

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ  
Школа бакалавриата

## ОТЧЕТ

По проекту  
«Разработка Web-сервиса для автоматизации проведения хоккейных  
турниров»

по дисциплине «Проектный практикум»

Заказчик: Полозов А.А.

Куратор: Шестеров М.А.

Ассистент кафедры интеллектуальных информационных  
технологий ИнФО УрФУ

Студенты команды «Фанатики информатики»

Медведев Р.В.

Лытченко-Меткий О.Д.

Манакин А.С

Мищенко Н.К.

Невструев Р.О.

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. Работа команды проекта .....	5
1.1 Работа тимлида – Медеведева Романа Вячеславовича.....	5
1.2 Работа аналитика, дизайнера – Лытченко-Меткий Олег Денисович .....	7
1.3 Работа Backend-разработчика – Манакин Андрей Сергеевич .....	9
1.4 Работа Backend-разработчика – Мищенко Никита Константинович ....	10
1.5 Работа Frontend-разработчика – Невструев Роман Олегович .....	11
1.6. Планирование деятельности в ходе разработки .....	12
2. Анализ требований и постановка задач .....	13
2.1. Требования заказчика и пользователей.....	13
2.2. Формирование backlog и план действий .....	17
3. Анализ аналогов и обоснование разработки.....	19
4. Архитектура программного продукта.....	23
4.1. Структура и компоненты системы .....	23
4.2. Выбор архитектурного подхода и его обоснование .....	26
5. Методология и процесс разработки .....	27
5.1. Методологический подход к разработке.....	27
5.2. Промежуточное тестирование, выявленные ошибки и пути их устранения .....	29
ЗАКЛЮЧЕНИЕ .....	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	33
ПРИЛОЖЕНИЕ А (справочное) Техническое задание.....	34

## ВВЕДЕНИЕ

Современные тенденции в сфере цифровых технологий охватывают все больше аспектов повседневной жизни, включая спортивные мероприятия, где автоматизация процессов становится неотъемлемым условием эффективной организации. Особенно остро эта необходимость ощущается при проведении массовых детско-юношеских спортивных турниров, в которых участие принимают десятки и сотни участников с различным уровнем подготовки, возрастными особенностями и индивидуальными характеристиками. Одной из таких актуальных задач стало создание системы, способной обеспечить быстрое и удобное проведение хоккейных турниров с минимальным участием человека в процессах администрирования.

Актуальность проекта обусловлена прямым запросом со стороны заказчика, который на данный момент не располагает решением, соответствующим его требованиям. Повышение цифровой зрелости спортивной отрасли и растущая потребность в быстрой обработке большого объема информации делают такой сервис не просто востребованным, но и необходимым. Использование разработанного web-продукта позволит заказчику перейти на качественно новый уровень управления турнирами, сократить затраты на организацию, снизить количество ошибок, связанных с человеческим фактором, и обеспечить прозрачность и справедливость при формировании рейтингов.

Целью данного проекта является разработка web-сервиса, автоматизирующего все ключевые этапы организации хоккейных соревнований: от регистрации участников и формирования турнирной сетки до подсчета индивидуальных рейтингов и отправки уведомлений.

Для достижения поставленной цели были сформулированы следующие задачи:

1. создание личного кабинета игрока с возможностью загрузки документов и подтверждения согласия на обработку данных;

2. реализация системы автоматического подсчета рейтингов на основе микроматчей и активности команд;
3. формирование групп с учетом возрастных и профессиональных характеристик участников;
4. разработка интерфейса для судей с возможностью оперативного внесения результатов;
5. организация надежной системы оповещений и обратной связи, включая SMS-уведомления и доску объявлений.

Область применения разрабатываемого программного продукта охватывает прежде всего детские и молодежные хоккейные турниры, проводимые в локальном или региональном масштабе. Благодаря предусмотренной гибкости и масштабируемости, система может использоваться как небольшими хоккейными секциями (до 30 участников), так и более крупными организациями с развитой инфраструктурой и регулярными соревнованиями.

В результате реализации проекта планируется получить полноценный web-сервис, способный в автоматическом режиме обрабатывать регистрацию участников, формировать расписание микроматчей, рассчитывать рейтинги с учетом пола, возраста и уровня подготовки, а также уведомлять игроков и тренеров о важных событиях турнира. Судьи получают удобный инструмент для ввода результатов прямо с мобильных устройств, что обеспечит оперативность и актуальность данных. По завершении проекта заказчик получит эффективный инструмент, повышающий прозрачность, технологичность и удобство организации хоккейных турниров, а также минимизирующий административную нагрузку на персонал.

## **1. Работа команды проекта**

### **1.1 Работа тимлида – Медведева Романа Вячеславовича**

Работа тимлида в рамках проекта по разработке web-сервиса для автоматизации хоккейных турниров сыграла ключевую роль как в организации командной деятельности, так и в стратегическом управлении всеми этапами реализации проекта. На данной позиции выступал Медведев Роман Вячеславович, который не только выполнял функции технического координатора, но и брал на себя значительную часть задач, связанных с планированием, контролем сроков, распределением обязанностей между участниками и обеспечением качества конечного результата.

С первых этапов работы Роман активно участвовал в постановке задач и формировании технического задания. Он взаимодействовал с заказчиком, уточняя функциональные требования, выявляя приоритеты и формируя общее видение продукта. Это позволило задать четкий вектор всей работе команды и обеспечить фокусировку на действительно востребованных функциях, таких как автоматическая регистрация участников, система SMS-оповещений, алгоритмы формирования рейтингов и составление расписаний матчей.

Особое внимание тимлид уделял вопросам контроля версий и организации командной работы через систему Git. Благодаря его инициативе в команде была введена практика регулярных созвонов и стендапов, на которых обсуждался прогресс, возникающие сложности и принимались оперативные решения.

Особое место в деятельности Медведева Романа Вячеславовича занимало взаимодействие с заказчиком, которое он осуществлял на протяжении всего жизненного цикла проекта. Именно он выступал связующим звеном между технической командой и представителями заказчика, своевременно передавая требования, собирая обратную связь и координируя доработки в соответствии с актуальными потребностями. На этапе постановки задачи Роман организовал несколько встреч, в ходе которых

были уточнены ключевые сценарии использования системы, определены приоритетные функции и согласован формат отчетности о ходе выполнения работ. В процессе разработки он регулярно информировал заказчика о достигнутом прогрессе, демонстрировал промежуточные версии сервиса и принимал предложения по корректировке интерфейсов, логики расписания и системы рейтингов. Благодаря его коммуникационным навыкам и ответственному подходу удалось выстроить доверительные и конструктивные отношения, что способствовало стабильной и результативной работе над проектом.

На этапах тестирования и отладки Медведев координировал проведение внутренних проверок, а также участвовал в тестировании отдельных компонентов. Он анализировал выявленные баги, назначал ответственных за их устранение и следил за тем, чтобы ошибки устранялись своевременно и без ущерба для стабильности системы. Кроме того, Роман организовал работу с документацией – как технической, так и пользовательской, что особенно важно в контексте передачи проекта заказчику.

Также стоит отметить, что Роман проявил высокую степень ответственности и инициативности в тех ситуациях, когда команда сталкивалась с трудностями. Благодаря его усилиям, несмотря на то что ранее другие группы не смогли довести проект до завершения, текущая команда смогла наладить продуктивную работу и выйти на стабильный результат. Он обеспечил слаженность действий и общую дисциплину в коллективе, при этом создавая атмосферу взаимопомощи и уважения.

