 2.11 – Прототип страницы редактирования движения пациента

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



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.

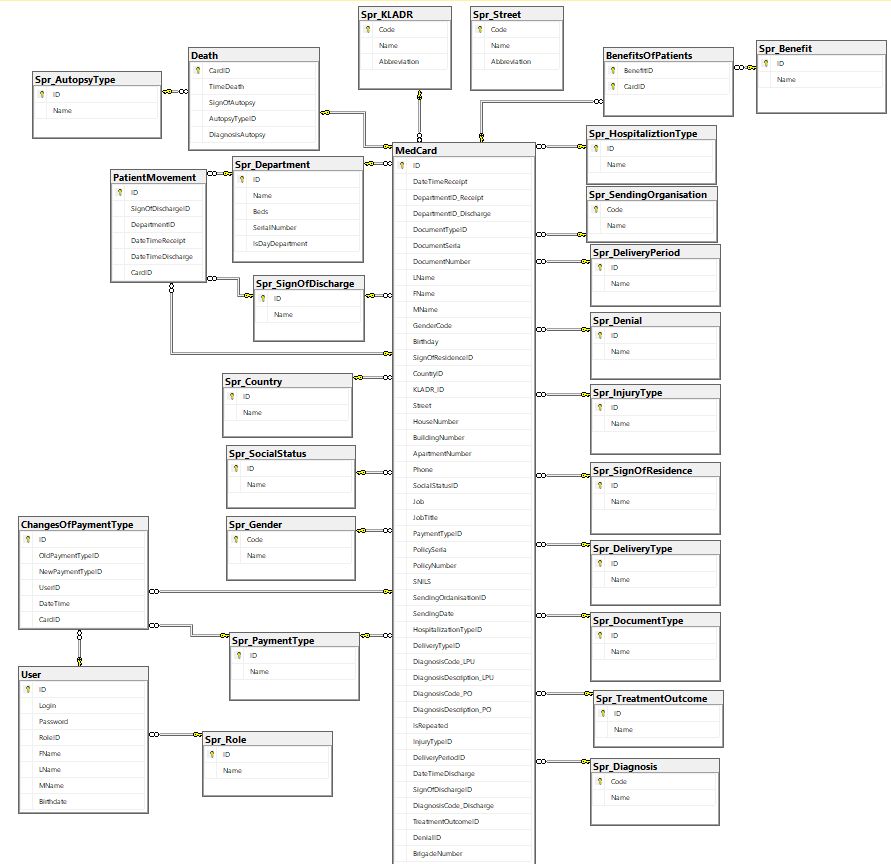


Рисунок 3.1 – Диаграмма БД в SSMS

На таблице 3.2 представлен перечень таблиц БД с кратким описанием.

Таблица 3.2 – Перечень таблиц БД



3.3 Описание программных модулей

На рисунке 3.2 показана структура проекта приложения MedPlus.

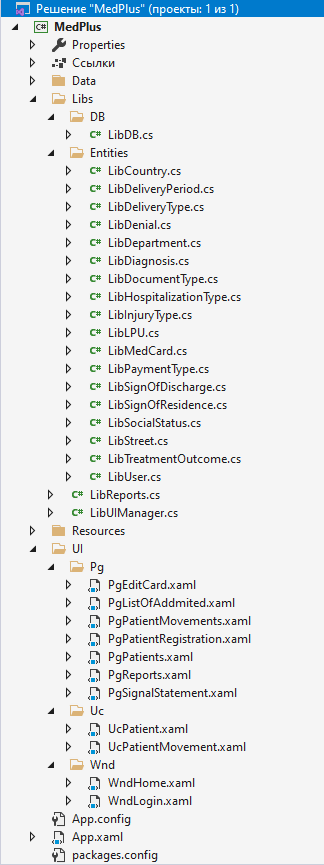


Рисунок 3.2 – Структура приложения MedPlus

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.

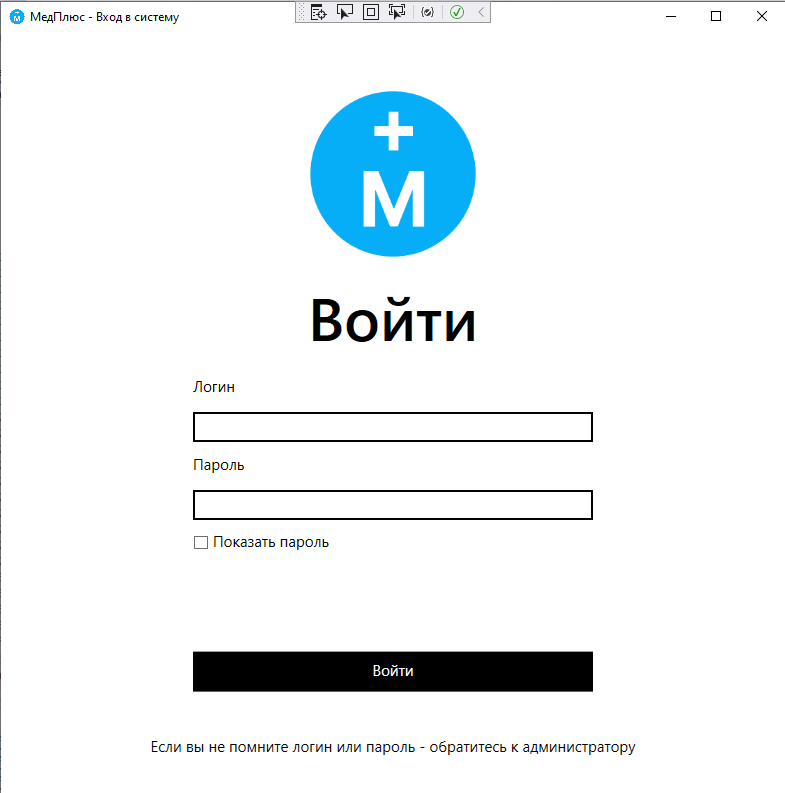


Рисунок 3.3 – Экранная форма окна «Вход в систему»

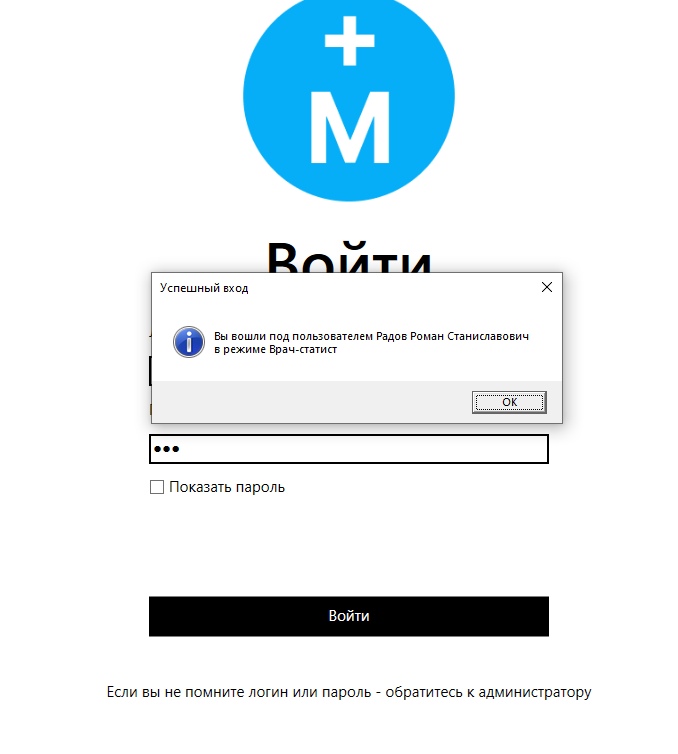


Рисунок 3.4 – Результат входа в окне «Вход в систему»

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.

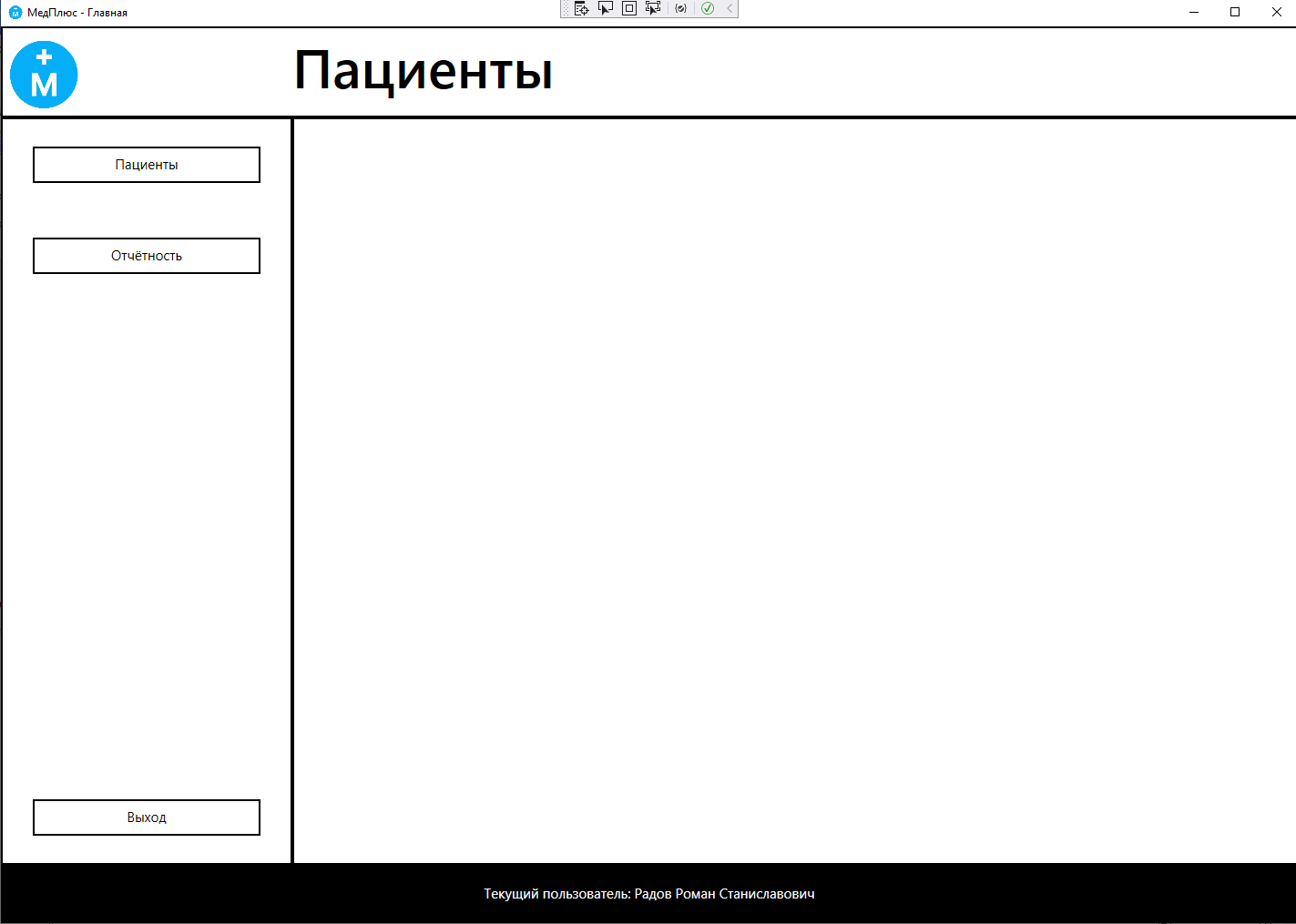


Рисунок 3.5 – Экранная форма окна «Главная»

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.

