МИНИСТЕРСТВО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И ЗАНЯТОСТИ НАСЕЛЕНИЯ ПРИМОРСКОГО КРАЯ

краевое государственное автономное

профессиональное образовательное учреждение

«Лесозаводский индустриальный колледж»

ДОПУСТИТЬ К ЗАЩИТЕ

Заместитель директора по УПР

\_\_\_\_\_\_\_\_\_\_\_\_\_Н. Н. Полосина

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

**ДИПЛОМНАЯ РАБОТА**

**на тему «Проектирование информационной системы по бронированию билетов в кинотеатре»**

|  |  |
| --- | --- |
| Работу выполнил: | Лозовой Илья Иванович |
| Группа: | ИС-4 |
| Специальность: | 09.07.02 Информационные системы и программирование |
| Руководитель: | Токарская Майя Сергеевна  \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) (подпись) |
| Нормоконтроль: | Губарева Юлия Александровна  \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата) (подпись) |

Лесозаводск, 2023

**Оглавление**

[ВВЕДЕНИЕ 5](#_Toc3345)

[1. Проектирование информационной системы 8](#_Toc16703)

[1.1. Предпроектное обследование 8](#_Toc14981)

[1.1.1. Анализ предметной области 8](#_Toc21808)

[1.1.2 Проблемы кинотеатра «Планета» 10](#_Toc23020)

[1.1.3 Сравнительный анализ существующих систем бронирования билетов в кинотеатрах 11](#_Toc21750)

[1.2 Математическая формализации, реинжиниринг, формирование требований 19](#_Toc29130)

[1.2.1 Анализ и выбор CASE - средств 19](#_Toc13410)

[1.2.2 Основные требования к информационной системе продажи и бронирования билетов в кинотеатре «Планета» 21](#_Toc6288)

[1.3 Проектирование информационной системы продажи и бронирования билетов в кинотеатре «Планета» 22](#_Toc17050)

[1.3.1 Выбор архитектуры информационной системы 22](#_Toc29037)

[1.3.2 Проектирование информационной структуры 23](#_Toc3085)

[2. разработка ИС 27](#_Toc32020)

[2.1 Обоснование выбора типа СУБД 27](#_Toc1205)

[2.2 Обоснование выбора среды разработки 31](#_Toc17759)

[2.2.1 Microsoft Visual Studio Code 2022 (VS Code) 32](#_Toc1449)

[2.2.2 Sublime Text 4 34](#_Toc25381)

[2.3 Обоснование выбора инструментов реализации. 37](#_Toc16404)

[2.3.1 HTML 37](#_Toc6092)

[2.3.2 CSS 38](#_Toc6409)

[2.3.3 JavaScript. 40](#_Toc16471)

[2.3.4 PHP 42](#_Toc22021)

[2.3.5 Adobe Photoshop 43](#_Toc28035)

[2.3.6 Figma 44](#_Toc4281)

[2.4 Расчёт затрат на проектирование и разработку 45](#_Toc28794)

[2.5 Техническое задание на разработку ИС по ГОСТ 19.201-78 49](#_Toc8638)

[2.5.1 Введение 49](#_Toc22096)

[2.5.1.1 Наименование программы 49](#_Toc29100)

[2.5.1.2 Краткая характеристика области применения 49](#_Toc32206)

[2.5.1.3 Основание для разработки 49](#_Toc25209)

[2.5.1.4 Назначение для разработки 49](#_Toc28213)

[2.5.2 Требования к программе или программному изделию 50](#_Toc24309)

[2.5.2.1 Общие требования 50](#_Toc19786)

[2.5.2.2 Требования к дизайну сайта 50](#_Toc2882)

[2.5.2.4 Требования к содержимому сайта 52](#_Toc5731)

[2.5.3 Технические требования к информационному продукту 52](#_Toc17936)

[2.5.3.1 Требования к организации входных и выходных данных 52](#_Toc9031)

[2.5.3.2 Требования к временным характеристикам 53](#_Toc26781)

[2.5.3.3 Требования к надёжности 53](#_Toc23709)

[2.5.3.4 Требования к обеспечению надежного (устойчивого) функционирования программы 53](#_Toc25543)

[2.5.3.4 Время восстановления после отказа 53](#_Toc22329)

[2.5.3.5 Условия эксплуатации 54](#_Toc2749)

[2.5.3.6 Климатические условия эксплуатации 54](#_Toc3224)

[2.5.3.7 Требования к видам обслуживания 54](#_Toc8196)

[2.5.3.8 Требования к численности и квалификации персонала 54](#_Toc4149)

[2.5.3.9 Требования к составу и параметрам технических средств 54](#_Toc8182)

[2.5.3.10 Требования к информационной и программной совместимости 55](#_Toc28839)

[2.5.3.11 Требование к маркировке и упаковке 55](#_Toc23524)

[2.5.3.12 Требования к транспортированию и хранению 55](#_Toc10928)

[2.5.3.13 Специальные требования 55](#_Toc21365)

[2.5.3.14 Требования к программной документации 55](#_Toc1432)

[2.5.4 Стадии и этапы разработки 55](#_Toc26124)

[2.5.5 Порядок контроля и приемки 56](#_Toc11146)

[2.6 Оптимизация работы онлайн-сервиса 57](#_Toc1071)

[Заключение 63](#_Toc9688)

[Социальная значимость разработки 63](#_Toc2368)

[Приложение А. Диаграммы IDEF-0 65](#_Toc25675)

[Приложение Б. Диаграммы IDEF-0 «TO-BE» 67](#_Toc26647)

[Приложение В. Макеты страниц 69](#_Toc28638)

# ВВЕДЕНИЕ

Современные информационные системы являются не просто средством автоматизации и повышения эффективности, но неотъемлемым элементом архитектуры компании. Организации все чаще вкладывают значительные средства в системы, способные помочь компании выжить в стремительно изменяющейся внешней среде и условиях конкуренции. Однако, что же именно способны дать предприятию информационные системы, и как предприятие может в свою очередь способствовать развитию ИС и максимизации выгод от их применения? Для ответа на этот вопрос необходимо рассмотреть само понятие информационной системы.

Наиболее часто цитируется определение М. Р. Когаловского:

«Информационной системой называется комплекс, включающий вычислительное и коммуникационное оборудование, программное обеспечение, лингвистические средства и информационные ресурсы, а также системный персонал, обеспечивающий поддержку динамической информационной модели некоторой части реального мира для удовлетворения информационных потребностей пользователей».

В настоящее время ИС используются для автоматизации информационных процессов практически во всех сторонах жизни общества, в том числе и кинотеатрах.

С каждым днем кинотеатры все больше и больше становятся неотъемлемой частью современного общества. Сейчас в городе есть много развлекательных учреждений, которые вынуждены конкурировать как друг с другом. Не секрет, что каждый стремится привлечь к себе как можно больше посетителей – кинотеатр «Планета» не исключение.

А это достигается за счёт таких факторов как:

* Правильно проведённая маркетинг компания:
  + Теле- и радио- реклама;
  + Баннеры;
  + Листовки;
  + СМС рассылки и т.д.
* Качество и скорость обслуживания.

В данном дипломном проекте будет рассматриваться такой аспект как привлечение аудитории.

Целью дипломной работы является создание web-сервиса киноцентра для отображения в онлайн режиме актуальной информации о транслируемых кинофильмах в залах киноцентра, а также о наличие свободных билетов на них с возможностью онлайн бронирования билетов.

Объектом исследования является деятельность кинотеатра «Планета»

Предмет исследования: возможность онлайн заказа и бронирования билета на кинофильм

Задачи:

* Описание предметной области
* Построение организационной структуры
* Разработка сценариев работы приложения
* Составление функционально-ориентированной модели
* Формулировка требований к ИС
* Составление экономической модели
* Разработка технического задания
* Разработка макета сайта
* Разработка сайта со стороны клиента и администратора
* Провести тестирование разработанного инструмента

Для решения задач были применены следующие методы:

* Теоретический анализ и синтез позволил рассмотреть изучаемый объект по частям, а затем объединить данные в общее знание о нем;
* Сравнение для выявления общего и отличий в данных разных источниках об предмете изучения;
* Индукция (восхождение от частного к общему, обобщение) необходима, чтобы на основе данных разных материалов сделать общий теоретический вывод;
* Дедукция (конкретизация, движение от общего к частному) для распространения общего вывода на отдельные случаи;
* С помощью классификации выделил существующие данные в структуру;

# Проектирование информационной системы

## Предпроектное обследование

### Анализ предметной области

Кинотеатр – это общественное учреждение для публичного показа кинофильмов. Главное помещение кинотеатра – зрительный зал со специальным широким экраном размером до 30 метров и акустической системой. В современных кинотеатрах часто имеется несколько зрительных залов, обязательна система кондиционирования воздуха, а современные акустические системы состоят из множества раздельных звуковых каналов. Для высококачественного кинотеатра характерны акустически приспособленные стены и потолок, декоративное освещение. В кинотеатрах обычно также имеются фойе для зрителей, гардероб, буфет, служебные помещения. Ранее кинотеатры СССР были довольно крупными и вмещали до 2500-4000 зрителей единовременно. Более современные кинотеатры рассчитаны на меньшее количество зрителей, обычно по 200-300 посадочных мест в одном кинозале. Кинопроекционный комплекс кинотеатра зачастую состоит из одного или нескольких кинопроекторов для демонстрации фильмов с кинопленки шириной 35 мм, но есть и специальные кинотеатры, приспособленные для демонстрации трехмерного кино.

Кинотеатр «Планета» - располагается на базе ДК Ружинского м-р, которое является обособленным структурным подразделением МБУК «Культурно-досуговый центр», который расположен на улице Урицкого, д. 14. Он имеет прекрасно оформленный зал, который оборудован на 400 посадочных мест, размещённых в 20 рядов.



Рисунок 1.1 - Организационная структура кинотеатра «Планета»

Из «фишек» в кинотеатре «Планета» присутствуют: цифровой звук DolbyDigitalSurroundEX, система вентиляции и кондиционирования. Кинотеатр располагает большой парковкой.

В киноцентре имеется один зал, в котором осуществляется показ фильма (или мультфильма).

Для того, чтобы зрителя пустили в какой-либо зал для просмотра фильма (или мультфильма), у него должен быть входной билет, в котором указывается название фильма, зал, в котором будет транслироваться фильм, а также дата и время сеанса и место зрителя в зале на время сеанса. По каждому билету может пройти только один зритель. Перед началом сеанса работник киноцентра открывает двери в зал и пускает зрителей, предварительно проверяя у каждого из них наличие билета. Во избежание входа в зал по поддельному билету на каждый билет наносится уникальный шрихкод, который контролер считывает специальным портативным сканером.

Факт проверки билета сотрудником отмечается им путем специальной части билета, предназначенной именно для этого.

Билеты приобретаются в кассах киноцентра непосредственно перед сеансом, либо за три дня до него. Для покупки билета никаких документов предоставлять не требуется, за исключением сеансов с возрастным ограничением. Для покупки билета на такой сеанс необходимо предоставить любой документ, подтверждающий возраст зрителей (по требованию кассира). При покупке билета кассир на экран, размещенный над кассой, выводит план зала с отмеченными на нем свободными местами для того, чтобы покупатель сам выбрал себе место, или места (если покупает сразу несколько билетов). Каждое место в зале идентифицируется номером ряда и номером места в этом ряду. В случае, если покупатель не смог присутствовать на сеансе, то деньги за билет не возвращаются.

Вся деятельность кассиров по продаже билетов фиксируется в специальной информационной системе, созданной для автоматизации деятельности киноцентра. В данной ИС хранятся данные обо всех сеансах, а также о проданных билетах на каждый из них

### Проблемы кинотеатра «Планета»

В настоящее время посещаемость кинотеатра оставляет желать лучшего. Этому способствовали следующие факторы:

1. Уход многих зарубежных кинокомпаний из РФ;
2. Отдаленность кинотеатра «Планета» от г.Лесозаводск;
3. Отсутствие возможности оплаты безналичным способом и по Пушкинской карте.

В качестве решения второй проблемы мы предложили создать систему для онлайн-бронирования и заказа билетов. Кроме этого, наличие яркого и красочного сайта будет способствовать привлечению большей аудитории.

В данном дипломном проекте ставится задача разработки инструмента, который позволил бы привлечь большее количество посетителей.

### Сравнительный анализ существующих систем бронирования билетов в кинотеатрах

Для решения поставленной задачи по созданию информационной системы бронирования билетов кинотеатра «Планета» был проведен анализ работы отдела по работе с клиентами, в который и будет внедряться разрабатываемая ИС. В процессе анализа были выявлены существующие проблемы, на основании которых и были получены требования и специфика разрабатываемой ИС, формальная оценка существующих бизнес процессов и проведение их реинжиниринга будет произведено в разделе 2, а пока рассмотрим существующие решения в данной предметной области.

Выбор системы с помощью метода анализа иерархий

На сегодняшний день во многих сферах деятельности для решения задач аналитического планирования широко используется метод анализа иерархий, созданный американским ученым Т. Саати.

Для объективности выбора ИС среди аналогов воспользуемся данным методом.

Сначала определяем перечень критериев, по которым будет осуществлён выбор ИС, а затем указываем для каждого из рассматриваемых вариантов оценки по каждому критерию.

Иерархия строится с вершины – цели анализа, через промежуточные уровни (критерии, по которым производится сравнение вариантов) к нижнему уровню (который является перечислением альтернатив).

Имеются три информационные системы: ИС «БроньМастер», ИС «Эверест» и **«**Разрабатываемая информационная система».

Цель анализа – это выбор информационной системы, которая будет соответствовать требованиям отдела по работе с клиентами, повысит гибкость и эффективность работы.

Для достижения цели произведём сравнение каждого критерия из альтернативных вариантов попарно.

Сравнительный анализ программ-аналогов и разрабатываемой информационной системы сведены в таблицу 1.1.

