Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Информационная система «Театр»

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

подпись, дата

Студент КИ19-06Б, 031940417 \_\_\_\_\_\_\_\_\_\_ А. В. Шнайдер

подпись, дата

Красноярск 2020

СОДЕРЖАНИЕ

1. Разработка спецификации требований 3

**1.1. Описание входных и выходных данных** 3

**1.2 Требования к составу выполняемых функций и макет интерфейса** 4

**1.3 Стадии и этапы разработки** 13

**2. Проектирование** 14

**2.1 Диаграммы пригодности** 14

**2.2 Диаграммы последовательности** 20

**2.3 Диаграмма классов** 25

3. Программирование 28

**3.2 Реализация программы** 28

**3.3 Рефакторинг** 28

4. Заключение 29

# **1. Разработка спецификации требований**

Информационная система «Театр» предназначена для организации списка актеров и выступлений в театре, а также продажи билетов для посещения этих выступлений.

Информационная система «Театр» должна удовлетворять следующим требованиям:

1. Данные хранить в двоичных файлах. При обработке использовать динамические списки. Ввод и вывод данных осуществлять только через пользовательский интерфейс разрабатываемой системы.

2. Предусмотреть контроль правильности введенных данных.

3. Предусмотреть поиск и сортировку данных по некоторым параметрам.

4. Предусмотреть интуитивно понятный интерфейс, использующий меню (графическое или текстовое) и разделение прав различных групп пользователей.

5. Для организации меню использовать указатели на функции.

Информационная система предусматривает следующие функции:

* Коррекция списка актеров и списка спектаклей;
* Составление и коррекция афиши;
* Продажа билетов на спектакли;
* Выведение справки (участие актеров в различных постановках, информация по спектаклям и др.).

Информационная система предусматривает 3 уровня доступа для пользователей: администратор, кассир и зритель. Все пользователи зарегистрированы в системе и имеют логин и пароль. Администратор имеет права на изменение информации об актёрах и спектаклях, а также регистрацию новых пользователей. Существует первичный пользователь-администратор, логин и пароль которого прописаны прямо в коде программы. Кассир может только оформлять продажу билетов зрителям. Зрители могут просматривать информацию об актёрах, предстоящих спектаклях и своих билетах.

## **1.1. Описание входных и выходных данных**

Строка представляется типом QString.

При сохранении в двоичный файл класс QString помещает сначала длину строки в байтах (quint32), за которой следуют данные в UTF-16.

Число – 32-битное целое число со знаком.

В качестве входных и выходных данных выступают файлы:

users.bin – бинарный файл с данными о пассажирах:

- логин: строка;

- пароль: строка;

- уровень доступа: число от 1 до 3, где 1 – администратор, 2 – кассир, 3 – зритель.

viewer.bin – бинарный файл с данными о билетах зрителей, состоящий из строк вида:

- логин: строка;

- количество билетов: число;

- список билетов: название спектакля (строка), дата и время спектакля (строка вида чч:мм ДД.ММ.ГГГГ), количество купленных мест (число) и тип мест (строка).

performance.bin – бинарный файл с данными о спектаклях:

- автор пьесы: строка;

- название спектакля: строка;

- постановщик: строка;

- список ролей и актёров: строка вида «актёр - роль, актёр - роль …»;

- время проведения спектакля: строка чисел вида «мм:чч ДД.ММ.ГГГГ, мм:чч ДД.ММ.ГГГГ …».

actors.bin – бинарный файл с данными об актёрах:

- ФИО актёра – строка;

- дата рождения актёра – строка чисел вида ДД.ММ.ГГГГ;

- роли актёра – строка вида «название спектакля – роль, название спектакля – роль…».

## **1.2 Требования к составу выполняемых функций и макет интерфейса**

На рисунке 1 представлена диаграмма прецендентов.

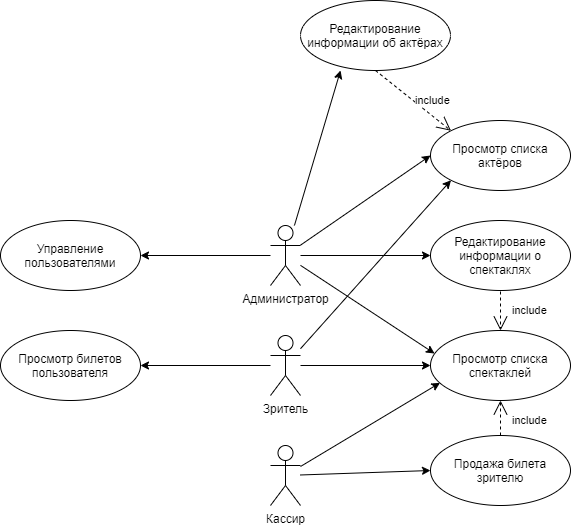


Рисунок 1 – Диаграмма прецедентов

Название прецедента: Управление пользователями

Действующее лицо: Администратор

Цель: Внести данные о новом пользователе в систему или удалить данные о пользователе

Предусловие: Наличие пользователя, которого необходимо зарегистрировать или удалить; администратор уже выполнил вход в систему

Главная последовательность:

1. При нажатии в главном окне (рисунок 2) кнопки «Список пользователей» открывается окно со списком пользователей системой (рисунок 3)



Рисунок 2 – Главное окно

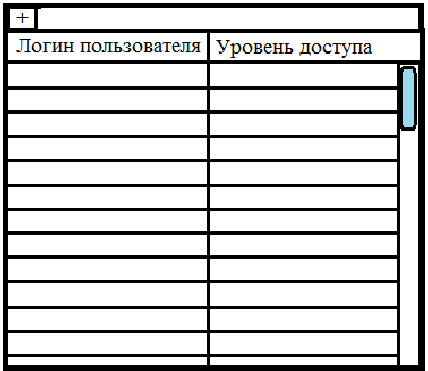


Рисунок 3 – Окно списка пользователей

2. Для добавления нового пользователя нужно нажать кнопку «+» в открывшемся окне. После этого откроется окно регистрации (рисунок 4).

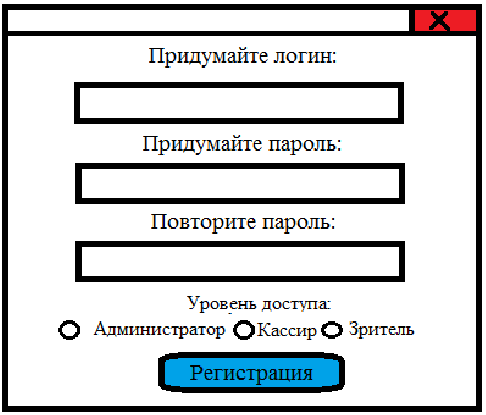


Рисунок 4 – Окно регистрации

3. Вводятся данные (логин и пароль, а также нужный уровень доступа).

4. После нажатия кнопки «Регистрация» проверяется правильность введённых данных. При неправильно введённых данных (уже существует пользователь с таким логином или поле пароля пусто) выводится окно с предупреждением об этом. Если же данные введены правильно, они вносятся в файл с логинами и паролями пользователей, а окно регистрации закрывается.