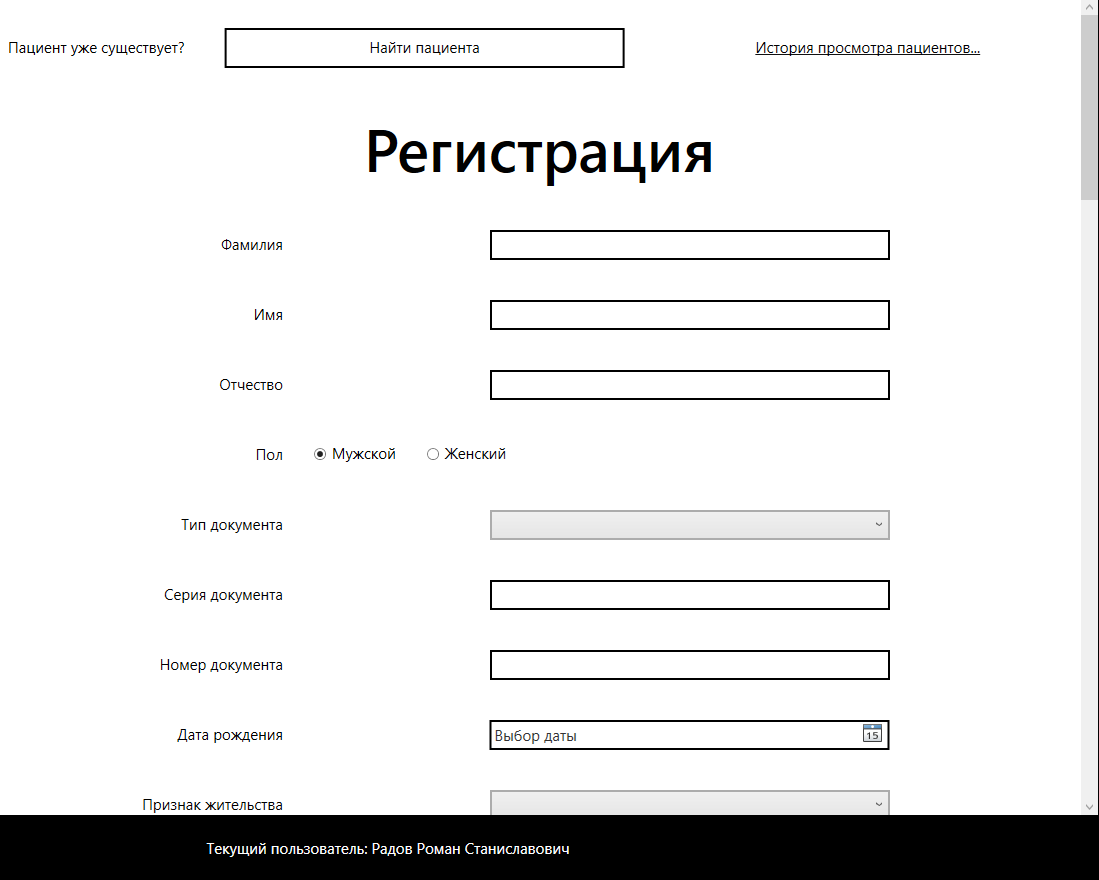


Рисунок 3.6 – Экранная форма страницы регистрации пациента

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.

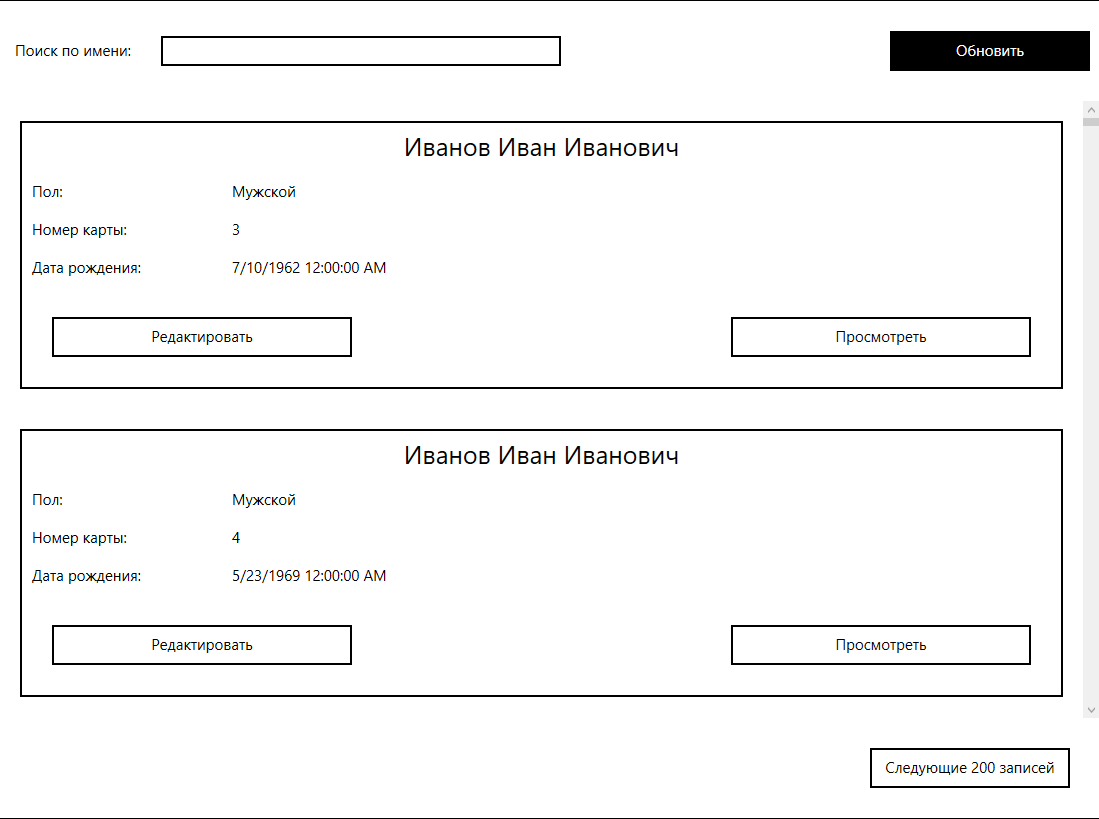


Рисунок 3.7 – Экранная форма страницы списка пациентов

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.

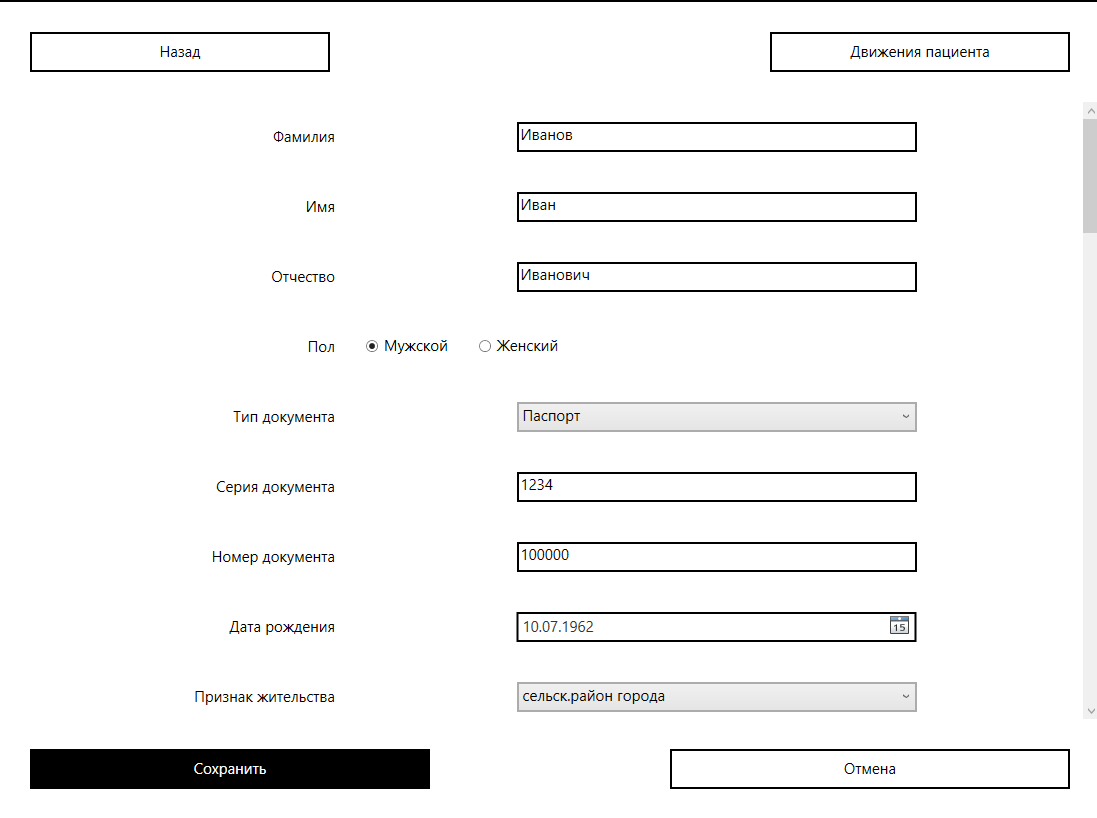


Рисунок 3.8 – Экранная форма страницы «Редактирование карты пациента»

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.

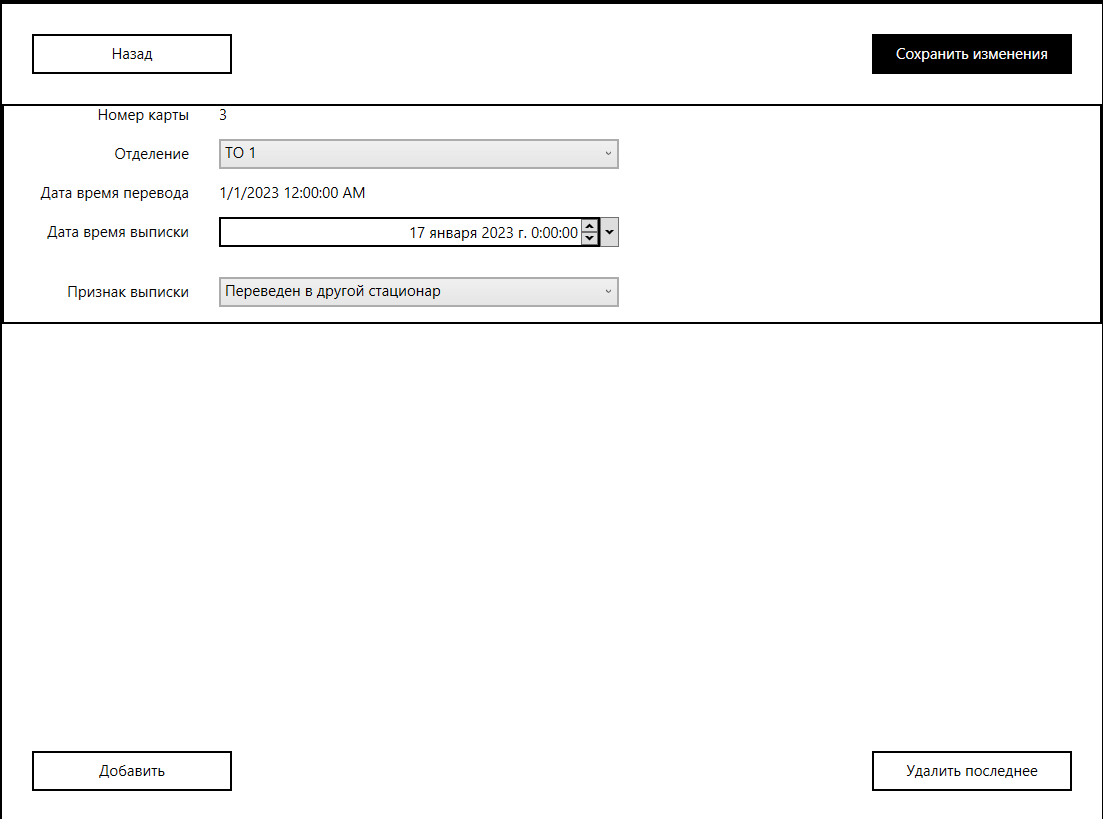


Рисунок 3.9 – Экранная форма страницы «Движения пациента»

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.

