Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра «Вычислительная техника»

**КУРСОВОЙ ПРОЕКТ**

Информационная система «Гостиница»

Преподаватель \_\_\_\_\_\_\_\_\_\_ В.С. Васильев

подпись, дата инициалы, фамилия

Студент КИ20-06Б, 032049103 \_\_\_\_\_\_\_\_\_\_ Д.А. Анциферов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2021

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc89250403)

[1 Разработка спецификации требования 5](#_Toc89250404)

[1.1 Выявление ролей и функций (графическая часть диаграммы прецедентов). 5](#_Toc89250405)

[1.2 Разработка макетов интерфейса 6](#_Toc89250406)

[1.2.1 Макеты интерфейса клиента 8](#_Toc89250407)

[1.2.2 Макеты интерфейса управляющего 10](#_Toc89250408)

[1.2.3 Макеты интерфейса администратора 13](#_Toc89250409)

[1.3 Описание формата данных 14](#_Toc89250410)

[1.3.1 Информация о комнатах 14](#_Toc89250411)

[1.3.2 Информация о пользователях 14](#_Toc89250412)

[1.3.3 Информация о забронированных комнатах 15](#_Toc89250413)

[1.4 Описание поведения программы (текстовая часть диаграммы прецедентов) 16](#_Toc89250414)

[AuthForm 17](#_Toc89250415)

[Recording Users 17](#_Toc89250416)

[1.4.1 Интерфейс клиента 18](#_Toc89250417)

[1.4.2 Интерфейс руководителя 19](#_Toc89250418)

[1.4.3 Интерфейс администратора 21](#_Toc89250419)

[2 Объектно-ориентированное проектирование 22](#_Toc89250420)

[2.1 Разработка диаграммы пригодности 22](#_Toc89250421)

[2.1.1 Интерфейс клиента 24](#_Toc89250422)

[2.1.2 Интерфейс руководителя 25](#_Toc89250423)

[2.1.3 Интерфейс администратора 27](#_Toc89250424)

[2.2 Разработка диаграммы последовательности 28](#_Toc89250425)

[AuthForm 28](#_Toc89250426)

[Recording Users 28](#_Toc89250427)

[2.2.1 Интерфейс клиента 29](#_Toc89250428)

[2.2.2 Интерфейс руководителя 30](#_Toc89250429)

[2.2.3 Интерфейс администратора 32](#_Toc89250430)

[2.2.3 Разработка диаграммы классов уровня проектирования 33](#_Toc89250431)

[3 Объектно-ориентированное программирование 34](#_Toc89250432)

[3.1 Разработка модульных тестов 34](#_Toc89250433)

[3.1.1 Проверки ввода данных при регистрации пользователем. 34](#_Toc89250434)

[3.1.2 Проверки ввода данных при создании или редактировании комнаты 35](#_Toc89250435)

[3.2 Реализация программы 37](#_Toc89250436)

[3.3 Рефакторинг 38](#_Toc89250437)

[3.3.1 Степень покрытия тестами 38](#_Toc89250438)

[3.3.2 Гибкость архитектуры 38](#_Toc89250439)

[3.3.3 Соответствие стандарту кодирования 38](#_Toc89250440)

[3.4 Документирование 39](#_Toc89250441)

[3.4.1 Документация, созданная с помощью систем генерации документов 39](#_Toc89250442)

[3.4.2 Инструкция по сборке программы 39](#_Toc89250443)

[3.4.3 Первый запуск программы 39](#_Toc89250444)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 40](#_Toc89250445)

# Введение

Гостиница предоставляет номера клиентам. Каждый номер характеризуется вместимостью, комфортностью (люкс, полулюкс, обычный) и ценой. О клиентах собирается определенная информация (фамилия, имя, отчество, паспортные данные, адрес жительства и некоторый комментарий). Сдача номера клиенту производится при наличии свободных мест в номерах, подходящих клиенту по указанным выше параметрам. При заселении фиксируется дата заселения. При выезде из гостиницы для каждого места запоминается дата освобождения. Необходимо также осуществлять бронирование номеров. Для постоянных клиентов, а также для определенных категорий клиентов предусмотрена система скидок. Скидки могут суммироваться.

# 1 Разработка спецификации требования

## 1.1 Выявление ролей и функций (графическая часть диаграммы прецедентов).

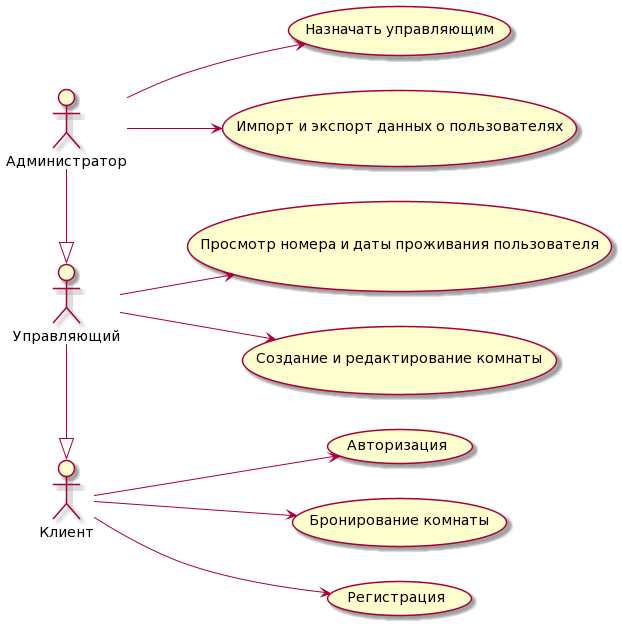


Рисунок 1.1 – Диаграмма прецедентов

## 1.2 Разработка макетов интерфейса

При запуске программы запускается окно регистрации, в котором вводится фамилия и серия и номер паспорта пользователя. В случае, если пользователь не был ранее зарегистрированным, есть кнопка регистрации, которая открывает окно регистрации. (рис. 1)

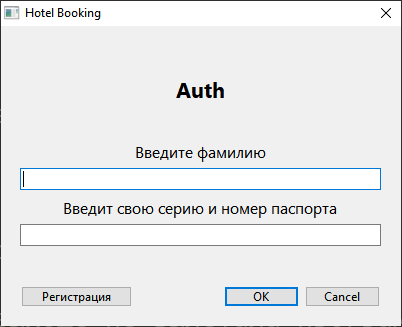


Рисунок 1.2 – Окно авторизации

Окно регистрации открывается по кнопке Register в окне авторизации. (рис. 2)

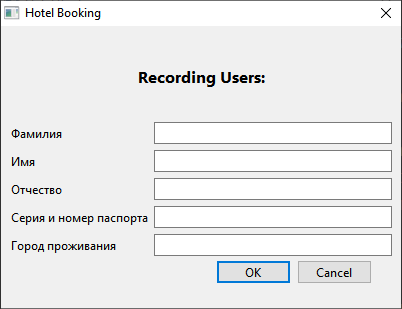


Рисунок 1.3 – Окно регистрации

### **1.2.1 Макеты интерфейса клиента**

В главном окне программы перед нами предстоит текущее время и таблица, в которой содержатся комнаты, их вместимость, комфорт, цена и доступность в данный момент (рис. 3).

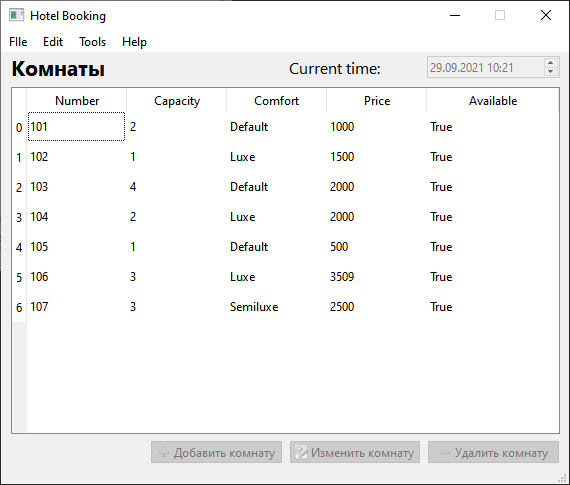


Рисунок 1.4 – Главное окно

При двойном нажатии на строчку открывается окно бронирования, в котором клиент может забронировать комнату на определенные даты и узнать итоговую цену, которую ему придется отдать за прожитие в этой комнате в промежуточных датах (рис.4).

