% Место для титульного листа %

% Место для задания %

**Реферат**

Расчетно-пояснительная записка квалификационной работы бакалавра содержит 75 страниц (с приложениями 90 страниц), 27 таблиц и 12 иллюстраций. В процессе выполнения данной работы были использованы 25 литературных и Интернет источников.

Объектом разработки является интерактивная система поддержки процесса обучения студентов с использованием облачных технологий.

Квалификационная работа на тему «интерактивная система поддержки процесса обучения студентов с использованием облачных технологий» посвящена разработке системы для студентов специальности «информатика и вычислительная техника», обладающих определенными навыками, умениями и компетенции, которые можно отнести к «лидерским». Система позволяет студентам находить нужную, полезную и актуальную информацию. В данной системе используется исследовательский подход к подбору информации, а также экспертная система для выявления предпочтений пользователя (студента).

Цель разработки – создание простого, интуитивно-понятного и клиент-ориентированного инструмента для студентов, содержащего актуальную информацию для развития профессиональных и личностных компетенций в соответствии с их персональными предпочтениями.

В процессе выполнения квалификационной работы бакалавра было проведено исследование предметной области, анализ аналогов и прототипов, разработка структуры и архитектуры системы, построена инфологическая и даталогическая модель, разработан интерфейс пользователя, разработан веб-сервис, разработана экспертная система, проведено сравнение с другими веб-сервисами.

В исследовательской части работы проведен что-то там и анализ, *выбранного языка программирования* (?). Пояснительная записка содержит N приложений.

С**одержание**

[Введение 7](#_Toc40398722)

[1. Конструкторская часть 8](#_Toc40398723)

[1.1. Общетехническое обоснование разработки 8](#_Toc40398724)

[1.1.1. Постановка задачи проектирования 8](#_Toc40398725)

[1.1.2. Описание предметной области 9](#_Toc40398726)

[1.1.3. Перечень процессов, подлежащих автоматизации 9](#_Toc40398727)

[1.1.4. Функциональные задачи системы 10](#_Toc40398728)

[1.1.5. Выбор и обоснование критериев качества 10](#_Toc40398729)

[1.1.6. Анализ аналогов и прототипов 12](#_Toc40398730)

[1.2. Разработка программного изделия 15](#_Toc40398731)

[1.2.1. Выбор облачной платформы 15](#_Toc40398732)

[1.2.2. Выбор облачной базы данных. 17](#_Toc40398733)

[1.2.3. Выбор облачного сервера 18](#_Toc40398734)

[1.2.4. Выбор облачной модели 18](#_Toc40398735)

[1.2.5. Выбор средств разработки 20](#_Toc40398736)

[1.2.5. Архитектура системы 21](#_Toc40398737)

[1.2.6. Проектирование базы данных 24](#_Toc40398738)

[1.2.7. Выбор программных средств 40](#_Toc40398739)

[Любая ОС, поддерживающая работу с браузером, который поддерживает HTML5, CSS3, JS5; 40](#_Toc40398740)

[Один из перечисленных веб-браузеров: 40](#_Toc40398741)

[- Облачный сервер 40](#_Toc40398742)

[1.2.8. Выбор аппаратных средств 40](#_Toc40398743)

[ Процессор с частотой не менее 1 ГГц; 40](#_Toc40398744)

[ Размер оперативной памяти не менее 512 МБ; 40](#_Toc40398745)

[ Видеоадаптер и монитор, способные обеспечить графический режим не менее 1024 x 768 точек с 16 битной цветопередачей; 40](#_Toc40398746)

[ Система устройства хранения данных не менее 1 ГБ; 40](#_Toc40398747)

[ Наличие манипулятора «мышь» или другого указывающего устройства; 41](#_Toc40398748)

[ Наличие клавиатуры или её экранного аналога; 41](#_Toc40398749)

[ Обеспечение доступа в интернет. 41](#_Toc40398750)

[1.2.9. Структура программного изделия 41](#_Toc40398751)

[1.2.10. Алгоритм экспертной системы 45](#_Toc40398752)

[2. Технологическая часть 46](#_Toc40398753)

[2.1. Граф диалога 46](#_Toc40398754)

[2.2. Разработка интерфейса взаимодействия с пользователем 46](#_Toc40398755)

[2.2.1. Экранные формы веб-сервиса 47](#_Toc40398756)

[3. Исследовательская часть 53](#_Toc40398757)

[Выбор способа развертывания приложения на облачной платформе Azure 53](#_Toc40398758)

[Заключение 66](#_Toc40398759)

[Список литературы 67](#_Toc40398760)

# 

# Введение

Квалификационная работа на тему «интерактивная система поддержки процесса обучения студентов с использованием облачных технологий» (в дальгейшем ИСППОС) посвящена разработке системы для студентов.

