МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОЙ РАБОТЫ**

**«РАЗРАБОТКА ПРОЕКТА ПО СОЗДАНИЮ СИСТЕМЫ**

**УПРАВЛЕНИЯ СЕКЦИЕЙ ТУРИЗМА»**

Студент Кислюк И. В.

Факультет Инфокоммуникационные технологии

Кафедра Программных систем

Группа К4120

Направление (специальность) 11.04.02 Программное обеспечение в инфокоммуникациях

Руководитель Иванов Сергей Евгеньевич, доцент, к. ф.-м. н.

Дисциплина Разработка web-сервисов в инфокоммуникационных системах

Санкт-Петербург

2018 г.

Оглавление

# Введение

В данный момент активно развиваются спортивные секции. Большинство молодых людей увлекаются походами в спортивные залы, принимают участие в спортивных соревнованиях среди большого количества дисциплин. Однако перенесение учета и контроля участников соревнований и спортивных секций в данной области остается на бумажных носителях.

В связи с этим встает острая потребность в модернизации системы учета соревнований и участников спортивных секций. Данное улучшение позволит ускорить обработку заявок участников, а также в режиме реального времени взаимодействовать с секцией спорта.

# Часть 1. Цели проекта

Целью данного проекта является создание проекта учета спортивных мероприятий, отвечающего функциональным, техническим и операционным требованиям и обеспечивающего необходимые параметры, такие как производительность, масштабируемость, безопасность и управляемость.

Система должна уметь сохранять информацию о:

* спортивных мероприятиях, о времени, месте проведения, необходимом спортивном разряде
* участниках спортивных мероприятий
* администраторах
* организаторах мероприятий

Основные цели работы:

• систематизация, закрепление и расширение теоретических и практических знаний по дисциплине «Разработка web-сервисов в инфокоммуникационных системах» и применение их при решении конкретных задач;

• приобретение навыков проектирования распределенных систем;

• развитие творческого мышления и навыков самостоятельной работы;

• приобретение навыков использования справочной, нормативной и научной литературы, а также Интернет-ресурсов;

• приобретение навыков оформления пояснительной записки

### Анализ предметной области

Основным профилем деятельности спортивной секции является создание и проведение спортивных мероприятий. Данная система включает в себя описание различных мероприятий, участников, тренеров, спортивных судей и пр.

Рассмотрим более подробно процесс проведения спортивного мероприятия. Администратор назначает менеджеров спортивной секции или сайта. В данном случае будет выступать организатор или руководитель спортивной секции. После получения необходимых прав, руководитель назначает спортивное мероприятие, в котором необходимо указать место проведения спортивного мероприятия, в данном случае это может быть «Спартакиада для магистрантов университета ИТМО»; дату проведения мероприятия, указать необходимые спортивные разряды, указать вид спорта, в случае если проводится какая-то конкретная олимпиада или соревнование. С свою очередь всем желающим участвовать необходимо произвести регистрацию на сайте, которая будет включать предоставление регистрирующимся своих личных данных, таких как фамилия, имя и отчества, дата рождения, выбор участников логина или пароля, предоставления номера телефона и пр. После регистрации участникам будут доступно принятие участия среди списка предоставленных спортивных мероприятий.

### Обзор аналогичных веб-сервисов.

Для понимания успешности будущей разработки необходимо провести анализ и выявление недостатков, а также для определения требований проведем аналитический обзор существующих сервисов для спортивных секций. Рассмотрим наиболее популярный из них.

Сайт “Спорт для детей и взрослых” (https://sportschools.ru) предоставлен как наиболее популярный ресурс по работе спортивных секций. Пример окна интерфейса приведен на рисунке 1.

