Министерство науки и высшего образования РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

**«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Институт космических и информационных технологий

Кафедра вычислительной техники

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
| Информационная система «Страховая компания» |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Руководитель |  |  |  |  | В.С. Васильев |
|  |  | подпись, дата |  | инициалы, фамилия |
|  |  |  |  |  |  |
| Студент | КИ22-06Б, 032215626 |  |  |  | В.А. Елисеев |
| номер группы, зачетной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2024

**РЕФЕРАТ**

Курсовой проект состоит из \_ страницы текста, \_ таблиц, \_ рисунков, \_ использованных источников и \_ приложения.

Проектирование программного обеспечения, информационная система, ICONIX, объектно-ориентированное программирование.

Целью проекта является освоение навыков проектирования программного обеспечения и развитие умений объектно-ориентированного программирования, разработки приложений с графическим интерфейсом.

Курсовой проект заключается в проектировании и реализации программного обеспечения, моделирующего информационную систему, которая позволяет управлять заключением договоров, а также фиксировать данные о клиентах и агентах.

В первой главе описана разработка спецификации требований, описаны прецеденты, а также приведены макеты интерфейса.

Во второй главе описано проектирование диаграмм пригодности, последовательности, классов, а также ER-диаграммы.

В третьей главе описаны особенности программной реализации, инструкция по сборке проекта, а также инструкция по эксплуатации.

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc168076286)

[1 Разработка спецификации требований 5](#_Toc168076287)

[1.1 Постановка задачи 5](#_Toc168076288)

[1.2 Выявление ролей и функций, диаграмма прецедентов 5](#_Toc168076289)

[1.3 Прецедент «Вход в систему» 7](#_Toc168076290)

[1.4 Прецедент «Добавление пользователя» 8](#_Toc168076291)

[1.5 Прецедент «Редактирование пользователя» 9](#_Toc168076292)

[1.6 Прецедент «Заключение договора» 10](#_Toc168076293)

[1.7 Прецедент «Перезаключение договора» 11](#_Toc168076294)

[1.8 Прецедент «Просмотр статистики» 13](#_Toc168076295)

[1.9 Прецедент «Подтверждение договора» 14](#_Toc168076296)

[1.10 Прецедент «Установка тарифной ставки» 16](#_Toc168076297)

[1.11 Прецедент «Просмотр заключённых договоров» 17](#_Toc168076298)

[1.12 Описание форматов данных 18](#_Toc168076299)

[2 Объектно-ориентированное проектирование 20](#_Toc168076300)

[2.1 Диаграммы пригодности и последовательности для прецедента «Вход в систему» 20](#_Toc168076301)

[2.2 Диаграммы пригодности и последовательности для прецедента «Добавление пользователя» 21](#_Toc168076302)

[2.3 Диаграммы пригодности и последовательности для прецедента «Редактирование пользователя» 22](#_Toc168076303)

[2.4 Диаграммы пригодности и последовательности для прецедента «Заключение договора» 24](#_Toc168076304)

[2.5 Диаграммы пригодности и последовательности для прецедента «Перезаключение договора» 25](#_Toc168076305)

[2.6 Диаграммы пригодности и последовательности для прецедента «Просмотр статистики» 26](#_Toc168076306)

[2.7 Диаграммы пригодности и последовательности для прецедента «Подтверждение договора» 28](#_Toc168076307)

[2.8 Диаграммы пригодности и последовательности для прецедента «Установка тарифной ставки» 29](#_Toc168076308)

[2.9 Диаграммы пригодности и последовательности для прецедента «Просмотр заключённых договоров» 30](#_Toc168076309)

[2.10 ER-диаграмма 31](#_Toc168076310)

[2.11 Диаграмма классов 32](#_Toc168076311)

[3 Объектно-ориентированное программирование 36](#_Toc168076312)

[3.1 Реализация 36](#_Toc168076313)

[3.2 Сборка и запуск 37](#_Toc168076314)

[3.3 Тестирование 37](#_Toc168076315)

[3.4 Инструкция 40](#_Toc168076316)

[Заключение 41](#_Toc168076317)

[Приложение А 43](#_Toc168076318)

[Приложение Б 45](#_Toc168076319)

[Список используемых источников 42](#_Toc168076320)

ВВЕДЕНИЕ

Сегодня страхование является неотъемлемой частью жизни современного человека. Поэтому растёт число компаний, оказывающих данные услуги. И для того, чтобы упростить и автоматизировать задачу составления договоров, необходимо разработать информационную систему.

Целью работы является разработка приложения «Страховая компания» с открытым исходным кодом, позволяющего автоматизировать операции: заключения договоров, расчёта заработной платы, регистрации новых сотрудников и пр.

1. Разработка спецификации требований
   1. Постановка задачи

В страховую компанию обращаются клиенты с целью заключения договора о страховании. В зависимости от принимаемых на страхование объектов договор заключается по определенному виду страхования (страхование автотранспорта от угона, страхование домашнего имущества, добровольное медицинское страхование). При заключении договора фиксируются: дата заключения, страховая сумма, вид страхования, тарифная ставка. Договоры заключают страховые агенты. Помимо информации об агентах (фамилия, имя, отчество, адрес, телефон) нужно хранить филиал, в котором они работают. Необходимо иметь возможность рассчитывать заработную плату агентам. Заработная плата составляет некоторый процент от страховой суммы. Процент зависит от вида страхования, по которому заключен договор.

* 1. Выявление ролей и функций, диаграмма прецедентов

Были выявлены следующие роли: администратор, агент, бухгалтер.

На рисунке 1, рисунке 2, рисунке 3 представлены диаграммы прецедентов для этих ролей соответственно.

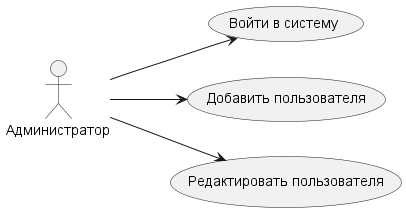


Рисунок 1 – Диаграмма прецедентов (Администратор)

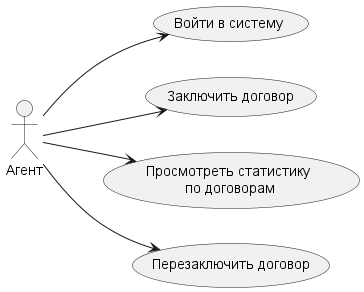


Рисунок 2 – Диаграмма прецедентов (Агент)

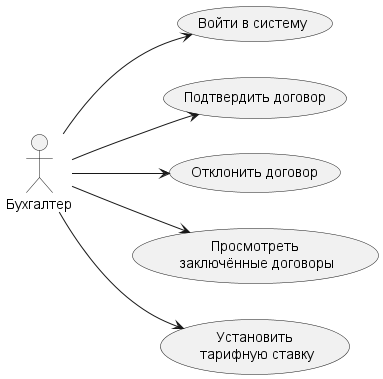


Рисунок 3 – Диаграмма прецедентов (Бухгалтер)

В приложении А представлена диаграммы потока экранов.

## Прецедент «Вход в систему»

Макет интерфейса для прецедента представлен на рисунке 4.

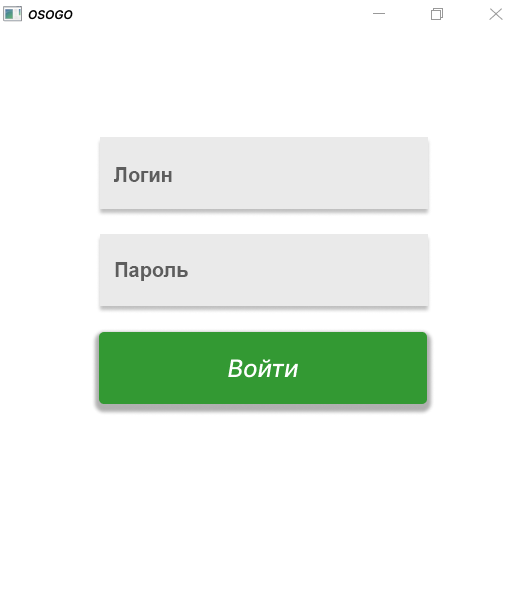


Рисунок 4 – Окно входа в программу

**Роль**: агент, бухгалтер, администратор.

**Предусловие**: открыто «Окно входа в программу» (Рисунок 4).

**Основной сценарий**:

1. Ввести логин и пароль;
2. Нажать кнопку «Войти».

**Постусловие**: если данные корректны, то откроется домашнее окно для соответствующей роли.

**Альтернативный сценарий**: введены некорректные данные.

**Постусловие**: выведено сообщение об ошибке.

* 1. Прецедент «Добавление пользователя»

Макет интерфейса для прецедента представлен на рисунке 5.

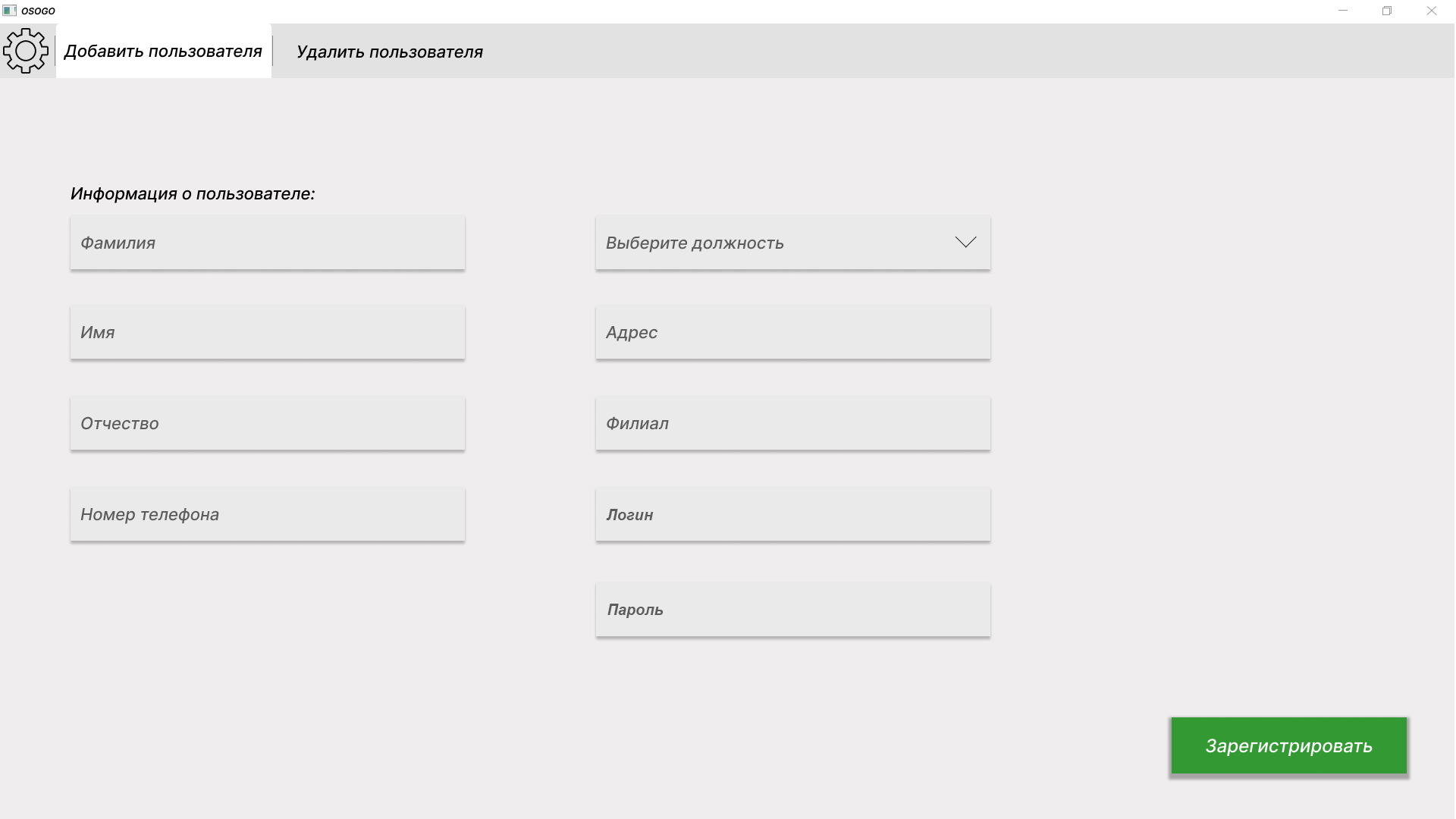


Рисунок 5 – Вкладка добавления пользователя

**Роль**: администратор.

**Предусловие**: открыта вкладка добавления пользователя.

**Основной сценарий**:

1. В «Окне добавления пользователей» (Рисунок 5) ввести ФИО, телефон, должность, адрес, филиал, логин и пароль;
2. Нажать кнопку «Зарегистрировать».