Таким образом, вклад Медведева Романа Вячеславовича в проект трудно переоценить. Его работа как тимлида обеспечила не только организационную и техническую устойчивость проекта, но и создала условия для продуктивной деятельности всей команды. Благодаря его компетенциям и вовлеченности, проект продвигался последовательно, уверенно и в соответствии с поставленными целями.

## **1.2 Работа аналитика, дизайнера – Лытченко-Меткий Олег Денисович**

В рамках проекта по разработке web-сервиса для автоматизации проведения хоккейных турниров важнейшую роль сыграл Лытченко-Меткий Олег Денисович, выполнявший функции аналитика и дизайнера. Его деятельность охватывала сразу несколько ключевых направлений, связанных с начальной проработкой концепции сервиса, анализом требований, проектированием пользовательских сценариев и визуальным оформлением интерфейса. Благодаря его усилиям была обеспечена логическая цельность, продуманность и удобство будущего программного продукта.

На первом этапе реализации проекта Олег сосредоточился на аналитической работе. Он детально изучал техническое задание, сформированное в результате общения тимлида с заказчиком, и проводил углублённый анализ задач, которые необходимо было реализовать. Особое внимание он уделил функциональной целесообразности каждого модуля, выявляя потенциальные сложности в реализации и предлагая оптимальные пути их решения. Олег тщательно проработал пользовательские потоки – от момента регистрации участника до финального этапа подведения итогов турнира. В результате была сформирована чёткая и структурированная логика взаимодействия пользователя с системой, учитывающая как удобство навигации, так и ограничения, связанные с возрастом и технической подготовкой участников и судей.

В качестве дизайнера Лытченко-Меткий отвечал за внешний вид интерфейсов и общий пользовательский опыт. Им была разработана система прототипов всех основных экранов: формы регистрации, личного кабинета, панели судьи, а также модуля расписания и рейтингов. В своей работе он придерживался принципов простоты и доступности, делая акцент на понятные элементы управления, читаемую типографику и лаконичную структуру экранов. Особое внимание он уделил адаптивности интерфейсов – так как

сервис должен был использоваться как на настольных компьютерах, так и на мобильных устройствах, в том числе судьями во время матчей.

В процессе разработки дизайна Олег также принимал участие в обсуждениях с заказчиком, демонстрируя макеты и получая обратную связь по визуальной части. Он быстро реагировал на замечания, гибко адаптируя графические элементы под предпочтения клиента, при этом не теряя из виду общий стиль и удобство использования. Важно отметить, что его работа не ограничивалась только эстетической составляющей – дизайнер последовательно следил за тем, чтобы каждое визуальное решение было обоснованным с точки зрения функциональности.

Кроме того, Лытченко-Меткий курировал оформление пользовательской документации, создавая инструкции и подсказки, которые планировались к интеграции в интерфейс сервиса. Это значительно повышало доступность проекта для конечных пользователей, особенно в части регистрации и навигации по функционалу. Его подход позволил создать образ сервиса, понятного даже тем, кто впервые сталкивается с подобными цифровыми решениями.

Таким образом, вклад Олега Денисовича в проект охватывал сразу несколько критически важных областей: от глубокой аналитики и продумывания пользовательских сценариев до визуального воплощения идей и формирования удобного и привлекательного интерфейса. Его системный подход, внимательность к деталям и способность соединять техническое с визуальным обеспечили высокий уровень качества и завершённости в проектных решениях.



### **1.3 Работа Backend-разработчика – Манакин Андрей Сергеевич**

Манакин Андрей Сергеевич принимал активное участие в создании серверной части web-сервиса и выполнял значительный объём backend-разработки самостоятельно. Его работа была направлена на реализацию ключевых функций системы, в том числе связанных с регистрацией участников, авторизацией, обработкой результатов матчей и формированием индивидуальных рейтингов.

Андрей занимался созданием и тестированием REST API, необходимого для взаимодействия клиентской части с сервером. Он разрабатывал серверную логику, отвечающую за приём и обработку данных от судей, регистрацию новых пользователей, а также формирование и обновление рейтингов с учётом пола, возраста и уровня подготовки участников. Особое внимание в его работе уделялось точности обработки информации и надёжности обмена данными между различными компонентами системы.

Кроме того, в рамках совместной работы с другим backend-разработчиком – Мищенко Никитой – Андрей взял на себя разработку части алгоритмов, в частности, по обработке и хранению информации о результатах микроматчей, а также их последующем учёте при построении финальных турнирных таблиц. Он реализовал часть логики распределения участников по категориям и обеспечил корректное взаимодействие серверных скриптов с базой данных.

В процессе разработки Манакин активно занимался отладкой и тестированием кода, участвовал в проверке работоспособности отдельных модулей, исправлял ошибки и вносил улучшения на основе результатов промежуточного тестирования. Его вклад стал важной составляющей стабильной и логичной работы backend-системы в целом.

## **1.4 Работа Backend-разработчика – Мищенко Никита Константинович**

Работа Мищенко Никиты заключалась в следующем:

1. Выбор стека и архитектуры;
2. Проектирование и реализация API (Регистрация и авторизация пользователей, работа с рейтингами и аналитикой, внесение и обработка результатов матчей);
3. Структура обработки алгоритмов (Распределение по возрастным категориям, автоматизация на сервере с помощью скриптов, формирование расписания, генерация матчей с учётом рейтингов, истории встреч)

В ходе проделанной работы были реализованы базовые алгоритмы работы веб-сервиса, построена и реализована логика обработки алгоритмов для генерации матчей, а также разработка REST API для функции меню регистрации и авторизации пользователей.

## **1.5 Работа Frontend-разработчика – Невструев Роман Олегович**

Невструев Роман Олегович выполнял задачи, связанные с разработкой клиентской части web-сервиса. Его работа заключалась в создании и настройке интерфейса, через который пользователи взаимодействуют с системой: участники турнира, судьи и администраторы. Он реализовал страницы регистрации и авторизации, личного кабинета участника, а также экранов с расписанием матчей и рейтингами.

В ходе работы Роман подключал интерфейс к серверной части через REST API, обеспечивая динамическое обновление данных и их корректное отображение. Также он отвечал за адаптацию страниц под разные устройства, в первую очередь под смартфоны, с которых судьи должны были вносить результаты матчей. Это потребовало продуманной структуры интерфейса и тестирования на различных разрешениях экранов.

Кроме основной вёрстки, Невструев занимался обработкой пользовательских действий и внедрением логики отображения данных в зависимости от статуса пользователя и текущего состояния турнира. Он оперативно реагировал на замечания от других участников команды, особенно при интеграции с backend и в процессе финального тестирования.

## 1.6. Планирование деятельности в ходе разработки

Планирование деятельности в ходе разработки веб-сервиса для автоматизации хоккейных турниров строилось на чётком распределении этапов работы и задач между участниками команды с учётом их специализации и опыта. В начале проекта была проведена совместная встреча, на которой определили ключевые цели, основные требования заказчика и составили первоначальный план действий. На основе этих данных сформировали backlog – список задач с расстановкой приоритетов и сроков выполнения.

