Оглавление

|  |  |
| --- | --- |
| Введение...………………………………………………………………….   1. Техническое задание на проектирование и разработку программного обеспечения………………………………………...…… 2. Технические характеристики…………………………………….. 3. Описание программного обеспечения…………………………. 4. Программная реализация и тестирование программного обеспечения………………………………………………………………..   Заключение…………………………………………………………………  Список литературы……………………………………………………….. | 4  5  20  25  35  50  58 |

Введение

С каждым годом темп жизни людей увеличивается. Количество информации, дел, планов и задач растет и не собирается тормозить. Если даже взрослые, состоявшиеся люди не могут должным образом организовать свою жизнь, то что можно говорить о тех, кто только начинает вступать во взрослую жизнь, студентов вузов, ведь для того чтобы уследить за всеми своими делами, правильно все спланировать и успеть в сроки, придется использовать огромное количество не связанных друг с другом инструментов.

Рабочий день студента в первую очередь основывается на уже сформированном расписании занятий. Своевременное посещение пар является основной, но далеко не единственной задачей учащегося в ВУЗе. Так, в первую очередь, студентам необходимо постоянно планировать процесс подготовки к практическим занятиям, выполнения домашних заданий и осуществления самостоятельной работы над долгосрочными задачами, такими как курсовые проекты. От эффективности планирования этих мероприятий порой непосредственно зависит успеваемость студента.

Для достижения данной цели будут выполнены следующие задачи:

* Написать техническое задание на разработку системы
* Описать технические характеристики системы
* Описать программное обеспечение
* Разработать систему и выполнить ее тестирование.

Необходимо разработать web-систему, поддерживающую составления плана дня пользователя отталкиваясь от расписания вуза. Это позволит студенту качественнее распределять свое время и повысить свою успеваемость в учебе и общую эффективность.

1. Техническое задание на проектирование и разработку программного обеспечения
   1. Назначение и цели создания ПО
      1. Назначение ПО

Система предназначена хранения и управления задачами пользователя.

* + 1. Цели создания ПО

Система создается с целью оптимизации процесса планирования деятельности пользователя. В результате создания ежедневника студента пользователь сможет без труда составить план на день отталкиваясь от расписания занятий в ВУЗе.

* + 1. Характеристика объекта автоматизации общее описание автоматизируемого процесса

Для построения моделей, описывающих бизнес-процессы, необходим сначала выделить структуру существующего процесса планирования дня студента. Студент может планировать свой день, не используя вспомогательных средств или применяя для этих целей бумажные носители или различные программный средства. Рассмотрим подробнее каждый из вариантов.

Первый, и один из самых не эффективных вариантов – это планирование без использования каких-либо вспомогательных средств. Даже личные дела и планы довольно сложно запомнить все, а уж тем более огромной количество университетских заданий не получиться удержать в голове. Процесс представлен в виде диаграммы на рисунке 1.1.



Рисунок.1.1. Модель «Не использует вспомогательных средств»

Второй, более жизнеспособный вариант, использование различных бумажных носителей: ежедневники, флипчарты. Но для студента, нужно будет перенести в такой график своей вузовское расписание или пытаться состыковывать в голове планы и учебные занятия. Использование бумажных носителей может облегчить процесс планирования, но это довольно трудоемко, и скорее может использоваться в дополнении к третьему варианту. Диаграмма планирования с использованием бумажных носителей представлена на рисунке 1.2.



Рисунок.1.2. Модель «Пользователь использует бумажные носители для планирования»

Третий, самый эффективный из рассмотренных – это использование различных программных средств. Проблема данного способа заключается в том, что расписание вуза, представлено в формате, не удобном для быстрого переноса в любое программной средство, и даже если студент выполнит эту трудоемкую задачу, ему еще нужно поддерживать расписание в актуальном состоянии

Процесс планирования дня с использование цифровых вспомогательных средств представлен на рисунке 1.3.

Анализ существующих бизнес-процессов показал, что любой из рассмотренных вариантов планирования имеет существенные недостатки, которые могут негативно сказаться на попытках студента спланировать свою деятельность. В связи с этим было принято решение разработать систему, сконцентрировавшись на устранении этих недостатков, то есть максимально удобную и заточенную под студентов высших учебных заведений.

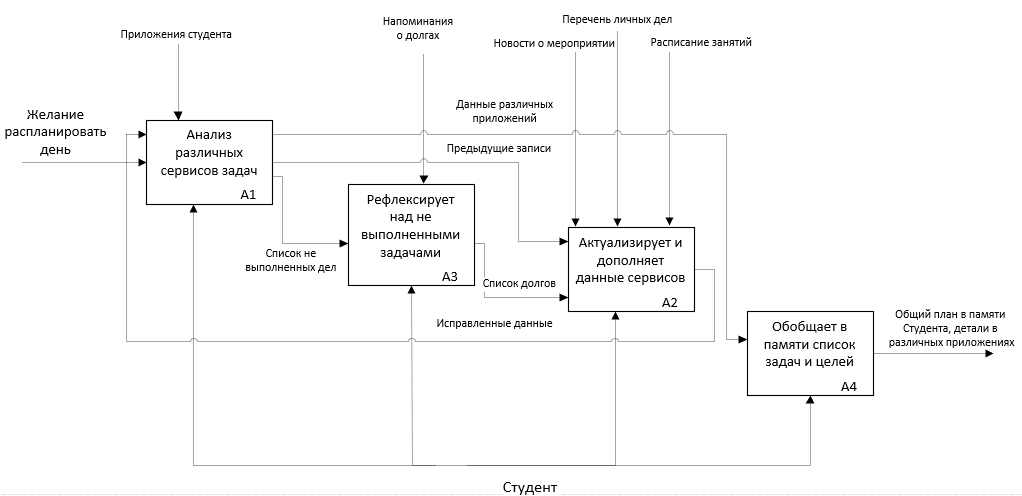


Рисунок 1.3 — Диаграмма процесса планирования с использование ПО.

* 1. Требования к ПО
     1. Требования к ПО в целом

В проектируемой системе выделяется всего две роли: Администратор и Пользователь.

Пользователь – могут добавлять личную информацию и просматривать общую

* Информация о своем календаре
* Информация о конкретном событии
* Добавление события
* Информация общих мероприятиях.
* Добавление мероприятий
* Просмотр личной статистики

Администратор – выполняет функции, связанные с добавлением общей информации

* Добавление и редактирование расписания групп
* Добавление и редактирование общих мероприятий

Общая диаграмма вариантов использования программного обеспечения представлена на рисунке 1.4.

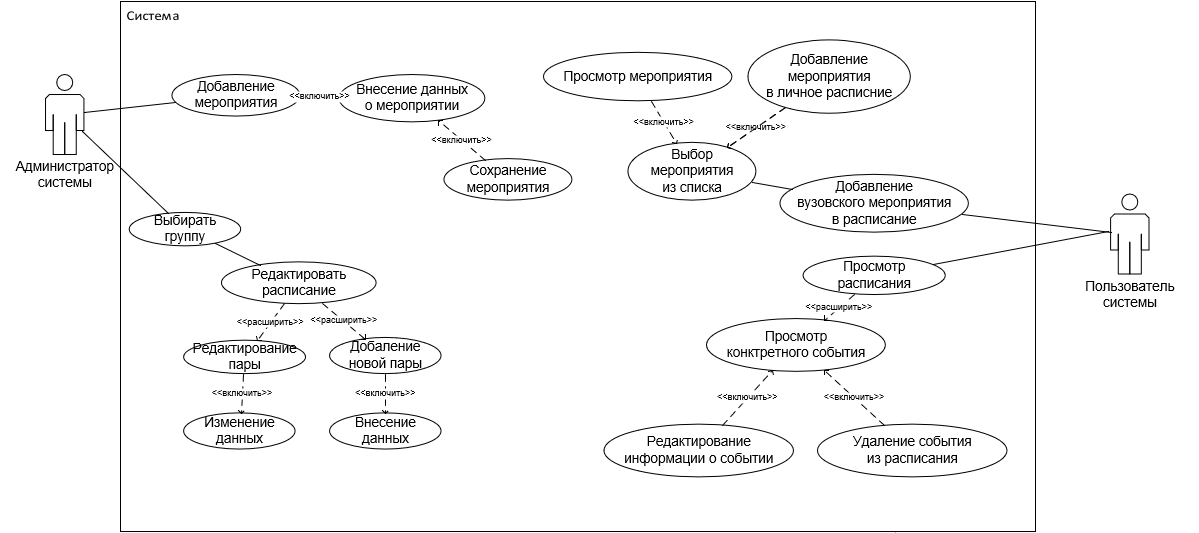


Рисунок 1.4 — Общая диаграмма вариантов использования ПО

Требования к структуре системы:

− Система должна представлять из себя веб-приложение;

− Система должна состоять из серверной части,

обрабатывающей запросы, и клиентского графического интерфейса;

− Система должна состоять из: подсистемы обработки данных

и подсистемы хранения данных;

К режимам функционирования: система должна функционировать

круглосуточно.

Требования к персоналу не предъявляются.

К надежности: Система должна работать без сбоев на протяжении

всего рабочего дня;

К эксплуатации и техническому обслуживанию: регулярное

обслуживание и осмотр сервера.