**Постусловие**: если все обязательные поля заполнены и введены корректные данные, то соответствующая запись о пользователе будет добавлена в базу данных.

**Альтернативный сценарий**: введены некорректные данные или заполнены не все обязательные поля.

**Постусловие**: выведено сообщение об ошибке.

* 1. Прецедент «Редактирование пользователя»

Макет интерфейса для прецедента представлен на рисунке 6 и рисунке 7.

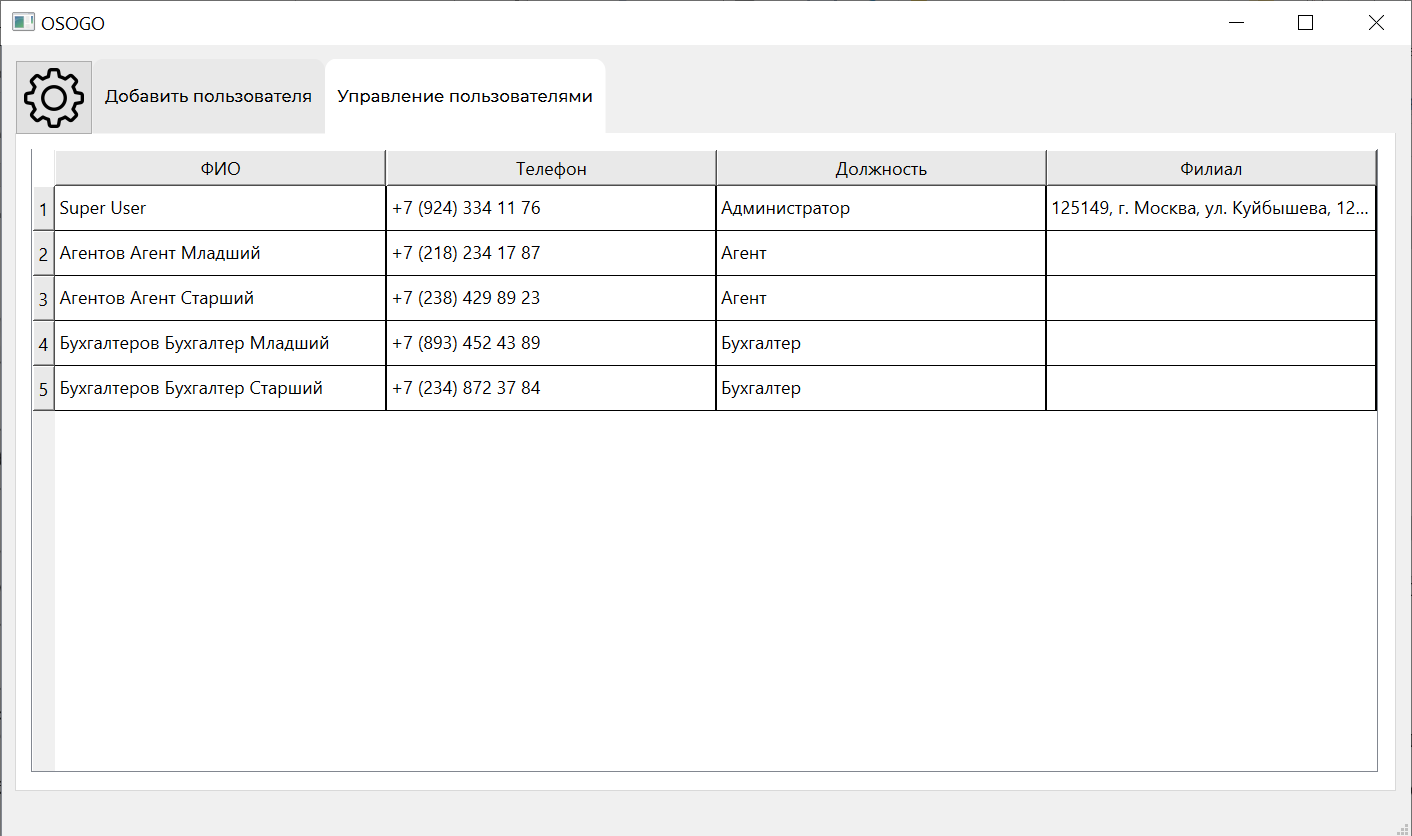


Рисунок 6 – Вкладка управления пользователями

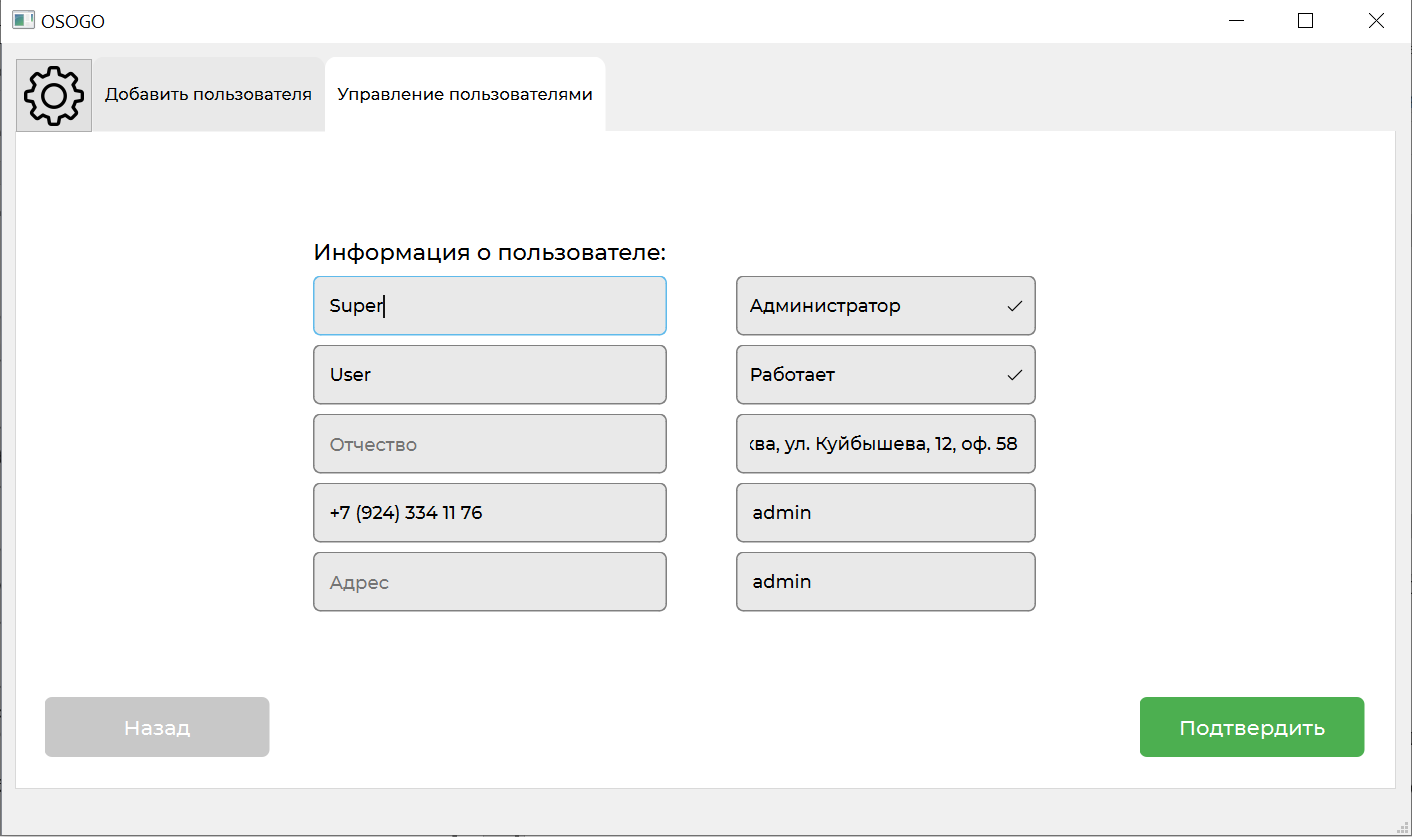


Рисунок 7 – Окно редактирования пользователя

**Роль**: администратор.

**Предусловие**: открыта «Вкладка управления пользователями» (рисунок 6).

**Основной сценарий**:

1. Дважды нажать на необходимого пользователя;
2. В «Окне редактирования пользователя» (Рисунок 7) изменить данные пользователя;
3. Нажать кнопку «Подтвердить».

**Постусловие**: если все обязательные поля заполнены и введены корректные данные, то соответствующая запись о пользователе будет обновлена в базе данных.

**Альтернативный сценарий**: введены некорректные данные или заполнены не все обязательные поля.

**Постусловие**: выведено сообщение об ошибке.

**Альтернативный сценарий**: нажата кнопка «Назад».

**Постусловие**: открыто «Окно управления пользователями», данные выбранного пользователя не изменены.

* 1. Прецедент «Заключение договора»

Макет интерфейса для прецедента представлен на рисунке 8.

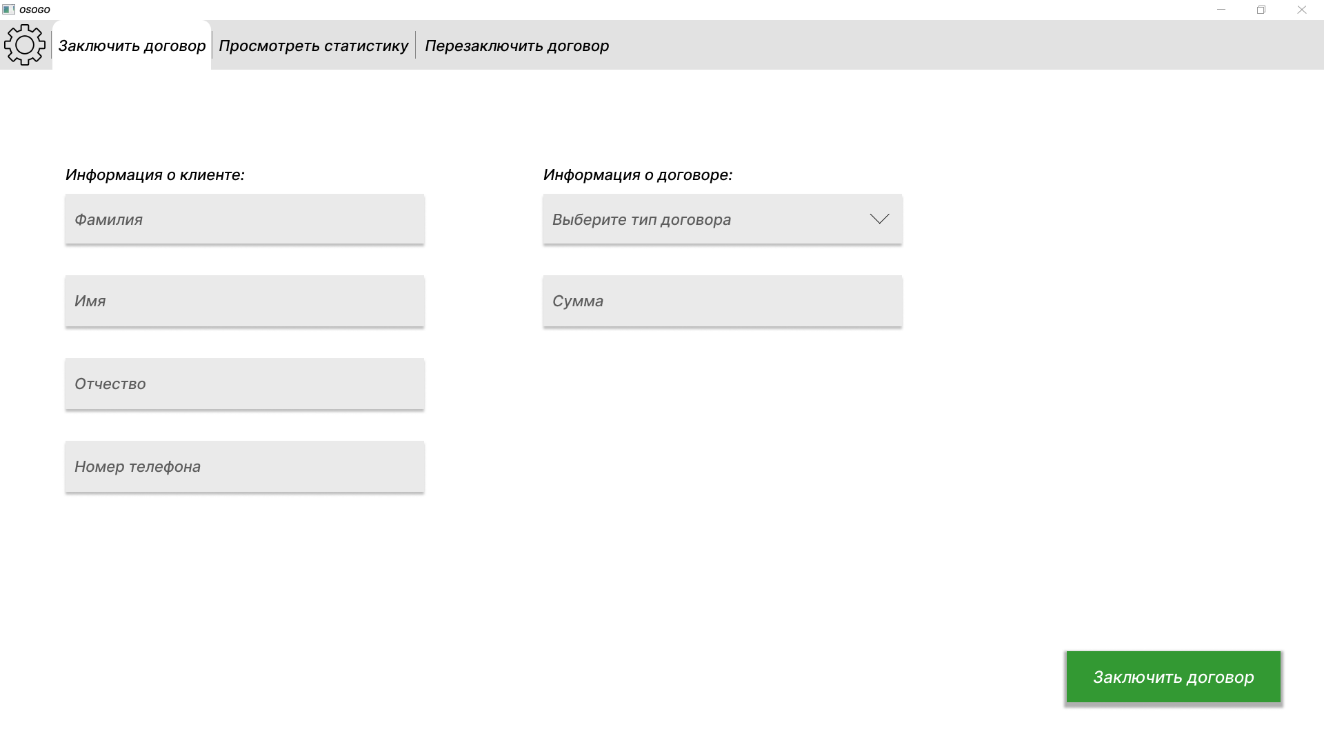


Рисунок 8 – Окно заключения договора

**Роль**: агент.

**Предусловие**: открыто «Окно заключения договора» (рисунок 8).

**Основной сценарий**:

1. Ввести информацию о клиенте и договоре;
2. Нажать кнопку «Заключить договор».

