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 «» |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Руководитель |  |  |  |  | Л.С. Артемьев |
|  |  | подпись, дата |  | инициалы, фамилия |
|  |  |  |  |  |  |
| Студент |  |  |  |  | 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, объектное ориентированное программирование

Целью проекта является

В первой главе

Во второй главе

В третьей главе

СОДЕРЖАНИЕ

[Введение 4](#_Toc167485330)

[1 Разработка спецификации требований 5](#_Toc167485331)

[1.1 Постановка задачи 5](#_Toc167485332)

[1.2 Выявление ролей и функций, диаграмма прецедентов 5](#_Toc167485333)

[1.3 Прецедент «Авторизация» 5](#_Toc167485334)

[1.4 Прецедент 6](#_Toc167485335)

[1.5 Описание форматов данных 6](#_Toc167485336)

[2 Объектно-ориентированное проектирование 6](#_Toc167485337)

[2.1 Диаграммы пригодности и последовательности для прецедента «Авторизация» 7](#_Toc167485338)

[2.2 Диаграммы пригодности и последовательности для прецедента 8](#_Toc167485339)

[2.3 ER-диаграмма 8](#_Toc167485340)

[2.4 Диаграмма классов 9](#_Toc167485341)

[3 Объектно-ориентированное программирование 9](#_Toc167485342)

[3.1 Реализация 9](#_Toc167485343)

[3.2 Сборка и запуск 9](#_Toc167485344)

[3.3 Тестирование 9](#_Toc167485345)

[3.4 Инструкция 9](#_Toc167485346)

[Заключение 10](#_Toc167485347)

[Приложение А 11](#_Toc167485348)

[Список используемых источников 12](#_Toc167485349)

Введение

Сегодня страхование является неотъемлемой частью жизни современного человека. Поэтому растёт число компаний, оказывающих данные услуги. И для того, чтобы увеличить прибыль, необходимо ускорить и упростить обмен данными, а также перевести всё в электронный вид.

**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

Текст

* 1. Постановка задачи

Страховая компания имеет филиалы, которые характеризуются наименованием, адресом и телефоном. В филиалы обращаются клиенты с целью заключения договора о страховании. В зависимости от принимаемых на страхование объектов и страхуемых рисков 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 и филиал, в котором заключался договор. 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 страхового платежа (платеж –страховая сумма, умноженная на тарифную ставку). 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.

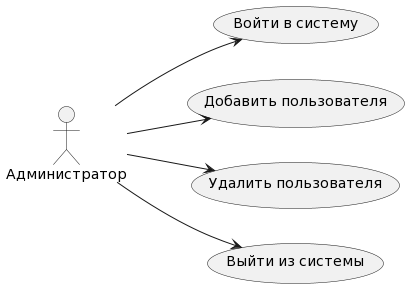


Рисунок 1 – Диаграмма прецедентов (Администратор)

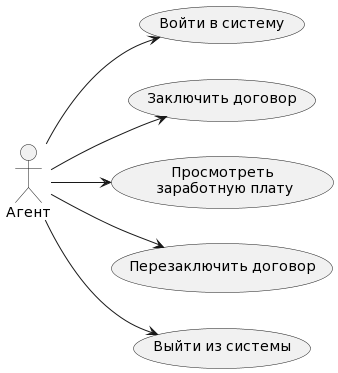


Рисунок 2 – Диаграмма прецедентов (Агент)



Рисунок 3 – Диаграмма прецедентов (Бухгалтер)

В приложении А представлена диаграмма потока экранов.

## Прецедент «Вход в систему»

Макет интерфейса для прецедента представлен на рисунке 4.