К сохранности информации при авариях: пользовательские

данные должны храниться на внешнем сервере (для защиты от их

утери).

К патентной чистоте: Разработка должна быть уникальной (не

менее 80% уникальности при проверке с помощью системы

“антиплагиат ру”).

К стандартизации и унификации: Разработка должна осуществляться в соответствии с стандартами IDEF0, IDEF1X, а также UML. При работе с БД необходимо использовать язык запросов SQL, а для спецификации этих запросов – реляционные выражения.

Требования к функциям (задачам) выполняемым в системе

представлены на рисунке 1.8, в виде UML диаграммы прецедентов.

* + 1. Требования к функциям (ВИ), выполняемым ПО

Рассмотрим более подробно такие сценарии как авторизация, заполнение расписания, создание мероприятий, просмотр расписаний, добавление событий, добавление мероприятия в свое расписание.

Вариант использования: «Пользователь авторизовывается в системе».

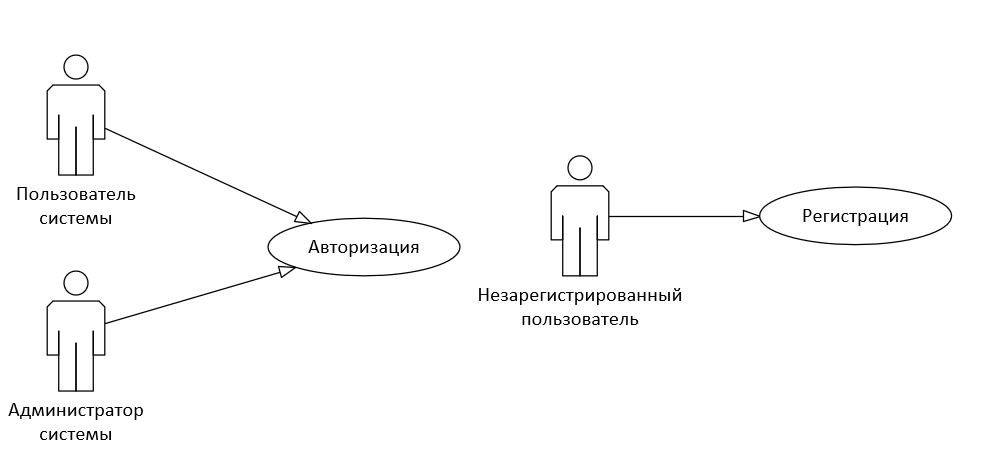
Описание: Пользователь хочет авторизоваться в системе, чтобы составить свое расписание.

Действующие субъекты: Пользователь системы, администратор системы, незарегистрированный пользователь.

Предварительные условия: Пользователь не авторизован.

Основной поток событий:  
 1. Пользователь перешел на сайт приложения  
 2. Пользователь нажал на кнопку авторизоваться.  
 3. Пользователь вводит свои данные и проходит процедуру авторизации

Альтернативный поток событий: если пользователь не зарегистрирован, то он регистрируется после чего переходит к пункту 3 данного сценария.

Рисунок 1.5 – Вход в систему

Вариант использования: «Администратор заполняет расписание».

Описание: Администратор заполняет расписание занятий одной из групп

Действующие субъекты: Администратор системы

Предварительные условия: Расписание группы не заполнено.

Основной поток событий:  
 1. Администратор выбирает группы для заполнения  
 2. Администратор нажимает на кнопку Редактировать  
 3. Администратор выбирает пару для заполнения данных  
Альтернативный поток событий: Администратор редактирует уже заполненное расписание

Рисунок 1.6 – Администратор заполняет расписание

Вариант использования: «Администратор добавляет мероприятие».

Описание: Администратор добавляет внеучебное мероприятие в систему

Действующие субъекты: Администратор системы

Предварительные условия: Мероприятие не добавлено.

Основной поток событий:

* Администратор выбирает Добавление мероприятия.
* Заполняет данные о мероприятии (название, описание, дата и
* время)
* Сохраняет данные

Альтернативный поток событий: Администратор редактирует уже добавленное мероприятие

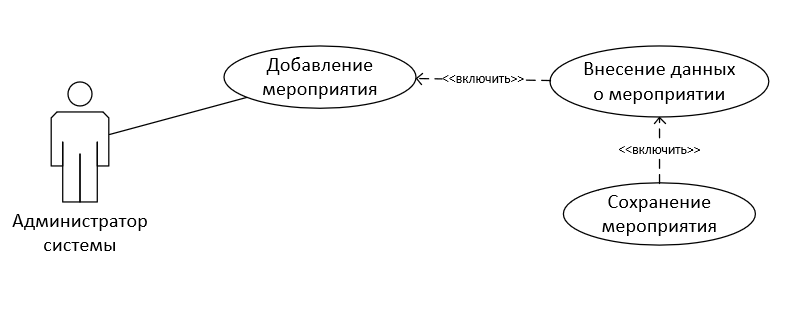


Рисунок 1.7 – Администратор добавляет мероприятие

Вариант использования: «Пользователь просматривает расписание».

Описание: Пользователь открывает и просматривает свое расписание

Действующие субъекты: Пользователь системы

Предварительные условия: Пользователь авторизован

Основной поток событий:

* Пользователь заходит на страницу расписание
* Пользователь просматривает расписание целиком
* Пользователь просматривает информацию о конкретном событии

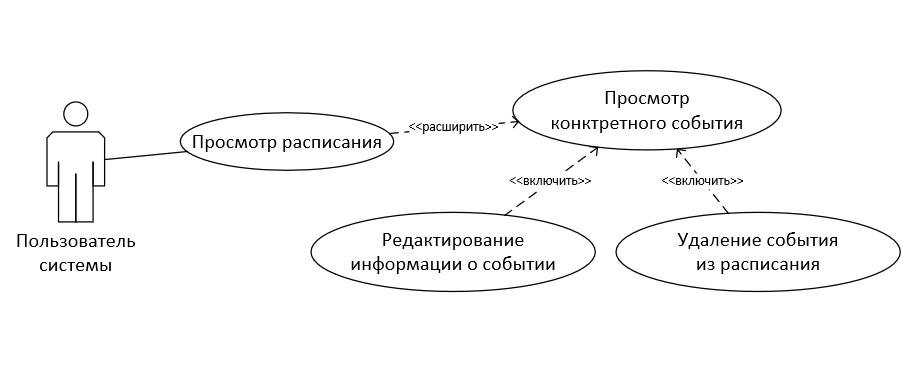
Альтернативный поток событий: при отсутствии расписания группы студенты, система отобразит только добавленные пользователем события. 

Рисунок 1.8 – Пользователь просматривает расписание

Вариант использования: «Пользователь добавляет событие».  
Описание: Пользователь добавляет свое событие в расписание  
Действующие субъекты: Пользователь системы  
Предварительные условия: Пользователь авторизован  
Основной поток событий:

1. Пользователь заходит на страницу добавления событии

2. Пользователь заполняет данные о событии

3. Пользователь сохраняет событие

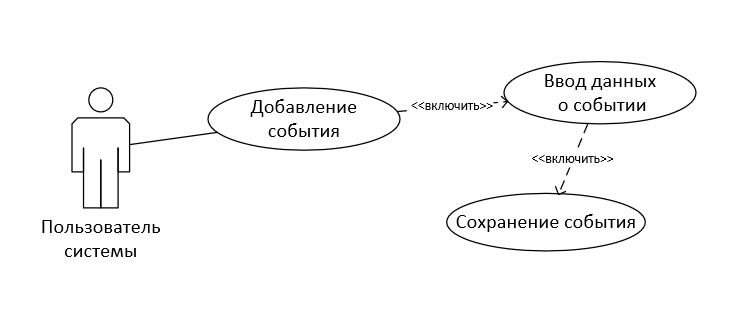


Рисунок 1.9 – Пользователь добавляет событие

Вариант использования: «Пользователь добавляет в свое расписание мероприятие вуза».  
Описание: Пользователь выбирает существующие в системе мероприятия вуза и добавляет их в личное расписание  
Действующие субъекты: Пользователь системы  
Предварительные условия: Пользователь авторизован  
Основной поток событий:

* Пользователь заходит на страницу со списком мероприятий
* Пользователь выбирает и просматривает интересные ему мероприятия
* Пользователь добавляет мероприятие в свое расписание

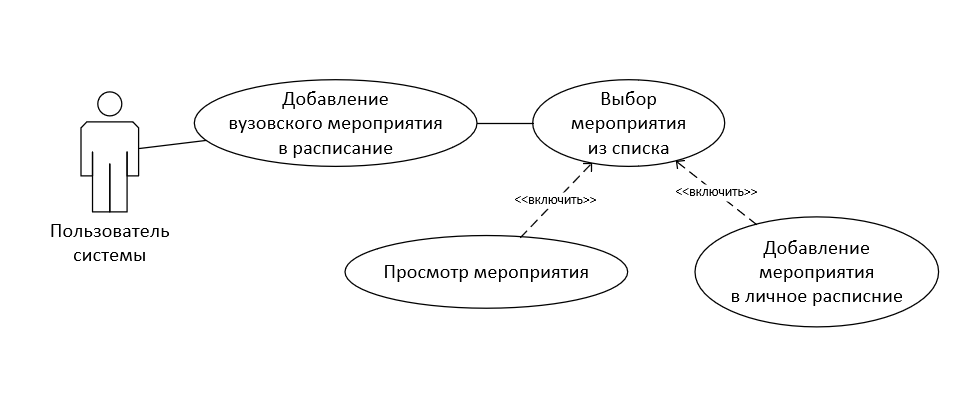
Альтернативный поток событий: Пользователь пытается добавить в личное расписание уже добавленное мероприятие. Система сообщит об ошибке.