**Постусловие**: если все обязательные поля заполнены и введены корректные данные, то соответствующая запись появится в базе данных.

**Альтернативный сценарий**: введены некорректные данные или заполнены не все обязательные поля.

**Постусловие**: выведено сообщение об ошибке.

* 1. Прецедент «Перезаключение договора»

Макет интерфейса для прецедента представлен на рисунке 9 и рисунке 10.

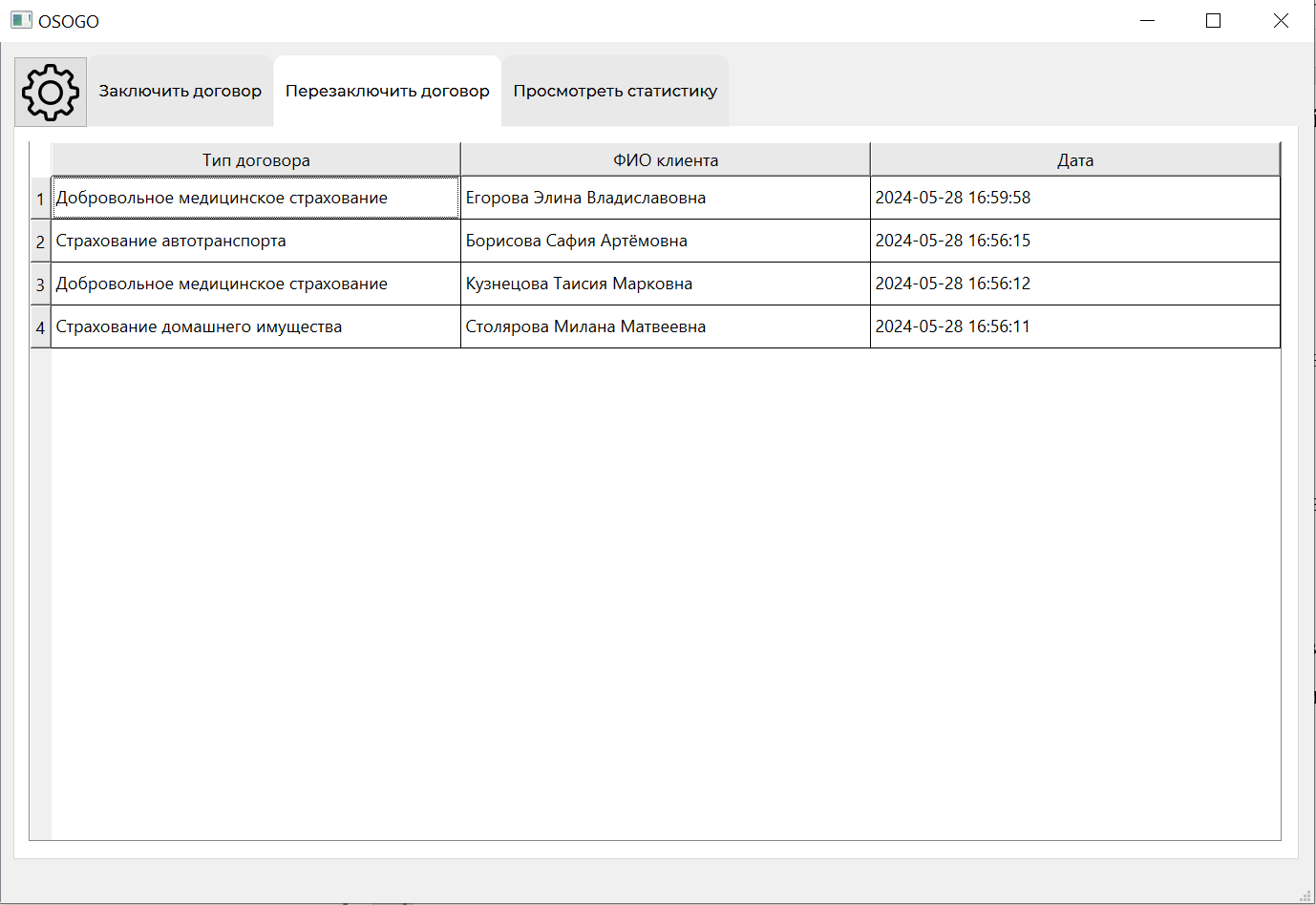


Рисунок 9 – Вкладка перезаключения договоров

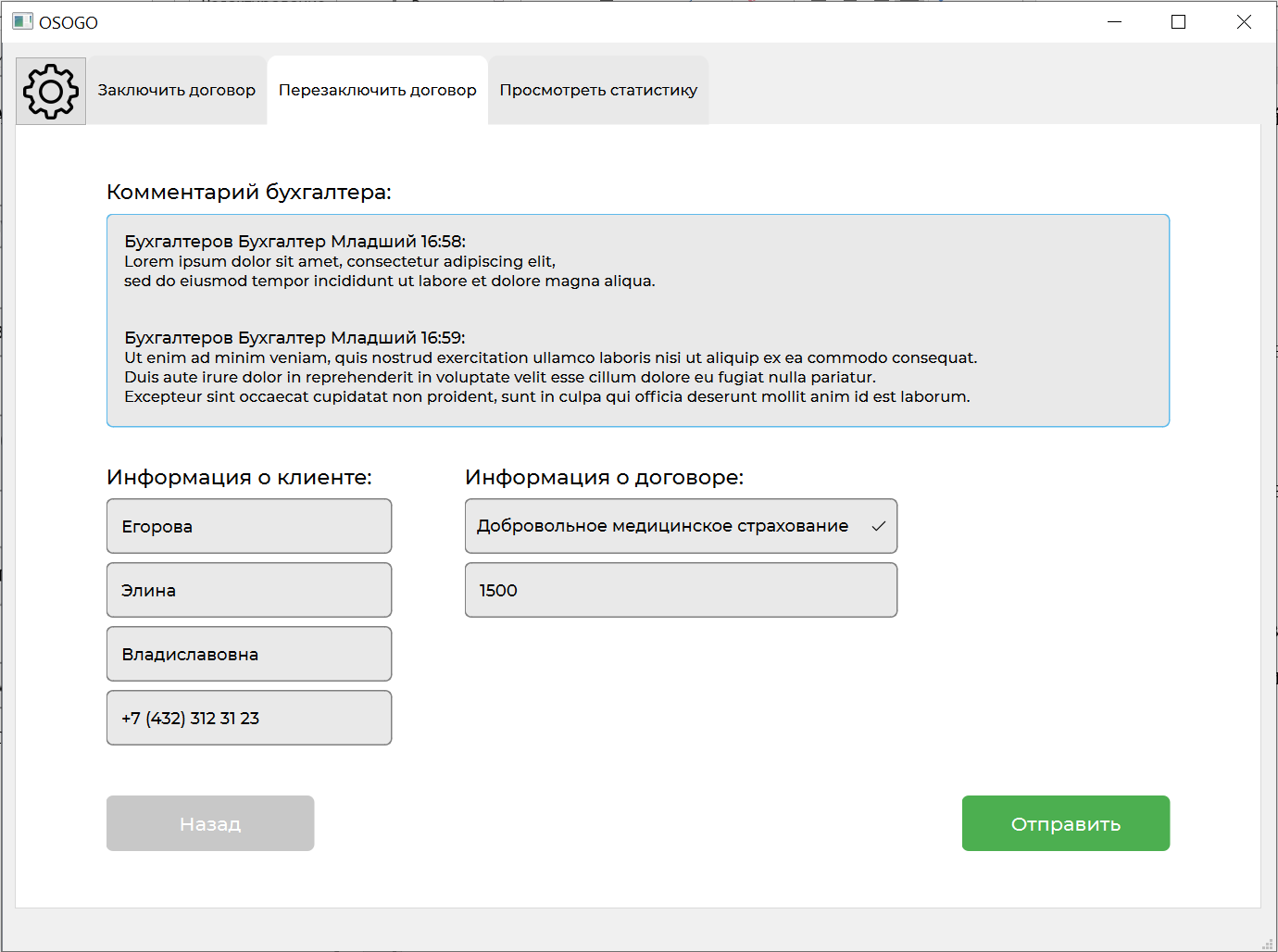


Рисунок 10 – Окно перезаключения договора

**Роль**: агент.

**Предусловие**: открыта «Вкладка перезаключения договора» (рисунок 9).

**Основной сценарий**:

1. Дважды нажать на какой-либо договор;
2. Ввести информацию о клиенте и договоре;
3. Нажать кнопку «Отправить».

**Постусловие**: если все обязательные поля заполнены и введены корректные данные, то соответствующая запись обновится в базе данных.

**Альтернативный сценарий**: введены некорректные данные или заполнены не все обязательные поля.

**Постусловие**: выведено сообщение об ошибке.

**Альтернативный сценарий**: нажата кнопка «Назад».

**Постусловие**: открыта «Вкладка перезаключения договоров», данные о договоре не изменены.

* 1. Прецедент «Просмотр статистики»

Макет интерфейса для прецедента представлен на рисунке 11.

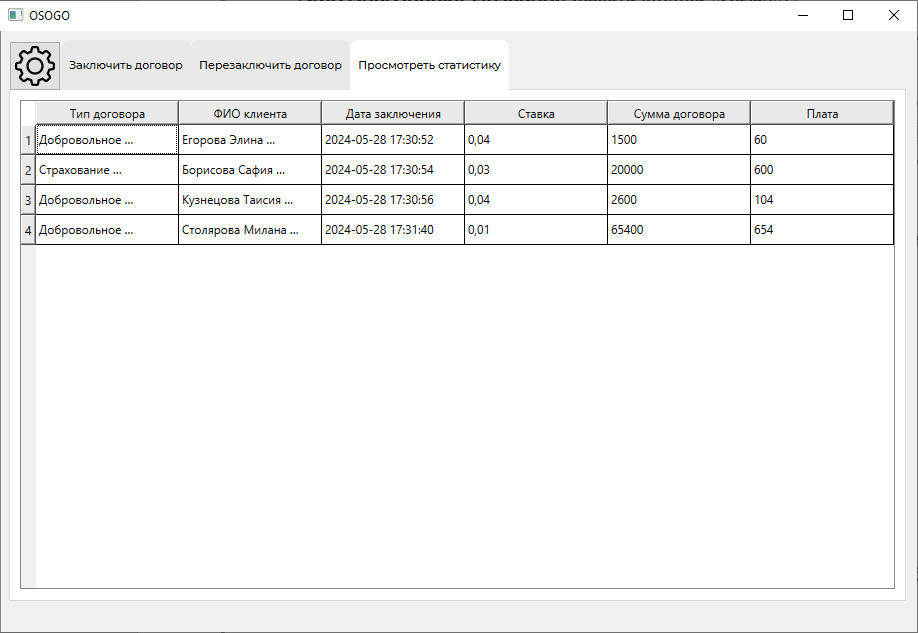


Рисунок 11 – Окно просмотра статистики

**Роль**: агент.

**Предусловие**: открыто «Окно просмотра статистики» (рисунок 11).

**Основной сценарий**:

1. Нажать на заголовок столбца для сортировки данных;
2. Просмотреть необходимую информацию.