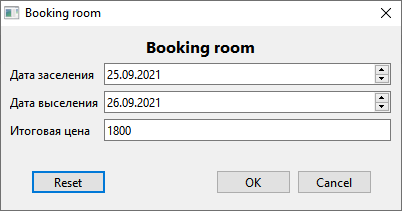


Рисунок 1.5 – Бронирование комнаты

### **1.2.2 Макеты интерфейса управляющего**

Существует список пользователей, который находится в . В этом окне администратор может узнать серию и номер паспорта пользователя, дата заселения и выселения, комнату и количество посещений. При нажатии Reset сбрасывается информация о комнате, дате заселения и выселения, количество посещений. (рис. 5).

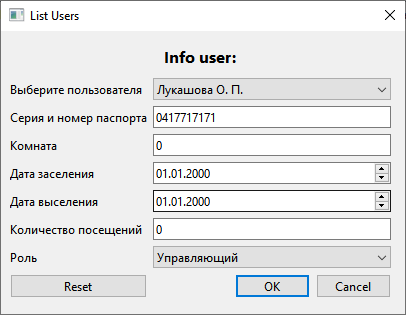


Рисунок 1.6 – Лист пользователей

Есть функция создания и редактирования комнаты в и в соответственно. Доступна эта функция управляющим и администратору. (рис. 6)

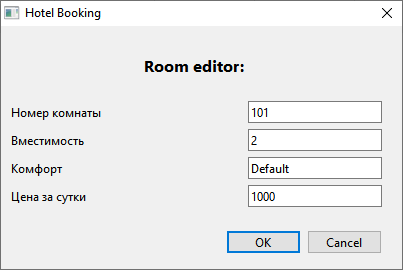


Рисунок 1.7 – Создание и редактирование комнат

При нажатии два раза по комнате в окне администратора открывается окно, в котором отображаются пользователи, которые забронировали комнату (рис. 7)

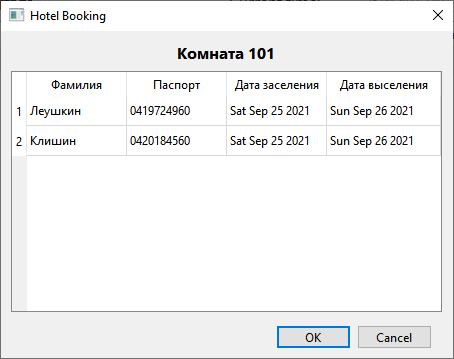


Рисунок 1.8 – Отображение пользователей, забронировавших комнату

### **1.2.3 Макеты интерфейса администратора**

Если в окно List Users по пути зайдет администратор, то откроется такое же окно с такими же возможностями, как и у руководителя, только с возможность назначать руководителем. (рис. 8)

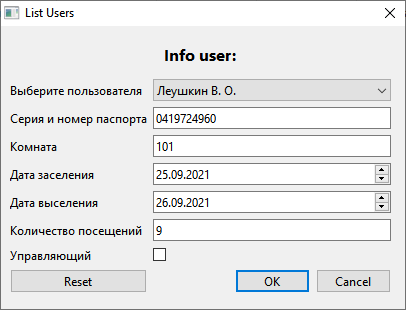


Рисунок 1.9 - Лист пользователей (администратор)

## 1.3 Описание формата данных

### **1.3.1 Информация о комнатах**

Программа обрабатывает комнаты, которые хранятся в файле комнаты.csv и используют формат CSV.

<номер комнаты>;< вместимость>;< комфортность >;<цена>;

В таблице 1 приведено описание форматов полей, а на рисунке 9 приведен пример содержимого комнаты.csv

Таблица 1 – Описание форматов полей

|  |  |
| --- | --- |
| Поле | Тип |
| Номер комнаты | Числовой |
| Вместимость | Числовой |
| Комфортность | Числовой |
| Цена | Числовой |

Номер;Вместимость;Комфортность;Цена

101;2;Default;1000

102;1;Luxe;1500

103;3;Semiluxe;2500

Рисунок 1.10 – Пример содержимого комнаты.csv

### **1.3.2 Информация о пользователях**

Программа обрабатывает пользоватей, которые хранятся в файле пользователи.csv и используют формат CSV.

<фамилия>;< имя>;<отчество>;< серия и номер паспорта >;<адрес>;<роль>

В таблице 1 приведено описание форматов полей, а на рисунке 10 приведен пример содержимого пользователи.csv

Таблица 1 – Описание форматов полей

|  |  |
| --- | --- |
| Поле | Тип |
| Фамилия | Строковый |
| Имя | Строковый |
| Отчество | Строковый |
| Серия и номер паспорта | Строковый |
| Адрес | Строковый |
| Роль | Строковый |

Фамилия;Имя;Отчество;Серия и номер паспорта;Адрес;Роль

Анциферов;Денис;Александрович;0415882152;Ачинск;Администратор

Леушкин;Владислав;Олегович;0417883812;Красноярск;Клиент

Лукашова;Ольга;Петровна;0419123456;Новосибирск;Руководитель

Рисунок 1.11 – Пример содержимого комнаты.csv

### **1.3.3 Информация о забронированных комнатах**

Программа обрабатывает забронированные комнаты, которые хранятся в файле комнаты\_пользователи.csv и используют формат CSV.

<номер комнаты>;< серия и номер паспорта клиента>;< дата заселения >;<дата выселения>;

В таблице 1 приведено описание форматов полей, а на рисунке 11 приведен пример содержимого комнаты\_пользователи.csv

Таблица 1 – Описание форматов полей

|  |  |
| --- | --- |
| Поле | Тип |
| Номер комнаты | Числовой |
| Серия и номер паспорта клиента | Строковый |
| Дата заселения | Дата формата день.месяц.год |
| Дата выселения | Дата формата день.месяц.год |

Номер;Серия;Заселение;Выселения

101;0415882152;19.09.2021;21.09.2021

102;0415882152;17.09.2021;20.09.2021

103; 0419123456;17.09.2021;18.09.2021

Рисунок 1.12 – Пример содержимого пользователи.csv

## 1.4 Описание поведения программы (текстовая часть диаграммы прецедентов)

Диаграмма прецедентов программы Hotel Booking показана на рисунке 1.