Таблица 1 - Сравнительный анализ программ-аналогов и разрабатываемой информационной системы.

| Функции | ИС «БроньМастер» | ИС «Эверест» | Разрабатываемая система |
| --- | --- | --- | --- |
| Удобство интерфейса | среднее | низкое | высокое |
| Скорость обработки запросов | низкая | средняя | высокая |
| Требования к аппаратным средствам | средние | высокие | низкое |
| Уровень защищенности | средний | низкий | высокий |
| Надёжность | высокая | средняя | высокая |

Начнём с построения матрицы попарных сравнений для критериев, т.е. со второго уровня иерархии (на первом уровне наша цель - выбор информационной системы, на третьем - альтернативы). Заполняя таблицу 1.2, попарно сравниваю критерий из строки с критерием из столбца по отношению к цели - выбору информационной системы. Значения из шкалы относительной важности вписываю в ячейки, образованные пересечением соответствующей строки и столбца. Относительные веса критериев сведены в таблицу 1.2.

Таблица 2 - Относительные веса критериев.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Удобство интерфейса | Скорость обработки запросов | Требования к аппаратным средствам | Уровень защищенности | Надёжность | Оценка компонент собственного вектора | Нормализованные оценки вектора приоритета | max |
| Удобство интерфейса | 1 | 1/3 | 1/5 | 1/5 | 1/8 | 0,278208 | 0,037392 | 0,822624 |
| Скорость обработки запросов | 3 | 1 | 1/3 | 1/5 | 1/7 | 0,491119 | 0,066008 | 1,452173 |
| Требования к аппаратным средствам | 5 | 3 | 1 | 1/3 | 1/3 | 1,107566 | 0,148860 | 3,274927 |
| Уровень защищенности | 5 | 5 | 3 | 1 | 1/5 | 1,718772 | 0,231008 | 5,082181 |
| Надежность | 8 | 7 | 3 | 5 | 1 | 3,844642 | 0,516732 | 11,368095 |
| ИТОГО: |  |  |  |  |  | 7,440307 |  | 5,121036 |

Заполнив Таблицу 1.2, сначала определяю оценки компонент собственного вектора, которые получаются как произведение относительных весов критерия по горизонтали, возведенного в степень 1/5 (где 5 – количество критериев). Например, рассчитаю оценку собственного вектора для критерия «Удобство интерфейса»:

(1\*1/3\*1/5\*1/5\*1/8\*)1/5 = 0,278208

Аналогично определяю остальные критерии.

Для того же критерия «Удобство интерфейса» вычисляю нормализованные оценки вектора приоритета, разделив оценки собственного вектора на их сумму, равную 7,440307.

0,278208/7,440307 = 0,037392

Так же рассчитываю остальные критерии.

Получив сумму по каждому критерию в столбцах, вычисляю значение матрицы, умножая нормализованные оценки вектора приоритета на эту сумму. Например, для критерия «Удобство интерфейса»:

22\*0,037392= 0,822624

Просуммировав, получаем максимальное собственное значение матрицы (лmax), л max = 5,121036

Определяю наибольшее значение критерия информационных систем, сравнительный анализ которых сведён в таблицу 1.3.

Таблица 1.3 – Сравнение критериев информационных систем.

|  |  |
| --- | --- |
| Критерии | Нормализованные оценки вектора приоритета |
| Удобство интерфейса | 0,037392 |
| Скорость обработки запросов | 0,066008 |
| Требования к аппаратным средствам | 0,14886 |
| Уровень защищенности | 0,231008 |
| Надежность | 0,516732 |

Сравнивая нормализованные оценки вектора приоритета можно сделать вывод, что наибольшее значение я придаю критерию «Надежность».

Далее определяю индекс согласованности (ИС), который дает информацию о степени нарушения согласованности, т.е. проверяю, насколько мои суждения были непротиворечивыми при составлении матрицы попарных сравнений критериев.

ИС = (л max - n)/(n - 1),

где лmax - максимальное собственное значение матрицы (лmax ≥ n),

n - размерность матрицы

ИС = (5,121036– 5)/ (5-1) = 0,030259

Разделив ИС на число, соответствующее случайной согласованности матрицы пятого порядка, равного 1,12, получим отношение согласованности (ОС). Величина ОС должна быть порядка 10% или менее, чтобы быть приемлемой. В некоторых случаях допускается ОС до 20%, но не более, иначе надо проверить свои суждения.

ОС = 0,030259 / 1,12 = 2,7% < 10%, т.е. пересматривать свои суждения нет нужды.

Следующим шагом выполняется сравнение информационных систем по каждому критерию отдельно. Данные об информационных системах по перечисленным выше критериям представлены в таблице 1.1.

Построю матрицу сравнений, сравнивая попарно альтернативу из строки с альтернативой из столбца по отношению к критерию «Удобство интерфейса». Сравнительные оценки систем по критерию «Удобство интерфейса» сведены в таблицу 1.4. Никакие другие критерии при этом не учитываю.

Значения из шкалы относительной важности вписываю в ячейки, образованные пересечением соответствующей строки и столбца. Диагональ этой матрицы заполняю значением «1», а ячейки, лежащие ниже диагонали – обратными значениями.

Относительная согласованность матрицы равна 7,4%, т.е. <10%.

Далее так же построю матрицу сравнений, сравнивая попарно альтернативу из строки с альтернативой из столбца по отношению к критерию «Скорость обработки запросов». Сравнительные оценки систем по критерию «Скорость обработки запросов» сведены в таблицу 1.5. Никакие другие критерии при этом не учитываю.

Таблица 1.4 – Сравнительные оценки систем по критерию «Удобство интерфейса».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ИС БроньМастер | ИС "Эверест" | Разрабатываемая система | Оценка компонент собственного вектора | Нормализованные оценка вектора приоритета | л мах |
| ИС БроньМастер | 1 | 1/2 | 1/5 | 0,464159 | 0,106839 | 0,870688 |
| ИС "Эверест" | 2 | 1 | 1/6 | 0,693361 | 0,159596 | 1,219346 |
| Разрабатываемая система | 5 | 6 | 1 | 3,107233 | 0,715214 | 0,995732 |
| ИТОГО: | 8 | 7,5 | 1 11/30 | 4,344481 |  | 3,085766 |

Таблица 1.5 – Сравнительные оценки систем по критерию «Скорость обработки запросов».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ИС БроньМастер | ИС "Эверест" | Разрабатываемая система | Оценка компонент собственного вектора | Нормализованные оценка вектора приоритета | л мах |
| ИС БроньМастер | 1 | 1/2 | 1/5 | 0,464159 | 0,121905 | 0,976162 |
| ИС "Эверест" | 2 | 1 | 1/3 | 0,873580 | 0,229433 | 1,033429 |
| Разрабатываемая система | 5 | 3 | 1 | 2,466212 | 0,647715 | 0,994104 |
| ИТОГО: | 8 | 4,5 | 1 8/15 | 3,807557 |  | 3,003695 |

Относительная согласованность матрицы равна 0,32%, т.е. <10%.

Далее так же построю матрицу сравнений по отношению к критерию «Требования к аппаратным средствам». Сравнительные оценки систем по критерию «Требования к аппаратным средствам» сведены в таблицу 1.6.

Таблица 1.6 – Сравнительные оценки систем по критерию «Требования к аппаратным средствам».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ИС БроньМастер | ИС "Эверест" | Разрабатываемая система | Оценка компонент собственного вектора | Нормализованные оценка вектора приоритета | л мах |
| ИС БроньМастер | 1 | 6 | 1/2 | 1,44225 | 0,33873 | 1,07785 |
| ИС "Эверест" | 1/6 | 1 | 1/8 | 0,27516 | 0,06463 | 0,97408 |
| Разрабатываемая система | 2 | 8 | 1 | 2,51984 | 0,59182 | 0,96637 |
| ИТОГО: | 3 1/6 | 15 | 1 5/8 | 4,25777 |  | 3,01830 |

Относительная согласованность матрицы равна 1,58%, т.е. <10%.

Далее построю матрицу сравнений по отношению к критерию «Уровень защиты». Сравнительные оценки систем по критерию «критерию «Уровень защиты» сведены в таблицу 1.7.

Таблица 1.7 – Сравнительные оценки систем по критерию «критерию «Уровень защиты».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ИС БроньМастер | ИС "Эверест" | Разрабатываемая система | Оценка компонент собственного вектора | Нормализованные оценка вектора приоритета | л мах |
| ИС БроньМастер | 1 | 5 | 1/2 | 1,357209 | 0,33037 | 1,066291 |
| ИС "Эверест" | 1/5 | 1 | 1/7 | 0,305711 | 0,07442 | 0,975738 |
| Разрабатываемая система | 2 | 7 | 1 | 2,410142 | 0,58667 | 0,972124 |
| ИТОГО: | 3 1/5 | 13 | 1 2/3 | 4,108160 |  | 3,014153 |

Относительная согласованность матрицы равна 1,22%, т.е. <10%.

Так же построю матрицу сравнений по отношению к критерию «Надежность».

Сравнительные оценки систем по критерию «Надежность» сведены в таблицу 1.8.

Таблица 1.8 – Сравнительные оценки систем по критерию «Надежность».

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ИС БроньМастер | ИС "Эверест" | Разрабатываемая система | Оценка компонент собственного вектора | Нормализованные оценка вектора приоритета | л мах |
| ИС БроньМастер | 1 | 1 | 1/3 | 0,69336 | 0,20000 | 1,000000 |
| ИС "Эверест" | 1 | 1 | 1/3 | 0,69336 | 0,20000 | 1,000000 |
| Разрабатываемая система | 3 | 3 | 1 | 2,08008 | 0,60000 | 1,000000 |
| ИТОГО: | 5 | 5 | 1 2/3 | 3,46681 |  | 3,000000 |

Относительная согласованность матрицы равна 0%, т.е. <10%.

Результаты оценок информационных систем по всем критериям сведены в таблицу 1.9, т.е. в самую верхнюю строку перенесу из таблицы 1.2 значения вектора приоритета для каждого критерия.

Для каждой из альтернатив заполняю столбцы критериев значениями локальных векторов приоритета, полученных соответственно в таблицах 1.4; 1.5; 1.6; 1.7; 1.8.

Далее подсчитываю значения глобального приоритета для каждой из альтернатив как сумму произведений значения вектора приоритета для критерия и значения вектора локального приоритета этой альтернативы в отношении данного критерия, т.е. для альтернативы ИС «БроньМастер» это будет:

0,034763 \* 0,0108836 + 0,061367\* 0,12202+ 0,138393\* 0,34037+ 0,214765\* 0,333216+ 0,480398\* 0,20000 = 0,226019027

Аналогично определяю остальные альтернативы.

Выбранной альтернативой считается альтернатива с максимальным значением глобального приоритета. Сравнения информационных систем сведены в таблицу 1.10.

Таблица 1.9 – Сравнительные оценки систем по всем критериям.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Альтернативы | Удобство интерфейса | Скорость обработки запросов | Требования к аппаратным средствам | Уровни защищенности | Надежность | Глобальные приоритеты |
| Численное значение вектора приоритета | | | | | | |
|  | 0,034763 | 0,061367 | 0,138393 | 0,214765 | 0,480398 |  |
| ИС БроньМастер | 0,106839 | 0,121905 | 0,33873 | 0,33037 | 0,2 | 0,225104362 |
| ИС "Эверест" | 0,159596 | 0,229433 | 0,06463 | 0,07442 | 0,2 | 0,140634402 |
| Разрабатываемая система | 0,715214 | 0,647715 | 0,59182 | 0,58667 | 0,6 | 0,560750038 |

Таблица 1.10– Сравнение информационных систем

|  |  |
| --- | --- |
| Наименование ИС | Глобальные приоритеты |
| ИС БроньМастер | 0,22510436 |
| ИС "Эверест" | 0,1406344 |
| Разрабатываемая система | 0,56075004 |

В данном случае это «Разрабатываемая система», на которой следует остановить свой выбор.

**Анализ существующих ИС показал, что разрабатываемая ИС будет отвечать более продуктивным показателям, низким требованиям к аппаратным средствам и удобству работы, отвечающему современным стандартам развивающихся технологий по сравнению с морально устаревшими и лишь частично отвечающими современным требованиям ИС.**

## Математическая формализации, реинжиниринг, формирование требований

### 1.2.1 Анализ и выбор CASE - средств

Для того чтобы начать процесс создания моделей системы, необходимо определиться с выбором средства, в котором данные модели будут создаваться.

Первый и основополагающим требованием к CASE - средству должна быть поддержка нотации UML версии 2.4.1 и свободное встраивание в модели произвольных графических объектов. Вторым немаловажным требованием является удобство применения CASE - средства.

На следующем этапе выберем несколько CASE - средств, среди которых будет производиться выбор. В данной работе в процессе анализа будут представлены следующие CASE - средства: BorlandTogether или MS Visio 2016. Все эти продукты поддерживают нотацию UML 2.4.1.

Достоинством MS Visio 2016 является тот факт, что данный продукт распространяется как надстройка к MS Office 2016. Размер дистрибутива у данного продукта не очень большой, что позволяет скачать его с сайта разработчика без каких-либо серьезных затрат.

Недостатком в данном CASE - средстве является тот факт, что оно поддерживает не все диаграммы нотации UML 2.4.1. Еще один минус – это отсутствие технической поддержки пользователей. Но! Большим плюсом является тот факт, что имеется возможность использовать не стандартные графические примитивы.

CASE-средство BorlandTogether поставляется в комплекте с пакетом BorlandDeveloperStudio.Данный пакет ориентирован на разработку приложений на языках Delphi, C++ и Java. Встроенное CASE-средство в данный пакет позволяет создавать модели и генерировать исходный код из диаграммы классов. Данный пакет изначально ориентирован на написание исходного кода приложения, его компиляцию и отладку. Функции разработки моделей в данном пакете являются дополнительными.