Цель разработки – создание простого, интуитивно-понятного и клиент-ориентированного инструмента для студентов, содержащего актуальную информацию для развития профессиональных и личностных компетенций в соответствии с их персональными предпочтениями.

ТУТ БУДЕТ ЕЩЕ ТЕКСТ

# 1. Конструкторская часть

## 1.1. Общетехническое обоснование разработки

### 1.1.1. Постановка задачи проектирования

Разрабатываемая интерактивная система поддержки процесса обучения студентов с использованием облачных технологий должна предоставлять удобный и простой интерфейс для поиска необходимой информации, опроса пользователей (студентов) с помощью экспертной сиситемы , чтения статей и новостей, а также программный интерфейс для добавления статей, рубрик и категорий; анализа данных, поступающих от пользователся с помощью прохождения опроса или обратной связи.

Таким образом получаем следующие задачи проектирования:

1. разработать веб-сервис для прохождения пользователем экспертной системы;
2. разработать веб-сервис для возможности написания обратной связи пользователем;
3. разработать программный интерфейс для анализа полученных данных от пользователя с помощью экспертной системы и системы обратной связи; для просмотра индивидуальной подборки пользователем; для просмотра общей подборки для всех пользователей;
4. разработать веб-сервис для просмотра статей, ранжированных по категориям и рубрикам;
5. в веб-сервисе предусмотреть возможность просмотра пользователем своей индивидуальной подборки;
6. предусмотреть возможность смены предпочтений пользователя и прохождения пользователем экспертной системы еще раз;

### 1.1.2. Описание предметной области

*Многие студенты сталкиваются с проблемой…*

*ТУТ БУДЕТ ЕЩЕ ТЕКСТ*

### 1.1.3. Перечень процессов, подлежащих автоматизации

При разработке веб-сервиса, автоматизации подлежат следующие процессы:

1. Регистрация пользователей;
2. Авторизация пользователей;
3. Анализ данных полученных с помощью прохождения пользователем экспертной системы;
4. Сохранение результатов прохождения экспертной системы пользователем в БД;
5. Отображение личной реккомендательной подборки статей каждому пользователю
6. Отображение общей реккомендательной подборки статей каждому пользователю в соответсвии с датой публикации
7. Отображение статей в каждую категорию и рубрику в соответсвии с датой публикации
8. Добавление, редактирование, удаление категорий
9. Добавление, редактирование, удаление рубрик
10. Добавление, редактирование, удаление статей
11. Интеграция Яндекс.новостей в категорию «новости»

### 1.1.4. Функциональные задачи системы

1. Регистрация пользователей
2. Авторизация пользователей
3. Завершение пользовательской сессии (выход)
4. Просмотр категории новости
5. Просмотр категории развитие
6. Просмотр категории образование, рубрики лекции
7. Просмотр категории образование, рубрики зарубеж
8. Просмотр категории события, рубрики конференции
9. Просмотр категории события, рубрики мероприятия
10. Просмотр категории стажировки, рубрики IT
11. Просмотр категории стажировки, рубрики маркетинг
12. Прохождение экспертной системы в виде опроса
13. Просмотр личной реккомендательной подборки статей
14. Просмотр общей подборки статей

### 1.1.5. Выбор и обоснование критериев качества

Для проектируеой системы приоритетными качествами являются следующие критерии:

* 1. Удобство доступа к системе;
  2. Понятность интерфейса;
  3. Кроссплатформенность;
  4. Наличие нужной информации;
  5. Наличие индивидуальных рекомендаций;
  6. Наличие общих рекомендаций;
  7. Возможность расширения функционала.
  8. Качество документации;

**Удобство доступа** к системе означет возможность воспользоваться системой из любой точки мира, имея только доступ в интернет. Интерфейс, базирующийся на web-страницах является более комфортным для пользователя, т.к. нет необходимости устанавливать специальную программу – нужно только наличие браузера.