Рисунок 1.10 – Пользователь добавляет в свое расписание мероприятие вуза

* + 1. Требования к обработке и хранению данных (БД)

Программная система должна обеспечивать возможность хранения данных с глубиной не менее 10 лет.

Система должна обеспечивать возможность одновременной

работы неограниченного количества пользователей.

Программная система должна восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Должна быть предусмотрена возможность организации автоматического и (или) ручного резервного копирования данных системы средствами системного и базового программного обеспечения (ОС, СУБД), входящего в состав программно-технического комплекса Заказчика.

Физическая схема БД представлена на рисунке 1.11.

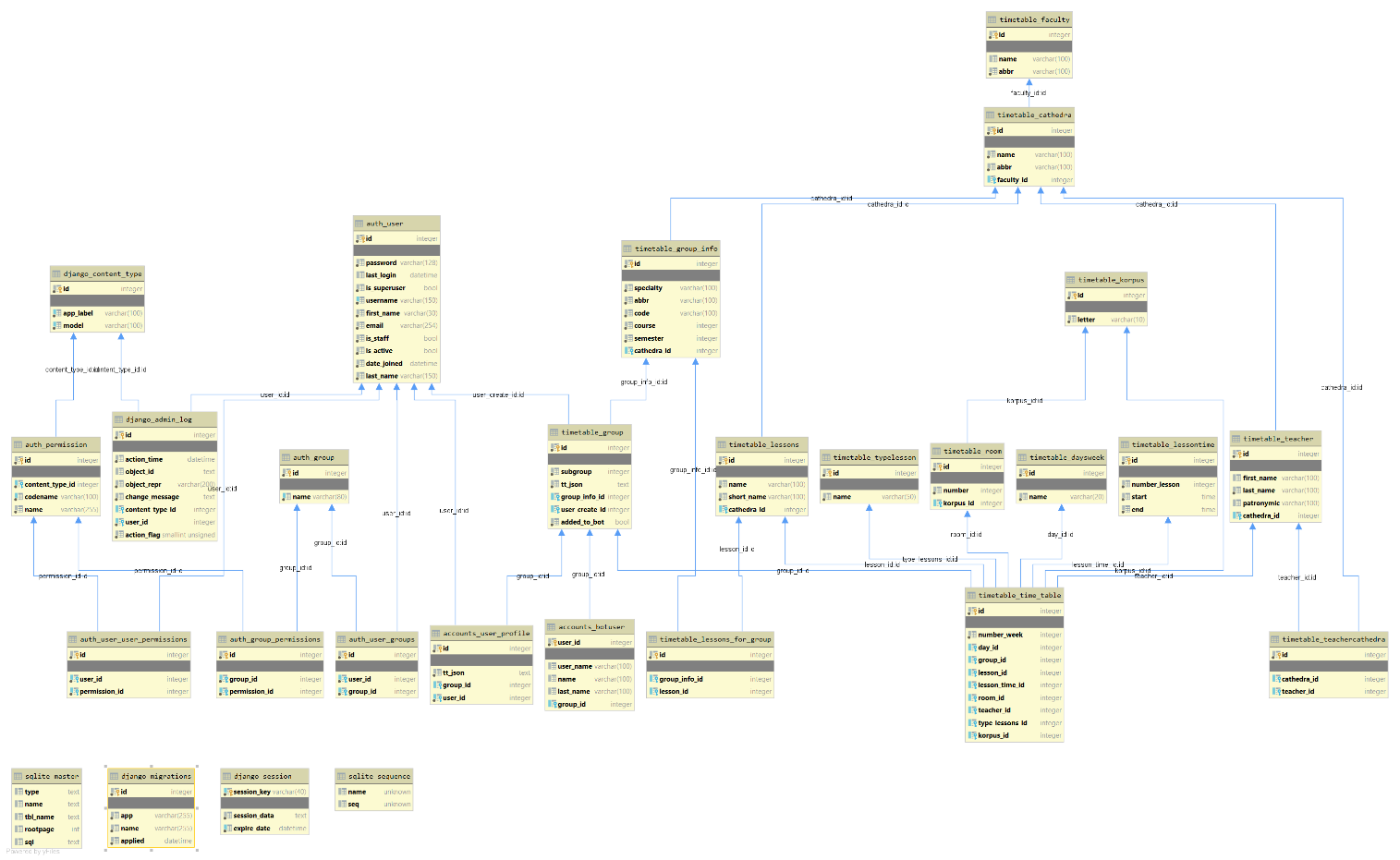


Рисунок 1.11 – Физическая схема базы данных

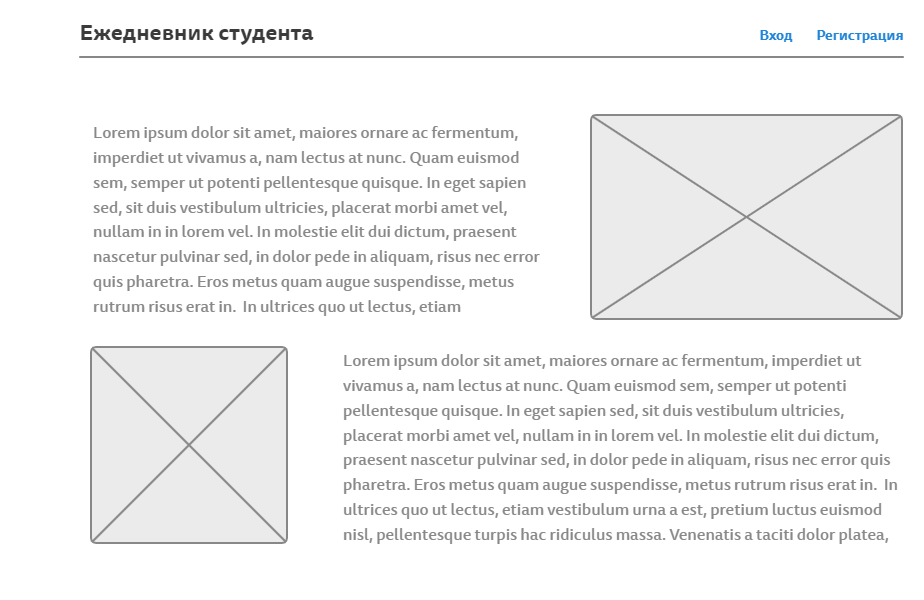
* + 1. Требования к пользовательскому интерфейсу. Эскиз дизайна сайта

Меню сайта должно быть выполнен в нежно голубом или нежно-зеленом цвете. Фон должен быть белым или светло-серым.

Ниже представленные некоторые эскизы.

Первая страница, которую встретит пользователь при открытии сайта, будет стартовая страница с необходимой информацией о ресурсе. На этой же странице пользователь может авторизоваться, нажав кнопку "Вход". Если у него нет учетной записи, то тут же может ее создать.

На странице регистрации, эскиз которой, пользователю необходимо ввести свою электронную почту, пароль, подтвердить пароль, дать согласие на обработку персональных данных, после чего нажать на кнопку зарегистрироваться. После регистрации, необходимо подтвердить свою электронную почту.

  
 Рисунок 1.12 — Стартовая страница системы

После авторизации, для пользователя откроется главная страница, она представлена на рисунке 1.12, на которой можно увидеть календарь пользователя и настройки системы. При нажатии на кнопку добавить новое событие, открывается специальная страница, содержащая пять полей (название, описание, дата, время, тип), все поля обязательны к заполнению, кроме поля с описанием. Страница добавления представлена на рисунке 1.13



Рисунок 1.13 — Календарь

При нажатии на само событие, открывается страница с подробным описанием и возможностью удалить или отредактировать событие. При редактировании открывается страница аналогичная Добавлению события. При удалении события появится диалоговое окно, где необходимо подтвердить удаление.