В течение разработки команда регулярно проводила совещания для контроля прогресса, обсуждения сложностей и корректировки плана. Тимлид выполнял координационную роль, обеспечивая коммуникацию между backend- и frontend-разработчиками, дизайнером и аналитиком, что позволяло синхронизировать работу разных модулей и избежать пересечений. Важным элементом планирования стала интеграция работы всех специалистов, поскольку успешное взаимодействие между клиентской и серверной частью напрямую влияло на качество сервиса.

В процессе реализации учитывались результаты промежуточного тестирования и обратная связь от заказчика. При необходимости план корректировался, задачи перераспределялись для устранения выявленных проблем или внедрения новых требований. Такой подход позволил сохранить гибкость разработки и обеспечить своевременное выполнение ключевых этапов.

Таким образом, системное планирование и регулярный контроль позволили организовать работу команды эффективно, что способствовало успешному достижению поставленных целей и своевременному завершению проекта.

## **2. Анализ требований и постановка задач**

### **2.1. Требования заказчика и пользователей**

Требования заказчика и пользователей к разрабатываемому веб-сервису сформированы на основе технического задания, определяющего ключевые функции и цели системы. Основной задачей веб-сервиса является автоматизация проведения личных соревнований по хоккею среди спортсменов ДЮСШ «Спартаковец» с числом участников до 600 человек и несколькими игровыми площадками. В системе должна обеспечиваться регистрация участников с учётом их персональных данных, уровня подготовки, пола и возраста, а также интеграция смс-оповещений для информирования игроков о предстоящих матчах и подтверждении их участия (Приложение А).

Функциональные требования включают создание расписания игр с учётом рейтингов участников, автоматизацию формирования микроматчей, соблюдение игровых правил и обеспечение справедливого распределения по возрастным и половым категориям. Рейтинги участников формируются на основе результатов матчей, при этом учитывается возможность переноса рейтинга с предыдущих турниров. Предусмотрена система контроля корректности внесённых данных и предотвращения возможных манипуляций.

Для различных категорий пользователей определён дифференцированный доступ к функциям веб-сервиса. Посетители имеют ограниченный доступ к общедоступной информации, участники турниров – возможность регистрации, просмотра расписания и внесения изменений в результаты с подтверждением, а судьи – расширенные права на внесение результатов без необходимости подтверждения. Такая система разграничения прав доступа обеспечивает безопасность и упорядоченность работы сервиса.

В технических требованиях уделено внимание удобству использования и внешнему виду интерфейса. Дизайн должен быть выполнен в светлых тонах

с контрастными элементами, исключая избыточные анимации и тёмные агрессивные цветовые решения [1]. Основные разделы сайта должны быть доступны с главной страницы, при этом текстовая информация ограничена для повышения удобства восприятия.

Техническое задание также включает спецификации по формату турниров, правилам формирования микроматчей, особенностям учёта рейтингов в зависимости от возраста и позиции игрока, а также интеграцию локального сервиса для проведения небольших соревнований. Особое внимание уделено возможности загрузки медицинских заключений и документов, а также поддержке системы стоп-листа для участников с нарушениями.

Для обеспечения корректного проектирования информационной системы необходимо детально рассмотреть потребности и сложности, с которыми сталкиваются ключевые категории пользователей: организаторы турниров, тренеры, а также игроки и их родители. Анализ их взаимодействия с текущими процессами выявляет широкий спектр неудовлетворённых требований, обуславливающих необходимость создания специализированного цифрового решения.

Организаторы (или администраторы) представляют центральное звено в процессе проведения соревнований. В их функциональные обязанности входит разработка расписания матчей, формирование групп и турнирных таблиц, коммуникация с участниками, контроль за ходом соревнования и документальное сопровождение результатов. На текущий момент данные процессы в подавляющем большинстве случаев реализуются вручную, с использованием разрозненных и неудобных инструментов – Excel-таблиц, бумажных журналов, мессенджеров и стандартных почтовых или SMS-рассылок.

Одной из ключевых проблем является ручное составление расписаний. Это не только трудоёмкий и времязатратный процесс, но и источник высокой вероятности ошибок, особенно при изменениях состава участников или сбоях

в ходе турнира. Ещё одной существенной проблемой является отсутствие централизованной системы учёта рейтингов игроков. Распределение участников по группам часто происходит субъективно и не всегда учитывает уровень подготовки, что влияет на справедливость и мотивацию игроков. Коммуникация между организаторами и участниками турнира также сопряжена с трудностями: информация часто доходит с опозданием или в искажённой форме. Дополнительную сложность создаёт фрагментарность инструментов – для каждой задачи (регистрация, рассылка, рейтинги) используются отдельные нерегламентированные и несвязанные между собой сервисы, что препятствует формированию единой информационной среды.

Тренеры, в свою очередь, выполняют не только функцию подготовки игроков, но и участвуют в организационных и аналитических аспектах турниров. Им необходим постоянный доступ к актуальному расписанию матчей, оперативная информация о результатах и объективные данные о прогрессе подопечных. На практике же тренеры вынуждены полагаться на личные записи, вести коммуникацию в неофициальных группах и чатах, а также запрашивать данные у судей и организаторов вручную. Такие подходы снижают эффективность анализа, затрудняют долгосрочное планирование и ведут к потере данных между турнирами.

Особенно остро стоит проблема отсутствия стабильной и прозрачной системы рейтинга игроков. В результате становится невозможным корректно оценивать развитие спортсменов и сравнивать результаты в рамках сезона или между турнирами. Это создаёт ощущение субъективности и неравенства, что может негативно сказаться как на мотивации спортсменов, так и на уровне доверия к организаторам соревнований.

Третьей ключевой категорией пользователей являются непосредственно участники турниров – дети и подростки в возрасте от 7 до 16 лет и их родители. Для данной группы особое значение имеет доступность, интуитивность и наглядность цифрового взаимодействия. Однако в существующих условиях участники часто оказываются в информационном вакууме. О расписании

матчей, результатах и изменениях они узнают, как правило, от тренеров либо из фотографий бумажных протоколов, размещённых в чатах. При этом информация может быть представлена не в полном объёме или поступать с опозданием, особенно при форс-мажорных изменениях в ходе турнира.

Регистрация на турниры и подтверждение участия также сопряжены с трудностями. В ряде случаев этот процесс осуществляется вручную или через устные договорённости, что порождает риски дублирования, ошибок и недопонимания. Личностная вовлечённость и интерес к собственному участию со стороны юных спортсменов также остаются на низком уровне: отсутствуют цифровые профили, истории достижений, системы уведомлений и элементы геймификации, которые могли бы стимулировать активное участие и ощущение принадлежности к соревновательному процессу.

В связи с изложенным, можно сделать вывод о наличии объективной потребности в комплексной цифровой системе, способной интегрировать процессы планирования, администрирования, информирования и аналитики в рамках детско-юношеских хоккейных турниров. Решение должно учитывать специфику каждой группы пользователей и стремиться к автоматизации, унификации и доступности всех функциональных компонентов.



## 2.2. Формирование backlog и план действий

Формирование backlog и план действий стали основой для организации работы над проектом. В процессе разработки первоначально проведён тщательный анализ технического задания, на основании которого были выделены ключевые функциональные модули и задачи, необходимые для реализации веб-сервиса. Каждая задача получила конкретные сроки выполнения, что позволило структурировать работу команды и обеспечить контроль за прогрессом.