**Понятнойсть интерфейса** означает понятность и простоту работы с веб-интерфейсом. Это позволяет минимизировать время, потраченное на поиск нужной информации и получить максимальную эффективность результата.

**Кроссплатформенность** включает в себя возможность комфортного доступа к системе с помощью любого устройства – компьютера, ноутбука, планшета или телефона.

**Наличие нужной информации** подрузамевает под собой наличие в системе различной информации, разделенной по тематикам или категориям, отвечающей потребностям студентов.

**Наличие индивидуальных рекомендаций** означает возможность просмотра индивидуальной личной подборки каждым пользователям в соответствии с его предпочтениями.

**Наличие общих рекомендаций** включает в себя наличие рекоммендательной подборки, доступной всем пользователям.

**Возможность расширения функционала** – при проектировании представленной системы предусматривалась способность к дальнейшему развитию путем добавления новых категорий и рубрик. Упрощению программного веб-сервиса с целью обеспечения новых возможностей с точки зрения хранения, обработки и анализа информации. Так же предусматривалось изменение отображения некоторых частей системы в более удобном для пользвателя формате для повышения эффективности работы пользователя с системой.

**Качество документация** предполагает её простоту, очередность и полноту. Документация должна содержать в себе такие части как обеспечение программной и эксплуатационную части. Также следует отметить необходимость соответствия языка документации языку пользователей (студентов) системы. При оценке по данному критерию необходимо принимать во внимание цели и задачи проектирования и создания данного программного продукта.

### 1.1.6. Анализ аналогов и прототипов

В качестве аналогов для сравнения с данным проектом были выбраны четыре похожих системы, найденных в интернете: x-student.ru (XS), tmstudent.ru (TS), changellenge.com (CH) и studroom.ru (SR)

Таблица 1. Сравнение аналогов и прототипов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Критерии** | ИСППОС | XS | TS | CH | SR |
| K1 | Удобство доступа  к системе; | оч хор | удовл | оч хор | отл | плохо |
| K2 | Понятность интерфейса | отл | удовл | оч хор | оч хор | плохо |
| K3 | Кроссплатфор-  менность | оч хор | плохо | хор | хор | плохо |
| K4 | Наличие нужной информации | отл | удовл | оч хор | отл | удовл |
| K5 | Наличие индивидуальных рекомендаций | да | нет | нет | нет | нет |
| K6 | Наличие общих рекомендаций | да | нет | нет | да | нет |

Переведём вербальную шкалу оценок в числовую по следующей таблице:

Таблица 2. Перевод вербальной шкалы оценок в числовую

|  |  |
| --- | --- |
| **Вербальная оценка** | **Числовая оценка** |
| отлично / да | 1 |
| очень хорошо | 0.8 |
| хорошо | 0.6 |
| удовлетворительно | 0.4 |
| плохо | 0.2 |
| отсутствует / нет | 0 |

Оценим важность каждого критерия, воспользовавшись методом парного сравнения критериев:

Таблица 3. Парное сравнение критериев

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерии** | **K1** | **K2** | **K3** | **K4** | **K5** | **K6** |  |  |
| **K1** | 1 | 0 | 0.5 | 0.5 | 0.5 | 0.5 | 3.0 | 0.143 |
| **K2** | 1 | 1 | 1 | 0.5 | 0.5 | 0.5 | 4.5 | 0.214 |
| **K3** | 0.5 | 0 | 1 | 0 | 0.5 | 0.5 | 2.5 | 0.119 |
| **K4** | 0.5 | 0.5 | 1 | 1 | 1 | 1 | 5.0 | 0.238 |
| **K5** | 0.5 | 0.5 | 0.5 | 0 | 1 | 0.5 | 3.0 | 0.143 |
| **K6** | 0.5 | 0.5 | 0.5 | 0 | 0.5 | 1 | 3.0 | 0.143 |

Итоговая сумма:

Вычисляем веса критериев:

