Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчёт**

по предмету «Проектирование программного обеспечения»

Лабораторная работа №7

«Разработка диаграммы поведения»

Студент: Стрелковская В. А.

ФИТ 3 курс 2 группа

Преподаватель: Якубенко К. Д.

# Описание функциональных требований

Функциональны требования к системе можно разделить на требования к функционалу для различных ролей приложения – пользователя, гостя, администратора.

1. Функционал для пользователя:

Для зарегистрированных пользователей приложение предлагает широкий спектр возможностей, направленных на удобство и улучшение пользовательского опыта:

* просмотр расписания сеансов и фильмов;
* просмотр информации о фильмах;
* онлайн покупка билетов на сеансы;
* выбор мест в зале при покупке билетов;
* просмотр истории бронирований и приобретенных билетов;
* авторизация пользователя.

1. Функционал для администратора:

Администраторы играют ключевую роль в управлении и поддержании работы приложения. Их функционал включает:

* управление расписанием сеансов и фильмами;
* добавление, редактирование и удаление информации о фильмах (включая название, описание, жанр и актеров);
* управление ценами на билеты.

1. Функционал для гостя:

Гостям предоставляется доступ к ограниченному набору функций, что позволяет им ознакомиться с приложением и принять решение о регистрации:

* регистрация;
* просмотр расписания сеансов и фильмов;
* просмотр информации о фильмах;
* поиск фильмов и сеансов по критериям.

# Описание программных средств

Для построения диаграмм IDEF0 использовался веб-ресурс Draw.io, разрабатываемый компанией JGraph Ltd. и направленный на построение диаграмм. Адрес веб-ресурса – https://www.drawio.com. Данный ресурс доступен на всех платформах, имеющих веб-браузер и доступ в Интернет.

Draw.io предлагает интуитивно понятный интерфейс, который позволяет легко создавать и редактировать диаграммы. В функционал данного ресурса входит широкий спектр возможностей, включая:

* построение графиков и смысловых карт: пользователи могут визуализировать свои идеи и концепции, создавая понятные и наглядные схемы;
* UML-диаграммы: этот инструмент поддерживает создание различных типов UML-диаграмм, что делает его полезным для разработчиков программного обеспечения и системных аналитиков;
* диаграммы Венна: draw.io позволяет создавать диаграммы Венна для визуального представления пересечений и различий между наборами данных;
* Agile и Kanban доски: инструмент поддерживает методологии Agile, что позволяет командам эффективно управлять проектами и отслеживать прогресс;
* графики мозговых штурмов: пользователи могут организовывать свои идеи и генерировать новые концепции в удобной и визуально привлекательной форме;
* диаграммы архитектур технических систем: draw.io предоставляет возможности для создания сложных архитектурных схем, что полезно для инженеров и проектировщиков.

Одним из ключевых преимуществ draw.io является возможность совместной работы в реальном времени, что позволяет командам эффективно сотрудничать и вносить изменения одновременно. Пользователи могут делиться своими диаграммами с коллегами и работать над проектами совместно, не беспокоясь о лицензиях или ограничениях платформ.

Кроме того, draw.io обеспечивает высокий уровень конфиденциальности и безопасности данных, позволяя пользователям хранить свои диаграммы локально или в облачных хранилищах по своему выбору. Это делает инструмент идеальным для команд, стремящихся к безопасному управлению своей информацией.

Таким образом, использование draw.io для построения диаграмм IDEF0 не только облегчает процесс визуализации, но и предоставляет мощные инструменты для поддержки различных методологий и повышения эффективности работы команды.

# Описание практического задания

* 1. **Постановка задачи**

Изучение методологии объектно-ориентированного моделирования средствами UML. Ознакомление с основными принципами объектно-ориентированного проектирования программного обеспечения, получение навыков проектирования структуры информационной системы с применением UML.

* 1. **Описание программных средств**

Для построения моделей использовалось программное средство Draw.io.

Draw.io – это инструмент для создания диаграмм, блок-схем, интеллект-карт, бизнес-макетов, диаграмм связей сущностей (ERD), программных блоков, схем сетевых архитектур и других визуальных представлений данных. Сервис распространяется бесплатно, с возможностью использования в онлайн-режиме, а также предлагает десктопные приложения для различных операционных систем. Draw.io поддерживает интеграцию с популярными облачными хранилищами, такими как Google Drive, Dropbox, OneDrive, и GitHub, что делает его удобным для совместной работы и версионирования диаграмм. Пользователи могут экспортировать диаграммы в различные форматы, такие как PNG, PDF, SVG, и другие.

Особенности Draw.io:

* более 500 шаблонов элементов и фигур;
* облегчённый интерфейс, в котором за короткий промежуток времени можно создать готовый проект;
* поддержка горячих клавиш, задействованных в большинстве графических редакторов;
* экспорт в форматы: JPG, PNG, SVG, VDSX;
* возможность совместной работы;
* наличие различных фоновых тем;
* мультиязычный интерфейс.

Программное средство draw.io также поддерживает импорт файлов в различных форматах, включая .vsdx (формат Microsoft Visio), Gliffy™ и Lucidchart™. Это позволяет вам работать с существующими диаграммами, созданными в других инструментах, и продолжать их редактирование и доработку в draw.io.