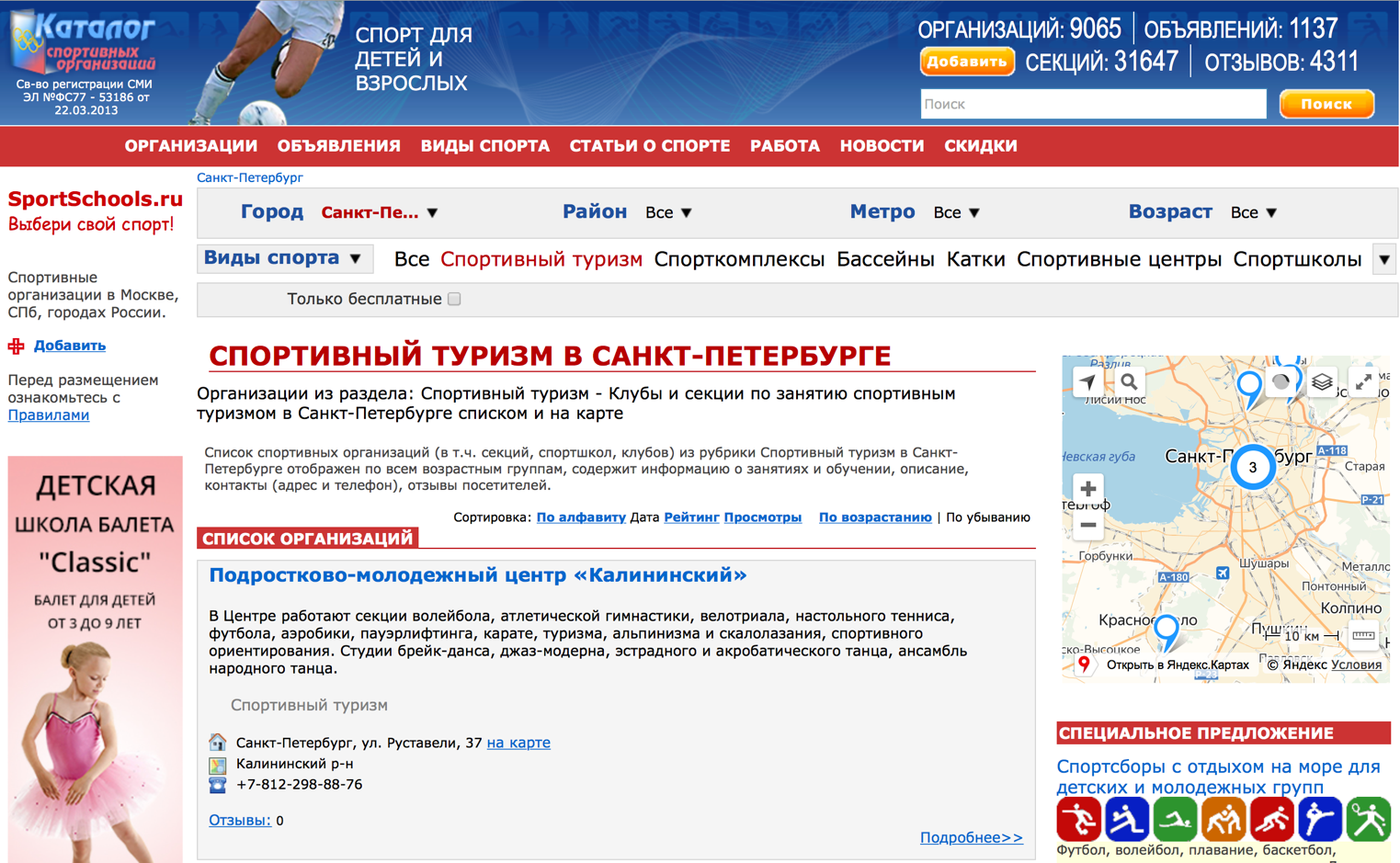


Рисунок 1 – пример сайта спортивного туризма

Сервис подойдет всем желающим, которые хотят ознакомиться с каталогом существующих спортивных секций туризма в определенном регионе России. Сервис, по заявлению разработчиков, предоставляет наиболее актуальную информацию обо всех видах спорта, предоставляет специальные предложения по работе со детскими спортивными юношескими школами (ДЮСШ), спортивными кружками и детскими спортивными школами.

Основные задачи, которые решает система:

- предоставление наиболее полной информации по существующим спортивным секциям в выбранных городах, регионах России

- возможность фильтрации результатов поиска по региональной принадлежности – по городу, региону, станции метро

- возможность расширенной фильтрации по возрастным ограничениям, а также по виду спорта

- просмотр наиболее актуальных статей о спорте и околоспортивных темах, таких как спортивное питание.

Рассмотрим преимущества и недостатки работы с данным web-сервисом.

Преимущества работы:

- просмотр актуальной контактной информации действующих спортивных секции и кружков

- возможность просмотра скидок и специальных предложений для посещения кружков и мероприятий, возможность посещения бесплатного занятия

- возможность добавление своей спортивной организации или секции

Недостатки работы с данным сервисом:

- отсутствие личного кабинета для участников соревнований

- невозможность контроля количества участников спортивных мероприятий

- недостаточная функциональность для регистрации спортивных мероприятий, а не секции или спортивных школ

# Глава 2. Обзор и выбор подходящей технической платформы

### Фундаментальная архитектура web-сервиса

В основе разрабатываемой архитектуры лежит известная «клиент-серверная» архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. В качестве среды взаимодействия клиента с сервером используется Интернет (рисунок 2).

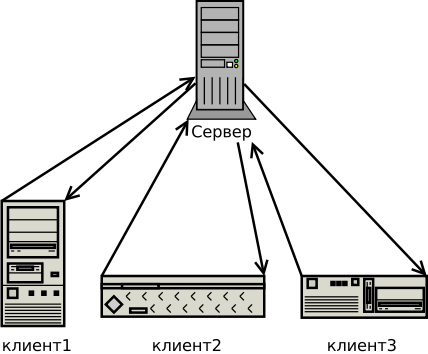


Рисунок 2 – пример клиент-серверного взаимодействия

Основные особенности работы с «клиент-серверной» архитектурой

Преимущества

- Отсутствие дублирования кода программы-сервера программами-клиентами.

- Так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются.

- Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще организовать контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.

Недостатки

- Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т. п.