Таблица 4.1 Числовые значения и коэффициенты критериев

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Коэффициент** | ИСППОС | XS | TS | CH | SR |
| K1 | 0.143 | 0.8 | 0.4 | 0.8 | 1 | 0.2 |
| K2 | 0.214 | 1 | 0.4 | 0.8 | 0.8 | 0.2 |
| K3 | 0.119 | 0.8 | 0.2 | 0.6 | 0.6 | 0.2 |
| K4 | 0.238 | 1 | 0.4 | 0.8 | 1 | 0.4 |
| K5 | 0.143 | 1 | 0 | 0 | 0 | 0 |
| K6 | 0.143 | 1 | 0 | 0 | 1 | 0 |
|  | **1** |  |  |  |  |  |

Таблица 4.2 Числовые значения и коэффициенты критериев

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Коэффициент** | ИСППОС | XS | TS | CH | SR |
| K1 | 0.143 | 0.114 | 0.057 | 0.114 | 0.143 | 0.029 |
| K2 | 0.214 | 0.214 | 0.086 | 0.171 | 0.171 | 0.043 |
| K3 | 0.119 | 0.095 | 0.024 | 0.071 | 0.071 | 0.024 |
| K4 | 0.238 | 0.238 | 0.095 | 0.190 | 0.238 | 0.095 |
| K5 | 0.143 | 0.143 | 0.000 | 0.000 | 0.000 | 0.000 |
| K6 | 0.143 | 0.143 | 0.000 | 0.000 | 0.143 | 0.000 |
|  | **1** | 0.947 | 0.262 | 0.546 | 0.766 | 0.191 |

Используя метод максимума взвешенной суммы получаем итоговые значения:

Таким образом, благодаря балансу между удобством для пользователя и образовательной составляющей, лучшим вариантом по указанным критериям является разрабатываемая «Интерактивная система поддержки процесса обучения студентов с использованием облачных технологий».

## 1.2. Разработка программного изделия

Для разработки ИСППОС будет использоваться:

1. Microsoft Azure — открытая и гибкая платформа облачных вычислений.
2. Облачная база данных mysql для хранения данных пользователей, категорий, рубрик, статей, результатов прохождения экспертной системы.
3. Облачный сервер для обслуживания запросов пользователей
4. Облачная модель Microsoft Azure - платформа как услуга (PaaS)
5. Язык программирования Python с использованием Django Framework.

### 1.2.1. Выбор облачной платформы

В качестве облачной платформы была выбрана Microsoft Azure. Microsoft Azure реализует две облачные модели — платформы как сервиса (PaaS) и инфраструктуры как сервиса (IaaS).

Выбор платформы производился среди нескольких популярных облачных платформ: Microsoft Azure, Amazon, Google Cloud

Таблица 5. Сравнение СУБД

|  |  |  |  |
| --- | --- | --- | --- |
|  | Microsoft Azure | Amazon | Google Cloud |
| Пробная версия | Пробная версия 30 дней | Пробная версия 30 дней | Пробная версия 60 дней |
| Оплата | Оплата только за те ресурсы, которые используются | Несколько моделей оплаты | Несколько моделей оплаты |
| Округление часов использования | Округление по минутам | Округление по часам | Округление по 10 минут |
| Наличие студенческой лицензии | да | нет | нет |
| Возможности развертки приложений с помощью платформы | да | нет | нет |
| Вид платформы | Paas, Iaas, Saas | Paas, Iaas, Saas | Iaas |
| Наличие документации | да | да | да |
| Наличие тех поддержки | да | да | да |
| Простота использования | Очень просто | Сложно | Просто |

Из облачных платформ наилучшим вариантом является Microsoft Azure, потому что она обладает всеми необходимыми требованиями: бесплатность (возможность приобритения студенческой лицензии), возможность развертки приложений с помощью платформы, простота использования.

**Microsoft Azure** реализует две облачные модели — платформы как сервиса (PaaS) и инфраструктуры как сервиса (IaaS). Работоспособность платформы Windows Azure обеспечивает сеть глобальных дата-центров Microsoft.

Основные особенности данной модели:

* оплата только потреблённых ресурсов;
* общая, многопоточная структура вычислений;
* абстракция от инфраструктуры.