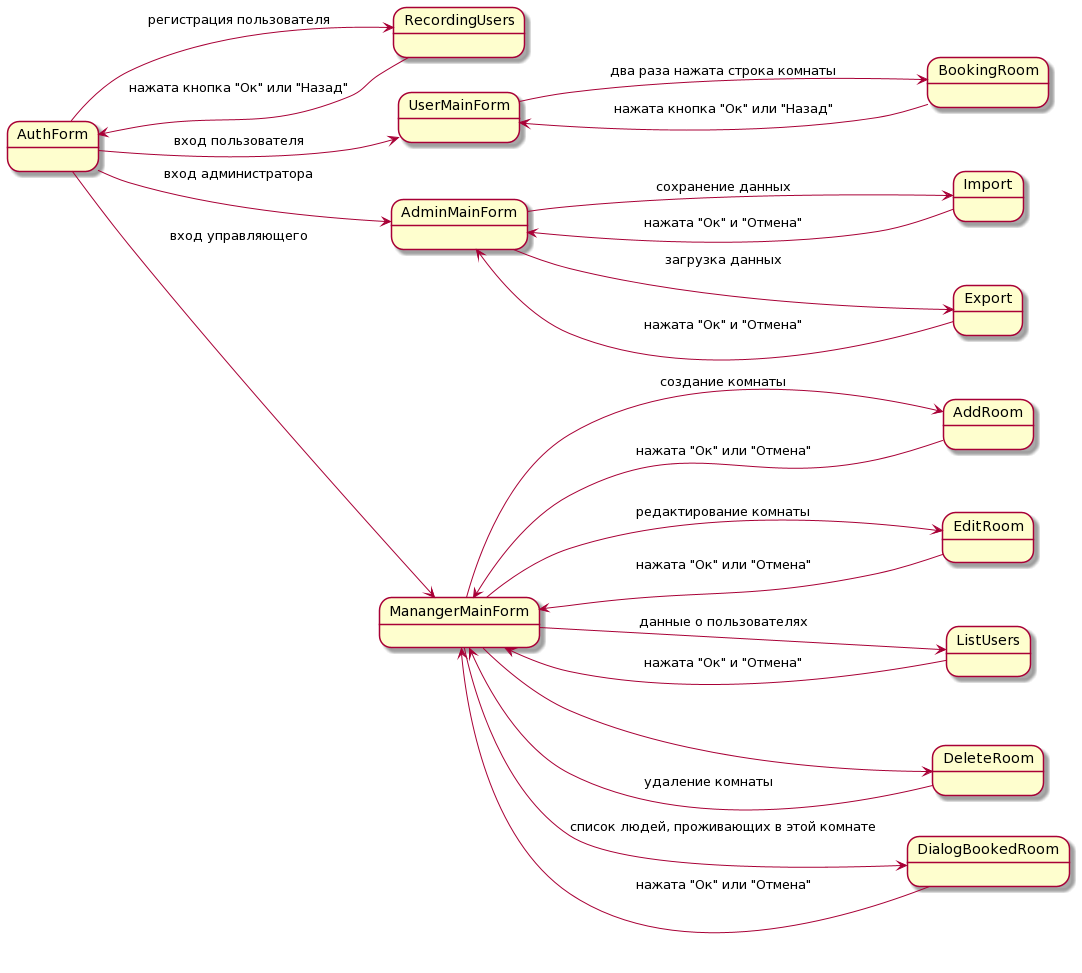


Рисунок 1.13 – Диаграмма прецедентов программы

### **AuthForm**

Прецедент: Авторизация пользователя

Предусловие: открыто окно AuthForm

Основная последовательность.

1. Ввести логин и пароль
2. При нажатии «ОК» и верной комбинации логин и пароль открывается главное окно с возможностями, зависящими от типа пользователя

Постусловие: открывается главное окно, соответствующее роли пользователя.

Альтернативная последовательность (неверный логин и пароль):

1. Ввести логин и пароль
2. Ввод неверного логина и пароля, нажатие кнопки «ОК»

Постусловие: поля формы очищаются, всплывает окно с ошибкой

### **Recording Users**

Прецедент: Регистрация пользователя

Предусловие: в окне AuthForm нажата кнопка Register

Основная последовательность.

1. Заполнить форму регистрации
2. При нажимании кнопки «ОК» пользователь регистрируется и окно закрывается

Постусловие: пользователь регистрируется, окно закрывается

Альтернативная последовательность (не все поля заполнены):

1. Неполное заполнение формы регистрации
2. Всплывает окно с ошибкой ввода данных.

Постусловие: поля формы остаются, окно не закрывается

### **1.4.1 Интерфейс клиента**

#### **1.4.1.1 Booking room**

Прецедент: Бронирование комнаты

Предусловие: Пользователь «клиент», из главного окна совершают двойное нажатие по строке комнаты

Основная последовательность.

1. Выбирается дата заселения и дата выселения
2. Автоматически считается цена
3. При нажатии кнопки «ОК» окно бронирования закрывается.

Постусловие: Комната забронирована

### **1.4.2 Интерфейс руководителя**

#### **1.4.2.1 Add Room**

Прецедент: создание комнаты

Предусловие: в главном окне управляющего или администратора нажимается сверху пункт .

Основная последовательность:

1. Заполняется номер комнаты, её вместимость, комфортность (Luxe, Semiluxe, Default) и цена за сутки.
2. При нажатии кнопки «ОК» окно закрывается

Постусловие: создается новая комната

#### **1.4.2.2 Edit Room**

Прецедент: Редактирование комнаты

Предусловие: в главном окне управляющего или администратора в таблице выделяется нужная комната и нажимается сверху пункт

Основная последовательность:

1. Изменяются параметры комнаты
2. При нажатии кнопки «ОК» окно закрывается

Постусловие: комната редактируется новыми данными

#### **1.4.2.3 List Users**

Прецедент: Список пользователей

Предусловие: в главном окне управляющего или администратора нажимается сверху пункт

Основная последовательность:

1. В списке выбирается необходимый пользователь и в поля ниже (серия и номер паспорта, комната, в которой проживает пользователь, дата заселения и выселения, количество посещений и управляющий или нет) заполняются автоматически.
2. Если в данное окно зашел администратор, то у него есть возможность сделать руководителем.
3. При нажатии кнопки «Reset» обнуляется информация о комнате, дате заселения и выселения, количество посещений.
4. При нажатии «ОК» данные сохраняются и окно закрывается, при нажатии «Отмена» окно закрывается.

Постусловие: закрывается окно

**1.4.2.4 Dialog Booked Room**

Прецедент: Окно забронированной комнаты

Предусловие: пользователь, авторизированный как администратор или

управляющий

Основная последовательность:

1. Двойное нажатие по комнате в главном окне
2. Таблица заполняется списком клиентов, которые забронировали эту комнату

Постусловие: окно закрывается

Альтернативная последовательность (комната полностью пустая)

1. Двойное нажатие по комнате в главном окне
2. Открывается информационное окно, в котором говорится о том, что комната никем не забронированная.

Постусловие: окно не открывается

### **1.4.3 Интерфейс администратора**

#### **1.4.3.1 Export**

Прецедент: экспорт данных

Постусловие: пользователь, авторизированный как администратор нажимает

Основная последовательность:

1. Открывается окно выбора папки, в которую будет сохранен файл (файл сохраняется в виде таблицы формата CSV).

Постусловие: сохранения данных, возращение в главное окно

#### **4.3.2 Import:**

Прецедент: импорт данных

Постусловие: пользователь, авторизированный как администратор

Основная последовательность:

1. Открывается окно выбора файла формата CSV.

