**Проект ООТРПО**

**Дайджест по второму этапу**

**Состав команды:**

* Шахов Саша, 5304;
* Третьяков Саша, 5304;
* Сальников Виталий, 5304;
* Закоморный Андрей, 5303;
* Владимирова Оля, 5381.

**Выбранный контейнер:** двудольный граф.

**В дайджесте представлены:**

* План итерации.
* Запросы заинтересованных лиц.
* Схема навигации.
* Прототип интерфейса.
* Архитектурный план:
  + диаграмма классов;
  + диаграмма деятельности;
  + диаграмма коммуникации;
  + диаграмма компонентов;
  + диаграмма развертывания;
  + диаграмма последовательностей;
  + диаграмма состояний;

## План итераций

**Введение**

**Цель**

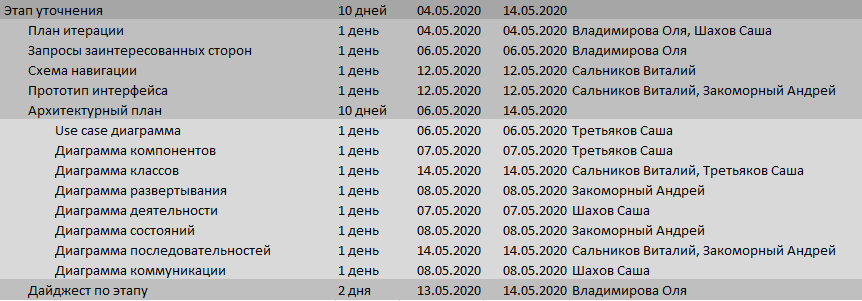
Цель документа – описание плана итерации проекта. В данной итерации проводится уточнение требований к ПО.

**Определения и сокращения**

Представлены в артефакте Глоссарий.

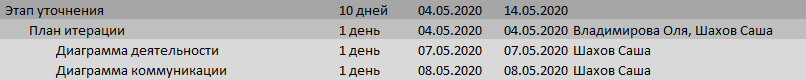
**План**

**Задачи**

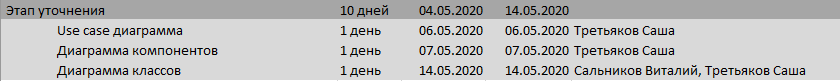


**Задачи каждого участника:**

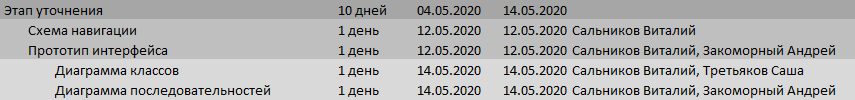
* Шахов Саша, 5304:



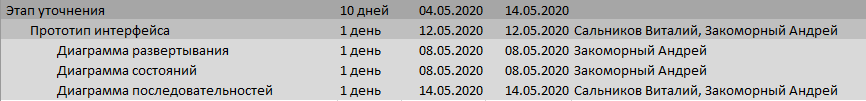
* Третьяков Саша, 5304:



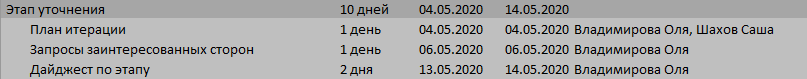
* Сальников Виталий, 5304:



* Закоморный Андрей, 5303:



* Владимирова Оля, 5381:

****

**Запросы заинтересованных лиц**

# 1. Введение

## Цель

Целью документа является определение, фиксация и анализ запросов к разрабатываемому проекту от заинтересованных лиц.

## Контекст

Документ содержит описание запросов заинтересованных лиц к различным аспектам разрабатываемого программного продукта, таких как описание проблем, потенциальных пользователей, возможностей, надежности, производительности и поддержки.

## Определения, акронимы и сокращения

См. Глоссарий.

# Определение заинтересованных лиц

При определении заинтересованных лиц проекта были выявлены следующие лица:

* Заказчик.
* Пользователь.
* Администратор.

# Оценка проблемы

Проблемы, связанные с разрабатываемым продуктом, отсутствуют. Возможные риски при разработке продукта и стратегии их устранения указаны в документе «Риски».