Backlog составлен с учётом приоритетности и логической последовательности этапов. Он включает как базовые компоненты, например, создание базы данных и API, так и модули, обеспечивающие пользовательский интерфейс и взаимодействие – регистрацию, личный кабинет, расписание, судейский модуль, а также элементы, связанные с логикой формирования рейтингов и рассылкой уведомлений. Такой подход обеспечивает поэтапную реализацию проекта с возможностью своевременной проверки и корректировки.

План действий чётко распределяет сроки выполнения основных этапов, начиная с уточнения технического задания и подготовки окружения для разработки, и завершая тестированием, исправлением ошибок и релизом продукта. Благодаря этому обеспечивается прозрачность процессов, что способствует эффективному управлению ресурсами и соблюдению установленных сроков. Каждый этап логично вытекает из предыдущего, что минимизирует риски и позволяет своевременно выявлять и устранять проблемы (таблица 1).

Таблица 1 – План работ (backlog)

Этап	Сроки
1. Уточнение ТЗ	18-26 марта
2. Подготовка окружения	27 марта-7 апреля
3. Реализация БД и API	8-14 апреля
4. Регистрация и личный кабинет	15-20 апреля
5. Расписание и распределение	21-26 апреля

Продолжение таблицы 1

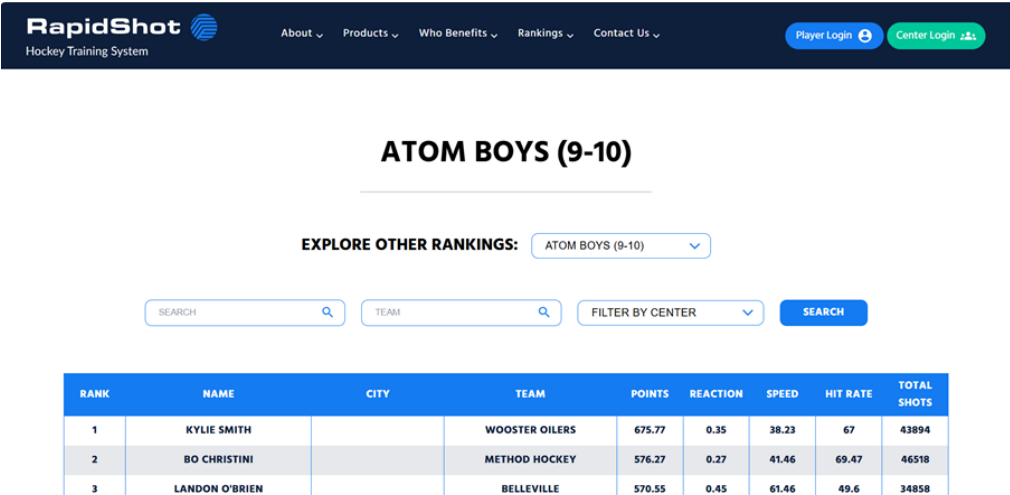
6. Судейский модуль	27 апреля-3 мая
7. Рейтинг и логика матчей	4-9 мая
8. Рассылка SMS	10-12 мая
9. Доска объявлений	13-15 мая
10. UI/UX + адаптация	16-21 мая
11. Тестирование и багфикс	22-27 мая
12. Завершение и релиз	28-31 мая

Таким образом, формирование backlog и детальный план действий обеспечивают системный подход к разработке веб-сервиса, способствуя согласованной работе команды и достижению поставленных целей в запланированные сроки.

### 3. Анализ аналогов и обоснование разработки

Существующие решения в области управления хоккейными соревнованиями и анализа игровых показателей представлены разными платформами, каждая из которых ориентирована на определённые задачи и аудитории. Анализ подобных сервисов позволяет выявить их преимущества и недостатки, что служит основой для обоснования необходимости разработки специализированного веб-сервиса для турниров ДЮСШ «Спартаковец».

Первым аналогом выступает сервис RapidShot Rankings, который фокусируется на предоставлении рейтингов хоккеистов по возрастным категориям (рисунок 1). Основой для формирования рейтингов здесь служат данные, получаемые с тренажёров, что позволяет детально анализировать навыки каждого игрока и отслеживать динамику его развития. Среди сильных сторон данного решения выделяется чёткая и понятная система рейтинга, а также удобная визуализация результатов, что способствует мотивации и развитию спортсменов. Однако RapidShot Rankings не предназначен для организации и управления соревнованиями [3]. В сервисе отсутствуют возможности регистрации участников, формирования расписания матчей и автоматизации турниров, что ограничивает его применение исключительно индивидуальным анализом вне соревновательного контекста.



**RapidShot** Hockey Training System

About Products Who Benefits Rankings Contact Us

Player Login Center Login

## ATOM BOYS (9-10)

EXPLORE OTHER RANKINGS: ATOM BOYS (9-10)

SEARCH TEAM FILTER BY CENTER SEARCH

RANK	NAME	CITY	TEAM	POINTS	REACTION	SPEED	HIT RATE	TOTAL SHOTS
1	KYLIE SMITH		WOOSTER OILERS	675.77	0.35	38.23	67	43894
2	BO CHRISTINI		METHOD HOCKEY	576.27	0.27	41.46	69.47	46518
3	LANDON O'BRIEN		BELLEVILLE	570.55	0.45	61.46	49.6	34858

Рисунок 1 – Первый аналог – сервис RapidShot Rankings

Вторым рассмотренным аналогом является платформа HockeyShift, которая ориентирована на управление хоккейными клубами и лигами (рисунок 2). Этот сервис предлагает расширенный функционал, включающий управление расписанием игр, составами команд и сбором статистики матчей. Такой комплексный подход обеспечивает удобство в администрировании лиг и клубов, позволяет вести учёт результатов и контролировать процесс проведения соревнований [2]. Вместе с тем, HockeyShift характеризуется сложностью настройки и высокой стоимостью, что может стать барьером для использования в малых детских и локальных турнирах. Кроме того, отсутствие русскоязычного интерфейса снижает удобство эксплуатации для российских пользователей, а недостаточная адаптация под формат детских соревнований ограничивает его практическую применимость для ДЮСШ и аналогичных организаций.