5. Для удаления пользователя в окне списка пользователей нужно дважды щёлкнуть левой кнопкой мыши по строке нужного пользователя. После этого откроется окно с подтверждением удаления.

6. Чтобы сохранить изменения, необходимо нажать кнопку сохранения в верхней части окна.

Название прецедента: Просмотр билетов пользователя

Действующее лицо: Зритель

Цель: Просмотреть билеты пользователя

Предусловие: Зритель выполнил вход в систему

Главная последовательность:

1. В главном окне при нажатии кнопки «Просмотреть список купленных билетов» открывается окно со списком билетов пользователя (рисунок 5).

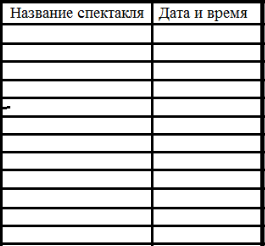


Рисунок 5 – Окно списка билетов

2. При двойном нажатии левой кнопкой мыши по интересующей строке с билетом будет открыто окно с подробной информацией о билете (рисунок 6).

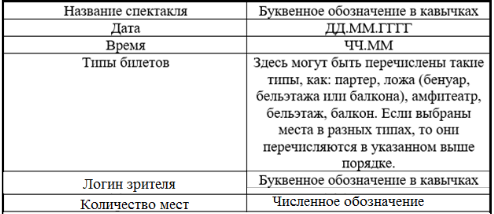


Рисунок 6 – Окно подробной информации о билете

Название прецедента: Просмотр списка спектаклей

Действующее лицо: Любой пользователь

Цель: Просмотреть список спектаклей

Предусловие: Пользователь должен выполнить вход в систему

Главная последовательность:

1. В главном окне при нажатии кнопки «Список спектаклей и покупка билетов» открывается окно со списком запланированных спектаклей (рисунок 7)

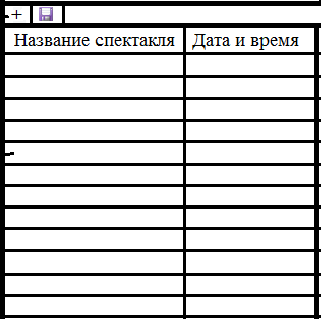


Рисунок 7 – Окно списка спектаклей

Название прецедента: Продажа билета зрителю

Действующее лицо: Кассир

Цель: Оформить билет для зрителя

Предусловие: Кассир должен выполнить вход в систему; должен существовать зритель с желанием купить билет; зритель должен быть зарегистрирован

Главная последовательность:

1. В главном окне при нажатии кнопки «Список спектаклей и покупка билетов» открывается окно со списком запланированных спектаклей.

2. При двойном нажатии левой кнопкой мыши по выбранной строке открывается окно с выбором типов билетов и количества мест (рисунок 8).

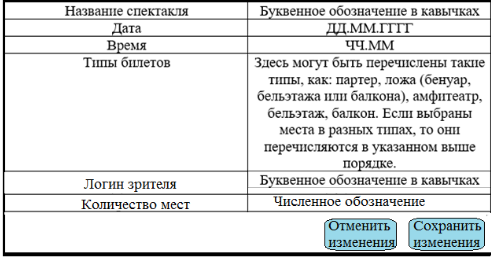


Рисунок 8 – Окно продажи билета

3. Вписывается логин пользователя в соответствующее поле. Заполняются остальные поля.

4. При нажатии кнопки «Сохранить изменения» данные о билете вносятся в файл данных о пользователях, а окно для покупки билета закрывается. Если нажимается кнопка «Отменить изменения», то данные о билете не вносятся в файл.

Название прецедента: Редактирование информации о спектаклях

Действующее лицо: Администратор

Цель: Редактировать данные о спектаклях

Предусловие: Администратор должен выполнить вход в систему

Главная последовательность:

1. В главном окне при нажатии кнопки «Список спектаклей и покупка билетов» открывается окно со списком запланированных спектаклей.

2. При двойном нажатии левой кнопкой мыши по выбранной ячейке с названием спектакля открывается окно с подробной информацией о спектакле (рисунок 9). Там же выполняется редактирование информации. Редактирование даты спектаклей производится без открытия данного окна, прямо в окне списка спектаклей.

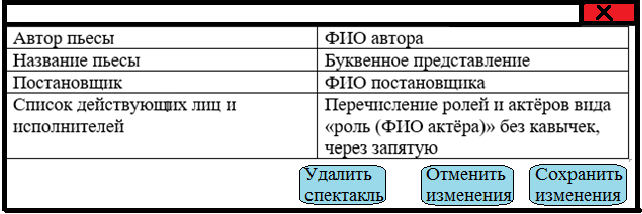


Рисунок 9 - Окно подробной информации о спектакле

3. Для добавления нового спектакля нужно нажать соответствующую кнопку в окне списка спектаклей.

Название прецедента: Просмотр списка актёров

Действующее лицо: Администратор или зритель

Цель: Просмотреть список актёров

Предусловие: Администратор или зритель должен выполнить вход в систему

Главная последовательность:

1. В главном окне при нажатии кнопки «Список актёров» откроется окно списка актёров (рисунок 10).



Рисунок 10 – Окно списка актёров

Название прецедента: Редактирование информации об актёрах

Действующее лицо: Администратор

Цель: Добавить информацию о новом актёре или изменить её у уже существующего

Предусловие: Администратор или должен выполнить вход в систему

Главная последовательность:

1. В главном окне при нажатии кнопки «Список актёров» откроется окно списка актёров.

2. Двойное нажатие левой кнопкой мыши по полю ФИО актёра откроет окно редактирования информации об актёре (рисунок 11).

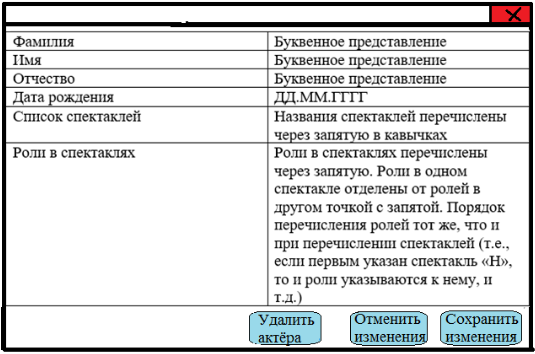


Рисунок 11 – Окно редактирования информации об актёре

3. После редактирования информации об актёре при нажатии кнопки «Сохранить изменения» данные об актёре будут перезаписаны. При нажатии кнопки «Отменить изменения» окно редактирования закроется, а данные не изменятся. При нажатии кнопки «Удалить актёра» все данные об актёре будут удалены.