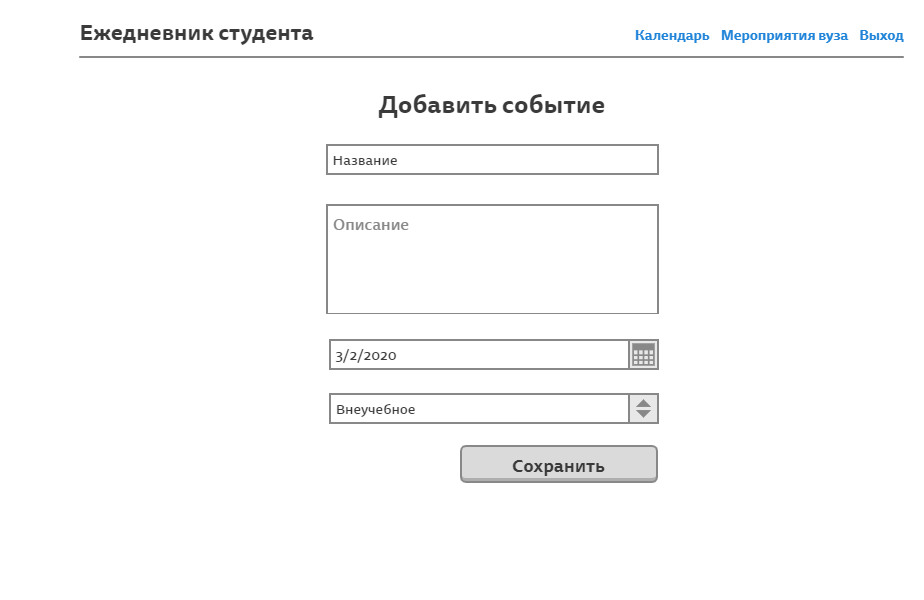


Рисунок 1.14 — Добавить событие



Рисунок 1.15 — Удалить событие

При выборе пункта "Мероприятия института", пользователю предоставляется список всех событий, добавленных администраторами системы.

Страница мероприятий представлена на рисунке 1.16

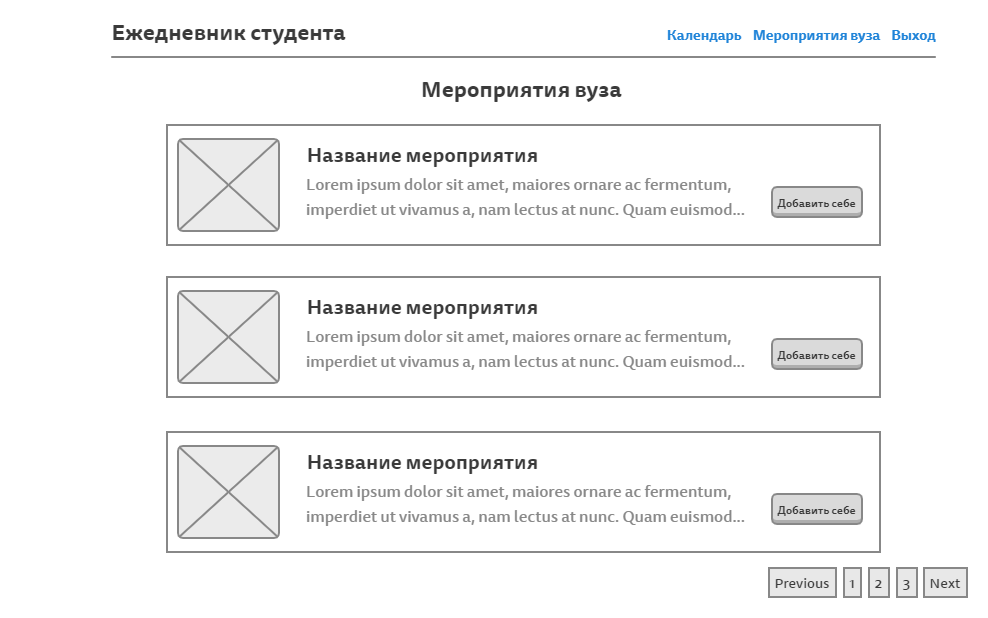


Рисунок 1.16 — Список мероприятий института

При выборе самого мероприятия открывается страница с подробным описания и кнопка для добавления мероприятия в свой календарь. Рисунок 1.17



Рисунок 1.17 — Список мероприятий института

* + 1. Требования к документации

К завершению разработки системы заказчику будут переданы следующие документы:

* + Техническое задание на разработку системы
  + Технические характеристики
  + Описание программного обеспечения
  + Программный реализацию и его исходные тексты
  + Модель функциональных требований к ПО
  + Общая диаграмма последовательности для ПО
  + Диаграмма вариантов использования ПО
  + Диаграммы деятельности функций (ВИ) ПО
  + Форматы входных и выходных данных
  + Логическая схема БД
  + Эскиз дизайна ПО (сайта)
    1. Требования по безопасности

Система не должна позволять пользователям (кроме администраторского аккаунта) просматривать персональные и регистрационные данные, такие как логин, пароль, электронная почта других пользователей.

* + 1. Требования к видам обеспечения

Минимальные требования к серверу:

* + - * CPU: Core 2 Duo(Quad), выпущенный не позже 2009 года
      * RAM: от 4 GB
      * HD: MD (программный) RAID-1 массив, 2x500gb SATA

Требования к хостингу:

* + - * Операционная система: любая
      * Веб сервер: Nginx
      * Язык: Python 3.7. Cистема может работать с Python 3.0+
      * База данных: PostgresSQL 5.0.15+, т.е. Любая современная версия
      * План хостинга: память 128+мб, место на диске 100+мб
      * Оперативная память min 32Mb. Для комфортной работы желательно 128Mb+
      * Место на диске 50Mb+. Для комфортной работы 400Mb+
  1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

УФПКП должна быть проста в использовании и не требовать сложной специальной подготовки пользователей. Функционирование Системы должно осуществляться без специалистов по информационно-программному обеспечению, но под контролем разработчика или администратора, обеспечивающих бесперебойную работу системы. При выходе системы из строя потребуется привлечение разработчика для диагностирования поломки и восстановления исправной работы. Основные мероприятия по вводу системы в действие включают: создание условий функционирования объекта автоматизации на основании, имеющихся у председателя компьютера и ПО, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ.

1. Технические характеристики
   1. Назначение и область применения

Система предназначена для упрощения планирования деятельности студента высшего заведения, в следствии чего, должна повысить успеваемость и эффективность студента.

* 1. Постановка задачи на разработку программы

Разработать систему многокритериального планирования деятельности студента вуза.

* 1. Описание и обоснование выбора метода организации входных и выходных данных

Разделы, не требующие динамической асинхронной подзагрузки

файлов, для отправки данных отправляются с помощью post-запросов. Данные помещаются в заголовках запроса. Таким образом, это обеспечивает некоторую минимальную безопасность, а также не

засоряет url запроса параметрами. Ответ сервер отправляет в виде

html кода, который в свою очередь интерпретирует браузер. Описание

языка разметки html можно найти в стандарте: “RFC: 1866”

Динамическая асинхронная подзагрузка данных обеспечивается

посредством ajax запросов. Они также передаются методом post,

однако в этом случае сервер возвращает ответ в нотации json.

Полученный ответ разбирается посредством скриптов, написанных на

языке java script, и встраивается в DOM страницы. Описание формата json можно найти в стандарте: “RFC: 7159”.

* 1. Описание и обоснование выбора состава технических и программных

Были рассмотрены пять прототипов: многофункциональное приложение «Google Calendar», приложение «Trello», приложение «Todoist», табличный процессор «Excel», сервис «ЛидерТаск».

На их основе был проведен сравнительный анализ с точки зрения их применения для решения задачи.

В качестве критериев для сравнительного анализа программных продуктов выберем следующие:

А1 - Созданием календаря

А2 - Заточенность под расписание вузов

А3 - Сохранением задач

А4 - Создание списков задач

А5 - Формирование статистики

Для определения весов критериев воспользуемся аналитической иерархической процедурой Саати.

Таблица 2.1 – матрица парных сравнений,

средние геометрические и веса критериев.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | А1 | А2 | А3 | А4 | А5 | Среднее геометрическое | Веса критериев |
| А1 | 1 | 1 | 2 1/3 | 5 | 5 | 2,26 | 0,34 |
| А2 | 3/7 | 1 | 5 | 2 1/3 | 5 | 1,90 | 0,29 |
| А3 | 3/5 | 5 | 1 | 1 | 1 2/3 | 1,38 | 0,21 |
| А4 | 1/5 | 3/7 | 1 | 1 | 1 2/3 | 0,68 | 0,10 |
| А5 | 1/5 | 1/5 | 3/5 | 3/5 | 1 | 0,43 | 0,06 |
|  | Сумма | | | | | 6,64 | 1,00 |

Используя полученные коэффициенты определим интегральный показатель качества программных продуктов планирования деятельности человека.

Таблица 2.2 – Интегральные показатели качества

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Критерии | Весовые коэффициенты | Программные продукты | | | | |  |
| Trello | Google Calendar | Excel | Лидер  Таск | Todoist | Базовые |
| Созданием календаря | 0,34 | 1 | 9 | 3 | 9 | 1 | 4,6 |
| Сохранением задач | 0,29 | 9 | 9 | 3 | 7 | 9 | 7,4 |
| Заточенность под расписание вузов | 0,21 | 1 | 1 | 1 | 7 | 1 | 2,2 |
| Создание списков задач | 0,10 | 9 | 9 | 3 | 9 | 9 | 7,8 |
| Формирование статистики | 0,06 | 1 | 3 | 3 | 1 | 1 | 1,8 |
| Интегральный показатель качества Q | | 4,10789 | 6,95208 | 2,58469 | 7,49609 | 4,10789 | 5,04973 |

Диаграмма весовых коэффициентов для критериев A1, A2, A3, A4, A5.

Рисунок 2.1 – Диаграмма весовых коэффициентов для критериев.

Лепестковая диаграмма интегрального показателя качества каждого программного продукта представлена на рисунке 2.

