МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

|  |  |
| --- | --- |
|  | **ШКОЛА ЕСТЕСТВЕННЫХ НАУК** |

**кафедра информатики, математического и компьютерного моделирования**

|  |
| --- |
| Система контроля успеваемости  на мастер-классах по программированию ДВФУ |

**ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА**

по основной образовательной программе подготовки бакалавров

по направлению подготовки 230700.62 - прикладная информатика

|  |
| --- |
| Студентов группы Б8319а    Мигдальского Артура  Колтунова Марка  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |
| Руководитель  Кленина Н. В. \_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |

|  |
| --- |
| Защищена с оценкой\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (ФИО)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |

**г. Владивосток**

**2015**

# Содержание

[Содержание 2](#_Toc423310765)

[Аннотация 3](#_Toc423310766)

[1. Введение 3](#_Toc423310767)

[1.1. Глоссарий 3](#_Toc423310768)

[1.2. Описание предметной области 3](#_Toc423310769)

[1.3. Неформальная постановка задачи 4](#_Toc423310770)

[1.4. Обзор существующих методов решения 5](#_Toc423310771)

[1.5. План работы 7](#_Toc423310772)

[2. Требования к окружению 8](#_Toc423310773)

[2.1. Требования к аппаратному обеспечению 8](#_Toc423310774)

[2.2. Требования к программному обеспечению 8](#_Toc423310775)

[2.3. Требования к пользователям 8](#_Toc423310776)

[3. Спецификация данных 9](#_Toc423310777)

[3.1. Описание сущностей 9](#_Toc423310778)

[4. Функциональные требования 10](#_Toc423310779)

[5. Проект 11](#_Toc423310780)

[5.1. Средства реализации 11](#_Toc423310781)

[5.2. Проект интерфейса 11](#_Toc423310782)

[6. Реализация и тестирование 15](#_Toc423310783)

[Заключение 16](#_Toc423310784)

[Список источников 17](#_Toc423310785)

# Аннотация

В рамках исследовательской работы планируется изучить предметную область (организация и проведение мастер-классов) и спроектировать систему, которая позволит упростить процесс ведения отчетности по посещаемости и успеваемости, а также будет предоставлять наглядную информацию о предстоящих занятиях в рамках мастер-классов.

# Введение

ДВФУ проводит мастер-классы для школьников с целью привлечения абитуриентов и создания у выпускников школ желания изучать программирование.

Организация занятий по программированию, проводимых в ДВФУ для юных программистов, подразумевает большой объем сопутствующей работы, такой как: формирование списков учеников, учет посещаемости, учет успеваемости и так далее. Также требуется эффективный метод как личного общения, так и удаленного взаимодействия между юными программистами и преподавателями.

Большую часть этой работы можно оптимизировать благодаря современным информационным технологиям.

## 1.1. Глоссарий

ДВФУ — Дальневосточный Федеральный университет.

Семинарное занятие — занятие проводимое в рамках мастер-классов ДВФУ.

Юный программист — школьник, который посещает семинарные занятия в ДВФУ по информатике.

Паттерн — повторимая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

CMS — информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления контентом (то есть содержимым).

Информационная система (сокр. ИС) — система обработки информации и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

## 1.2. Описание предметной области

Мастер-классы ДВФУ — форма занятий школьников, которые хотят попробовать себя в спортивном программировании, подготовиться к поступлению в ВУЗ на специальности, связанные с программированием, а также понять, хотелось бы заниматься этим в дальнейшей жизни или нет.

Занятия по мастер-классам проводятся в кампусе ДВФУ, куда юные программисты добираются на автобусе, в корпусе D, в субботу, во второй половине дня. Занятия ведут преимущественно студенты или выпускники Дальневосточного Федерального Университета, что положительно сказывается на процессе обучения.

Группы школьников разделены на три ступени: подготовительная, начальная и высшая. Когда школьник приходит в первый раз на занятия, его определяют в какую-либо группу, в зависимости от уровня его подготовки. Обычно это подготовительная ступень, но бывает и сразу начальная. Также набор осуществляется по результатам городских олимпиад по информатике среди школьников. Ступени отличаются между собой сложностью заданий. Если на подготовительной ступени происходит знакомство с языком программирования, то на высшей ступени решаются задачи, приближенные к реальности, которые не по силам многим профессиональным программистам.

Количество обучающихся в каждой из групп разное, в подготовительной ступени оно может достигать ста человек, но до высшей ступени доходят около десяти. Переход на другую ступень осуществляется по результатам олимпиад, а также летней школы, также по результатам его личной работы и успехам в классе. Летняя школа – те же мастер-классы, но проходящие ежедневно в течение двух недель на летних каникулах, тогда как мастер классы проводятся один раз в неделю на протяжении всего учебного года.

Из-за значительного притока школьников существуют проблемы ведения учета обучающихся, который существует в бумажном и электронном виде, а именно:

* Рутинное заполнение списков утомляет преподавателей.
* Информация в данных списках не является актуальной вследствие нестабильной посещаемости со стороны юных программистов.
* Учащиеся не в состоянии отследить свои успехи, так как нет общих списков с результатами.
* Родители не способны удостовериться, действительно ли их ребенок посещает мастер-классы.
* Ограниченная возможность просмотра информации о заданиях для юных программистов, которые следует сделать дома.

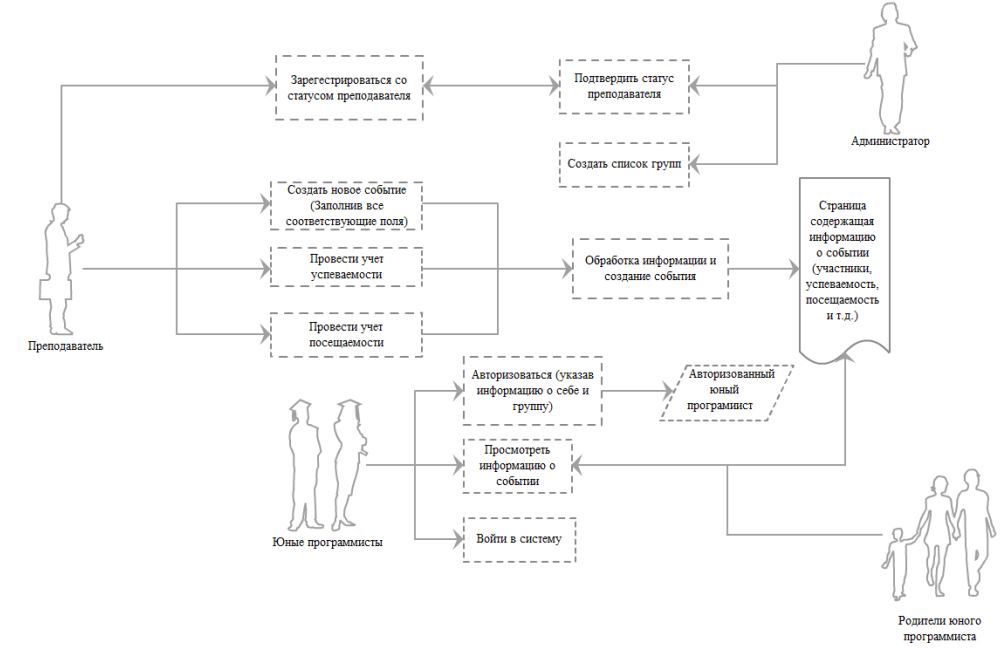
Прогрессирование данных проблем оказывает достаточно сильное влияние на эффективность семинарных занятий.