Стоимость данного пакета является высокой. Но возможность приобрести лицензионную версию доступна более широкому кругу разработчиков.

В идеальном случае для создания моделей и проведения моделирования предметной области необходимо использовать CASE-средство BorlandTogether. Но ввиду высокой стоимости и недоступности данного средства, в данной работе будет использовано CASE-средство MSVisio 20016. Оно в данном случае удовлетворяет всем требованиям и является доступным.

Таблица 1.2 Анализ CASE средств

|  |  |  |  |
| --- | --- | --- | --- |
|  | MS Visio 2016 | RationalRose 7.0 | BorlandTogether |
| Поддержка UML 2.4.1 и выше | + | + | + |
| Генерация кода программы | + | + | + |
| Работа в комплексе | - | + | + |
| Поддержка | - | + | + |
| Экспертная оценка | Удовлетворительно | Отлично | Хорошо |
| Размер дистрибутива | 350 Мбайт | 8 400 Мбайт | 4 500 Мбайт |
| Аппаратные требования | 512 Мб оперативной памяти, 400 Мб свободного места на HDD. | Минимум 1 Гб оперативной памяти, от 1200 Мб свободного места на HDD. | Минимум 1 Гб оперативной памяти (рекомендуется больший объем), 700 Мб свободного места HDD |
| Стоимость | Бесплатно, при условии покупки MsOffice | > 130 000 рублей | >55 000 рублей |

Проведя анализ достоинств и недостатков, представленных CASE-средств, можно сделать выбор какое средство необходимо использовать в данном случае. Я склоняюсь к использованию для создания модели работы объекта исследования и модели разрабатываемой системы CASE - средства MS Visio 2016, так как оно удовлетворяет требованиям по использованию нотации UML 2.4.1 в создаваемых моделях.

На основании вышеизложенного обоснования использования MS Visio были сделаны диаграммы IDEF-0 (см. Приложение А)

### 1.2.2 Основные требования к информационной системе продажи и бронирования билетов в кинотеатре «Планета»

Исходя из проведенных исследований сформированных предложений и сделанных выводов выявим основные требования к информационной системе продажи и бронирования билетов в кинотеатре:

* Централизованное хранилище информации о сеансах, свободных местах в залах, количестве проданных билетов;
* Возможность отображать информацию о состоянии свободных мест в различных кинозалах на информирующие билеты;
* Возможность продавать билеты через интернет.

Необходимо в ходе проектирования информационной системы предусмотреть формирования следующих страниц для разных групп пользователей:

* Администратора;
* Пользователь.

Разработку ИС будем осуществлять в формате сайта

## Проектирование информационной системы продажи и бронирования билетов в кинотеатре «Планета»

### 1.3.1 Выбор архитектуры информационной системы

По способу организации групповые и корпоративные информационные системы разделяются на следующие архитектуры:

* системы на основе архитектуры файл-сервер;
* системы на основе архитектуры клиент-сервер;
* системы на основе многоуровневой архитектуры;
* системы на основе Интернет/интранет-технологий.

Определим наиболее важные критерии для разрабатываемой ИС.

Для выявления подходящего средства разработки воспользуемся методом вариантных обоснований. Этот метод предназначен для выбора наилучшего варианта из нескольких предложенных и состоит из следующих этапов:

* Определение критериев, по которым будет произведено сравнение и степени их важности.
* Каждый вариант оценивается по полученному перечню критериев. Получается численное значение – оценка показателю качества по 5-ти бальной шкале.
* Нахождение общего количества баллов для каждого из вариантов (можно учитывать важность критериев).
* Лучшим считается вариант, который набрал максимальное количество баллов.

Вывод: в результате выполненного анализа архитектур выявили, наибольшее соответствие архитектуры на основе «интернет/интранет» для разрабатываемой ИС.

Таким образом, для реализации ИС выбор архитектуры останавливаем на архитектуре, основанной на интернет/интранет технологиях, так как в настоящее время данная архитектура получила признание и широкое распространение как способ организации приложений для рабочих групп и информационных систем корпоративного уровня. Подобная организация работы повышает эффективность выполнения приложений за счет использования возможностей сервера БД, разгрузки сети и обеспечения контроля целостности данных. Указанные преимущества делают архитектуру клиент-сервер наиболее подходящей для построения информационных систем, удовлетворяющих современным требованиям.

### 1.3.2 Проектирование информационной структуры

При выбранной архитектуре системы, вся информация должна храниться в единой базе данных. Поэтому необходимо преступить к проектированию такой базы данных. Для моделирования логической структуры воспользуемся методологией IDEF1x. Согласно данной методологии, процесс построения информационной модели состоит из следующих шагов:

* определение сущностей; определение зависимостей между сущностями;
* задание первичных и альтернативных ключей;
* определение атрибутов сущностей;
* приведение модели к требуемому уровню нормальной формы;
* переход к физическому описанию модели: назначение соответствий имя сущности – имя таблицы, атрибут сущности – атрибут таблицы;
* задание триггеров, процедур и ограничений;
* генерация базы данных.

Диаграмма сущность-связь, описывающая БД в терминах IDEF1.X, строится из трех основных блоков – сущностей, атрибутов и связей. Если рассматривать диаграмму как графическое представление правил предметной области, то сущности и атрибуты являются существительными, а связи – глаголами.

Логическая структура базы показана на рис. 3.2.



Рис. 3.1 - Логическая модель данных

Структура базы данных состоит из следующих сущностей:

* Сеанс. Сущность, характеризующая тот или иной временной сеанс в кинотеатре. Содержит необходимую информацию о каждом сеансе в следующих атрибутах:
  + IdСеанса – Уникальный идентификатор, который присваивается сеансу при его открытии в кинотеатре;
  + Кодовое название – обычно используется в маркетинговых целя;
  + Время начала – атрибут указывающий на час, минуту и секунду в которые начинается сеанс;
  + Время окончания – атрибут, который указывает на час, минуту и секунду в которые заканчивается сеанс;
* Фильм. Сущность, которая характеризует фильмы, показ которых проходит в кинотеатре. Содержит необходимую информацию о каждом кинофильме в следующих атрибутах:
  + Id Фильма – уникальный идентификатор, который присваивается фильму при его появлении в репертуаре кинотеатра;
  + Название фильма;
  + Актеры;
  + Компания производитель;
  + Продолжительность;
  + Жанр;
  + Id Сеанса – атрибут который служит для связи с сущностью сеанс;
* Кинозал. Сущность, которая характеризует кинозалы, имеющиеся в кинотеатре. Содержит всю необходимую информацию о каждом кинозале в следующих атрибутах:
  + Id Зала – Уникальный идентификатор, который присваивается каждому имеющемуся кинозалу;
  + Название зала;
  + Количество мест;
  + Id Фильма – атрибут который служит для связи с сущностью сеанс;
* Билет. Сущность, которая характеризует билеты продаваемые в кинотеатре. Содержит всю необходимую информацию о каждом проданном и не проданном билете в следующих атрибутах:
  + Id Билета;
  + Номер места;
  + Цена;
  + Дополнительные параметры;
  + Скидка;
  + Id Зала – атрибут, который необходим для связи с сущностью кинозал.
* Способ оплаты. Сущность которая характеризует способ оплаты билета. Содержит следующие атрибуты:
  + Id Оплата;
  + Наименование;
  + Id Билета – необходим для связи с сущностью билет.

Следующим шагом в разработке БД является переход от логической модели данных к физической. Используемая методология предполагает разработку реляционной БД, в которой физическая модель идентична логической. Заметим, что при переходе от логического уровня к физическому необходимо устранить связи «многие-ко-многим» посредством введения дополнительной сущности. Таким образом, физическая модель данных будет выглядеть, как показано на рисунке 3.3.

Таким образом, разработана структура БД, которая позволяет хранить всю необходимую для корректной работы системы информацию.

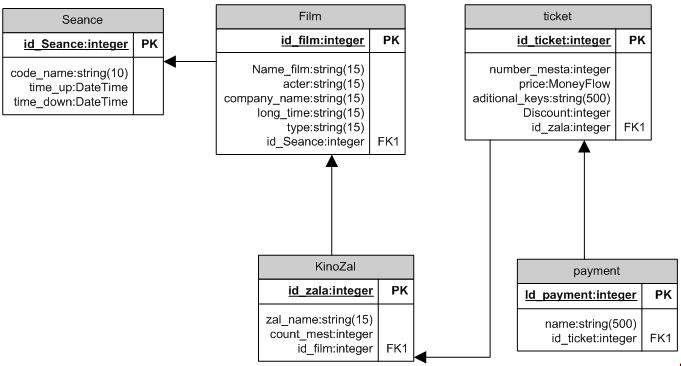


Рис. 3.3 Физическая модель данных

# 2. разработка ИС

## 2.1 Обоснование выбора типа СУБД

Система управления базами данных (СУБД) — специализированная программа (чаще комплекс программ), предназначенная для манипулирования базой данных. Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор.

Основные функции СУБД:

* Управление данными во внешней памяти (на дисках).
* Управление данными в оперативной памяти.
* Журнализация изменений и восстановление базы данных после сбоев.
* Поддержка языков БД (язык определения данных, язык манипулирования данными).

Обычно современная СУБД содержит следующие компоненты:

* Ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию.
* Процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода.
* Подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД.
* Сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

Классификация СУБД:

* + По модели данных:

По типу управляемой базы данных СУБД разделяются на:

* + Сетевые.
  + Иерархические.
  + Реляционные.
  + Объектно-реляционные.
  + Объектно-ориентированные.
  + По архитектуре организации хранения данных:
  + локальные СУБД (все части локальной СУБД размещаются на одном компьютере).
  + распределенные СУБД (части СУБД могут размещаться на двух и более компьютерах).
  + По способу доступа к БД:
  + Архитектура «файл-сервер» не имеет сетевого разделения компонентов диалога и использует компьютер для функции отображения, что облегчает построение графического интерфейса. «Файл-сервер» только извлекает данные из файлов, так что дополнительные пользователи добавляют лишь незначительную нагрузку на ЦП и каждый новый клиент добавляет вычислительную мощность сети. Минус: высокая загрузка сети.
  + Клиент-серверные СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера. Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера (что плохо для локальных программ — в них удобнее встраиваемые СУБД) и больших вычислительных ресурсах, потребляемых сервером.
  + Встраиваемая СУБД — библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных.

СУБД позволяют создавать, использовать и администрировать базы данных. СУБД представляют собой незаменимый инструмент разработчиков, бизнес-пользователей, администраторов БД и т.д. Благодаря тесной интеграции со средствами разработки приложений, СУБД используются разработчиками при создании информационных систем. [9].

Наиболее распространенные типы СУБД:

* OracleDatabase.
* Microsoft SQL Server.
* MySQL.

Рассмотрим данные типы СУБД более подробно:

1. Oracle: обладает возможностями кластеризации, мощными и экономичными средствами безопасности, полностью исключает потери данных и позволяет интерактивно обмениваться информацией. Основные достоинства - масштабируемость, безопасность, особенно важно, лидирует на платформах Unix и Windows. Пакет Oracle, наделенный самым развитым набором функций для работы с языком Java и доступа к данным через Интернет, системой оптимизации одновременного доступа. Единственным недостатком данной СУБД является сложность администрирования, однако все затраты на ее внедрение и освоение в последствии окупятся эффективной и надежной работой.
2. MicrosoftSQLServer: её важнейшие характеристики – это простота администрирования, возможность подключения к Web, быстродействие и функциональные возможности механизма сервера СУБД, наличие средств удаленного доступа. В комплект средств административного управления данной СУБД входит целый набор специальных мастеров и средств автоматической настройки параметров конфигурации. Также данная БД оснащена замечательными средствами тиражирования, позволяющими синхронизировать данные ПК с информацией БД и наоборот. Входящий в комплект поставки сервер OLAP дает возможность сохранять и анализировать все имеющиеся у пользователя данные. SQL Server уступает другим рассматриваемым СУБД по двум важным показателям: программируемость и средства работы. SQL Server функционирует только в среде Windows. Поэтому использование SQL Server целесообразно только если для доступа к содержимому БД используется исключительно стандарт ODBC. Microsoft SQL Server относятся к классу многопользовательских систем.
3. MySQL: является решением для малых и средних приложений. Входит в LAMP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, благодаря хорошей системе безопасности этого пакета, стабильной работе, высокому быстродействию и хорошей интеграции с соответствующими средствами программирования. В дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Разработчики MySQL всегда считали стабильность предметом особой важности.

Приведем перечень факторов, по которым так же была произведена оценка при обосновании выбора СУБД:

* Поддержка языка запросов - MySQL "понимает" команды языка SQL (StructuredQueryLanguage — структурированный язык запросов). Этот язык применяется во всех современных СУБД. MySQL также поддерживает интерфейс ODBC (OpenDatabaseConnectivity), протокол интерфейса с базами данных, разработанный компанией Microsoft.
* Возможности - сервер позволяет одновременно подключаться неограниченному количеству пользователей. Доступ к серверу СУБД MySQL можно осуществить в интерактивном режиме с помощью различных интерфейсов, позволяющих вводить запросы и просматривать полученные результаты: это программы-клиенты, работающие с командной строкой, Web-браузеры или программы-клиенты, работающие в системе X Window. Кроме того, в наличии имеются программные интерфейсы для таких языков, как C, Perl, Java, PHP и Python. Таким образом, можно использовать как готовое клиентское программное обеспечение, так и создавать свое собственное.
* Взаимодействие и безопасность - MySQL предназначена для работы в сети и может быть доступна через Internet, таким образом, с данными можно работать в любой точке земного шара. Но при этом СУБД MySQL снабжена развитой системой защиты от несанкционированного доступа.
* Переносимость - СУБД MySQL отлично работает как под управлением самых различных версий UNIX, так и под управлением систем, не использующих UNIX, таких как Windows и OS/2. СУБД MySQL работает как на домашних ПК, так и на мощных серверах.
* Открытое распространение - дистрибуция СУБД MySQL легкодоступна. Для этого достаточно воспользоваться Web-броузером. Если вы не понимаете, как что-либо работает, просмотрите исходный код. Если вам что-то в работе не нравится, можно внести коррективы.
* Дистрибуция MySQL включает следующие программные продукты - SQL-сервер - Эта программа обеспечивает доступ к базам данных.