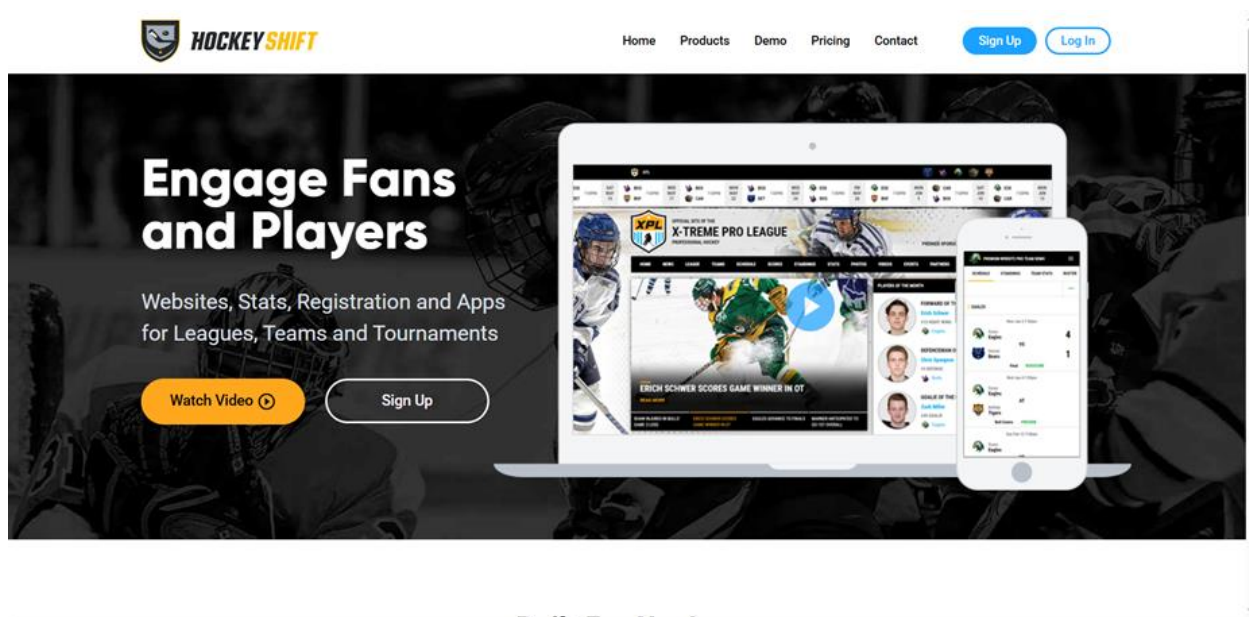


Рисунок 2 – Второй аналог – сервис HockeyShift

Оценка данных аналогов показывает, что на рынке отсутствует специализированное решение, которое одновременно обеспечивало бы автоматизацию турниров детско-юношеских спортивных школ с учётом их специфики, доступный интерфейс на русском языке, гибкую систему формирования расписаний и рейтингов, а также интеграцию с SMS-оповещениями участников. Это подчеркивает актуальность и необходимость

создания собственного веб-сервиса, который будет учитывать особенности проведения локальных турниров, обеспечивать простоту использования и высокую функциональность, необходимую для эффективного управления соревнованиями и анализа игровых результатов.

Проведение сравнительного анализа существующих аналогов позволяет более чётко обозначить уникальные характеристики разрабатываемого веб-сервиса и выявить те функциональные аспекты, которые требуют особого внимания в процессе проектирования. В данном случае анализируются две платформы – RapidShot Rankings и HockeyShift. Обе системы представляют различные подходы к цифровому сопровождению хоккейной деятельности, однако они существенно различаются по своим целям, возможностям и предполагаемой аудитории.

Сервис RapidShot Rankings ориентирован преимущественно на индивидуальную работу с игроками. Его основная задача – сбор и обработка технических показателей, полученных при помощи специализированных хоккейных тренажёров. Это делает систему особенно полезной в рамках тренировочного процесса, где важно отслеживать динамику развития конкретных навыков, таких как сила броска, точность или скорость реакции. Однако, несмотря на развитую систему рейтинга и качественную визуализацию данных, данный сервис абсолютно не приспособлен к решению задач по организации соревновательной деятельности. Он не предусматривает механизмов регистрации участников турниров, не поддерживает формирование расписания матчей, не ведёт турнирную статистику и не автоматизирует проведение игр. В связи с этим использование RapidShot Rankings ограничено индивидуальной аналитикой, что делает его недостаточно универсальным.

Платформа HockeyShift, в отличие от RapidShot, предназначена именно для комплексного управления турнирами и лигами. Она предоставляет возможности по созданию расписаний, распределению игроков по командам, ведению матчевой статистики и управлению различными аспектами

хоккейных соревнований. Это делает её привлекательной для спортивных организаций, особенно на уровне профессиональных или полупрофессиональных команд. Однако важным ограничением HockeyShift является его недостаточная адаптированность под потребности детско-юношеского хоккея.

Во-первых, интерфейс полностью англоязычный, что усложняет использование платформы в русскоязычных регионах без предварительной локализации. Во-вторых, платформа требует высокой степени технической подготовки от администраторов и тренеров, так как содержит множество сложных параметров и настроек. Кроме того, использование сервиса предполагает значительные финансовые затраты, что может оказаться неприемлемым для спортивных школ с ограниченным бюджетом. Наконец, HockeyShift ориентирован на командные соревнования и не предусматривает полноценной реализации личных турниров.

Сравнительный анализ показывает, что ни один из рассмотренных аналогов не охватывает в полной мере те функции, которые ставятся в качестве задачи при разработке собственного веб-сервиса. RapidShot Rankings полезен для индивидуального развития, но не решает задач турнирной логистики. HockeyShift закрывает задачи администрирования турниров, но остаётся громоздким, дорогостоящим и не приспособленным к формату локальных личных соревнований среди детей. Таким образом, разрабатываемая система должна стать интеграцией лучших аспектов обоих подходов, при этом дополняясь уникальными модулями – такими как рейтинг по микроматчам, автоматическая генерация расписаний, фильтрация по возрасту и уровню подготовки, регистрация и подтверждение участия через SMS, а также полноценная работа с индивидуальными профилями и медицинскими данными игроков. Подобная комплексность и целевая адаптация делают обоснованным создание нового продукта, специально ориентированного на детско-юношеский хоккейный турнир локального уровня.

## **4. Архитектура программного продукта**

### **4.1. Структура и компоненты системы**

Система автоматизации проведения хоккейных турниров выстроена в соответствии с классической трёхзвенной архитектурой, в которой ясно разграничены уровни хранения данных, обработки бизнес-логики и представления пользовательского интерфейса.

На уровне хранения данных реализована реляционная модель, включающая сущности турниров, пользователей, команд и матчей. Таблица, содержащая сведения о турнирах, фиксирует параметры соревнования: идентификатор, наименование, формат проведения, географическое расположение и календарные рамки (даты начала и завершения). Пользовательские аккаунты представлены отдельной таблицей, где регистрируются личные данные (имя, фамилия, отчество, дата рождения), контактная информация, а также атрибутивные характеристики участника – роль (игрок, судья, администратор), уровень мастерства и рейтинговые показатели. Модель команд строится на связи «многие-ко-одному» с турниром посредством внешнего ключа, что позволяет однозначно привязать каждый состав к конкретному соревнованию. При этом участие игроков в командах оформлено в виде вспомогательной таблицы «team\_player», обеспечивающей двустороннюю связь «многие-ко-многим» между записями пользователей и команд, что важно для поддержки сценариев смены состава и участия одного спортсмена в нескольких турнирах. Для учёта негровых ролей (судей и организаторов) разработана дополнительная таблица, связывающая пользователей и турниры вне привязки к командам. Отдельное внимание уделено структуре таблицы матчей: она содержит ссылки на оба участвующих коллектива, хранит результаты по каждому периоду игры, а также фиксирует текущее состояние поединка и временные метки.

Бизнес-логика инкапсулирована в серверном приложении на базе фреймворка Java Spring. Архитектурная схема разделена на слои

контроллеров, сервисов и репозиторийев. Контроллеры отвечают за приём HTTP-запросов, их первичную валидацию и трансформацию входящих данных в объекты доменной модели. Сервисный слой реализует ключевые алгоритмы: генерацию турнирных сеток, расчёт статистических показателей, обработку результатов матчей и управление правами доступа. Репозитории, построенные на Spring Data JPA, обеспечивают прозрачное взаимодействие с базой, автоматически переводя операции CRUD в SQL-запросы. Благодаря декларативной конфигурации транзакций достигается целостность и согласованность записей при массовых обновлениях результатов и составе участников (рисунок 3).