## 1.3. Неформальная постановка задачи

Система контроля, предоставляющая доступ как для преподавателей, так и для учеников; имеющая свою базу данных с возможностью редактирования; имеющая возможность создания событий(уроков) и учёт успеваемости и посещаемости учащихся преподавателем.

Цели данной работы:

* Упростить работу преподавателя в области учета посещаемости занятий.
* Создать систему, которая помогает в информировании юных программистов о прошедших и предстоящих событиях.
* Проинформировать родителей об успеваемости юных программистов.

Рис. 1. Диаграмма вариантов использования.

## 1.4. Обзор существующих методов решения

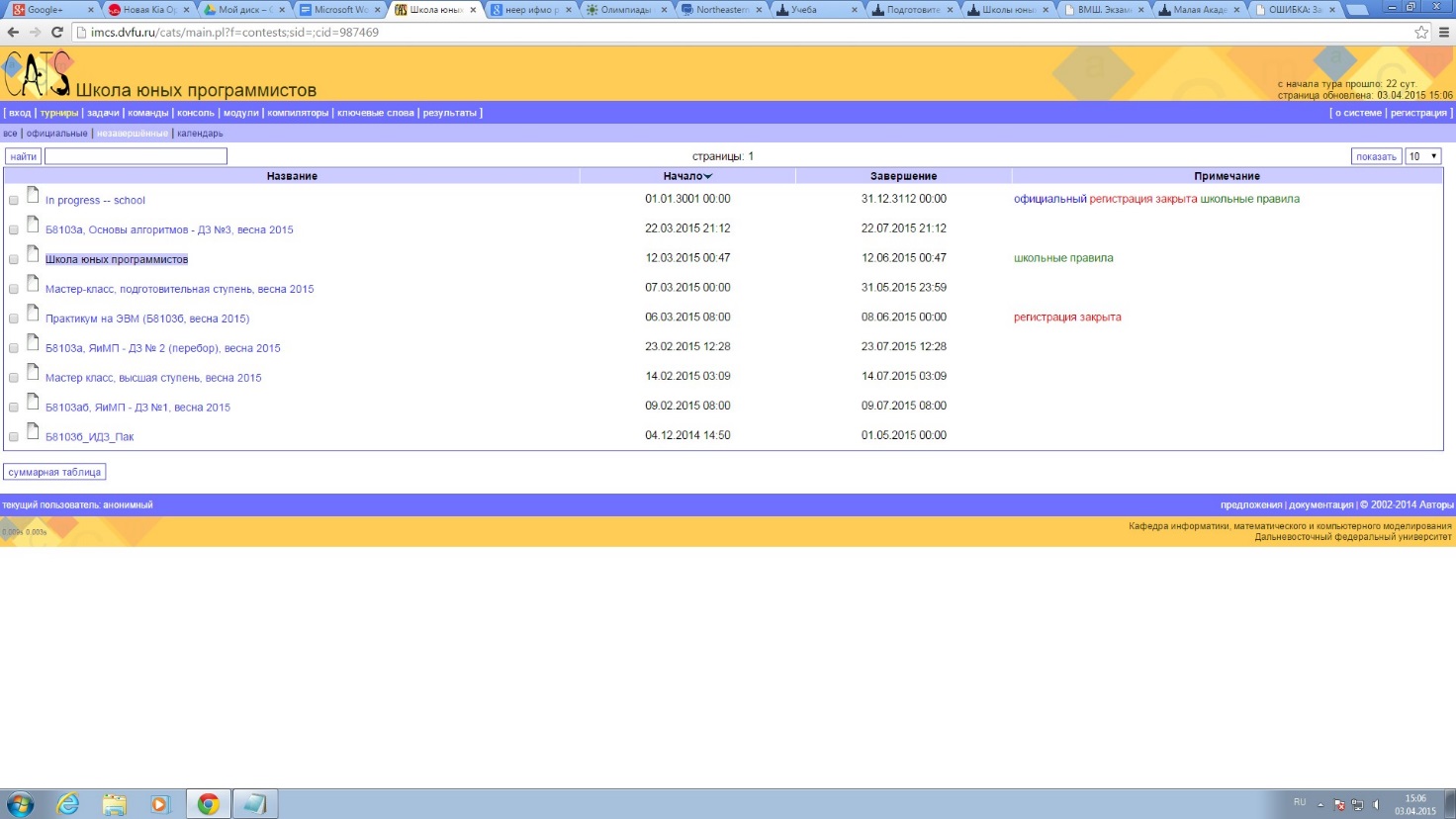
### 1.4.1. Cats

Cats — это система с помощью которой проводятся чемпионаты ACM по Дальневосточному региону, соревнования школьников, тренировки.

Использование системы позволит пользователю:

* Принимать участие в соревнованиях, в том числе виртуально — в прошедших соревнованиях.
* Решать задачи из архива от своего имени.
* Наблюдать свои достижения в общей рейтинговой таблице участников.

Количество заданий в системе: 1205.

Рис. 2. Пример интерфейса системы Cats.

### 1.4.2. Blackboard Learn

Blackboard Learn разработан с ориентацией на студентов. Он предоставляет доступ к инструментам, необходимым студентам.

* Глобальная навигация позволяет иметь быстрый доступ к своим оценкам и заданиям.
* Обновления и уведомления держат вас в курсе предстоящих дат и событий.
* Совместные пространства позволяют вам быть на связи с академическим сообществом.
* Мобильные возможности позволяют учиться в любом месте, в любое время.

Blackboard Learn расширяет основные учебные задачи, такие как сортировка и создание оценок. С интуитивным дизайном, это единственная LMS, которая сэкономит вам время внутри и вне классной комнаты.

* Обеспечивает возможности обратной связи в виде текстов и сообщений
* Создавать и управлять - не требуется никаких навыков программирования.
* Легко создавать оценки и оценивать работы студентов.
* Анализ данных и превращение их в полезную информацию.

