Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

**КУРСОВАЯ РАБОТА**

Информационная система «Театр»

Руководитель \_\_\_\_\_\_\_\_\_\_ В.С. Васильев

подпись, дата

Студент КИ19–08Б, 031940570 \_\_\_\_\_\_\_\_\_\_ И.В. Минин

подпись, дата

Красноярск 2020

РЕФЕРАТ

Информационная система «Театр». Информация: актеры – фамилия, имя, отчество, дата рождения, список спектаклей, в которых занят актер, роли, исполняемые в данных спектаклях; спектакли – автор пьесы, название, постановщик, список действующих лиц и исполнителей; афиша – название спектакля, дата, время; билеты – дата, количество билетов разного типа (партер, ложи, балкон и др.). Операции: коррекция списка актеров и списка спектаклей; составление и коррекция афиши; продажа билетов на спектакли; справки (участие актеров в различных постановках, информация по спектаклям и др.). Предусмотреть следующие группы пользователей: администратор, кассир, зритель.

СОДЕРЖАНИЕ

[Введение 4](#_Toc60144977)

[1 Разработка технического задания 5](#_Toc60144978)

[1.1 Модель программы 7](#_Toc60144979)

[1.2 Описание пользовательского интерфейса 7](#_Toc60144980)

[1.3 Функциональное назначение 9](#_Toc60144981)

[1.4 Входные и выходные данные 12](#_Toc60144982)

[1.5 Стадии и этапы разработки 13](#_Toc60144982)

[2 Подходы к решению поставленной задачи 12](#_Toc60144983)

[2.1 Предметная область 12](#_Toc60144984)

[2.2 Входные и выходные данные 13](#_Toc60144985)

[3 Подробное описание алгоритма 16](#_Toc60144987)

[4 Тестирование программы 22](#_Toc60144988)

[5 Инструкция к программе 23](#_Toc60144989)

[6 Заключение 24](#_Toc60144990)

[Список использованных источников 25](#_Toc60144991)

Введение

Система «Театр» предназначается для предоставления текущей информации спектаклях, актерах, билетах. Она помогает клиенту приобрести билет на необходимый спектакль и получить информацию о нем.

# 1 Разработка технического задания

Программа не предусмотрена для покупки билетов, она несет информационный характер о спектаклях. Данная система предназначена для предоставления информации о спектаклях, актерах, билетах, кассирах.

С данной информационной системой работают следующие группы пользователей: администратор, кассир, обычный пользователь. В группу пользователей «обычный пользователь» входят незарегистрированные или не авторизированные пользователи. Кассир, помимо получения информации, может полностью управлять информацией о заказах. Администратор имеет полные права добавления, удаления редактирования любой информации.

Информационная система «Театр» должна удовлетворять следующим требованиям:

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

2. Предусмотреть контроль правильности введенных данных, поиск и сортировку данных по некоторым параметрам.

3. Предусмотреть интуитивно понятный интерфейс, использующий меню и разделение прав различных групп пользователей.

4. Для организации меню использовать указатели на функции.

Разработка данной информационной системы планируется на языке программирования C++: Qt.

# 1.1 Модель программы

Каждый спектакль характеризуется следующими параметрами:

- наименование;

- дата спектакля;

- время спектакля;

Каждый спектакль имеет следующие разделы:

- организационная группа;

- билеты;

Информационная система “Театр” должна обеспечивать:

- добавление информации о спектаклях, билетах и организационной группе;

- удаление информации о спектаклях, билетах и организационной группе;

С данной информационной системой должны работать следующие группы пользователей:

- администратор;

- кассир;

- пользователь.

Представляя нашу информационную систему можно выделить данную поверхностную диаграмму классов:

Диаграмма классов указана на рисунке 1.

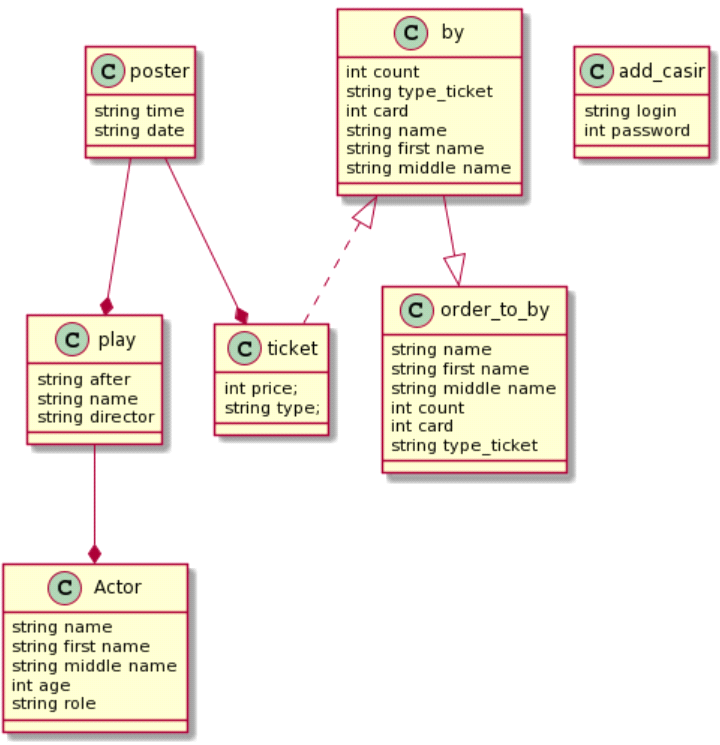


Рисунок 1 – Поверхностная диаграмма классов

# 1.2 Описание пользовательского интерфейса

Пользовательский интерфейс программы:

1) Главное окно, в котором отображается афиша, со спектаклями.

2) Окно авторизации, которое позволяет зарегистрироваться или авторизоваться, если пользователь еще не вошел в систему.

3) Окно организационной группы, которое показывает актеров, сценаристов и режиссеров.

4) Окно информации о билете, его цена, тип и количество свободных экземпляров.

5) Окно информации о существующих кассирах, поля логин и пароль, есть возможность добавлять удалять и редактировать информацию.

5) Окно информации о текущих ордерах.

Макеты пользовательского интерфейса:

Макеты окон пользовательского интерфейса указаны на рисунках: 2, 3, 4, 5, 6.

