

Министерство науки и высшего образования Российской Федерации Федеральное  
государственное бюджетное образовательное учреждение высшего образования  
**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ  
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра автоматизации обработки информации (АОИ)

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ  
ОРГАНИЗАЦИИ И ПРОВЕДЕНИЯ СПОРТИВНЫХ СОРЕВНОВАНИЙ**

**ОТЧЁТ  
ПО РЕЗУЛЬТАТАМ**

Проектного практикума

Выполнил:

Студент гр. 429-3

\_\_\_\_\_ Кутольвас В.Б.

« \_\_\_\_ » \_\_\_\_\_ 2022 г.

Проверил:

доцент каф. АОИ

\_\_\_\_\_ Морозова Ю.В.

« \_\_\_\_ » \_\_\_\_\_ 2022 г.

Томск 2022

## Содержание

Выбор объекта автоматизации.....	3
Выбор фреймворка для разработки.....	4
Аналоги системы.....	10
Техническое задание.....	11
Графический макет.....	15
Заключение.....	19
Список литературы.....	20

## Выбор объекта автоматизации

Любая сфера жизнедеятельности человека в настоящее время подвергается автоматизации, будь то производство или быт, хобби или бизнес. Для любой сферы необходимо наличие информационных систем (далее ИС), способствующих облегчению деятельности и освобождению времени от рутинных задач. Проблема автоматизации стоит и перед организаторами спортивных мероприятий, так как ведение регистрации команд-участников и протоколов соревнований в бумажном виде не соответствуют требованиям третьего десятилетия XXI века.

В данной работе целью ставится создание программного комплекса для организации спортивных турниров. В процессе создания подразумевается сравнение аналогичных ИС, исследование возможностей различных фреймворков и библиотек, позволяющих реализовать данный комплекс, сравнение их качественных и количественных характеристик с последующим выбором тех из них, чей функционал наиболее удовлетворяет поставленным требованиям, а также исследование возможностей выбранного стека технологий.

Первым делом была изучена предметная область, получены первичные знания о терминах и понятиях обозреваемой области знаний и составлена интеллектуальная карта, визуализирующая общие представления о реализуемой ИС (рисунок 1).

