МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра технологий обработки и защиты информации

Техническое задание

в соответствии с ГОСТ 34.602-89

Исполнители\_\_\_\_\_\_\_\_\_\_\_\_ Янкевич А. А., Капкин К. А., Коротеева Е. Д.

Заказчик \_\_\_\_\_\_\_\_\_\_\_\_ Тарасов В. С.

Воронеж 2021

**Содержание**

[1 Общие сведения](#_heading=h.30j0zll) 3

[1.1 Наименование сайта](#_heading=h.1fob9te) 3

[1.2 Наименование заказчика](#_heading=h.3znysh7) 3

[1.3 Наименование исполнителя](#_heading=h.2et92p0) 3

[1.4 Основание для разработки](#_heading=h.tyjcwt) 3

[1.5 Плановые сроки начала и окончания работ](#_heading=h.3dy6vkm) 3

[1.6 Термины и сокращения](#_heading=h.1t3h5sf) 3

[2 Назначение и цели создания](#_heading=h.4d34og8) 5

[2.1 Назначение и цели системы](#_heading=h.2s8eyo1) 5

[2.2 Цели создания системы](#_heading=h.17dp8vu) 5

[3 Характеристика объектов автоматизации](#_heading=h.3rdcrjn) 6

[4 Требования к системе](#_heading=h.26in1rg) 7

[4.1 Требования к системе в целом](#_heading=h.lnxbz9) 7

[4.1.1 Требования к структуре и функционированию системы](#_heading=h.35nkun2) 7

[4.1.2 Показатели назначения системы](#_heading=h.1ksv4uv) 8

[4.1.3 Требования к защите информации от несанкционированного доступа](#_heading=h.44sinio) 8

[4.1.4 Требования к численности и квалификации персонала](#_heading=h.2jxsxqh) 8

[4.1.5 Требования к патентной чистоте](#_heading=h.z337ya) 9

[4.1.6 Требования к масштабируемости и открытости](#_heading=h.3j2qqm3) 9

[4.1.7 Обработка ошибок](#_heading=h.1y810tw) 9

[4.2 Требования к функциям, выполняемым системой](#_heading=h.4i7ojhp) 9

[4.3 Требования к видам обеспечения 1](#_heading=h.2xcytpi)2

[4.3.1 Требования к информационному обеспечению 1](#_heading=h.1ci93xb)2

[4.3.2 Требования к программному обеспечению 1](#_heading=h.3whwml4)2

[4.3.3 Требования к лингвистическому обеспечению 1](#_heading=h.2bn6wsx)3

[4.3.4 Требования к техническому обеспечению 1](#_heading=h.qsh70q)3

[4.4 Требования к дизайну системы 1](#_heading=h.3as4poj)3

[5 Состав и содержание работ по созданию (развитию) системы 14](#_heading=h.1pxezwc)

[6 Порядок контроля и приемки системы](#_heading=h.147n2zr) 15

[7 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие](#_heading=h.3o7alnk) 16

[8 Требования к документированию](#_heading=h.23ckvvd) 17

9 Источники разработки 18

# Общие сведения

## Наименование сайта

Сервис помощи организации концертов и покупки билетов на концерты “ConcertSpace”.

## Наименование заказчика

Старший преподаватель Тарасов Вячеслав Сергеевич, кафедра программирования и информационных технологий.

## Наименование исполнителя

Студентка Янкевич Анна Артёмовна, кафедра технологий обработки и защиты информации

Студент Капкин Кирилл Алексеевич, кафедра технологий обработки и защиты информации

Студентка Коротеева Елизавета Дмитриевна, кафедра технологий обработки и защиты информации

## Основание для разработки

Необходимость создания курсового проекта по дисциплине «Технологии программирования» для успешного прохождения курса.

## Плановые сроки начала и окончания работ

Плановый срок начала работ - Март 2020 г.

Плановый срок окончания работ - Июнь 2020 г.

## Термины и сокращения

ВИ - Вариант Использования или Use Case, описание см. <https://ru.wikipedia.org/wiki/> Сценарий\_использования.

Система – Сервис помощи создания команды “ConcertSpace”, требования к которому указаны в данном документе.

Фреймворк – программная платформа, определяющая структуру программной системы.

Валидация – проверка вводимых пользователем данным на корректность.

Модерация – процесс контроля действий пользователей на соответствие принципам ресурса.

Клик – нажатие левой клавишей мыши на объект.

Front-end – клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса.

Back-end – программно-аппаратная часть сервиса.

REST API – это стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов.

Github – крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис, интернет-сервис, система, веб-приложение, проект – идентифицируемая уникальным веб-адресом (URL-адресом) программная система со стандартизированными интерфейсами, а также HTML-документ сайта, отображаемый браузером пользователя.