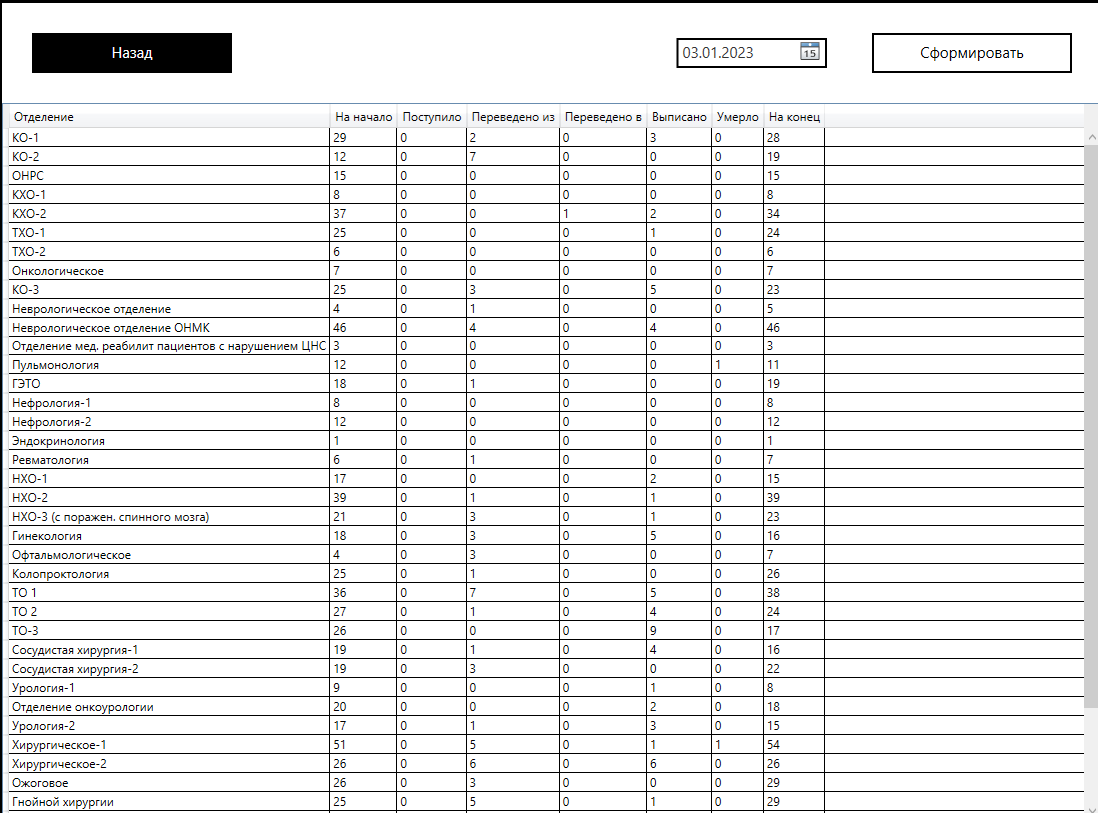


Рисунок 3.10 – Экранная форма страницы «Сигнальная ведомость»

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.

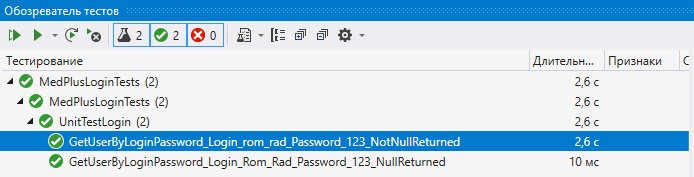


Рисунок 4.1 – Результаты тестирования модуля входа в систему

Для модуля пациента были созданы два тест-кейса, показанные на рисунках Г.2 и Г3.

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.

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.

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 определенны 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 быдла составлена эксплуатационная документация в виде руководства пользователя.

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 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: Стандартинформ, 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.

Приложение А

Техническое задание. Требования к программным модулям

1 Введение

1.1 Наименование программы

Разработка модулей для автоматизации процесса анализа движения пациентов в лечебных отделениях больницы Наименование программной системы «МедПлюс»

1.2 Краткая характеристика области применения

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

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 Молокотиным 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».

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 организация 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.

Приложение Б

Программный код

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



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

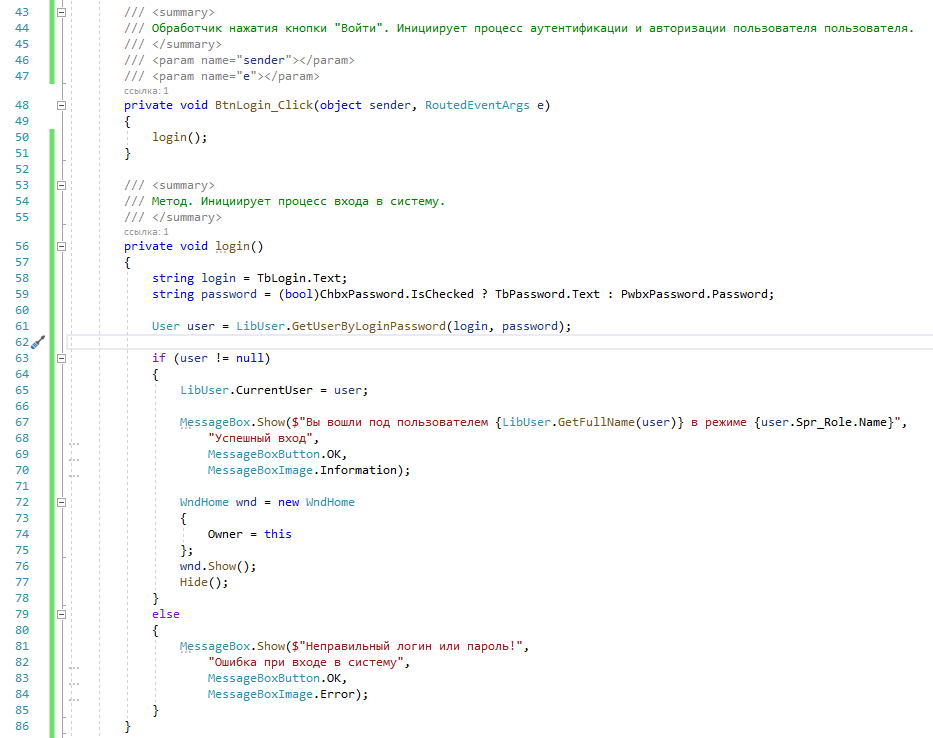


Рисунок Б.2 – Код обработчика для кнопки «Войти» на окне «Вход в систему»

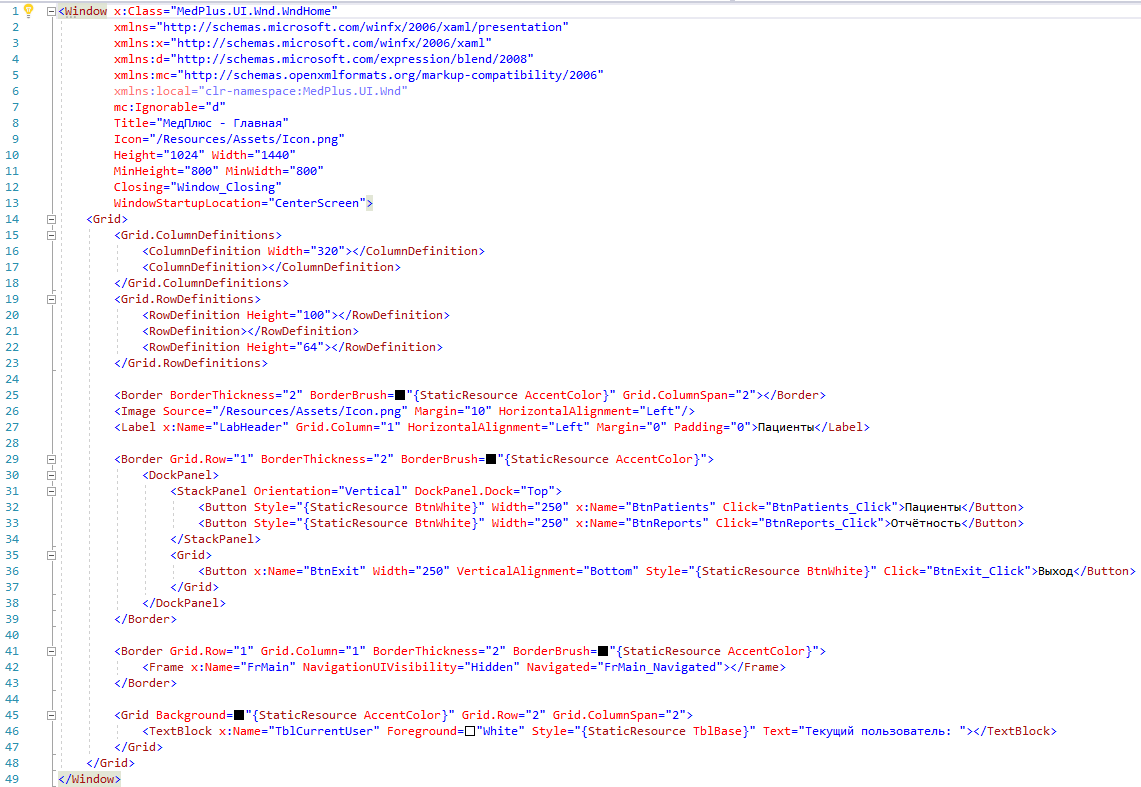


Рисунок Б.3 – Код разметки окна «Главная»

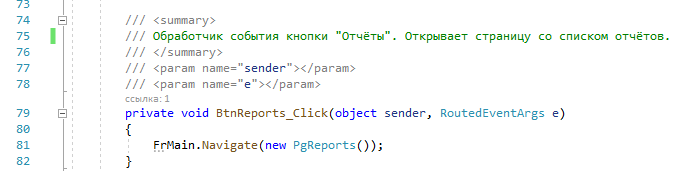


Рисунок Б.4 – Код обработчика нажатия кнопки «Отчёты» на окне «Главная»