## **1.3 Стадии и этапы разработки**

Этапы разработки и сроки приведены в таблице 1

Таблица 1 – Этапы разработки

|  |  |
| --- | --- |
| Этапы разработки | Сроки |
| Реализация списка пользователей и четырёх окон (авторизация, регистрация, главное окно, список пользователей) | 28.09.2020 – 18.10.2020 |
| Реализация списка спектаклей и двух окон (список спектаклей, добавление/редактирование спектакля) | 19.10.2020 – 08.11.2020 |
| Реализация списка актёров и двух окон (список актёров, редактирование/добавление информации об актёре) | 09.11.2020 – 29.11.2020. |
| Реализация списка билетов и двух окон (покупка билета / информация о билете, список билетов пользователя) | 30.11.2020 – 20.12.2020 |
| Составление пояснительной записки | 21.12.2020 – 30.12.2020 |

## **2. Проектирование**

## **2.1 Диаграммы пригодности**

Согласно указанным в пункте 1.2 последовательностям были разработаны диаграммы пригодности с помощью онлайн-сервиса Drawio. Диаграммы пригодности для открытия списков актёров и спектаклей представлены на рисунках 12-13.



Рисунок 12 – Диаграмма пригодности прецедента «Просмотр списка актёров»



Рисунок 13 – Диаграмма пригодности прецедента «Просмотр списка спектаклей»

Диаграммы пригодности для прецедентов редактирования различной информации представлены на рисунках 14-16.

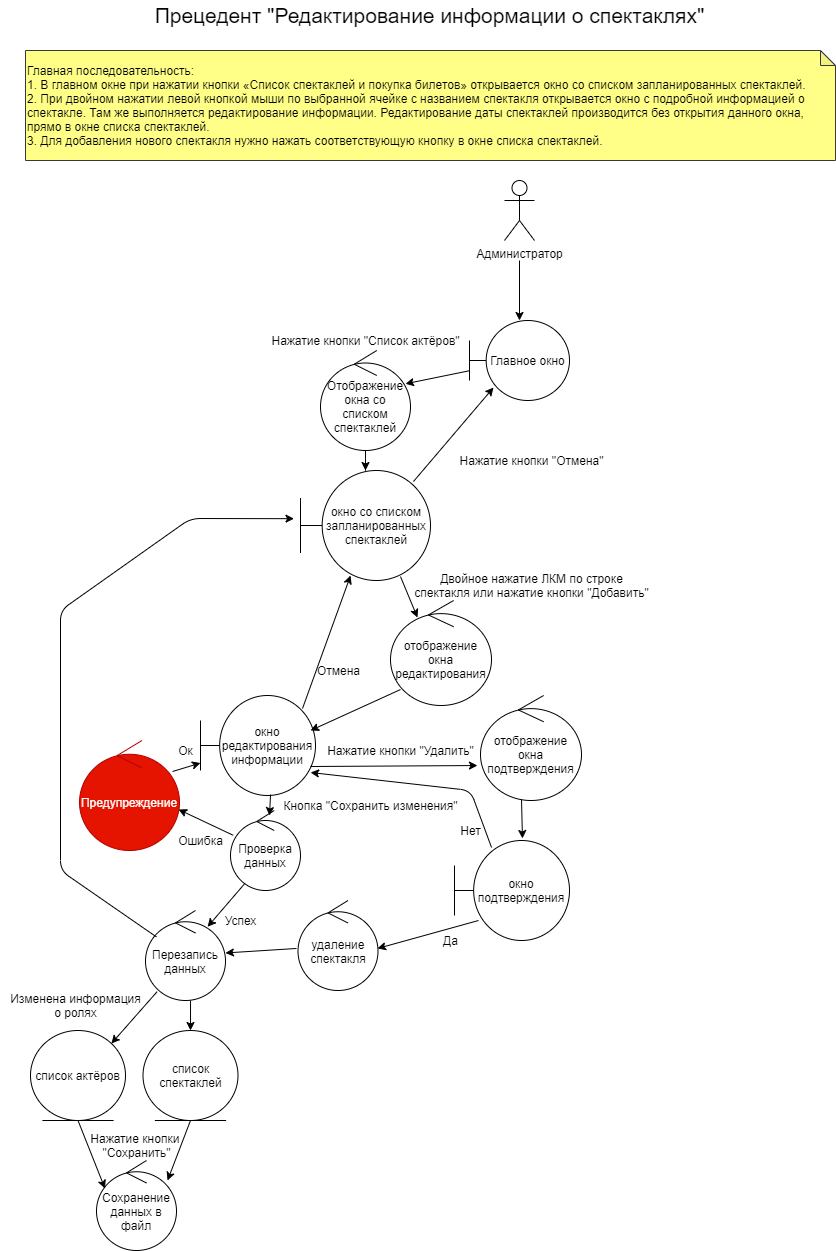


Рисунок 14 - Диаграмма пригодности прецедента «Редактирование информации о спектаклях»