Контент – наполнение веб-страницы.

# Назначение и цели создания

## Назначение и цели системы

Назначением системы является автоматизация процесса бронирования площадок, организации мероприятий, продажи и покупки билетов на мероприятия.

## Цели создания системы

Цели создания:

* упрощение процесса организации концерта;
* упрощения процесса покупки и продажи билетов;
* мониторинг доступных площадок в городе;
* получения информации об актуальных концертах в городе.

# Характеристика объектов автоматизации

Данная система предназначена для заказчика, которому необходим интернет-сервис организации концертов и продажи/покупки билетов. Для обслуживания в критической ситуации может потребоваться модератор, который будет вносить изменения в базу данных.

Сайт должен предполагать одну роль пользователя с двумя разными сценариями:

* Пользователь-организатор
* Пользователь-покупатель

# Требования к системе

# Требования к системе в целом

Разрабатываемый проект должен удовлетворять следующим основным требованиям:

* Должен стабильно работать в браузере Google Chrome, Opera.
* Иметь доступный и понятный интерфейс для пользователя.
* Реализовывать основные задачи, стоящие перед данным проектом.
* Просмотр списка всех доступных площадок
* Просмотр списка всех доступных концертов
* Поиск концертов по названию или адресу
* Поиск площадки по названию или адресу
* Покупка билета
* Бронирование площадки
* Возможность настраивать параметры при бронировании площадки
* Возможность продавать билеты на сайте приложения при бронировании площадки

### 4.1.1 Требования к структуре и функционированию системы

Созданное приложение будет иметь разделение на back-end и front-end, взаимодействие между которыми происходит с помощью REST API.

Базовая архитектура приложения указана ниже на рисунке 1.

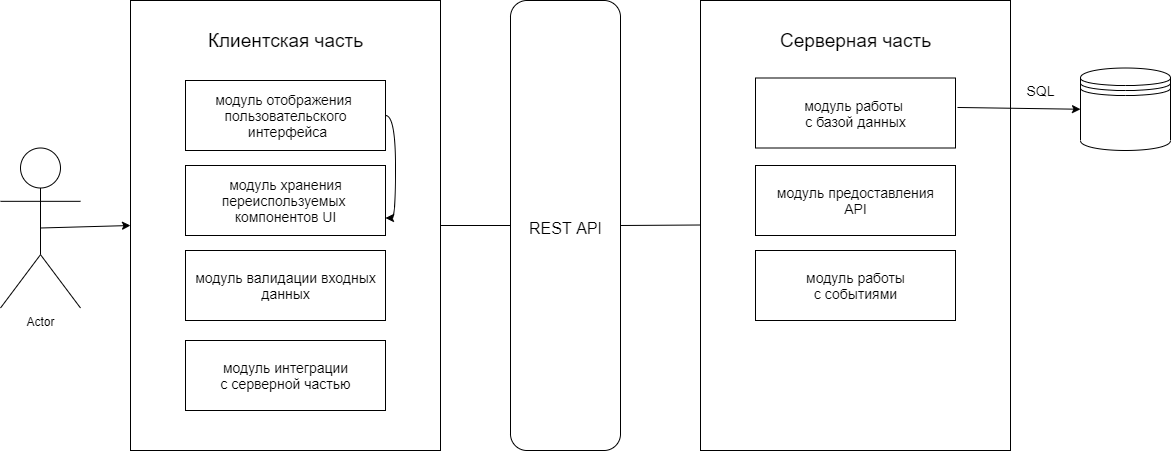


Рисунок 1 – Архитектура приложения

Стек используемых технологий (может меняться и дополняться в виду отсутствия конкретных требований заказчика, не противореча требованиям системы установленных в текущем документе):

1. Разработка системы планируется с использованием каскадной модели разработки программного обеспечения;
2. Для разработки клиентской части используется ReactJS, Material UI;
3. Для разработки серверной части Java - Spring Boot, Hibernate;
4. Для работы с базой данных MySQL.

### 4.1.2 Показатели назначения системы

* Реализация функции покупки билетов различных типов на мероприятия;
* Реализация функции бронирования площадки под концерты;
* Реализация функции продажи билетов на сайте приложения;
* Реализация просмотра описания концерта;
* Реализация просмотра описания площадки;
* Реализация поиска по названию или адресу площадки;
* Реализация просмотра списка всех мероприятий;
* Реализация просмотра списка всех площадок;
* Реализация функции редактирования параметров при бронировании.

### 4.1.3 Требования к защите информации от несанкционированного доступа

Система должны быть надёжно защищены от самых распространённых способов взлома сайтов и программ, а именно от SQL и XSS инъекций.