Рисунок 2.2 – Лепестковая диаграмма интегральных показателей качества программ.

Лепестковая диаграмма значений характеристик качества функциональных возможностей (критериев) представлена на рисунке 2.3.

Рисунок 2.3 – Лепестковая диаграмма значений характеристик качества функциональных возможностей.

Сравнительный анализ программных продуктов, предназначенных для планирования, показал, что только программы сервиса «Лидер Таск» имеют значения интегрального показателя выше базового. Сравнение аналогов программных продуктов методом определения экспертных оценок с помощью иерархической аналитической процедуры Саати позволило выявить, что ни одна из рассматриваемых программных систем не удовлетворяет всем заданным критериям.

Для разработки данной системы будет использовать фреймворк Django. Данный фреймворк использует стандартные способы решения задач, что уменьшает или устраняет запутанность кода; облегчает поддержку кода с использованием общей архитектуры и методов. Также Django имеет активное сообщество разработчиков, которые поддерживают фреймворк, обобщенные задачи и новые возможности; сокращает время, затрачиваемое на рутинные задачи, такие как проверка форм и безопасность.

* 1. Источники, использованные при разработке

Услуги по организации и проведению конгрессов, симпозиумов,

съездов, конференций, семинаров, круглых столов, выставок, встреч,

презентаций, переговоров и т.п. Мероприятия в сфере делового

туризма в соответствии с ГОСТ Р 53524-2009 объединяют под общим

названием: конгрессные и выставочные мероприятия.

− ГОСТ 34.003 - 90 ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ.

Автоматизированные системы. Термины и определения.

− ГОСТ 24.104-85 ЕСС АСУ. Автоматизированные системы

управления. Общие требования.

− ГОСТ 34.201-89 ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ. Виды,

комплектность и обозначение документов при создании

автоматизированных систем.

− ГОСТ 34.601-90 ЕСС АСУ. Автоматизированные системы.

Стадии создания.

− ГОСТ 34.602-89 ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ.

Комплекс стандартов на автоматизированные системы. Техническое

задание на создание автоматизированной системы.

− РД 50-34.698-90 МЕТОДИЧЕСКИЕ УКАЗАНИЯ.

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ. Автоматизированные системы.

Требования к содержанию документов.

− ГОСТ 21.404-85 Автоматизация технологических процессов.

Обозначения условные приборов и средств автоматизации в схемах.

− ГОСТ 34.603-92 ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ. Виды

испытаний автоматизированных систем.

− ГОСТ 2.119-2013 Единая система конструкторской

документации (ЕСКД). Эскизный проект.

− ГОСТ Р ИСО/МЭК 15910-2002 Процесс создания

документации пользователя программного средства.

− ГОСТ Р ИСО/МЭК 15271-02 Процессы жизненного цикла

программных средств.

− ГОСТ Р ИСО/МЭК 10746-3-2001 Управление данными и

открытая распределенная обработка.

− ГОСТ Р ИСО/МЭК 8824-3-2002 Информационная технология.

Абстрактная синтаксическая нотация версии один.

− ГОСТ 24.302-80 Общие требования к выполнению схем.

− ГОСТ 24.205-80 Требования к содержанию документов по

информационному обеспечению.

30− ГОСТ 24.210-82 Требования к содержанию документов по

функциональной части.

− ГОСТ 24.703-85 Типовые проектные решения. Основные

положения.

* 1. Выводы. Планирование процесса разработки.

Весь процесс был разделен на 5 этапов:

1. Исследование предметной области
2. Разработка технического задания
3. Разработка эскизного проекта
4. Разработка программы и методики испытаний
5. Разработка программного продукта

На данный момент были пройдены первые 4 этапа.

1. Описание программного обеспечения
   1. Общие сведения:
      1. Обозначение и наименование программного обеспечения

Полное наименование: веб-приложение для многокритериального планирования деятельности студента вуза

Краткое наименование: ЕС

* + 1. Программное обеспечение, необходимое для функционирования программы

Минимальные требования:

Процессор – Intel Xeon L5420 c двумя отключенными ядрами, итого 2 ядра на сокете 775 с частотой 2.5 ггц. Оперативная память 1 ГБ. Жесткий диск Seagate Barracuda 500ГБ. Операционная система Windows XP. Internet Explorer 8. Стандартное приложение для просмотра фотографий. Программа для просмотра PDF-документов.

* + 1. Языки программирования, на которых написана программа.

Для разработки системы были использованы следующие языки программирования:

* Python
* JavaScript
  1. Функциональное назначение:
     1. Особенности и преимущества программного обеспечения

В ходе проведенного исследования у аналогичного программного обеспечения выявлены определенные недостатки, которые будут реализованы в разрабатываемой системе, среди них: автоматическое заполнение расписания вуза, интеграция мероприятий вуза в систему и возможность добавлять их в свой календарь.

* + 1. Классы решаемых задач и (или) назначение ПО

Программная система предназначена для создания личного календаря пользователя на основе расписания института.

* + 1. Сведения о функциональных ограничениях на применение

ЕС предназначен для планирования дня студента, а также составлению статистики по затраченному времени.

Объектами автоматизации являются процессы составления плана дня.

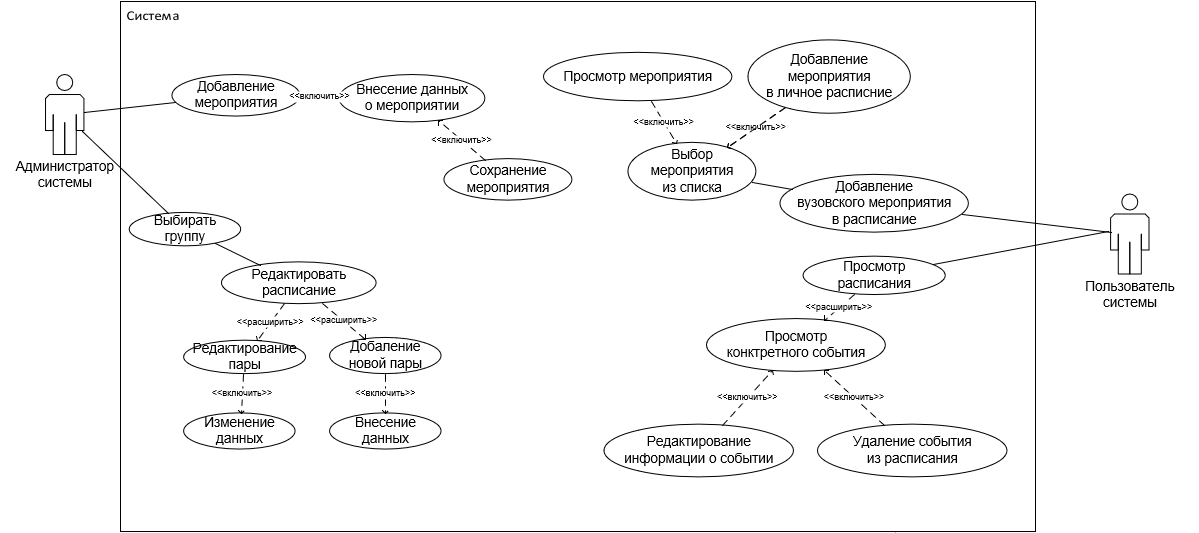
К данным процессам будут относится:

* Составление расписания
* Составление статистики

Осуществляться данные процессы будут следующими специалистами:

* Администратор
* Пользователь

Общая диаграмма вариантов использования программного обеспечения представлена на рисунке 3.1.

Рисунок 3.1 ⎯ Общая диаграмма вариантов использования ПО

* + 1. Назначение и описание основных функций ПО в целом
* Система предназначен для частичной автоматизации процесса планирования дня студента. На рисунке 3.2 представлена общая диаграмма деятельности системы.

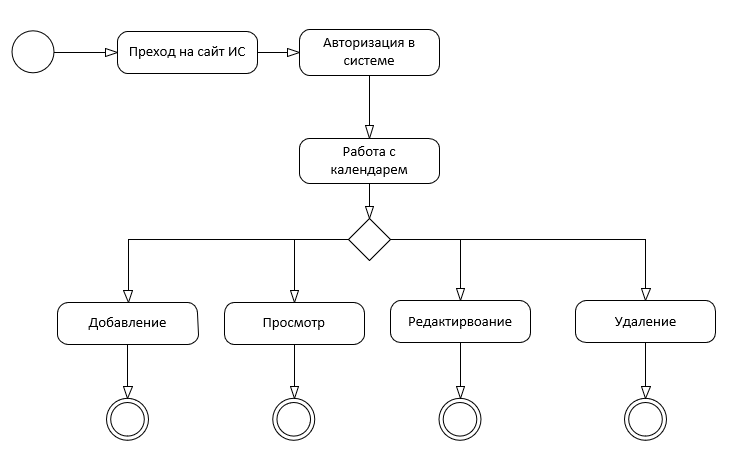
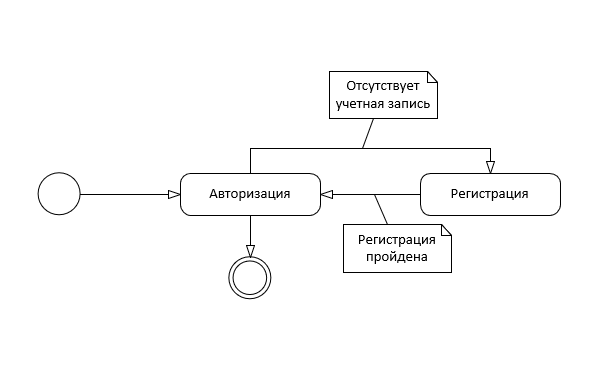


Рисунок 3.2 — Общая диаграмма деятельности

Основные функции ПО:

* Авторизация пользователя;
* Просмотр списка участков и информации о каждом из них;
* Просмотр/добавление/редактирование событий пользователя
* Просмотр/добавление/редактирования информации об общих мероприятий
  + 1. Назначение и описание основных функций для каждого варианта использования ПО

Вход в систему изображен с помощью диаграммы состояний на рисунке 3.3

  
Рисунок 3.3 — Диаграмма состояний в нотации UML. Вход в систему.

