МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет *компьютерных наук*

Кафедра *информационных систем*

***Система продажи концертных билетов* «YourTickets»**

*Курсовой проект*

*по дисциплине Технологии программирования*

09.03.02 *Информационные системы и технологии*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Анисимов А.В., 3 курс, д/о*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Рагимов Р.Т., 3 курс, д/о*

Обучающийся *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кравцов Н.Р., 3 курс, д/о*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Тарасов В. С.*

Воронеж 2021

Оглавление

[Введение 4](#_Toc87446518)

[1. Постановка задачи 5](#_Toc87446519)

[2. Анализ предметной области 6](#_Toc87446520)

[2.1. Цели создания приложения 6](#_Toc87446521)

[2.2. Сфера применения 6](#_Toc87446522)

[2.3. Сравнение аналогов. Анализ сильных и слабых сторон 7](#_Toc87446523)

[2.4. Перечень принятых определений и сокращений 10](#_Toc87446524)

[2.5. Требования к функциональности 11](#_Toc87446525)

[3. Диаграммы 12](#_Toc87446526)

[3.1. Диаграмма последовательностей 12](#_Toc87446527)

[3.2. Диаграмма коопераций 14](#_Toc87446528)

[3.3. Диаграмма состояний 15](#_Toc87446529)

[3.4. Диаграмма объектов 16](#_Toc87446530)

[3.5. Диаграмма классов 17](#_Toc87446531)

[3.6. Диаграмма активностей 19](#_Toc87446532)

[3.7. IDEF0 22](#_Toc87446533)

[3.8. Диаграмма прецедентов 23](#_Toc87446534)

[3.9. Диаграмма развертывания 24](#_Toc87446535)

[4. Анализ ключевых сценариев 26](#_Toc87446536)

[5. Тестирование 28](#_Toc87446537)

[5.1. Приёмочное тестирование 28](#_Toc87446538)

[5.2. Дымовое тестирование 28](#_Toc87446539)

[5.3. UI тесты 30](#_Toc87446540)

[5.4. Негативное тестирование 32](#_Toc87446541)

[5.5. Позитивное тестирование 32](#_Toc87446542)

[6. Реализация 33](#_Toc87446543)

[6.1. Анализ средств реализации 33](#_Toc87446544)

[6.2. Шапка сайта 34](#_Toc87446545)

[6.3. Главная страница 34](#_Toc87446546)

[6.4. Страница регистрации 35](#_Toc87446547)

[6.5. Страница авторизации 35](#_Toc87446548)

[Заключение 36](#_Toc87446549)

[Список источников 37](#_Toc87446550)

# Введение

В современном мире человек привык совершать многие действия онлайн. Покупать продукты, бронировать отели и многое другое уже можно совершать с помощью гаджетов. Тоже самое можно сказать и о покупке концертных билетов. Намного удобнее выбрать мероприятие которое тебе по душе не выходя из дома. По статистике – 60% клиентов хотят покупать билеты через интернет, а не через кассы.

Из-за этого возникает потребность в сайте, на котором можно выбрать мероприятие и купить на него билет. Приложение “Yourtickets” поможет пользователям найти необходимый им концерт и приобрести на него билет, тем самым значительно экономя им время, так как пользователю не придётся идти в кассу концертного зала, и стоять в очереди на покупку. Благодаря приложению все концерты от разных площадок будут собраны в одном месте.

Приложение должно обладать следующим функционалом:

- Авторизация пользователей.

- Регистрация пользователей

- Выбор концерта и покупка билетов.

# Постановка задачи

Задача данной курсовой работы реализовать приложение, которое соответствовало бы следующим требованиям:

-Поддержка всех актуальных на данный момент браузеров (Maxthon, Google Chrome, Opera, Mozilla Firefox, Microsoft Internet Explorer, Safari, YandexBrowser)

- Быстрая работа на различных браузерах

- Отсутствие ярких цветов

Основной функционал:

- Регистрация и авторизация

- Поиск концертов

- Просмотр информации о концертах

- Покупка и получение билетов.

Процесс создания должен быть разделен на несколько частей.

1. Создание визуальной составляющей сайта(Front-end)
2. Создание внутренней составляющей сайта(Back-end)
3. Создание и подключение Базы данных
4. Подключение всех вышеперечисленных частей и их корректная совместная работа

Завершенный проект представляет собой полностью функционирующее web-приложение.

# Анализ предметной области

## Цели создания приложения

Большинство людей ценят свое личное время и хотели бы уменьшить время на выполнение различных бытовых задач.

Приложение “Yourtickets” создается с целью сокращения времени пользователя, затрачиваемого им на выполнение такой частой задачи как поиск и покупка билетов на концерт.

## Сфера применения

Приложение ориентированно на пользователей с высокой степенью занятости, фанатов и просто людей которые желают развлечься.

Цель приложения сократить затраты времени на поиск билетов на концерт и предоставить удобный интерфейс взаимодействия между конечными пользователями и концертными площадками.

## Сравнение аналогов. Анализ сильных и слабых сторон

**1**. <www.redkassa.ru>

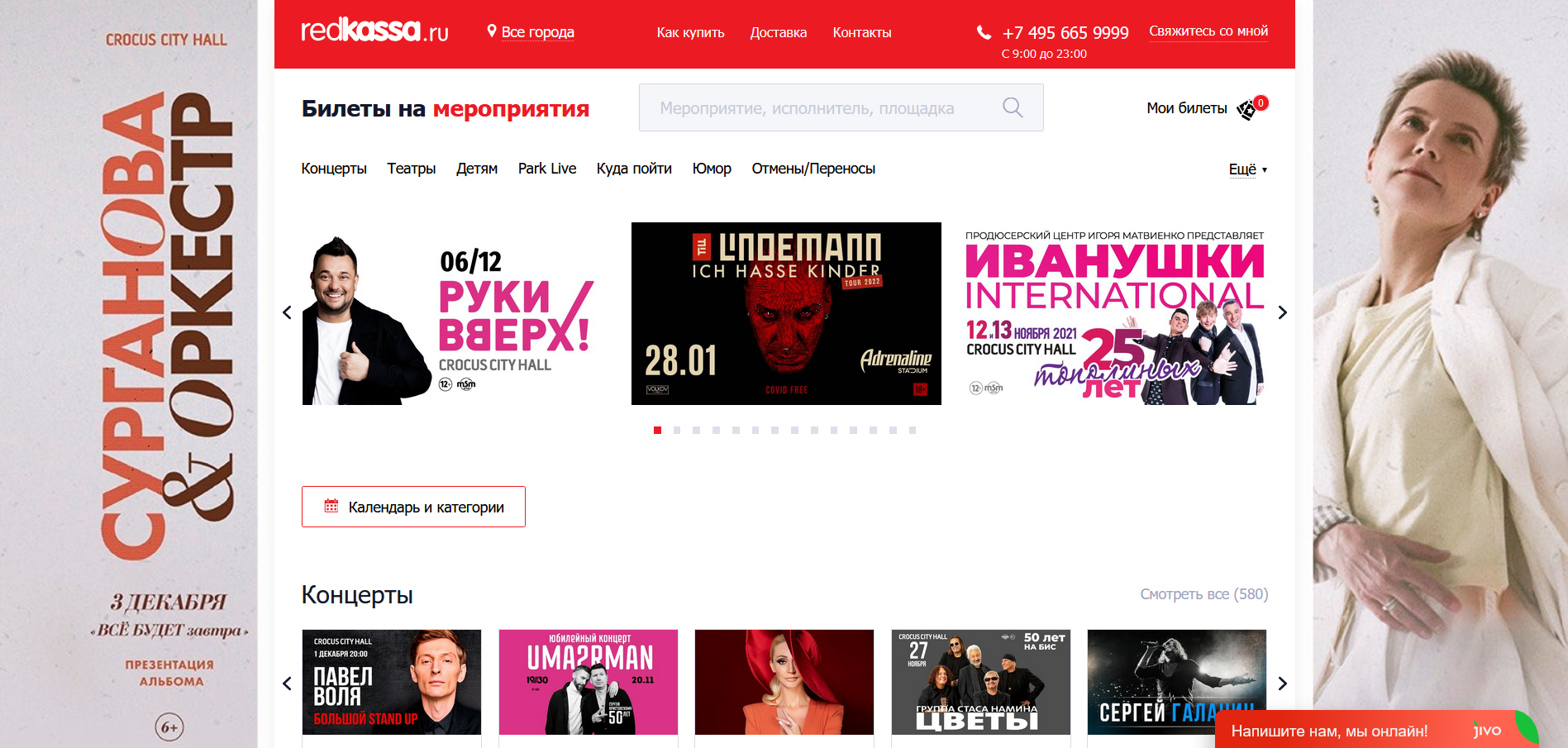


Рисунок 1 – www.redkassa.ru

Данный веб-сайт предлагает большое количество мероприятий таких как: концерты, театральные выступления, и тд .