**Постусловие**: данные отсортированы.

* 1. Прецедент «Подтверждение договора»

Макет интерфейса для прецедента представлен на рисунке 12 и рисунке 13.

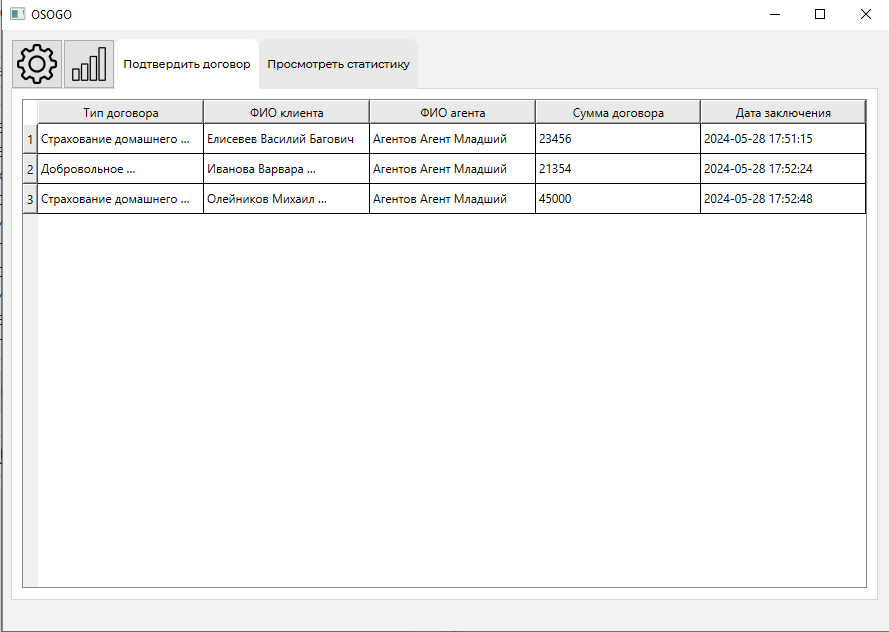


Рисунок 12 – Вкладка подтверждения договоров

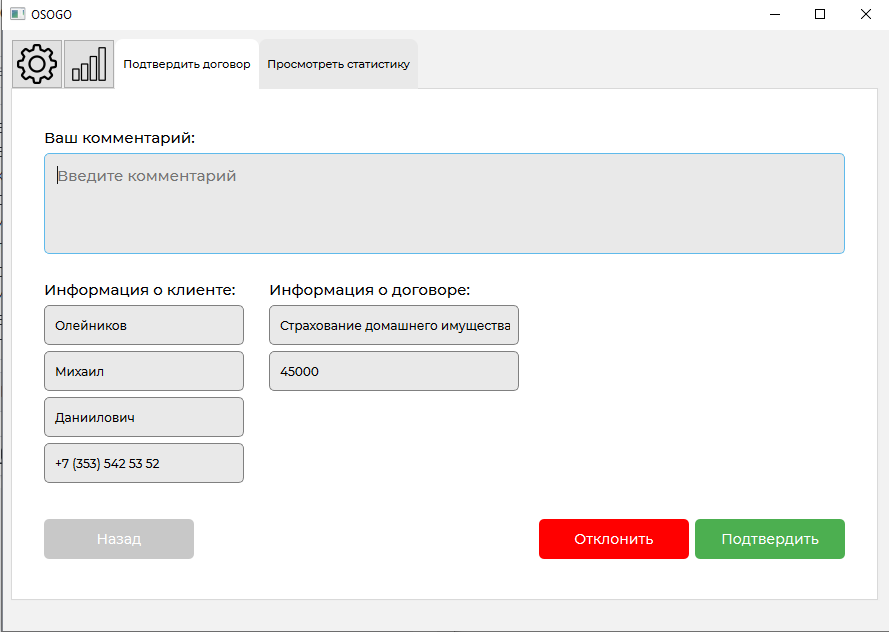


Рисунок 13 – Окно подтверждения договора

**Роль**: Бухгалтер.

**Предусловие**: открыта «Вкладка подтверждения договоров» (рисунок 12).

**Основной сценарий**:

1. Дважды нажать на какой-либо договор, откроется «Окно подтверждения договора» (рисунок 13);
2. Нажать на кнопку «Подтвердить».

**Постусловие**: соответствующая запись в базе данных обновлена, договор является подтверждённым.

* 1. Прецедент «Отклонение договора»

**Роль**: Бухгалтер.

**Предусловие**: открыта «Вкладка подтверждения договоров» (рисунок 12).

**Основной сценарий**:

1. Дважды нажать на какой-либо договор, откроется «Окно подтверждения договора» (рисунок 13);
2. Нажать на кнопку «Отклонить».

**Постусловие**: соответствующая запись в базе данных обновлена, договор является отклонённым.

* 1. Прецедент «Установка тарифной ставки»

Макет интерфейса для прецедента представлен на рисунке 14.

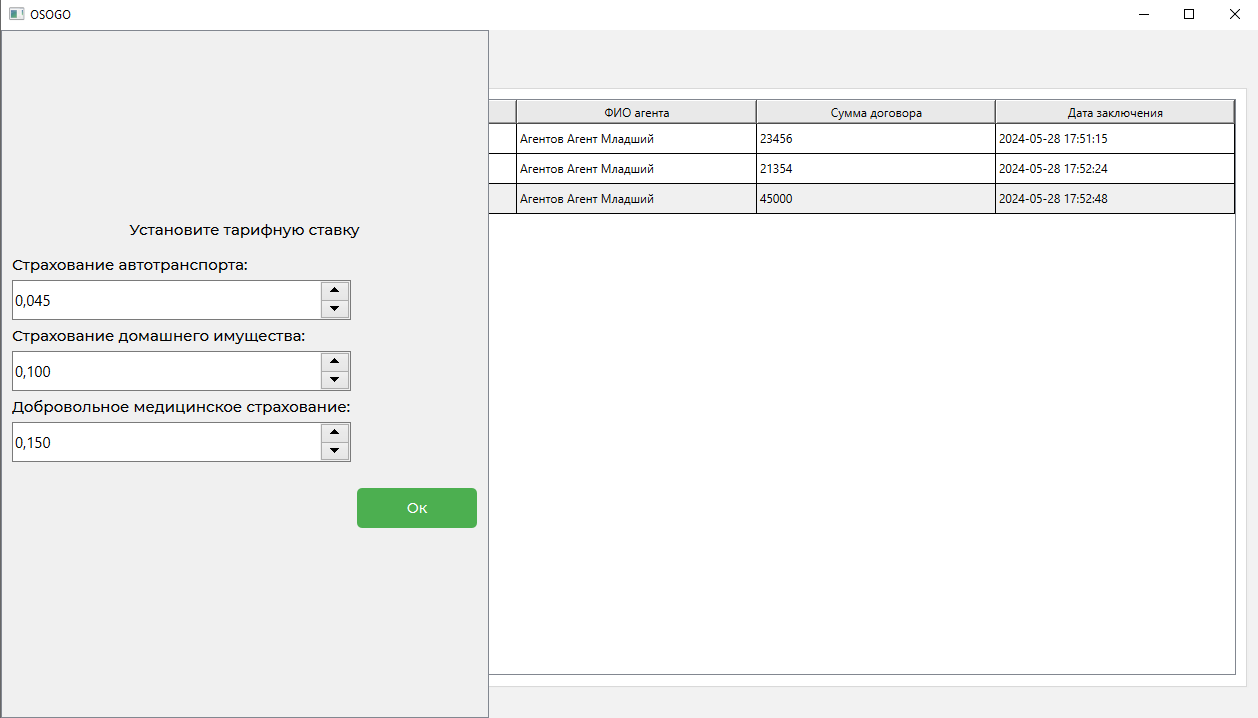


Рисунок 14 – Окно установки тарифной ставки

**Роль**: Бухгалтер.

**Предусловие**: открыто «Окно установки тарифной ставки» (рисунок 14).

**Основной сценарий**:

1. Указать тарифные ставки;
2. Нажать на кнопку «Ок».

**Постусловие**: значения тарифных ставок сохранены.

* 1. Прецедент «Просмотр заключённых договоров»

Макет интерфейса для прецедента представлен на рисунке 15.

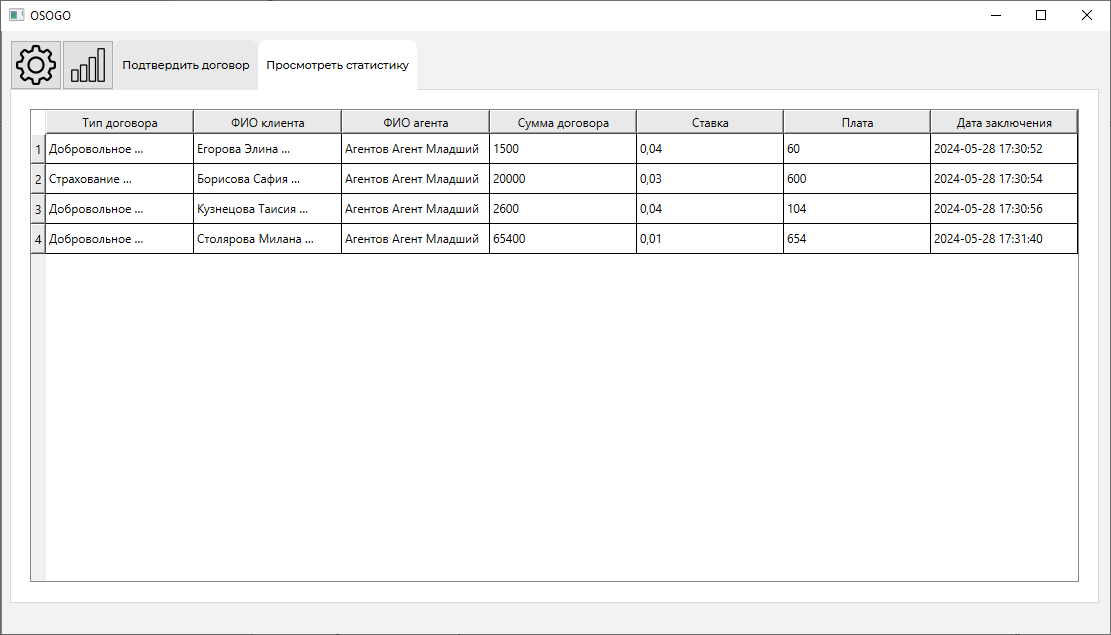


Рисунок 15 – Вкладка просмотра статистики

**Роль**: Бухгалтер.

**Предусловие**: открыта «Вкладка просмотра статистики» (рисунок 15).

**Основной сценарий**:

1. Нажать на заголовок столбца для сортировки данных;
2. Просмотреть необходимую информацию.

**Постусловие**: данные отсортированы.

* 1. Описание форматов данных

Важной частью функционирования системы является хранение и обработка данных. Для этого используется SQLite база данных с таблицами, представленными на рисунках 16, 17, 18, 19, 20.

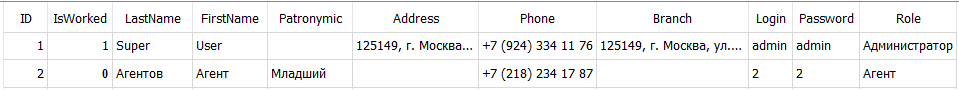


Рисунок 16 – Пример заполнения таблицы Employee



Рисунок 17 – Пример заполнения таблицы Contract

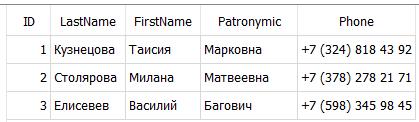


Рисунок 18 – Пример заполнения таблицы Client

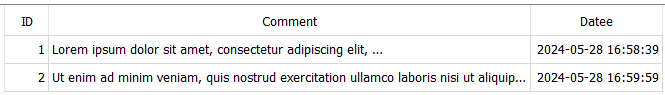


Рисунок 19 – Пример заполнения таблицы Comment

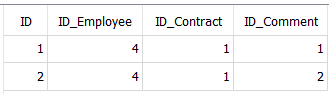


Рисунок 20 – Пример заполнения таблицы CommentsOn

2 **Объектно-ориентированное проектирование**

1. Диаграммы пригодности и последовательности для прецедента «Вход в систему»

На рисунке 21 представлена диаграмма пригодности для данного прецедента.