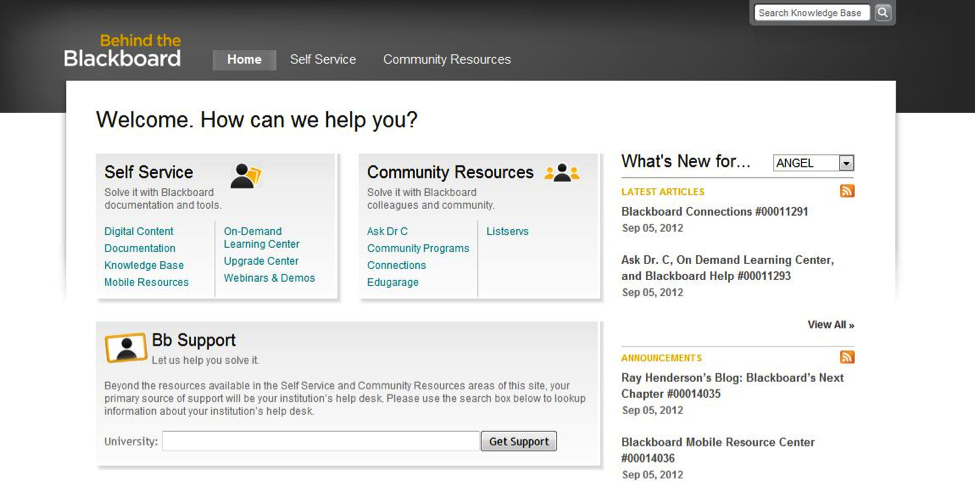
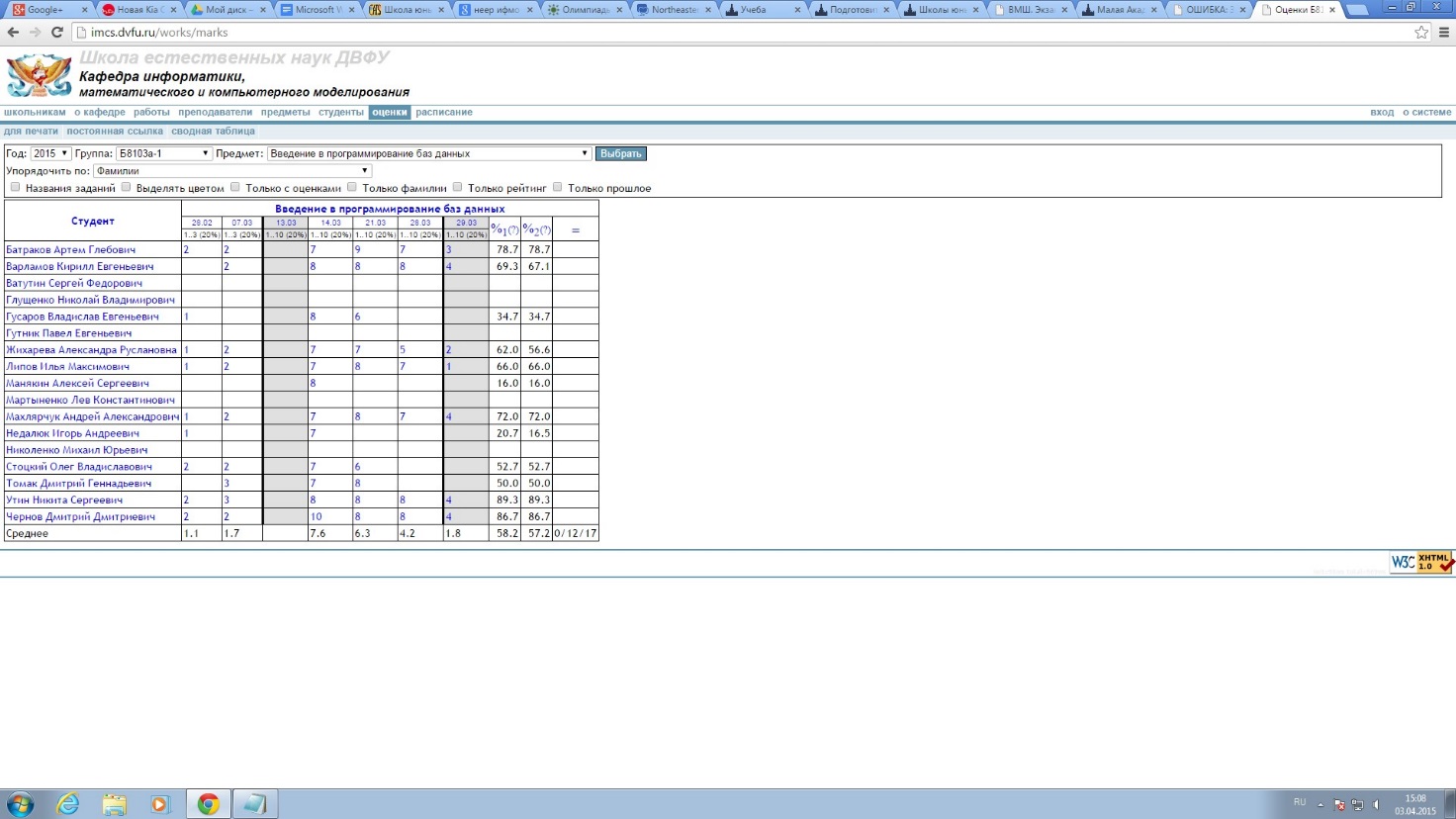


Рис. 3. Пример интерфейса системы Blackboard.

### 1.4.3. AWorks

Наиболее близким по функционалу конкурентом для данного программного решения является система AWorks, внедренная на кафедре информатики ДВФУ. Данная система позволяет преподавателям кафедры размещать описания предметов, практические задания, задания на курсовые, дипломные и реферативные работы, сведения о расписании, проводить тестирование студентов. Система поддерживает рейтинговую оценку успеваемости с учётом опоздания при сдаче заданий.

Рис. 4. Пример интерфейса системы Aworks.

### 1.4.4. СИТО

Система тестирования, применяемая во Владивостокском Государственном Университете Экономики и Сервиса, позволяет проводить тестирование, которое включает в себя учебные тесты, промежуточные аттестации и экзамены. Также, в системе реализованы различные курсы, по аналогии с Blackboard.

Сравнительная таблица существующих методов решений.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Сats | AWorks | Blackboard | СИТО |
| Описание учебных предметов | **-** | **+** | **+** | **+** |
| Описание практических заданий | **-** | **+** | **+** | **+** |
| Сведения о расписании | **-** | **+** | **-** | **-** |
| Тестирование студентов | **+** | **+** | **+** | **+** |
| Удобный интерфейс | **+** | **+** | **+** | **+** |
| Общий рейтинг учащихся | **+** | **+** | **-** | **-** |

### Вывод

В результате анализа было выявлено, что существующие системы не предоставляют необходимый функционал в полной мере, однако обладают лишними возможностями, усложняющими работу с системой. На основе этого было принято решение о реализации собственной системы.

## 1.5. План работы

Задачи:

* Продумать и создать схему базы данных, соответствующую требованиям данной системы.
* Выбрать наиболее подходящие инструменты для реализации поставленных задач.
* Выбрать наиболее подходящий архитектурный паттерн или использовать существующее решение в виде CMS.
* Удостовериться в возможности реализации поставленных задач средствами выбранных инструментов.
* Упорядочить и разделить всю работу по разработке проекта на модули и разделить работу между участниками команды.
* Решить проблемы, которые возникнут при разработке модулей.
* Объединить модули в готовый проект.
* Протестировать проект на возможные ошибки и недочеты.
* Исправить выявленные ошибки и недочеты.