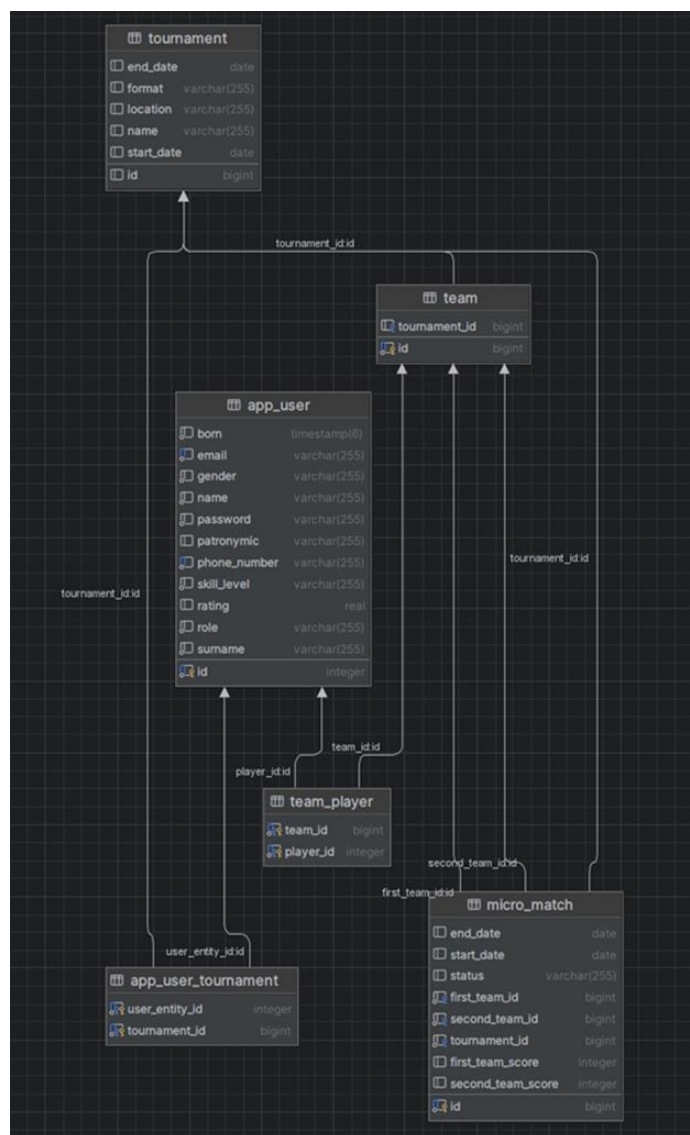


Рисунок 3 – схема базы данных



Интерфейс представления разработан на React с применением TypeScript и инструментов сборки Vite. Компонентная структура приложения модульно выстраивает функциональные блоки: формы создания и редактирования турниров, таблицы просмотра расписания матчей, карточки команд и профилей игроков. В качестве транспортного протокола используется REST-архитектура с JSON-обменом, что упрощает интеграцию сторонних клиентов и тестирование. Vite обеспечивает мгновенный HMR-обновление при разработке и эффективную минификацию скриптов в продакшн-сборке, что снижает время загрузки и повышает отзывчивость интерфейса.

Межуровневое взаимодействие осуществляется по схеме:

1. Пользовательский запрос инициируется в браузере и передаётся React-компонентом к серверу посредством HTTP-метода (GET, POST, PUT, DELETE);
2. Серверная сторона, пройдя контроллерный и сервисный слой, формирует соответствующий запрос к СУБД через репозиторий и получает результат;
3. Полученные данные сериализуются в формат JSON и возвращаются клиенту;
4. React-приложение обновляет представление, обеспечивая динамическое отображение актуальной информации.

Отделение логики хранения, обработки и отображения данных не только упрощает сопровождение и развитие системы, но и позволяет при необходимости масштабировать каждый из уровней независимо, интегрировать новые модули (например, аналитический движок или мобильный клиент), а также гарантировать высокую отказоустойчивость и безопасность при одновременной многопользовательской работе.

## 4.2. Выбор архитектурного подхода и его обоснование

В качестве серверного фреймворка выбран Java Spring, поскольку его модульная структура контроллер-сервис-репозиторий даёт возможность ясно формализовать бизнес-правила, упрощает реализацию транзакционного управления и предоставляет развитый инструментарий для обработки REST-запросов. Кроме того, Spring Data JPA автоматизирует работу с объектно-реляционным отображением, что сокращает объём шаблонного кода и минимизирует риск ошибок при формировании SQL-запросов вручную.

В фронтенд-слое предпочтение было отдано React и TypeScript в связке с Vite [6]. Компонентная модель React позволяет инкапсулировать визуальную и поведенческую логику каждого элемента интерфейса, а статическая типизация TypeScript повышает надёжность клиентского кода за счёт раннего обнаружения несоответствий типов. Инструмент сборки Vite гарантирует молниеносную перезагрузку модулей во время разработки и эффективную оптимизацию бандла для продакшн-окружения/

Таким образом, сочетание реляционной базы данных, зрелого серверного фреймворка и современного клиентского стека делает архитектуру системы масштабируемой, отказоустойчивой и удобной для расширения функциональности. Каждая из используемых технологий выбиралась с учётом её зрелости, широты сообщества и наличия инструментов для обеспечения безопасности и мониторинга, что в совокупности повышает надёжность и управляемость разрабатываемого сервиса.

## **5. Методология и процесс разработки**

### **5.1. Методологический подход к разработке**

Процесс разработки web-сервиса для автоматизации проведения хоккейных турниров был организован с учётом требований, сформулированных на этапе взаимодействия с заказчиком. На начальной стадии проекта состоялась встреча, в ходе которой заказчик определил ключевые функциональные блоки, обозначил приоритеты и обозначил ожидаемый результат. Эти вводные легли в основу формирования целей разработки, а также способствовали определению набора минимально жизнеспособных функций (MVP), которые необходимо было реализовать в первую очередь.

Для эффективного управления проектом был выбран итерационный методологический подход, сочетающий элементы Scrum и канбан-организации задач. Основой планирования выступала методология Scrum, позволяющая разбить весь цикл работ на относительно короткие итерации (спринты) продолжительностью от одной до двух недель. В рамках каждого спринта проводились планёрки, где формировался список задач на текущий цикл, и устраивались созвоны для синхронизации участников команды. Такой подход обеспечил гибкость и возможность оперативного реагирования на изменяющиеся требования или обратную связь от заказчика.

Параллельно с элементами Scrum применялась визуальная система управления задачами в формате канбан-доски, реализованная в системе Kaiten. Каждая задача (функциональная, аналитическая, инфраструктурная или связанная с дизайном) фиксировалась как отдельная карточка, перемещаемая по стадиям жизненного цикла: «Создано», «В работе», «На ревью», «Готово». Благодаря этому визуальному представлению команда в любой момент могла оценить текущее состояние проекта, степень загрузки участников и определить потенциальные узкие места в процессе. Также это позволило

равномерно распределять задачи между ролями (разработчики, аналитики, дизайнеры и др.) и отслеживать динамику выполнения работы по каждому направлению.