PhpMyAdmin - веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования системы управления базами данных MySQL. Данное приложение позволяет через браузер осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять системой управления базами данных MySQL без непосредственного ввода SQL команд, предоставляя дружественный интерфейс.

Вывод: Исходя из представленных возможностей и потребностей разрабатываемой будет целесообразно использовать СУБД MySQL/PhpMyAdmin

## 2.2 Обоснование выбора среды разработки

Современные средства разработки программного обеспечения характеризуются большим разнообразием критериев, используя которые разработчик имеет возможность автоматизировать процесс разработки приложений. Так, в настоящее время инструментальные средства позволяют:

1. Создавать интерфейс, используя стандартные компоненты;
2. Передавать управление различным процессам, в зависимости от состояния системы;
3. Создавать оболочки для баз данных, как и сами базы данных;
4. Разрабатывать более надежное ПО, путем обработки исключительных ситуаций, возникающих при некорректной работе ПО.

Рассмотрим наиболее распространенные среды программирования от ведущих компаний-производителей, к ним можно отнести основание выбора среды программирования: Microsoft Visual Studio Code 2022, Sublime Text 4, Atom, Webstorm, Notepad++ и Vim.

Но я буду сравнивать два самых популярных из списка чтобы понять какой лучше подходит для использования в моей работе, это Microsoft Visual Studio Code 2022 и Sublime Text 3.

2.2.1 Microsoft Visual Studio Code 2022 (VS Code) — это редактор кода для разных языков программирования. Он относительно немного весит, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках.

VS Code не стоит путать с Visual Studio — это IDE, очень мощная и масштабная, но одновременно с этим тяжеловесная. Названия похожи, потому что у обоих средств разработки один и тот же создатель, но продукты разные по своей сути. VS Code меньше весит, проще в освоении и подходит в том числе для начинающих разработчиков.

Редактор кода существует для всех популярных операционных систем: Windows, Linux и macOS. Он бесплатный, в отличие от большинства версий «старшего брата» Visual Studio.

С VS Code работают программисты на разных языках. Например, им активно пользуются веб-разработчики, пишущие на HTML/CSS, JavaScript, PHP. Но редактор поддерживает намного большее количество языков: Python, Go, Ruby, C#, TypeScript и так далее. Он работает и с расширениями и фреймворками для популярных языков — например, с React JS и Vue.js, с языками стилей SCSS и LESS, которые дополняют CSS.

VS Code позволяет легко писать, форматировать и редактировать код на разных языках. С его помощью можно быстро создать проект и структуру файлов в нем, он подсвечивает синтаксис кода и помогает автоматически править ошибки. В нем есть возможности для отладки и запуска кода на некоторых языках.

Редактор легко расширяется, поэтому к перечисленным функциям можно добавить новые — достаточно просто скачать нужное дополнение из официального каталога. Дополнения тоже распространяются бесплатно.

Одна из самых популярных функций — функция автоматической отладки. Как правило, поиск ошибок в коде может занять много времени, но, нажав всего одну горячую клавишу, Visual Studio Code выделит ошибки, которые необходимо исправить. С добавлением этой удобной функции Visual Studio Code официально квалифицируется как IDE, что означает, что он может работать за пределами возможностей обычного текстового редактора. Visual Studio Code может отлаживать, записывать и управлять ресурсами в коде, таким образом обрабатывая приличный объём рабочей нагрузки.

Пользовательский интерфейс программного обеспечения является безупречным, практически без задержек, а текст и панель инструментов понятна.

**Особенности VS Code**

* VS Code позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ.
* В редакторе присутствуют встроенный отладчик, инструменты для работы с Git и средства рефакторинга, навигации по коду, автодополнения типовых конструкций и контекстной подсказки.
* Продукт поддерживает разработку для платформ ASP.NET и Node.js, и считается легковесным решение, которое позволяет обойтись без полной интегрированной среды разработки.
* Большим плюсом редактора является поддержка большого количества языков, таких как C++, C#, Python, PHP, JavaScript и других.

**Преимущества VS Code**

* **Простота освоения.** У редактора интуитивно понятный интерфейс, его легко установить и настроить. А после этого им сразу можно начинать пользоваться. Единственной проблемой поначалу могут стать горячие клавиши и дополнительные настройки — они необычно организованы. Но со временем легко разобраться, особенно если вначале держать под рукой памятку или пользоваться туториалами.
* **Малый вес.** По сравнению с IDE редактор быстро работает и мало весит, не так требователен к ресурсам компьютера. При этом он довольно функционален, а огромное количество расширений позволяет добавлять в него новые возможности.
* **Кроссплатформенность.** Visual Studio Code есть для всех популярных операционных систем, так что с ним можно работать вне зависимости от ОС, которую вы предпочитаете. К тому же это позволяет быстрее и удобнее переносить проекты с одного устройства на другое.
* **Поддержка разных языков.** Есть инструменты, которые предназначены специально для какого-то языка, например Python или C++. Visual Studio Code поддерживает огромное количество языков программирования, и они легко подключаются. Поэтому с его помощью можно вести проекты, написанные на нескольких языках одновременно.
* **Гибкость.** У VS Code множество расширений и удобных функций. Помимо вещей, стандартных для редактора кода, вроде подсветки синтаксиса, он «умеет» много чего еще. Его можно легко и гибко настроить под себя в зависимости от потребностей разработчика и технологий, с которыми он работает.

2.2.2 Sublime Text 4 - кроссплатформенный текстовый редактор для написания программного кода на различных языках программирования (Groovy, Erlang, C+, Java и т.д.), а также верстки веб-документов. Это проприетарное, распространяемое по платной лицензии ПО, имеющее бесплатную версию для ознакомления. За счет широкой функциональности, которую можно расширять путем подключения плагинов, а также кроссплатформенности Sublime Text популярен среди программистов и веб-разработчиков.

Возможности Sublime Text:

Этот текстовый редактор (правильнее его назвать редактором кода) изначально создавался с ориентацией на программистов и веб-разработчиков. Отсюда и соответствующий функционал, включающий следующие ключевые возможности:

* **Настраиваемый интерфейс.** Внешний вид редактора «Саблайм Текст» можно адаптировать под конкретные задачи. Например, для работы с несколькими файлами одновременно можно настроить режим, при котором сразу будут показываться до 4 панелей. Есть и полноценный (free modes) режим, при котором отображается одна рабочая область без дополнительных окон и панелей. В дополнение к этому можно настраивать визуальную часть интерфейса — для этого предусмотрены встроенные темы и возможность загрузки дополнительных. Для удобства работы также есть настраиваемые комбинации командных клавиш, палитра часто используемых (приоритетных) команд и т.д.
* **Множественное выделение.** Эта функция позволяет с помощью указателей, расставленных в различных частях кода, одновременно редактировать сразу несколько фрагментов. При этом есть возможность и одиночного редактирования с быстрым переключением на множественное. Это позволяет быстро редактировать сложно структурированный код без использования макрокоманд и тем самым ускорить его написание.
* **Автоматическое заполнение.** При написании кода редактор, в зависимости от выбранного языка программирования, предлагает несколько вариантов окончания записи. Кроме того, программа также завершает созданные программистом переменные. Это не только ускоряет написание кода, но и снижает вероятность ошибок.
* **Подсветка синтаксиса и высокая контрастность.** Фон рабочей области в Sublime Text можно сделать темным или светлым для более четкого отображения вводимого текста. Основные элементы синтаксиса выделяются различными цветами, что не только улучшает их визуальное восприятие, но и делает код более понятным разработчику. Также в процессе написания программы редактор автоматически в режиме реального времени подсвечивает синтаксис, уменьшая вероятность ошибки.
* **Заготовки (сниппеты).** При написании программы разработчики часто применяют одни и те же фрагменты кода. Чтобы не нужно было тратить время на их повторный набор, в Sublime Text можно сохранить такие повторяющиеся фрагменты для повторного использования, а также ключевые слова для их запуска.
* Раздельное редактирование. Если программист использует сразу несколько языков для различных частей программы, он может настроить в рабочей области отображение нескольких столбцов. Таким образом, не придется тратить дополнительное время на переключение между ними.
* **Подключение плагинов.** С помощью подключаемых модулей программист может серьёзно расширять базовый функционал редактора. Плагины в Sublime Text используются как для работы с различными языками программирования, так и для получения дополнительных возможностей, таких как совместная работа над одним проектом в нескольких редакторах (не обязательно Sublime), организация рабочих задач, подключение к системам управления версиями и т.д. Некоторые плагины абсолютно бесплатные, другие требуют регистрации, третьи полностью платные. Но их разнообразие покрывает практически все потребности разработчика.
* **Поддержка систем сборки.** В Sublime Text пользователь может собрать программу и запустить ее без обращения к командной строке. Есть возможность настроить автоматическую сборку, которая будет запускаться каждый раз при сохранении кода. Кстати, сохраняется код тоже автоматически в режиме реального времени. Что исключает потерю его фрагментов при внезапном прекращении работы редактора.

Недостатки Sublime Text:

* **Платная лицензия.** Стоимость 3-летней лицензии на Sublime Text составляет 99 долларов США. Достаточно большая сумма, однако она вполне оправдана широкими возможностями редактора. Кроме того, для него существует множество подключаемых бесплатных модулей. Также есть бесплатная версия для ознакомления, которая предлагает почти полный функционал.
* **Время загрузки программы.** Хотя редактор загружается достаточно быстро, по этому параметру он может уступать некоторым аналогам. Особенно это касается ранних версий, где плагины были встроены в общий процесс, а не запускались отдельно.
* **Ошибки в работе с плагинами.** Так как некоторые подключаемые модули разрабатываются сторонними программистами, в том числе непрофессионалами, при их работе могут возникать сбои. Причем в первых трех версиях «поломка» плагина часто приводила к аварийному прерыванию работы самого редактора.

На основании приведенных выше фактов было принято решение использовать в качестве среды разработки VS Code в виду его бесплатного распространения, встроенной системы контроля версий Git, большего количества плагинов

## 2.3 Обоснование выбора инструментов реализации.

В качестве инструментов реализации были выбраны язык гипертекстовой разметки HTML, каскадные таблицы стилей CSS, языки программирования JS, PHP, для создания логотипа был использован фоторедактор Adobe Photoshop, для создания макета Figma. А теперь расскажем о каждом чуть подробнее:

2.3.1 HTML — это язык разметки гипертекстовых документов. Он нужен, чтобы отображать в браузере специальным образом отформатированный документ с множеством вложенных элементов: заголовками, абзацами, списками, гиперссылками, медиаисточниками, расположением изображений, видео и аудио.

Дословно HTML означает Hypertext Markup Language (язык гипертекстовой разметки). Из расшифровки названия понятно, что инструмент применяется для управления отображением контента на интернет-странице, его структуризации.

Технология гипертекстовой разметки веб-страниц была предложена в 1989 году британским специалистом Тимом Бернерсом-Ли. Сначала язык применялся для обмена научной рабочей документацией между инженерами института CERN, сотрудником которого был Бернерс-Ли. Немного позднее применение языка HTML было расширено настолько, что он, наряду с такими базовыми элементами, как HTTP и URL лег в основу Всемирной паутины.

Когда пользователь посещает сайт, браузер «подтягивает» файл HTML с данными о структуре и содержании веб-страницы. Функция HTML состоит в выстраивании внешней базы, фундамента, но сам запуск сайта в функционал не входит. HTML только указывает, где должны располагаться элементы, каков их базовый визуал, где брать стили для элементов и скрипты.

HTML-документ можно составлять в любом редакторе, который есть в операционной системе: Notepad на MS Windows, TextEdit в Mac, Pico на Linux. Браузер для работы HTML–документа желателен, но необязателен. Он нужен для того, чтобы показать отформатированный документ.

HTML представляет собой основу внутренней структуры сайта, его базовый каркас. Необходимо учитывать, что этот код является не языком программирования, а инструментом для разметки гипертекста. С его помощью браузер выстраивает интернет-страницу в виде, который понятен для людей. HTML оптимален для начинающих программистов, он прост в освоении, а приобретённые навыки помогут уже в изучении языков программирования.

2.3.2 CSS — это язык описания внешнего вида документа, то есть он отвечает за то, как выглядят веб-страницы: цвет фона и декоративных элементов, размер и стиль шрифтов. Термин расшифровывается как Cascading Style Sheets (каскадные таблицы стилей). CSS взаимодействует с другим языком разметки — HTML, который отвечает за размещение элементов на странице.

Допустим, с помощью HTML текст уже расположен в верхней части документа. Задать для него цвет, тип шрифта и его кегль, сделать текст жирным или выделить курсивом можно и без использования CSS. Для этого в исходный код нужно добавить тег, который определяет начертание текста.

Использование тегов для форматирования текста в HTML загромождает исходный код, усложняет его, а значит, вероятность допустить в нем ошибку становится выше. Для того чтобы этого избежать, создали отдельный язык для стилевой разметки — CSS. Кроме размера и цвета шрифтов, этот язык разметки регулирует разделение заголовков, подзаголовков и основного текста, размер полей и отступов, отдельные цветовые фреймы для выделения текста, цвет основного фона, шапки и подвала.