**Сильные стороны.**

Есть много фильтров.

Позволяет увидеть свободные места и их расположение.

Есть подборки праздничных билетов

Позволяет выбрать разные условия доставки билетов

**Слабые стороны.**

Нет системы регистрации и авторизации.

**2**. <www.parter.ru>

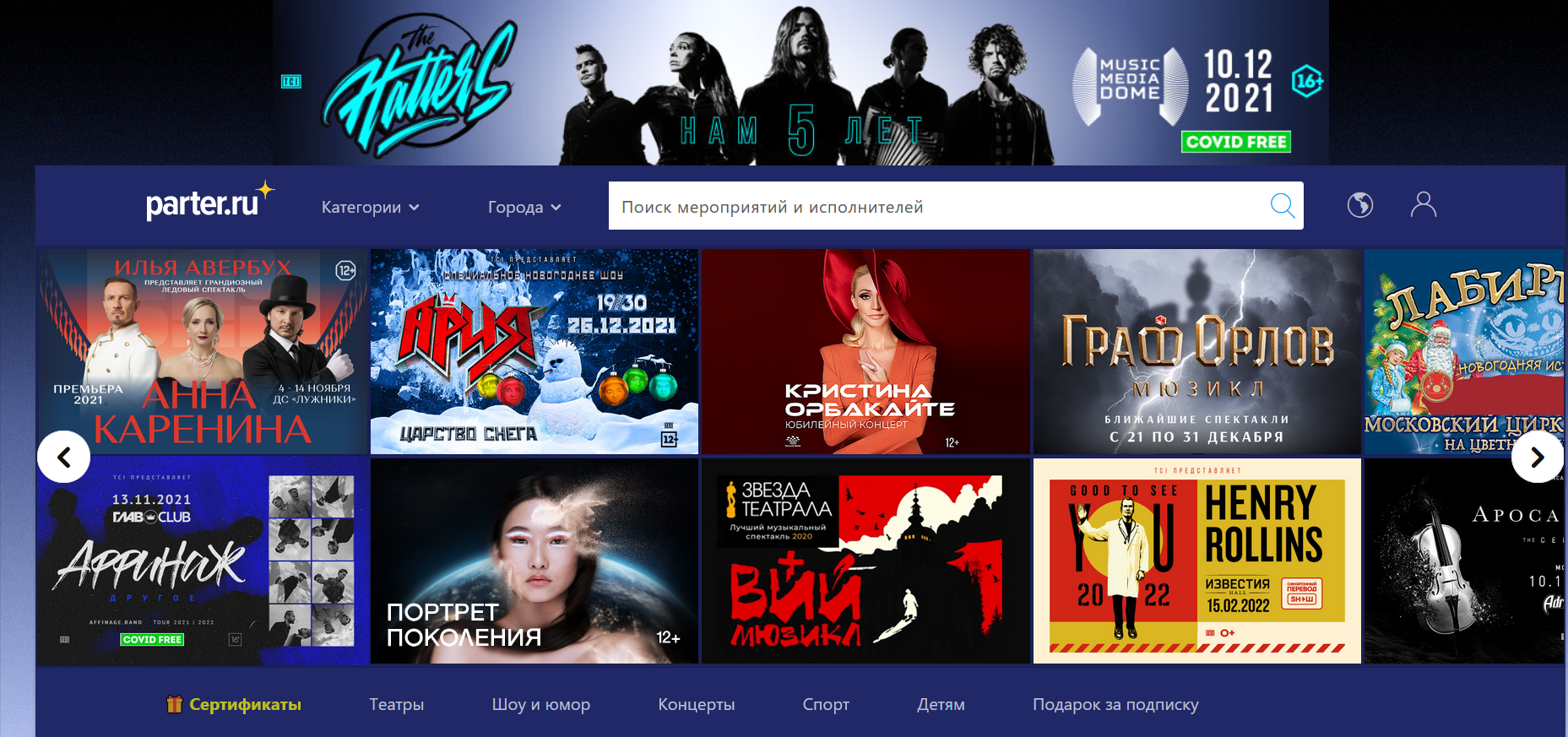


Рисунок 2 –www.parter.ru

Позволяет приобретать билеты в театр, на концерты, спортивные мероприятия.

**Сильные стороны.**

Предоставляет приобрести фанбилет с уникальным дизайном .

**Слабые стороны.**

Перегруженный интерфейс.

**3**. [www.kassir.ru](http://www.kassir.ru)

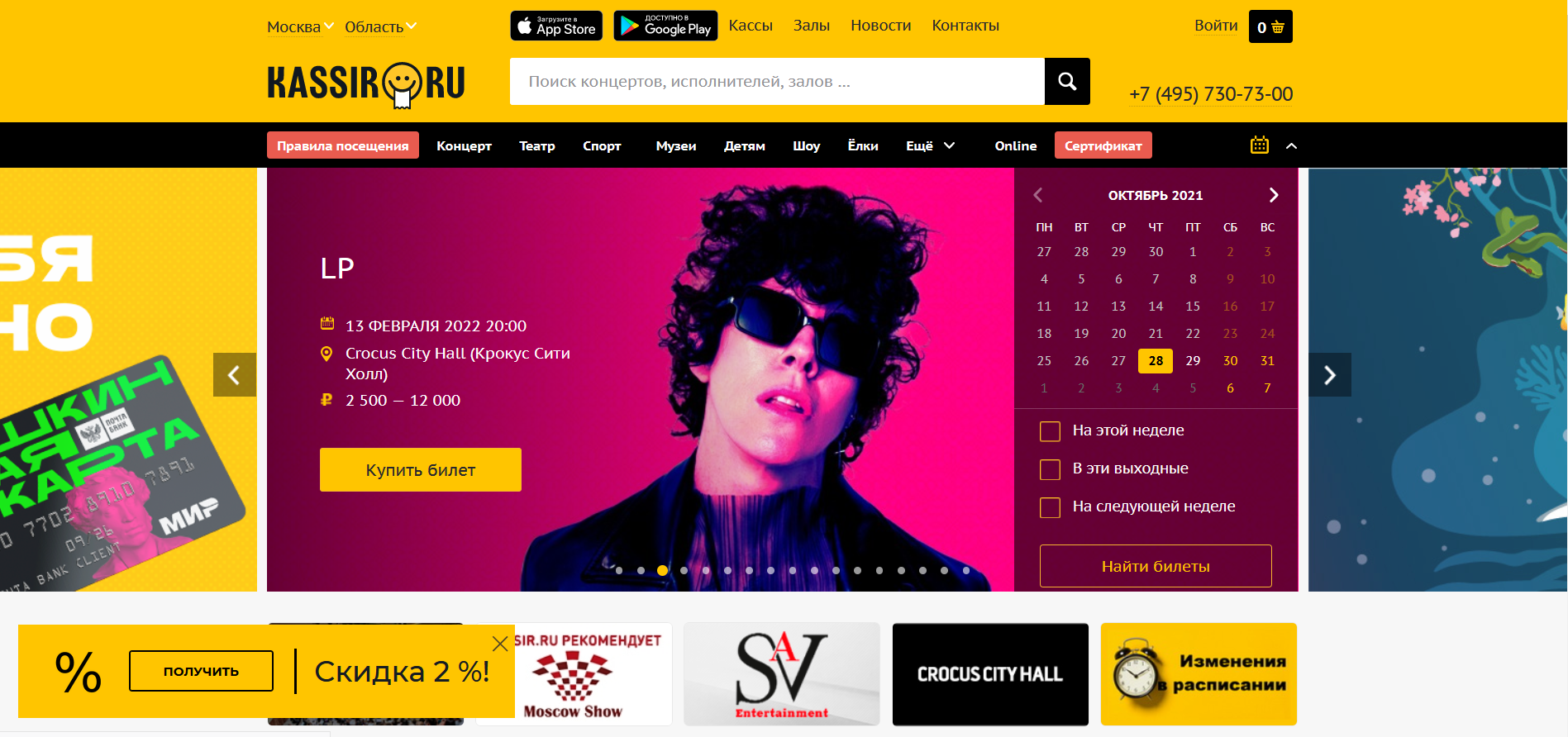


Рисунок 3 – www.kassir.ru

Сайт по продаже билетов по всей России.