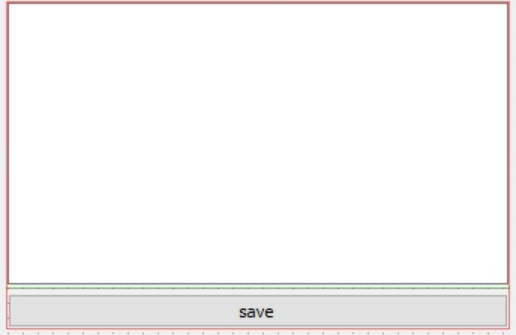


Рисунок 2 – Окно информации о текущей персоне

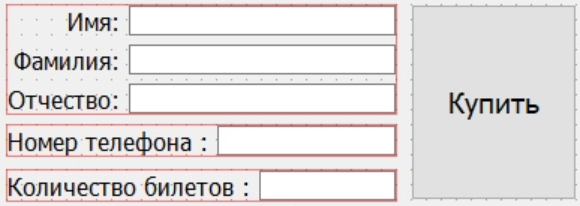


Рисунок 3 – Окно оформления заявки

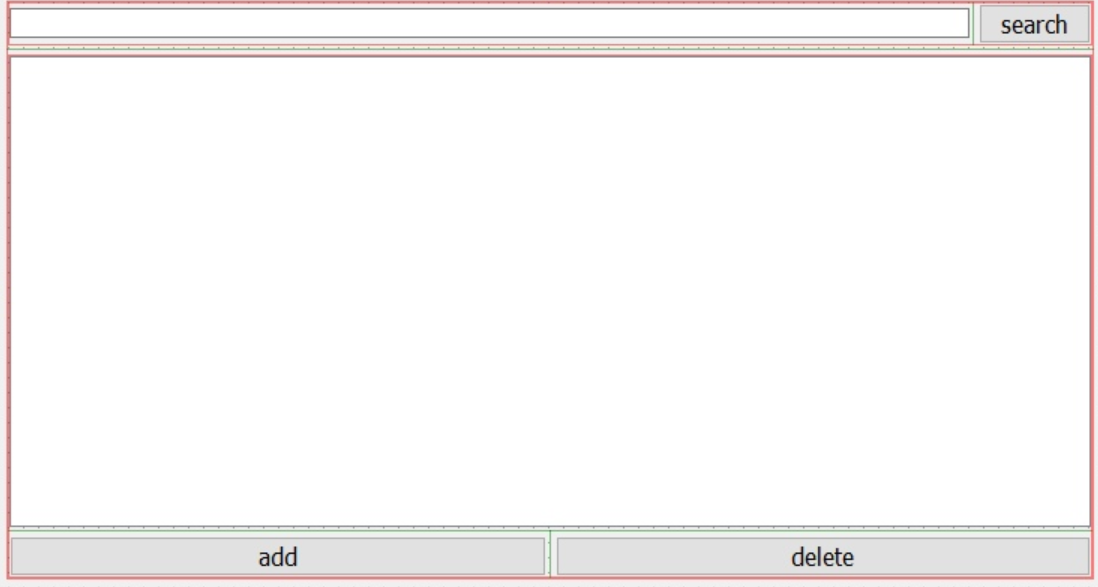


Рисунок 4 – Макет главного окна



Рисунок 5 – Вторичное окно авторизации

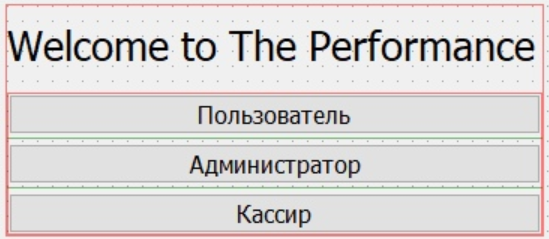


Рисунок 6 – Первичное окно авторизации

# 1.3 Функциональное назначение

Программа будет использоваться в театре тремя группами пользователей: читатель, кассир, администратор.

Эксплуатационное назначение подробно расписано в диаграмме прецедентов, каждый из которых расписан более подробно после рисунка.



Рисунок 7 – Диаграмма прецедентов.

**User**

Название прецедента: **покупка билета**

Действующие лица: пользователь

Предусловия: пользователь нажимает на ячейку “билеты” – общие количество билетов, в открывшемся окне выбирает ячейку с нужным типом билета, всплывает окно покупки билета

Цель: покупка билета

Действия: пользователь вводит ФИО, количество билетов, номер карты, нажимает кнопку купить.

**Admin\Casheir**

Название прецедента: **редактирование спектаклей**

Действующие лица: администратор

Предусловия: администратор нажимает на ячейку спектакля

Цель: редактирование информации о спектакле (постановщик, автор, действующие лица

Действия: нажав двойным кликом на любую из ячеек (постановщик, автор, действующие лица), появится возможность редактировать информацию, нажав на кнопку добавить, появляется строка, в которую можно вписать информацию, выделив строку и нажав на кнопку удалить, можно удалить данную информационную ячейку.

Название прецедента: **редактирование пользователей**

Действующие лица: администратор

Предусловия: администратор нажимает на ячейку спектакля, затем в открывшемся окне, нажимает на ячейку интересующего актера

Цель: редактирование информации об актере

Действия: в открывшемся окне, при двойном нажатии на ячейку актера, появляется возможность редактировать (ФИО, Дату рождения, Спектакли).

Название прецедента: **редактирование количества билетов**

Действующие лица: администратор, кассир

Предусловия: администратор нажимает на ячейку “Билеты”

Цель: изменить количество билетов

Действия: в открывшемся окне, при двойном нажатии на ячейку билета появляется возможность редактирования.

Название прецедента: **редактирование афиши**

Действующие лица: администратор

Предусловия: нет

Цель: редактирование информации (Спектакли, Дата премьеры, Время премьеры)

Действия: в открывшемся окне, при двойном нажатии на ячейку билета, появляется возможность редактирования данной ячейки, так же можно выделить строку с данными и удалить ее, при нажатии на кнопку добавить, появляется дополнительная строка куда нужно вписать данные (спектакль, дата премьеры, время премьеры, билеты).

Название прецедента: **подтверждение покупки**

Действующие лица: администратор, кассир

Предусловия: в главном окне выбрать раздел “Заявки на покупку”

Цель: подтвердить или опровергнуть покупку

Действия: выбрать заявку, затем через интерфейс одобрить или отклонить

Название прецедента: **добавить кассира**

Действующие лица: администратор

Предусловия: в главном окне выбрать раздел “Добавить кассира”

Цель: добавить или удалить кассира

Действия: при нажатии на кнопку добавить кассира появится дополнительная строка в таблице, где можно будет задать данные логина и пароля, так же можно выделить определенную строку данных и удалить ее.