# Требования к окружению

## 2.1. Требования к аппаратному обеспечению

Для использования системы подойдет любой современный компьютер, либо мобильное устройство с возможностью выхода в интернет.

## 2.2. Требования к программному обеспечению

Для использования системы достаточно использовать любой современный интернет-браузер.

## 2.3. Требования к пользователям

В системе присутствуют пользователи двух разных типов: юный программист и администратор.

Пользователь «Администратор». От данного типа требуются:

* Опыт преподавания и ведения преподавательской деятельности для ведения учета посещаемости и успеваемости.
* Создавать события.
* Подтверждение статуса администратора для зарегистрированных в системе пользователей.

Пользователь «Юный программист». К данному типу пользователей не предъявляется особых требований, кроме опыта работы в интернет-браузере.

# Спецификация данных

Рис. 5. Диаграмма сущностей БД.

## 3.1. Описание сущностей

### 3.1.1. Сущность «Пользователь»

Данная сущность предназначена для хранения информации о пользователях: юный программист. Все атрибуты данной таблицы являются обязательными. Для редактирования недоступен атрибут - идентификатор пользователя. Атрибут Статус изменяется только после подтверждения администратором системы.

Атрибуты:

* Идентификатор пользователя. Целое число. Уникальное.
* Имя. Текст.
* Фамилия. Текст.
* Статус. Бинарное.
* Логин. Текст. Уникальное.
* Пароль. Текст.
* Хэш-код. Текст.

### 3.1.2. Сущность «Информация»

Данная сущность предназначена для хранения информации о пользователях. Все атрибуты данной таблицы являются обязательными.

Атрибуты:

* Идентификатор пользователя. Целое число.
* Имя. Текст.
* Фамилия. Текст.
* Отчество. Текст.
* Адрес электронной почты. Текст.
* Номер телефона. Текст.

### 3.1.3. Сущность «Группа»

Данная сущность предназначена для хранения информации о группах. Все атрибуты данной таблицы являются обязательными. Для редактирования недоступен только один атрибут - идентификатор группы.

Атрибуты:

* Идентификатор группы. Целое число. Уникальное.
* Название группы. Текст. Уникальное.
* Идентификатор куратора. Целое число. Ссылка на сущность Персонал.

### 3.1.4. Сущность «Оценки»

Данная сущность предназначена для хранения информации о полученных оценках. Все атрибуты данной таблицы являются обязательными. Все атрибуты доступны для редактирования.

Атрибуты:

* Идентификатор пользователя. Целое число. Ссылка на сущность Пользователь.
* Идентификатор события. Целое число. Ссылка на сущность Событие.
* Оценка. Целое число.
* Присутствие. Логическое.

### 3.1.5. Сущность «Событие»

Данная сущность предназначена для хранения информации о событиях. Обязательными атрибутами являются Идентификатор, Название и Дата.

* Идентификатор события. Целое число. Уникальное.
* Название. Текст.
* Дата. Дата/время.
* Описание. Текст.
* Задание. Текст.
* Идентификатор группы. Целое число. Ссылка на сущность Группа.
* Идентификатор преподавателя. Целое число. Ссылка на сущность Персонал.

### 3.1.6. Сущность «Персонал»

Сущность для персонала, в ней находится администратор, а также преподаватели. Для подтверждения статуса преподавателя подразумевается использование бинарных масок. То есть: 0011 — статус преподавателя не подтвержден, 0111 — статус преподавателя подтвержден, 1111 — администратор. Генерация хэш-кода по правилам описанным в сущности пользователя.

* Идентификатор пользователя. Целое число. Уникальное.
* Логин. Текст. Уникальное.
* Пароль. Текст.
* Имя. Текст.
* Фамилия. Текст.
* Идентификатор группы. Целое число
* Статус. Бинарное.
* Хэш-код. Текст.

# Функциональные требования

Необходимо обеспечить следующие функции:

* **Регистрировать пользовательские аккаунты** —аккаунт будет включать в себя информацию для идентификации пользователя в системе и определения его статуса. Поле статус будет являться бинарным числом, что будет отражать права доступа по отношению к системе. При регистрации пользователь может указать определенную группу, если администратор подтвердит запрос, то пользователь перейдет в категорию зарегистрированных юных программистов.
* **Создавать новые события (Администратор)** — в статусе преподавателя будет возможно создавать новые события, событие должно быть привязано к определенной дате. Создание события происходит по нажатию кнопки "Создать событие", которая находится на странице календаря. Видна данная кнопка только пользователям со статусом "Администратор".
* **Просмотр календаря и страницы «Событие»** — Любой посетитель веб-сайта может посмотреть календарь, который включает события, привязанные к определенной дате, а также по ссылке перейти на страницу события для более детальной информации. Страница события содержит название и описание события, рейтинг и посещаемость учащихся. Данная страница доступна всем посетителям веб-сайта для просмотра рейтингов любого юного программиста.
* **Редактирование посещаемости и успеваемости (Администратор)** — Администратор может редактировать сведения об успеваемости и посещаемости: возможность указать оценки, а также поставить отметки о посещении занятий.
* **Функциональная кнопка «домой»** - она будет возвращать на главную страницу сайта (это ее единственная функция). Данная кнопка будет располагаться в левом верхнем углу страницы и будет доступна во всех разделах сайта.

# *Проект*

## 5.1. Средства реализации

При реализации проекта будут использованы языки программирования PHP, JavaScript, а также язык разметки HTML. PHP - скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

## 5.2. Проект интерфейса

## Интерфейс выполнен в стиле минимализма. Основные цвета: белый, серый, синий. Шапка сайта состоит из ссылки «Домой» в левом углу и «Войти в систему» в правом. После авторизации на сайте ссылка «Войти в систему» сменяется ссылкой, ведущей в личный кабинет пользователя.



Рис. 6. Макет интерфейса главной страницы. Главная страница состоит из заголовка и двух ссылок на основные разделы сайта: «Расписание» и «Рейтинг».