CSS, как и любой язык, имеет свой синтаксис. В нем есть правила — значения, которые определяют внешний вид элементов. CSS-правило состоит из селектора, CSS-свойств и их значений:

* **Селекторы —** это метки, которые помогают браузеру понять, к какой части HTML-кода нужно применить заданные параметры.
* **CSS-свойства —** это определенные параметры оформления, например цвет элемента или текста (color) или цвет фона (background).
* **Значение —** это просто значение, оно выражается текстом или числом, например черный (black).

В примере селектором является <p>, и он выбирает все теги с именем <p>, color — это CSS-свойство а black — значение CSS-свойства. Связка «свойство: значение» называется блоком объявления стилей. Внутри него свойство отделяется от значения двоеточием, а один блок от другого отделяет точка с запятой.

Таблицы называются каскадными, потому что работают по принципу каскада — то есть правило, прописанное ниже, считается приоритетным.

2.3.3 JavaScript. — это язык программирования, который используют для написания frontend- и backend-частей сайтов, а также мобильных приложений. Часто в текстах и обучающих материалах название языка сокращают до JS. Это язык программирования высокого уровня, то есть код на нем понятный и хорошо читается.

JS поддерживают все популярные браузеры. Во frontend-части сайтов язык используют для создания интерактива (анимаций, всплывающих форм, автозаполнения), так как он связан с HTML и CSS и может ими манипулировать. В backend-части с языком JavaScript работают на платформе Node.js. С ее помощью, например, разрабатывают серверные веб-приложения и подключают библиотеки.

JavaScript называют языком скриптов или сценариев. Скрипты — это набор инструкций, которые выполняются при загрузке страницы. Браузер самостоятельно интерпретирует код на JavaScript, для этого даже не требуется компиляция (перевод языка программирования в машинный код).

Скрипты можно прописать внутри кода страницы или подключить к HTML отдельным файлом.

Но обычно для этого создается отдельный файл с расширением .js. Его называют script.js и прописывают код в нем.

JS — мультипарадигменный язык, так как он поддерживает разные парадигмы (стили) программирования:

* Объектно-ориентированный — представляет программу как систему отношений между объектами — наборами данных и функций.
* Императивный — в этом стиле программирования используют инструкции, которые выполняются последовательно, а для доступа к данным используются переменные, значение которых может быть переписано в ходе вычисления.
* Функциональный — в нем вычисление основано на математических функциях. Так как функциональное программирование не опирается на переменные, которые могут менять значение, результаты запросов будут всегда одинаковыми.

Веб-страницы частично обрабатываются с помощью JavaScript на компьютере пользователя. Это снижает нагрузку на сервер: часть операций выполняется без запросов к нему, что экономит время и трафик.

Где применяется JavaScript

* Клиентская часть веб—приложений (frontend). Это интерфейс страницы, то есть всё, что видит пользователь: контент, кнопки, формы обратной связи, меню. С помощью JS интерфейс реагирует на действия пользователя (клики мыши, нажатия клавиш), также язык отвечает за запоминание данных и автозаполнение форм.
* Серверная часть веб—приложений (backend). Серверный код пишут на платформе Node.js. На JS работают, например, запросы AJAX (asynchronous javascript and XML), которые отправляются на сервер в фоновом режиме, без перезагрузки веб-страницы, и push-уведомления — всплывающие сообщения в браузере, которые реализуются с помощью технологии Comet.
* Мобильные приложения на Android, iOS, Windows Mobile

Для чего нужен JS

* В веб-приложениях HTML отвечает за разметку страницы. Расположение контента, внешний вид интерфейса описывается с помощью CSS, а JavaScript отвечает за интерактивность и функционал этого интерфейса. Вот примеры задач, которые решает JS:
* Взаимодействие с пользователем. Всплывающие окна с полем для ввода почты или других данных, уведомления об использовании cookie.
* Обработка данных HTML. JS помогает заполнить необходимые поля в формах, проверять буквенные и числовые значения, а также управлять HTML-элементами на странице.
* Анимация. Это не только движущиеся объекты на странице и кнопки, но и карусель в галерее или плавная прокрутка страницы. Например, Pinterest с помощью JavaScript создал эффект бесконечной ленты.
* Математические вычисления. С помощью JS-скриптов на сайтах реализованы калькуляторы, например подсчет суммы заказа в корзине интернет-магазина.

2.3.4 PHP (Hypertext PreProcessor, «препроцессор гипертекста») — скриптовый язык программирования. Имеет открытый исходный код. Изначально создавался для разработки веб-приложений, но в процессе обновлений стал языком общего назначения.

Язык программирования PHP спроектировал датский программист Расмус Лердорф в 1995 году как инструмент для создания интерактивных и динамических веб-сайтов. Сейчас PHP активно используют веб-разработчики для создания сайтов и веб-приложений. Это наименее конкурентный язык для работы. Его использует большинство сервисов, и, скорее всего, ситуация не изменится в ближайшие 10 лет. Поэтому специалисты очень востребованы.

Основная область применения — разработка скриптов, которые работают на стороне сервера. Также можно создавать скрипты командной строки и GUI-приложения (приложения с графическим интерфейсом пользователя).

PHP используется в проектах разных масштабов: от браузерной игры и социальной сети до крупных веб-приложений в интернете, таких как Facebook, ВКонтакте, Google, W3C, Yahoo, NASA. На PHP написаны многие системы управления контентом (CMS), например Drupal, WordPress, Joomla. Также он использовался при разработке популярных фреймворков для создания сайтов, например Yii2, Laravel, Symfony.

PHP — язык с динамической типизацией. Это означает, что переменная не определяется жестко и заранее (если число, то int, если строка — string). Динамическая типизация позволяет присваивать одной и той же переменной несколько значений: она может быть массивом, строкой, числом, объектом и т. д. Динамическая типизация удобная и гибкая, но приводит к потреблению большого количества оперативной памяти и уменьшает скорость работы.

Также PHP — интерпретируемый язык. Это означает, что код, который написан на PHP, не компилируется (преобразуется в машинный), а выполняется строка за строкой с помощью программы-интерпретатора, которая читает каждую из них и выполняет то, что там указано.

CSS- и НТМL-файлы, как и файлы изображений, сервер отправляет сразу клиенту (браузеру), независимо от содержания. В отличие от них PHP-файлы содержат код, который интерпретируется на сервере. Клиент (браузер) получает не сам PHP-код, а результат его исполнения, который, как правило, является чистым CSS и HTML. Пользователь не может увидеть исходный код программы.

2.3.5 Adobe Photoshop – многофункциональный редактор для работы с фото и видеофайлами (растровые изображения и несколько векторных инструментов). Программный продукт работает на ПК с операционными системами macOS, Windows и в мобильных версиях iOS, Windows Phone и Android. Используется графический редактор для коммерческих целей (телевидение, кинематограф, реклама, игры, ретушь и так далее). Чтобы просматривать, редактировать и сохранять изображения и видеофайлы с использованием расширенного функционала, необходимо [купить лицензированный Adobe Photoshop](https://www.softmagazin.ru/soft/rabota-s-grafikoy/graficheskie-redaktory/tag-adobe-photoshop/" \t "https://www.softmagazin.ru/blog/vozmozhnosti_adobe_photoshop_obzor_versiy_i_dostoinstv/_blank" \o "купить лицензированный Adobe Photoshop).

Для работы полиграфистов, дизайнеров, фотографов, мультипликаторов, архитекторов и других творческих профессий важно иметь полный набор инструментов для коррекции мультимедийных файлов. Adobe Photoshop – необычный редактор, в котором можно не только корректировать и ретушировать изображения, но и преобразовывать формы, сохранять GIF-анимации, использовать слои и эффекты. Эти функции позволяют легко трансформировать, клонировать, улучшать изображения, применять различные фильтры для создания красочной, продающей картинки.

Основные возможности Adobe Photoshop

* Лицензированная версия Adobe Photoshop оснащена большим количеством функциональных возможностей для создания качественных изображений:
* улучшение качества отсканированных, отпечатанных снимков, устранение дефектов цветопередачи во время съемки;
* создание многослойного изображения с возможностью редактирования каждого в отдельности для передачи глубины цвета, работы со сложной графикой;
* монтирование фотографий и картинок, отдельных элементов в коллажи;
* восстановление старых снимков, ретушь фотографий;
* работа с эскизами и чертежами;
* большой выбор инструментов для обработки текстовых элементов (шрифты, эффекты, фильтры);
* обработка 2D и 3D моделей, добавление к ним текстур, работа с формами;
* создание интересных графических элементов для полиграфии, дизайна, оформления сайтов и так далее;
* подготовка к печати на принтере или в интернете и мгновенная публикация файлов;
* работа со стандартными форматами для растровых и векторных изображений;
* раскрашивание черно-белых файлов.

**Достоинства Adobe Photoshop**

Adobe Photoshop – основной инструмент компьютерной графики, который используется для всех видов коммерческой обработки медиа файлов. Суть Photoshop в том, чтобы скорректировать изображение, улучшить его восприятие, придать фото и видео четкости и гармоничности. Эта программа позволяет усилить эффект от картинки, создать сложные и красивые конструкции из простых элементов.

2.3.6 Figma — это популярный кроссплатформенный графический редактор для UX UI дизайнеров и веб разработчиков, с помощью которого можно создавать дизайн сложных интерфейсов, веб сайтов и мобильных приложений. А также в ней можно создавать интерактивные прототипы и небольшие анимации.

Figma является удобным графическим редактором для веб-дизайнеров и дизайнеров интерфейсов. Но по мимо дизайнеров, ей часто пользуются верстальщики и веб-разработчики для работы с макетами и переносом дизайна в HTML и CSS.

У Figma есть 2-е отличительные особенности:

* Возможность работать командой дизайнеров над одним проектом
* Доступ к интерфейсу и макетам прямо через браузер

Можно сказать, что Figma — это веб-сервис, а не программа. Благодаря своей простоте и удобству, большинство дизайнеров делают свой выбор в пользу Figma

Также у Figma есть огромное сообщество, которое с каждым годом только растет. Благодаря этому, Figma постоянно улучшает свой интерфейс и периодически показывает своим пользователям новые фичи, которые упрощают работу над созданием дизайна.

Весь интерфейс программы выполнен на английском языке.

Figma является условно бесплатным графическим редактором. Ведь помимо бесплатного тарифа у нее есть и платные.

Главное отличие Figma от Photoshop заключается в том, что Figma — это специализированная программа для разработки интерфейсов с удобным и простым функционалом. Photoshop является более сложным редактором и изначально задумывался лишь для обработки фотографий

## 2.4 Расчёт затрат на проектирование и разработку

Под проектированием будем понимать совокупность работ, которые необходимо выполнить, чтобы решить поставленную задачу – разработать алгоритм и реализовать его программно.

Для расчета затрат на этапе проектирования необходимо определить продолжительность каждой работы начиная с составления технического задания и заканчивая оформлением документации. Продолжительность работ определяется либо по нормативам (с использованием справочников), либо расчетом с помощью экспертных оценок по формуле (определение среднего времени продолжительности работ на каждом из этапов)

to = (3tmin +2tmax)/5,

где tо - ожидаемая длительность работ; tmin, tmax - наименьшая и наибольшая по мнению эксперта длительность работ.

Расчеты длительности всех работ на этапе проектирования сведены в таблицу 6.1.1

Для определения продолжительности этапа проектирования ТП по данным таблицы 6.1.1 построим график организации работ во времени.

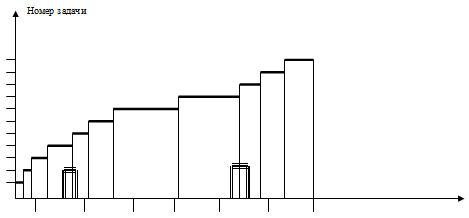


Рисунок 6.1.1 - Ленточный график

Таблица 6.1.1 - Длительность всех работ на этапе проектирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование работ | Длительность работ (дней) | | | Расход машинного времени |
| tmin | tmax | t0 | tM |
| 1. Разработка ТЗ | 8 | 12 | 5,2 | - |
| 2. Анализ ТЗ | 5 | 7 | 3,8 | - |
| 3. Поиск и изучение литературы | 3 | 8 | 5,0 | - |
| 4.Обзор существующих аналогов системы; | 7 | 14 | 9,8 | - |
| 5. Разработка основных этапов работы | 9 | 11 | 5,8 | - |
| 6. Разработка алгоритма | 10 | 14 | 11,6 | - |
| 7. Разработка программы | 25 | 40 | 31,0 | 190 |
| 8. Отладка работы программы | 21 | 31 | 25,0 | 100 |
| 9.БЖ и экологичность разработки | 10 | 15 | 7,0 | 35 |
| 10.Технико- экономическое обоснование работы | 9 | 14 | 5,8 | 30 |
| 11. Оформление пояснительной записки | 6 | 10 | 12,0 | 80 |
| Итого: | 113 | 176 | 139,2 | 435 |

Так как некоторые процессы можно проводить параллельно, общая продолжительность работ уменьшается. По графику видно, что общее время проектирования ТП=130 дней.

Капитальные затраты на этапе проектирования Кп рассчитываются по формуле:

KП = ZП + MП + НП,

где ZП - заработная плата проектировщика задачи на всем этапе проектирования;

MП - затраты на использование ЭВМ на этапе проектирования;

НП - накладные расходы на этапе проектирования.

Одним из основных видов затрат на этапе проектирования является заработная плата проектировщика, которая рассчитывается по формуле:



где Zд - дневная заработная плата разработчика задачи на этапе проектирования;

Ас - процент отчислений на социальное страхование (26%);

Ап - процент премий.

Средняя дневная плата рассчитывается по формуле:

Zд= ОК / Др,

где: ОК – оклад разработчика (10 000 руб.);

Др – среднее число рабочих дней (21 дней);

Получим,

Zд== 10 000 / 21 = 476 руб.

Отсюда,

ZП = 476\*130\* (1+0,07)⋅(1+0,26) = 83 426 руб.

Стоимость одного часа машинного времени примем 15 руб, тогда затраты на использовании ЭВМ равны:

МП = С \* 435 = 15 \* 435 = 6525 руб.