# 1.4 Входные и выходные данные

Все входные данные будут вводиться или устанавливаться пользователем через пользовательский интерфейс программы и сохранятся в текстовые файлы.

Данные текстовых файлов также являются входными и выходными данными.

Таблица 1 – афиша

|  |  |  |
| --- | --- | --- |
| Спектакли | Дата премьеры | Время премьеры |
| ...... | ...... | ...... |
| ...... | ...... | ...... |

Пример хранения в файле \“В тени виноградника”\23.10.2020\14:20

Таблица 2 - спектакль

|  |  |  |
| --- | --- | --- |
| Автор | Постановщик | Действующие лица |
| ..... | ..... | ..... |
| ..... | ..... | ..... |

Пример хранения в файле \1\Любовский.А.Т\2\Андросов.В.Ю\3\Егорова.А.Г\3\Чурим.Н.А\3\Суриков.П.Д\3\Калуга.Е.А.

Таблица 3 - билеты

|  |  |  |
| --- | --- | --- |
| Балкон | Ложи | Партер |
| цена | цена | цена |
| количество | количество | количество |

Пример хранения в файле \1\1000\2\20\1\500\2\300\1\400\2\300\

Таблица 4 - кассиры

|  |  |
| --- | --- |
| Login | Password |
| ….. | ….. |
| ….. | ….. |

Пример хранения в файле \Login\12345\

Таблица 5 - заявки

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Фамилия | Имя | Отчество | Номер крт. | Кол-во б. | Тип билета |
| .... | .... | .... | ..... | .... | ..... |
| .... | .... | .... | ..... | .... | ..... |

Пример хранения в файле \1\Минин\2\Иван\3\Владимирович\4\123\5\2\6\ложи\

При запуске программы, все шесть файлов загружаются в оперативную память. При добавлении/изменении данных в базе, они перезаписываются в новый файл этой БД.

# 1.5 Стадии и этапы разработки

Разработка должна быть проведена в три стадии:

* 1. техническое задание;
  2. технический (и рабочий) проекты;
  3. внедрение.

На стадии «Техническое задание» должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии «Технический (и рабочий) проект» должны быть выполнены перечисленные ниже этапы работ:

- разработка программы;

- разработка программной документации;

- испытания программы.

На стадии «Внедрение» должен быть выполнен этап разработки «Подготовка и передача программы».

Создание работ по этапам:  
На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- постановка задачи;

- определение и уточнение требований к техническим средствам;

- определение требований к программе;

- определение стадий, этапов и сроков разработки программы и документации на нее;

- согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

# 2 Проектирование

# 2.1 Разработка диаграмм пригодности

На рисунках 9–14 приведены диаграммы пригодности для описанных ниже прецедентов.