**Сильные стороны.**

Можно найти анонс любого интересующего мероприятия, рассмотреть схему зала, приобрести электронный билет, ознакомится с программой и организаторами. Можно оформить рассрочку.

**Слабые стороны.**

Нельзя выбрать несколько фильтров для поиска.

## Перечень принятых определений и сокращений

|  |  |
| --- | --- |
| **Термин** | **Описание** |
| Сайт | часть проекта «YourTickets», являющийся web-приложением с использование базы данных MySQL. |
| HTML-страница (веб-страница, страница) | Основной носитель информации в World ide Web. Особым образом сформатированный файл (набор файлов), просматриваемый с помощью www-браузера как единое целое (без перехода по гиперссылкам) |
| Гиперссылка (ссылка, линк) | Активный элемент HTML-страницы, задаваемый специальным тегом. Выделенный фрагмент текста или изображения, позволяющий загрузить другую страницу или выполнить определенное действие |
| WWW-браузер (браузер) | Клиентская программа, поставляемая третьими сторонами и позволяющая просматривать содержимое HTML-страниц |
| Администратор (менеджер, редактор) сайта | Лицо, осуществляющее от имени Заказчика информационную поддержку сайта |
| Дизайн веб-сайта | Уникальные для конкретного веб-сайта структура, графическое оформление и способы представления информации |
| Веб-интерфейс | Совокупность экранов и элементов управления системы, позволяющих пользователю, осуществляющему доступ к системе через веб-браузер, осуществлять поддержку и управление системой. |
| Функционал | Конкретные действия, разрешенные системой, лицу или группе лиц, пользующимся системой. |
| Неавторизованный пользователь | Человек непрошедший процедуру авторизации в системе, а также пользующийся ограниченным пользовательским функционалом. |
| Авторизованный пользователь | Пользователь прошедший процедуру авторизациина сайте, имеющий право использовать предоставленный ему функционал сайта. |
| БД | База данных |

## Требования к функциональности

Приложение должно с точностью удовлетворять функциональным требованиям заявленной таблицы

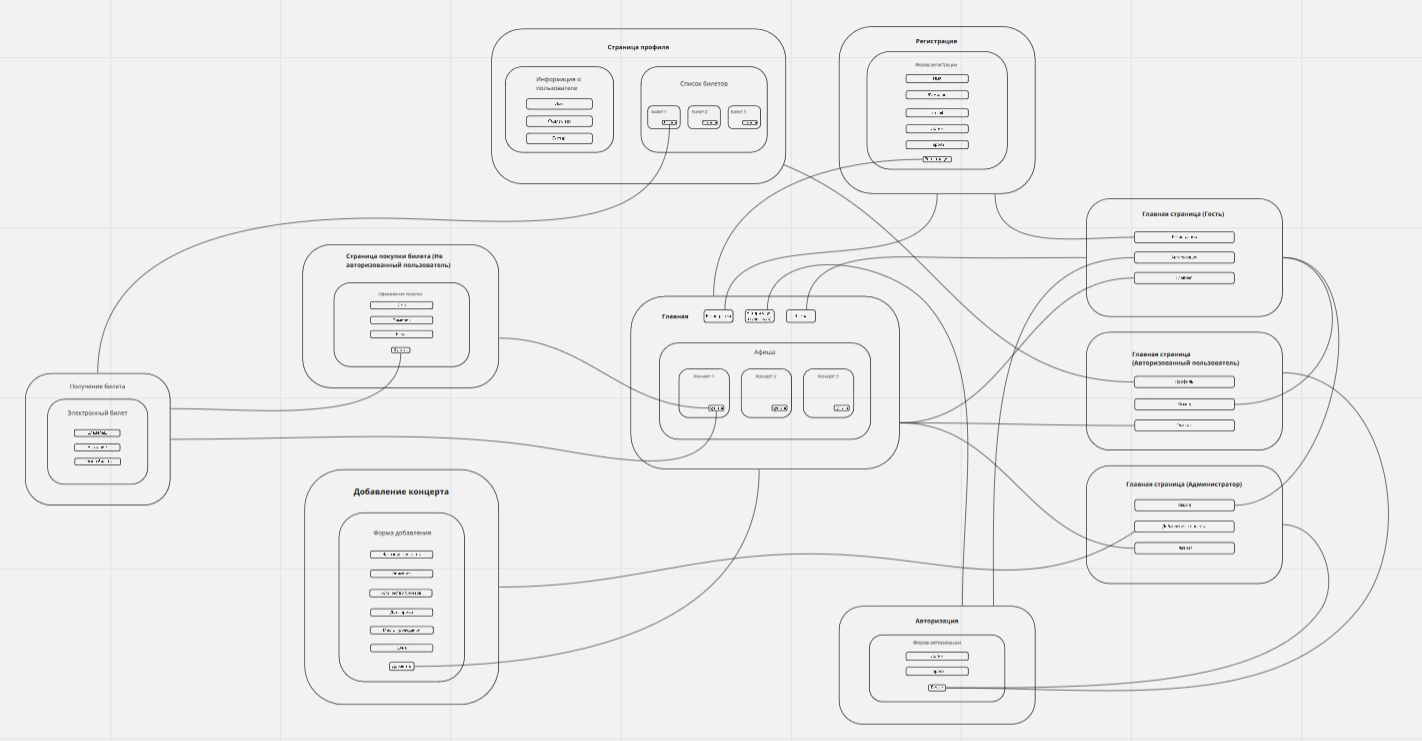


Рисунок 6 – Схема в miro

# Диаграммы

## Диаграмма последовательностей

Диаграмма последовательностей отображает взаимодействие объектов в динамике.

На первой диаграмме мы видим взаимодействие Незарегистрированного пользователя при Регистрации. Незарегистрированный пользователь передаёт данные приложению, после этого приложение делает запрос к Базе данных и получает от неё ответ, содержащий либо информацию о добавлении нового Пользователя, либо о невозможности данного действия в связи с существованием такого Пользователя в БД.При успешном сценарии происходит перенаправление на страницу регистрации с уведомлением о успешном её прохождении.