Microsoft Azure как PaaS обеспечит не только все базовые функции операционной системы, но и дополнительные: выделение ресурсов по требованию для неограниченного масштабирования, автоматическую синхронную репликацию данных для повышения отказоустойчивости, обработку отказов инфраструктуры для обеспечения постоянной доступности и многое другое. ЦИТАТКА Из документации <https://azure.microsoft.com/ru-ru/overview/what-is-azure/>

### 1.2.2. Выбор облачной базы данных.

**MySQL** использует интеллектуальную технологию кластеризации и сквозную защиту, а также поддерживает администрирование базы данных через PhpAdmin. НАДО ЧТО-ТО ДОПИСАТЬ

**База данных SQL** — это облачная служба баз данных, созданная для разработчиков приложений, которая позволяет легко масштабироваться без простоев и эффективно доставлять приложения. Встроенные помощники быстро определяют уникальные характеристики приложения и динамически адаптируются для максимально высокой производительности, надежности и защиты данных.

**Microsoft Azure DocumentDB** — это полностью управляемая, быстрая служба баз данных NoSQL с возможностью неограниченного глобального масштабирования. DocumentDB включает в себя встроенную высокую доступность, возможность распространения в масштабе всей планеты "под ключ", обеспечивает прогнозируемую производительность, неограниченное эластичное масштабирование и низкие задержки (на уровне одной цифры, в миллисекундах) и поставляется с соглашением об уровне обслуживания на время бесперебойной работы уровня 99,99 %.

### 1.2.3. Выбор облачного сервера

В качестве облачного сервера Microsoft Azure предлагает - SQL Server . Это пустой логический сервер, на котором могут размещаться базы данных и пулы эластичных баз данных для базы данных SQL. Есть возможность размещения баз данных хранилища данных SQL или использовать его в качестве удаленной конечной точки для базы данных SQL Server Stretch.

### 1.2.4. Выбор облачной модели

Microsoft Azure предоставляет три варианта облачной модели:

- Программное обеспечение как услуга (SaaS)

- Платформа как услуга (PaaS)

- Инфраструктура как услуга (IaaS)

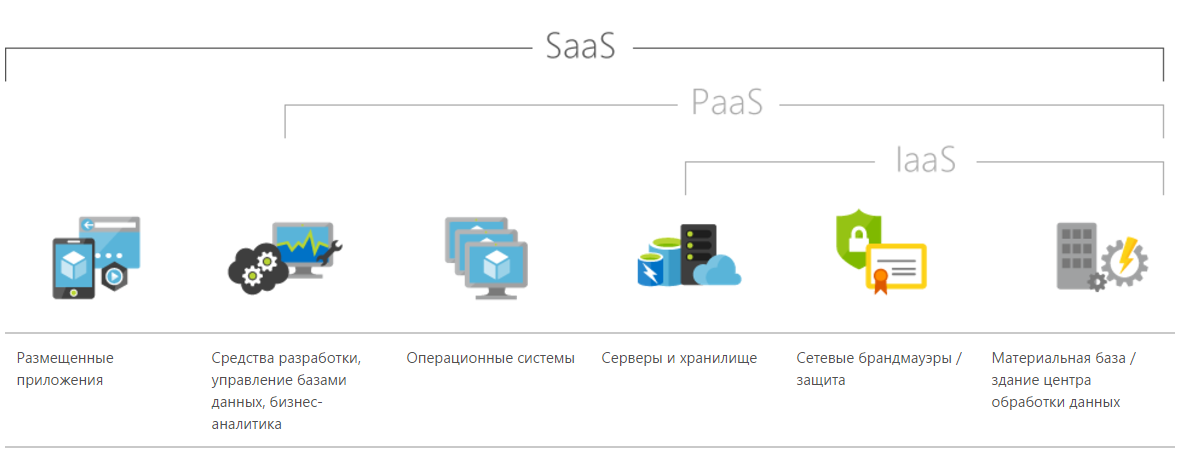


Рисунок - Архитектура облачного сервиса

**Программное обеспечение как услуга (SaaS)**

Программное обеспечение как услуга (SaaS) позволяет пользователям подключаться к облачным приложениям и использовать их через Интернет. SaaS предоставляет полноценный набор программного обеспечения. Вся базовая инфраструктура, ПО промежуточного слоя, программное обеспечение приложений и данные приложений находятся в центре обработки данных поставщика. Поставщик служб управляет оборудованием и программным обеспечением на основе соответствующего соглашения об обслуживании и обеспечивает доступность и безопасность приложений и данных.