Постусловие: загрузка данных, возращение в главное окно

# 2 Объектно-ориентированное проектирование

## 2.1 Разработка диаграммы пригодности

#### AuthForm

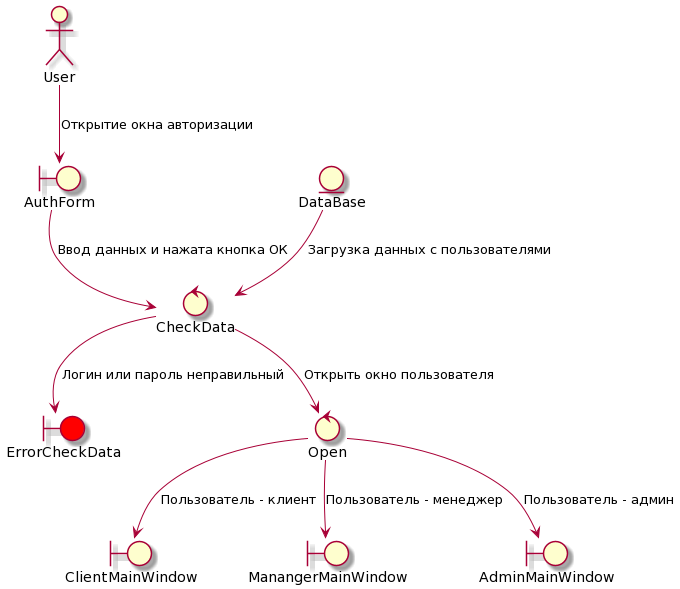
****

Рисунок 2.1 – Диаграмма пригодности для AuthForm

#### Recording Users

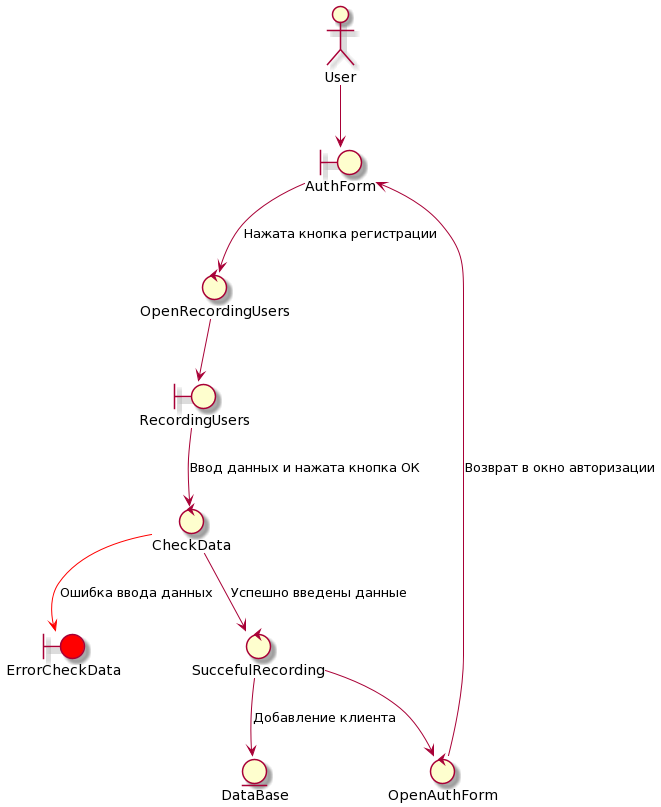
****

Рисунок 2.2 – Диаграмма пригодности для Recording Users

### 2.1.1 Интерфейс клиента

#### 2.1.1.1 Booking Room

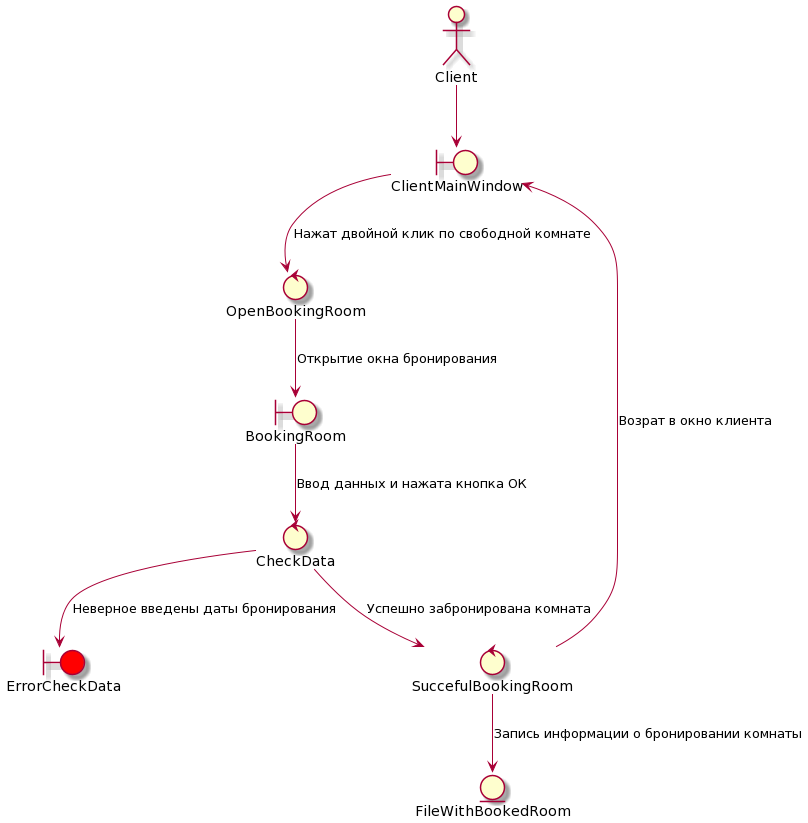
****

Рисунок 2.3 – Диаграмма пригодности для Booking Room

### 2.1.2 Интерфейс руководителя

#### 2.1.2.1 Add Room

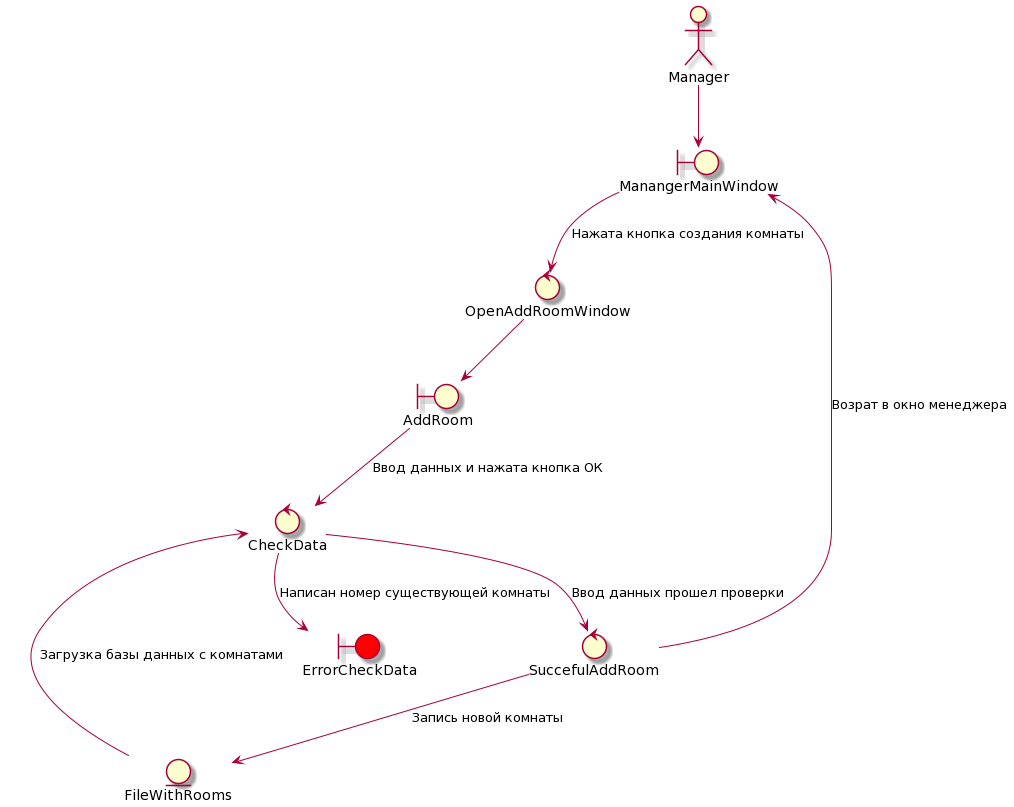
****

Рисунок 2.4 – Диаграмма пригодности для Add Room

#### 2.1.2.2 Edit Room

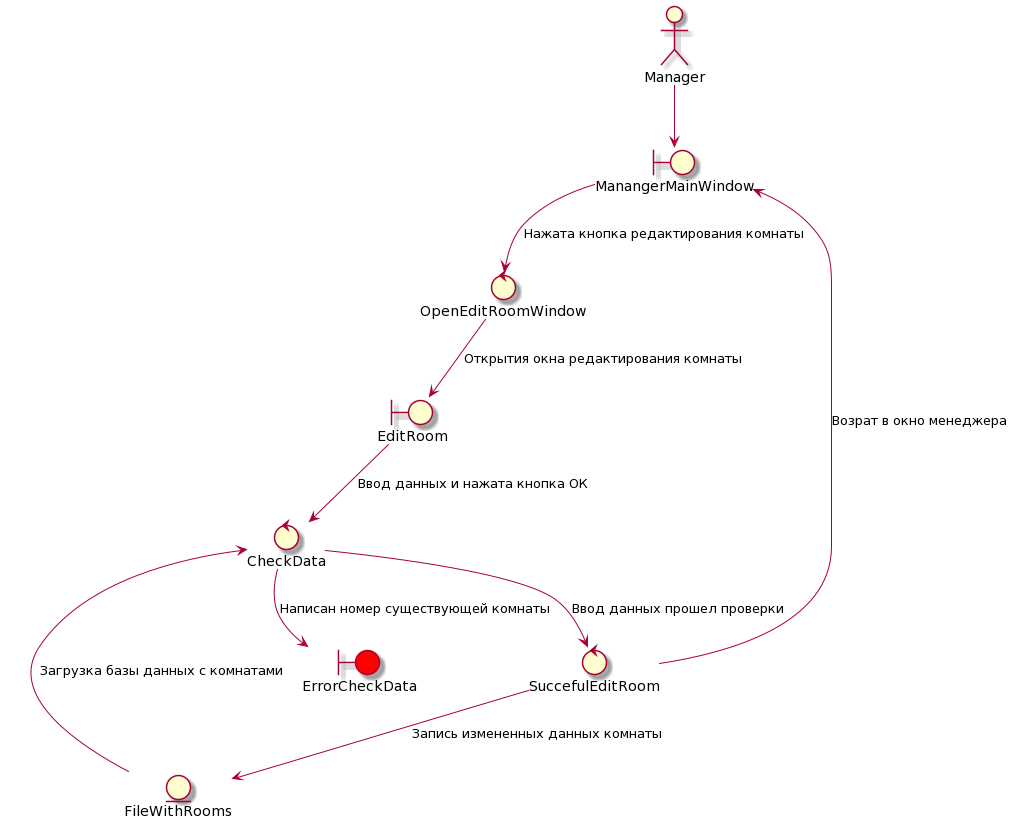
****

Рисунок 2.5 – Диаграмма пригодности для Edit Room

#### 2.1.2.3 List Users

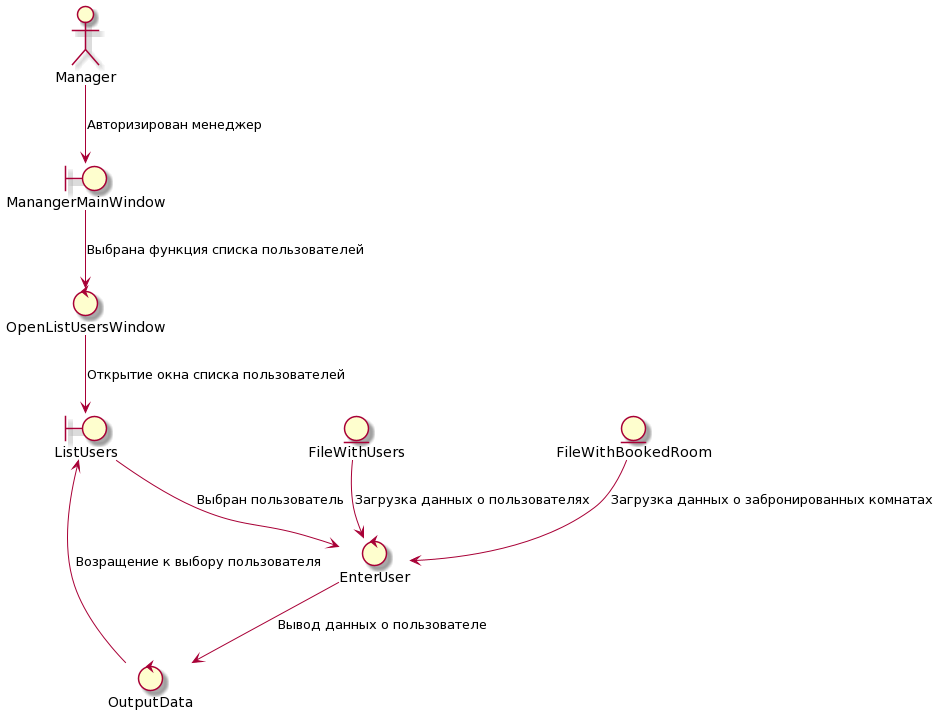
****

Рисунок 2.6 – Диаграмма пригодности для List Users

#### 2.1.2.4 Dialog Booked Room

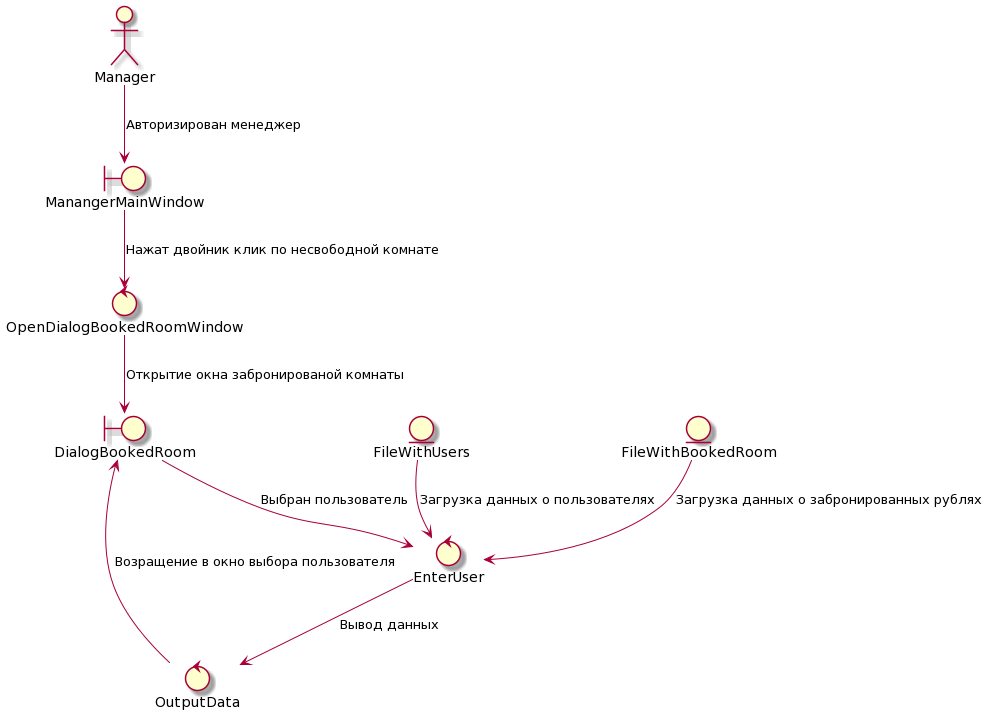
****

Рисунок 2.7 – Диаграмма пригодности для Dialog Booked Room

### 2.1.3 Интерфейс администратора

#### 2.1.3.1 Export

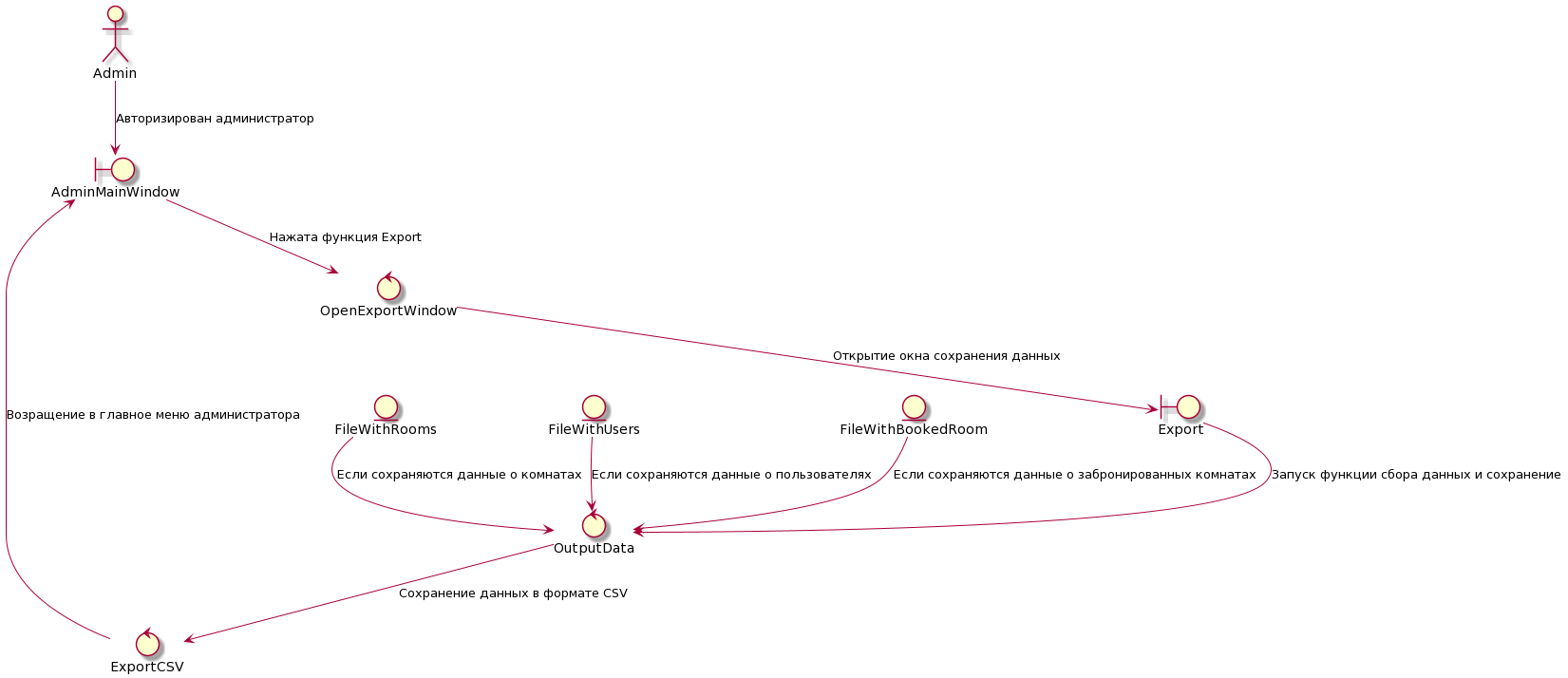
****

Рисунок 2.8 – Диаграмма пригодности для Export