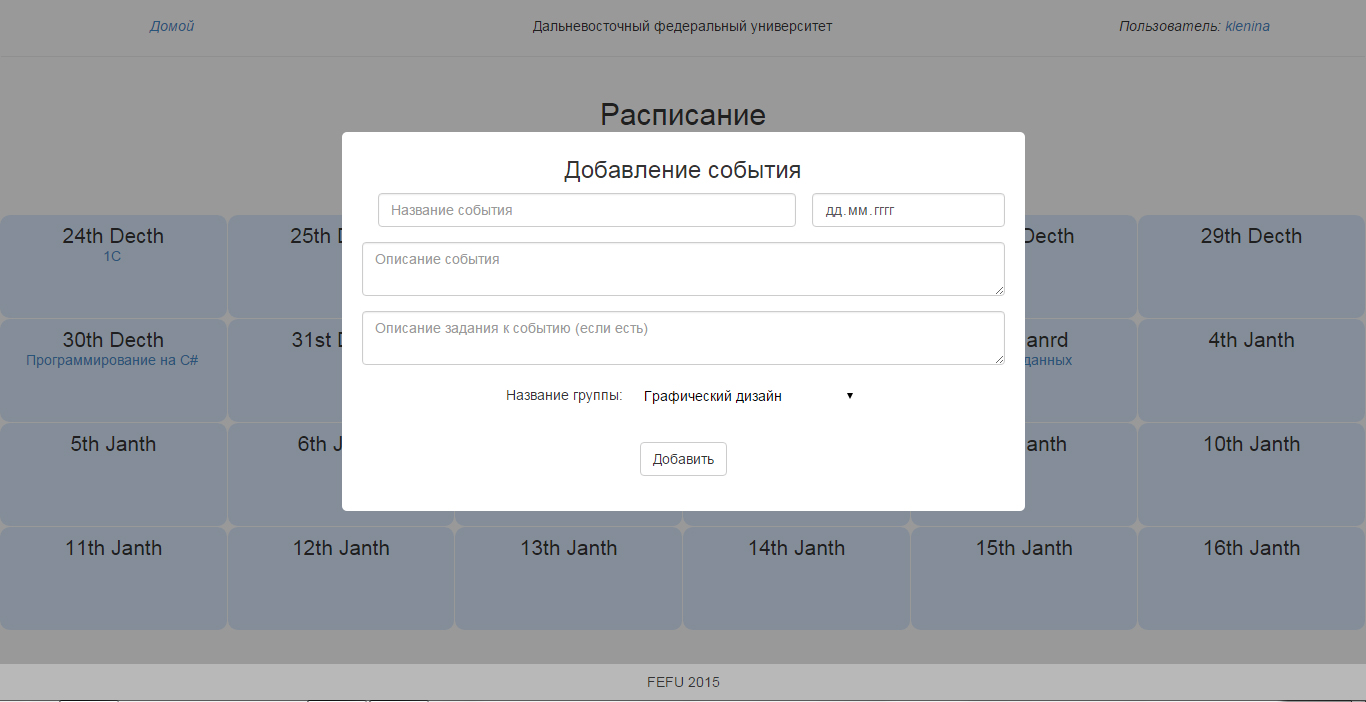


Рис. 7. Макет интерфейса формы добавления события.

На одну дату можно создать неограниченное количество мероприятий.

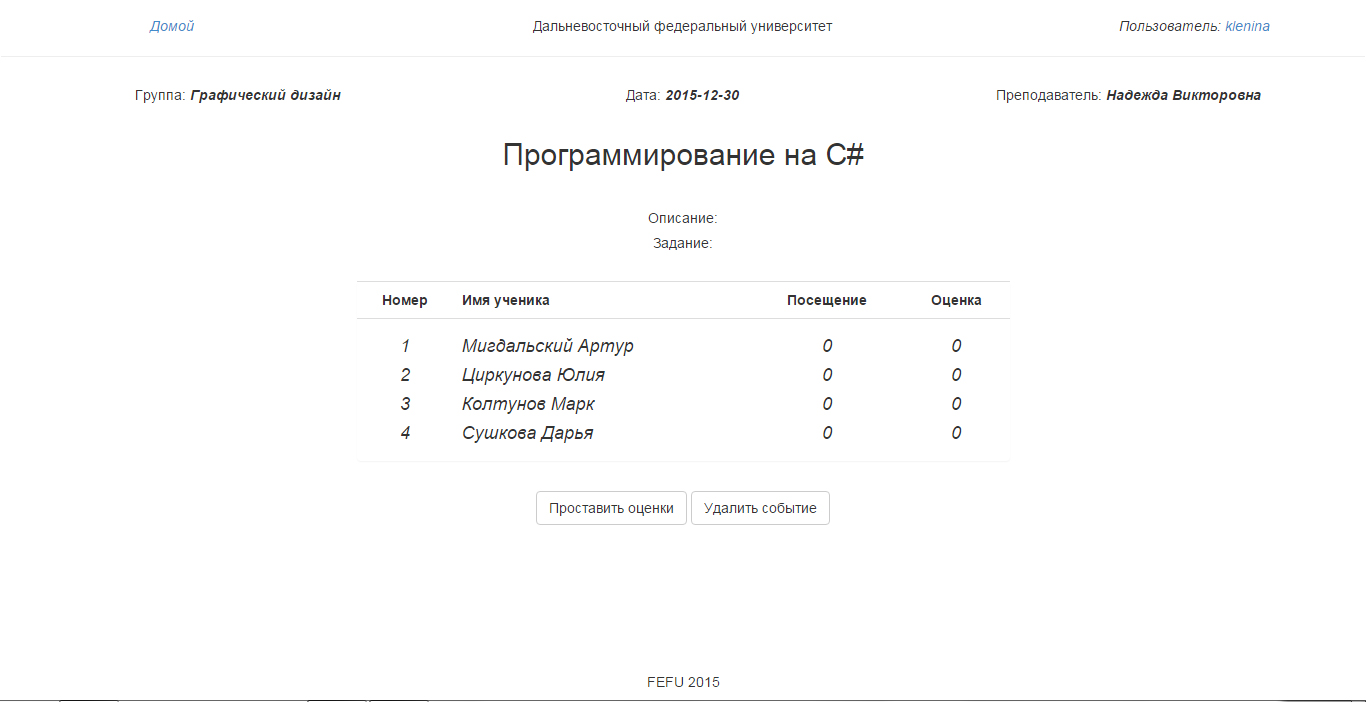


Рис. 8. Макет интерфейса страницы событие.

Заголовок страницы информирует о теме занятия, далее идет описание занятия и задание к выполнению по данной теме. По нажатию кнопки «Проставить оценки» открывается форма, позволяющая преподавателю возможность поставить ученикам оценки и посещения за данное занятие. Кнопка «Удалить событие» служит для удаления случайно созданного (либо потерявшего целесообразность хранения) события.