**Инфраструктура как услуга (IaaS)**

Инфраструктура как услуга (IaaS) — это вычислительная инфраструктура, мгновенно выделяемая и управляемая через Интернет. IaaS помогает избежать затрат и трудностей, связанных с приобретением собственных физических серверов и другой инфраструктуры центра обработки данных и управлением этой инфраструктурой. Каждый ресурс предоставляется как отдельный компонент службы, и необходимо арендовать только конкретный компонент на определенное время. Поставщик служб облачных вычислений управляет инфраструктурой. Приобретение, установка, настройка, упраление программным обеспеченем происходит самостоятельно (включая операционные системы, ПО промежуточного слоя и приложения).

**Платформа как услуга (PaaS)**

Платформа как услуга (PaaS) — это полноценная среда разработки и развертывания в облаке с ресурсами. Как и в случае с IaaS, PaaS включает инфраструктуру (серверы, хранилище и сетевое оборудование), а также ПО промежуточного слоя, средства разработки, бизнес-аналитику (BI), службы системы управления базами данных и другое. Услуга PaaS предназначена для поддержки полного жизненного цикла веб-приложения: разработки, тестирования, развертывания, управления и обновления. Управление разработанными приложениями и службами происходит самостоятельно, а поставщик облачных служб обычно управляет всем остальным.

Для разработки «интерактивная система поддержки процесса обучения студентов» используется модель Платформа как услуга (PaaS)

### 1.2.5. Выбор средств разработки

Проект содержит части, имеющие различные требования к средствам разработки.

Microsoft Azure предоставляет 6 вариантов развертывания приложений: ASP.NET, ASP.NET Core, Java, Node.Js, Php и Python. В ASP.NET, ASP.NET Core и Pythin возможно прямое развертывание из Visual Studio или системы управления версиями, что достаточно удобно.

В ходе анализа средством для разработки был выбран Python/Django, по таким причинам как:

1. Python, как язык программирования имеет обширную библиотеку

классов, хорошую документацию и достаточно компактный и интуитивно понятный синтаксис.

1. Хорошая документация конкретно у Django (множество примеров,

объяснений и самое главное — открытый исходный код, который очень хорошо написан).

1. Встроенный ORM (Object-relational mapper). Безусловно, существуют

средства раработки с более мощным ORM, но для данных задач возможностей Django достаточно.

1. Наличие интерфейса администратора.
2. Поддержка MTV (Model-Template-View). Он позволяет хорошо

отделять бизнес-логику от дизайна.

1. Django работает достаточно быстро.

Более подробно выбор средства разработке рассматривается в исследовательской части в пункте 3.2.

### 1.2.5. Архитектура системы

Архитектура системы состоит из трех узлов:

1. Клиент (веб-интерфейс)

написан на JavaScript, HTML и CSS

1. Облачный сервер

SQL сервер

1. Облачная база данных

MySQL

**Принцип работы данной архитектуры:**

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

Архитектура Django похожа на архитектуру «Модель-Представление-Контроллер» (MVC). Контроллер классической модели MVC примерно соответствует уровню, который в Django называется Представление (англ. View), а презентационная логика представления реализуется в Django уровнем шаблонов (англ. Template). Из-за этого уровневую архитектуру Django часто называют «Модель-Шаблон-Представление» (MTV).