Важно отметить, что при наличии базовой структуры Scrum команда не придерживалась жёстких рамок полной agile-итеративности, а адаптировала методологию под особенности проекта. Например, фокус смещался с ежедневных стендапов на работу по результату, а задачи могли передвигаться между этапами вне зависимости от завершения спринта, если того требовала ситуация [5].

Таким образом, совмещение структурированной итерационной методики (Scrum) с визуальным контролем процессов (канбан) позволило обеспечить управляемость разработки, сохранить прозрачность выполнения задач, а также достичь высокой степени согласованности внутри команды при постоянной связи с заказчиком.

## **5.2. Промежуточное тестирование, выявленные ошибки и пути их устранения**

На протяжении всего цикла разработки web-сервиса проводилось промежуточное тестирование, целью которого было выявление функциональных и логических ошибок до ввода системы в эксплуатацию. Для тестирования использовались тестовые данные, предоставленные заказчиком, соответствующие реальным сценариям использования: информация о турнирах, командах, игроках и расписании матчей. Это обеспечило максимальное приближение проверок к реальным условиям эксплуатации системы.

Первоначально проверялась корректность работы базовых операций: регистрация пользователей, создание турниров, формирование команд и запись матчей. На этом этапе была обнаружена одна из первых критических ошибок – невозможность регистрации пользователя через форму на клиентской стороне. После детального анализа выяснилось, что ошибка была связана с некорректной сериализацией данных: информация, поступающая от клиента, в процессе трансформации теряла часть обязательных полей, вследствие чего в базе данных создавалась неполная или неконсистентная запись [4]. Проблема была устранена путём переписывания серверного метода обработки регистрационной формы: логика валидации входящих данных была усилена, а структура запроса приведена к строгому соответствию схеме модели в базе.

Другой значимой проблемой стал некорректный пересчёт рейтинга игроков после завершения матчей. Согласно бизнес-логике, каждый матч должен влиять на рейтинг участников в зависимости от результата игры и индивидуальной статистики игрока. Однако при проверке выяснилось, что по завершении матчей значения рейтингов в базе не изменялись. Причина заключалась в том, что логика пересчёта рейтинга находилась вне

транзакционного блока и вызывалась асинхронно, в результате чего данные в момент вызова функции ещё не были окончательно зафиксированы. Решение было реализовано следующим образом: пересчёт рейтинга был встроен непосредственно в метод фиксации результатов матча, а сама операция была обёрнута в транзакцию, чтобы гарантировать атомарность обновлений. Это обеспечило согласованное состояние базы после каждого изменения счёта.

Также в ходе тестирования были выявлены несостыковки в логике назначения матчей, особенно при формировании расписания для турниров с нечётным количеством команд. Система не предусматривала автоматического добавления «технических» выходных (бай-раундов), в результате чего происходила генерация дублирующихся матчей либо одновременное назначение одной команды в два параллельных поединка. В рамках согласования с заказчиком было принято решение ввести отдельный алгоритм, проверяющий целостность расписания и при необходимости автоматически добавляющий холостые туры для выравнивания сетки. Этот алгоритм был интегрирован в backend-сервис генерации матчей и прошёл успешную проверку на множестве тестовых кейсов.

В целом, благодаря регулярному промежуточному тестированию и активному взаимодействию с заказчиком удалось своевременно выявлять и устранять ошибки, не допуская их накопления на поздних стадиях разработки. Такой подход обеспечил высокое качество и предсказуемость работы ключевых компонентов системы.

## ЗАКЛЮЧЕНИЕ

Реализация проекта по разработке web-сервиса для автоматизации проведения хоккейных турниров продемонстрировала, что поставленные на начальном этапе цели были достигнуты в полном объёме. Разработанный программный продукт в своей текущей версии способен обеспечить заказчику не только выполнение всех ключевых функций, предусмотренных техническим заданием, но и предоставить качественно новый уровень организации соревнований. Проведённая работа над сбором, анализом и формализацией требований позволила точно отразить потребности конечных пользователей – организаторов, судей и участников турниров – в функциональности системы. Это выразилось в точной интеграции механизмов, таких как автоматическое распределение игроков, расчёт рейтингов, оперативное внесение результатов судьями и централизованная система уведомлений.

Качество программного продукта подтверждено результатами промежуточного и итогового тестирования. Несмотря на выявленные ошибки, касающиеся, в основном, отображения информации и валидации данных, они не повлияли критически на стабильность или функциональность системы. Более того, их устранение в ходе отладки позволило не только повысить надёжность сервиса, но и улучшить пользовательский опыт за счёт оптимизации взаимодействия с интерфейсом. Особое внимание в процессе разработки уделялось вопросам отказоустойчивости и консистентности данных, что также повысило уровень доверия к программному продукту со стороны заказчика.

Анализ проделанной работы позволяет утверждать, что созданная архитектура сервиса закладывает прочную основу для его дальнейшего развития. В системе изначально заложена гибкость, позволяющая масштабировать её под турниры различного уровня и формата. В перспективе

возможна реализация интеграции с внешними платформами (например, календарями, платёжными системами или сторонними CRM), а также развитие системы аналитики на основе накопленных данных о матчах и игроках. Также обоснованной видится задача разработки мобильного приложения, синхронизированного с основным сервисом, что ещё больше повысит доступность продукта для конечных пользователей.

Таким образом, проект показал высокую степень реализуемости на практике и соответствие современным требованиям в сфере цифровизации спортивных мероприятий. Созданный сервис отвечает текущим ожиданиям заказчика и пользователей, демонстрирует устойчивую работу, а также обладает потенциалом к масштабированию и функциональному расширению в будущем.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лаурен, Дж. HTML и CSS. Разработка и дизайн веб-сайтов / Дж. Лаурен ; пер. с англ. – СПб. : Питер, 2020. – 512 с.
2. Официальный сайт HockeyShift [Электронный ресурс] / HockeyShift. – <https://www.hockeyshift.com/> (дата обращения: 20.03.2025).
3. Официальный сайт RapidShot Rankings [Электронный ресурс] / RapidShot Rankings. – URL: <https://www.rapidshot.com/rankings/> (дата обращения: 20.03.2025).
4. Соммервилл, И. Инженерия программного обеспечения : пер. с англ. / И. Соммервилл. – 10-е изд. – М. : Вильямс, 2016. – 832 с.
5. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер ; пер. с англ. – М. : Вильямс, 2013. – 576 с.
6. Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган ; пер. с англ. – 6-е изд. – СПб. : Питер, 2020. – 1072 с.

## **ПРИЛОЖЕНИЕ А**

### **(справочное)**

#### **Техническое задание**

##### **1. Назначение и цели создания веб-сервиса.**

###### **1.1. Назначение веб-сервиса.**

Веб-сервис должен использоваться при проведении личных соревнований по хоккею среди хоккеистов ДЮСШ «Спартаковец». Предполагаемое число участников турнира – до 600 при наличии порядка 2-3 хоккейных площадок.

###### **1.2. Цели создания веб-сервиса**

Цель веб-сервиса – автоматизированная система проведения соревнования по хоккею, реализуемая через смс-оповещение участников в ходе турнира, предоставляющая им доступ к регистрации на турниры через Интернет, информацию о расписании игр и об индивидуальных игровых рейтингах.

##### **2. Описание работы веб-сервиса.**

При регистрации абитуриенты/студенты заполняют ФИО, номер сотового телефона, мейл, пол, возраст, уровень подготовки в данном виде спорта (разряд, мс, любитель, новичок).