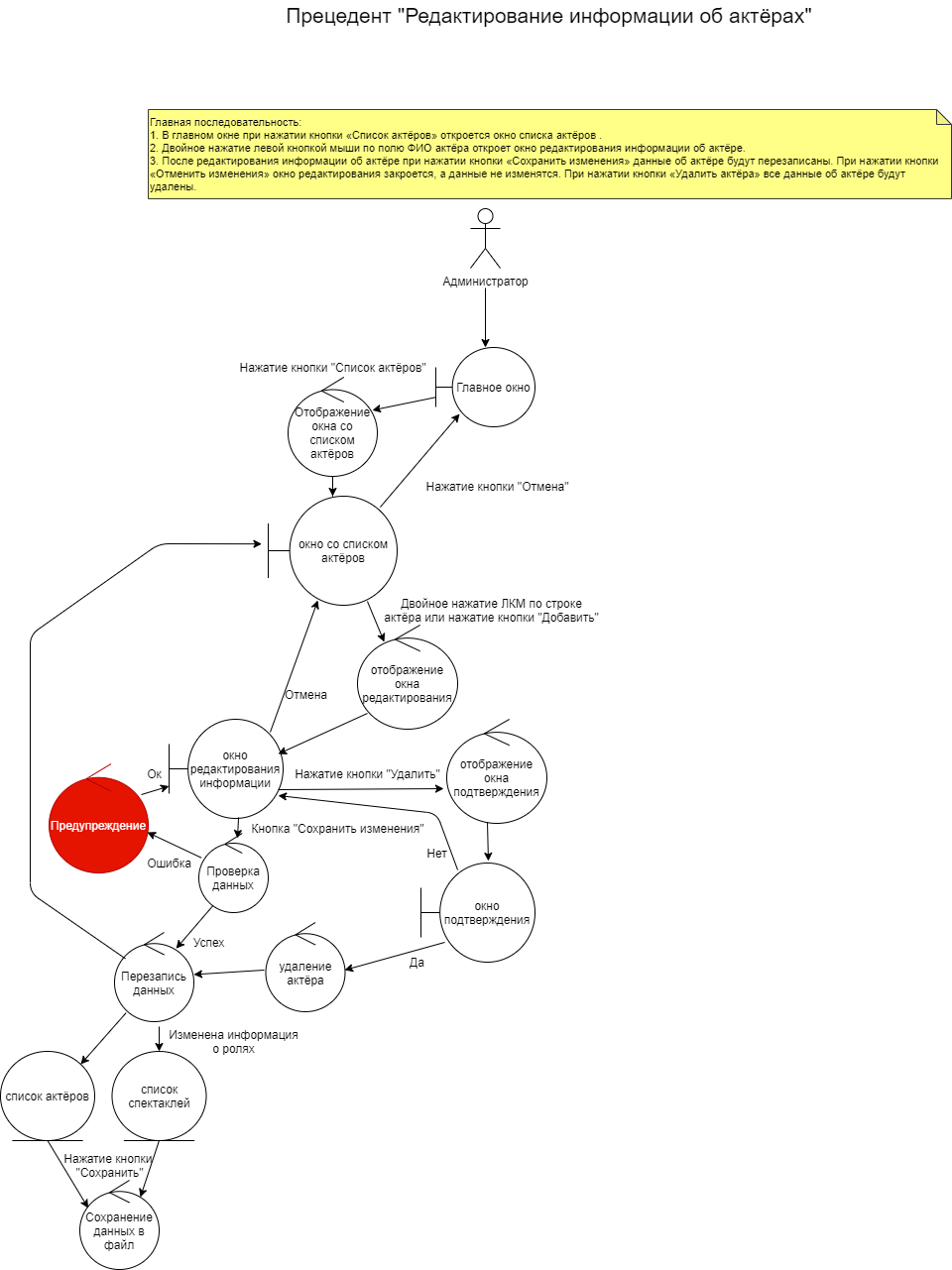


Рисунок 15 - Диаграмма пригодности прецедента «Редактирование информации об актёрах»

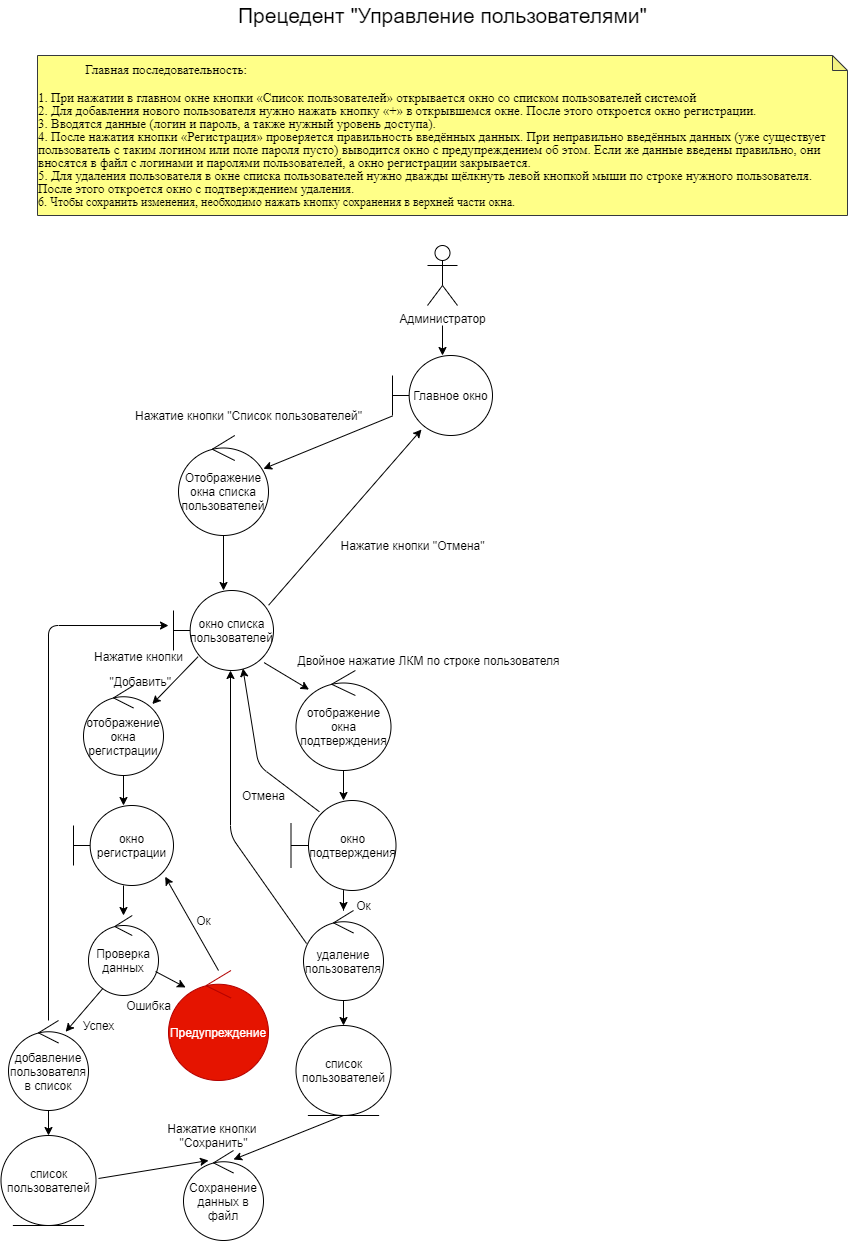


Рисунок 16 - Диаграмма пригодности прецедента «Управление пользователями»

Диаграммы пригодности для продажи билета зрителю представлена на рисунке 17, а для просмотра купленных билетов на рисунке 18.