Рисунок - Архитектура фреймворка

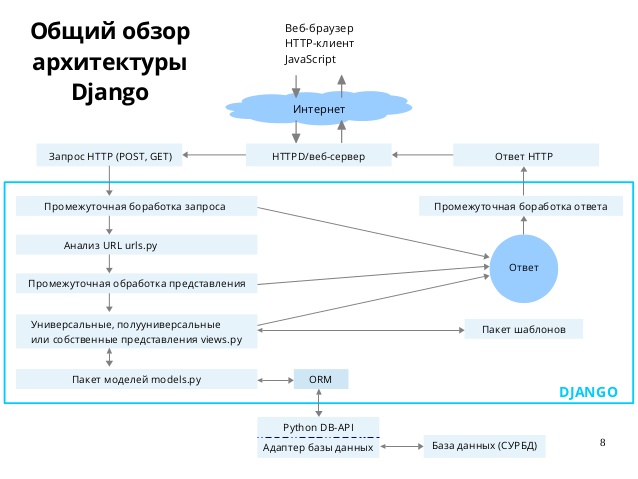


Рисунок - Архитектура системы

### 1.2.6. Проектирование базы данных

#### 1.2.6.2. Инфологическая модель

**Теоретическая часть**

На этапе анализа предметной области была проведена предварительная структуризация объектов предметной области и выделение её главных составляющих. Далее решается вопрос о том, какая информация об объектах должна храниться в базе данных и как ее представить с помощью данных. Сутью этого этапа проектирования базы данных является установление соответствия между состоянием предметной области, его восприятием и представлением в базе данных.

Для описания инфологической схемы используются диаграммы «сущность – связь» (ER-диаграмма). Такая диаграмма является методом представления информационной структуры базы данных в графическом виде для более простого и наглядного отображения основных компонентов конкретного проекта базы данных. Диаграмма содержит информацию о сущностях системы и способах их взаимодействия, включает идентификацию объектов, важных для предметной области (сущностей), свойств этих объектов (атрибутов) и их отношений с другими объектами (связей).

*Сущность* – поименованная конструкция для моделирования объектов предметной области. Это собирательное понятие, некоторая абстракция реально существующего объекта, процесса или явления, о котором необходимо хранить информацию в базе данных. Каждый экземпляр сущности должен быть уникален. Для этого вводят первичные и вторичные ключи.

Первичный ключ (Primary Key) – это атрибут (или группа атрибутов), однозначно идентифицирующий экземпляр сущности. При этом значение идентифицирующего атрибута находится во взаимнооднозначном соответствии с экземплярами сущности. Первичный ключ должен удовлетворять ряду требований:

* 1. Уникальность. Два экземпляра сущности не должны иметь одинаковых значений ключа.
  2. Минимум атрибутов в ключе. Составной первичный ключ (группа атрибутов) не должен содержать ни одного атрибута, удаление которого не приводило бы к утрате уникальности.

Вторичный (или внешний) ключ (Foreign Key) – он появляется в процессе установления связей между сущностями. Это описательный атрибут, который назначается в дополнение к идентифицирующему атрибуту и служит не для идентификации этого уникального экземпляра, а для выделения из набора тех объектов, которые обладают интересующими нас свойствами. Этот ключ часто называют "ключом поиска". Ограничения на состав вторичных ключей определяются логикой запроса

*Атрибут* – поименованная конструкция для моделирования свойств объекта предметной области (сущности). Основное назначение атрибута – описание свойства сущности, а также идентификация экземпляра сущностей.

*Связь* – поименованная конструкция для моделирования отношений между объектами (сущностями). При анализе связей между сущностями могут встречаться бинарные (между двумя сущностями), тернарные (между тремя сущностями) и, в общем случае n-арные связи. Может также встречаться унарные (рекурсивные) связи, когда экземпляр определенного типа сущности связан с другим экземпляром той же самой сущности. Наиболее часто встречаются бинарные связи. При проведении классификации видов связей обычно выделяют следующие виды связей:

* отображение типа 1:1 (связь "один-к-одному");
* отображение типа 1:М (связь "один-ко-многим");
* отображение типа М:1 (связь "многие-к-одному");
* отображение типа М:М (связь "многие-ко-многим").

Различают зависимые и независимые сущности. Тип сущности определяется связью между сущностями.

Независимая сущность – обычная сущность, которая не зависит при идентификации от других объектов в модели, или просто независимая, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.

Зависимая сущность – это дочерняя сущность, уникальность которой зависит от атрибута внешнего ключа, или просто сущность, у которой однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности.

Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности, и сущности, которые не могут быть идентифицированы без использования ключа родителя (сущности, зависящие от идентификации).

Идентифицирующая связь – это связь, которая характеризуется глагольной формой со сказуемым "должен", где подлежащим является имя дочерней таблицы. Она используется, когда требуется, чтобы внешний ключ передавался в дочернюю сущность (и, в результате, создавал зависимую сущность). Идентифицирующая связь устанавливается между независимой (родительской – начало связи) и зависимой (дочерней – конец связи) сущностями.