# Определение пользовательской среды

Приложение предполагает двух основных пользователей: пользователь и администратор. Пользователь – сотрудник кинотеатра, обязанностью которого является составление расписания. Система должна упростить для пользователя планирование работы залов кинотеатра. Администратором системы должен быть сотрудник отдела технической поддержки. Администратор отвечает за наполнение системы, корректность работы и, при возникновении, устранение неполадок. Пользователи могут получать доступ к Cinema3000 локально. Для доступа к приложению необходим компьютер с ОС Windows.

# Оценка возможностей

При определении пользовательской среды было установлено, что пользователями конечного продукта будут являться администраторы кинотеатров. Основными пользователями данного приложения считаются люди, участвующие в организации работы кинотеатров, а именно – имеющие доступ к созданию расписания.

# Оценка надежности, производительности и поддержки

При оценке надежности следует учесть, что разрабатываемом продуктом будут пользоваться люди, которые регулярно занимаются организацией работы кинотеатров. Приложение работает с большим количеством данных, поэтому следует добиться высокой надежности приложения. Данная задача будет достигаться путем использования оптимальных инструментов и алгоритмов при разработке, а также на этапе тестирования продукта. К тому же, важна производительность используемого приложения. Поэтому следует добиться оптимальной производительности для первой версии конечного продукта и в дальнейшем повышать производительность приложения, путём оптимизации работы приложения. Поддержка программного продукта планируется путем обновлений и улучшений предыдущих версий приложения. В приложении не планируется вводить использование денежных средств и личных данных, поэтому требования к безопасности минимальны. Системные требования для корректной установки и работы приложения будут ориентированы на компьютеры различной конфигурации. Уточненные системные требования появятся на этапе тестирования программного продукта.

## Схема навигации

**Цель**

Создать схему навигации согласно возможным сценариям поведения пользователя для приложения Cinema3000.

**Определения и сокращения**

См. Глоссарий.