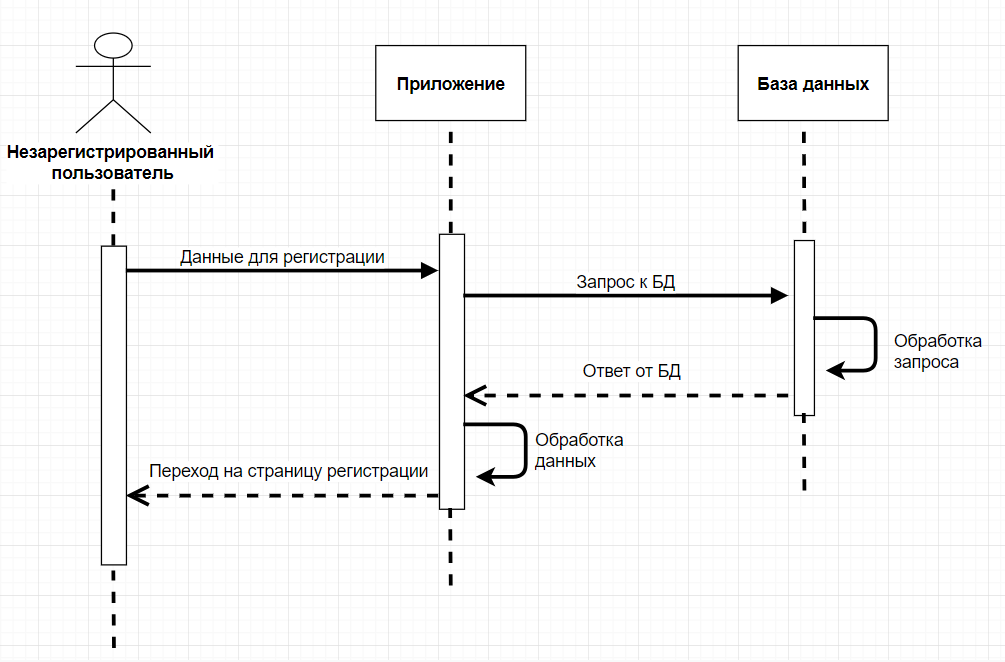


Рисунок 7 – Диаграмма последовательностей для регистрации

На второй диаграмме мы видим взаимодействие пользователя при Авторизации. Пользователь передаёт данные приложению, после этого приложение делает запрос к Базе данных и получает от неё ответ содержащий информацию о Авторизации, либо о невозможности данного действия в связи с не существованием данного пользователя в Базе данных

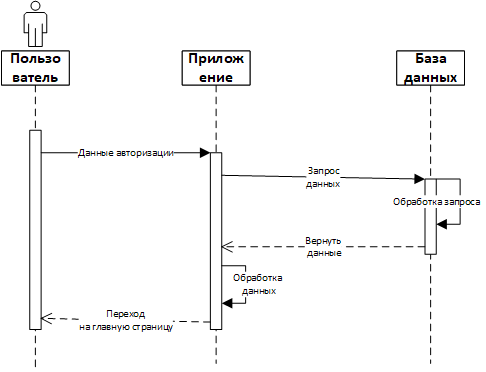


Рисунок 8 - Диаграмма последовательностей для авторизации

На третьей диаграмме показано взаимодействие при покупке билета. Пользователь отправляет запрос приложению, после чего происходит запрос к Базе данных. Приложение получает ответ от БД и отправляет пользователя на страницу с уведомлением о покупке билета. По нажатию на кнопку “Ссылка на электронный билет” отправляется запрос к приложению, и после обработки данных Пользователь получает билет.

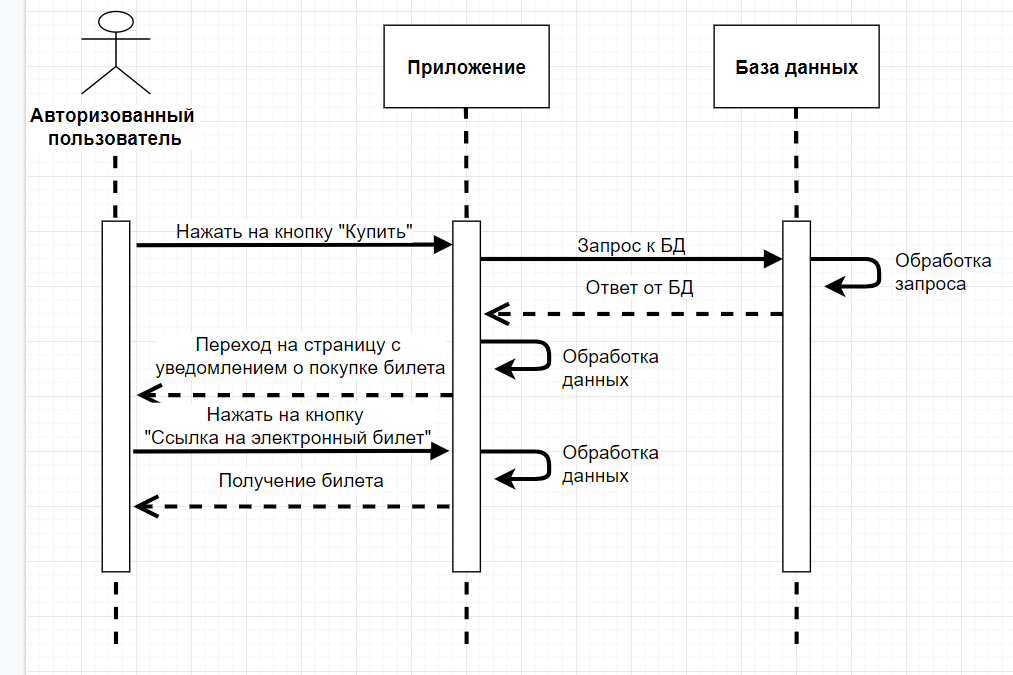


Рисунок 9 – Диаграмма последовательностей при покупке билета

## Диаграмма коопераций

С помощью диаграммы кооперации можно описать полный контекст взаимодействий как своеобразный временной "среза" совокупности объектов, взаимодействующих между собой для выполнения определенной задачи или бизнес-цели программной системы.

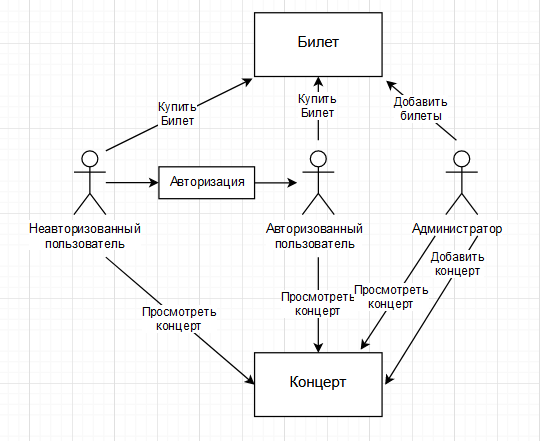


Рисунок 10 – Диаграмма кооперации

## Диаграмма состояний

Они определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий.

На рисунке 11 мы видим состояние запроса на покупку Билета. На первом этапе происходит покупка, и у пользователя отображается уведомление о покупке билета. После чего Пользователь получает Билет.

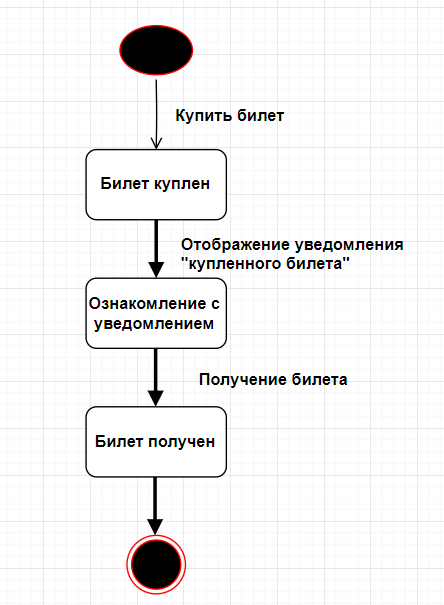


Рисунок 11 – Диаграмма состояния

## Диаграмма объектов

На диаграмме объектов отображаются экземпляры классов (объекты) системы с указанием текущих значений их атрибутов и связей между объектами.