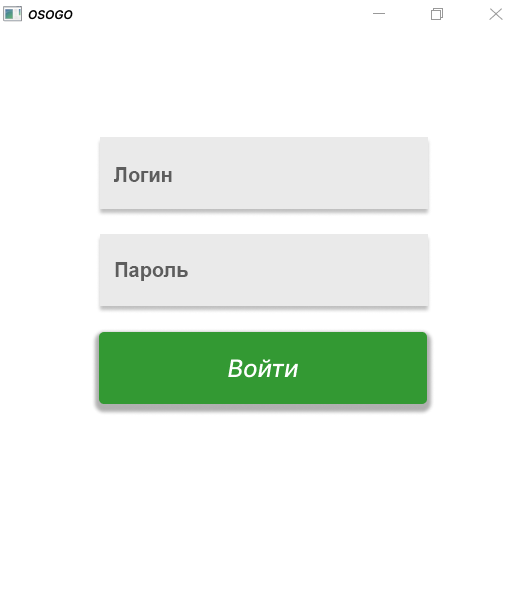


Рисунок 4 – Окно для авторизации

**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.

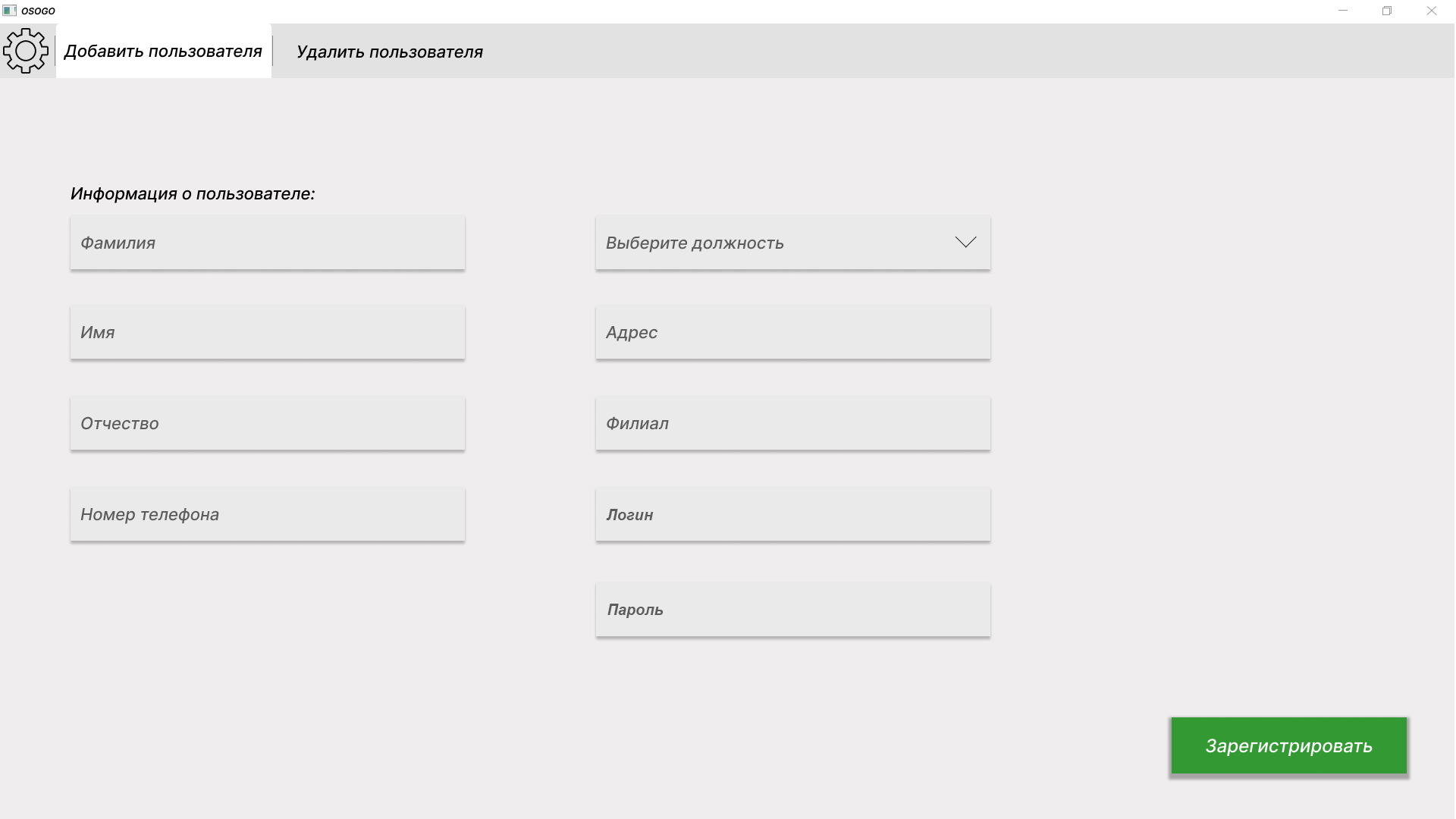


Рисунок 5 – вкладка добавления пользователя

Роль: администратор.

Предусловие: открыта вкладка добавления пользователя.

Основной сценарий:

1. В «Окне добавления пользователей» (Рисунок 9) ввести ФИО, телефон, должность, адрес, филиал;
2. Нажать кнопку «Зарегистрировать»

Постусловие:

Альтернативный сценарий:

Постусловие:

* 1. Описание форматов данных

Важной частью функционирования системы является хранение и обработка данных.