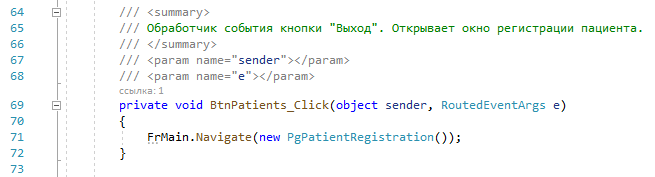


Рисунок Б.5 – Код обработчика нажатия кнопки «Пациенты» на окне «Главная»

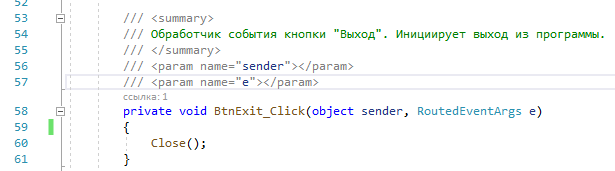


Рисунок Б.6 – Код обработчика нажатия кнопки «Выход» на окне «Главная»



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

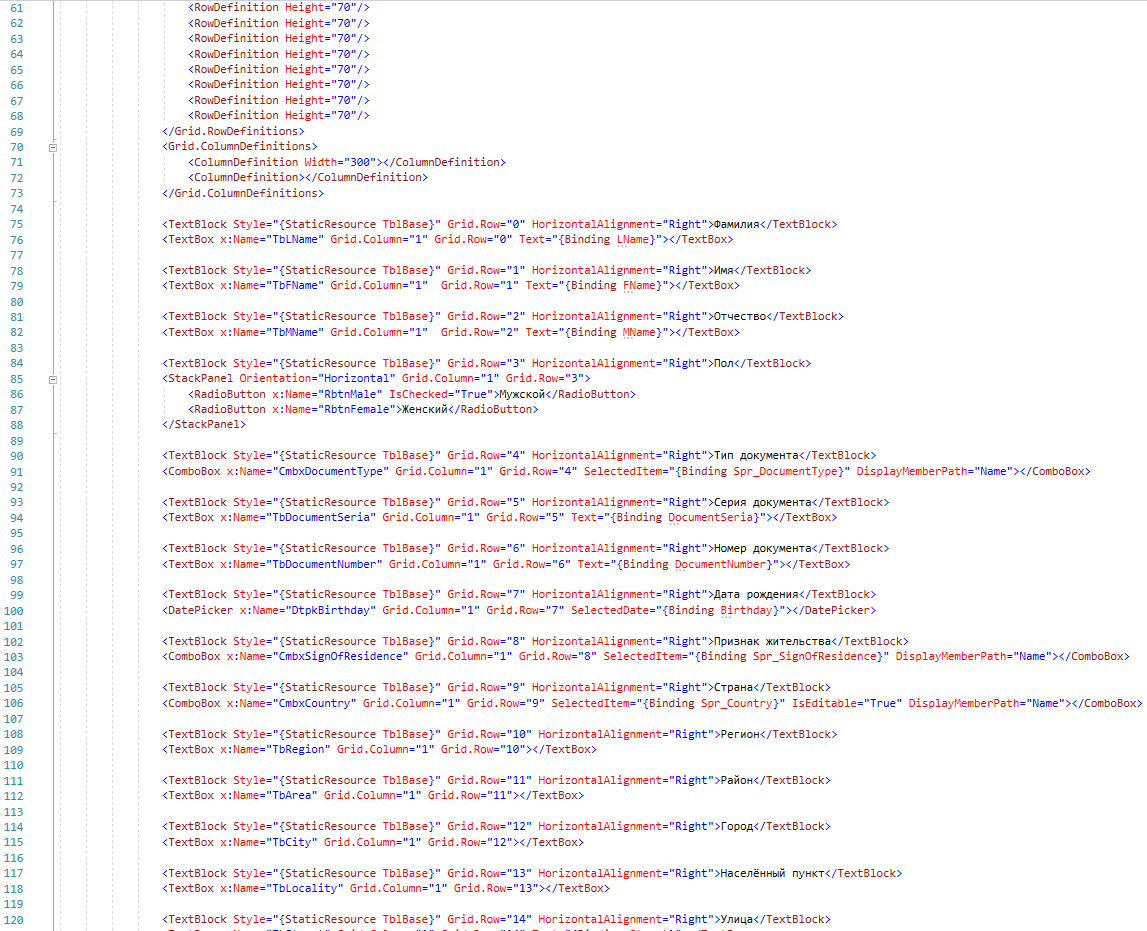


Рисунок Б.8 – Код разметки страницы «Регистрация пациента» (2)

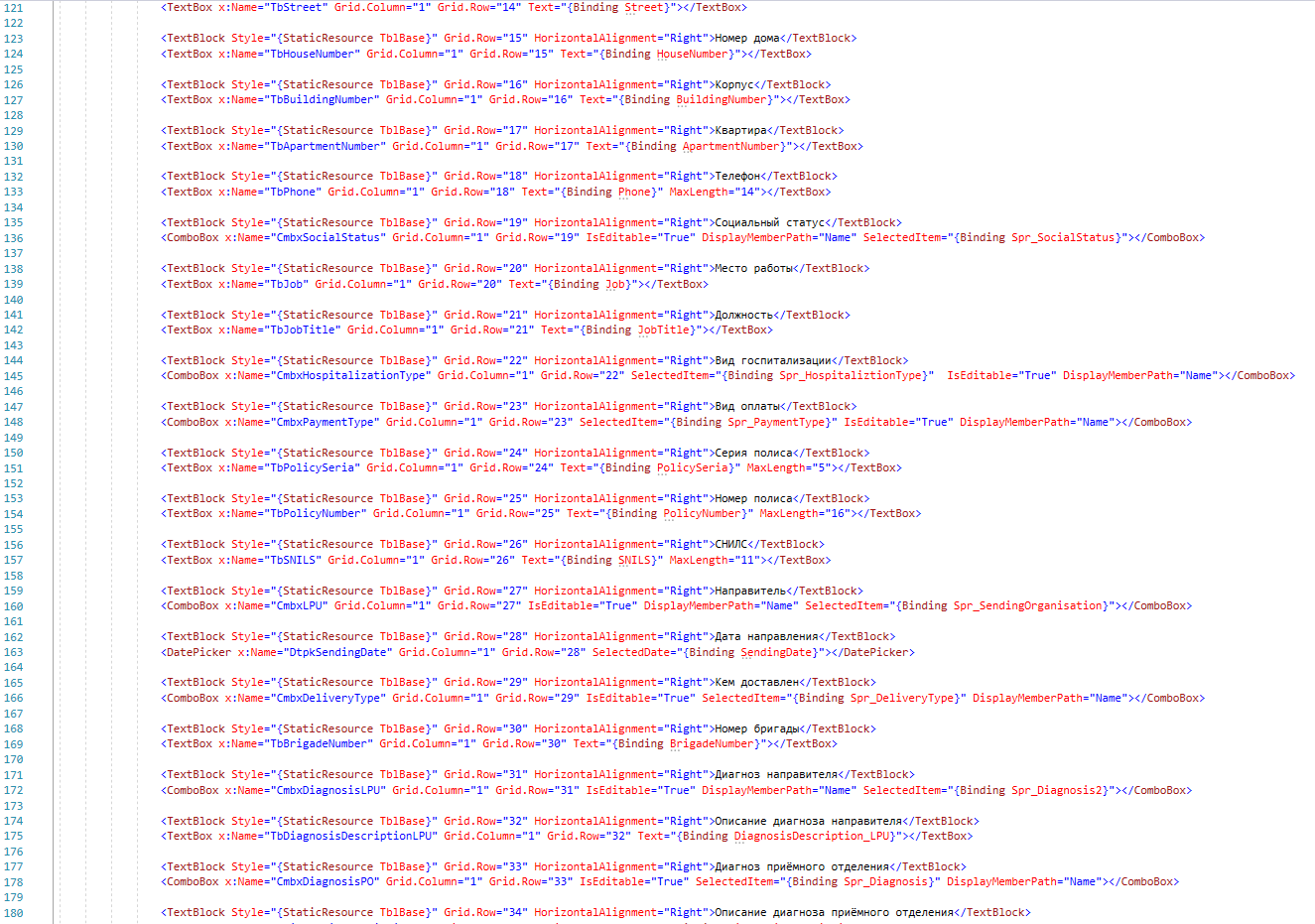


Рисунок Б.9 – Код разметки страницы «Регистрация пациента» (2)



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

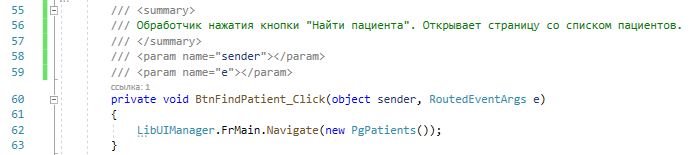


Рисунок Б.11 – Код обработчика нажатия кнопки «Найти пациента» на странице «Регистрация пациента»



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

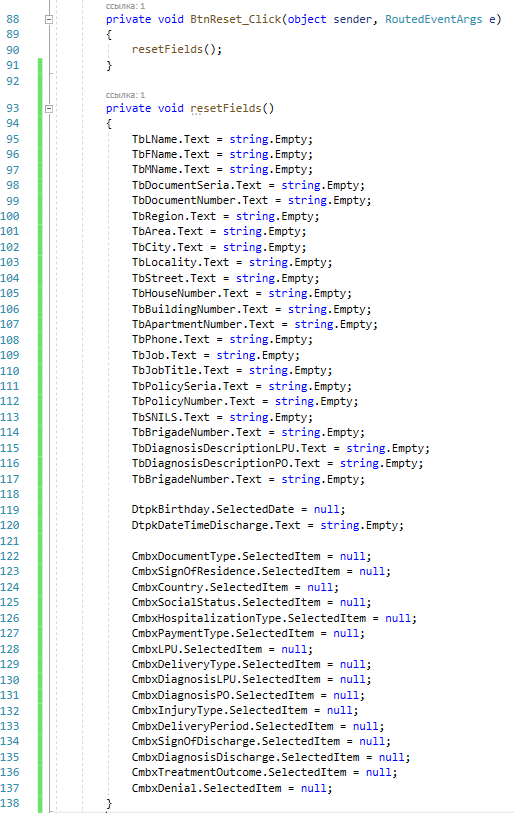


Рисунок Б.13 – Код обработчика нажатия кнопки «Очистить поля» на странице «Регистрация пациента»



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



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

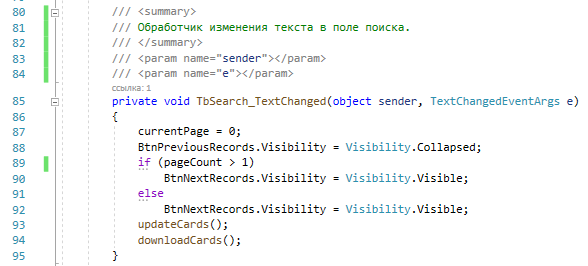


Рисунок Б.16 – Код обработчика текстового поля поиска на странице «Список пациентов»