Накладные расходы составляют 80% от заработной платы персонала, занятого эксплуатацией программы, и вычисляются по формуле:

НП= (ZП \* 80) / 100

То есть НП=(83 426 \*80)/100=66 740 руб.

Таким образом, капитальные затраты на этапе проектирования продукта составят:

КП= 83 426 + 6525 + 66 740 =156 691 руб.

## 2.5 Техническое задание на разработку ИС по ГОСТ 19.201-78

### 2.5.1 Введение

#### 2.5.1.1 Наименование программы

Наименование программы – веб-приложение по бронированию билетов МБУК «КДЦ кинотеатр «Планета».

#### 2.5.1.2 Краткая характеристика области применения

Веб-приложение кинотеатра «Планета» предназначен для онлайн бронирования и заказа билетов на кинофильм.

Клиент будет заполнять выбирать кинофильм, на который желает пойти, выбирает место и ряд, и переходит на форму заказа билета. Форма будет отправлена в БД, а оттуда на страницу администратора. Связь между базой данных и администратором будут выполнять специальные для этого страницы.

#### 2.5.1.3 Основание для разработки

Основанием для разработки является прохождение преддипломной практики. Заявление на прохождение производственной практики утверждено и.о. директора МБУК «Культурно-досуговый центр» Костиным Сергеем Викторовичем, именуемым в дальнейшем руководителем производственной практики на предприятии, и Лозовым Ильей Ивановичем (студент), именуемым в дальнейшем исполнителем.

#### 2.5.1.4 Назначение для разработки

Сайт будет использоваться двумя группами пользователей: администратором и пользователь.

Функциональное значение:

Для пользователей системы веб-приложение предоставляет возможность ознакомиться с фильмами, розыгрышами, и будущими новинками кинопроката, которые предоставляет кинотеатр, бронировать билеты.

Для администратора веб-приложение предоставляет возможность смотреть забронированные и купленные места на сеанс и при необходимости редактировать заявки.

Эксплуатационное назначение:

Веб-приложение предоставляет информацию о идущих в прокате фильмов и возможность забронировать билет на желаемый фильм. Сайт будет доступен к просмотру и бронированию билетов всем желающим без регистрации.

Страница администратора доступна только администратору.

### 2.5.2 Требования к программе или программному изделию

#### 2.5.2.1 Общие требования

Веб приложение должно быть разработано с использованием языка гипертекстовой разметки HTML, каскадных таблиц стилей CSS и языка программирования JavaScript, серверного языка программирования PHP, и его фреймворка Laravel, базы данных MySQL.

По окончании работ исполнитель обязан предоставить полностью функционирующий сайт, исходные графические материалы по дизайну, все необходимые данные для доступа к системе управления сайтом (аккаунты, пароли и т.п.).

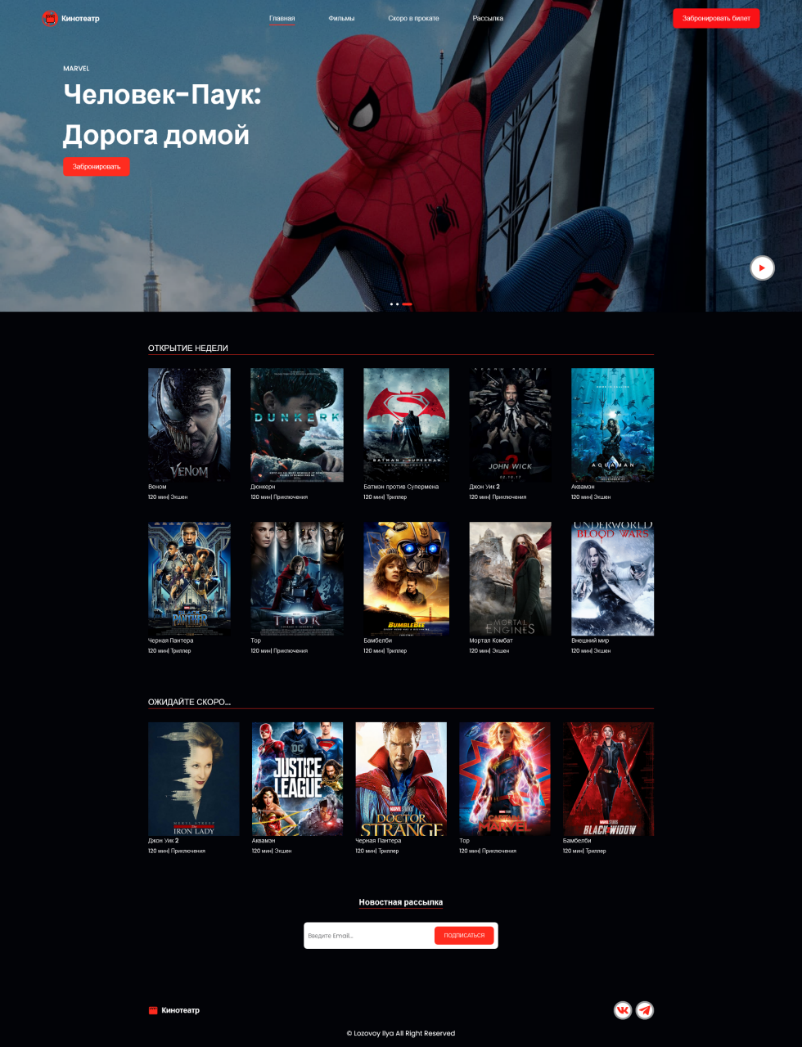
#### 2.5.2.2 Требования к дизайну сайта

Дизайн должен быть в черном и красном цветах, для большего привлечения аудитории. Дизайн сайта должен быть выполнен с использованием языка HTML и CSS и JS.

Сайт должен корректно отображаться во всех браузерах и на разных платформах.

Необходимо создать структуру (шаблон) главной страницы сайта, состоящую из следующих элементов:

* 1. Шапка
  2. Баннер-слайдер
  3. Открытие недели
  4. Скоро в прокате
  5. Новостная рассылка
  6. Подвал



**2.5.2.3 Требования к функциональности сайта**

Необходимо обеспечить возможность предоставлять пользователям информацию на русском языке.

Веб-приложение должно позволять пользователям:

* + - 1. осуществлять навигацию по сайту (переход между страницами);
      2. оформлять заказ билета;
      3. Просматривать видеоролики/трейлеры к фильмам

Система управления сайтом должна позволять:

1. управлять страницами сайта (добавлять, удалять, изменять их содержимое);

2. управлять элементами меню;

3.Бронировать и покупать билеты на показ

4. загружать на сайт графический материал (фото-видео изображения, различные файлы и т.п.).

#### 2.5.2.4 Требования к содержимому сайта

Необходимо создать следующие страницы сайта

1. Главная страница в формате LP
2. Страница «О нас»
3. Галерея фильмов
4. Галерея ожидаемых фильмов
5. Страница «Фильм»
6. Страница администратора

Макеты страниц (см. Приложение В)

Необходимо создать меню навигацию по сайту. Меню должно состоять из следующих элементов:

1. Главная
2. Фильмы
3. Скоро в прокате
4. О нас

### 2.5.3 Технические требования к информационному продукту

#### 2.5.3.1 Требования к организации входных и выходных данных

Все данные о фильмах, местах и способах оплаты хранятся в базе данных. СУБД обеспечивает разграничение прав доступа к данным – дает клиенту права на просмотр сайта, оформление заказа, заполнения формы, то есть полноценную работоспособность сайта со стороны клиента, а администратору – с помощью страницы администратора ввод и изменения данных сайта

#### 2.5.3.2 Требования к временным характеристикам

После изменения данных, находящихся в базе данных, новая информация отображается на сайте со стороны клиента (пользователя интернета) не позднее чем через 7 секунд.

#### 2.5.3.3 Требования к надёжности

Вероятность безотказной работы сайта должна составлять не менее 99.99% при условии исправности сети (со стороны клиента и также со стороны администратора).

#### 2.5.3.4 Требования к обеспечению надежного (устойчивого) функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:

* организацией бесперебойного питания технических средств;
* использованием лицензионного программного обеспечения;
* регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
* регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов.

#### 2.5.3.4 Время восстановления после отказа

Время восстановления после отказа, вызванная сбоем электропитания технических средств, не фатальным сбоем операционной системы, не должно превышать 15 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, не фатальным сбоем операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

#### 2.5.3.5 Условия эксплуатации

Сайт запускается на любом электроном носителе, позволяющий выход в интернет. Сам сайт и все данные выгружаются на хостинг. Благодаря этому имеется рабочая связь между клиентом и администратором все рабочее время.

Сайт должен открываться при клике на ссылку, во весь экран, также должна быть возможность закрыть сайт, свернуть или запустить любое стороннее программное обеспечение.

#### 2.5.3.6 Климатические условия эксплуатации

Специальные условия не требуются.

#### 2.5.3.7 Требования к видам обслуживания

Сайт не требует проведения каких-либо видов обслуживания.

#### 2.5.3.8 Требования к численности и квалификации персонала

В процессе эксплуатации с сайтом работают администратор и клиент (пользователь интернета)

Пользователь сайта (администратор) должен обладать практическими навыками работы графическим пользовательским интерфейсом.

Специальные требования для клиента (пользователя интернета) не требуются.

#### 2.5.3.9 Требования к составу и параметрам технических средств

Состав технических средств:

* Компьютер менеджера, включающий в себя:
  + Процессор Pentium – 1.6Hz;
  + Оперативную память не менее 1 Гб;
  + Видеокарту, монитор, мышь, клавиатура.
* Компьютер посетителя, включающий в себя:
  + Процессор Pentium – 1.6Hz;
  + Оперативную память не менее 1 Гб;
  + Видеокарту, монитор, мышь.

#### 2.5.3.10 Требования к информационной и программной совместимости

К программному изделию имеют доступ все пользователи интернета. Для безопасности программного изделия сайт будет защищен с помощью пароля в личный кабинет.

#### 2.5.3.11 Требование к маркировке и упаковке

Программное изделие хранится на хостинге и открывается на всех браузерах. Специальные требования к маркировке не требуется.

#### 2.5.3.12 Требования к транспортированию и хранению

Специальных требований не предъявляется.

#### 2.5.3.13 Специальные требования

Программное изделие должно обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса.

#### 2.5.3.14 Требования к программной документации

Предварительный состав программной документации:

* Техническое задание;
* Программа и методика испытаний;
* Руководство системного программиста;
* Руководство оператора;
* Руководство программиста;
* Ведомость эксплуатационных документов;
* Формуляр.

### 2.5.4 Стадии и этапы разработки

Разработка должна быть проведена в четыре стадии:

1. Анализ предприятия;
2. Техническое задание;
3. Разработка информационной системы;
4. Внедрение.

На стадии «Техническое задание» должно быть выполнено документация с требованиями заказчика, необходимыми непосредственно для разработки информационной системы. Данная документация не позволит выйти за рамки дозволенного.

На стадии «Разработка информационной системы» будет выполнена работа по программированию системы и ее отладки.

На стадии «Внедрение» будет выполнена работа по подготовке и передаче информационной системы.

### 2.5.5 Порядок контроля и приемки

Приемосдаточные испытания информационной системы должны проводиться непосредственно студентом и согласованно руководителем производственной практики на предприятии.

2.6 Оптимизация работы онлайн-сервиса

Техническая оптимизация сайта - техническая оптимизация представляет собой комплекс работ, которые позволяют улучшить взаимодействие сайта с роботами и алгоритмами поисковых систем. Игнорирование такого рода ошибок может значительно снизить эффективность от продвижения ресурса в целом.

Ошибки могут возникать по ряду причина, например, наиболее встречающиеся: особенности работы CMS, недостаточная компетенция специалистов, которые занимались разработкой сайта.

Начнём с проверки наличия и корректности заполнения файла ***[robots.txt](https://www.directline.pro/blog/sozdanie-robots/" \t "https://www.directline.pro/blog/tekhnicheskaya-optimizatsiya-sayta/_blank)***. Если вкратце, то это технический файл, в котором перечисляются директории сайта, доступные или закрытые для индексации поисковыми роботами. К слову, при посещении ресурса краулером, данный файл проверяется им в первую очередь.

Далее используем файл ***[sitemap.xml](https://www.directline.pro/blog/karta-sayta/" \t "https://www.directline.pro/blog/tekhnicheskaya-optimizatsiya-sayta/_blank)***, который содержит в себе структуру сайта в виде URL-адресов страниц, его наличие является обязательным для больших проектов. В идеале реализовать карту сайта таким образом, чтобы она обновлялась автоматически средствами.

Требования к файлу карты сайта:

* Содержит только актуальные URL-адреса.
* В перечень документов не входят неканонические страницы.
* Корректно указан протокол соединения (например, https).
* Код ответа всех документов 200.
* Исключены страницы, которые закрыты от индексации (в ***robots.txt*** или иным способом).
* Карта сайта валидна, проверить можно в [Яндекс.Вебмастере](https://webmaster.yandex.ru/tools/sitemap/" \t "https://www.directline.pro/blog/tekhnicheskaya-optimizatsiya-sayta/_blank).

Проверяем наличие лишних цепочек редиректов, что приводят к излишнему расходу краулингового бюджета.

Редиректы (301 – постоянный, 302 – временный) показывают краулеру путь, по которому перемещена текущая страница.

Краулинговый бюджет – некий лимит количества страниц, который поисковый робот проиндексирует за определенный временной период. При его нецелесообразном расходовании могут возникнуть проблемы с индексацией целевых страниц.

Для проверки редиректов на сайте булет использован специализированный софт - Screaming Frog SEO Spider.

После проверки редиректов проверяем наличие «битых» ссылок.

Битой считается такая ссылка, которая ведет на несуществующую страницу. Переходя по ней, система уведомляет об ошибке.