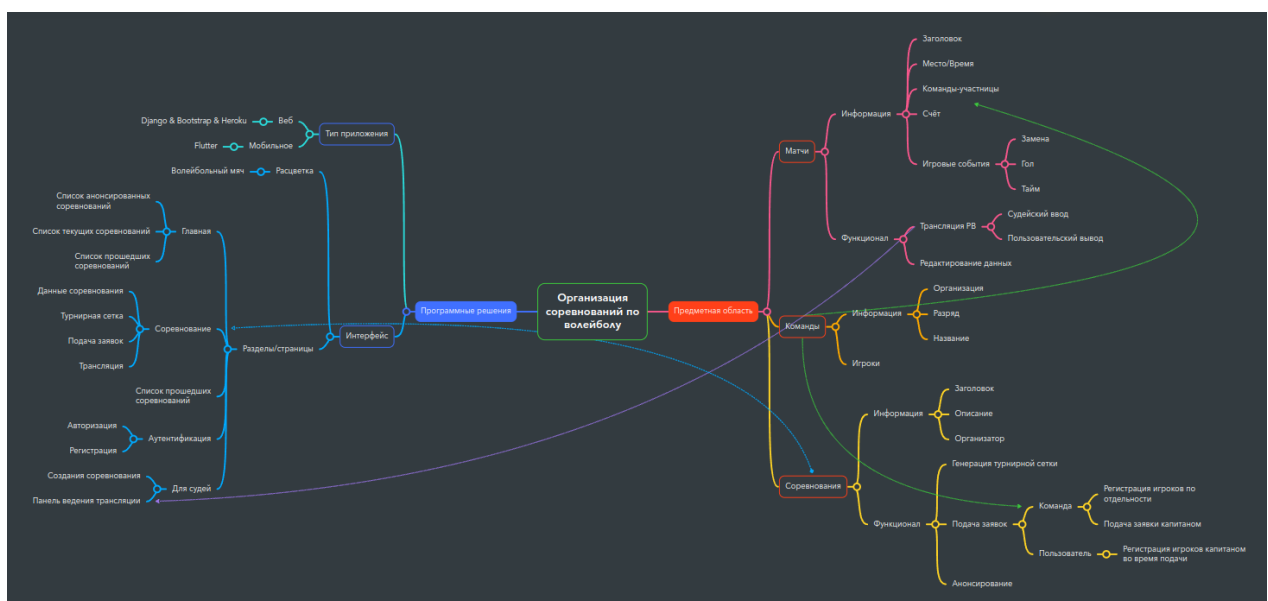


Рисунок 1 - Интеллектуальная карта

## Выбор фреймворка для разработки

Создание веб-приложения возможно несколькими способами. Первый и самый простой из них - блочные конструкторы сайтов, такие как Tilda или Wix. Данная технология ставит значительные ограничения в функционале приложения, так как предназначена для создания узкоспециализированных однотипных сайтов, например интернет магазинов, лендингов, сайтов-визиток. Так как в случае создания системы для проведения спортивных соревнований необходимо наличие расширенного списка функций, необходимо воспользоваться другим типом инструмента, которым является фреймворк.

В данном разделе будет производиться сравнение и выбор фреймворка для разработки серверной части приложения, так как функционал приложения реализуется именно в серверной части.

Подход к выбору фреймворка предусматривает оценку того, какой функционал уже реализован в его модулях, наличие в нём архитектурного шаблона, обширность документации и прочих параметров. В процессе выбора инструмента необходимо определить, для решения каких задач он будет использован, для каких целей конструируется сайт и с какими технологиями будет интегрирован веб-ресурс.

Рассмотрим наиболее популярные фреймворки для веб-бэкенд разработки, такие как: Django, Symfony, Flask.

**Django.** Популярный python веб-фреймворк, работающий что называется, из коробки, при этом имеет большой перечень библиотек, предоставляющих разработчику самый разнообразный функционал при низком пороге вхождения и большом сообществе[1].

Некоторые возможности Django:

- ORM, API доступа к БД с поддержкой транзакций
- встроенный интерфейс администратора, с уже имеющимися переводами на многие языки
- диспетчер URL на основе регулярных выражений
- расширяемая система шаблонов с тегами и наследованием

- подключаемая архитектура приложений, которые можно устанавливать на любые Django-сайты
- «generic views» — шаблоны функций контроллеров
- система фильтров («middleware») для построения дополнительных обработчиков запросов, как например включённые в дистрибутив фильтры для кеширования, сжатия, нормализации URL и поддержки анонимных сессий.

**Symfony.** Это полнофункциональный веб-фреймворк. Он содержит набор повторно используемых компонентов PHP. Вы можете использовать любые компоненты Symfony в приложениях независимо от платформы[2].

Symfony обладает огромным количеством функциональных возможностей и активным сообществом. Он имеет гибкую конфигурацию с использованием YAML, XML или аннотаций. Symfony интегрируется с независимой библиотекой и модулем PHP. Symfony в основном вдохновлен средами веб-приложений Ruby on Rails, Django и Spring. Компоненты Symfony используются во многих проектах с открытым исходным кодом, включая Composer, Drupal и phpBB.

Некоторые из характерных особенностей Symfony Framework следующие:

- Архитектура системы на основе модели-представления-контроллера
- Гибкая маршрутизация URL
- Код многоразовый и простой в обслуживании
- Полнофункциональные классы баз данных с поддержкой нескольких платформ
- Поддерживает огромное и активное сообщество
- Защита от подделки межсайтовых запросов и других атак

**Flask** — это фреймворк, который зависит от внешних библиотек. Перед тем, как его использовать, необходимо убедиться в том, что установлены Werkzeug и Jinja 2[3].

Werkzeug – это набор инструментов для WSGI, который используется как для разработки приложения, так и развертывания.

Jinja2 – это шаблонизатор, который позволяет создавать HTML-шаблоны с помощью Python. Flask – это удобный инструмент для генерирования таких шаблонов.