**Схема навигации**

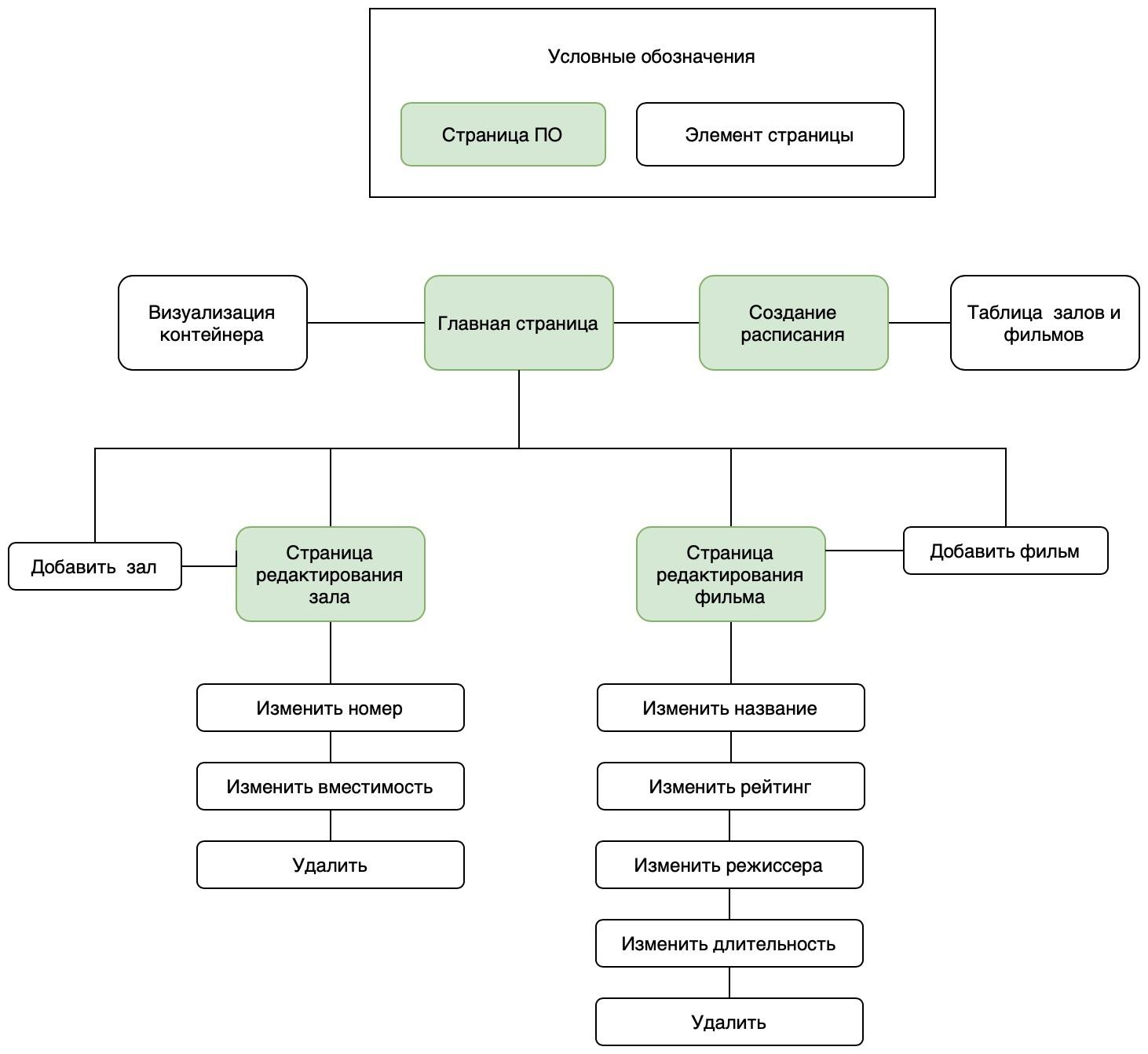


Рисунок 1 – Схема навигации

## Прототип интерфейса

1. **Введение**

Целью этого документа является описание и визуальное представление пользовательского интерфейса системы Cinema3000.

1. **Описание**

Пользовательский интерфейс — совокупность средств и методов, при помощи которых пользователь взаимодействует с различными устройствами и аппаратурой.Интерфейс воспринимают как то, что помогает пользователюполучить доступ к функционалу устройства.

Традиционно, интерфейс воспринимают как посредника между человеком и возможностями продукта. Специалисты (UI-дизайнеры, UX-проектировщика) ставят своей целью создать интерфейс, который бы позволял потребителю как можно проще, удобнее, быстрее использовать возможности продукта. Но такой подход представляется в корне неверным, главным образом потому, что он не учитывает основные цели и истинные желания пользователя.Пользователь требуется лишь одного – решить свою задачу. Именно это истинное желания потребителей, которое и должно служить главным ориентиром при разработке продуктов.

При разработке, интерфейс должен восприниматься не как мост между пользователем и функционалом устройства, он должен быть, в первую очередь, мостом между пользователем и решением его реальных задач. Если продукт вместо того, чтобы решать задачи пользователя, сам ставит пользователям свои задачи, то этот продукт скорее усложняют жизнь, чем упрощают ее. Воспринимая разработку продукта не как производство конкретного приложения, а как некий инструмент, решающий конкретные задачи потребителя, можно прийти к новым решениям, улучшающим жизнь.

1. **Внешний вид и поведение программной среды**
   1. Профили пользователей
      1. Абстрактный администратор кинотеатра
         1. Контекст и точка входа

Администратору нужно постоянно поддерживать работу кинотеатра. Причем расписание должно быть составлено заранее и уже доступно клиентам за, допустим, месяц до сегодняшнего дня. Администратор заранее знает, какие фильмы будут транслироваться в то время, поэтому накануне даты выкладывания расписания администратор использует приложение Cinema3000.

* + - 1. Страхи и опасения

Основным опасением у администратора является то, что расписание может быть составлено неверно, что влечет за собой либо излишний простой кинозалов, либо – наложение сеансов. В результате – материальные потери в обоих случаях.

* + - 1. Приоритеты и ограничения

Приоритетом для администратора будет возможность быстрого формирования расписания без какой-либо аналитической работы. Также важно исключение ошибки, связанной с человеческим фактором.

* 1. Use case

Пользователь (администратор кинотеатра) может редактировать список фильмов, менять свойства фильма, редактировать список залов, менять свойства залов, связывать фильмы и залы, составлять расписание, просматривать расписание и экспортировать расписание.

На рис. 2 представлена Use-case диаграмма.