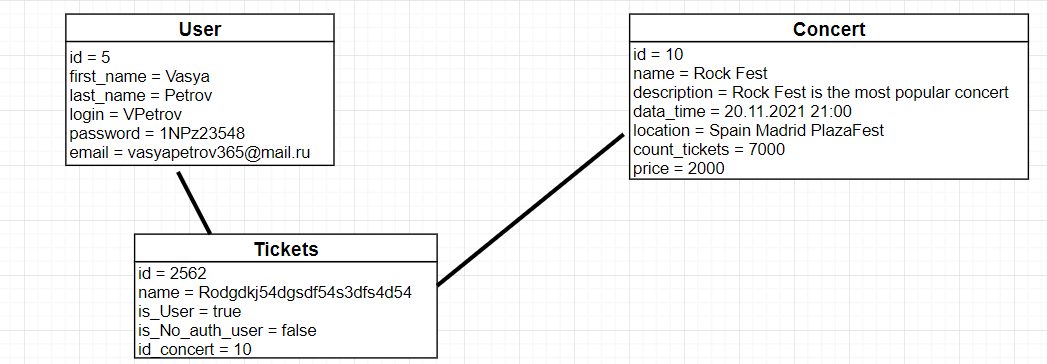


Рисунок 12 – Диаграмма объектов

## Диаграмма классов

Диаграмма классов – это структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

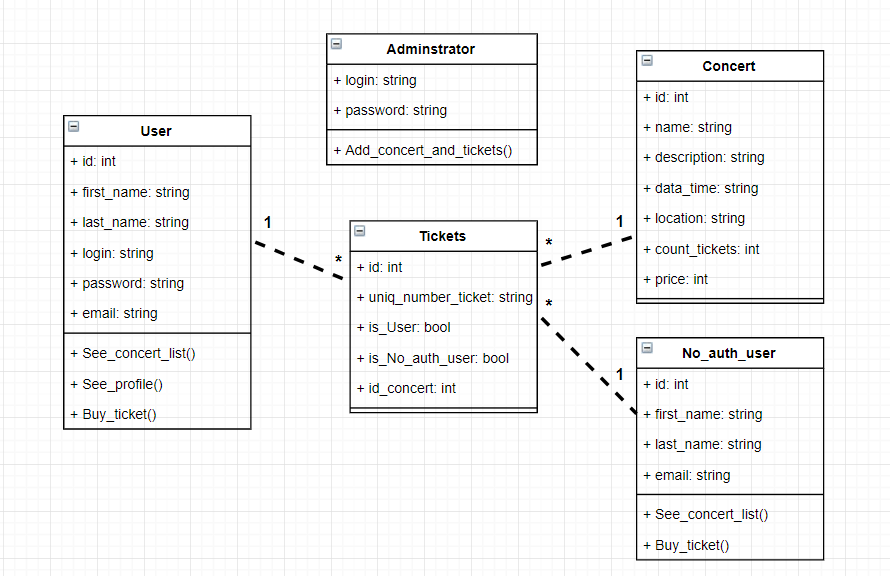


Рисунок 13 – Диаграмма классов

## Диаграмма активностей

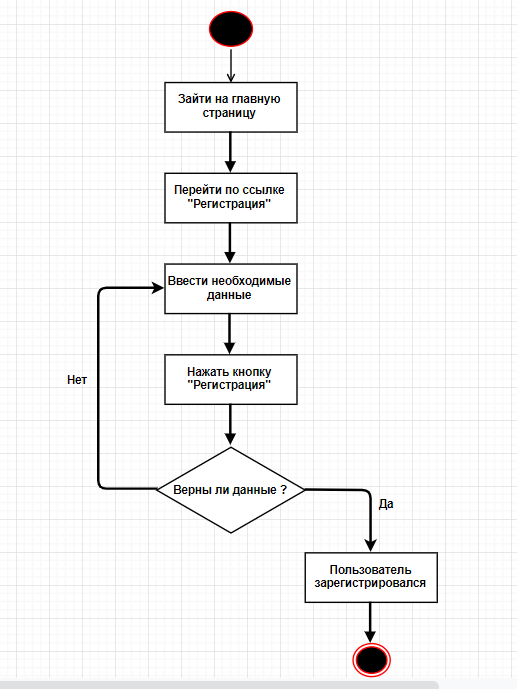


Рисунок 14 – Диаграмма активности для регистрации

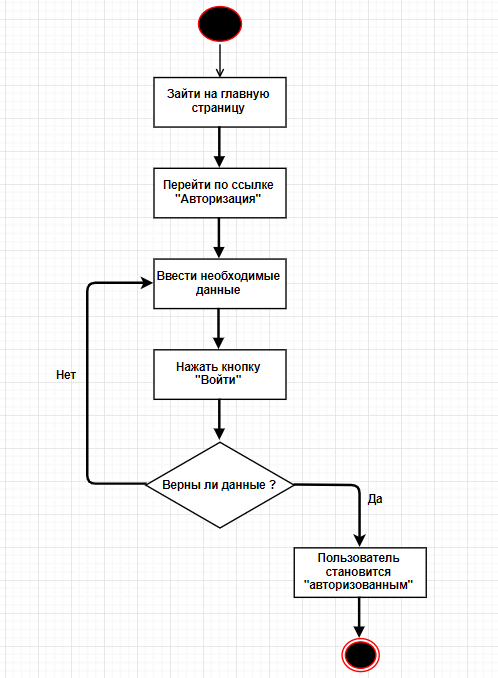


Рисунок 15 – Диаграмма активности для авторизации

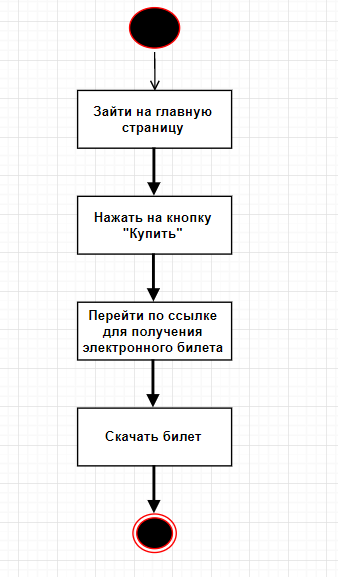


Рисунок 16 – Диаграмма активности для покупки билета

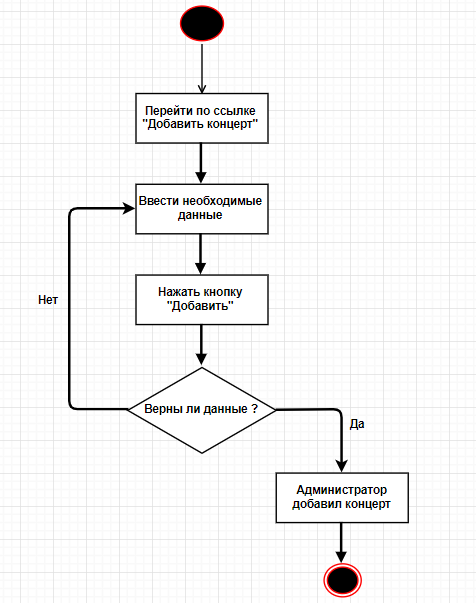


Рисунок 17 – Диаграмма активности для добавления концерта

## IDEF0

На данной диаграмме представлено видение добавления Концертов и покупки Билета.

В блок «Добавить концерт» на вход поступает информация о концерте, Управление осуществляет администратор, Механизм – Правила добавления концерта. На выходе получаем концертные данные.

После чего эти данные поступают в блок «Купить билет на концерт». Управление осуществляет Пользователь, Механизмы – Правила покупки билета. На выходе получаем концертный билет.

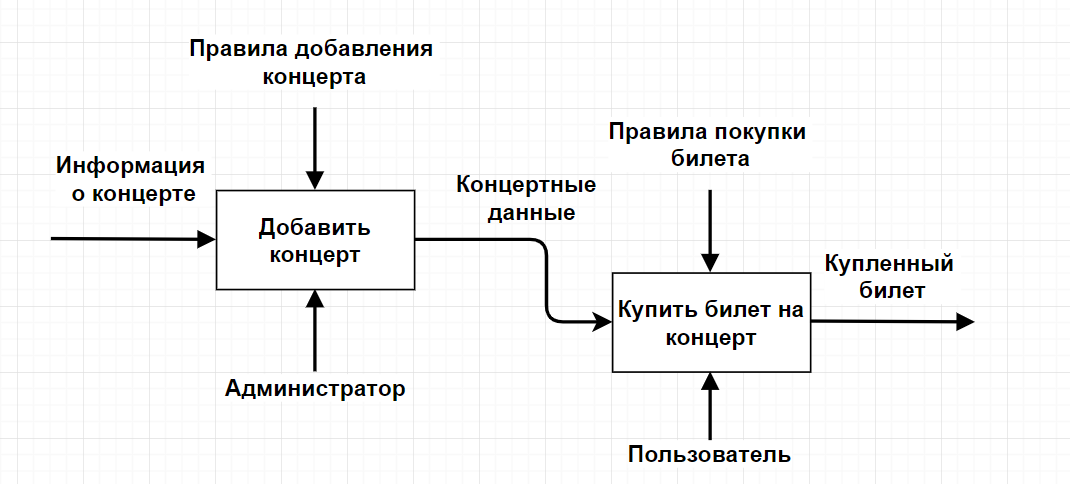


Рисунок 18 – IDEF0

## Диаграмма прецедентов

Как мы видим на диаграмме, в нашей Системе будет существовать 3 вида пользователей: Гость, Пользователь, Администратор, каждый из которых способен взаимодействовать с Билетами.