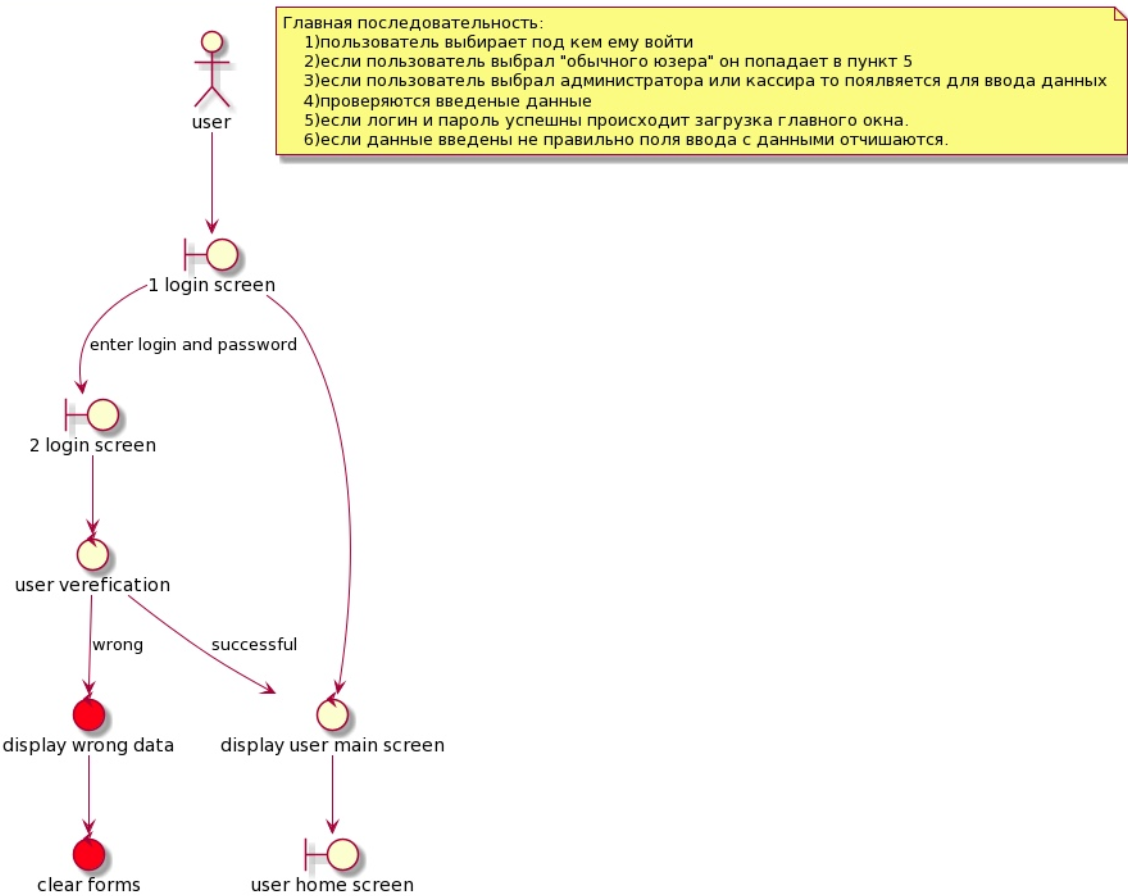


Рисунок 8 – Диаграмма пригодности прецедента входа в систему

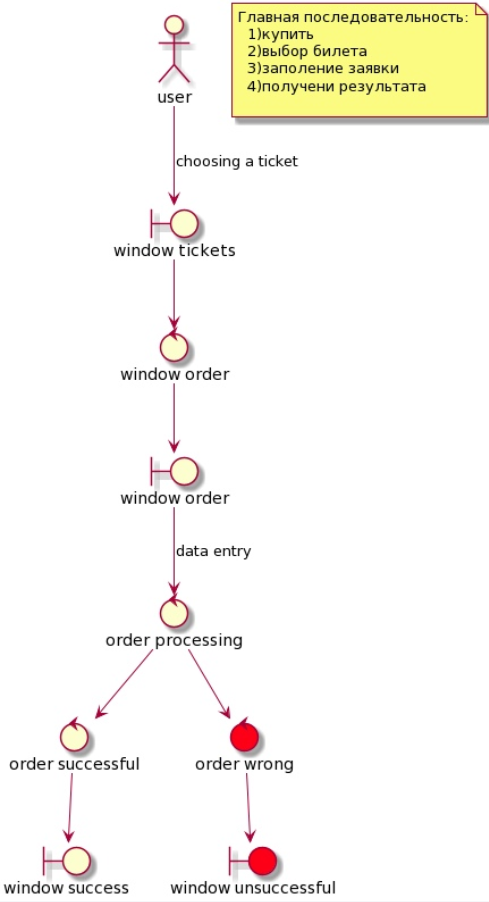


Рисунок 9 – Диаграмма пригодности прецедента приобретения билета

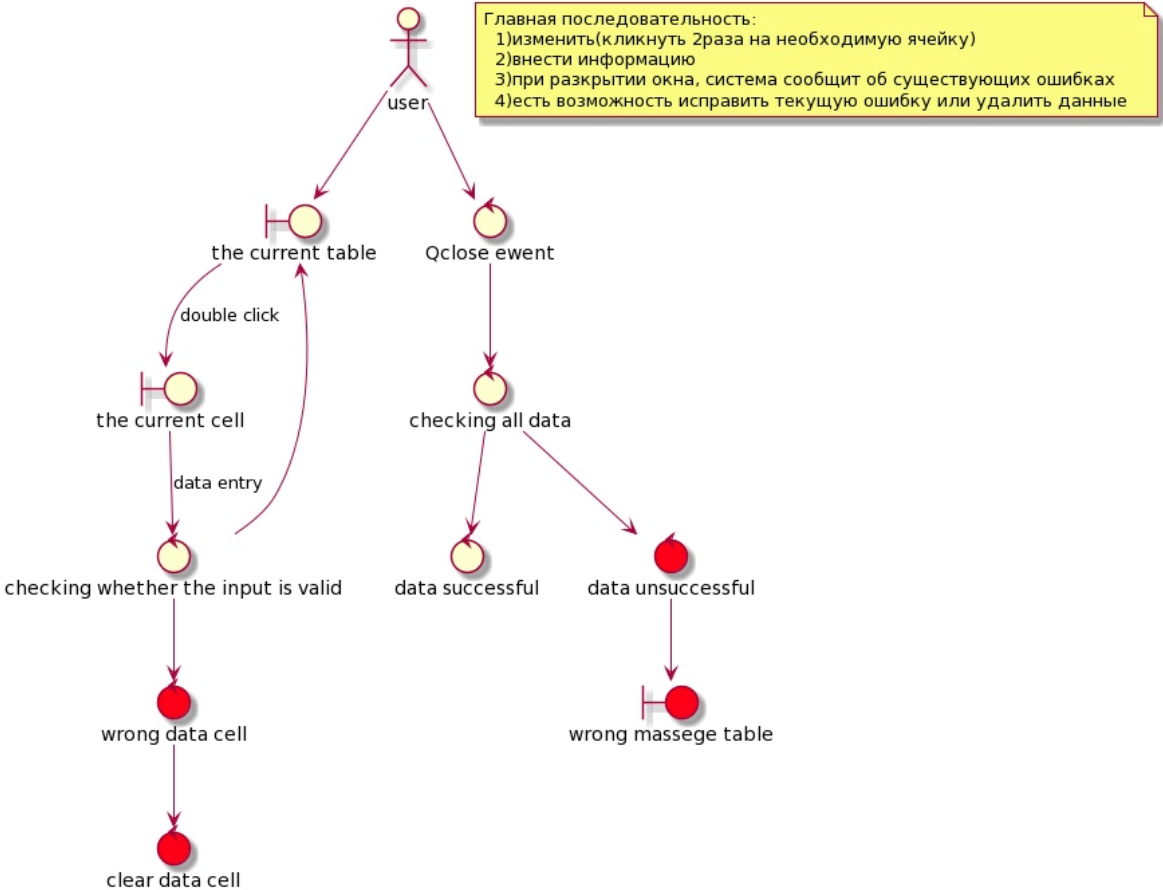


Рисунок 10 – Диаграмма пригодности прецедента редактирование таблицы