Таблица 1 – Пример заполнения users

|  |  |  |  |
| --- | --- | --- | --- |
| **Идентификатор** | **Логин** | **Пароль** | **Роль** |
| 1 | Admin | Admin |  |
| 2 | Test | Test |  |

## 2 Объектно-ориентированное проектирование

1. Диаграммы пригодности и последовательности для прецедента «Авторизация»

На рисунке 5 представлена диаграмма пригодности для данного прецедента.

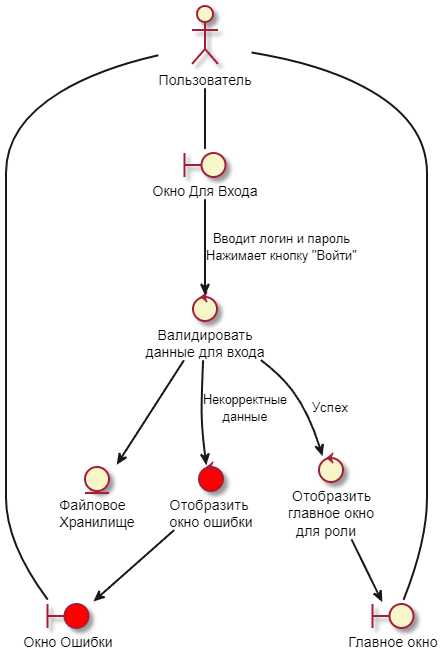


Рисунок 5 – «Авторизация»

На рисунке 6 представлена диаграмма последовательности для этого прецедента.

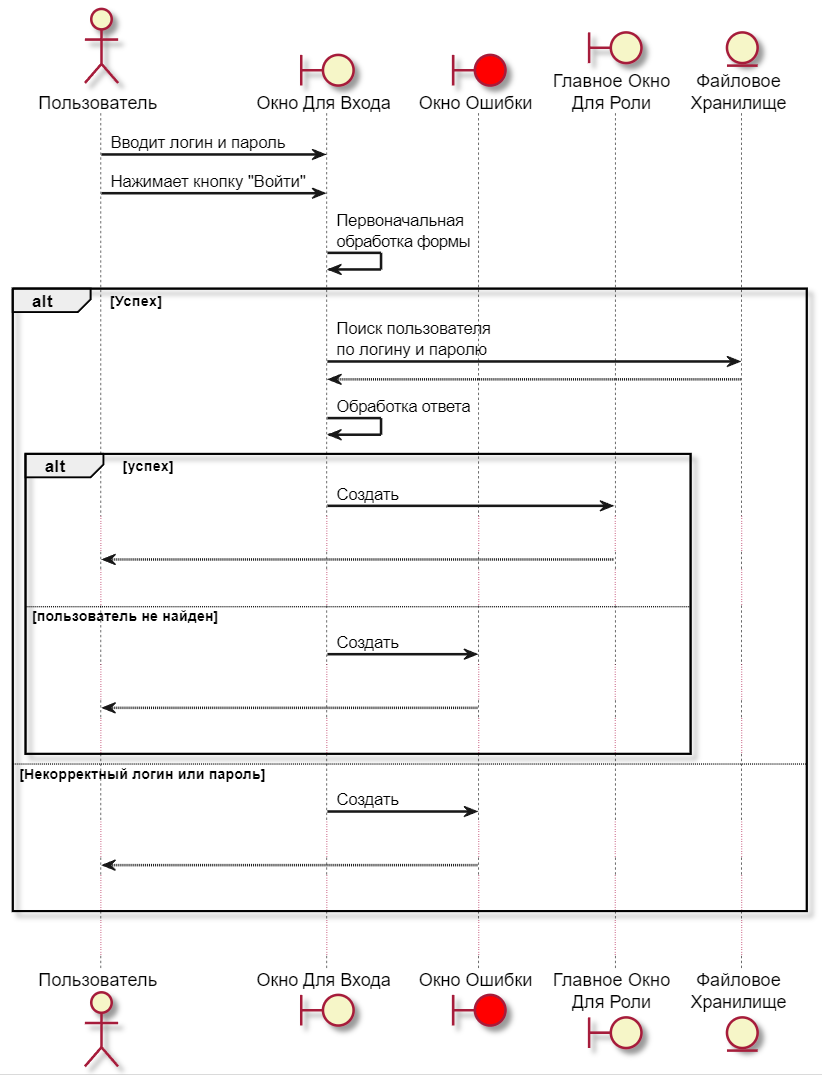


Рисунок 6 – «Авторизация»

1. Диаграммы пригодности и последовательности для прецедента
2. ER-диаграмма

На рисунке \_ представлена ER-диаграмма в нотации Мартина.

Рисунок \_ – ER-диаграмма

1. Диаграмма классов

На рисунке \_ представлена диаграмма классов.