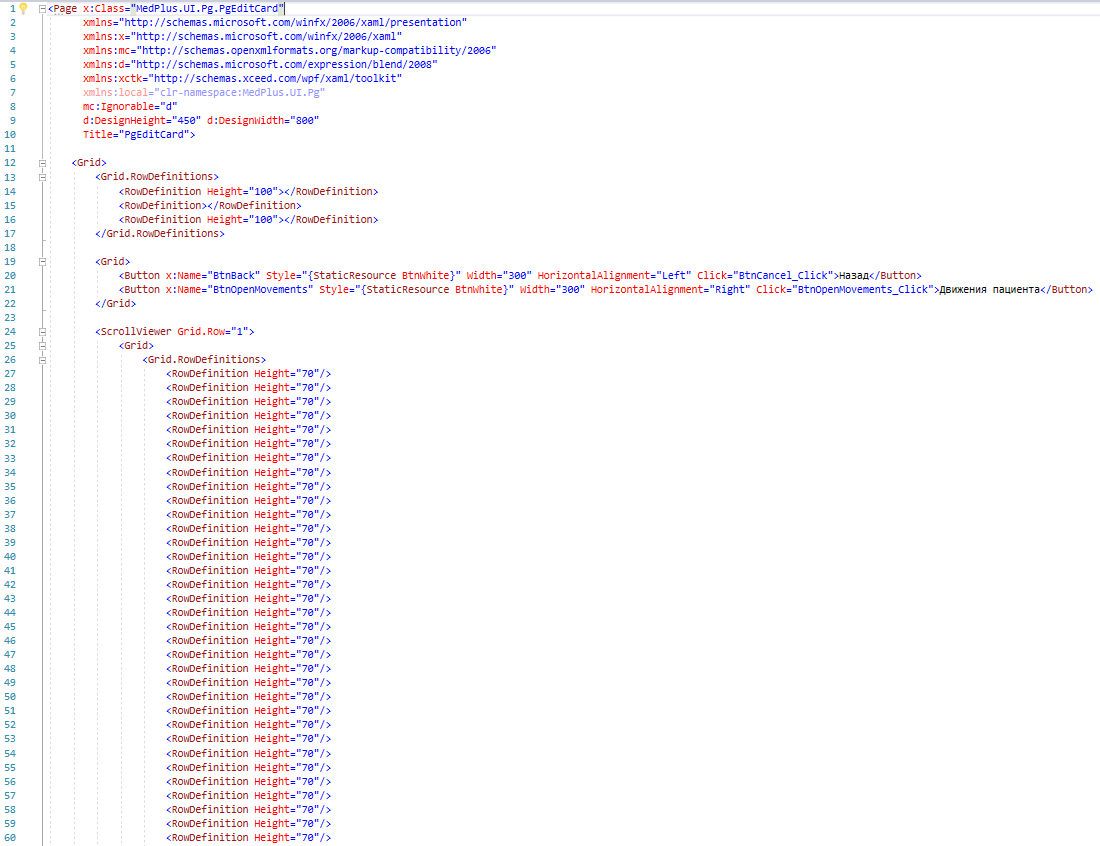


Рисунок Б.17 – Код разметки страницы «Редактирование карты пациента» (1)

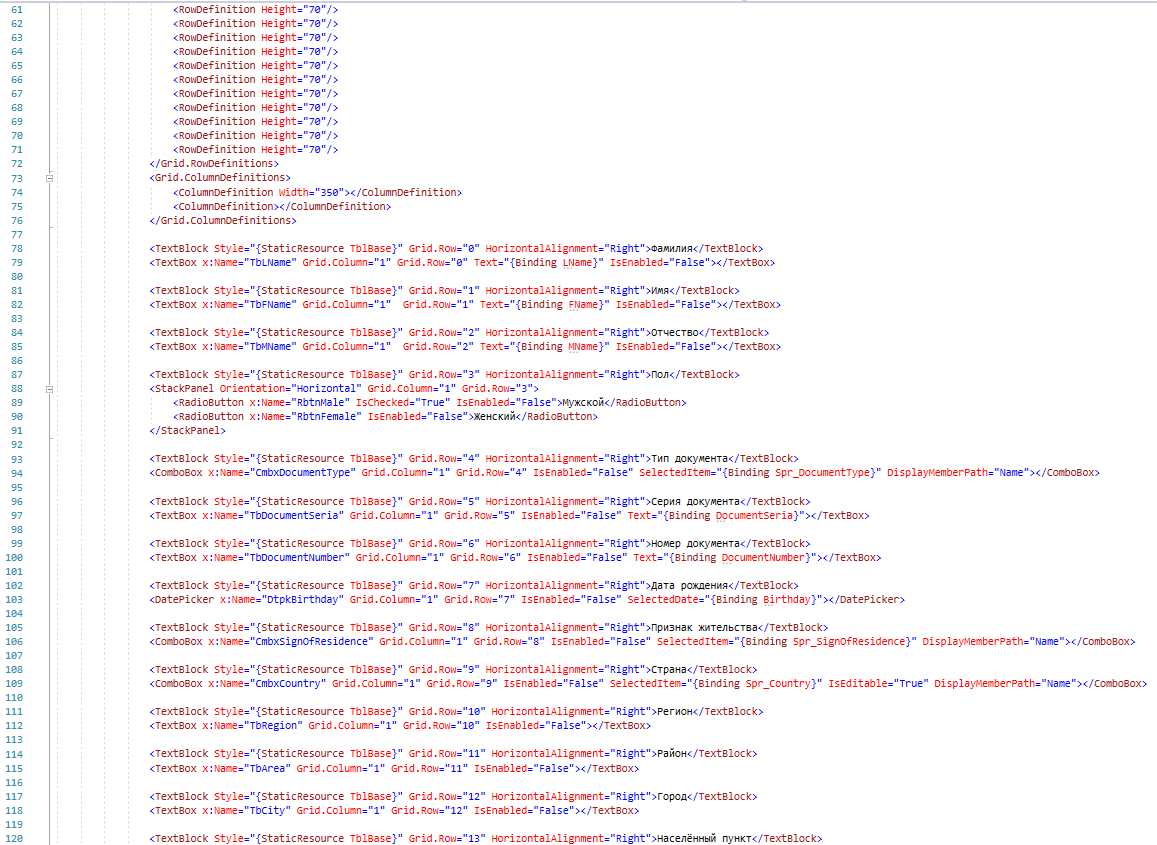


Рисунок Б.18 – Код разметки страницы «Редактирование карты пациента» (2)

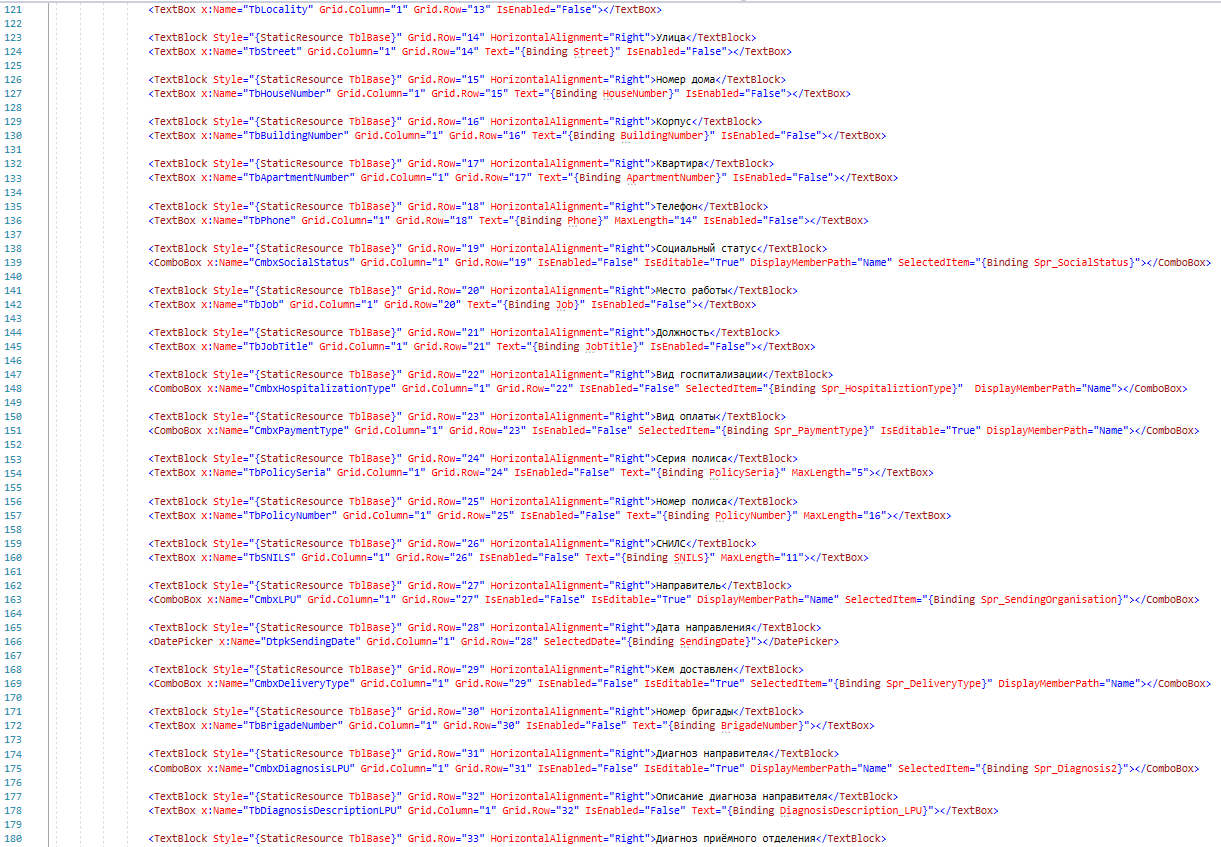


Рисунок Б.19 – Код разметки страницы «Редактирование карты пациента» (3)



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

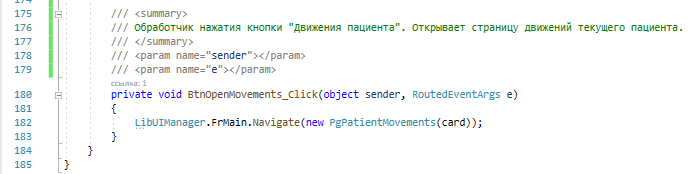


Рисунок Б.21 – Код обработчика нажатия кнопки «Движения пациента» на странице «Редактирование карты пациента»

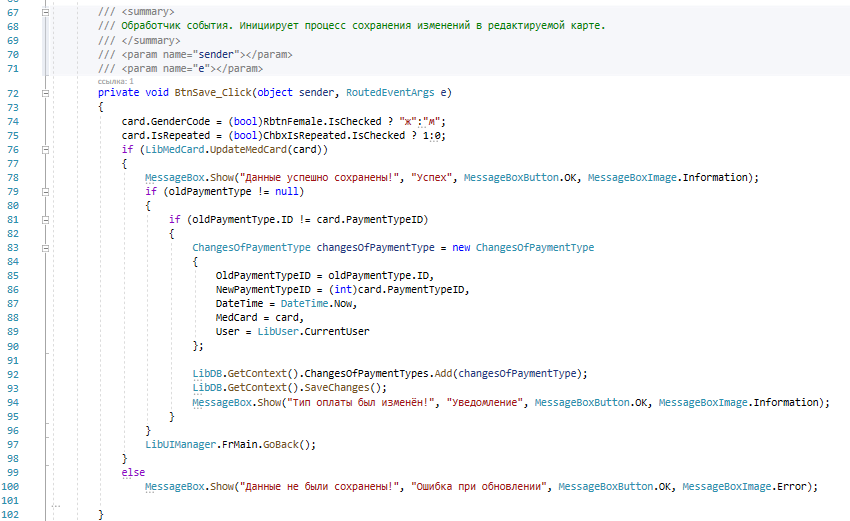


Рисунок Б.22 – Код обработчика нажатия кнопки «Сохранить» на странице «Редактирование карты пациента»