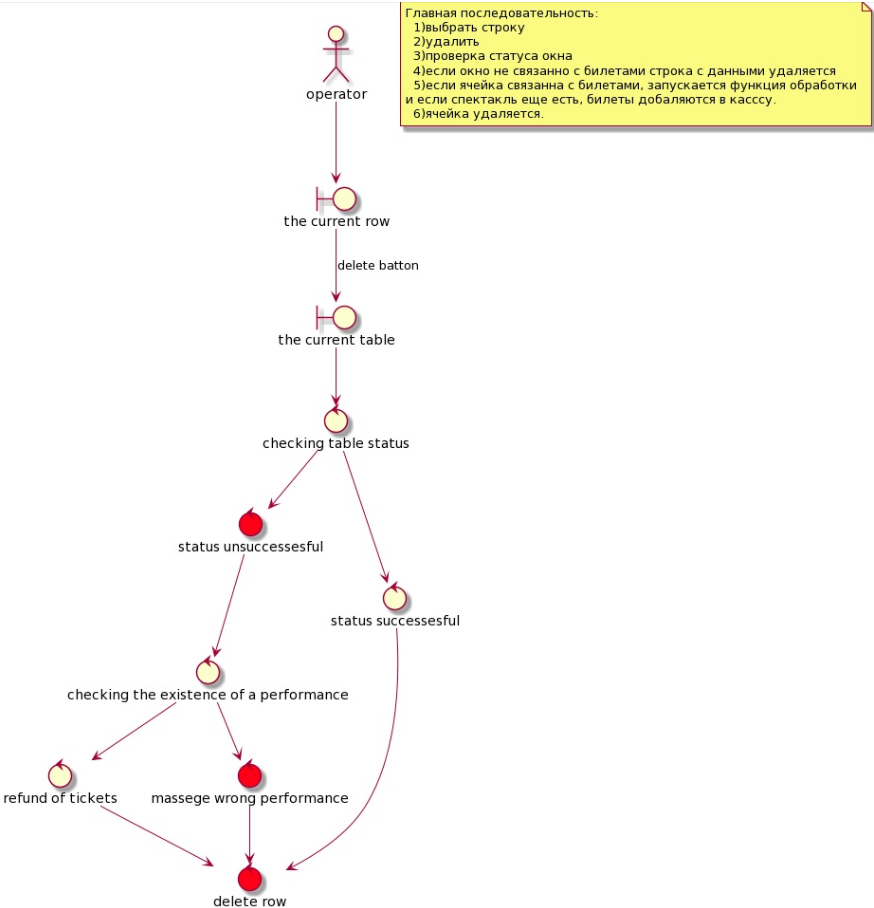


Рисунок 11 – Диаграмма пригодности прецедента удаление строки

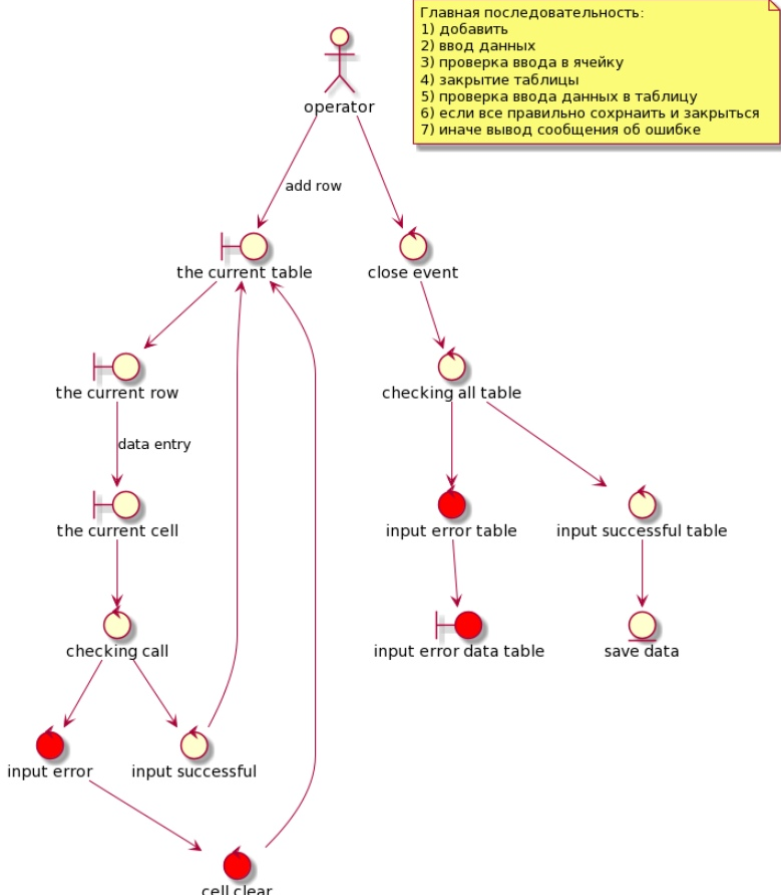


Рисунок 12 – Диаграмма пригодности прецедента добавление строки

При разработке этих диаграмм выявленные новые сущности, диаграммы последовательности на рисунках 15 – 21:

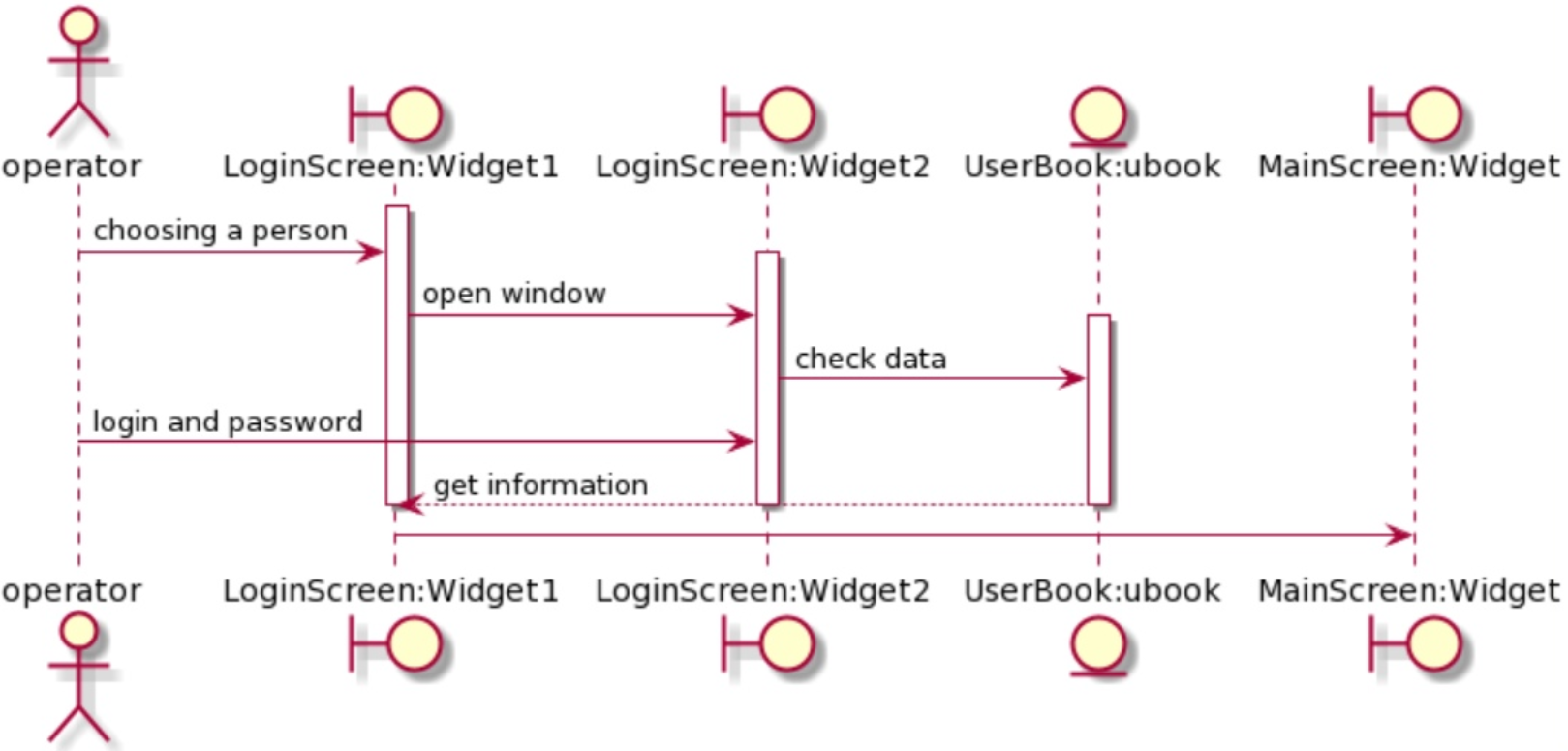


Рисунок 13 – Вход в систему

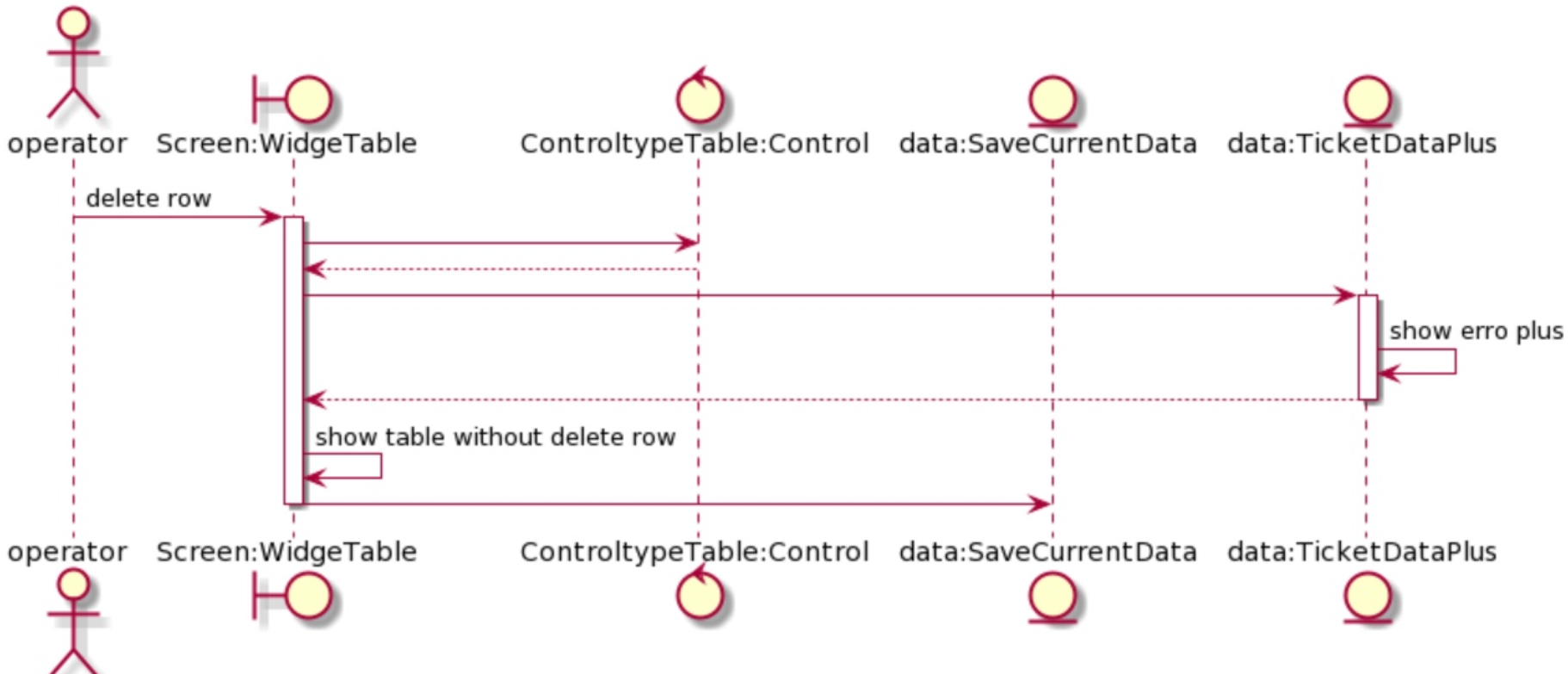


Рисунок 14 – Удалить строку

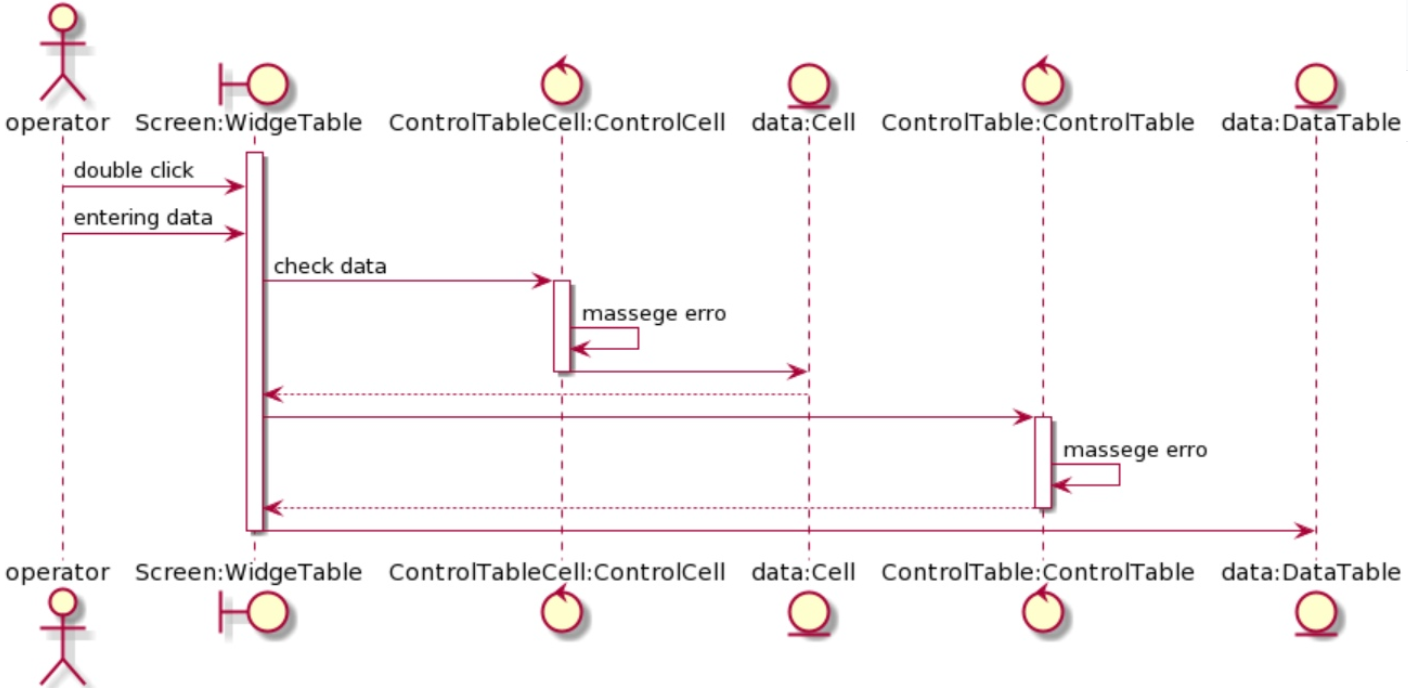


Рисунок 15 – Редактирование информации

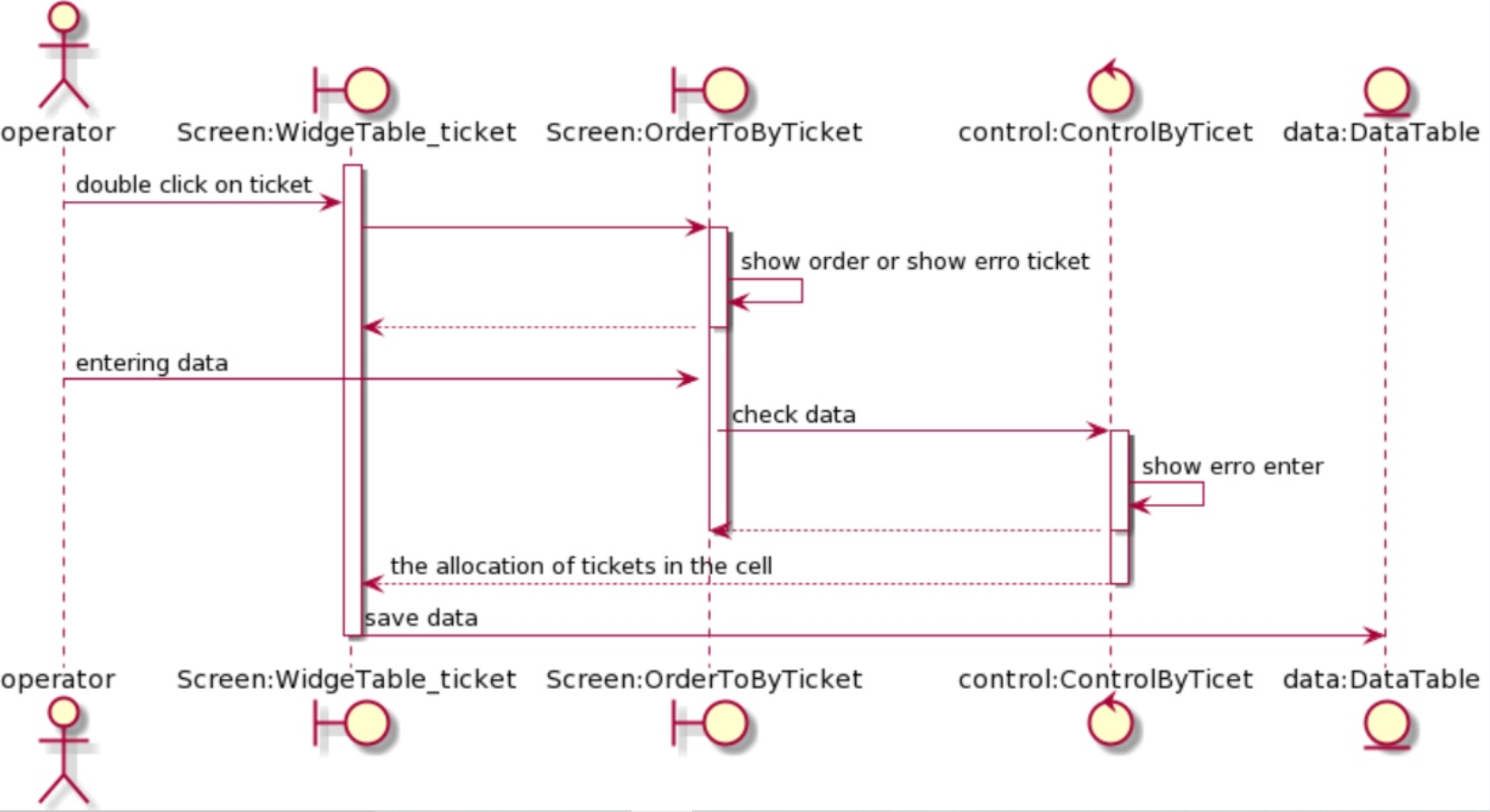


Рисунок 16 – Заказ билета

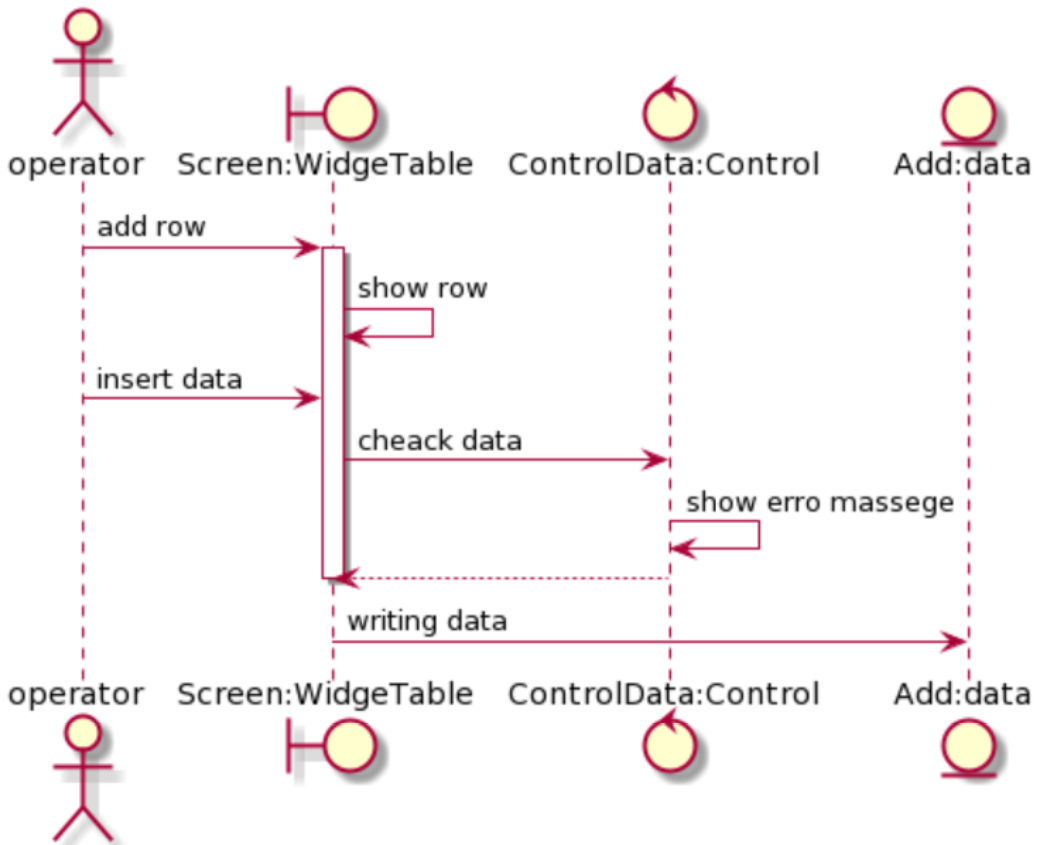


Рисунок 17 – Добавить ячейку

# 3 Подробное описание алгоритма

Описание алгоритма можно продемонстрировать, расписав диаграммы пригодности для прецедента.

1) Название прецедента: добавление новой строки.

Действующее лицо: администратор.

Цель: добавление новой строки.

Последовательность:

1. Администратор нажимает в текущей таблице на кнопу “добавить”.

Заполняет текущую ячейку информацией.

2. Система проверяет корректность ввода данных.

3.Если данные соответствуют стандарту, то таблица сохраняет введенные данные, иначе веденные данные в ячейки затираются.

4. При закрытии текущей таблицы данные всей таблицу перепроверяются

5. Если ошибок не обнаружено данные перезаписываются из таблицы в файл.

2) Название прецедента: удалить строку.

Действующее лицо: администратор.

Цель: удалить рейс.

Последовательность:

1. Администратор нажимает кнопку «удалить».

2. Система проверяет тип таблицы.

3. Если текущая таблица “таблица ордеров”, происходит проверка информации спектакля по билету, если спектакль действующий, происходит добавление билетов в кассу, иначе всплывает окно предупреждения, о том, что спектакль не обнаружен.

4. Происходит уничтожение строки

5. При выходе из программы данные перезаписываются и текстовый файл.

3) Название прецедента: заказ билета.

Действующее лицо: пользователь.

Цель: заказ билета.

Последовательность:

1. Пользователь находится в таблице билетов по выбранному спектаклю.

2. Пользователь нажимает на выбранный тип билета, затем заполняет всплывшее регистрационное окно.

3. Пользователь нажимает «купить».