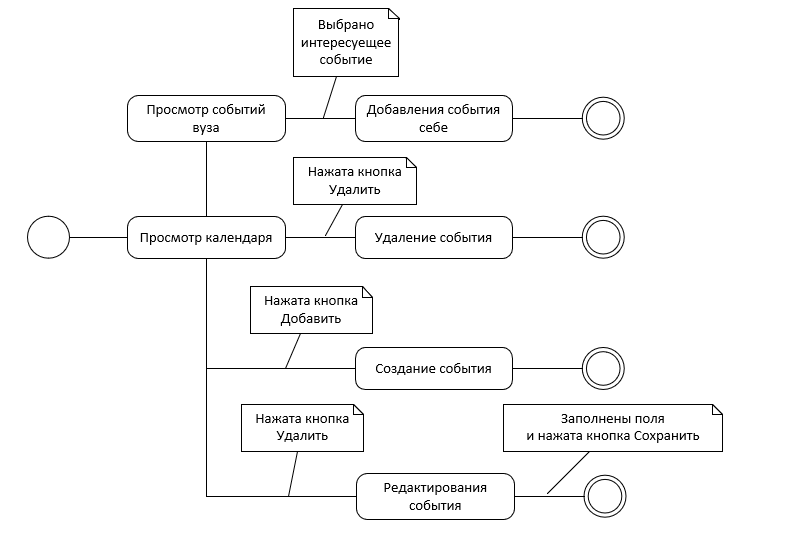
На рисунке 3.3 представлено графическое описание процесса входа в систему пользователя. При отсутствии учетной записи, ему необходимо зарегистрироваться на специальной странице.  
 Работа с календарем изображена с помощью диаграммы состояний на рисунке 3.4  
 На рисунке 3.4 представлено графическое описание процесса работы пользователя со своим календарем. Пользователь просматривает весь календарь. Пользователь может создать новую событие в календаре, создавая его, он заполняет специальные поля. Уже созданные события можно редактировать и удалять.  
 

Рисунок 3.4 — Диаграмма состояний. Работа с событиями.

* 1. Описание логической структуры:
     1. Структура ПО (компонентная диаграмма);

ЕС будет состоять из четырех модулей (рис. 3.4):

1. Модуль работы с авторизацией
2. Модуль работы с расписанием
3. Модуль работы с календарем
4. Модуль работы с мероприятиями

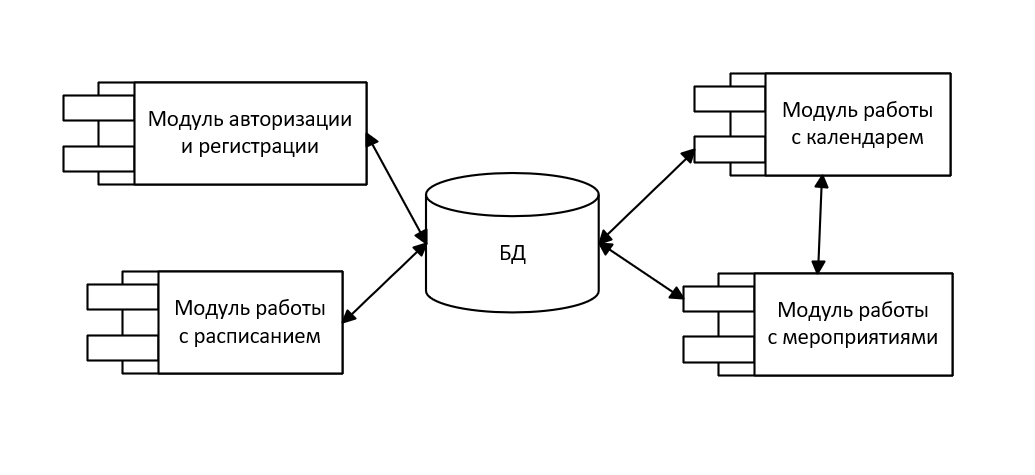


Рисунок 3.4 — Компонентная диаграмма

* + 1. Связи ПО с другими программами.

Программное обеспечение имеет прямую связь с браузерами, так как является Web – приложением. Также в программную реализацию интегрирована возможность связи с другими сервисами с помощью REST API.

* 1. Вызов и загрузка
     1. Способ вызова программы с соответствующего носителя данных

На стороне сервера, необходимо запустить сервер PostgreSQL и Nginx. Способ запуска зависит от операционной системы и указан на сайте производителя.

На стороне клиента, достаточно запустить браузер и ввести url-адрес (на момент реализации системы использовалось estudent.edu в качестве доменного имени) сайта в командную строку.

* + 1. Входные точки в программу.

Входной точкой программы будет считаться страница с авторизацией пользователя, так как авторизация позволит обезопасить систему от внешних вмешательств.

* 1. Входные данные
     1. Характер, организация и предварительная подготовка входных данных

Введенные пользователем данные проверяются на валидность.

Пароль, электронная почта или логин – обязательные поля.

Электронная почта должна соответствовать своему формату.

Ниже приведено соответствие типов для каждого поля:

• email адрес: строка формата email;

• пароль: строка, состоящая из латинских символов, знаков

препинания и символов; описание приведено для всех полей

типа «пароль», используемых в системе;

• повторный пароль: строка, состоящая из латинских символов и

знаков препинания, должен совпадать с значением «пароль»;

• логин: строка, состоящая из латинских символов и знаков препинания;

• даты: даты в формате ISO 8601

• названия событий: строка состоящая из любых символов

* + 1. Формат, описание и способ кодирования входных данных.

Разделы, не требующие динамической асинхронной подзагрузки

файлов, для отправки данных отправляются с помощью post-запросов. Данные помещаются в заголовках запроса. Таким образом, это обеспечивает некоторую минимальную безопасность, а также не

засоряет url запроса параметрами. Ответ сервер отправляет в виде

html кода, который в свою очередь интерпретирует браузер. Описание

языка разметки html можно найти в стандарте: “RFC: 1866”

Динамическая асинхронная подзагрузка данных обеспечивается

посредством ajax запросов. Они также передаются методом post,

однако в этом случае сервер возвращает ответ в нотации json.

Полученный ответ разбирается посредством скриптов, написанных на

языке java script, и встраивается в DOM страницы. Описание формата json можно найти в стандарте: “RFC: 7159”.

* 1. Выходные данные
     1. Характер и организация выходных данных
     2. Формат, описание и способ кодирования выходных данных.

Система предусматривает всего два типа выходных файлов:

− HTML документы – основной формат выходных файлов.

Представляет собой страницы сайта.

− JSON – используется для передачи информационных объектов в

случае динамической асинхронной передачи данных.

* 1. Структура баз данных

Все данные, хранимые системой, будут сохранятся на сервере. Часть данных будет организована с помощь баз данных. Управление и доступ к базам данных будет осуществляется посредством СУБД PostgreSQL. На рисунке 3.5 представлена физическая схема базы данных.