Фреймворк имеет следующие возможности:

- Сервер разработки и отладчик
- Отправка запросов RESTful
- Использование шаблонизатора Jinja2
- Поддержка безопасных файлов cookie (сеансы на стороне клиента)
- 100% соответствие WSGI 1.0
- Совместимость с Google App Engine
- Расширения для улучшения желаемых функций

Основные характеристики фреймворков приводятся в таблице 1.1.

Таблица 1 - Сравнение различных фреймворков

Функция	Django	Symfony	Flask
Архитектурный шаблон MVC	+	+	-
Защита от атак	+	+	+-
Модульность	-	+	+
Шаблонизатор	+	+	+

(+-) - есть в подключаемых модулях.

Теперь необходимо обозначить критерии, в соответствии с которыми будет производиться оценка того или иного фреймворка.

Все критерии можно разделить на несколько групп:

- назначение и возможности фреймворка;
- стоимость;

- интеграция ПП;
- бизнес-логика решения;
- опыт внедрений;
- архитектура;
- документация.

Всего было выделено 7 критериев, по которым будем оценивать предложенные выше инструменты.

Для обоснования выбора среды разработки введем некую балльную шкалу от 0 до 10 баллов, где:

0 – функция отсутствует в имеющейся конфигурации программного обеспечения;

2 – функция реализована частично, для ее реализации необходима серьезная доработка программного кода при настройке/внедрении;

4 – функция реализована частично, для ее реализации необходима незначительная доработка программного кода при настройке/внедрении;

6 – функция реализована удовлетворительно, требуется адаптация под нужды Предприятия в процессе настройки/внедрения средствами ИСУ;

8 – функция реализована хорошо, однако в перспективе могут понадобиться ее доработки;

10 – функция реализована полностью, удовлетворяет требованиям, в том числе – на перспективу.

Для наглядности сведём все оценки в единую таблицу (Таблица 2.2).

Таблица 2 - Оценка представленных фреймворков

Название фреймворка	Критерии						
	1	2	3	4	5	6	7
Django	10	10	8	8	8	8	8
Symfony	8	10	6	8	8	8	10
Flask	4	8	6	6	8	2	10

Теперь используя формулы:

1) Нахождение средневзвешенной оценки каждого пакета

$$Q_{kj} = (1/m_k) \sum_{i=1}^{m_k} \lambda_i (q_{ij})$$

Здесь  $Q_{kj}$  – оценка  $k$ -й группы критериев для варианта  $j$ ;  $(q_{ij})_k$  – оценка  $i$ -го показателя в группе  $k$  для варианта  $j$ ;  $\lambda_i$  – взвешенные коэффициенты, учитывающие важность признака  $i$ -го показателя для  $k$ -й группы критериев;  $m_k$  – Количество показателей в группе  $k$ .

2) Комплексная оценка  $W_j$  пакета для варианта  $j$ :

$$W_j = (1/n) \sum_{k=1}^n \lambda_k Q_{kj}$$

Здесь  $Q_{lj}$  – оценка  $l$ -й группы критериев для варианта  $j$ ;  $\lambda_l$  – взвешенные коэффициенты, учитывающие важность признака для  $l$ -й группы критериев.

Установим наиболее подходящий пакет прикладных программ для реализации нашей ИС. Выводы представим также в виде таблицы 2.3.



Таблица 3 - Результаты оценки фреймворков

Название	$Q_{kj}$	$W_j$
Django	8.6	8.34
Symfony	8.3	8.13
Flask	6.3	6.19

Из результатов видно, что наиболее предпочтительным фреймворком для реализации проекта является Django.

## Аналоги системы

После составления ТЗ был проведён обзор существующих аналогов создаваемой системы. На данном этапе были выделены следующие ресурсы, предоставляющие возможность создать турнирную сетку для самых разнообразных видов спорта, реальных или виртуальных: [challonge.com](http://challonge.com) и [tlab.pro](http://tlab.pro). Все перечисленные программные системы являются веб-приложениями. Такой формат является удобнее нежели мобильное приложение в случае несистемного участия, организации или просто наблюдении пользователем за соревнованиями за счёт отсутствия необходимости устанавливать приложение на устройство и возможности получить доступ к ресурсу с любого устройства, на котором есть браузер.