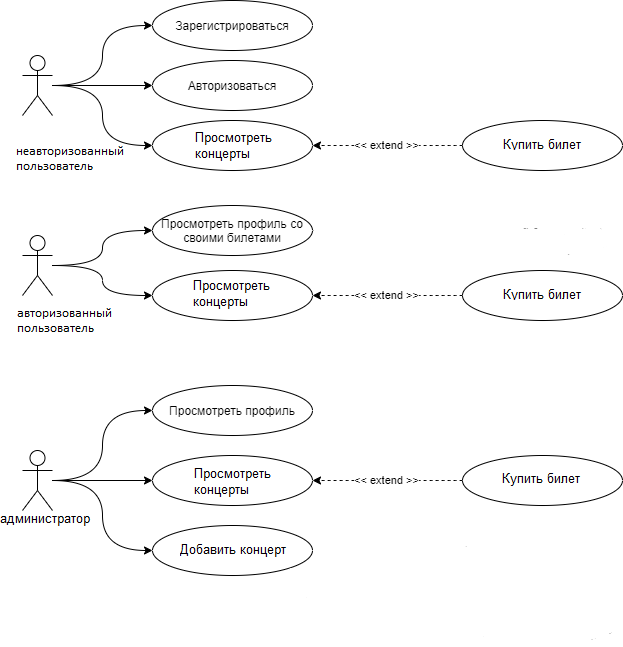


Рисунок 19 – Диаграмма прецедентов

## Диаграмма развертывания

Диаграмма развертывания предназначена для визуализации элементов и компонентов системы, существующих лишь на этапе ее исполнения, к которым относятся исполнимые файлы, динамические библиотеки, таблицы БД и т. д.

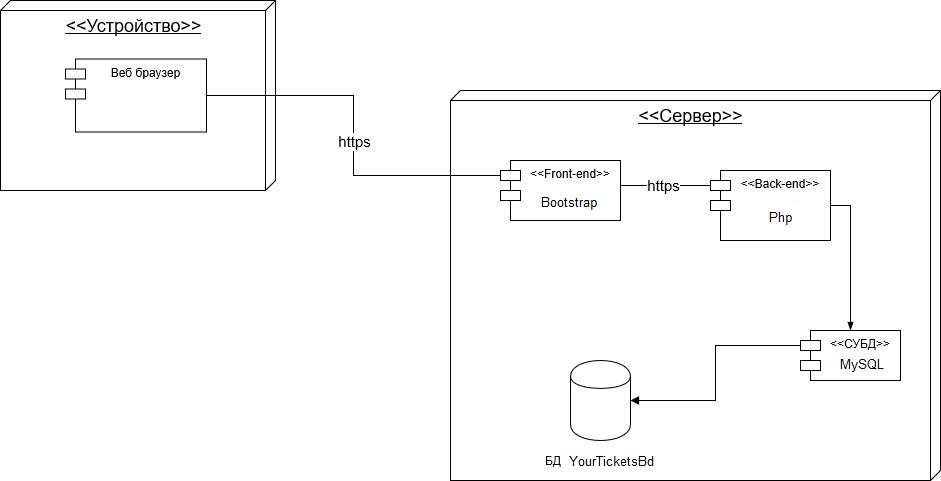


Рисунок 20 – Диаграмма развертывания

# Анализ ключевых сценариев

Чтобы выяснить, насколько приложение отвечает потребностям пользователей, к нему подключается система веб-аналитики. Получая данные о посетителях, мы можем анализировать их действия и на основе выводов из этих данных принимать решения о доработке системы для увеличения конверсии.

В системе аналитики будет вестись учет трех ключевых вариантов использования:

* Регистрация
* Авторизация
* Покупка билета

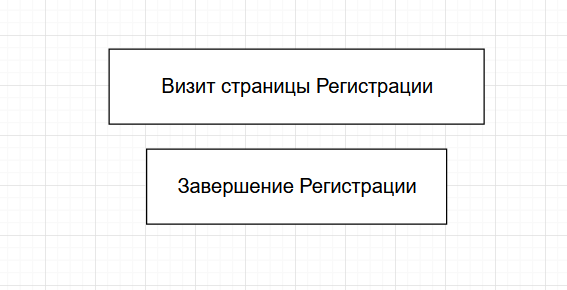


Рисунок 21 – Воронка «Регистрация»

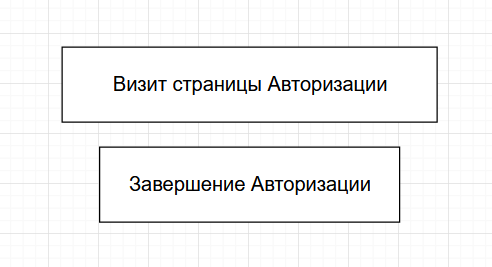


Рисунок 22 – Воронка «Авторизация»

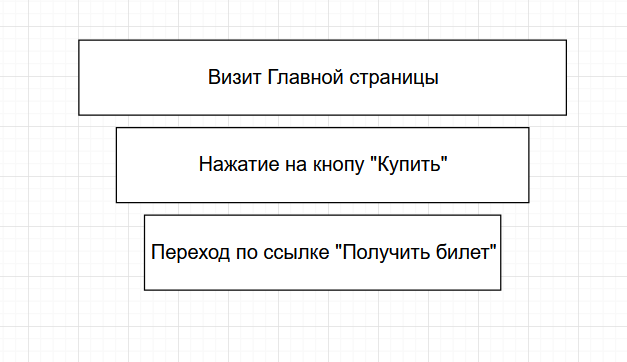


Рисунок 23 – Воронка «Покупка билета»

# Тестирование

После реализации всех задач был проведен запланированный набор тестов, а именно:

* Приемочное тестирование
* Дымовое тестирование
* UI тесты
* Позитивное тестирование
* Негативное тестирование

## Приёмочное тестирование

|  |  |
| --- | --- |
| Наименование | Статус |
| Проверка всех функций приложения | Выполнено |

Таблица 1- Приёмочное тестирование

## **Дымовое тестирование**

Для данного тестирования необходимо проверить работоспособность приложения на всех основных сценариях:

* Покупка билетов
* Авторизация
* Регистрация
* Добавление концертов

Тестирование проводилось вручную и охватило также дополнительные сценарии. Его результаты представлены на таблице 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Название | Шаги для воспроизведения | Ожидаемый результат | Статус теста |
| 1 | Регистрация | Нажатие на кнопку “Регистрация”  Ввод имени  Ввод фамилии  Ввод логина  Ввод пароля  Ввод email  Нажатие на кнопку “Регистрация” | Успешная регистрация и отображение главной страницы | Пройден |
| 2 | Авторизация | Нажатие на кнопку “Авторизация”  Ввод логина  Ввод пароля  Нажатие на кнопку “Войти” | Успешная авторизация и отображение главной страницы | Пройден |
| 3 | Покупка билета | Выбор концерта  Нажатие на кнопку “Купить” | Авторизованный пользователь переходит на страницу с ссылкой на билет | Пройден |
| Неавторизованный пользователь переходит на страницу с ссылкой на билет |
| 4 | Добавление концертов | Ввод концертных данных  Нажатие на кнопку “Добавить” | Успешное добавление концерта | Пройден |