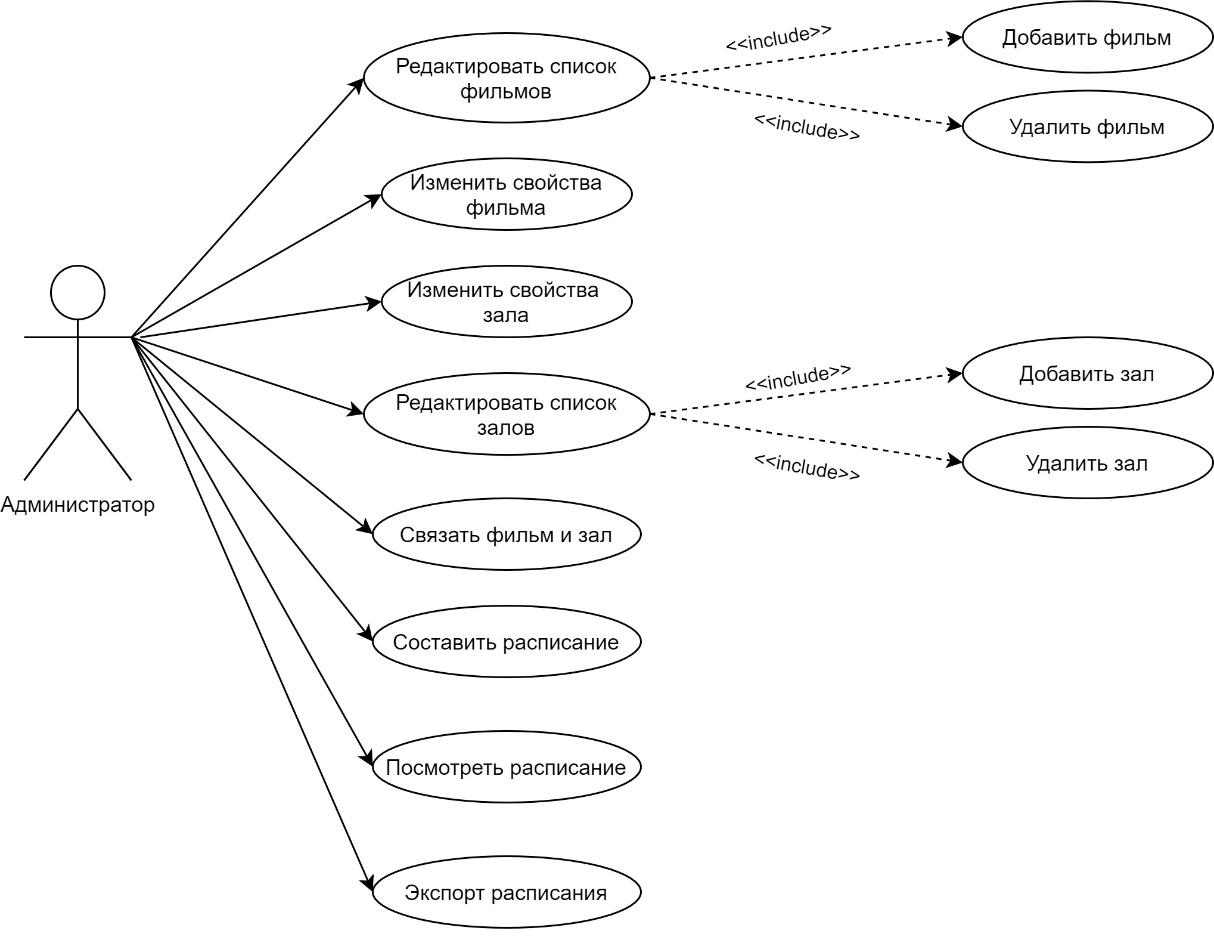


Рисунок 2 – Use case диаграмма

Таблица 1. Сценарий использования “Редактировать список фильмов” с добавлением фильма

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Изменение списка фильмов |
| Успешный сценарий:   * Пользователь выбирает «Редактировать список фильмов». * Выводится поле, в котором можно выбрать действие: удалить или добавить фильм. * Пользователь выбирает «Добавить». * Открывается поле для добавления фильма. * Пользователь вводит данные, сохраняет информацию. * Фильм добавляется в список фильмов.   Альтернативный сценарий (Наступает при уже созданном фильме):   * Пользователь выбирает «Редактировать список фильмов». * Выводится поле, в котором можно выбрать действие: удалить или добавить фильм. * Пользователь выбирает «Добавить». * Открывается поле для добавления фильма. * Пользователь вводит данные, сохраняет информацию. * Приложение сообщает, что такой фильм уже существует. | |
| Успешный результат | Фильм добавлен. |
| Отрицательный результат | Вывод сообщения о невозможности добавления фильма. |