#### 2.1.2.2 Import

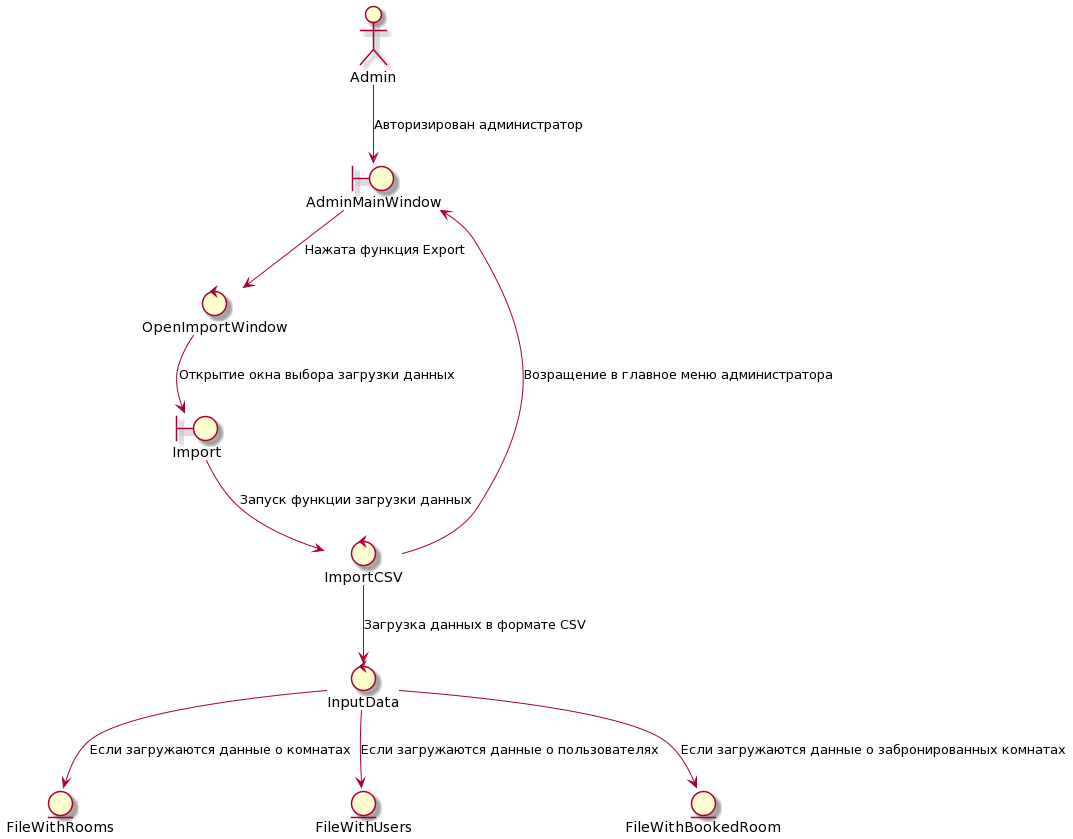


Рисунок 2.9 – Диаграмма пригодности для Import

## 2.2 Разработка диаграммы последовательности

### AuthForm

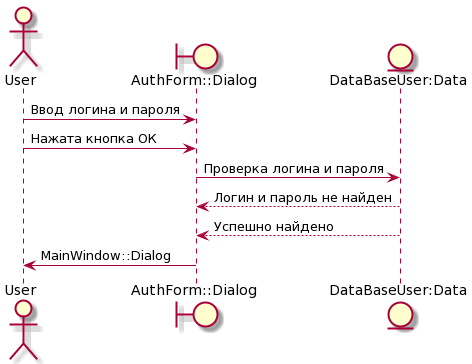
****

Рисунок 2.10 – Диаграмма последовательности для AuthForm

### Recording Users

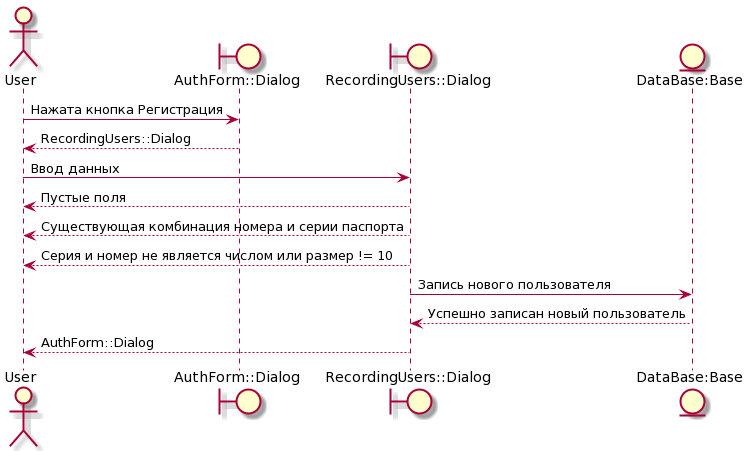
****

Рисунок 2.11 – Диаграмма последовательности для Recording Users

### 2.2.1 Интерфейс клиента

#### 2.2.1.1 Booking Room

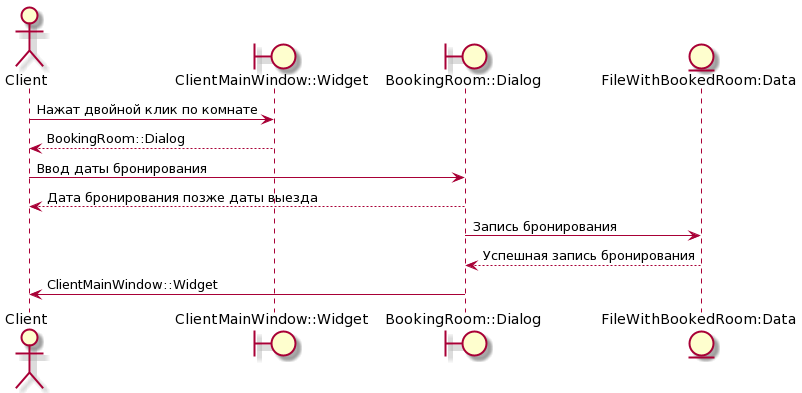
****

Рисунок 2.12 – Диаграмма последовательности для Booking Room

### 2.2.2 Интерфейс руководителя

#### 2.2.2.1 Add Room

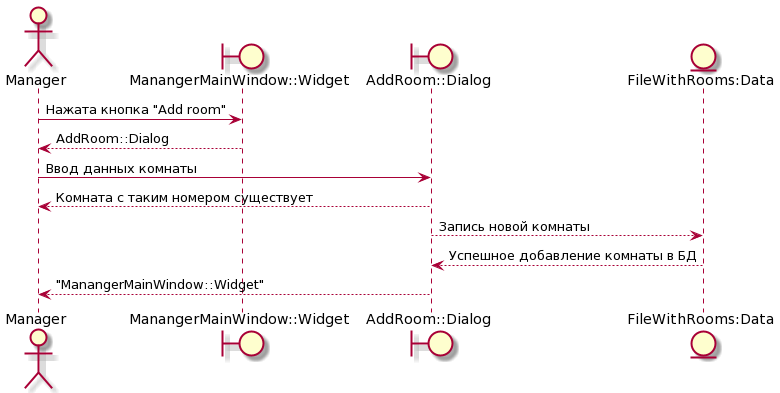
****

Рисунок 2.13 – Диаграмма последовательности для Add Room

#### 2.2.2.2 Edit Room

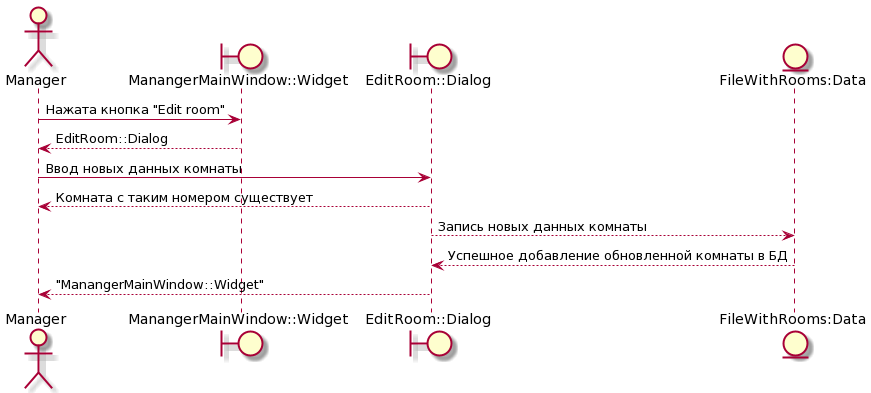
****

Рисунок 2.14 – Диаграмма последовательности для Edit Room

#### 2.2.2.3 List Users

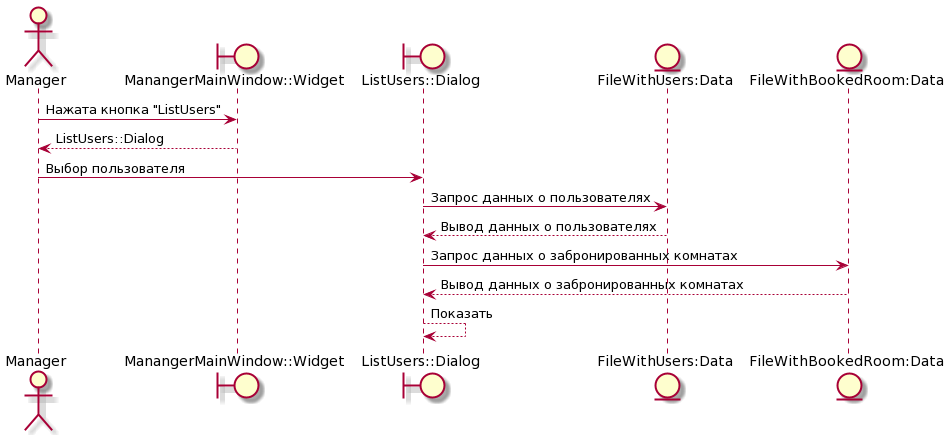
****

Рисунок 2.15 – Диаграмма последовательности для List Users

#### 2.2.2.4 Dialog Booked Room

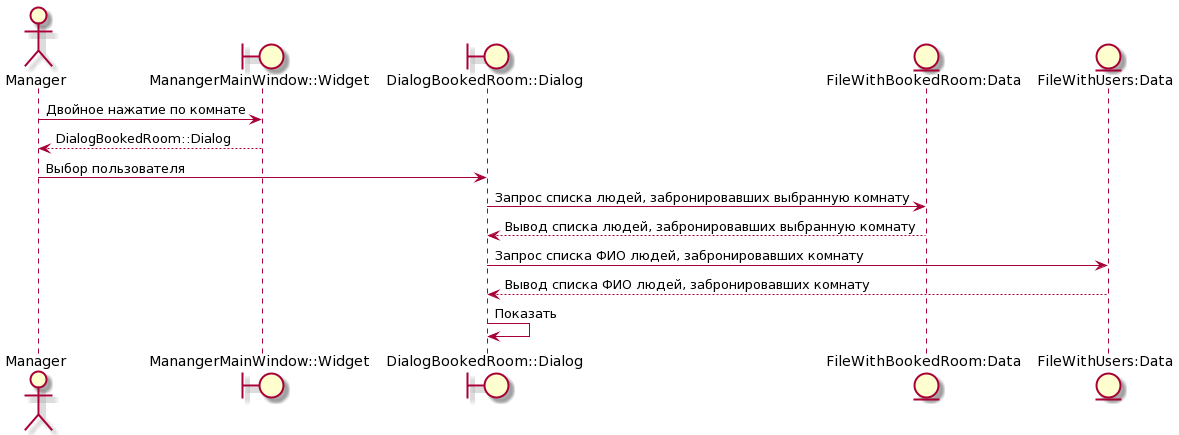
****

Рисунок 2.16 – Диаграмма последовательности для Dialog Booked Room

### 2.2.3 Интерфейс администратора

#### 2.2.3.1 Export

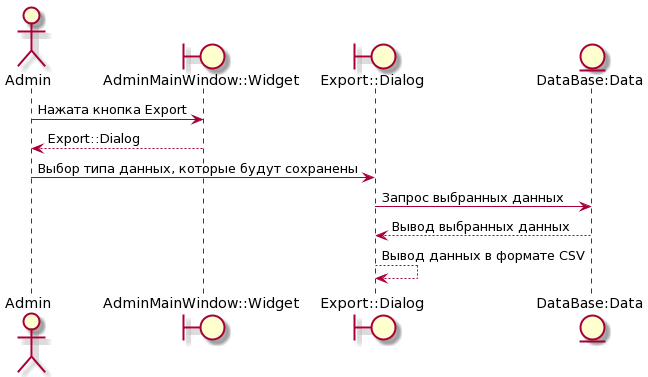