### 4.1.4 Требования к численности и квалификации персонала

В системе конкретных требований к численности персонала не приводится. В Системе предполагается наличие сценариев пользователя – пользователь-организатор и пользователь-посетитель.

Пользователи, работающие с данной системой, должны обладать базовыми навыками работы на персональном компьютере и в сети интернет.

### 4.1.5 Требования к патентной чистоте

Данный проект должен не нарушать никаких лицензий и патентов. В случае нарушения всю ответственность несет сторона Исполнителя.

### 4.1.6 Требования к масштабируемости и открытости

Проект должен предоставлять возможность добавлять новую функциональность с минимальным изменением существующего кода.

### 4.1.7 Обработка ошибок

В случае возникновения ошибок пользователь должен получать соответствующие сообщение об ошибке.

Приложение должно поддерживать обработку следующих основных ошибок:

* Некорректно введенные данные;
* Обработка ошибок со стороны клиента;
* Обработка ошибок со стороны сервера.

## Требования к функциям, выполняемым системой

Система должна удовлетворять следующим функциональным требованиям:

* Возможность покупки билетов нескольких типов
* Бронирование площадки под концерты
* Возможность продажи билетов на концерты на сайте приложения
* Просмотр краткого описания площадки при наведении курсора
* Просмотр полной характеристики площадки
* Просмотр краткого описания концерта при наведении курсора
* Просмотр полной характеристики концерта
* Поиск площадки по названию или адресу
* Поиск концерта по названию или адресу
* Просмотр списка всех мероприятий
* Просмотр списка всех площадок
* Возможность редактировать параметры при бронировании площадки

На Диаграммах вариантов использования представлены основные варианты использования системы.

**Действующие лица**

На данной диаграмме представлены 2 вида сценариев, которыми может воспользоваться любой пользователь нашего сервиса: Пользователь-организатор и Пользователь-покупатель. Эти пользователи никак друг от друга не зависят, но в то же время любой Пользователь может воспользоваться любым из двух сценариев в любой момент времени.

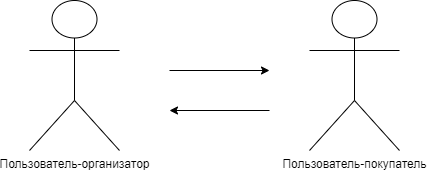


Диаграмма 1 − действующие лица.

**ВИ “Бронирование площадки”**

Пользователь-организатор (далее Пользователь) вводит поисковый запрос по названию или адресу, при наведении курсора на карточку площадки, Пользователь может видеть краткое описание этой площадки, при клике на карточку ему открывается страничка с полный описанием, где он может произвести настройку параметров по организации концерта (например, заказать диджея или фуршет), узнать итоговую стоимость организации концерта и забронировать площадку, а также, при желании, запросить продажу билетов на свой концерт на сайте приложения.

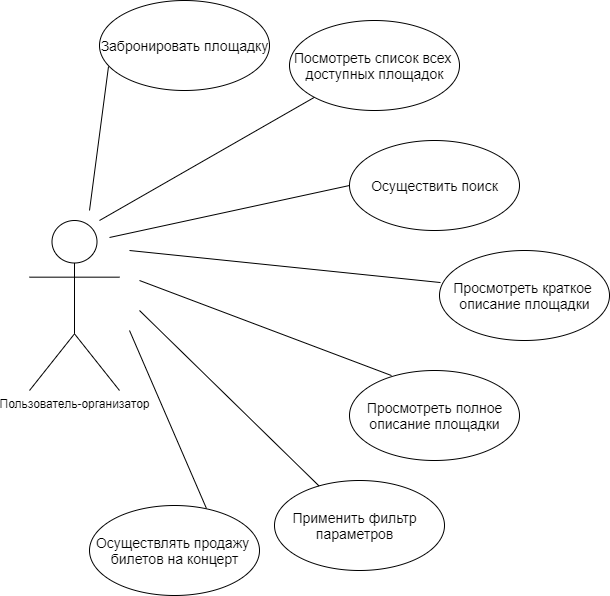


Диаграмма 2 − ВИ бронирования площадки.

**ВИ “Покупка билета”**

Пользователь-покупатель (далее Пользователь) вводит поисковый запрос по названию или адресу, при наведении курсора на карточку концерта, Пользователь может видеть краткое описание этого концерта, при клике на карточку ему открывается страничка с полным описанием концерта, где он может выбрать тип мест, их количество и узнать итоговую стоимость посещения.