Таблица 2. Сценарий использования “Редактировать список фильмов” с удалением фильма

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Изменение списка фильмов |
| Успешный сценарий:   * Пользователь выбирает «Редактировать список фильмов». * Выводится поле, в котором можно выбрать действие: удалить или добавить фильм. * Пользователь выбирает «Удалить». * Открывается список фильмов. * Пользователь выбирает фильм. * Подтверждает желание удалить фильм. * Фильм удаляется из списка фильмов.   Альтернативный сценарий (Наступает при отсутствии подтверждения):   * Пользователь выбирает «Редактировать список фильмов». * Выводится поле, в котором можно выбрать действие: удалить или добавить фильм. * Пользователь выбирает «Удалить». * Открывается список фильмов. * Пользователь выбирает фильм. * Не подтверждает желание удалить фильм. * Список фильмов остается без изменений. | |
| Успешный результат | Фильм удален. |
| Отрицательный результат | Список фильмов без изменений. |

Таблица 3. Сценарий использования “Изменить свойства фильма”

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Изменение свойства фильма |
| Успешный сценарий:   * Пользователь выбирает «Изменить свойства фильма». * Выводится поле, в котором необходимо выбрать фильм. * Пользователь фильм. * Открывается поле для редактирования свойства фильмов. * Пользователь вводит данные. * Сохраняет информацию. * Свойства фильма изменяются.   Альтернативный сценарий (Наступает при вводе некорректных данных):   * Пользователь выбирает «Изменить свойства фильма». * Выводится поле, в котором необходимо выбрать фильм. * Пользователь выбирает фильм. * Открывается поле для редактирования свойства фильмов. * Пользователь вводит некорректные данные. * Сохраняет информацию. * Выводится сообщение об ошибке. | |
| Успешный результат | Изменены свойства фильма. |
| Отрицательный результат | Выводится сообщение об ошибке. |

Таблица 4. Сценарий использования “Редактировать список залов” с добавлением зала

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Изменение списка залов |
| Успешный сценарий:   * Пользователь выбирает «Редактировать список залов». * Выводится поле, в котором можно выбрать действие: удалить или добавить зал. * Пользователь выбирает «Добавить». * Открывается поле для добавления зала. * Пользователь вводит данные, сохраняет информацию. * Зал добавляется в список залов.   Альтернативный сценарий (Наступает при уже созданном зале):   * Пользователь выбирает «Редактировать список залов». * Выводится поле, в котором можно выбрать действие: удалить или добавить зал. * Пользователь выбирает «Добавить». * Открывается поле для добавления зала. * Пользователь вводит данные, сохраняет информацию. * Приложение сообщает, что такой зал уже существует. | |
| Успешный результат | Зал добавлен. |
| Отрицательный результат | Вывод сообщения о невозможности добавления зала. |

Таблица 5. Сценарий использования “Редактировать список фильмов” с удалением зала

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Изменение списка залов |
| Успешный сценарий:   * Пользователь выбирает «Редактировать список залов». * Выводится поле, в котором можно выбрать действие: удалить или добавить зал. * Пользователь выбирает «Удалить». * Открывается список залов. * Пользователь выбирает зал. * Подтверждает желание удалить зал. * Зал удаляется из списка залов.   Альтернативный сценарий (Наступает при отсутствии подтверждения):   * Пользователь выбирает «Редактировать список залов». * Выводится поле, в котором можно выбрать действие: удалить или добавить зал. * Пользователь выбирает «Удалить». * Открывается список залов. * Пользователь выбирает зал. * Не подтверждает желание удалить зал. * Список залов остается без изменений. | |
| Успешный результат | Зал удален. |
| Отрицательный результат | Список залов без изменений. |

Таблица 6. Сценарий использования “Изменить свойства зала”

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Изменение свойства зала |
| Успешный сценарий:   * Пользователь выбирает «Изменить свойства зала». * Выводится поле, в котором необходимо выбрать зал. * Пользователь выбирает зал. * Открывается поле для редактирования свойства залов. * Пользователь вводит данные. * Сохраняет информацию. * Свойства зала изменяются.   Альтернативный сценарий (Наступает при вводе некорректных данных):   * Пользователь выбирает «Изменить свойства зала». * Выводится поле, в котором необходимо выбрать зал. * Пользователь выбирает зал. * Открывается поле для редактирования свойства залов. * Пользователь вводит некорректные данные. * Сохраняет информацию. * Выводится сообщение об ошибке. | |
| Успешный результат | Изменены свойства зала. |
| Отрицательный результат | Выводится сообщение об ошибке. |