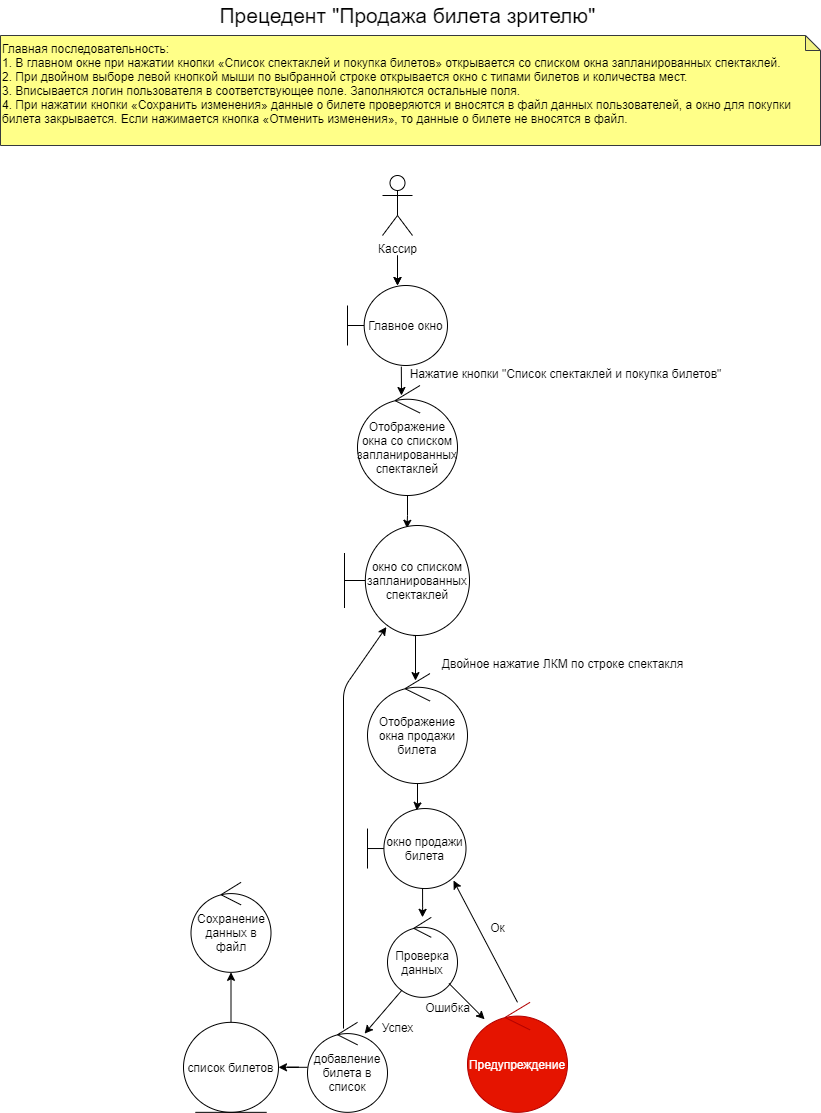


Рисунок 17 - Диаграмма пригодности прецедента «Продажа билета зрителю»

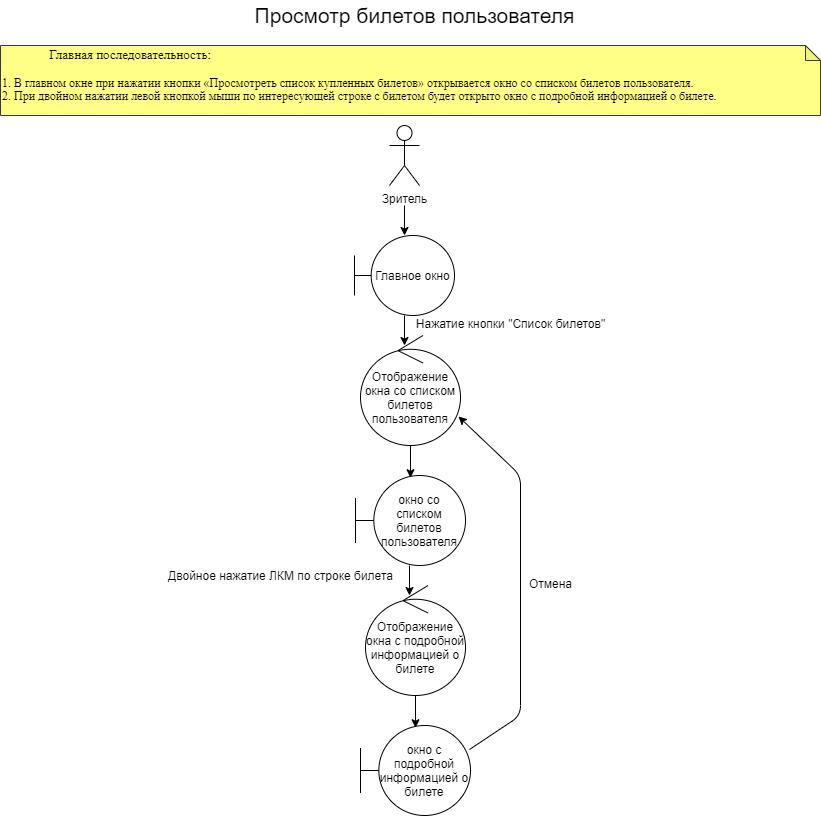


Рисунок 18 - Диаграмма пригодности прецедента «Просмотр билетов пользователя»

## **2.2 Диаграммы последовательности**

В соответствии с построенными ранее диаграммами пригодности были разработаны и диаграммы последовательности, наглядно показывающие работу каждого из прецедентов. В отличии от диаграмм пригодности, диаграммы последовательности были разработаны при помощи онлайн-сервиса PlantumUML. Диаграммы последовательности для прецедентов просмотра списков актёров и представлений показаны на рисунках 19-20.

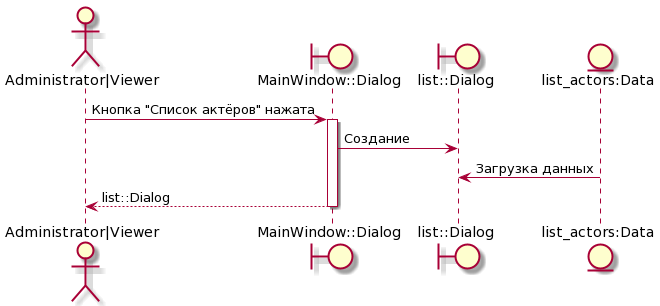


Рисунок 19 – Диаграмма последовательности прецедента «Просмотр списка актёров»

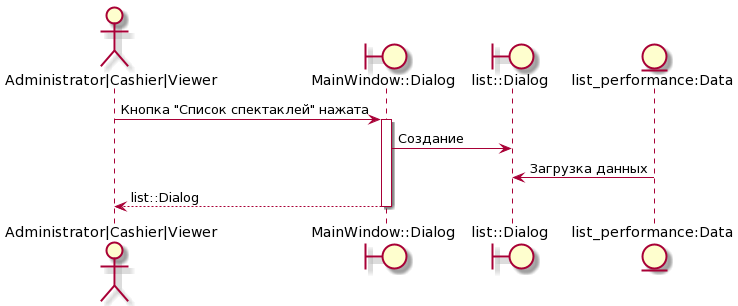


Рисунок 20 – Диаграмма последовательности прецедента «Просмотр списка спектаклей»

Диаграммы последовательности для прецедентов редактирования различной информации представлены на рисунках 21-23.