- Поддержка работы данной системы требует отдельного специалиста — системного администратора.

- Высокая стоимость оборудования.

### Выбор системы управления базой данных

Поскольку основная работа сервера направлена на долговременное хранение данных, благоразумным будет включение изначально базы данных в предоставляемый проект.

Для обеспечения доступа к информации, которая будет связана со спортивными секциями, хранящимися в базе данных, а также для управления ею, применяют систему управления базами данных (СУБД). СУБД — это комплекс языковых и программных средств, предназначенный для создания, ведения и совместного использования БД многими пользователями. Обычно СУБД различают по используемой модели данных. Так, СУБД, базирующиеся на использовании реляционной модели данных, называют реляционными СУБД. Системы управления базами данных помогают отсортировать информацию, а также связать базы данных между собой, при этом предоставив отчет об изменениях и зарегистрированных событиях.

Приведем сравнение наиболее популярных баз данных. Например, если речь идёт о выборе СУБД для предприятия, то следует принять во внимание возможность СУБД «расти» вместе с развитием организации. Малому бизнесу могут потребоваться только базовые функции и возможности, а также небольшое количество информации, размещаемой в БД. Но требования могут существенно расти с течением времени, а переход на другую СУБД может стать проблемой.

**Oracle**

Корпорация Oracle предлагает одноимённый продукт, с которого обычно начинается рассмотрение вариантов популярных СУБД. Первая версия Oracle была создана в конце 70-х годов, имея на данный момент блестящую репутацию. Кроме того, существует несколько версий этого продукта для удовлетворения потребностей конкретной организации. Безопасность в этой версии доведена до высочайшего уровня, потому что каждая транзакция изолирована от других.

Достоинства

* самые свежие инновации и большое количество функционала
* высокая степень надежности

Недостатки

* высокая стоимость и требовательность к ресурсам

**MySQL**

MySQL – одна из самых популярных баз данных для веб-приложений. Фактически, является стандартом для веб-серверов, которые работают под управлением операционной системы Linux. MySQL – это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала, который может стать и как достоинством, так и недостатком – в зависимости от области внедрения.

Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. Она также имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных. Система надежна и не стремится подчинить себе все доступные аппаратные ресурсы.

Достоинства

* бесплатное распространение
* документированность
* функциональность
* поддерживает набора пользовательских интерфейсов
* работа с другими базами данных, включая DB2 и Oracle.

Недостатки

* сложность настройки
* отсутствие встроенная поддержка XML или OLAP
* для бесплатной версии доступна только платная поддержка

**Microsoft SQL сервер**

Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем возможно комбинация типов применяемых серверов одновременно. Среди особенностей версии 2016 года – temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. А последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

Достоинства

* простота в использовании
* скорость и стабильность
* визуализация на мобильных устройствах
* взаимодействие с другими продуктами Microsoft

Недостатки

* высокая стоимость
* ресурсоемкость

**PostgreSQL**

PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов. Это одна из первых разработанных систем управления базами данных, поэтому в настоящее время хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использована на большинстве основных платформ.

Достоинства

* масштабируемость и способность обрабатывать терабайты данных
* поддержка формата json
* наличие множества предопределенных функций

Недостатки

* отсутствие документации
* сложность конфигурации

**MongoDB**

Предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядро является очень гибким и работает при подключении базы данных к приложениям через драйверы MongoDB. Существует широкий выбор доступных драйверов, поэтому легко найти драйвер, который будет работать с требуемым языком программирования.

Поскольку изначально система MongoDB не была разработана для обработки моделей реляционных данных, могут возникнуть проблемы производительности, если использовать её таким образом. Однако, движок предназначен для обработки различных данных, которые нельзя отнести к реляционным, и может хорошо справляться там, где другие движки работают медленно или бессильны.

Достоинства

* скорость и простота в использовании
* поддержка json и традиционных документов NoSQL
* данные любой структуры могут быть сохранены/прочитаны быстро

Недостатки

* SQL не используется в качестве языка запросов
* сложность установки

Среди представленного сравнения базы данных была выбрана PostgreSQL СУБД, как наиболее старая в развитии и с поддержкой большого объема данных, а также из-за простоты настройки и использования.

### Выбор языков программирования

Web-сервис состоит из двух равноценных частей: серверной и клиентской. Серверная часть будет реализована на Spring – фреймворке для Java, как для наиболее популярного и самого старого кроссплатформенного языка. Это позволит не привязываться к операционной системе веб-сервера и иметь возможность запускать его на любом из выбранных серверов.

Для клиентской части будут использоваться соответствующие технологии:

- HTML – язык разметки документов во всемирной паутине

- CSS – язык таблиц каскадных стилей

- JavaScript – язык для управление контентом на стороне пользователя

# Глава 3. Проектирование

Формирование требований к проекту

Описать возможность автоматической генерации базы данных, указать недостатки

Пример базы данных (скриншоты и описание)

Прототипирование графического интерфейса

Сценарии использования

Тестирование

Вывод, в котором будет сказано, про нормальную реализацию

Оформление

# Вывод