**challonge.com.** В сгенерированной турнирной сетке возможно проведение матча за третье место, но нет возможности внесения результатов матча, а также дополнительной информации по матчу.

**tlab.pro.** Минималистичный интуитивно понятный интерфейс, две системы проведения турниров.

В вышеприведённых системах имена участников и названия команд вносятся организатором. В разрабатываемой же системе заявку на участие будут оставлять сами участники, после чего она автоматически распределяются на матчи.

Таблица 4 - Сравнение аналогов системы

	challonge.com	tlab.pro
Внесение/хранение результатов матчей	-	-
Ведение трансляции игр	+	+
Автоматическая регистрация команд	-	-
Две системы проведения соревнований	-	+
Удобство интерфейса	-	+
Большое количество спортивных дисциплин	+	-

## **Техническое задание на разработку системы организации спортивных соревнования “SCS”**

### **1. ЭТАПЫ И СРОКИ СОЗДАНИЯ САЙТА:**

Общий срок работ по созданию веб-приложения для организации спортивных соревнований “SCS” составляет 30 дней (5 дней проектирование, создание макета сайта + 10 дней вёрстка шаблона и разработка полного статического прототипа сайта + 10 дней программирование динамики и подключение системы управления контентом + 5 дней тестирование).

Этапы создания сайта:

- Разработка концепции сайта, информационное проектирование, оформление Технического задания;
- Разработка эскиза дизайна;
- Программирование серверной части на базе фреймворка Django;
- Вёрстка страниц веб-ресурса;
- Тестирование;
- Развертывание сервера в сети интернет.

### **2. ТЕХНОЛОГИЧЕСКИЕ ТРЕБОВАНИЯ**

**2.1.** Сайт разрабатывается с адаптивным разрешением экрана.

**2.2.** Корректное отображение браузерами Google Chrome 79 и новее, Firefox 96 и новее, Яндекс.Браузер 18.2.1 и новее.

**2.3.** Использование цветов волейбольного мяча.

### **3. USER STORY**

**3.1** Как пользователь сервиса, я хочу получать рассылку о проводимых соревнованиях и матчах, чтобы знать время трансляции интересующего меня матча.

**3.2** Как пользователь сервиса, я хочу просматривать трансляции матчей, чтобы быть в курсе успехов сборной своего университета.

**3.3** Как участник соревнований, я хочу иметь возможность регистрации на участие в любом событии, чтобы в последующем принять в нём участие.

**3.4** Как судья, я хочу иметь возможность анонсирования соревнований, чтобы оповестить об этом всех участников и зрителей.

**3.5** Как судья, я хочу иметь возможность администрирования соревнований, чтобы транслировать и хранить информацию о событиях проводимых соревнований.

### **4. СТРУКТУРА И ОПИСАНИЕ САЙТА:**

#### **4.1 Шапка сайта**

- Содержит ссылки на главную страницу, кнопки входа, выхода и регистрации.
- При судейских правах доступа содержит кнопку-ссылку создания нового соревнования.
- Является общим элементом для всех страниц приложения.

#### **4.2 Главная страница (список соревнований)**

- Содержит список анонсированных, прошедших, текущих соревнований.
- Список анонсированных соревнований является основным контентом и занимает всю ширину контент-области страницы, ниже него занимая по пол страницы идут списки прошедших и текущих соревнований.
- При переполнении списка прошедших соревнований должна появляться ссылка на страницу списка прошедших соревнований.
- Элемент каждого списка несёт в себе основную информацию о соревновании: заголовок, описание, организатор, время начала, окончания, либо же время проведения следующего матча.

#### **4.3 Страница прошедших соревнований**

- Представляет список прошедших соревнований, представленных в виде элементов, аналогичных элементам главной страницы.
- Имеет пагинатор, пагинация по 10 соревнований.

#### **4.4 Страница соревнования**

- Содержит форму регистрации на соревнования в случае анонсированного матча.
- Редактирование данных матчей и соревнования в случае судейского доступа.

#### **4.5 Страница администрирования матча**

- Содержит панель администрирования матча, позволяющую вносить и удалять данные событий, происходящих во время матча.
- Должны быть предусмотрены следующие события: гол, замена, тайм-аут, начала раунда.