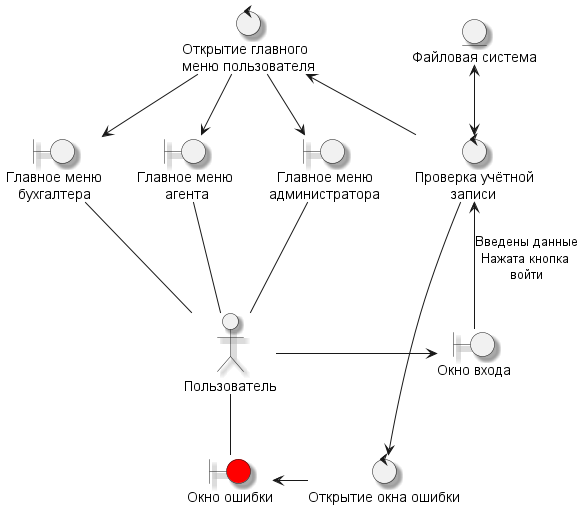


Рисунок 21 – «Вход в систему»

На рисунке 22 представлена диаграмма последовательности для этого прецедента.

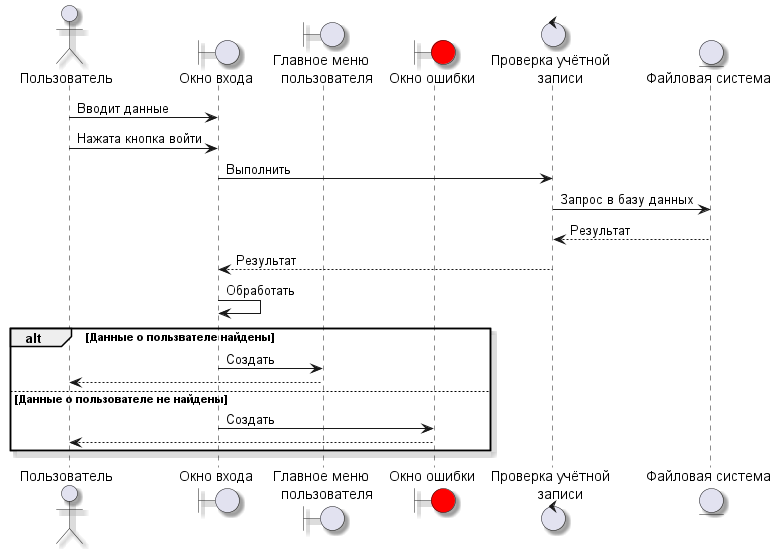


Рисунок 22 – «Вход в систему»

1. Диаграммы пригодности и последовательности для прецедента «Добавление пользователя»

На рисунке 23 представлена диаграмма пригодности для данного прецедента.

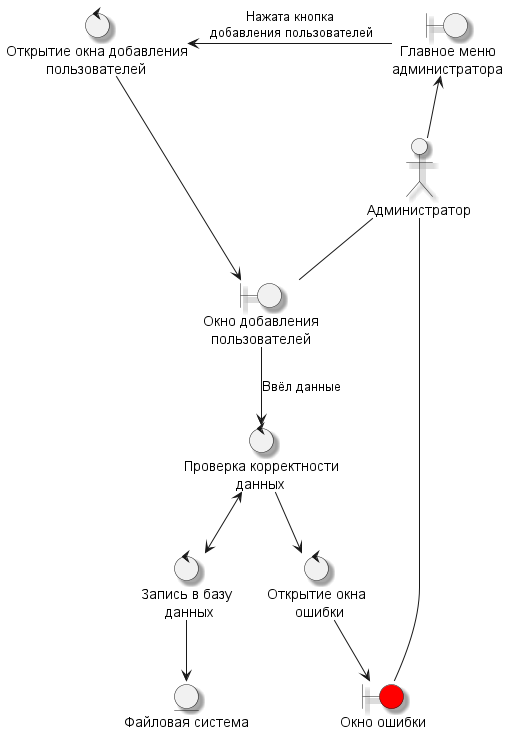


Рисунок 23 – «Добавление пользователя»

На рисунке 24 представлена диаграмма последовательности для этого прецедента.

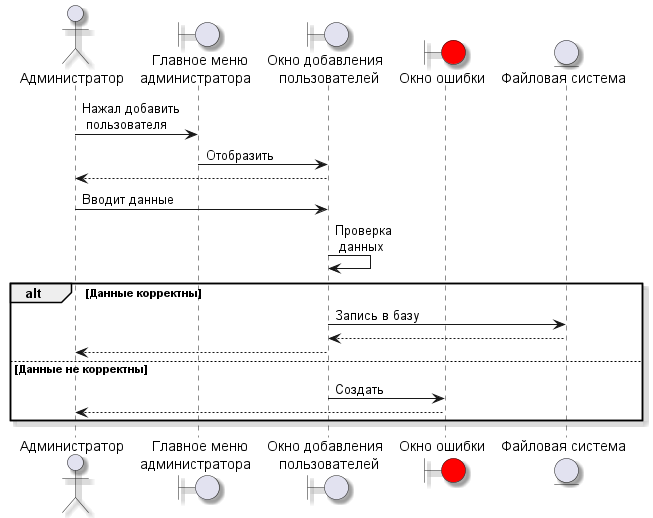


Рисунок 24 – «Добавление пользователя»

1. Диаграммы пригодности и последовательности для прецедента «Редактирование пользователя»

На рисунке 25 представлена диаграмма пригодности для данного прецедента.

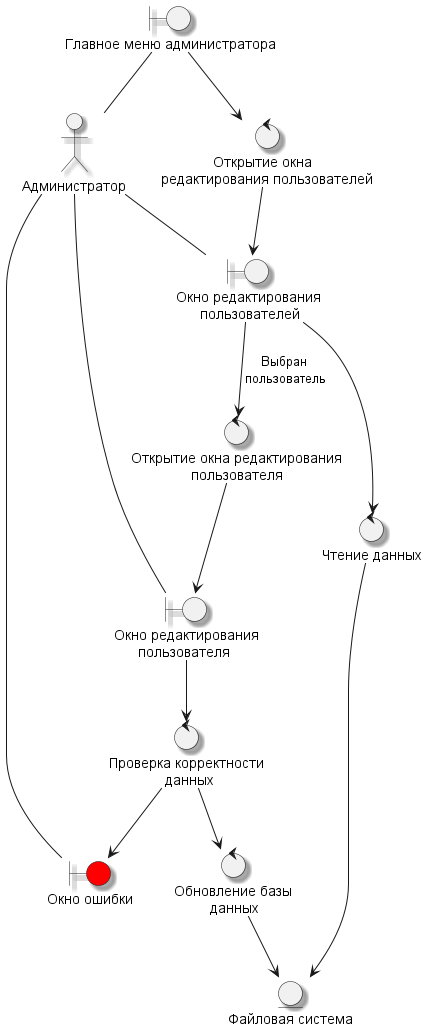


Рисунок 25 – «Редактирование пользователя»

На рисунке 26 представлена диаграмма последовательности для этого прецедента.

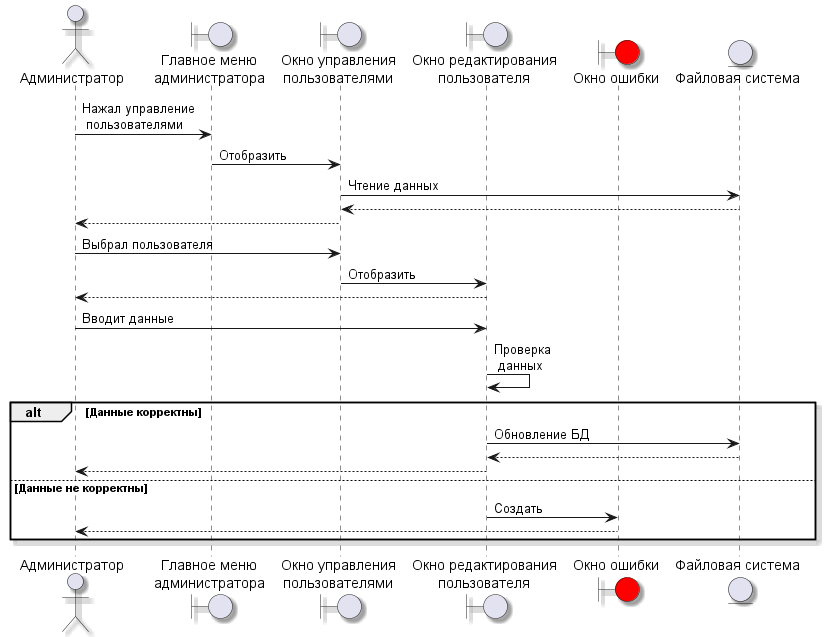


Рисунок 26 – «Редактирование пользователя»

1. Диаграммы пригодности и последовательности для прецедента «Заключение договора»

На рисунке 27 представлена диаграмма пригодности для данного прецедента.

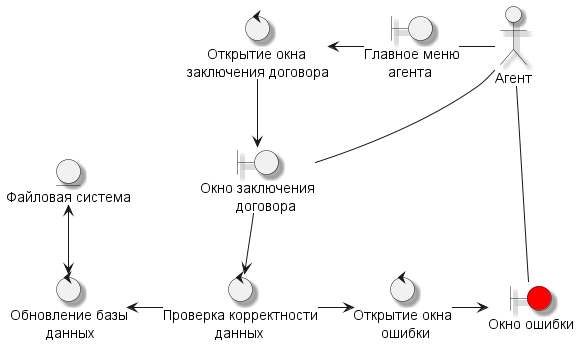


Рисунок 27 – «Заключение договора»

На рисунке 28 представлена диаграмма последовательности для этого прецедента.

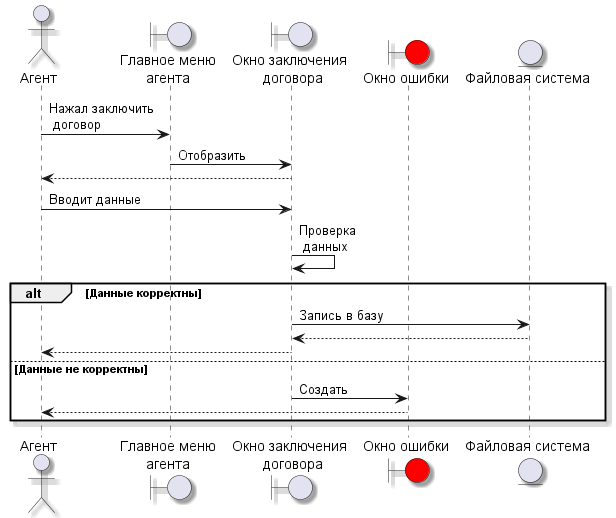


Рисунок 28 – «Заключение договора»

1. Диаграммы пригодности и последовательности для прецедента «Перезаключение договора»

На рисунке 29 представлена диаграмма пригодности для данного прецедента.

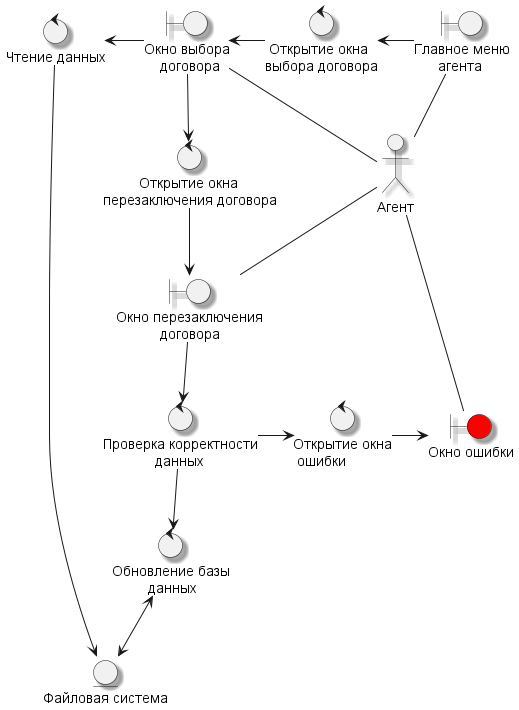


Рисунок 29 – «Перезаключение договора»

На рисунке 30 представлена диаграмма последовательности для этого прецедента.