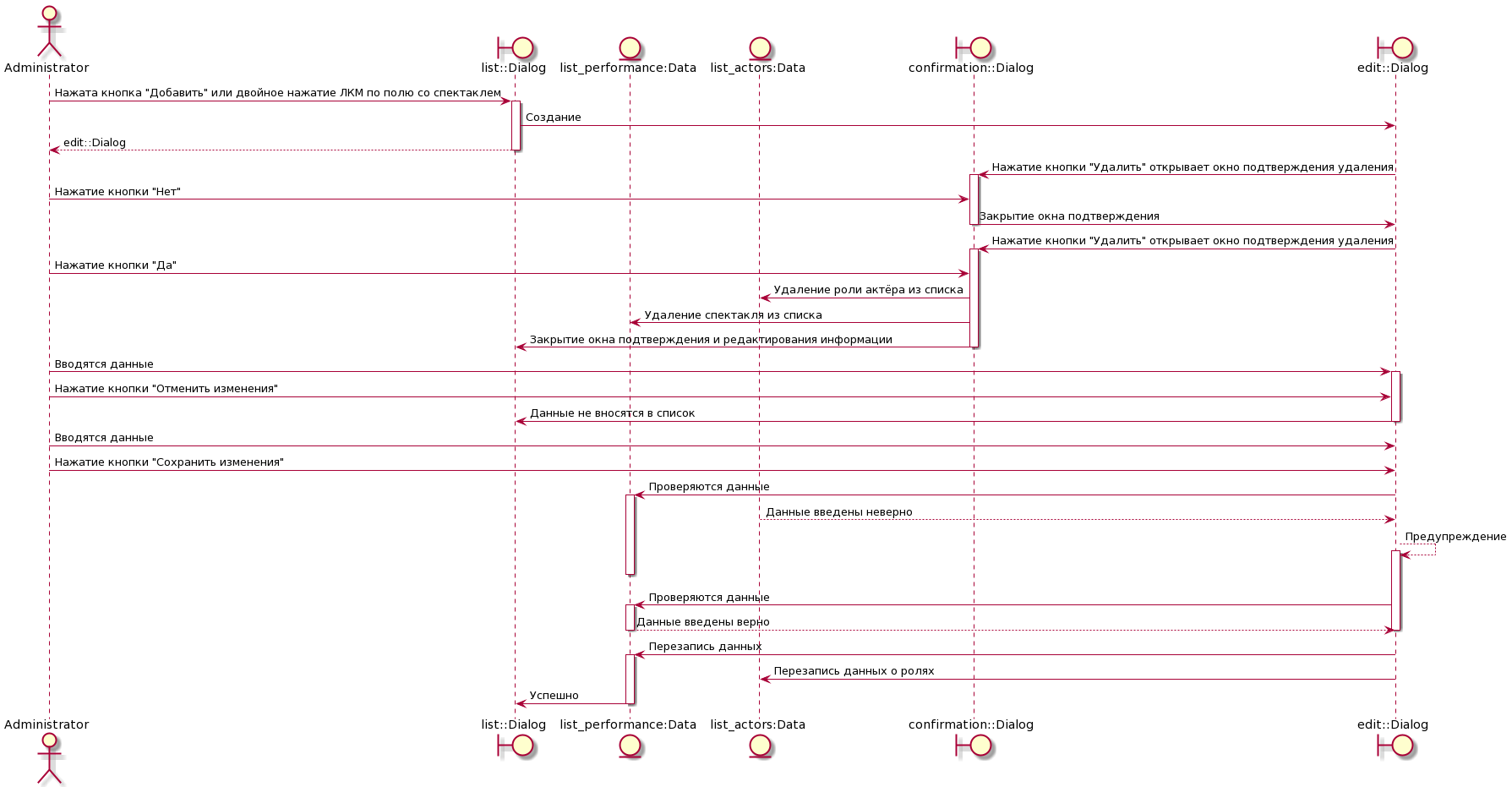


Рисунок 21 - Диаграмма последовательности прецедента «Редактирование информации о спектаклях»

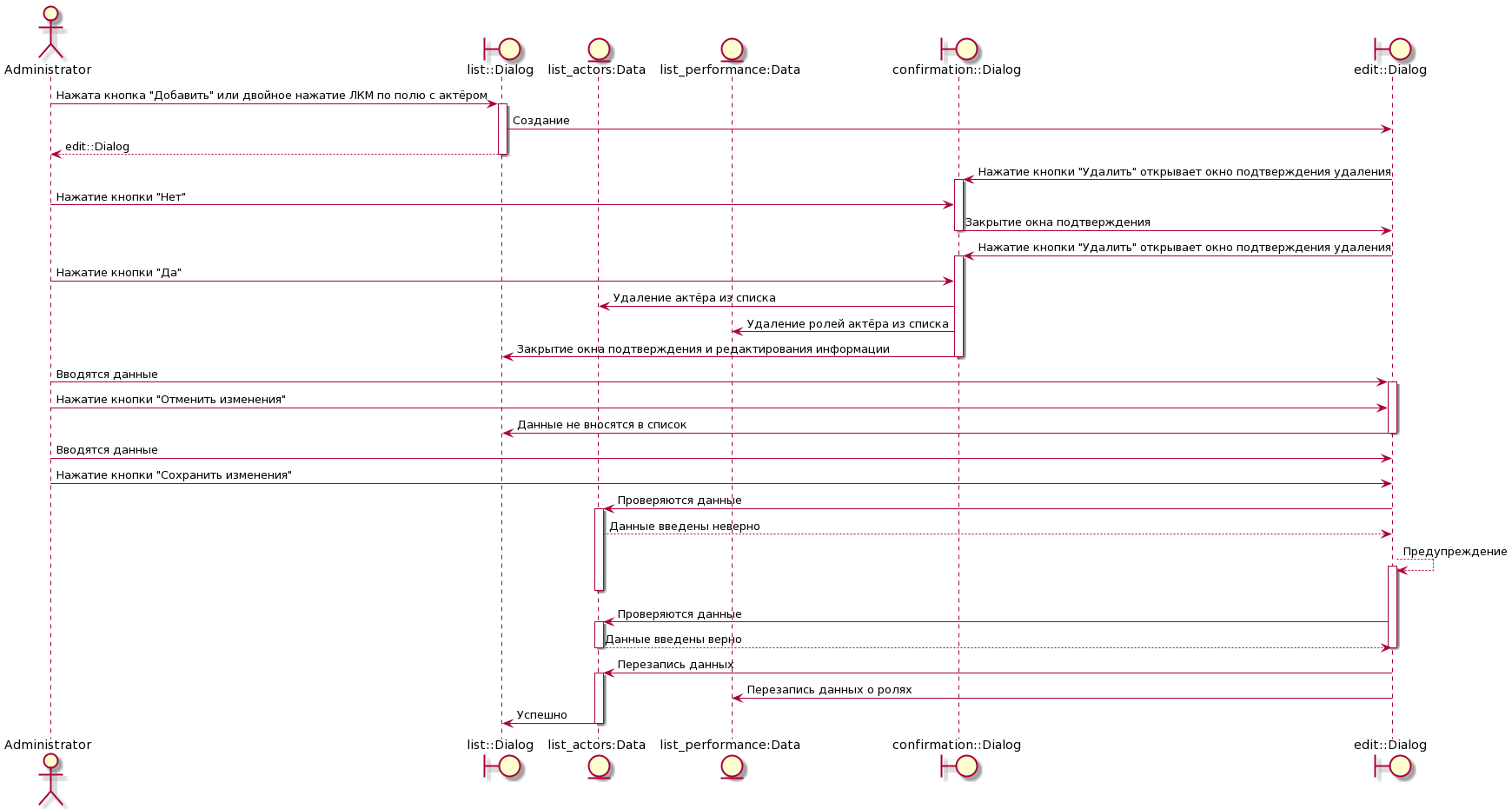


Рисунок 22 - Диаграмма последовательности прецедента «Редактирование информации об актёрах»