#### **4.6 Другие страницы**

- Страница авторизации
- Страница регистрации

### **5. КОНТЕНТ**

Основным разделом информационной структуры является список соревнований. В системе существует три списка:

- Прошедшие соревнования
- Текущие соревнования
- Анонсированные соревнования

Элементом каждого из списков является соревнования. Каждое соревнование несёт в себе следующую информацию:

- Заголовок;

- Описание;
- Организация, проводящая соревнования;
- Время подачи заявок (отображается только для анонсированных соревнований)
- Время следующего матча (отображается только для текущих соревнований)
- Время завершения соревнований (отображается только для завершённых соревнований)
- Протокол соревнования (отображается только для завершённых соревнований)
- Турнирная сетка

Турнирная сетка содержит в себе информацию о матчах, являющихся следующим разделом информационной структуры. Каждый матч имеет:

- Название
- Команды участники
- Счёт
- Время проведения
- Место проведения
- Протокол матча
- Перечень игровых событий, с их временем

## **6. РАЗВЁРТЫВАНИЕ**

### **6.1 Требования к хостинг-провайдеру:**

- Хостинг типа VPS;
- Поддержка доступа по SSH, защита от DDoS-атак, возможность бекапов;
- Наличие встроенного файлового менеджера
- Возможность бесплатного переноса сайта в дальнейшем

## Графический макет

В последнюю очередь на основе функциональных требований и предметной области был разработан дизайн веб-страниц. Так как система предназначена для проведения турниров преимущественно по волейболу, то и цвета для оформления были подобраны в соответствии с цветовой гаммой волейбольного мяча, к которой был добавлен чёрный цвет для прошедших соревнований (рисунок 2.1).



Рисунок 2 - Цветовая гамма

Шапка сайта выполнена в жёлтом цвете, и содержит ссылку перехода на главную страницу, кнопку входа и регистрации, либо же выхода, а также кнопку создание нового соревнования при судейских правах доступа.

Для удовлетворения требований основного функционала необходимы были страницы со списком соревнований, страница конкретного соревнования, страница создания соревнования, страницы входа и регистрации:

Страница со списком всех соревнований. Для упрощения структуры веб-ресурса было решено поместить анонсированные, текущие и прошедшие соревнования на одну страницу. Так как основной задачей информационной системы является анонсирование соревнований судьями и запись на них командами или их представителями, то основным элементом страницы является список анонсированных соревнований, занимающий всю ширину контент-блока страницы. Ниже идут текущие и прошедшие соревнования, занимающие по ширине по половине ширины страницы (рисунок 2.1);