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

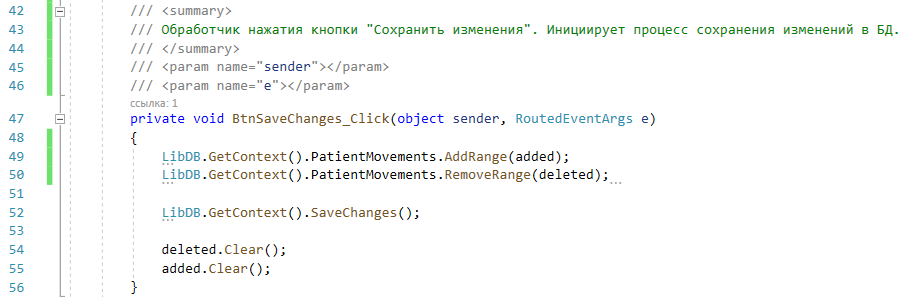


Рисунок Б.24 – Код обработчика нажатия кнопки «Сохранить изменения» на странице «Движения пациента»

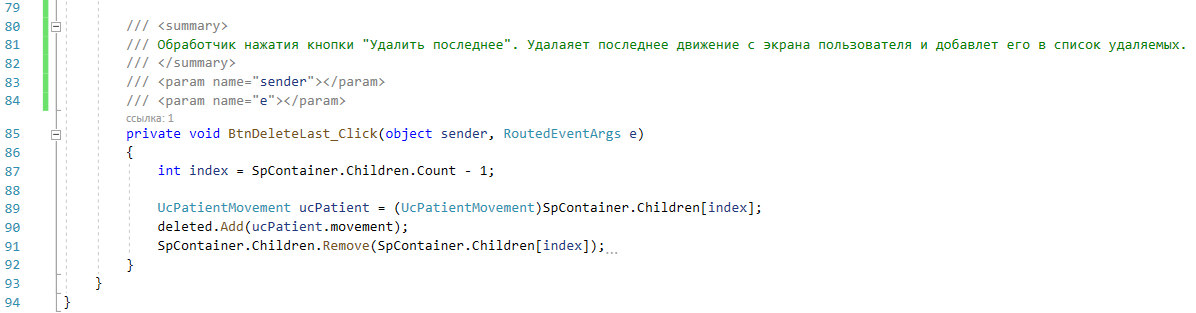


Рисунок Б.25 – Код обработчика нажатия кнопки «Удалить последнее» на странице «Движения пациента»

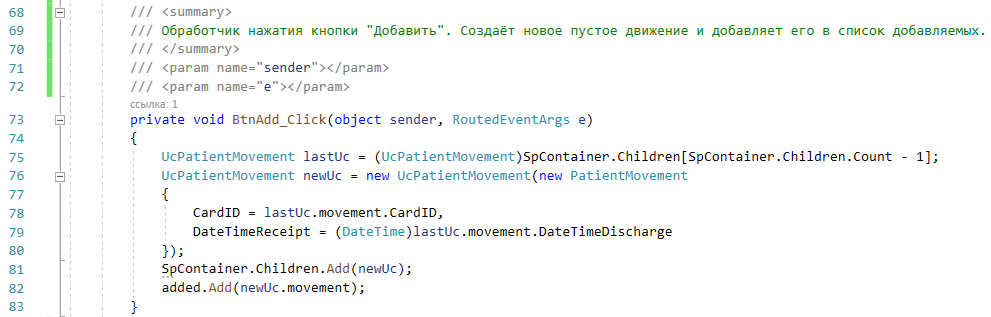


Рисунок Б.26 – Код обработчика нажатия кнопки «Добавить» на странице «Движения пациента»

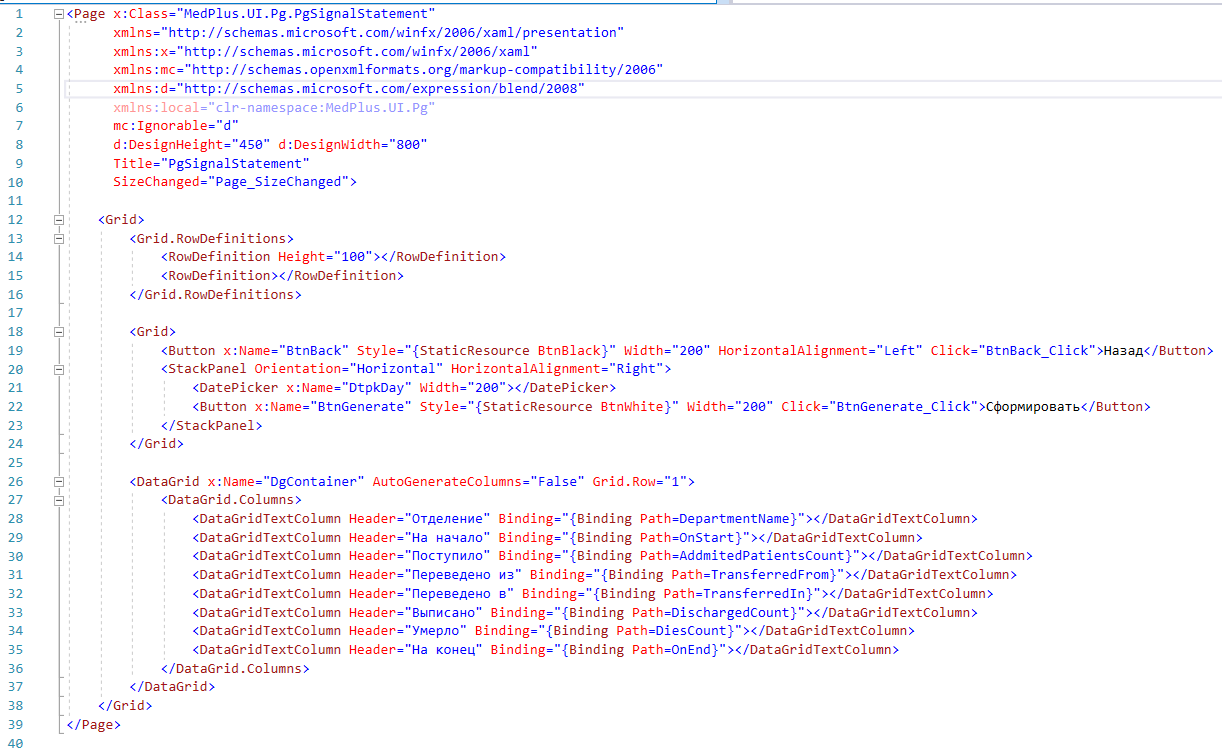


Рисунок Б.27 – Код разметки страницы «Сигнальная ведомость»

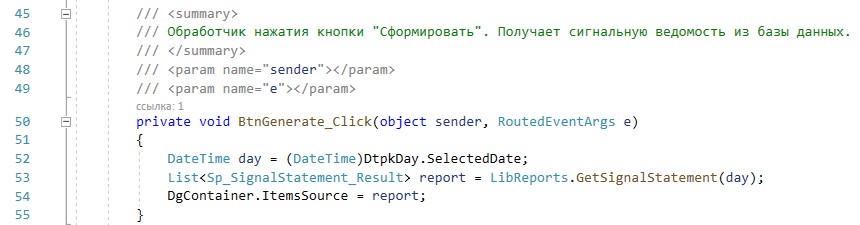


Рисунок Б.28 – Код обработчика нажатия кнопки «Сформировать» на странице «Сигнальная ведомость»