****

Рисунок 2.17 – Диаграмма последовательности для Export

#### 2.2.3.2 Import

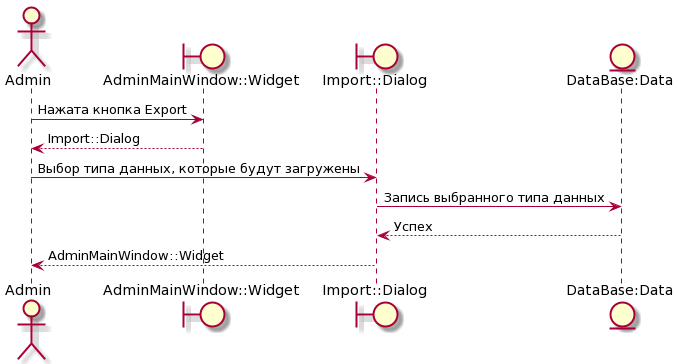


Рисунок 2.18 – Диаграмма последовательности для Import

## 2.2.3 Разработка диаграммы классов уровня проектирования

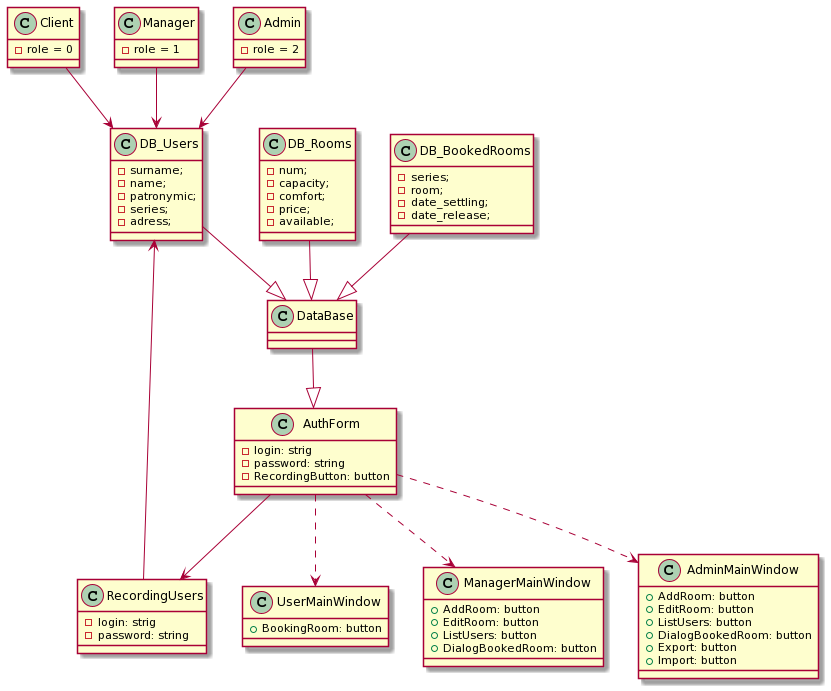


Рисунок 2.19 – Диаграмма классов уровня проектирования

# 3 Объектно-ориентированное программирование

## 3.1 Разработка модульных тестов

### 3.1.1 Проверки ввода данных при регистрации пользователем.

Проверки проводятся на пустые поля, длина серии и номера паспорта и проверка серии и номера паспорта на занятость (рисунок 3.1).



Рисунок 3.1 – Тесты для класса регистрации пользователя

### 3.1.2 Проверки ввода данных при создании или редактировании комнаты

Проверки проводятся на пустые поля (рисунок 3.2), занятость номера комнаты и неправильный ввод значений для номера комнаты, цены, вместимости и комфорта (рисунок 3.3).



Рисунок 3.2 – Тесты на проверку пустых полей при создании комнаты



Рисунок 3.3 – Проверка на неправильный ввод при создании комнаты

## 3.2 Реализация программы

Реализация программы размещена по ссылке <https://github.com/anonumka/hotelBooking>.