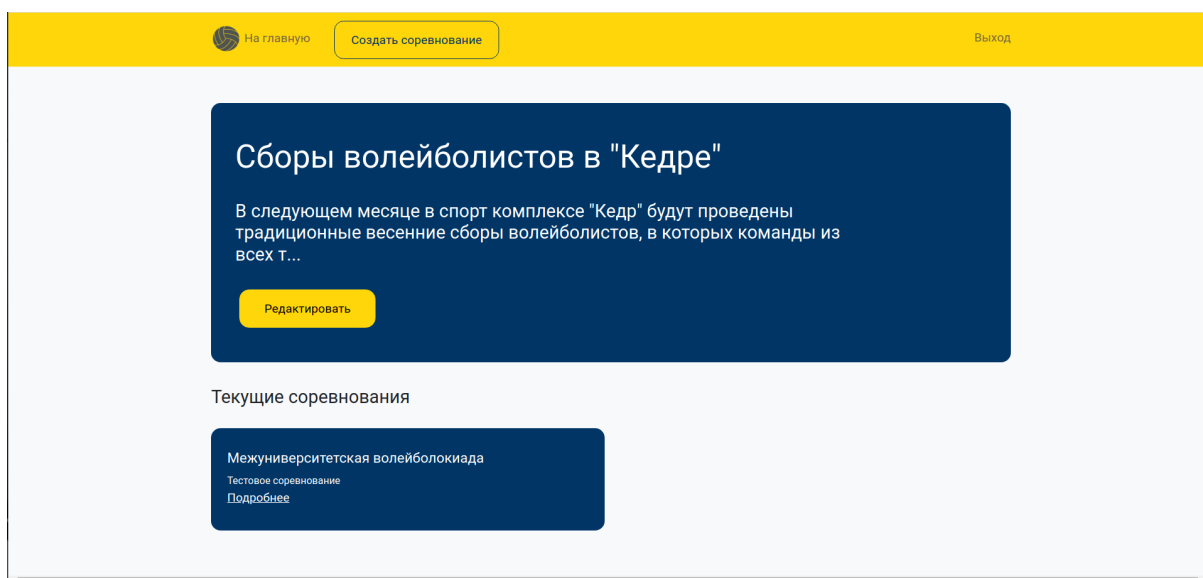


Рисунок 3 - Главная страница

Страница со списком прошедших соревнований. Когда прошедших турниров станет слишком много, предусмотрена веб-страница, реализующая их постраничный вывод. В её нижней части находится пейджинатор, позволяющий перемещаться по истории соревнований;

Страница турнира. В зависимости от статуса соревнования и прав доступа пользователя на странице будут присутствовать следующие виджеты:

- Основная информация. На странице находится постоянно, при судейском доступе предоставляет возможность редактирования данных соревнования. В данном виджете находится такая информация как заголовок, описание, наименование организации, проводящей соревнования, а также в зависимости от статуса соревнования время подачи заявок (для анонсированного), время следующего матча (для текущих) или время завершения соревнования (для завершённых). В случае анонсированного соревнования также предоставляет возможность редактирования количества команд-участниц.

- Виджет подачи заявки на участие в соревновании. Содержит форму для регистрации команды (название и описание), а также набор форм для ввода информации о каждом игроке команды. Данный виджет виден только когда соревнование анонсировано, а пользователь не имеет судейских прав доступа и ещё не подавал заявки на участие в данном соревновании (рисунок 2.2).



Рисунок 4 - Форма подачи заявок на участие

- Виджет со списком матчей данного соревнования. Включает в себя матчи, отсортированные по времени. Каждый матч имеет информация о командах в данном матче участвующих, о времени и месте проведения, а также игровой счёт. Каждый матч имеет возможность редактирования данных при судейских правах доступа, а статус матча помечается фоновым цветом - синим или чёрным (для предстоящих и прошедших матчей соответственно) (рисунок 2.3).

Рисунок 5 - страница текущего соревнования при судейском доступе

Страница создания соревнований. Доступна только при судейских правах доступа, имеет форму для создания соревнований, включающую следующие поля:

заголовок, описание, время подачи заявок, организатор и минимально необходимое количество команд (рисунок 2.4).

На главную      Регистрация      Вход

Заголовок      Заголовок

Организатор      Наименование организации, проводящей соревнования

Время подачи заявок      Время подачи заявок на участие

Мин. кол-во команд      4

Описание предстоящих соревнований

Описание

Создать

Рисунок 6 - Страница создания соревнования

## **Заключение**

В ходе выполнения проектного практикума был выбран объекта автоматизации - проведение спортивных соревнований. Первым делом была исследована предметная область и проведён обзор систем аналогичных той, разработка которой предстояла в контексте данной работы. Далее на основе полученных данных была произведена выработка ТЗ, в котором были обозначены сроки и этапы выполняемой работы, а также описаны функциональные и нефункциональные требования. Параллельно с этим был выбран фреймворк, предоставляющий функционал наиболее подходящий для решений поставленных задач.

Последним этапом практикума являлась разработка макета проекта, во время которой был создан и описан интерфейс системы.

Помимо разработанной системы результатом выполнения работы стало приобретение навыков написания ТЗ, анализа рынка аналогичного ПО и создание макета интерфейса системы.

**Список литературы**

1. Документация Django. [Электронный ресурс] - Режим доступа:  
<https://docs.djangoproject.com/en/4.0/>
2. Документация Symfony. [Электронный ресурс] - Режим доступа:  
<https://symfony.ru/doc/current/index.html>
3. Документация Flask. [Электронный ресурс] - Режим доступа:  
<https://flask-russian-docs.readthedocs.io/ru/latest/>