4. Если данные были введены верно, то пользователь увидит сообщение о том что заявка была принята, и ему нужно ждать звонка о подтверждении покупки от кассира.

5. После закрытия окна система переносит покупателя обратно в таблицу билетов.

6. Данные в системе перезаписываются в текстовый файл.

4) Название прецедента: редактирование информации.

Действующее лицо: администратор.

Цель: отредактировать информацию

Главная последовательность:

1. Администратор в текущей таблице выбирает ячейку, двойным кликом активирует редактирование информации.

2. Вводит новую информацию.

3. Система проверяет редактируемую ячейку.

4. Если информации введена верно ячейка сохраняет введенную информацию, иначе происходит восстановление прежних данных.

5. При закрытии таблицы информация перезаписывается в текстовый файл.

5) Название прецедента: вход в систему.

Действующее лицо: любой.

Цель: вход в систему

Последовательность:

1. Пользователь запускает приложение и выбирает в открывшемся окне способ входа в систему.

2. Если пользователь выбрал “Обычный пользователь”, то сразу же попадает на главное окно “Афиша”.

3. Если пользователь выбрал Администратора или Кассира, то всплывает окно ввода данных, которое нужно заполнить, далее после заполнения, данные проверяются и пользователь попадает на главное окно или всплывает окно ошибки входа в систему.

# 5 Тестирование программы

Тест «Авторизация»:

1. Ввод неверных данных - ошибка о неправильно введенных данных.
2. Незаполненные поля - ошибка о существовании незаполненных полей.
3. Пользователь авторизуется под правами кассира/администратора - Подтверждение авторизации от пользователя со статусом «Администратор».
4. Если пользователь верно ввел данные, то пользователь будет авторизован и перенаправлен в главное окно.

Тест «Редактирование таблицы»:

1. Существуют не заполненные поля - ошибка о существовании незаполненных полей.
2. Несоответствие правилам ввода – ошибка о существовании не корректности ввода.