## 3.3 Рефакторинг

### 3.3.1 Степень покрытия тестами

Программа покрыта тестами для следующих классов: RecordingUsers и EditRoom, так как от этих классов зависит работа других классов. Проверка этих классов обеспечивает стабильность работы программы.

### 3.3.2 Гибкость архитектуры

Архитектура достаточно гибкая. Внесения в программу каких-либо изменений происходят достаточно просто. Единственные трудности, которые могут появиться, это дополнение полей класса User и Room. В случае, если будут добавлены дополнительные поля, нужно будет добавить в классе user.cpp и room.cpp считывание дополнительных полей из потока QDataStream (также добавить и в выводе/вводе CSV таблиц). В остальном, программа написана так, чтобы любые изменения никак не влияли на другие классы программы.

### 3.3.3 Соответствие стандарту кодирования

Я старался писать код, соответствуя стандарту кодирования [ISO C++](https://isocpp.org/).

## 3.4 Документирование

### 3.4.1 Документация, созданная с помощью систем генерации документов

Прикреплен к отчету.

### 3.4.2 Инструкция по сборке программы

Процесс сборки на Windows 10 и Windows 11.

Для сборки программы нужно установить [Qt Creator](https://www.qt.io/download) версии 6.2.0 и запустить проект. Следом, собираем проект типом *Release*. Заходим в папку, в которой установлен Qt и переходим в папку *bin*. Из папки, в которой собрался проект, берем *lib.exe* и переносим на файл *windeployqt.exe* (рисунок 3.4). Запускаем программу. Если программа пишет, что недостает каких-либо библиотек, то из папки *bin* следует взять данные *DLL* файлы.

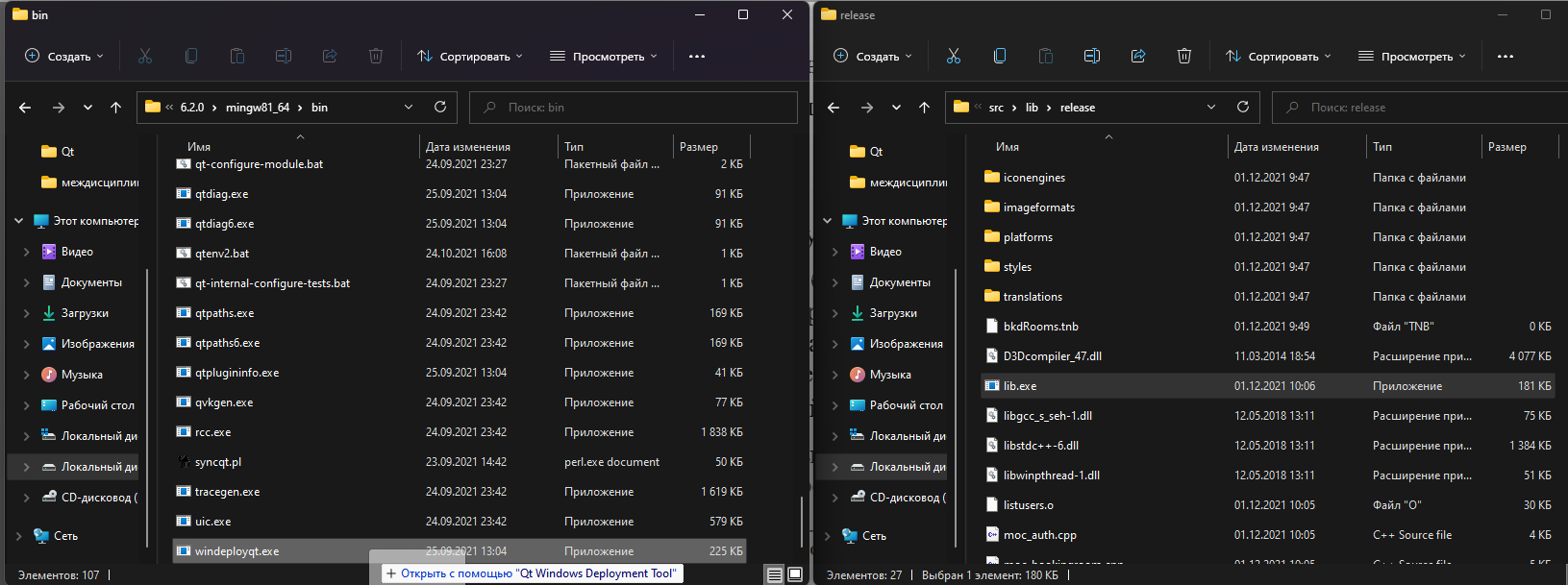


Рисунок 3.4 – Компиляция программы

### 3.4.3 Первый запуск программы

Первый запуск проходит в таком же формате, как и остальные. Первый зарегистрированный пользователь является администратором (рисунок 3.5).



Рисунок 3.5 – Первый запуск

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. СТО 4.2-07-2014. Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Красноярск, 2014.

2.Открытый кроссплатформенный инструмент для построения UML-диаграмм из текстового описания: Диаграмма классов [Электронный ресурс]. – Режим доступа: https://plantuml.com/ru/, вход свободный (2021-12-01).

3. QT : Кроссплатформенная свободная IDE / «QT Development Frameworks».– Режим доступа: https://www.qt.io/, вход свободный (2021-12-01).

4. Doxygen : Система генерации документации / «Dimitri van Heesch».– Режим доступа: https://www.doxygen.nl/index.html/, вход свободный (2021-12-01).