Приложение В

Скрипт базы данных

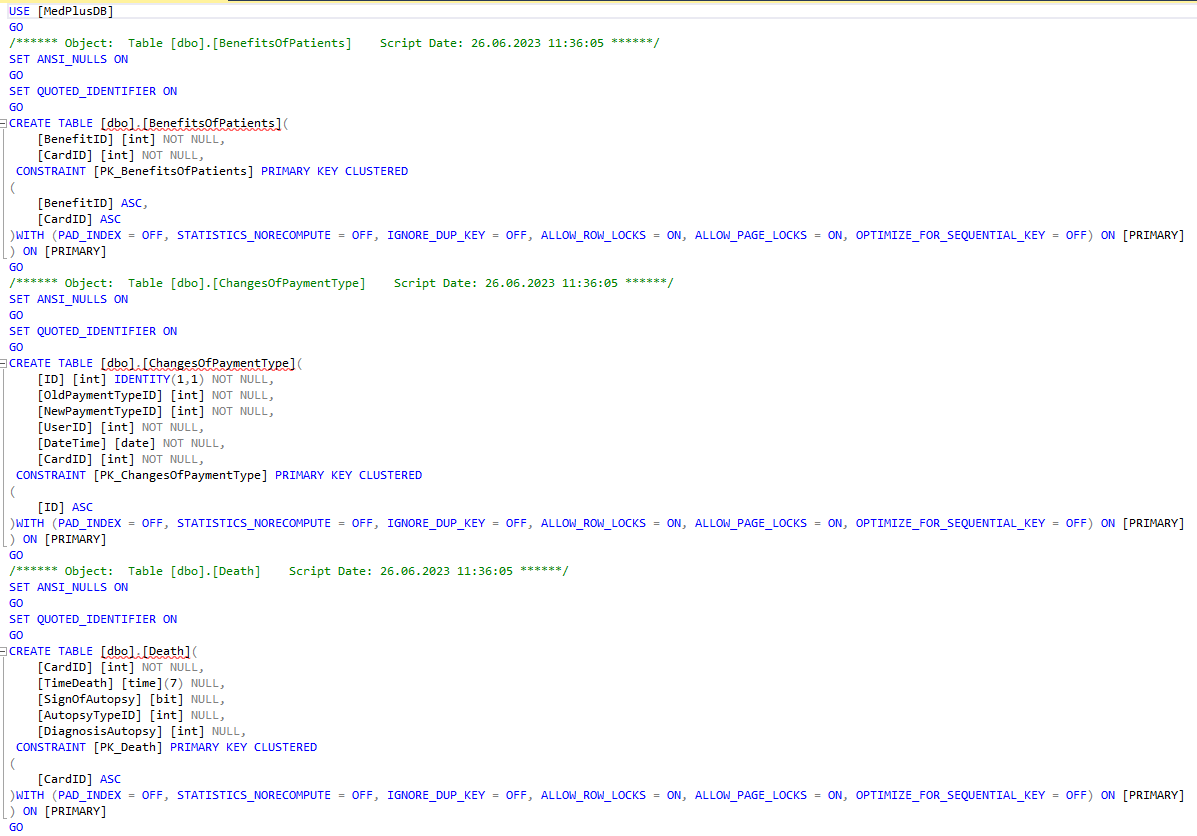


Рисунок В.1

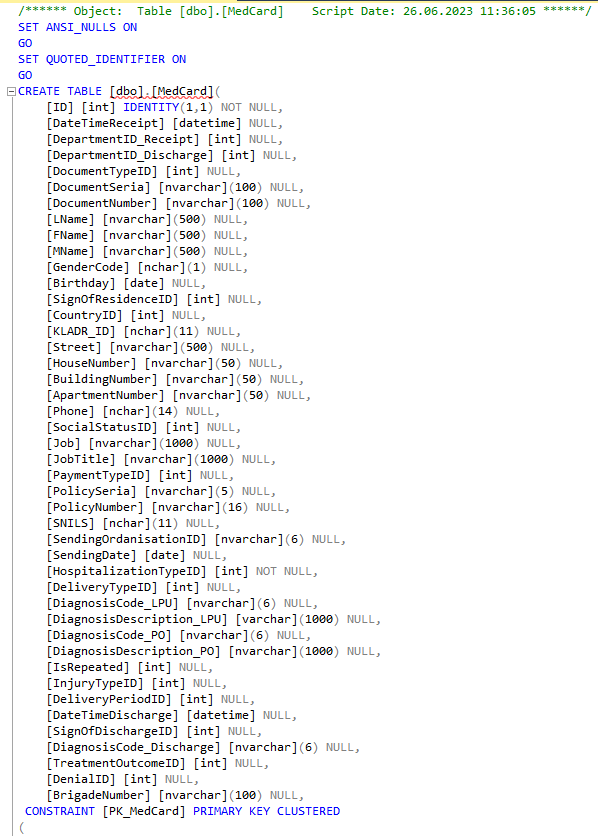


Рисунок В.2



Рисунок В.3

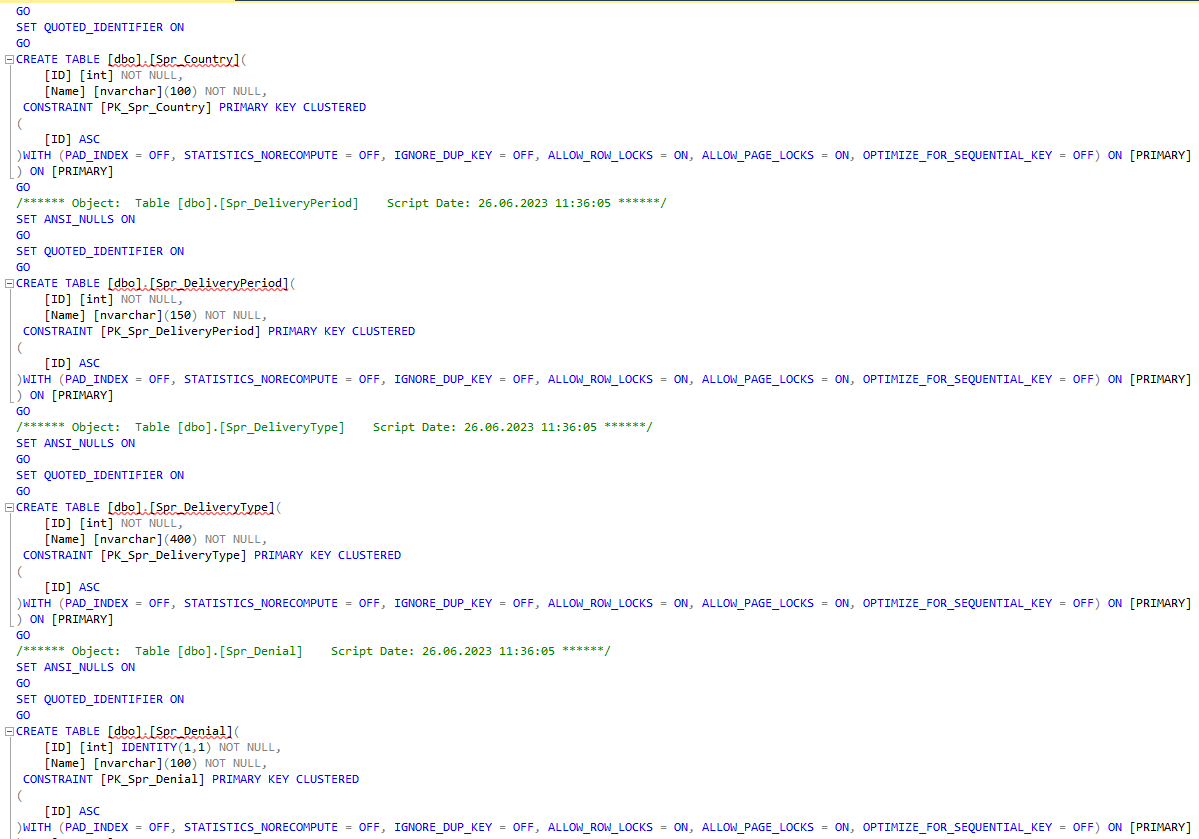


Рисунок В.4

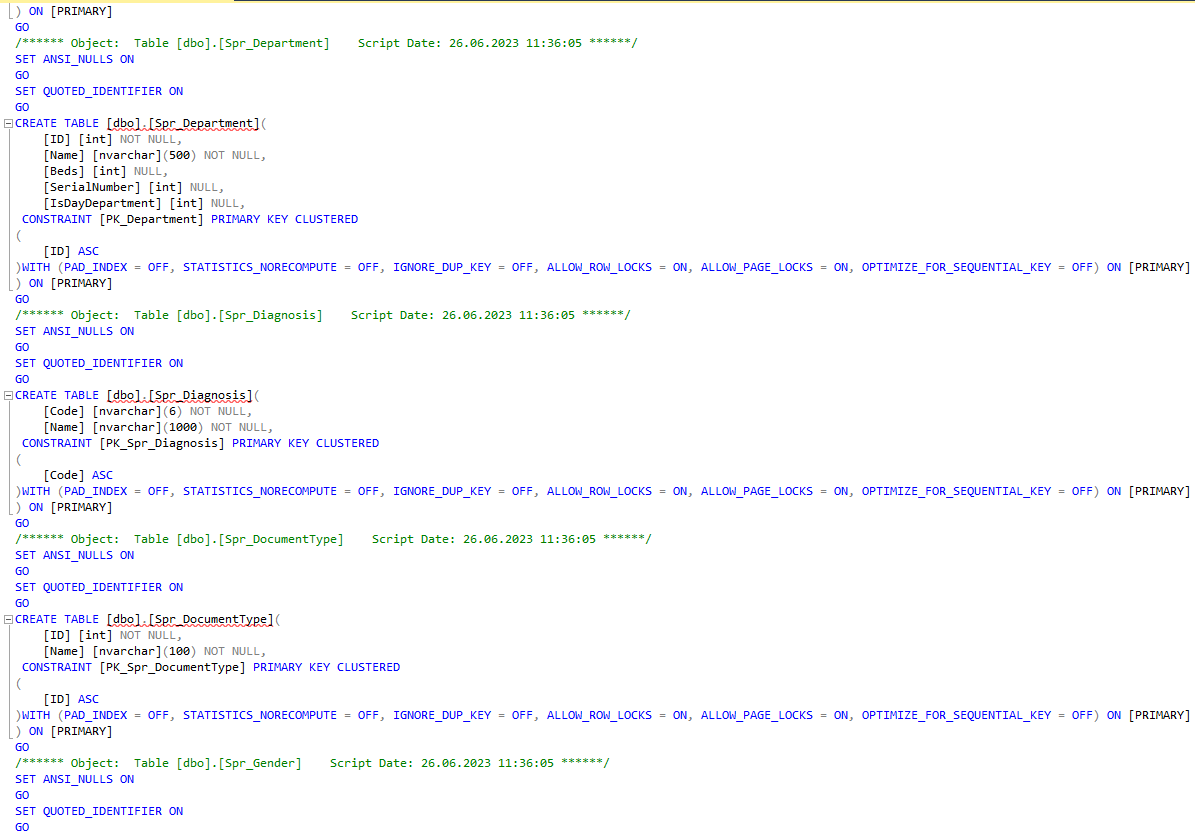


Рисунок В.5



Рисунок В.6



Рисунок В.7



Рисунок В.8

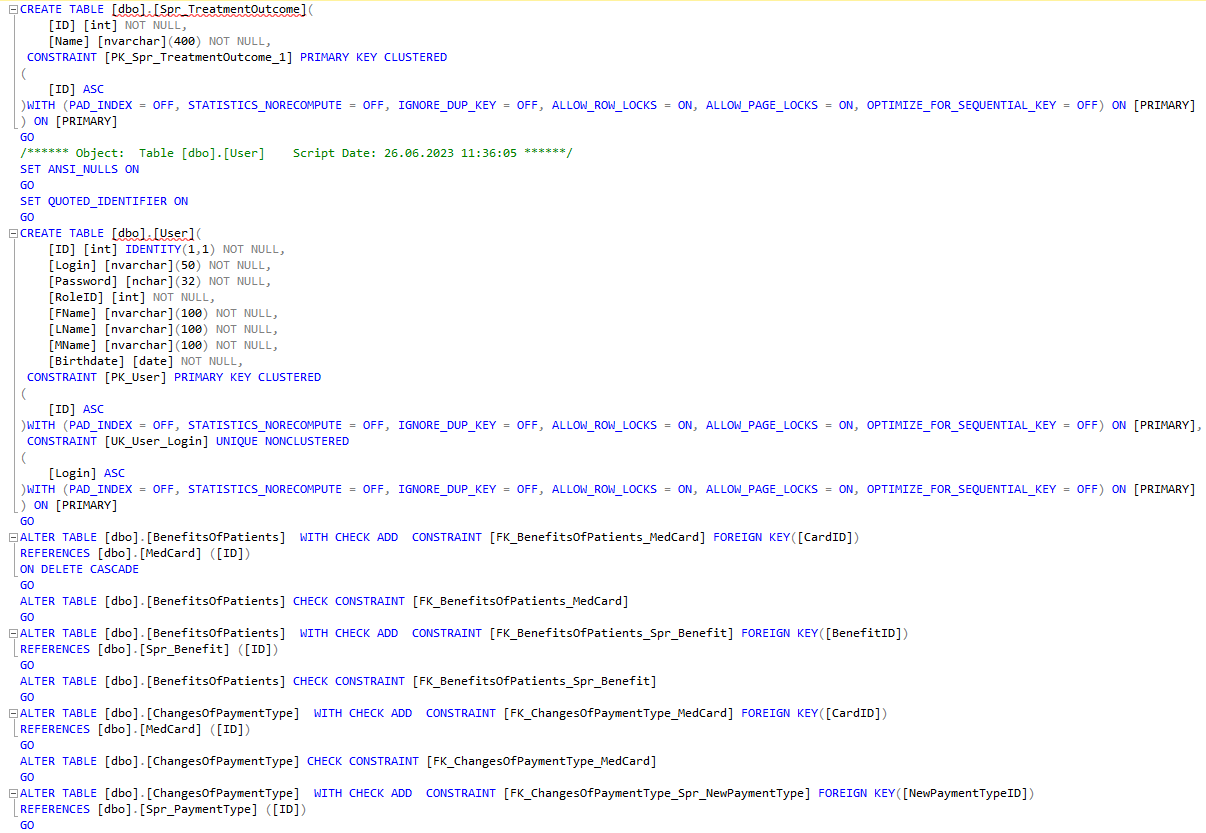


Рисунок В.9

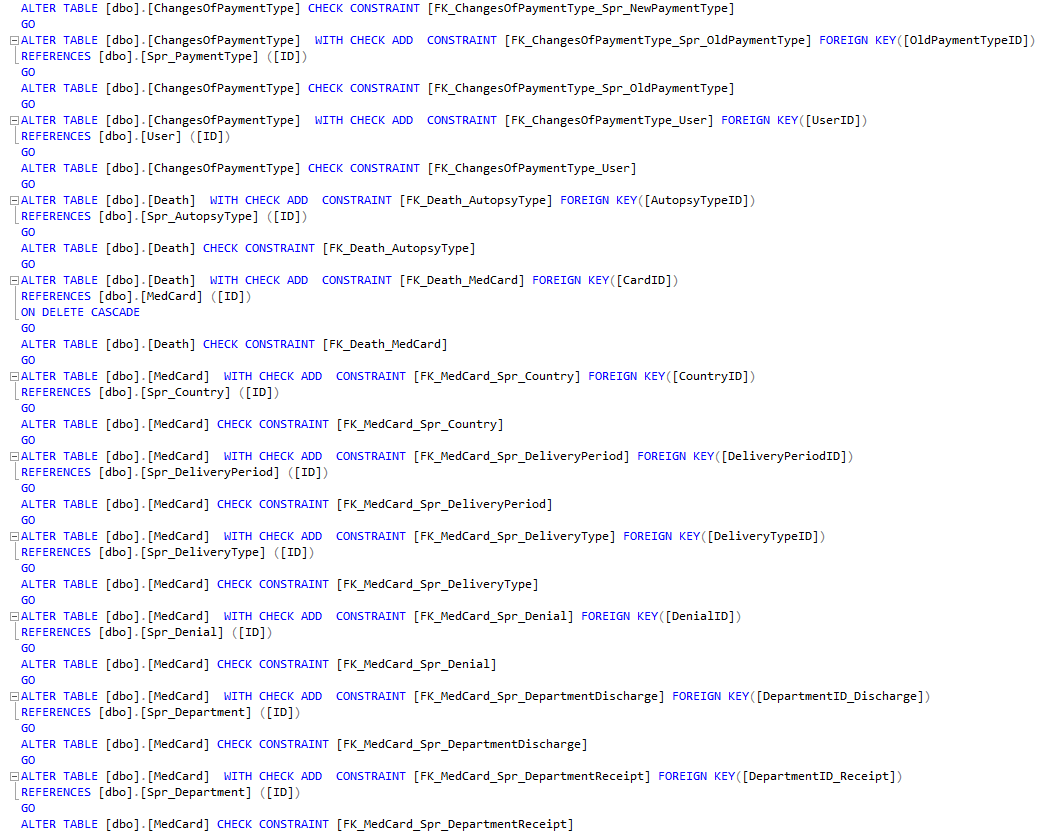


Рисунок В.10



Рисунок В.11

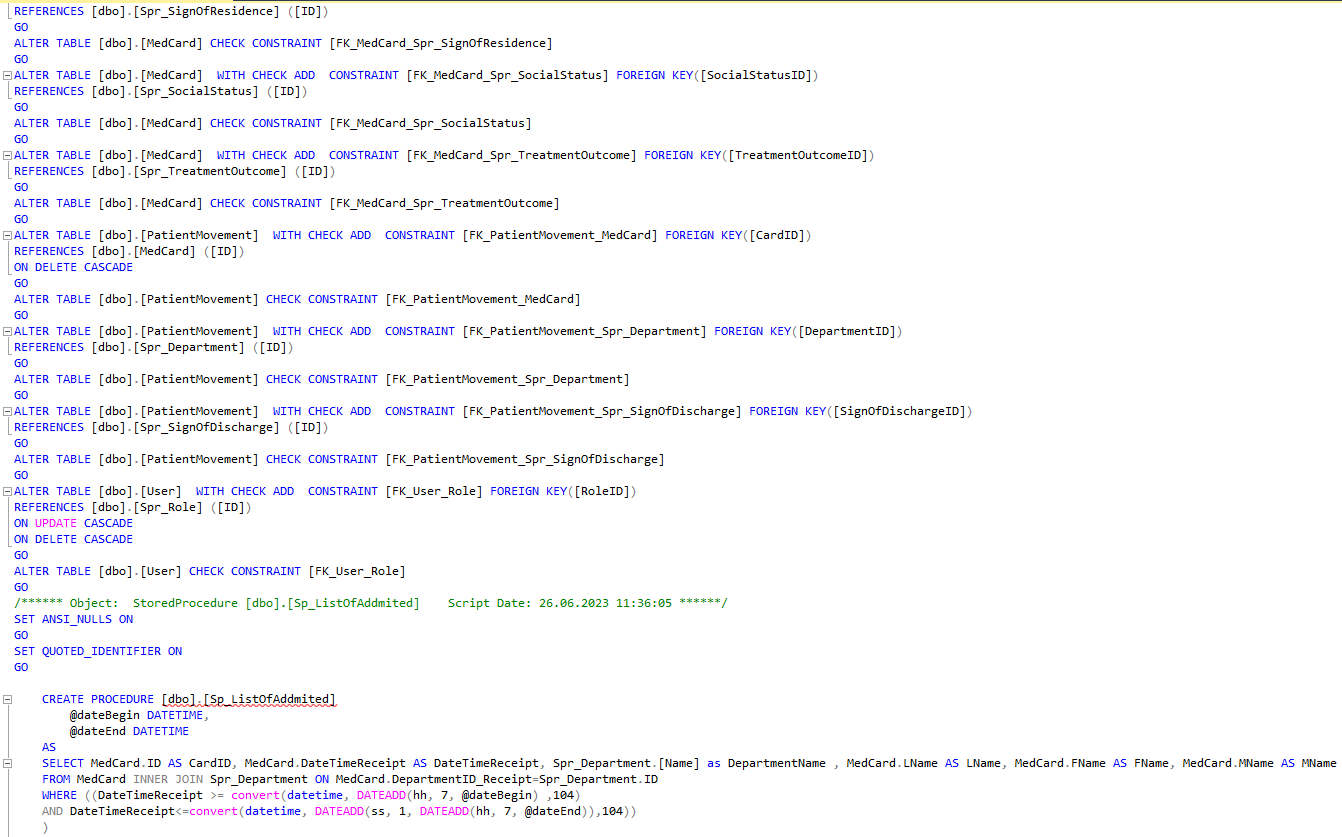


Рисунок В.12

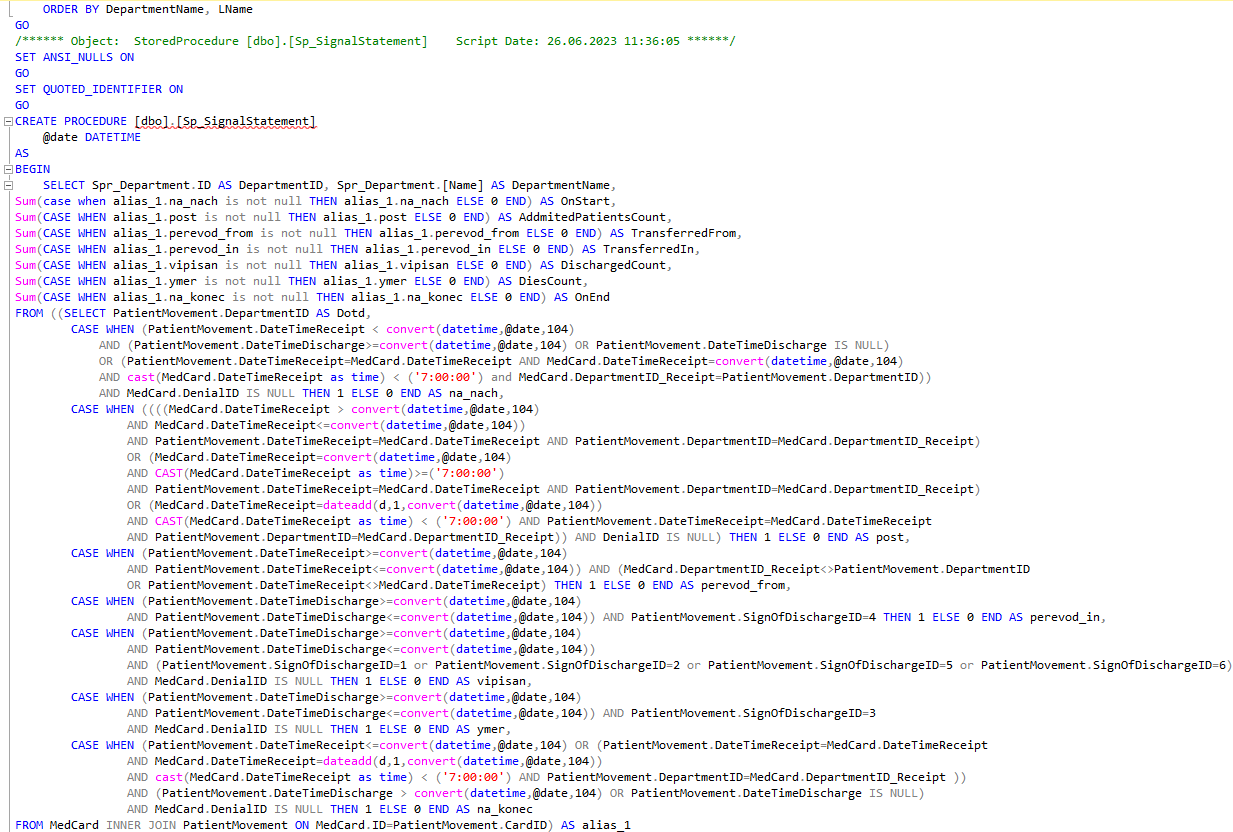


Рисунок В.13

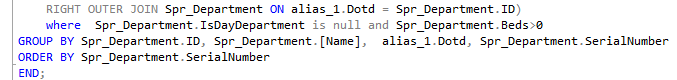


Рисунок В.14

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

|  |  |
| --- | --- |
| **Test Case #** | TC\_Module\_Patient\_1 |
| **Приоритет теста** | Высокий |
| **Название тестирования/Имя** | Попытка регистрации пациента с заполненными обязательными полями |
| **Резюме испытания** | Необходимо определить реакцию системы на попытку регистрации пациента с заполненными 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. Заполнить обязательные поля  5. Нажать кнопку сохранить |
| **Данные тестирования** | Логин: rom\_rad  Пароль: 123 |
| **Ожидаемый результат** | После попытки регистрации пациента с заполненными обязательными полями система сохранит карту в БД и выведет сообщение об успешной регистрации пациента |
| **Фактический результат** | Система сохранит карту в БД и выведет сообщение об успешной регистрации пациента |
| **Предпосылки** |  |
| **Постусловия** |  |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** |  |

Таблица Г.4

|  |  |
| --- | --- |
| **Test Case #** | TC\_Integration\_Patient\_PatientMovements\_1 |
| **Приоритет теста** | Средний |
| **Название тестирования/Имя** | Проверка взаимодействия модулей пациента и движений |
| **Резюме испытания** | Необходимо проверить корректность данных при открытии движений пациента. Должны быть доступны только движения выбранного пациента. |
| **Шаги тестирования** | 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 |
| **Данные тестирования** | Логин: rom\_rad  Пароль: 123  Номер медкарты: 3 |
| **Ожидаемый результат** | В списке движений все движения имеют значение 3 для поля «Номер карты» |
| **Фактический результат** | В списке движений все движения имеют значение 3 для поля «Номер карты» |
| **Предпосылки** |  |
| **Постусловия** |  |
| **Статус (Pass/Fail)** | Pass |
| **Комментарии** |  |