Таблица 2 – Результаты дымового тестирования

## UI тесты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №  п/п | Начальная страница | Название | Шаги | Ожидаемый результат | Статус теста |
| 1 | Главная | Открытие главной страницы | Нажатие на кнопку “Главная” | Переход к главной странице | Пройден |
| 2 | Главная | Открытие формы регистрации | Нажатие на кнопку “Регистрация” | Переход к форме регистрации | Пройден |
| 3 | Главная | Открытие формы авторизации | Нажатие на кнопку “Авторизация” | Переход к форме авторизации | Пройден |
| 4 | Авторизация | Переход в учетную запись | Ввод верного логина | Отображение главной страницы авторизованного пользователя | Пройден |
| Ввод верного пароля |
| Нажатие на кнопку "Войти" |
| 5 | Авторизация | Неверный ввод данных при авторизации | Ввод верного логина | Отображение окна неверного ввода | Пройден |
| Ввод неверного пароля |
| Нажатие на кнопку "Войти" |
| 6 | Авторизация | Неверный ввод данных при авторизации | Ввод неверного логина | Отображение окна неверного ввода | Пройден |
| Ввод верного пароля |
| Нажатие на кнопку "Войти" |
| 7 | Регистрация | Приватное отображение пароля | Ввод пароля пользователем | Отображение символов "\*" | Пройден |
| 8 | Регистрация | Переход в учетную запись | Заполнение формы регистрации | Отображение главной страницы авторизованного пользователя | Пройден |
| Нажатие на кнопку "Зарегистрироваться |
| 9 | Учётная запись | Выход из учетной записи и отображение главной страницы | Нажатие на кнопку "Выход" | Отображение главной страницы неавторизованного пользователя | Пройден |
| 10 | Главная | Страница покупки и получения билета | Неавторизованный пользователь заполняет форму с данными | Получение билета | Пройден |
| Авторизованный пользователь переход к странице покупки билета |

Таблица 3 – Результаты UI тестирования

## Негативное тестирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Название | Шаги для воспроизведения | Ожидаемый результат | Статус теста |
| 1 | Авторизация | Ввод некорректных данных при авторизации | Сообщение «введён неверный логин или пароль» | Пройден |
| 2 | Регистрация | Ввод некорректных данных при регистрации | Сообщение «введены неверные данные» | Пройден |
|  |  |  |  |  |

Таблица 4 – Результаты Негативного тестирования

## Позитивное тестирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Название | Шаги для воспроизведения | Ожидаемый результат | Статус теста |
| 1 | Авторизация | Ввод корректных данных при авторизации | Ввод корректных данных в поля авторизации | Успешная авторизация |
| 2 | Регистрация | Ввод корректных данных при регистрации | Ввод корректных данных в поля регистрации | Успешная регистрация |

Таблица 5 – Результаты Позитивного тестирования

# Реализация

## Анализ средств реализации

В ходе разработки приложения будут использованы следующие образцы свободно распространяемого ПО:

-Веб-фреймворк Bootstrap

-PHP 7.3

-HTML 5

-MySQL

-CSS 3

Bootstrap– это **веб-фреймворк, который используется разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений. Основная его область применения – это FrontEnd разработка.**

PHP 7.3 является специализированным (но не единственным) языком, сконструированным для веб-разработок, благодаря чему его код может быть непосредственно внедрён в HTML. Версия 7.3 является оптимальной и стабильной версией языка, гарантирующей отказоустойчивость.

HTML 5 - язык для структурирования и представления содержимого всемирной паутины. Его цель — улучшение уровня поддержки мультимедиа-технологий с одновременным сохранением обратной совместимости, удобочитаемости кода для человека и простоты анализа для парсеров.

MySQL способно предоставить высокий уровень безопасности для бд, благодаря программе сервера базы данных, расположенной на отдельном компьютере вместе с самим хранилищем бд, как правило клиент и сервер MySQL находятся на разных устройствах, в совокупности с собственной системе защиты с настройкой доступа к бд для каждого пользователя.

CSS является языком описания стилей, необходимым и незаменимым для создания удобного и привлекательного для потенциальной аудитории пользователей интерфейсом, совместимым с HTML.

## Шапка сайта

На определенных страницах сайта есть «шапка», которая предоставляет следующие возможности пользователю:

1. перейти на главную страницу,
2. перейти на страницу регистрации,
3. перейти на страницу авторизации,

## Главная страница

Предоставляет пользователю возможность перейти на страницу с информацией о сайте и данными о концертах. Главная страница предоставляет следующие возможности пользователю:

* Позволяет ознакомиться с текущими концертами.
* Позволяет приобрести билеты пользователю.

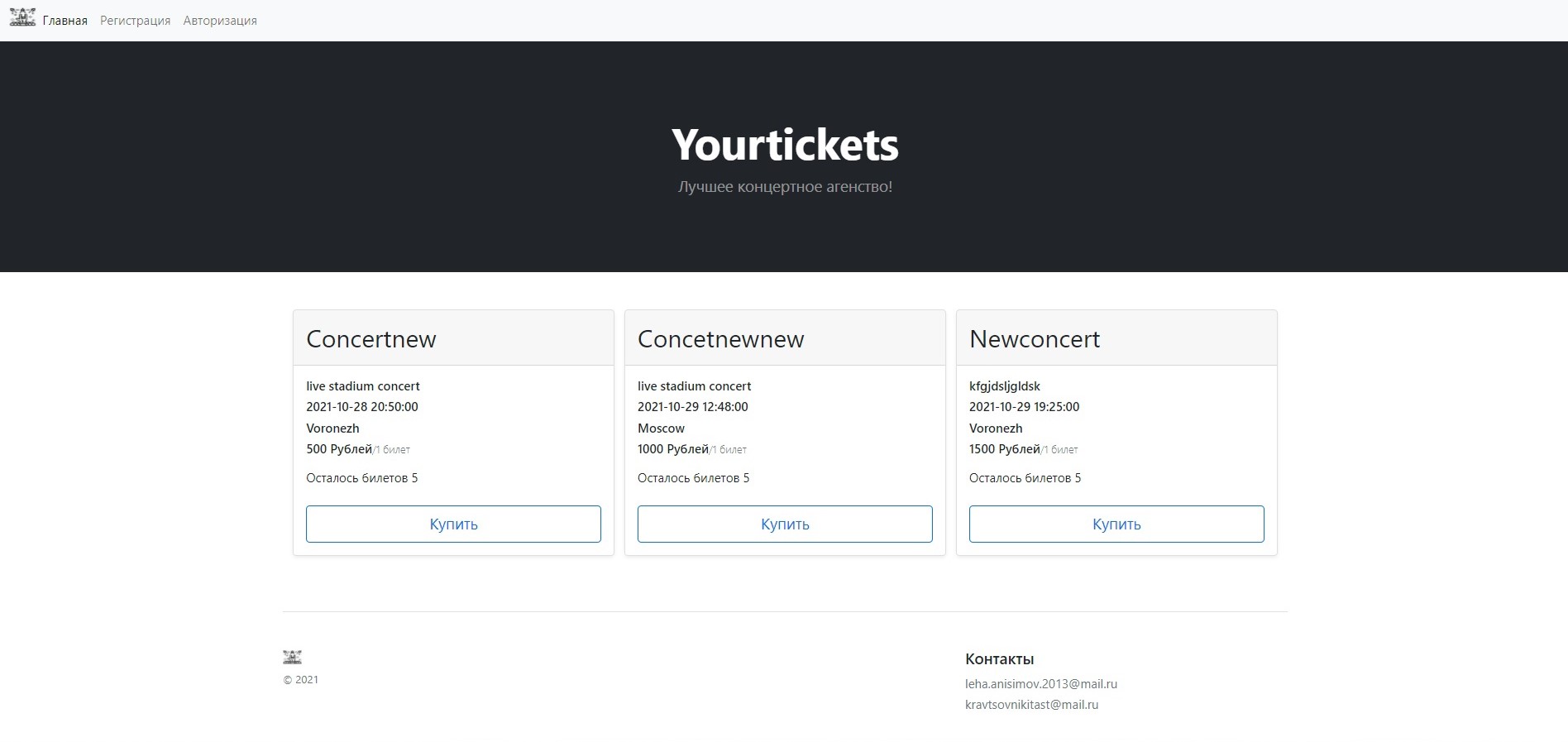


Рисунок 21 – Главная страница сайта

## Страница регистрации

Предоставляет пользователю возможность регистрации. Страница регистрации изображена на рисунке 22.