3. Если пользователь, верно ввел данные, то данные в таблице обновятся.

Тест «Добавление строки»:

1. Незаполненные поля - ошибка о существовании незаполненных полей.
2. Ввод данных - ошибка ввода данных.
3. Если администратор, верно ввел данные, то данные в таблице будут обновлены.

# 6 Инструкция к программе

При запуске программы пользователя встречает главное окно, где можно выбрать необходимый спектакль и получить необходимую информацию о нем.

После входа в систему функционал программы разделяется в зависимости от статуса пользователя.

Обычный пользователь может просматривать спектакли и любую информацию связанную с ними, может заказать билет.

Пользователь со статусом кассир может выполнять следующие действия:

- просматривать список заказов, удалять или добавлять их.

Пользователь со статусом администратор может выполнять следующие действия:

- изменять, добавлять или удалять любую информацию, связанную со спектаклями, билетами, кассирами.

# 7 Заключение

В ходе выполнения проекта были решены следующие поставленные задачи:

- Подготовить описание предметной области.

- Разработать техническое задание.

- Написать проект на тему информационная система «Театр».

- Оформить пояснительную записку.

В процессе работы над проектом, я улучшил знания в познании Qt Creator. Функции программы полностью соответствуют заданию.

Список использованных источников

1. Система электронного обучения СФУ: Шаблон пояснительной записки для LibreOffice [Электронный ресурс]. – Режим доступа: https://e.sfu-kras.ru/mod/folder/view.php?id=203018;

2. Открытый кроссплатформенный инструмент для построения UML-диаграмм из текстового описания: Диаграмма классов [Электронный ресурс]. – Режим доступа:www.plantuml.com/plantuml/uml/SoWkIImgAStDuN98pKi1qW00;

3. Qt Coding Style/ru: правила кодирования с использованием Qt [Электронный ресурс]. – Режим доступа: http:// wiki.qt.io/Qt\_Coding\_Style/ru;