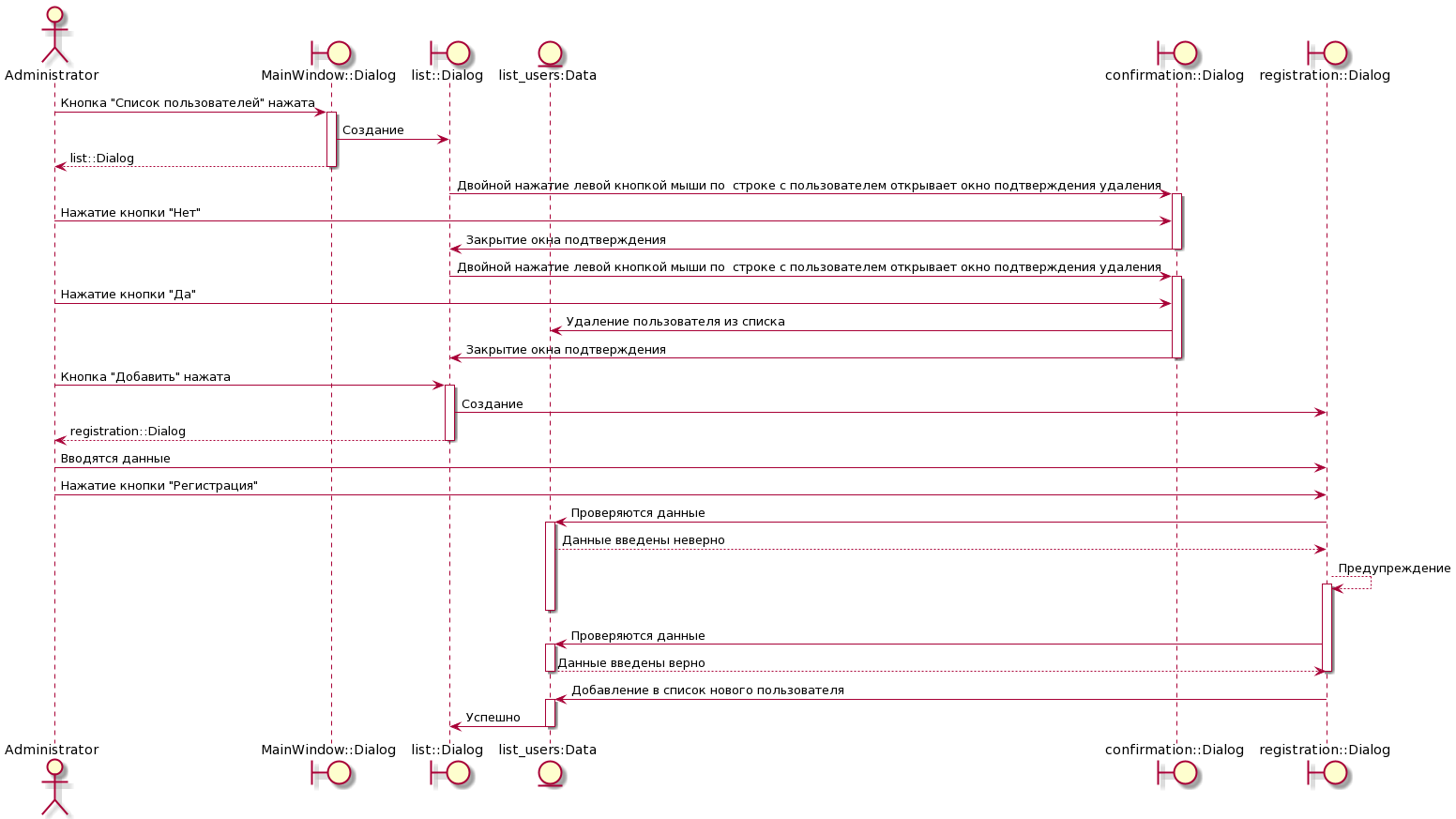


Рисунок 23 - Диаграмма последовательности прецедента «Управление пользователями»

Диаграммы последовательности для продажи билета зрителю представлена на рисунке 24, а для просмотра купленных билетов на рисунке 25.

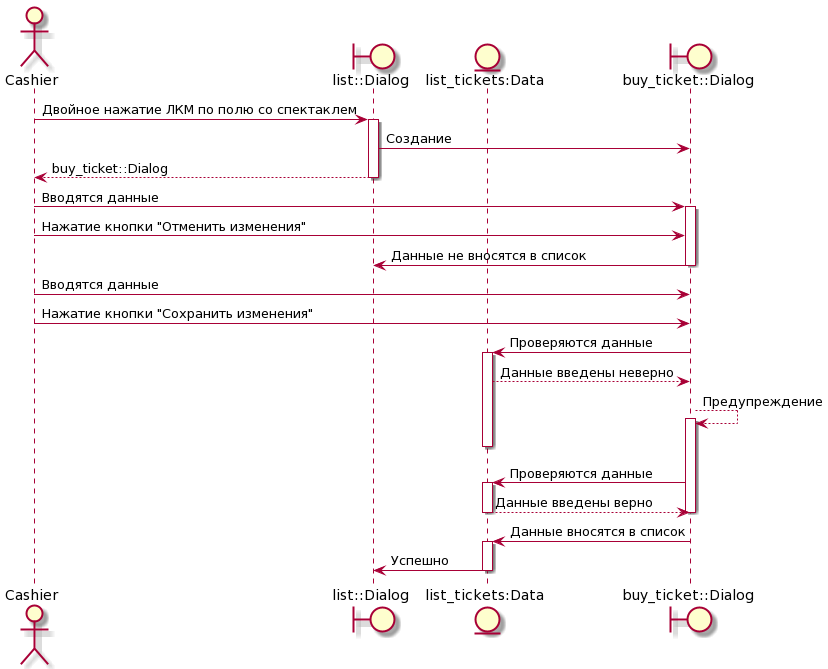


Рисунок 24 - Диаграмма последовательности прецедента «Продажа билета зрителю»

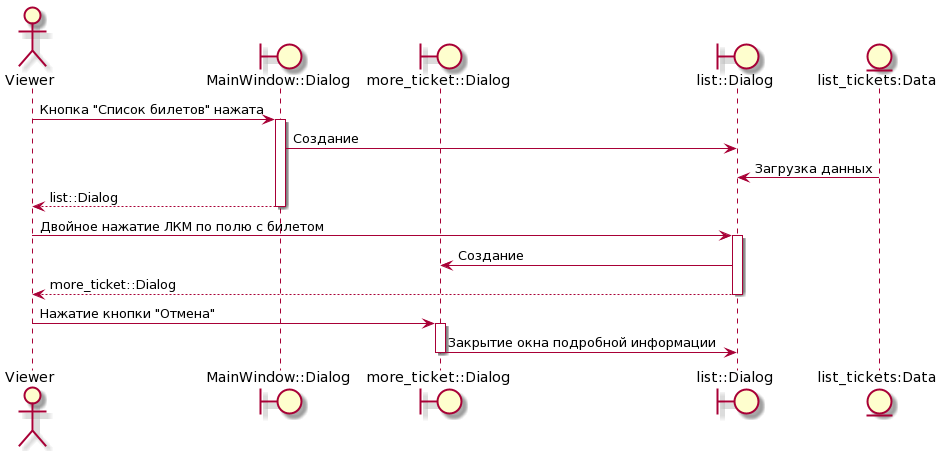


Рисунок 25 - Диаграмма последовательности прецедента «Просмотр билетов пользователя»

## **2.3 Диаграмма классов**

Для объединения всех разработанных ранее диаграмм в единую систему была спроектирована диаграмма классов (рисунок 26). На ней наглядно показаны связи между классами программы.