1. **Практическое задание**

Первой диаграммой была выбрана диаграмма деятельности – рисунок 4.1. Она посвящена процессу бронирования места.

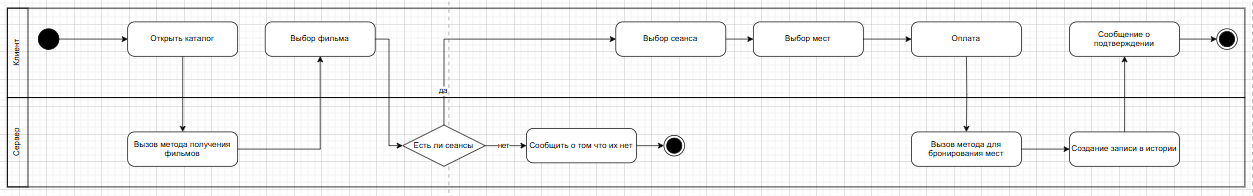


Рисунок 4.1 – Диаграмма деятельности

Вторая диаграмма – диаграмма состояния. Отобразим на ней процесс оплаты брони. Представлено на рисунке 4.2.

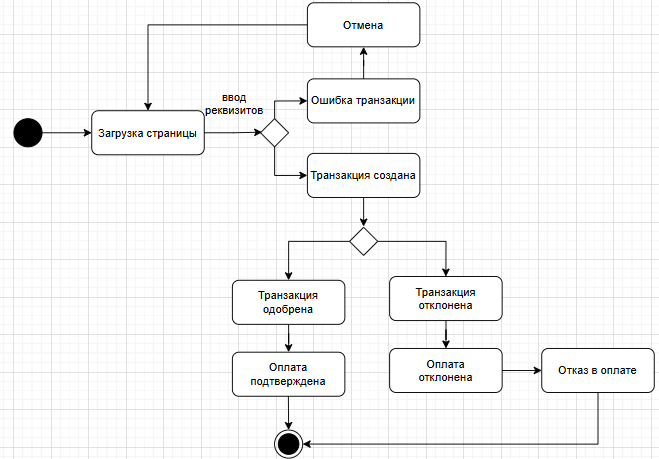


Рисунок 4.2 – Диаграмма состояния

Основными элементами диаграммы выступают следующие объекты: начальное псевдосостояние, состояния «Переход на страницу оплаты», «Ввод реквизитов», «Подтверждение», «Отмена», а также конечное состояние.

Переходы определяют статус транзакции оплаты, а также статус взаимодействия пользователя с системой.

Вывод: изучена методология объектно-ориентированного моделирования средствами UML. Получил дополнительные навыки проектирования моделей информационной системы с применением возможностей UML диаграмм поведения в контексте программного средства «REALFILM».

1. **Ответы на вопросы**
   1. Укажите виды диаграмм поведения.

Всего существует 3 вида диаграмм поведения:

* диаграмма деятельности;
* диаграмма состояний;
* диаграмма вариантов использования.
  1. Опишите назначение диаграммы деятельности.

Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

* 1. Опишите основные нотации, которые используются на диаграмме состояний.

| **Элемент/Нотация** | **Предназначение** |
| --- | --- |
| Пример | Класс (Class) |
| Пример | Состояние (State) |
| Пример | Состояние (StateEx) |
| Пример | Составное состояние (Composite state) |
| Пример | Разделитель (Concurrent state) |
| Пример | История (History) |
| Пример | Глубокая история (Deep history) |
| Пример | Начальное состояние (Start state) |
| Пример | Конечное состояние (Final state) |
| ПримерПример | Синхронизатор/разветвитель (Complex transition) |
| Пример | Переход (Transition) |
| Пример | Сообщение (Event message) |
| Пример | Точка изгиба связей (Point) |
| Пример | Комментарий (Note) |
| Пример | Коннектор комментария (Note connector) |

* 1. Укажите виды связей между объектами на диаграмме последовательностей.

Синхронное сообщение — отправитель передаёт ход управления актёру-получателю, которому необходимо провести в прецеденте некоторое действие. Пока проводимое получателем действие не будет завершено (не будет получено ответное сообщение), отправитель теряет возможность производить какие-либо действия. Графически изображается как сплошная линия со стрелкой в виде закрашенного треугольника, после которой идёт прямоугольник, отражающий деятельность объекта, в конце которого находится ответное сообщение.

Ответное сообщение — данное сообщение является ответом на синхронное сообщение. Обычно, содержит какое-либо возвращаемое изначальному отправителю значение, также возвращающее ему управление (возможность действовать). Графически изображается пунктирной линией с открытой стрелкой.

Асинхронное сообщение — отправитель передаёт ход управления получателю, которому необходимо провести в прецеденте некоторое действие. Основное отличие от синхронного сообщения состоит в том, что отправитель не теряет возможности совершать другие действия. Графически изображается сплошной линией с открытой стрелкой. Потерянное сообщение — сообщение без адресата. Найденное сообщение — сообщение без отправителя.

Последние два вида стрелок (взаимодействий) используются крайне редко. В основном они используются для демонстрации взаимодействия имеющихся объектов в данном прецеденте с внешними системами.

* 1. Какая диаграмма позволяет моделировать параллельные вычисления?

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.