Таблица 7. Сценарий использования “Связать фильм и зал”

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Добавить связь «фильм-зал» |
| Успешный сценарий:   * Пользователь выбирает «Связать фильм и зал». * Пользователь выбирает фильм и зал. * Приложение отображает связь между фильмом и залом.   Альтернативный сценарий (Наступает при уже существующей связи):   * Пользователь выбирает «Связать фильм и зал». * Пользователь выбирает фильм и зал. * Приложение выводит сообщение, что фильм и зал уже связаны. | |
| Успешный результат | Добавлена новая связь. |
| Отрицательный результат | Связи не изменились. |

Таблица 8. Сценарий использования “Составить расписание”

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Составить расписание |
| Успешный сценарий:   * Пользователь нажимает «Составить расписание». * Приложение выводит сообщение об успешном составлении расписания.   Альтернативный сценарий (Наступает при отсутствии данных):   * Пользователь нажимает «Составить расписание». * Приложение выводит сообщение об ошибке. | |
| Успешный результат | Расписание составлено. |
| Отрицательный результат | Вывод сообщения об ошибке. |

Таблица 9. Сценарий использования “Посмотреть расписание”

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Просмотр расписание |
| Успешный сценарий:   * Пользователь нажимает «Показать расписание». * Приложение выводит расписание.   Альтернативный сценарий (Наступает при отсутствии данных):   * Пользователь нажимает «Показать расписание». * Приложение выводит сообщение об ошибке. | |
| Успешный результат | Демонстрация расписания. |
| Отрицательный результат | Вывод сообщения об ошибке. |

Таблица 10. Сценарий использования “Экспорт расписания”

|  |  |
| --- | --- |
| Действующие лица | Пользователь |
| Цель | Экспорт расписания |
| Успешный сценарий:   * Пользователь нажимает «Экспорт расписания». * Приложение выводит поле экспорта. * Пользователь выбирает формат. * Пользователь нажимает «Экспортировать». * Расписание сохраняется.   Альтернативный сценарий (Наступает при отсутствии данных):   * Пользователь нажимает «Экспорт расписания». * Приложение выводит сообщение об ошибке. | |
| Успешный результат | Сохраненное расписание. |
| Отрицательный результат | Вывод сообщения об ошибке. |

**Архитектурный план**

# 1. Введение

При разработке программного продукта задачей разработчиков является проектирование архитектуры приложения.

## 1.1. Цель

Документ Архитектурный план призван провести краткий обзор разрабатываемой системы и описать различные архитектурные решения, принятые при проектировании ПО.

## 1.2. Контекст

Архитектурный план предоставляет обзор архитектуры приложения «Cinema3000». «Cinema3000» разработан командой в рамках курса «Объектно-ориентированные технологии разработки программного обеспечения».

## 1.3. Определения, акронимы, сокращения

См. глоссарий проекта.

# 2. Архитектурные цели и ограничения

Основной целью разработки является процесс создания рабочего продукта в соответствии с методологией разработки RUP. К реализации продукт имеет следующие требования и ограничения:

1) Наличие аллокатора.

2) Наличие итератора.

3) Наличие системы классов исключительных ситуаций.

4) Возможность сериализации.

5) Наличие манипулятора.

6) Наличие инварианта и проверки предусловия, постусловия.

7) Пространство имен.

Этот документ описывает архитектуру программной системы с помощью UML-диаграмм:

* Use case диаграмма.
* Диаграмма компонентов.
* Диаграмма состояний.
* Диаграмма деятельности.
* Диаграмма коммуникации.
* Диаграмма классов.
* Диаграмма последовательности.

# 3. Представление вариантов использования

Use-case диаграмма, являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне, представлена на рис. 1.

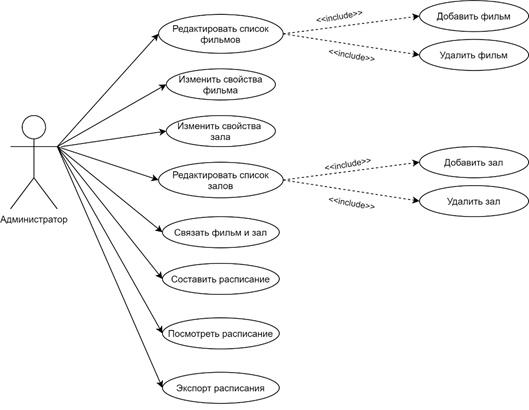


Рисунок 1 – Use-case диаграмма

# 4. Логическое представление

Для представления архитектуры используется диаграмма компонентов. Диаграмма компонентов показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п. Диаграмма компонентов представлена на рис. 2.