Рисунок \_ – Диаграмма классов

## 3 Объектно-ориентированное программирование

* 1. Реализация

Исходный код программы представлен в свободном доступе на Github

* 1. Сборка и запуск
  2. Тестирование

Тестирование проводилось

* 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

# Приложение А

**Диаграммы потоков экранов**



Рисунок А.1 – Диаграмма потоков экранов (Администратор)

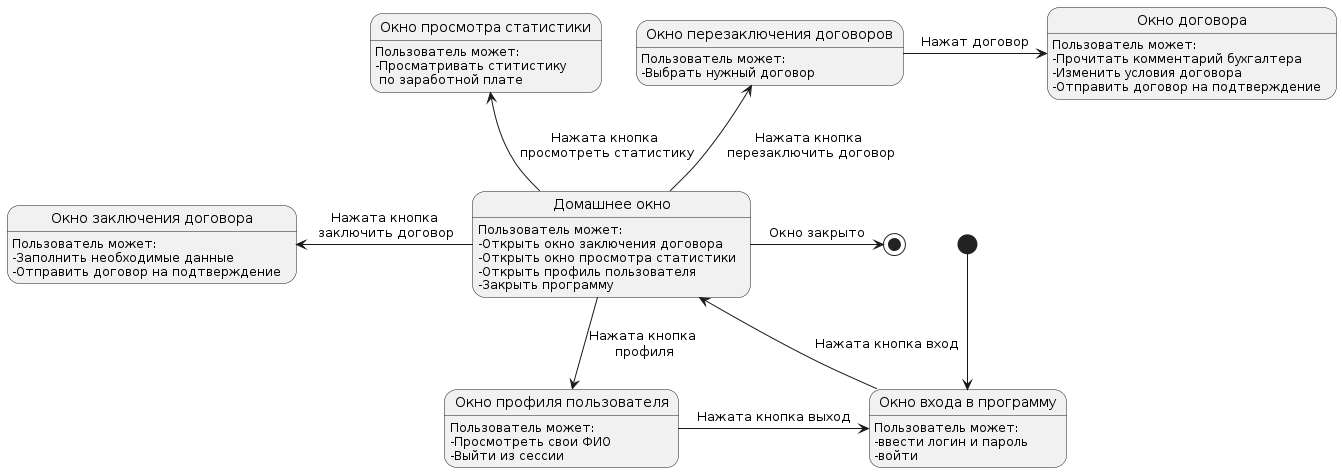


Рисунок А.2 – Диаграмма потоков экранов (Агент)

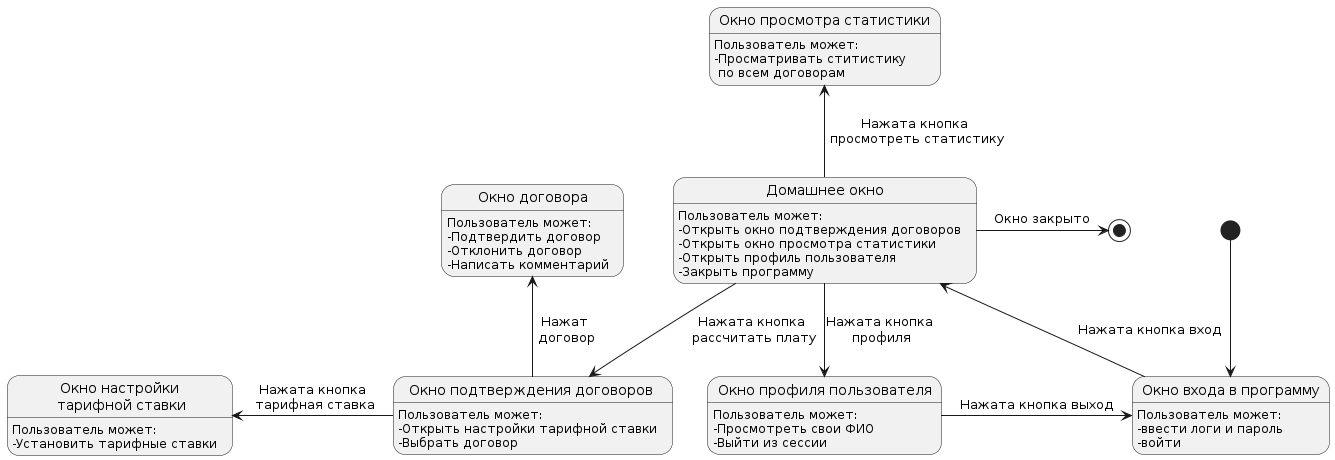


Рисунок А.3 – Диаграмма потоков экранов (Бухгалтер)

# 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/>.