За день до турнира им приходит смс оповещение о том, что школьник/абитуриент/студент участвует в турнире и должен подтвердить свое участие.

Все полученные рейтинги всех участников сохраняются в личном кабинете и при повторном участии используется последнее значение их рейтинга.

В зависимости от заявленного пола, возраста и уровня подготовки формируется исходный рейтинг. При этом, если человек уже участвовал в предыдущем турнире и в нем получил рейтинг, то его прежний рейтинг переносится с предыдущего турнира на новый.

Формат турнира предполагает соблюдение соотношения игроков на поле, правил хоккея, однако в отличии от обычной игры проводится серия микроматчей («замен»), продолжительностью 1 минута.

Формируется расписание на ближайший тур. Правила формирования состава микроматчей:

- а) Список рейтинга просматривается сверху вниз;
- б) Если участники уже играли, то они не играют во второй раз;
- в) Ищется соперник с максимально близким значением рейтинга.

Участнику заблаговременно приходит смс оповещение – место соревнования, время начала микроматча.

Судья заносит результат поединка в виде баланса забитых и пропущенных голов (З И П). Баланс переводится в рейтинг. Судьи имеют возможность вносить результаты со своего смартфона.

По итогам 1 тура формируется рейтинг лист и расписание на 2 тур.

При задержке игры рейтинг считается, исходя из тех результатов, которые появились в системе в предполагаемое время окончания матча, формируется новое расписание игр.

В первой встрече всем участникам присваивается средний рейтинг – обычно 2200. Соответственно, средний по контрольной игре – такой же. Потом его придется считать до игры, исходя из состава участников. Новичкам продолжаем присваивать 2200. Можно на рейтинг-лист занести этот (к+1) результат. Но обычно тренера интересуют данные по игроку за последний месяц (последние 5-10 результатов). Тогда после к+1 игры предыдущий к-й рейтинг изменится так, чтобы дать усредненную оценку в этом интервале:

$$Rti(k+1) = Rti(k) + \left(\frac{1}{10}\right) \times (Rti(k+1) - Rti(k)) \quad (1)$$

$$\left\{ \begin{array}{l} \delta_I \times Rt_I + \sum_{\substack{J=1 \\ I \neq J}}^N \delta_{I\text{напрм}} \times Rt_J - \sum_{\substack{J=1 \\ I \neq J}}^N \delta_{I\text{con}} \times Rt_J = \Delta_I, \text{ где } i = \underline{1, \dots, N}; \\ Rt_{\text{срел}} = \left(\frac{1}{n}\right) \times \sum_{I=1}^n Rt_I. \end{array} \right. \quad (2)$$

В приведенной формуле величина  $\delta$  – доля З и П  $i$ -м игроком от общей суммы этого игрока, которая приходится на его противостояние с  $j$ -м игроком. Величина  $\Delta i = 5500 \times (З - П) / (З + П)$  – баланс  $i$ -го игрока. Также необходимо  $n+1$  уравнение, задающее средний рейтинг участников контрольной игры. Иначе СЛУ не имеет решения.

По итогам турнира из общего списка рейтинга необходимо сформировать также частные списки методом сепарации: рейтинг-лист юношей, рейтинг-лист девушек, по возрастам

Предусмотреть стоп-лист, когда участнику может быть отказано в участии в турнире в связи с ранее сделанными проступками.

По возможности, сделать проверку на возможные искажения результатов. Для этого у любого участника можно подставить значения в его уравнение СЛУ и показать равенство левой и правой части.

Для детей в возрасте до 10 лет включительно использовать формат игры 4 на 4 без вратаря. Для более старших по возрасту – обычный формат 5 на 5 с участием вратарей.

Отдельно на сайт переносится web-сервис <http://personal-champ.ra-first.com/championships>, который позволяет провести личное первенство локально для небольшого числа участников (до 30 человек). Сделать по аналогии. Рейтинги такого локального соревнования должны заноситься в рейтинг лист игрока и личный кабинет. Рейтинг такого соревнования определяется по среднему рейтингу его участников. Подсчет результатов для самих участников идет также по сумме забитых и пропущенных ими мячей. При генерации игр, микроматчей добиваемся максимально возможного равенства по балансу партнеры/соперники. Если игрок вынужден играть две смены подряд, то расписание оставляем без изменений. Если игрок вынужден играть 3,4 и т.д. микроматчей подряд, то в расписании предусматриваем отдых 1 минуту между 2-3 микроматчами, 3-4 и т.д.

Допускается участие в микроматче игроков с разницей в возрасте \_\_\_\_ до 14 лет - 1 год, после – 2 года, после 16 лет нет ограничения

Если игрок выбыл до окончания турнира, то судья имеет возможность его исключить из генерации составов микроматчей.

В личный кабинет участник вносит скан медицинского заключения, удостоверение личности, согласие на обработку данных, персональные данные: мейл, фио, телефон, профильная позиция, профессиональный хоккеист

Выделить цветом игроков, претендующих на контракт.

При регистрации все игроки Спартаковца бесплатно, остальные платно.

При этом должен быть запрос на бесплатное участие.

Микроматч длится 1 минуту и 30 секунд пауза.

Сдвигка расписания судьей.

Возможность коррекции среднего рейтинга

3. Требования к веб-сервису

2.1. Требования к разграничению доступа

Информация, размещаемая на сайте, является общедоступной.

Пользователей сайта можно разделить на 3 части в соответствии с правами доступа:

1. Посетители
2. Участники турниров
3. Судьи

Посетители имеют доступ только к общедоступной части сайта.

Доступ к расписанию игр турниров и информации об индивидуальных игровых рейтингах имеют пользователи с правами участника или судьи.

Участник может:

1. Регистрироваться на турнир
2. Просматривать расписание игр турнира после регистрации
3. Вносить изменения в счет после окончания матча (изменения вносятся в систему только после подтверждения счета обоими участниками матча)
4. Просматривать информацию об индивидуальных рейтингах и занимаемых местах участников турнира

Судья, тренер может выполнять все те же действия, что и Участник, кроме того:

1. Не может регистрироваться на турнир
2. Может вносить изменения в счет после окончания матча без подтверждения со стороны других лиц

Для получения прав участник или судья пользователь должен пройти регистрацию на сайте и указать нужную роль.

## 2.2. Требования к графическому дизайну сайта

При разработке сайта должны быть использованы преимущественно светлые и контрастные цветовые решения. В качестве примера предлагается ориентироваться на сайт <https://www.rapidshot.com/>

Основные разделы сайта должны быть доступны с первой страницы.

На первой странице не должно быть большого объема текстовой информации.

В дизайне сайта не должны присутствовать:

1. мелькающие баннеры;
2. много сливающегося текста;
3. тёмные и агрессивные цветовые сочетания и графические решения.

## 2.3. Приблизительное пожелание по вкладкам сайта

Необходимо посчитать рейтинг игрока в динамике от возраста и показать перспективу контракта с профессиональным клубом. Для этого используем данные за несколько лет (рейтинг/возраст), используем логистическую кривую, отмечаем принимающего участие в соревновании игрока основного состава ХК «Автомобилист».

Рейтинг тренера определяем по средней динамике роста его подопечных в сравнении со средней динамикой.

Необходимо посчитать не только рейтинг, а его модификацию – как изменяется рейтинг игрока при смене позиции (амплуа), при игре с тем или иным партнером.