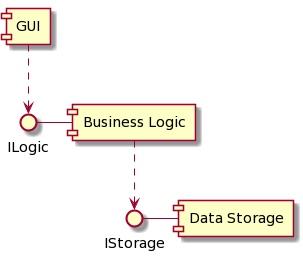


Рисунок 2 – Диаграмма компонентов

# 5. Представление развертывания

Программное обеспечение будет развертываться на ОС Microsoft Windows 7 и более старших версиях данной операционной системы. Диаграмма развёртывания представлена на рис. 3.

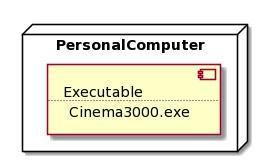


Рисунок 3 – Диаграмма развёртывания

# 6. Представление поведения

Основное представление поведения показано на диаграммах деятельности, состояний и коммуникации. В частности, на ней изображаются участвующие во взаимодействии объекты и действия. На рис. 4, 5 и 6 представлены диаграммы деятельности в соответствии с процессами деятельности. На рис. 7 представлена диаграмма состояний. Диаграмма коммуникации показана на рис. 8.



Рисунок 4 – Диаграмма деятельности (1)

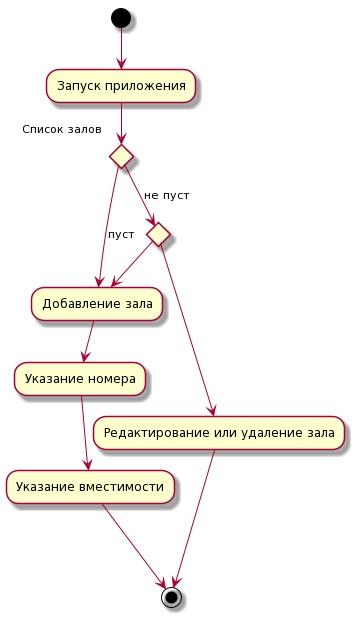


Рисунок 5 – Диаграмма деятельности (2)

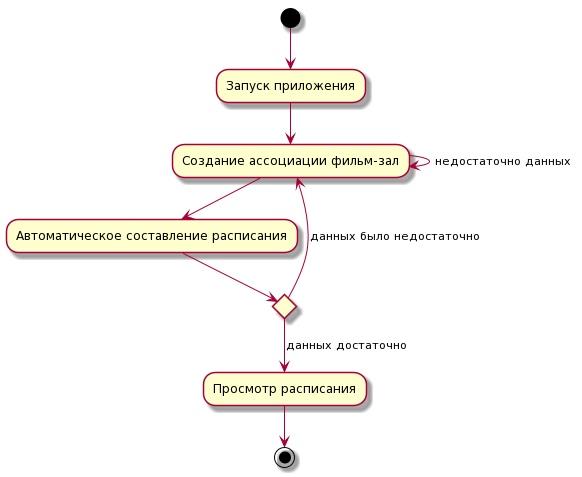


Рисунок 6 – Диаграмма деятельности (3)

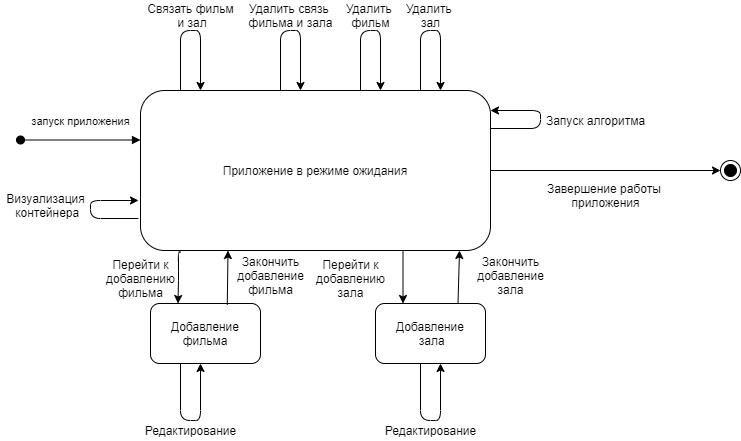


Рисунок 7 – Диаграмма состояний

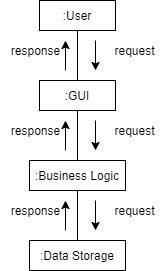


Рисунок 8 – Диаграмма коммуникации

# 7. Представление структуры

Для демонстрации классов системы, их атрибутов, методов и взаимосвязей между ними используется диаграмма классов. Диаграмма классов представлена на рис. 9.