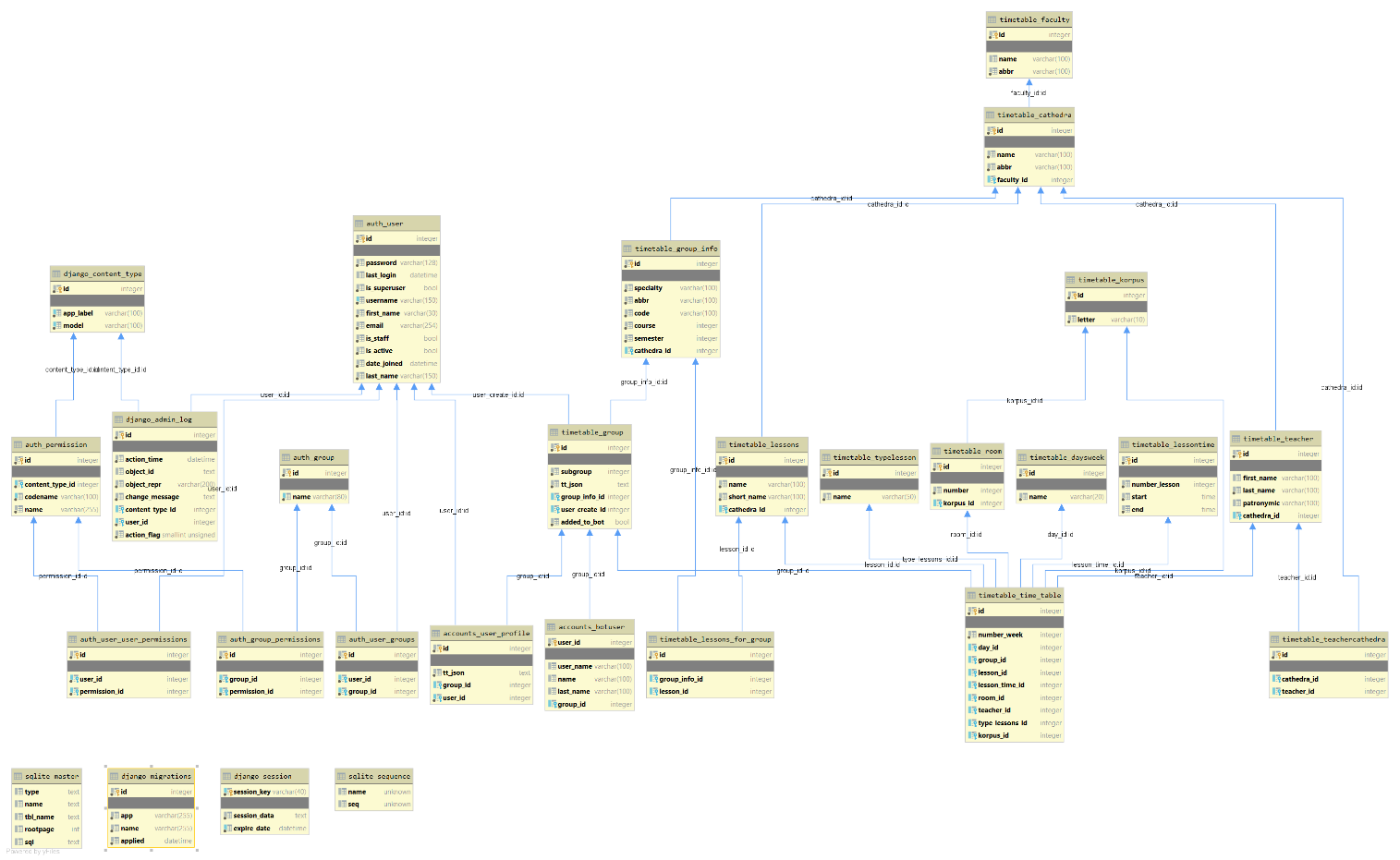


Рисунок 3.5 – Физическая схема базы данных

Для передачи данных между сервером и клиентом, будет использован формат данных json, описанный в стандарте RFC 8259.

1. Программная реализация и тестирование программного обеспечения
   1. Сценарий тестирования

Будет проводится следующие виды тестирования:

1. Модульное тестирование

Выполняется проверка работоспособности каждого модуля системы. Будет проводиться до тех пор, пока все неисправности не будут устранены.

1. Функциональное тестирование

Выполняется проверка работоспособности каждой функции системы. В течение недели проводится данное тестирование.

Испытаниям подвергнутся следующие функции:

* Авторизация пользователя;
* Просмотр календаря
* Просмотр/добавление/редактирование события
* Просмотр/добавление/редактирования мероприятия
  1. Протокол испытаний программного обеспечения
     1. Объект испытаний

Объектом испытаний является система многокритериального планирования деятельности студента вуза.

* + 1. Цель испытаний

Проверка надежности функционирования программы. Выявление критических ошибок, приводящих к аварийному завершению работы программы

* + 1. Требования к программе

Функционирование программы не должно приводить к сбою (фатальному нарушению работы системы). Организация диалога должна предусматривать защиту от ввода некорректных данных. Программа должна выдавать сообщения о любых возникших ошибках.

* + 1. Средства и порядок испытаний

Порядок тестирования модуля работы с расписанием:

1. Запускается программа
2. Выбирается модуль заполнения расписания
3. Выполняем необходимые действия

Порядок тестирования модуля работы со мероприятиями:

1. Запускается программа
2. Выбирается модуль работы с мероприятиями
3. Выбираем заполнение данных
4. Заполняем данные, сохраняем

Порядок тестирования модуля работы со календарем:

1. Запускается программа
2. Выбирается модуль работы с календарем
3. Выполняем необходимые действия
   * 1. Тестовые примеры

Некоторые тестовые примеры представлены в таблице 4.1

Таблица 4.1 — Тестовые примеры

| № | Действие | Ожидаемый результат | Полученный результат | |
| --- | --- | --- | --- | --- |
| 1 | Сценарий “Авторизация пользователя” | |  | |
|  | Ввести аутентификационные данные: [логин: “(логин зарегистрированного пользователя)”; пароль: “(неверный пароль пользователя)”] | Должно выводится сообщение об ошибке: “Неверный логин или пароль”. | Выводится сообщение об ошибке: “Неверный логин или пароль”. | |
|  | Ввести аутентификационные данные: [логин: “(не верный логин пользователя)”; пароль: “(верный пароль пользователя)”] | Должно выводится сообщение об ошибке: “Неверный логин или пароль”. | Выводится сообщение об ошибке: “Неверный логин или пароль”. | |
|  | Ввести аутентификационные данные: [(почта: “верная логин пользователя)”; пароль: “(верный пароль пользователя)”] | Должна открыться главная страница | Открывается главная страница | |
| 2 | Сценарий “Добавления события в календарь” | | | |
|  | Ввести данные: [одно из полей не заполнено] | Должно выводится сообщение об ошибке возле незаполненного поля: “Пустое поле”. | | Выводится сообщение об ошибке возле незаполненного поля: “Пустое поле”. |
|  | Ввести данные: [одно из полей заполнено неверными данными] | Должно выводится сообщение об ошибке возле незаполненного поля: “Неверные данные”. | | Выводится сообщение об ошибке возле незаполненного поля: “Неверные данные”. |
|  | Ввести данные: [все данные введены верно] | Должен открываться календарь | | Открывается календарь |
| 3 | Сценарий “Добавления мероприятия вуза” |  | |  |
|  | Ввести данные: [одно из полей не заполнено] | Должно выводится сообщение об ошибке возле незаполненного поля: “Пустое поле”. | | Выводится сообщение об ошибке возле незаполненного поля: “Пустое поле”. |
|  | Ввести данные: [одно из полей заполнено неверными данными] | Должно выводится сообщение об ошибке возле незаполненного поля: “Неверные данные”. | | Выводится сообщение об ошибке возле незаполненного поля: “Неверные данные”. |
|  | Ввести данные: [все данные введены верно] | Мероприятие вуза добавлено в базу | | Открывается список мероприятий |
| 4 | Сценарий «Создание расписания для группы» |  | |  |
|  | Выбрать данные из предложенных [одно из полей не выбрано] | Должно выводится сообщение об ошибке возле незаполненного поля: “Это поле обязательно”. | | Должно выводится сообщение об ошибке возле незаполненного поля: “Это поле обязательно”. |
|  | Нажать кнопку «Добавить»  [группа с выбранными параметрами уже существует] | Должно выводится сообщение об ошибке «Расписание группы уже существует» | | Выводится сообщение об ошибке «Расписание группы уже существует |
|  | Нажать кнопку «Добавить»  [группа создана впервые] | Расписание создается в базе | | Открывается список расписаний пользователя с добавленной в него новой группой. |
| 5 | Сценарий «Добавление пары для группы» |  | |  |
|  | Выбрать данные из предложенных [одно из полей не выбрано] | Должно выводится сообщение об ошибке возле незаполненного поля: “Это поле обязательно”. | | Должно выводится сообщение об ошибке возле незаполненного поля: “Это поле обязательно”. |
|  | Нажать кнопку «Добавить»  [Заполнены не все поля] | Кнопка не активна, пока не будут заполнены все поля и нажатие не должно ничего сделать | | Нажатие на кнопку не произвело никаких действий |
|  | Нажать кнопку «Добавить»  [Заполнены все поля] | Кнопка активна, нажатие добавляет пару в общую таблицу | | Нажатие на кнопку закрыло окно добавления, пара появилась в общей таблице |
| 6 | Сценарий «Добавления предмета группе» |  | |  |
|  | Нажатие на кнопку «Добавить» | Должно открыться окно добавления предмета с полем поиска | | Открылось окно добавления предмета с полем поиска |
|  | Ввод названия [введен несуществующий предмет] | Должно высвечиваться сообщение, о том, что предмет с таким название не найдет | | Появилось сообщение о невозможности найти предмет |
|  | Ввод названия [Введен существующий предмет] | Должно  появиться поле выбора предметов название, которые совпадает с предложенным и появиться кнопка «Добавить» в окне | | Появилось поле выбора предметов название, которые совпадает с предложенным, появилась кнопка «Добавить» |
|  | Нажатие на кнопку «Добавить» | Должно закрыться окно, а список предметов группы должен обновиться новым предметом | | Окно закрылось, а список предметов группы обновился новым предметом |
|  | Нажатие на кнопку «Крестик» напротив одного из предметов | Должен удалиться предмет, напротив которого нажат крестик | | Удаляемый предмет пропал из списка группы. |
| 7 | Сценарий «Добавление мероприятия вуза себе в календарь» |  | |  |
|  | Нажать на кнопку «Добавить» напротив мероприятия [время мероприятия совпадает с учебным занятием] | Должно появиться окно подтверждения с текстом «Вы уверены, что хотите добавить мероприятие во время учебного занятия» | | Появилось окно подтверждения |
|  | Нажать на кнопку «Добавить» напротив мероприятия [время мероприятия свободно в календаре пользователя] | Мероприятие добавилось в календарь пользователя | | Кнопка «Добавить изменилась на «Добавлено» |

Для подсчета показателей качества интерфейса использовался специальный инструмент CogTool.