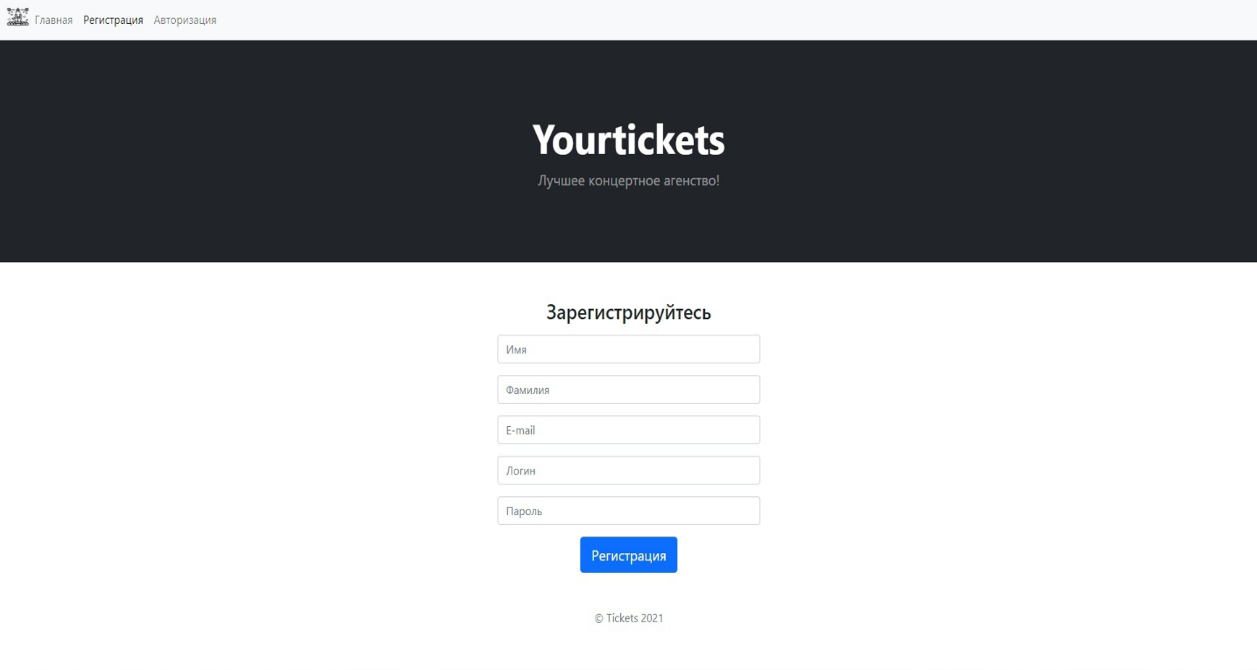


Рисунок 22 – Страница регистрации пользователя

## Страница авторизации

Предоставляет пользователю возможность авторизации. Страница авторизации изображена на рисунке 23.

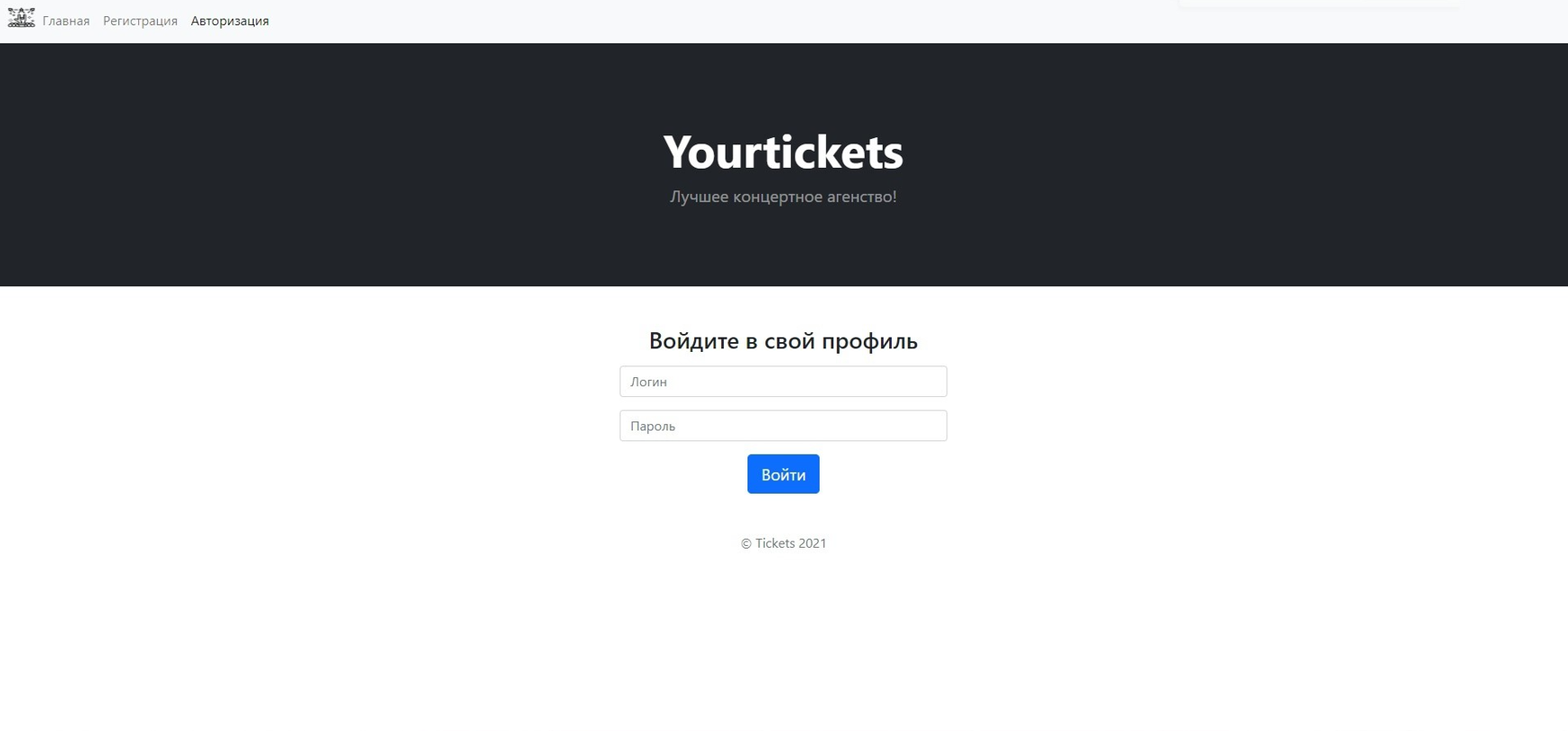


Рисунок 23 – Страница авторизации пользователя

# Заключение

В ходе выполнения курсового проекта была выявлена необходимость в создании приложения по поиску и покупке концертных билетов. Был проведен анализ предметной области и установлен список ключевых сценариев, а именно:

* Выбор концерта;
* Покупка билетов.

В результате, реализовано приложение, позволившее решить поставленные задачи:

* Регистрацию, авторизацию, выход из Учетной записи;
* Покупка и получение билетов пользователем.

Это приложение удовлетворяет изначальным требованиям:

* Разделение приложения на Front-end и Back-end;
* Наличие базы данных, содержащей не менее трех таблиц;
* Подключение всех вышеперечисленных частей и корректная их работа.

# Список источников

* <https://www.php.net> – справочник языка PHP
* <https://getbootstrap.com> – официальный источник фреймворка Bootstrap
* <http://www.fpdf.org> – библиотека генерации PDFфайлов на PHP