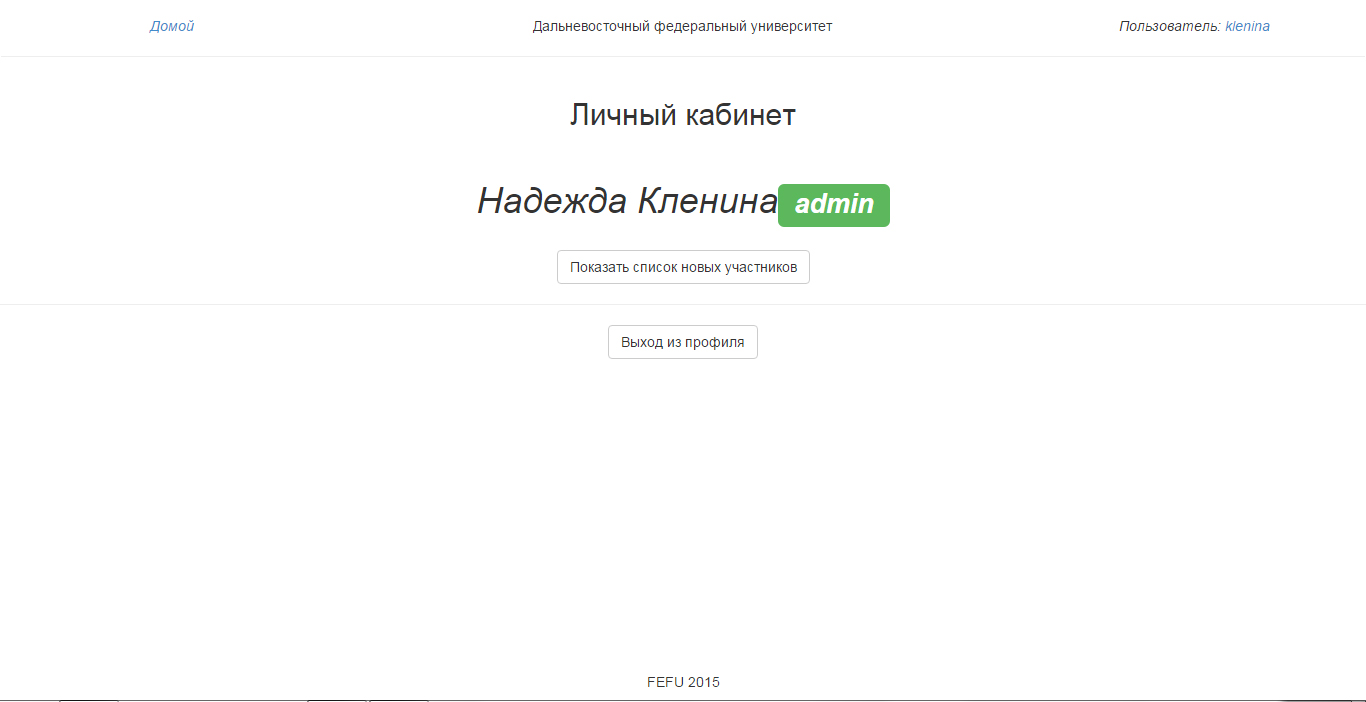


Рис. 9. Макет интерфейса страницы администратора.

Кнопка «Показать список новых участников» открывает форму, которая позволяет подтвердить статус еще не подтвержденных пользователей.

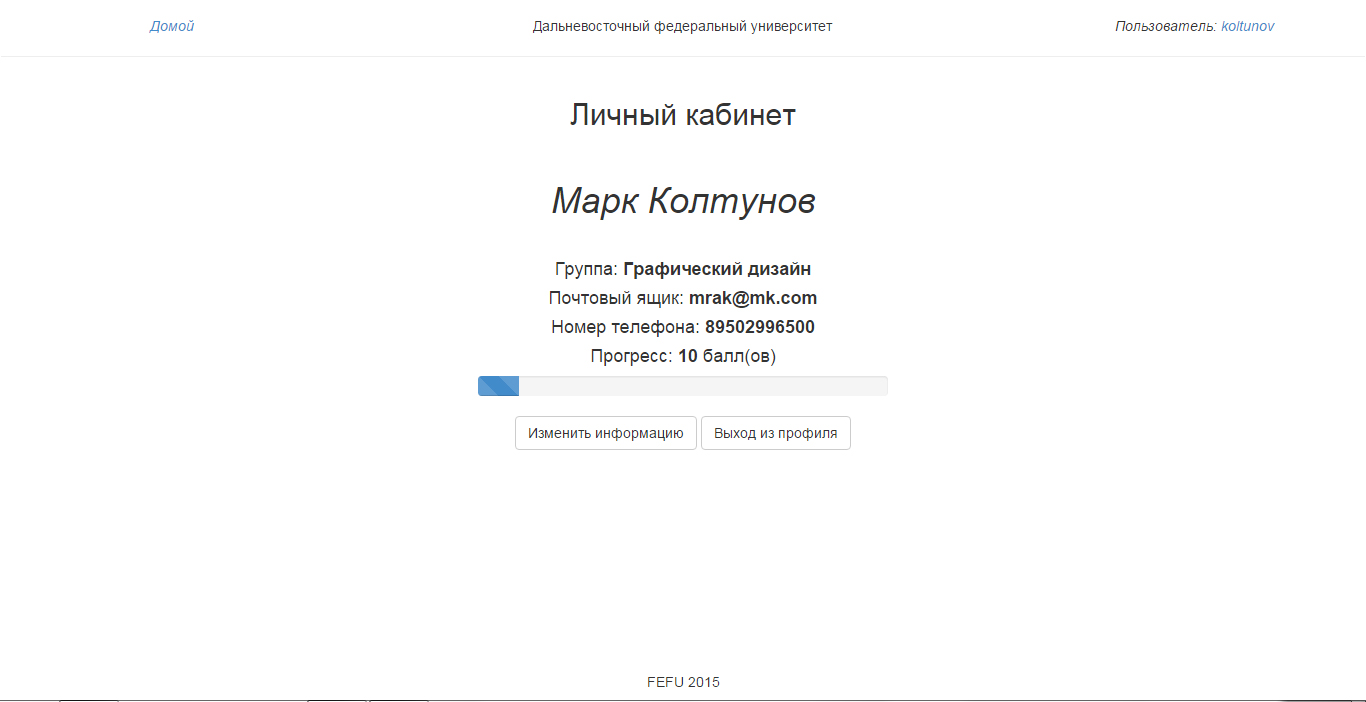


Рис. 10. Макет интерфейса страницы профиля пользователя.

Отображает контактную информацию о пользователе, его группу и успеваемость в текущем семестре. Также реализована функция редактирования пользователем своей контактной информации.