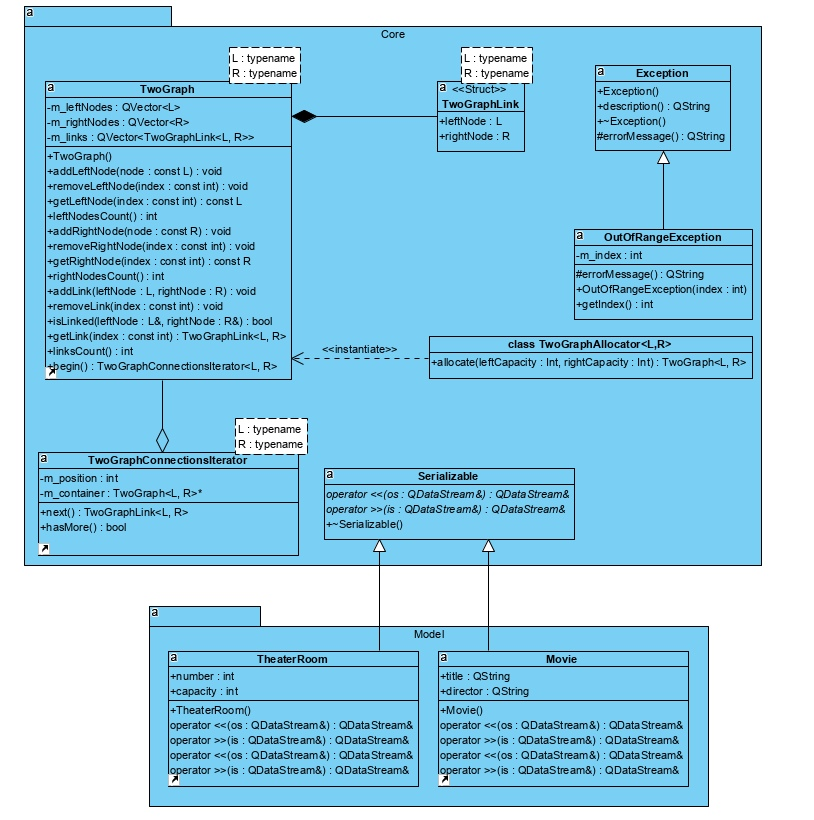


Рисунок 9 – Диаграмма классов

# 8. Диаграмма последовательностей

Диаграмма последовательностей представлена на рис. 10.

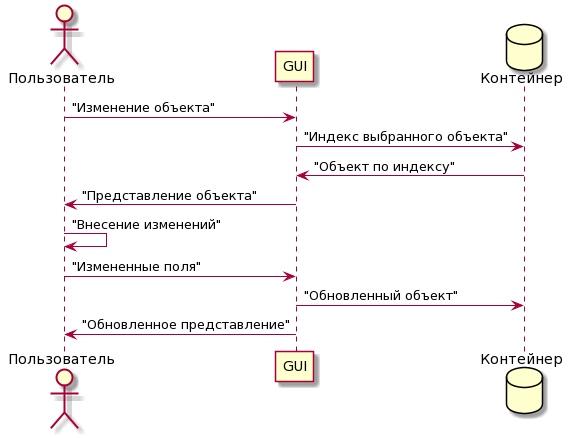


Рисунок 10 – Диаграмма последовательностей

# 9. Размер и работа

Размеры проекта ограничены учебным курсом. Временные рамки сужены, поэтому проект имеет достаточно большие размеры по сравнению со стандартными курсовыми проектами, но достаточно малые размеры относительно реальных проектов.

# 10. Качество

Целью разработки является освоение самого процесса разработки, поэтому к качеству предъявляются минимально возможные требования, с условием выполнения дополнительных функций, обеспечения надёжности работы программы.