Битые ссылки могут появляться в процессе редактирования сайта: удаления/переезда страниц, опечатка при создании Url-адреса и т.д. Чем сложнее сайт, тем, как правило, чаще появляются подобные ссылки.

Наличие битых ссылок усложняет процесс продвижения сайта тем, что при их большом количестве алгоритмы могут посчитать ресурс низкокачественным. К тому же, подобное явление ухудшает пользовательский опыт, негативно влияя на поведенческие факторы (например, растет процент отказов), которые напрямую влияют на ранжирование сайта.

Проверить наличие битых ссылок можно в панели Яндекс.Вебмастер: «Индексирование» – «Статистика обхода». Также можно просканировать сайт программой Screaming Frog SEO Spider.

Нельзя забывать и про скорость загрузки сайта, которая может напрямую влиять на качество индексации краулерами, ранжирование в органической выдаче и на количество конверсий. Если говорить о конкретных цифрах, то в идеале страница должна загружаться не более 2-4 секунд. Все значения, что выше, могут вести к появлению следующих проблем:

Посетители покидают сайт, не дождавшись его загрузки.

Это ведет не только к снижению числа конверсий, но и к росту процента отказов.

Процент отказа является одной из составляющих поведенческих факторов, которые оказывают влияние на ранжирование сайта и его страниц в поиске.

Для проверки скорости загрузки сайта можно воспользоваться бесплатным инструментом от Google. Кроме оценки, он покажет иерархию загрузки элементов страницы, с помощью чего можно определить наиболее тяжеловесные объекты и в дальнейшем оптимизировать их.

Одним из ключевых этапов создания и продвижения сайта является создание его структуры. Она определяет иерархию страниц ресурса, их принадлежность к категориям, каталогам или папкам.

Структура бывает:

* Двухуровневая. Встречается на простых (узконишевых) сайтах, ее суть заключается в том, что ресурс состоит из главной страницы и внутренних 2-го уровня.
* Древовидная. В этом случае присутствуют категории (подкатегории), которые объединяют страницы 3-го и последующих уровней.

Вот основные рекомендации от Яндекса касательно создания правильной структуры сайта:

* Каждая из страниц должна относиться к своей категории.
* Если позволяет структура, то не следует располагать страницы более чем на пятом уровне. В противном случае может увеличиться время индексации подобных документов.
* В идеале URL-адрес страницы должен отображать всю иерархию пути, а также отражать ее содержимое.
* Исключить ситуацию, при которой в URL-адресе содержится множество повторений.
* Все URL-адреса должны быть в единой кодировке.
* Применяйте хлебные крошки для того, чтобы пользователь мог легко перемещаться между уровнями вложенности.

Поговорим о том, что является частью структуры сайта, а именно о внутренних переклиновках, но в этом пункте речь пойдет о самих ссылках. Правильно выстроенная внутренняя перелинковка документов позволяет:

* Поднять авторитетность определенных страниц в «глазах» поисковиков. Чем больше ссылок ведет на страницу, тем более она значимая, причем это касается не только внешних, но и внутренних ссылок.
* Наиболее авторитетные страницы сканируются роботами в первую очередь. Поэтому грамотная перелинковка позволяет значительно ускорить попадание новых страниц в индекс.
* Анкоры во внутренних ссылках увеличивают релевантность страницы относительно ключевых фраз, по которым она продвигается в поиске.
* Если ресурс имеет сложную структуру, то перелинковка улучшает юзабилити и положительно сказывается на поведенческих факторах.

Чтобы определить, какие страницы имеют наибольший авторитет в текущей структуре сайта, его можно просканировать в Screaming Frog SEO Spider.

Так как сайт имеет большое количество страниц и на нем ведутся постоянные работы по наполнению, часто возникают ситуации, при которых появляются битые ссылки. Перейдя по такому адресу, в ряде случаев пользователи покидают сайт. Чтобы этого избежать, помимо постоянно мониторинга и устранения битых ссылок, можно грамотно оформить страницу 404-ошибки:

* Выразите сожаление о том, что страница не найдена, при этом изъявив желание помочь.
* Разместите на ней ссылки на другие страницы сайта, которые пользуются наибольшим спросом. Это могут быть акции, скидки, главная, форма поиска и прочее.
* На случай, если пользователь пришел на данную страницу путем ручного ввода ее адреса, предложите перепроверить корректность указанной ссылки.
* Если уместно, используйте юмор и прочие креативные идеи.

Так как сайт написан на языке разметки HTML, который включает в себя ряд правил и спецификаций (HTML5). Соблюдение валидности верстки может давать следующие преимущества:

* Увеличивается скорость загрузки страниц.
* [Кроссбраузерность](https://www.directline.pro/blog/krossbrauzernost-sayta/" \t "https://www.directline.pro/blog/tekhnicheskaya-optimizatsiya-sayta/_blank). Сайт корректно (так, как задумали разработчики) отображается в популярных браузерах.
* Улучшается индексация. Ошибки в коде могут привести к тому, что робот некорректно проиндексирует страницу.
* Улучшение поведенческих факторов. Если страницы отображаются с ошибками расположения элементов, то это может привести к ухудшению пользовательского опыта.

На практике большинство современных браузеров способны корректно отображать сайты с невалидной версткой. Но в идеале, конечно же, проверять страницы ресурса в специальном [валидаторе](https://validator.w3.org/" \t "https://www.directline.pro/blog/tekhnicheskaya-optimizatsiya-sayta/_blank).

Учитывая тот факт, что мобильный трафик во многих сферах начинает обгонять десктопный, поисковики обращают внимание на адаптивность сайта для всех типов устройств. Поэтому, чтобы участвовать в мобильной выдаче ресурс должен корректно отображаться на всех разрешениях экранов. Интересно то, что Яндекс имеет общую поисковую выдачу (для мобильных и десктопных устройств), а Google разделяет ее. Т.е., если у сайта нет мобильной версии, то в Google это повлияет лишь на ранжирование в соответствующей выдаче, а у Яндекса в целом пессимизирует ресурс.

Чтобы проверить адаптивность страниц, рекомендуется использовать бесплатный инструмент от Google.

Затронем такую обширную, но безумно важную тему, как оптимизация контента, но выделим основные её пункты, которые относятся к технической оптимизации:

* Структурирование текста: использование заголовков H1-H6, списков, абзацев и всего прочего, что улучшает его восприятие пользователем.
* Не стоит забывать о мета-тегах title и description, которые могут попадать в сниппеты поисковых систем, а также влиять на ранжирование.

Не забывайте также использовать качественные изображения, релевантные опубликованному контенту на странице. Графика в первую очередь влияет на поведенческие и социальные факторы. Но не забывайте про оптимизацию веса картинок.

При разработке сайта закладывайте возможность прописывать уникальные alt и title у картинок.

Также по возможности стоит использовать уникальные изображения, поисковики подобное ценят.

# Заключение

В процессе рассмотрения предметной области были выявлены проблемы, связанные и предложены методы их решения, которые заключались в появлении нового формата заказов билетов. Такое решение проблем непосредственно повлияет на благосостояние компании.

Применение новых информационных технологий, таких как электронные информационные панели, электронные киоски и системы Интернет бронирования и покупки билетов могут бить реализованы благодаря современным Интернет/Интранет технологиям.

На основании разработанных моделей был предложен проект информационной системы, проведено технико-экономическое обоснование разработки и рассмотрены вопросы экологичности и безопасности жизнедеятельности.

## Социальная значимость разработки

Как было отмечено выше, современное общество нуждается в культурном отдыхе, который в частности обеспечивают и кинотеатры. В каждом городе уже имеется множество данных заведений. Не секрет, что каждое предприятие стремится минимизировать затраты и увеличить прибыль, повысить свою конкурентоспособность.

В данной работе были предложены методы и средства автоматизации работы кинотеатров, которые обеспечивают большую клиентоориентированность кинотеатра, а значит большую привлекательностью. Сейчас век компьютерных технологий. Технические средства контроля и управления на предприятии становятся все более доступными, поэтому для повышения конкурентоспособности и качества обслуживания клиентов, просто необходимо их использование.

Построенные функциональные схемы, сценарии и функционально ориентированные модели в дальнейшем подвергшиеся формализации с помощью математического аппарата теории систем массового обслуживания доказывают необходимость внедрения автоматизированных технологий.

Предложенный вариант организации автоматизированной системы продажи и бронирования билетов основанный на интернет/интранет технологиях является наиболее оптимальным, так как обладает способностью расширения функциональности путем использования электронных киосков и глобальной вычислительной сети Интернет.

# Приложение А. Диаграммы IDEF-0

Диаграммы IDEF-0 «AS-IS»







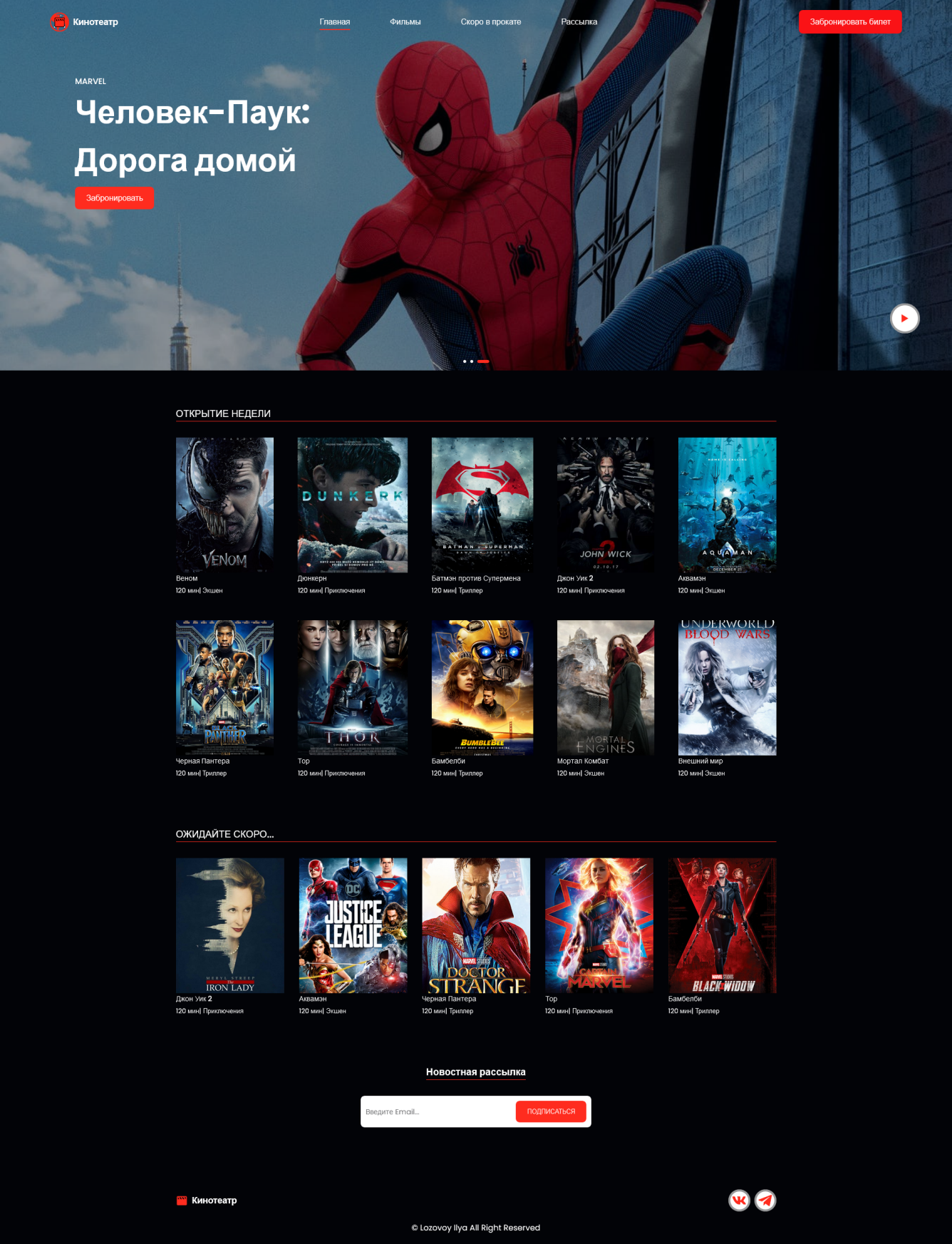
# Приложение Б. Диаграммы IDEF-0 «TO-BE»