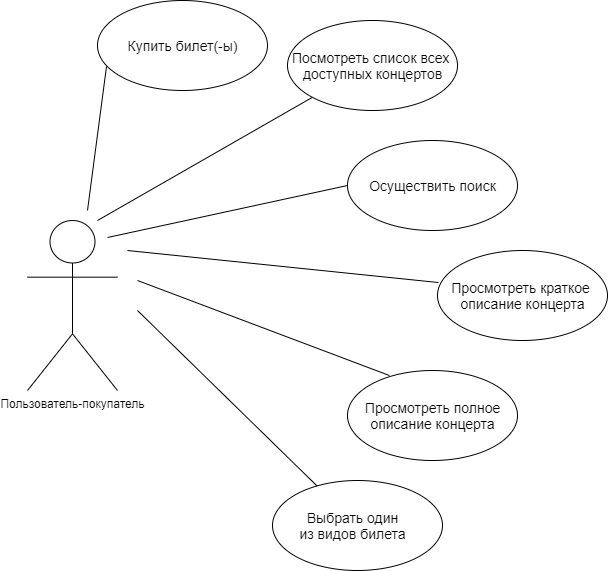


Диаграмма 3 ВИ − покупка билета.

## Требования к видам обеспечения

### 4.3.1 Требования к информационному обеспечению

Требования по применению систем управления базами данных: использование СУБД MySQL.

### 4.3.2 Требования к программному обеспечению

Для реализации серверной части должен использоваться высокоуровневый язык программирования Java с использованием фреймворка Spring Boot.

Для реализации алгоритмов манипулирования данными в системе поиска команд необходимо использовать стандартный язык запроса к данным SQL.

Для организации диалога системы с пользователем должен применяться графический пользовательский интерфейс.

В результате разработки данные требования могут расширяться.

### 4.3.3 Требования к лингвистическому обеспечению

Приложение должно поддерживать русский язык. Язык ввода-вывода данных русский. Язык манипулирования данными MySQL.

### 4.3.4 Требования к техническому обеспечению

Система должна выполняться в последних стабильных версиях браузеров Google Chrome и Opera.

## Требования к дизайну системы

Дизайн должен быть достаточно ярким, привлекающим внимание Пользователя, но не отвлекающим от контента. На всех страницах содержится Header - основное навигационное меню.

# Состав и содержание работ по созданию (развитию) системы

|  |  |  |
| --- | --- | --- |
| Этапы | Содержание работ | Результат работ |
| Этап 1  (проектирование) | Анализ предметной области, разработка технического задания, разработка модели программы. | Документация на разрабатываемое приложение, в которой описаны все необходимые организационные требования к разработке. |
| Этап 2  (реализация) | Разработка программы и документации. | Приложение, готовое к тестированию; полная документация. |
| Этап 3  (окончание работ) | Проведение тестов, исправление программы, подготовка презентации, сдача проекта Заказчику. | Акт сдачи-приемки  выполненных работ. |

# Порядок контроля и приемки системы

Контроль разработки системы осуществляется посредством предоставления необходимой Заказчику отчетности Исполнителем. Готовая система с полной документацией должна быть представлена Заказчику в требуемые им сроки. Заказчик определяет соответствие Системы требованиям и осуществляет её приём.

При приеме системы Исполнитель обязан предоставить:

* Техническое задание;
* Исходный код системы;
* Демонстрационное видео проекта со всеми ключевыми сценариями;
* Тест кейсы;
* Курсовой проект;
* Презентация проекта.

# Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

При подготовке объекта автоматизации к вводу системы в действие необходимо выполнить следующие работы:

* Необходимо осуществить набор персонала: минимум одного модератора для внесения данных в БД
* Провести обучение персонала, ознакомить персонал с возможностями системы
* Настройка интеграции со смежными системами;
* C технической точки зрения, это приложение может быть развернуто в любой из основных операционных систем, список, который включает в себя большое количество дистрибутивов Linux и BSD с открытым исходным кодом, а также коммерческую ОС X и Microsoft Windows. Для приложения будет необходимо минимум 512 Мб ОЗУ и одноядерный процессор с тактовой частотой 1 ГГц. Также необходим объем памяти в 512 Мб;
* Необходимо провести настройку системы: создать и наполнить базу данных дефолтными или начальными значениями.

# Требования к документированию

Документирование проекта в рамках Техническое Задания ведётся в соответствии с ГОСТ 34.602-89.

Вся документация должна быть подготовлена и передана, как в печатном, так и в электронном виде (в формате docx или pdf), а также размещена на GitHub.

Также осуществляется предоставление Курсового проекта на основе данного Технического Задания.

# Источники разработки

1. Spring Boot Reference Documentation / [сайт]. – URL: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
2. React documentation / [сайт]. – URL: <https://reactjs.org/>
3. Современный учебник JavaScript / [сайт]. – URL: <https://learn.javascript.ru/>
4. Material UI documentation / [сайт]. – URL: <https://material-ui.com/ru/>