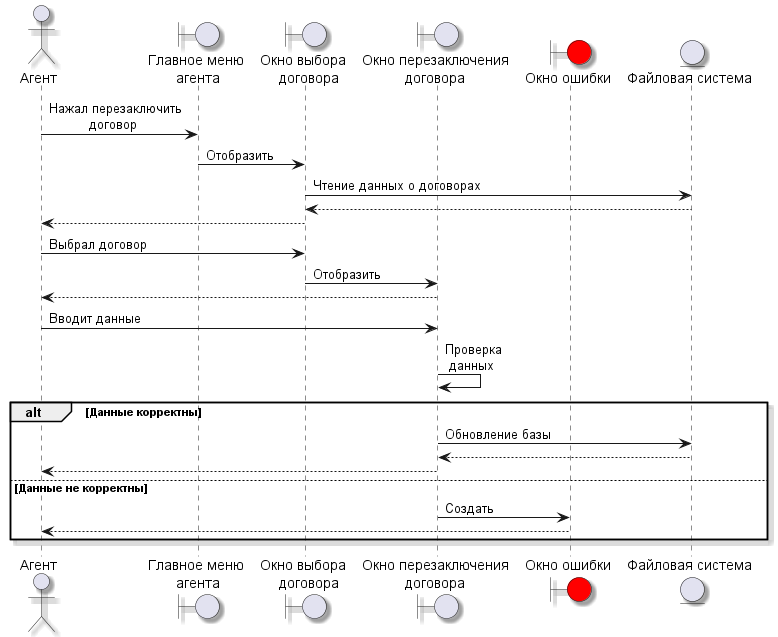


Рисунок 30 – «Перезаключение договора»

1. Диаграммы пригодности и последовательности для прецедента «Отклонение договора»

На рисунке 31 представлена диаграмма пригодности для данного прецедента.

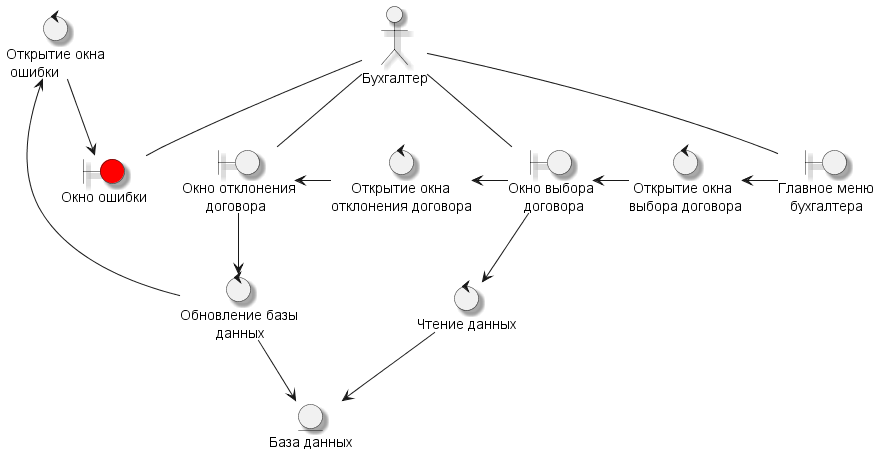


Рисунок 31 – «Отклонение договора»

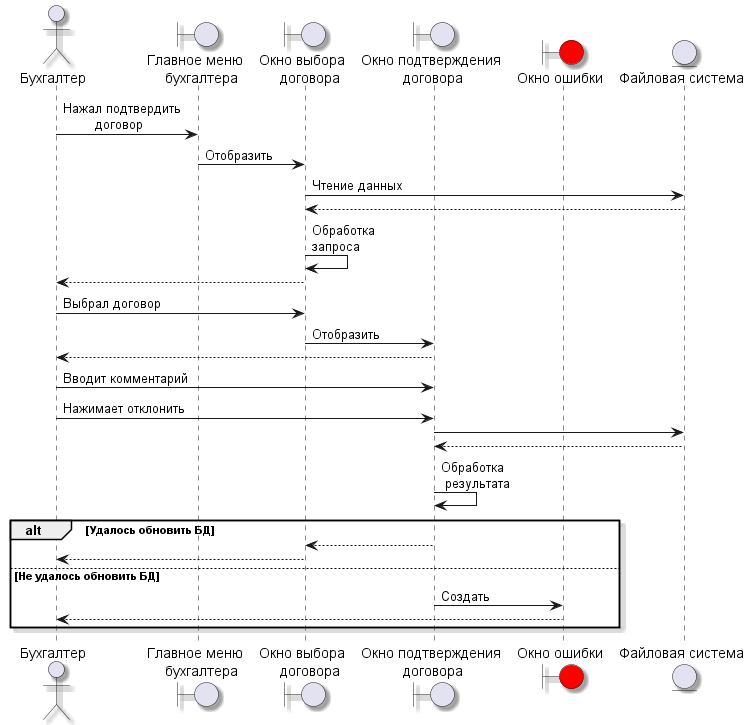


Рисунок 32 – «Отклонение договора»

1. Диаграммы пригодности и последовательности для прецедента «Просмотр статистики»

На рисунке 31 представлена диаграмма пригодности для данного прецедента.

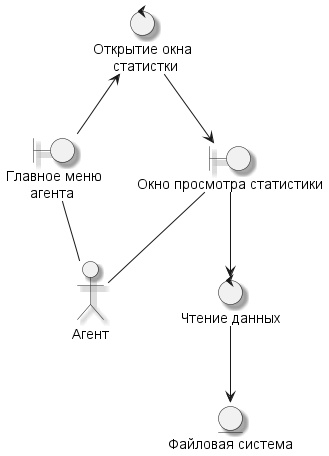


Рисунок 31 – «Просмотр статистики»

На рисунке 32 представлена диаграмма последовательности для этого прецедента.

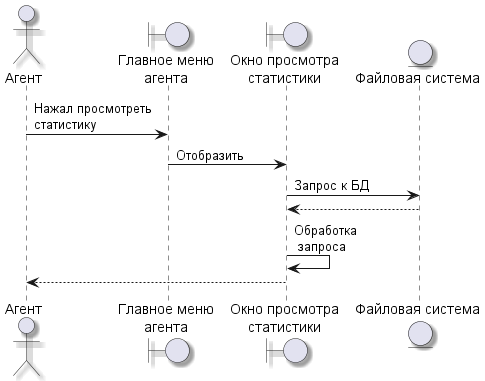


Рисунок 32 – «Просмотр статистики»

1. Диаграммы пригодности и последовательности для прецедента «Подтверждение договора»

На рисунке 33 представлена диаграмма пригодности для данного прецедента.

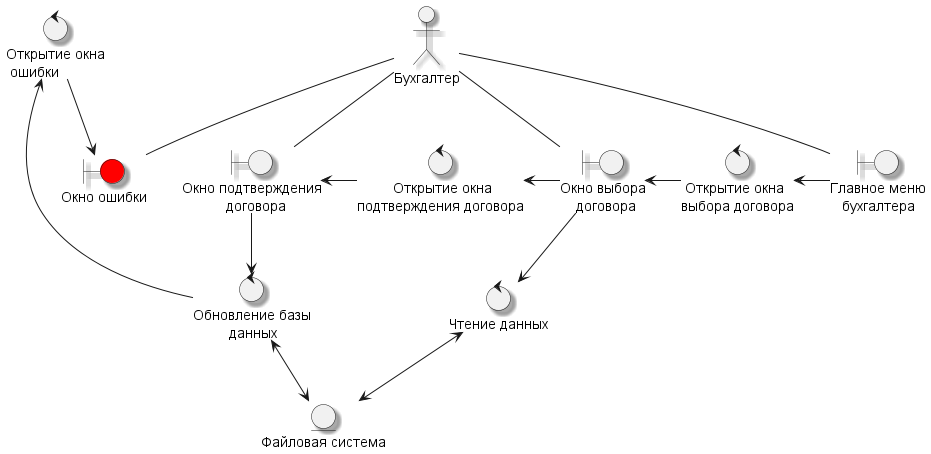


Рисунок 33 – «Подтверждение договора»

На рисунке 34 представлена диаграмма последовательности для этого прецедента.

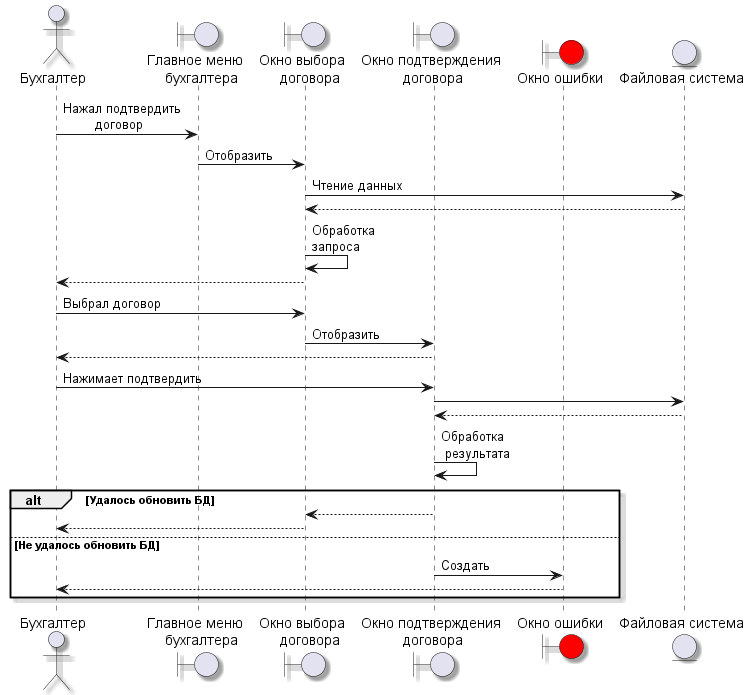


Рисунок 34 – «Подтверждение договора»

1. Диаграммы пригодности и последовательности для прецедента «Установка тарифной ставки»

На рисунке 35 представлена диаграмма пригодности для данного прецедента.

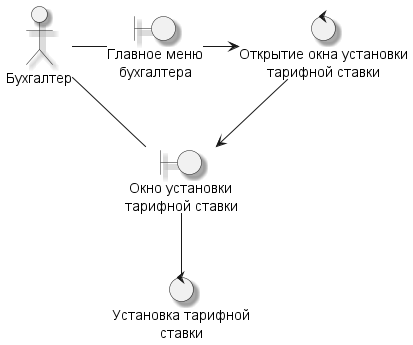


Рисунок 35 – «Установка тарифной ставки»

На рисунке 36 представлена диаграмма последовательности для этого прецедента.

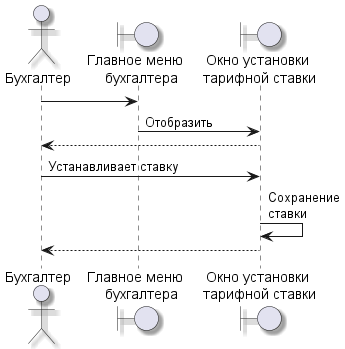


Рисунок 36 – «Установка тарифной ставки»

1. Диаграммы пригодности и последовательности для прецедента «Просмотр заключённых договоров»

На рисунке 37 представлена диаграмма пригодности для данного прецедента.

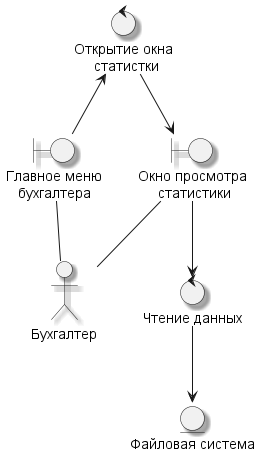


Рисунок 37 – «Просмотр заключённых договоров»

На рисунке 38 представлена диаграмма последовательности для этого прецедента.