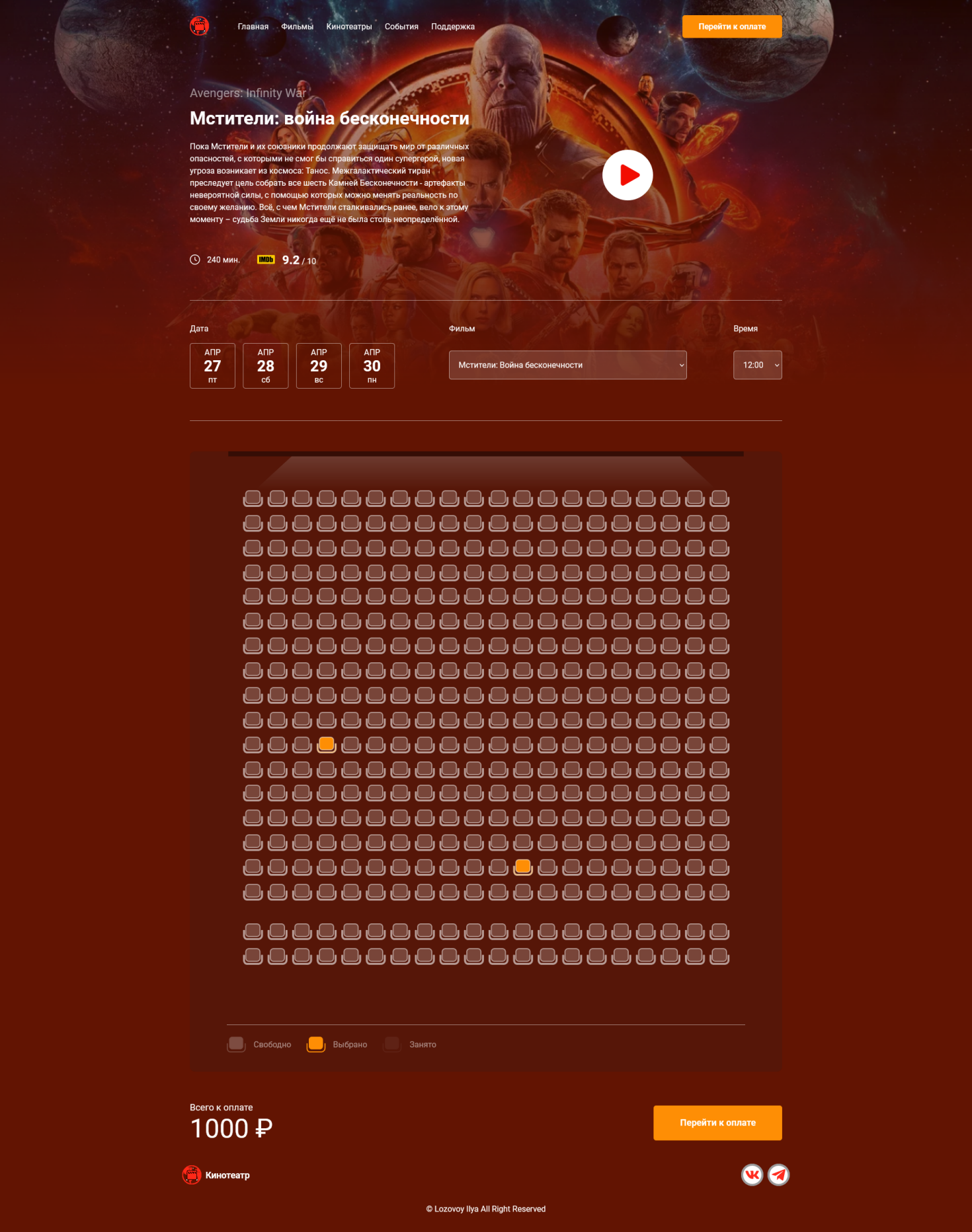


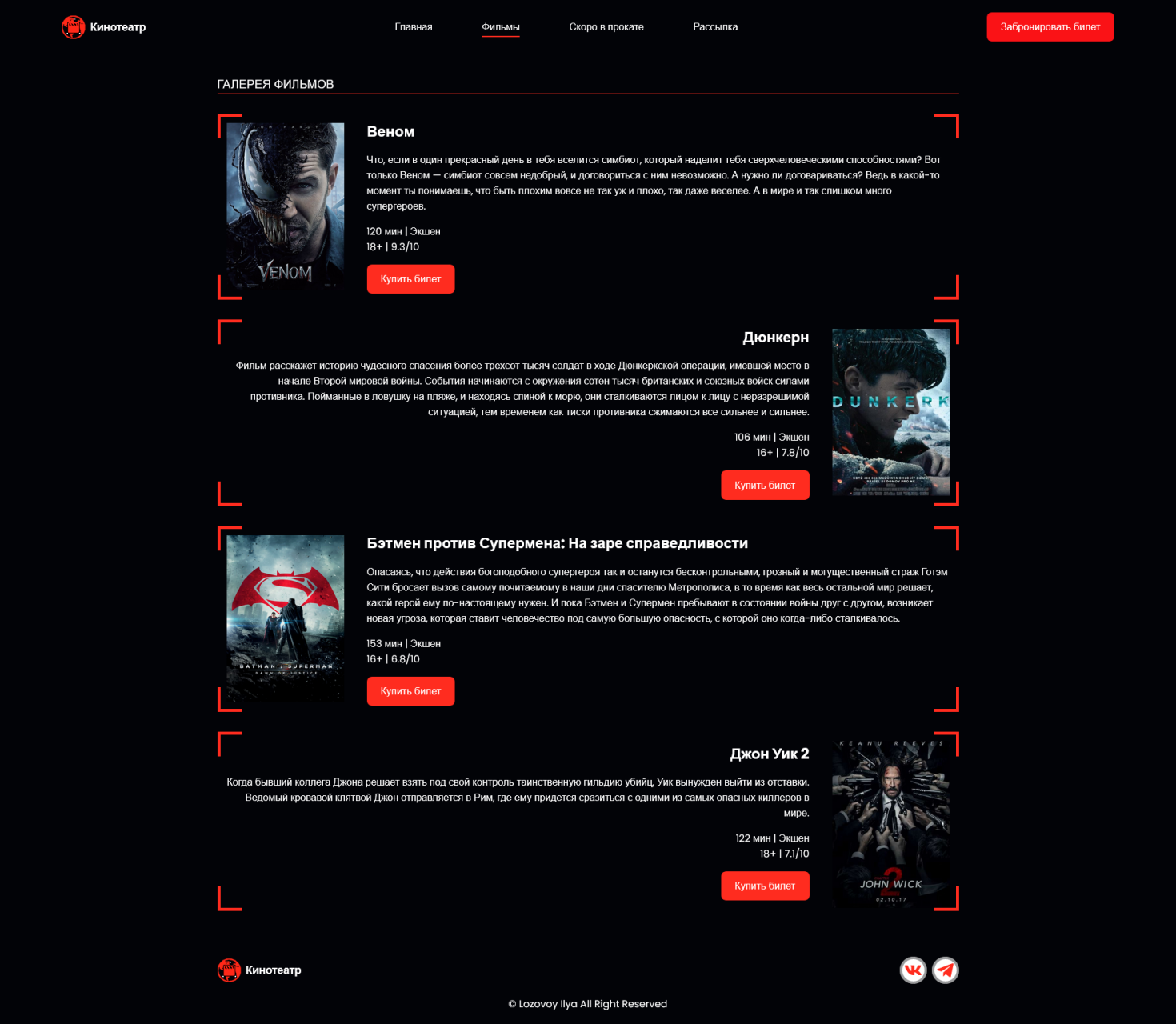


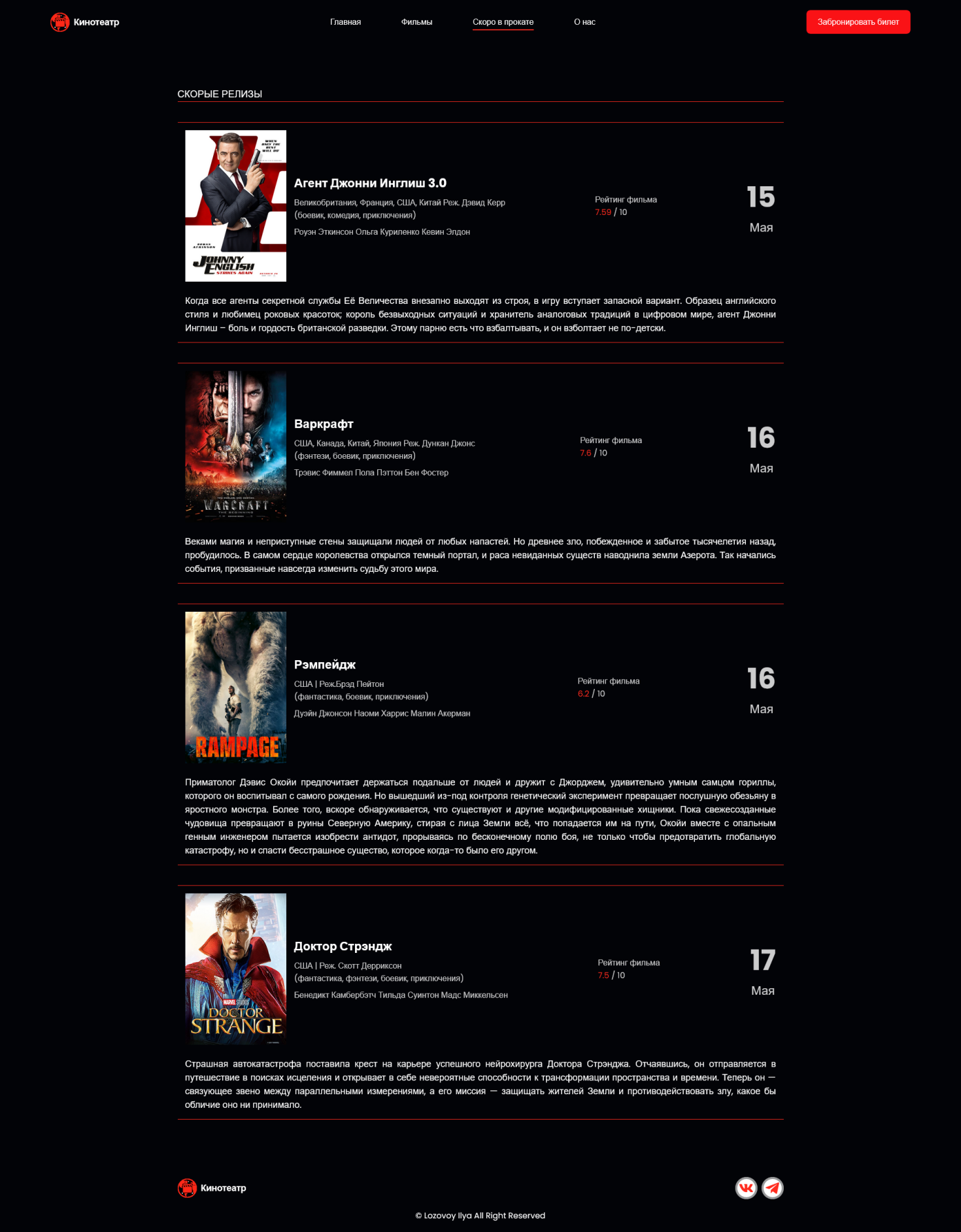


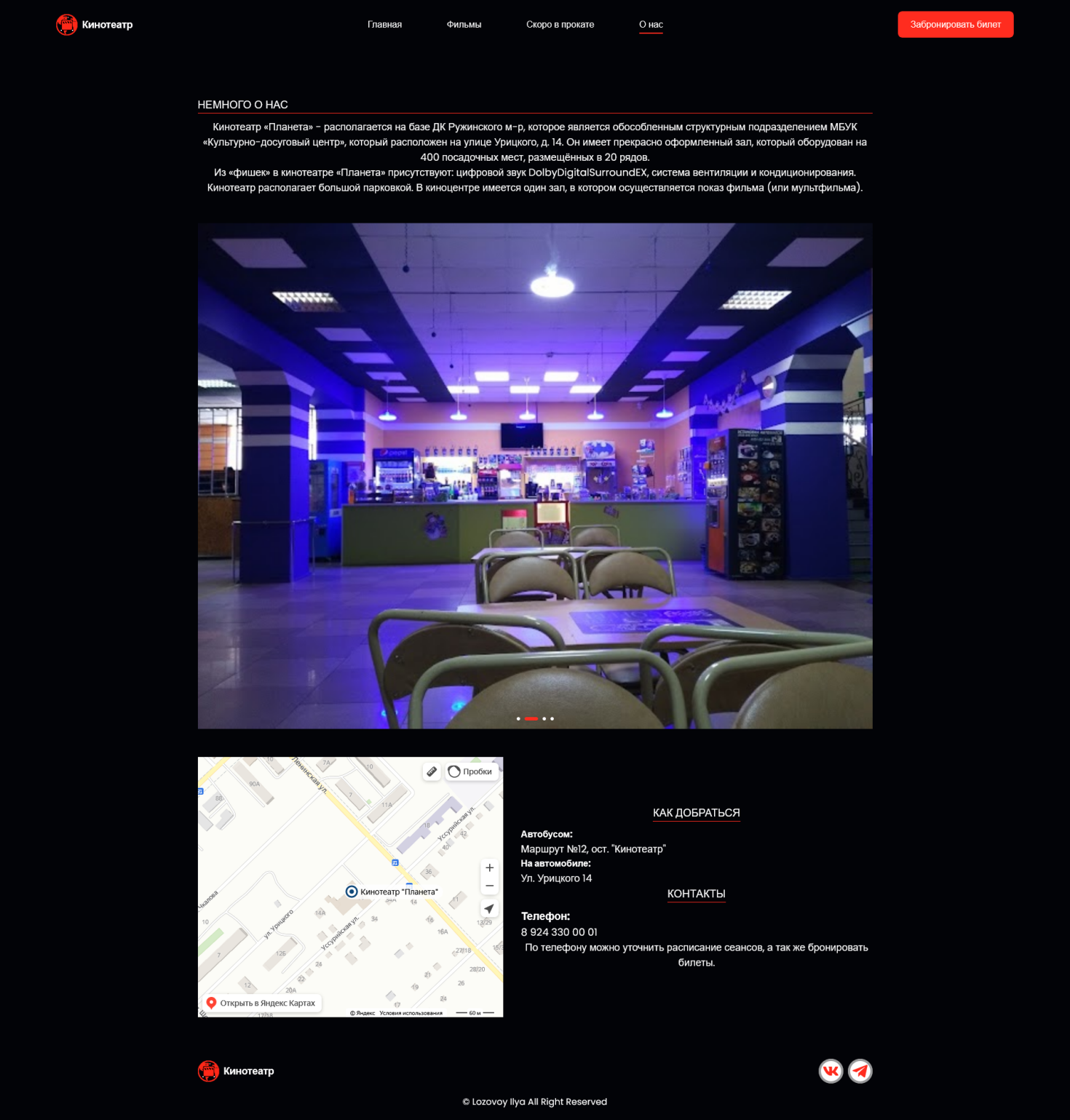
# Приложение В. Макеты страниц











[Сайт](https://inguntov.github.io/diplom/index.html)