Для оценки качества интерфейса администратора было проведено тестирование некоторых функций подсистемы. Эксперимент заключался в оценке эффективности пользовательского интерфейса для администратора. Администратору был поставлено несколько задач:

1. Создать расписание группы;
2. Заполнить в расписании одно занятие;
3. Удалить предмет для группы и добавить новый;

Результат эксперимента представлен в таблице 4.2.

Таблица 4.2 — Результат тестирования административной подсистемы

|  |  |
| --- | --- |
| Задача | Время, сек |
| Создать расписание группы | 9.5 |
| Заполнить расписание на одно занятие | 12.8 |
| Удалить предмет для группы и добавить новый предмет | 14.4 |

Для тестирования интерфейса пользовательской части системы были проведены эксперименты по оценке разработанного интерфейса в сравнении с интерфейсом систем аналогов.

В качестве объекта эксперимента тестирования пользовательского интерфейса была выбрана одна из основных при взаимодействии с системами планирования функций – Добавление события в Календарь. Результаты эксперимента представлены в таблице 4.3.

Результат эксперимента представлен на рисунке 11.

Таблица 4.2 — Результаты эксперимента сравнения с аналогами

|  |  |
| --- | --- |
| Задача | Время, сек |
| Ежедневник студента | 12.46 |
| Google | 13.54 |
| Trello | 18.2 |
| TodoIst | 12.67 |
| Excel | 20.43 |

Также в рамках реализации был создан чат-бот для социальной сети «Вконтакте». Чат-бот, проработал в боевом режиме на протяжении года и собрал аудиторию активных пользователей в 300 человек. Весь год, собиралась обратная связь с мнением студентов о боте, и дописывались новые функции по запросам пользователей.

* 1. Выводы

Проведенное тестирование программного средства показало нам, что в нем предусмотрены различные сообщения об ошибках, дающие понять пользователю, что он сделал не так.

В планах к данной разработке добавить модуль составления статистики, необходимого для контроля потраченного времени на различные сферы жизни. Исходя из этого и проведенного тестирования, можно сделать вывод, что система разработана на 65%.

Заключение

Описание применения программного обеспечения

Программная система для многокритериального планирования деятельности студента вуза, может использоваться учебными заведениями для повышения эффективности учебной и внеучебной деятельности.

Выводы

Таблица 5.1 Итоговая трассировка требований

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Бизнес-требования** | **Функциональные требования** | **Вариант использования** | **Сценарий тестирования** | **Комментарии** |
| Функциональная пригодность | Вход в систему | Актер: пользователь  Предусловия: пользователь не авторизован, пользователь зарегистрирован.  Основной поток событий:   1. Пользователь вводит логин и пароль для входа в систему 2. Пользователь после ввода верных данных для авторизации, если он зарегистрирован, авторизовывается под своей учетной записью.   Альтернативные потоки: Если пользователь не зарегистрирован, он регистрируется после чего переходит к пункту 1 данного сценария.  Постусловия: при успешном выполнении сценария пользователь получает доступ к функционалу системы. | Действия:  Ввести аутентификационные данные: [логин: “(логин зарегистрированного пользователя)”; пароль: “(неверный пароль пользователя)”]  Ожидаемый результат:  Должно выводится сообщение об ошибке: “Не верный логин или пароль”.  Действия:  Ввести аутентификационные данные: [логин: “(не верный логин пользователя)”; пароль: “(верный пароль пользователя)”]  Ожидаемый результат:  Должно выводится сообщение об ошибке: “Не верный логин или пароль”.  Действия:  Ввести аутентификационные данные: [(почта: “верная логин пользователя)”; пароль: “(верный пароль пользователя)”]  Ожидаемый результат:  Должна открыться главная страница |  |
| Функциональная пригодность | Работа с расписанием | Актер: Администратор  Предусловия: пользователь авторизовался.  Основной поток событий:   1. Администратор заполняет предмет 2. Администратор заполняет преподавателя 3. Администратор заполняет корпус и аудиторию, а система проверяет данные на корректность 4. Администратор выбирает тип пары, а система проверяет данные на корректность 5. Если все данные корректны, то система подает запрос на добавление записи в базу данных.   Альтернативные потоки: если какие-то данные не корректны, то система сообщает об этом администратору.  Постусловия: при успешном выполнении сценария:   * Если данные не дублируются, в базе данных появляется новая запись с парой. | Действия:  Ввести данные: [одно из полей не заполнено]  Ожидаемый результат:  Должно выводится сообщение об ошибке возле незаполненного поля: “Пустое поле”.  Действия:  Ввести данные: [некорректные данные]  Ожидаемый результат:  Должно выводится сообщение об ошибке возле поля: «Некорректные данные».  Действия:  Ввести данные: [пустая дата]  Ожидаемый результат:  Должно выводится сообщение об ошибке возле поля: «Дата не выбрана.». |  |

Список литературы

1. Джон Дакетт. HTML и CSS. Разработка и дизайн веб-сайтов

/ Джон Дакетт. - Эксмо, 2017. - 480 с.

2. Джон Дакетт. Javascript и jQuery. Интерактивная веб-

разработка / Джон Дакетт. - Эксмо, 2017. - 640 с.

3. HTML 5.2 W3C Recommendation [Электронный ресурс] —

2017 — Режим доступа : https://www.w3.org/TR/html5/

4. Ересь А.В. Создание и отправка форм с использованием

фреймворка YII2 // Постулат. 2019. № 1-1 (39). С. 45.

5. Морозов А.О., Рыбанов А.А. Экспертная оценка

программных продуктов для расчета метрических характеристик

физической схемы базы данных // Современные научные

исследования и инновации. 2015. № 1. Ч. 1 [Электронный ресурс]

— 2018 — Режим доступа :

http://web.snauka.ru/issues/2015/01/42101

6. Прозорова А.П. Управление тестированием / Прозорова А.П.

// Электронный журнал: наука, техника и образование. — 2017. —

№ 2 (12). — С. 194-197.

7. Абрамова О.Ф. Case-технологии: изучать или исключить? /

О.Ф. Абрамова // Alma mater (Вестник высшей школы). — 2012. —

№ 9. — С. 109-110.

8. Абрамова О.Ф., Белова С.В. Использование мультимедийных

технологий в процессе обучения дисциплине

«компьютерная графика» / О.Ф. Абрамова, С.В. Белова // Успехи

современного естествознания. — 2012. — № 3. — С. 90.

9. Сагиндыкова, А.С. Актуальность дистанционного

образования / А.С. Сагиндыкова // Молодой ученый. – 2015. – № 20.

– С. 495–498.

10. Главная – yii framework [Электронный ресурс] — 2019

— Режим доступа : https://www.yiiframework.com/doc/guide/2.0/ru

11. Сагиндыкова, А.С. Актуальность дистанционного

образования / А.С. Сагиндыкова // Молодой ученый. – 2015. – № 20.

– С. 495–498.

12. Главная – yii framework [Электронный ресурс] — 2019

— Режим доступа : https://www.yiiframework.com/doc/guide/2.0/ru

13. Программных платформ (php-фреймворков) / Косарев Н.О.

// В сборнике: Фундаментальные и прикладные научные

исследования: актуальные вопросы, достижения и инновации

сборник статей XIV Международной научно-практической

конференции: — в 2 ч. — 2018. — С. 37-40.

14. Анализ языков программирования для веб-разработки: php,

python и ruby / Паршикова С.Р., Кузина В.В. // Образование и наука

в современном мире. Инновации. — 2016. — № 6-1. — С. 195-202.

15. Методика разработки web -сайта средствами mysql, php и

css / Тажимаматова М, Токтосунова А, Самидинов А // Вестник

Жалал-Абадского государственного университета. – 2018. – №

2(37). – С. 142–4146.

16. Обзор фреймворков / Назиков Б.А., Гибадуллина Э.А. // В

сборнике: РОЛЬ И МЕСТО ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В

СОВРЕМЕННОЙ НАУКЕ сборник статей Международной научно-

практической конференции. – 2019. – С. 109-11

17. PHP, RUBY, PYTHON - на чем остановить свой выбор

студенту при изучении веб-разработки? / Потапова Н.В.,

Большакова А.А. // В сборнике: Новые информационные

технологии в образовании и науке Материалы X международной

научно-практической конференции. – 2017. – С. 409–413.

19. Джон Дакетт. Javascript и jQuery. Интерактивная веб

разработка / Джон Дакетт. - Эксмо, 2017. - 640 с.

20. Джон Дакетт. HTML и CSS. Разработка и дизайн веб-сайтов /

Джон Дакетт. - Эксмо, 2017. - 480 с