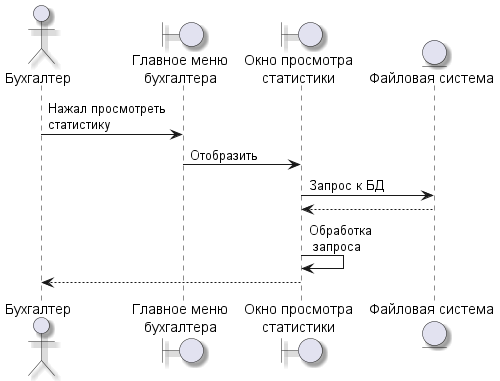


Рисунок 38 – «Просмотр заключённых договоров»

1. ER-диаграмма

На рисунке 39 представлена ER-диаграмма в нотации Мартина.

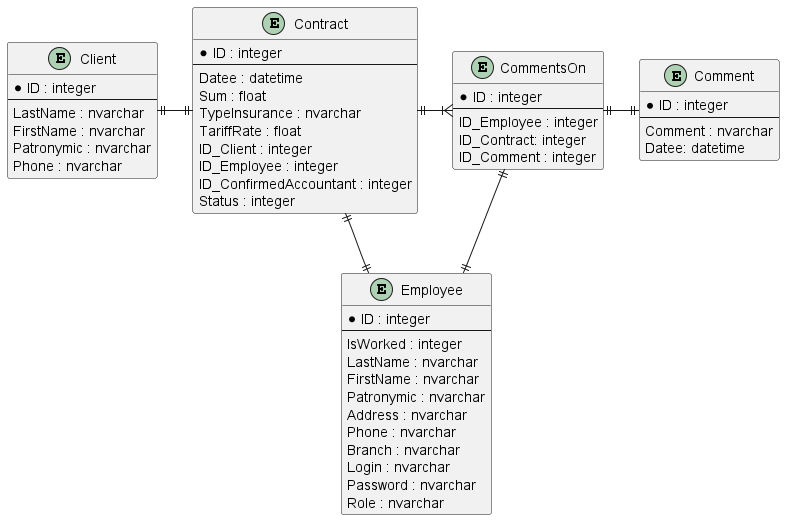


Рисунок 39 – ER-диаграмма

1. Диаграмма классов

На рисунках 40, 41, 42 представлены части диаграммы классов для соответствующих ролей. В приложении Б полностью представлена диаграмма классов.

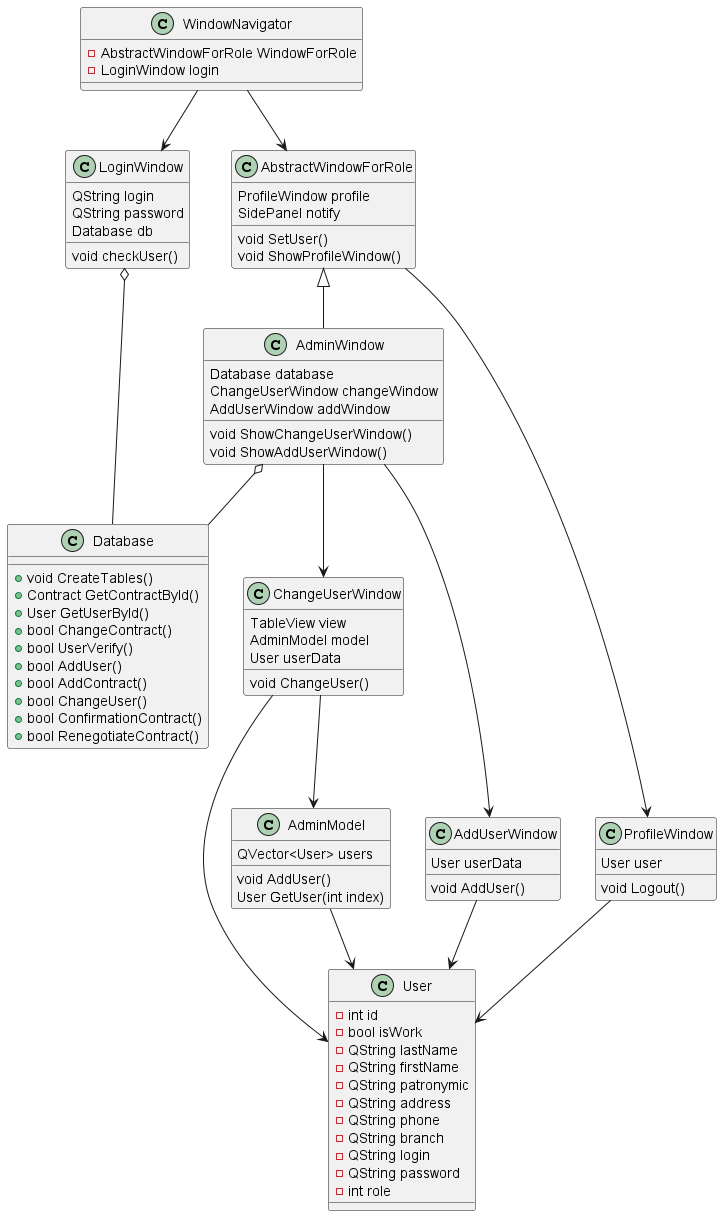


Рисунок 40 – Диаграмма классов для администратора

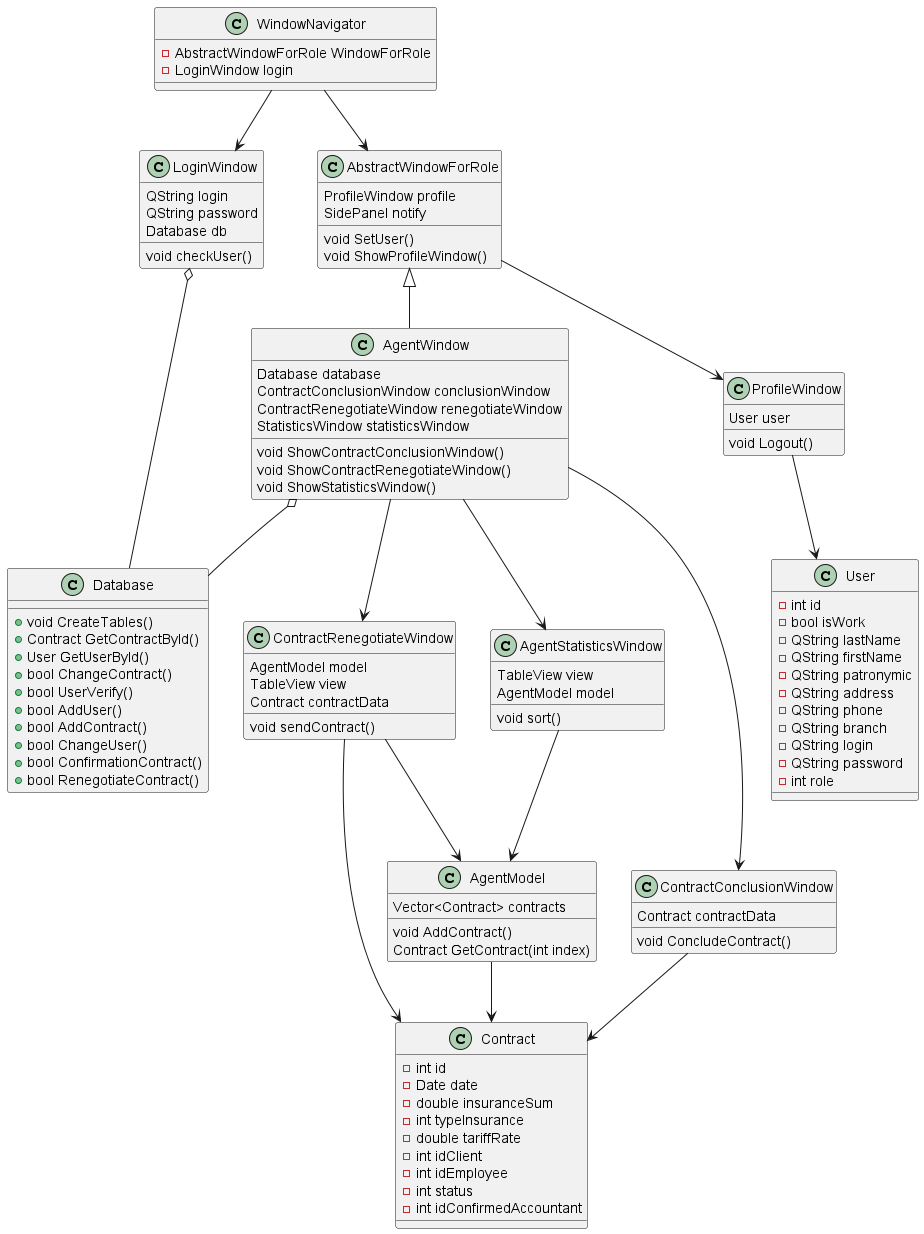


Рисунок 41 – Диаграмма классов для агента

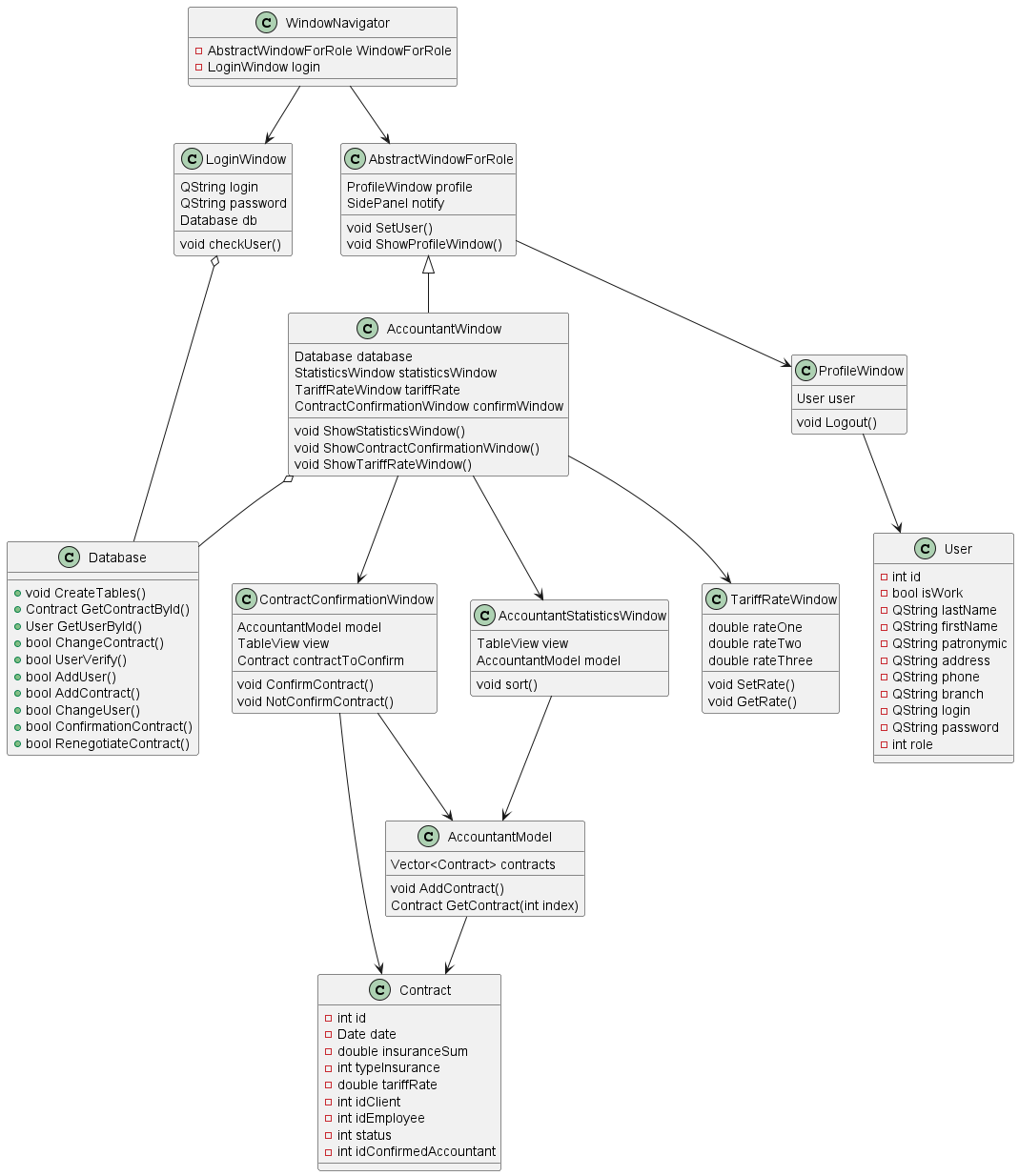


Рисунок 42 – Диаграмма классов для бухгалтера

## 3 **Объектно-ориентированное программирование**

* 1. Реализация

В приложении используется достаточно большое количество окон, в которых необходимо взаимодействовать с различными данными. Для организации работы с данными была выбрана архитектура модель-представление. В качестве модели используется QSqlQueryModel, потому что для неё достаточно лишь указать SQL запрос, и нет необходимости наследоваться (например, от QAbstractListModel) и переопределять методы. В качестве представления используется QTableView.