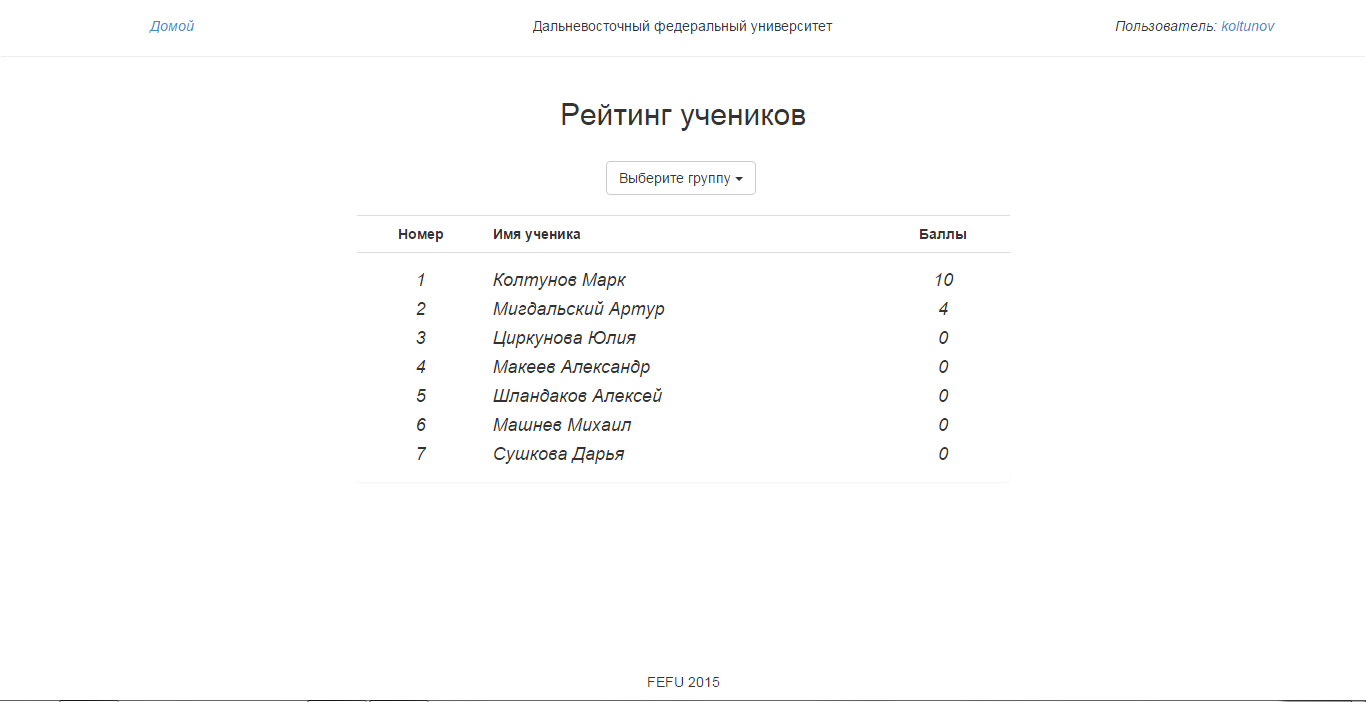


Рис. 11. Макет интерфейса страницы рейтинга пользователей. Рейтинг отображается за текущий период обучения (семестр). Для того чтобы попасть на данную страницу следует зайти в раздел рейтинг на главной странице сайта.



Рис. 12. Макет интерфейса страницы календаря.

В календарь выводятся 24 дня, начиная с текущего.

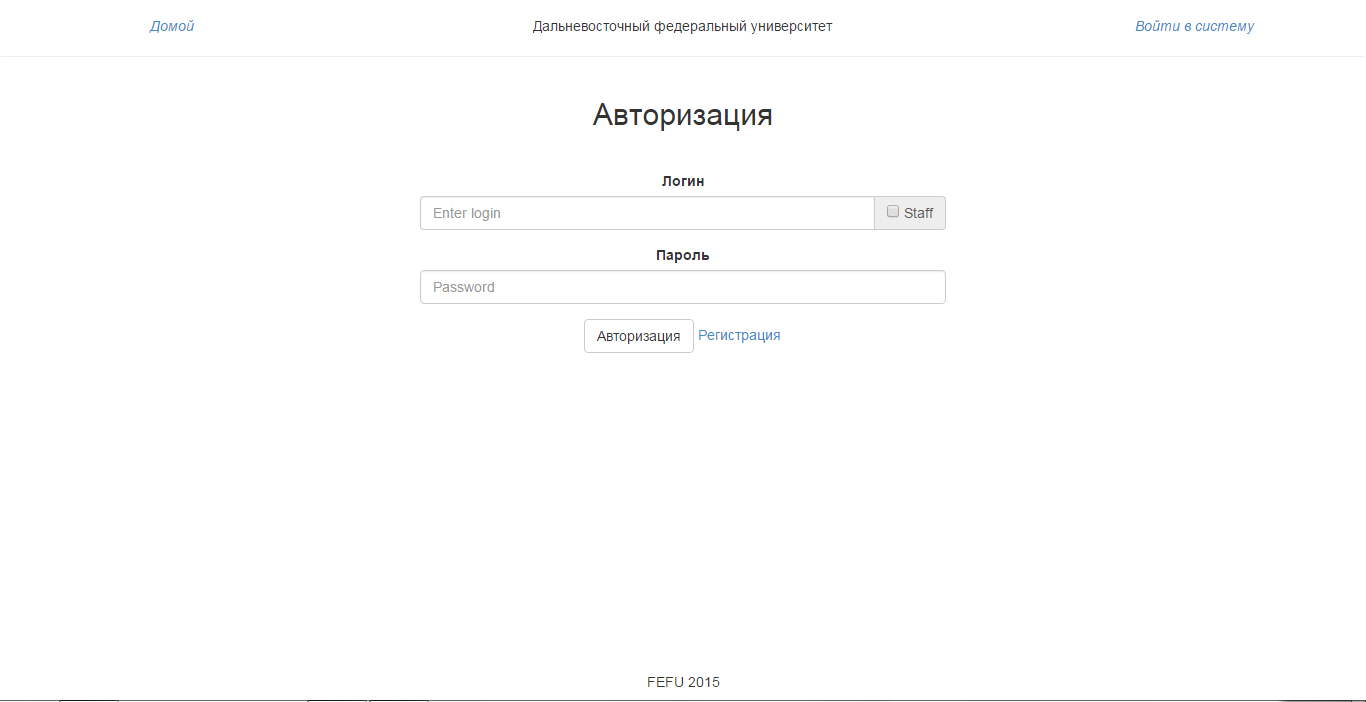


Рис. 13. Макет интерфейса окна авторизации.

Форма авторизации состоит из двух текстовых полей «Логин» и «Пароль» и переключателя «Staff», который отвечает за тип авторизующегося пользователя (если галочка стоит, то авторизуется администратор, иначе, юный программист)