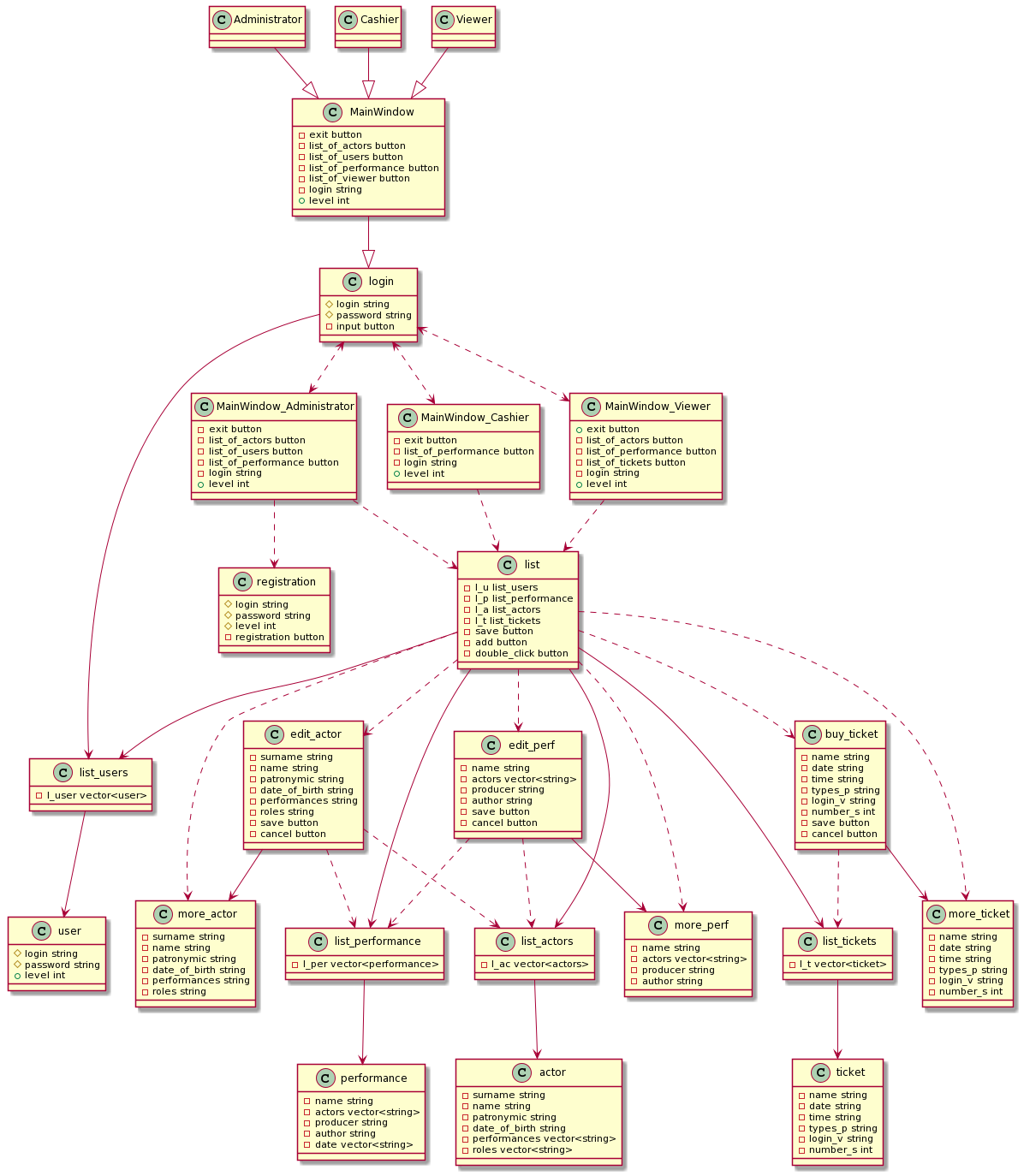


Рисунок 26 – Диаграмма классов

С виду диаграмма классов может показаться запутанной, но принцип работы итоговой программы прост. Т.к. в информационной системе необходимо отображать множество различной информации, было решено использовать так называемые «списки». Для хранения информации об актёрах, пользователях, выступлениях и билетах были созданы специальные классы: actor, user, performance и ticket соответственно. А т.к. каждый класс описывает лишь один объект (актёра, представление и т.д.), то для хранения всей информации были созданы вышестоящие классы, которые использовали вектора с упомянутыми выше классами. Получившиеся же классы-списки уже отображаются при открытии соответствующего окна программы. Т.к. информация в списках взаимосвязана (например, актёры связаны с представлениями ролями), то и такие списки изменяются совместно, если это требуется. Также, в режиме зрителя можно узнать подробную информацию о том, или ином объекте (актёре, представлении, билете). Для предоставления этой информации используются те же окна, что и для редактирования (добавления) записи, но возможности изменения отключены.

# **3. Программирование**

Для реализации программы была использована среда разработки «QtCreator».

## **3.2 Реализация программы**

Ссылка на репозиторий с реализованной программой находится в списке использованных источников. [[1]](#_[1]_-_https://github.com/Shnayder10)

В процессе написания программы я был вынужден немного отклониться от намеченного в процессе проектирования курса – так, например, в классах “actor” и “more\_actor” (“edit\_actor”) данные о ролях в спектаклях было решено представлять одной строкой, чтобы не усложнять отображение этих данных в таблице. Аналогично было решено поступить с датой и временем спектакля в классе “ticket”. В классе “performance” было решено при создании объекта автоматически добавлять 5 дат проведения спектакля. Впоследствии, именно эти 5 дат администратор редактирует.

## **3.3 Рефакторинг**

В процессе написания программы я старался придерживаться соглашению о кодировании от компании Google. В процессе рефакторинга был улучшен визуальный вид кода и исправлены названия некоторых переменных.

# **4. Заключение**

Во время выполнения работы были проведены следующие работы:

* Разработано техническое задание;
* Созданы динамическая и статическая модель системы;
* Реализована информационная система «Театр»;
* Проведены тесты разработанной программы;
* Оформлена пояснительная записка.

В процессе выполнения работы над проектом я улучшил свои навыки разработки программ (и тестов к ним) и написания сопутствующих им документам.

**Список использованных источников**

## **[1] - https://github.com/Shnayder102/teatr**