Для всплывающих окон используется библиотека QSydePanel из репозитория [4]. Она позволяет легко создавать всплывающие виджеты, устанавливать кривые для управления анимацией, а также работает при изменении размеров окна. Код библиотеки был модифицирован (были вырезаны кнопки на панелях и добавлена возможность задавать размер и положение всплывающего виджета).

В интерфейсе используются тени, добавленные с помощью QGragphicsDropShadowEffect. В последствии были признаны лишними, поэтому была добавлена возможность их выключить в окне профиля (эта настройка сохраняется с помощью QSettings и не сбрасывается при перезагрузке программы).

Для хранения данных используется база данных SQLite. Она была выбрана, потому что является встраиваемой и пользователю не придётся дополнительно устанавливать серверную часть СУБД. Все функции для работы с БД собраны в классе Database. При первом заходе пользователя в базе данных создаётся учётная запись администратора с логином «admin» и паролем «admin».

* 1. Сборка и запуск

Для динамической сборки проекта на Windows необходимо:

* проверить наличие путей до папок «…\Qt\6.6.0\mingw\_64\bin» и «…\Qt\Tools\mingw1120\_64\bin» в переменной path;
* перейти в папку проекта (командой «cd .\MDKP\src\MDKP»);
* запустить систему сборки (командой «qmake»);
* собрать проект (командой «mingw32-make -j8», где -j8 – число параллельных процессов, в которых будет выполняться компиляция).

В результате появится папка release, в которой будет .exe файл проекта.

Для запуска приложения необходимы зависимости. Их нужно добавить в папку с исполняемым файлом с помощью утилиты «windeployqt6.exe». Для этого необходимо использовать команду «path\_to\_bin\_folder\_qt\windeployqt6.exe path\_to\_your\_application\MDKP.exe» и затем запустить приложение. Если появится ошибка, то нужно вручную переместить необходимые библиотеки из папки bin в папку с исполняемым файлом.

* 1. Тестирование

Тестирование проводилось вручную. Для проверки корректности прецедента «Вход в систему» выполнялись следующие действия:

* вводились неверные данные;
* вводились данные уволенного сотрудника;
* одно или несколько полей оставлялись пустыми.

Для проверки корректности прецедента «Добавление пользователя» выполнялись следующие действия:

* одно или несколько полей оставлялись пустыми;
* поле «Телефон» заполнялось не полностью;
* вводился уже занятый номер телефона;
* ввод уже занятого логина;
* попытка ввода неразрешённых символов.

Для проверки корректности прецедента «Редактирование пользователя» выполнялись те же действия, что и для предыдущего прецедента.

Для проверки корректности прецедента «Заключение договора» выполнялись следующие действия:

* одно или несколько полей оставлялись пустыми;
* в поле «Сумма» водились нечисловые символы;
* вводились слишком длинные строки;
* поле «Телефон» заполнялось не полностью.

Для проверки корректности прецедента «Установка тарифной ставки» выполнялись следующие действия:

* попытка установки отрицательных значений;
* попытка установки значений, выходящих за диапазон.

Аналогичное тестирование было проведено для всех прецедентов, описанных в первом разделе.

Кроме того, для тестирования отдельных модулей были написаны тест с использованием Qt Test. В таблице 1 приведено описание.

Таблица 1 – Описание тестирования

| **Объект тестирования** | **Входные значения** | **Ожидаемый результат** | **Результат тестирования** |
| --- | --- | --- | --- |
| ValidationConstant::  EXP\_ON\_FIO | QString testString = "" | Строка пустая. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString =  "sssssssssssssssssssssssssssssssssssssssss" | Строка слишком длинная. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "\_Vasa\_" | Строка содержит цифры. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "Olgerd Фон Everek-third" | Строка корректна. Тест должен показать соответствие регулярному выражению. | Верно. |
| ValidationConstant::  EXP\_ON\_BRANCH \_AND\_ADDRESS | QString testString = "" | Строка пустая. Тест должен показать несоответствие регулярному выражению. | Верно. |

Окончание таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Объект тестирования** | **Входные значения** | **Ожидаемый результат** | **Результат тестирования** |
|  | QString testString = "Планета Земля, страна Россия, Красноярский край, г. Красноярск, ул Ленина, д 43, кв 915 подвал на платформе 9/3 четверти, школа магии Хогвартс, палатка Хагрида." | Строка слишком длинная. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "г. Красноярск, ул. Ленина, д. 517, кв. 777." | Строка корректна. Тест должен показать соответствие регулярному выражению. | Верно. |
| QString testString = "+==г. Красноярск, ул. Ленина, д. 517, кв. 777.+==" | Строка некорректна. Тест должен показать несоответствие регулярному выражению. | Верно. |
| ValidationConstant::  EXP\_ON\_LOGIN | QString testString = "" | Строка пустая. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "-666-\_\_\_\_\_\_\_\_\_\_VASAN\_\_\_228\_\_\_\_\_\_\_\_\_\_-666-" | Строка слишком длинная. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "\_Petechkin 228\_" | Строка корректна. Тест должен показать соответствие регулярному выражению. | Верно. |
| QString testString = "\_\_=++666DIZEL++=\_\_" | Строка некорректна. Тест должен показать несоответствие регулярному выражению. | Верно. |
| ValidationConstant::  EXP\_ON\_NUMBER \_LINE | QString testString = "" | Строка пустая. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "12345678911234.00" | Строка слишком длинная. Тест должен показать несоответствие регулярному выражению. | Верно. |
| QString testString = "-324.00" | Строка корректна. Тест должен показать соответствие регулярному выражению. | Верно. |
| QString testString = "517" | Строка некорректна. Тест должен показать несоответствие регулярному выражению. | Верно. |

В результате тестирования были выявлены и устранены следующие ошибки:

* возможность написать недопустимые символы в полях «Фамилия», «Имя», «Отчество»;
* возможность изменить телефон или логин пользователя на уже занятый в окне редактирования пользователей (администратор);
* возможность войти с помощью учётной записи уволенного сотрудника.
  1. Инструкция

В качестве инструкций для пользователя можно использовать прецеденты, описанные в первом разделе и диаграммы последовательности, разработанные во втором разделе.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была выполнена разработка спецификации требований, объектно-ориентированное проектирование, разработка приложения, а также его тестирование. Задача была выполнена полностью.

# Список используемых источников

1. СТУ 7.5–07–2021. Стандарт университета «Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности».
2. Документация Qt // QT | Tools for Each Stage of Software Development Lifecycle. – URL: <https://doc.qt.io/> (дата обращения 24.02.2024).
3. Основы UML – диаграммы использования (use-case) // Блог программиста – программирование и алгоритмы URL: <https://pro-prof.com/>.
4. <https://github.com/inobelar/QSidePanel>

# Приложение А

**Диаграммы потоков экранов**



Рисунок А.1 – Диаграмма потоков экранов (Администратор)

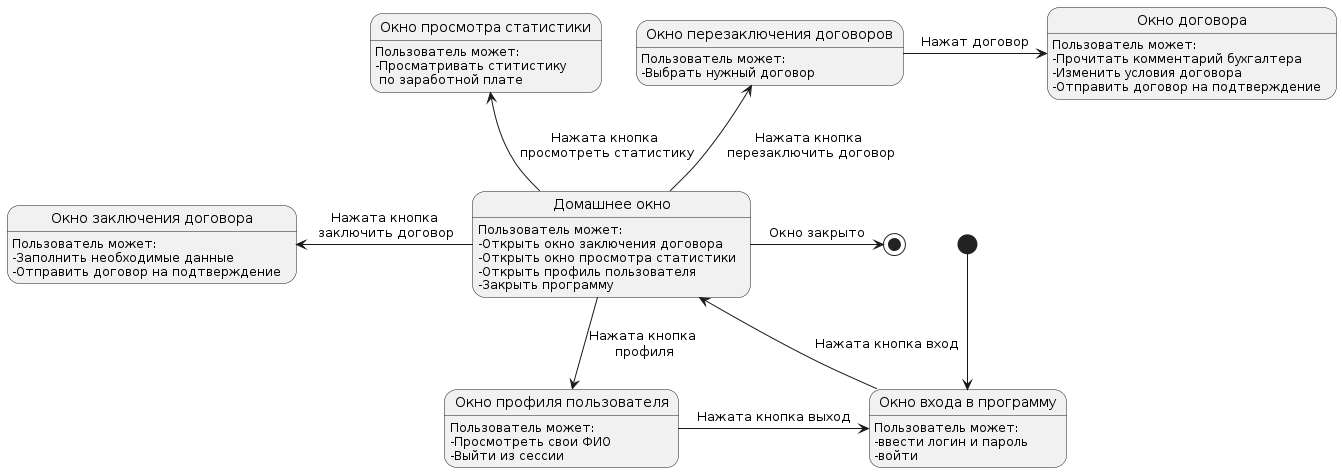


Рисунок А.2 – Диаграмма потоков экранов (Агент)

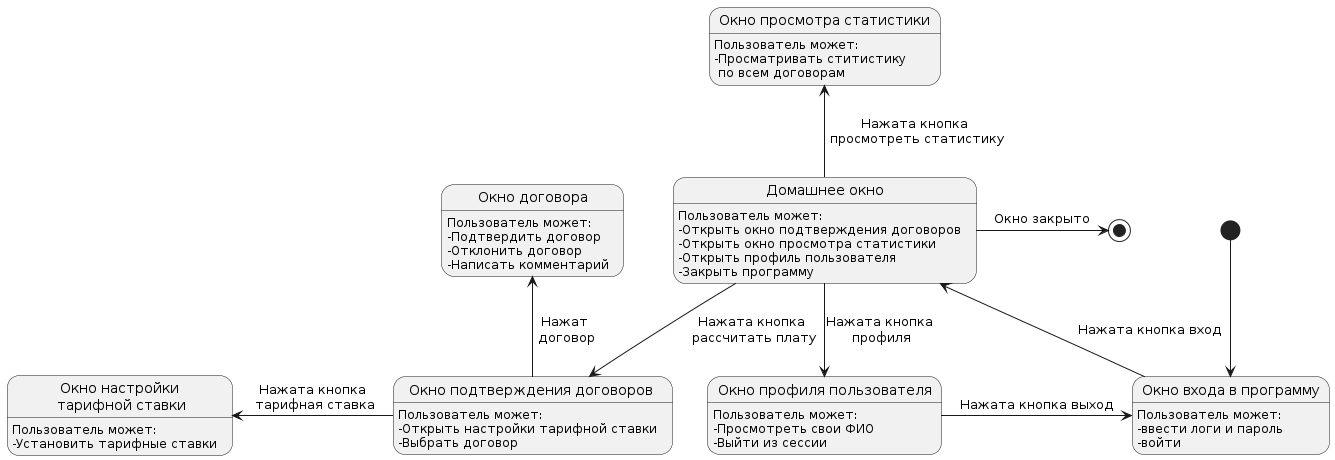


Рисунок А.3 – Диаграмма потоков экранов (Бухгалтер)

# Приложение Б

**Диаграмма классов**