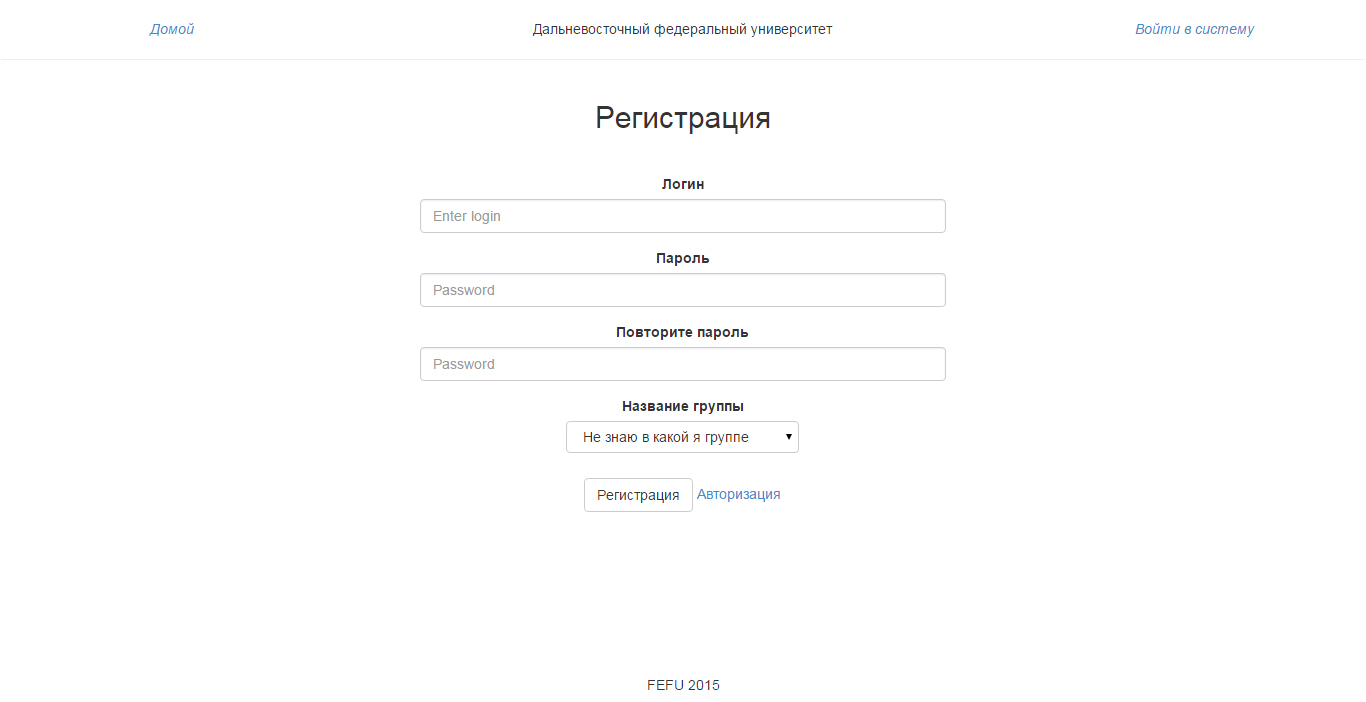


Рис. 14. Макет интерфейса окна регистрации нового пользователя.

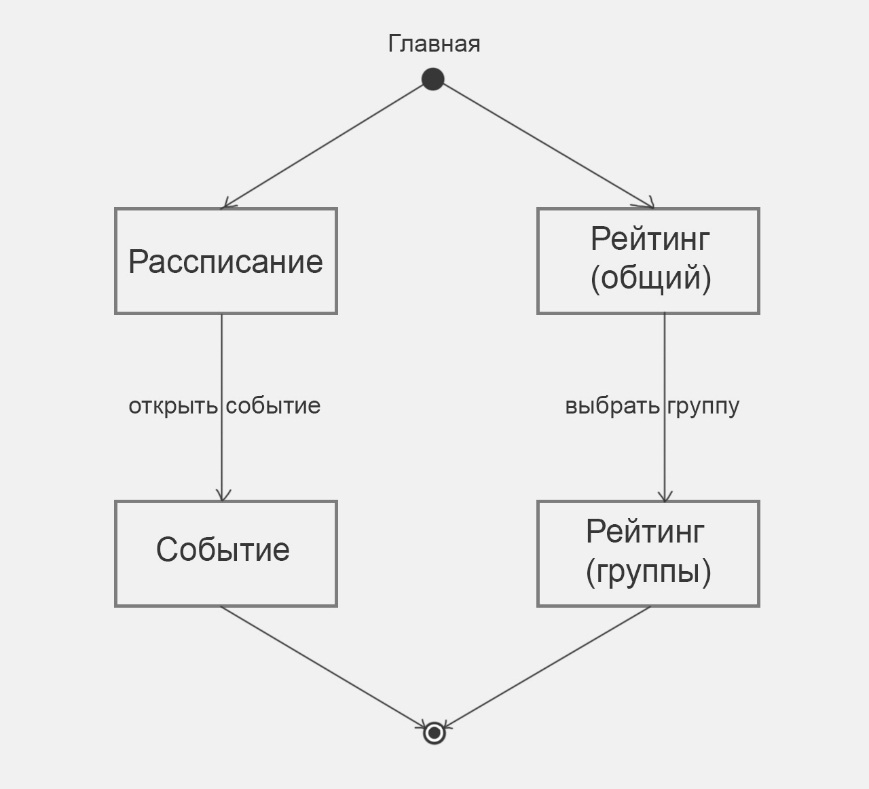


Рис. 15. Диаграмма состояний гостя сайта.

# Реализация и тестирование

В ходе реализации системы было написано 1513 строк кода на PHP и HTML. Система состоит из 21 модуля, необходимых для правильного функционирования архитектуры MVC. 13 модулей view, 5 модулей controller и 3 модуля model.

Среда разработки: Sublime Text 3

Тест-браузер: Safari и Google Chrome

Данный веб-сайт был создан с применением схемы Model-View-Controller. Согласно этой схеме используется несколько шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем оказываются разделены на три отдельных компонента. Преимуществом этой схемы является возможность модификации одного из компонентов с оказанием минимального воздействия на остальные компоненты.

Основные модули проекта:

Controller\_main – логика работы главной страницы сайта

Controller\_timetable – логика работы календаря

Controller\_user – логика работы регистрации, авторизации, а также проверка статуса пользователя(администратора, гостя)

Controller\_404 – логика работы страницы 404

Model\_timetable – работа с базой данных(календарь)

Model\_user – работа с базой данных(авторизация, регистрация, проверка статуса пользователя)

404\_view – отображение страницы 404

event\_view – отображение страницы события

login\_view – отображение страницы авторизации

main\_view – отображение главной страницы

privaterating\_view – отображение страницы частного рейтинга пользователей

register\_view – отображение страницы регистрации пользователей

registerinfo\_view – отображение страницы регистрации дополнительной информации о пользователе

rating\_view – отображение страницы  общего рейтинга юных программистов

timetable\_view – отображение страницы календаря с событиями мастер-класса

cabinet\_view – отображение страницы профиля пользователя

template\_view – шаблон вида в MVC

route.php, controller.php, model.php, view.php - ядро MVC

Остальные пункты проекта имеют подобную реализацию, интуитивно понятную потенциальному разработчику

# Заключение

Таким образом, в ходе исследовательской работы были углублены знания о PHP, JavaScript и архитектуре MVC.

Была проведена исследовательская работа в предметной области (организация и проведение мастер-классов) и реализован проект системы, позволяющей автоматизировать процесс проведения занятий по программированию для школьников и, как следствие, упростить работу преподавателей, ведущих эти занятия.

Изменения, которые были внесены по мере реализации проекта:

1. Изменен дизайн интерфейса. В предыдущем варианте использовалась цветовая гамма в преимущественно серых тонах. В интерфейсе данной системы применялись голубые оттенки.

2. Была упрощена база данных: убрана таблица "Members" из-за ненадобности.

3. Куратор, преподаватель и администратор в данной системе имеют равные права: возможность редактирования страниц и базы данных у них одинаковая.

На данный момент система реализована и отвечает всем требованиям технического задания.

# Список источников

1. http://imcs.dvfu.ru/cats/ — система для проведения чемпионатов Cats
2. http://imcs.dvfu.ru/works/home — Система для учета успеваемости\посещаемости и т.д. AWorks
3. http://eva.vvsu.ru/cito/ — Система для тестирования СИТО ВГУЭС.
4. https://bb.dvfu.ru/ — Система для учета успеваемости, дистанционного образования и т.д. Blackboard FEFU.