Очень важно обратить внимание на то, что при установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Происходит миграция атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ (foreign key).

Неидентифицирующая связь – это связь, которая характеризуется глагольной формой со сказуемым "может", где подлежащим является имя дочерней таблицы. Она используется для отображения другого типа передачи атрибутов внешних ключей – передача в область данных дочерней сущности. Очень важно обратить внимание на то, что при установлении неидентифицирующей связи происходит миграция атрибутов первичного ключа родительской сущности не в состав первичного ключа дочерней сущности, а в состав неключевых атрибутов дочерней сущности.

Существенно, что при идентифицирующей связи внешний ключ в дочерней сущности не может принимать значение NULL. Если внешний ключ должен существовать, то это означает, что запись в дочерней сущности может существовать только при наличии ассоциированной с ним родительской записи.

**Сущности и их атрибуты**

Таблица 5. Сущность «Пользователь»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID пользователь | Числовой | Первичный (PK) |
| ID группы | Числовой | Внешний (FK) |
| Фамилия | Текстовый |  |
| Имя | Текстовый |  |
| Отчество | Текстовый |  |
| Телефон | Числовой |  |
| Email | Текстовый |  |
| ВУЗ | Текстовый |  |
| Пол | Текстовый |  |
| Курс | Числовой |  |
| Логин | Текстовый |  |
| Пароль | Текстовый |  |
| Роль | Текстовый |  |

Таблица 6. Сущность «Личная подборка»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID Подборки | Числовой | Первичный (PK) |
| ID Пользователя | Числовой | Внешний (FK) |
| ID статья short | Числовой | Внешний (FK) |
| ID рубрика | Числовой | Внешний (FK) |
| ID категория | Числовой | Внешний (FK) |
| ID группы | Числовой | Внешний (FK) |
| Дата личной подборки | Дата и время |  |

Таблица 7. Сущность «Аннотация статьи»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID статья short | Числовой | Первичный (PK) |
| ID рубрика | Числовой | Внешний (FK) |
| ID категория | Числовой | Внешний (FK) |
| Название аннотации | Текстовый |  |
| Текст сокращенный | Текстовый |  |

Таблица 8. Сущность «Категория»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID категория | Числовой | Первичный (PK) |
| Название | Текстовый |  |
| Тэг | Текстовый |  |
| Актуальность категории | Числовой |  |
| Посещаемость категории | Числовой |  |

Таблица 9. Сущность «Рубрика»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID рубрика | Числовой | Первичный (PK) |
| ID категория | Числовой | Внешний (FK) |
| Тэг рубрика | Текстовый |  |
| Актуальность рубрики | Числовой |  |
| Посещаемость рубрики | Числовой |  |

Таблица 10. Сущность «Выбор пользователя»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID выбор пользователя | Числовой | Первичный (PK) |
| ID пользователь | Числовой | Внешний (FK) |
| ID выбор | Числовой | Внешний (FK) |
| ID вопрос | Числовой | Внешний (FK) |
| ID группы | Числовой | Внешний (FK) |
| ID категория | Числовой | Внешний (FK) |
| ID рубрика | Числовой | Внешний (FK) |
| ID статья | Числовой | Внешний (FK) |
| ID статья short | Числовой | Внешний (FK) |

Таблица 11. Сущность «Опрос»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID вопрос | Числовой | Первичный (PK) |
| Текст | Текстовый |  |
| Дата | Дата и время |  |

Таблица 12. Сущность «Выбор»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID выбор | Числовой | Первичный (PK) |
| ID вопрос | Дерево | Внешний (FK) |
| Текст | Числовой |  |
| Количество | Дата и время |  |

Таблица 13. Сущность «Исследование»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID исследование | Числовой | Первичный (PK) |
| Цель | Текстовый |  |
| Текст исследования | Текстовый |  |
| Даты исследованич | Дата и время |  |

Таблица 14. Сущность «Общая подборка»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID подборка | Числовой | Первичный (PK) |
| ID исследования | Числовой | Внешний (FK) |
| Количество | Числовой |  |
| Текст | Текстовый |  |
| Начало периода | Дата и время |  |
| ID категория | Числовой | Внешний (FK) |
| ID рубрика | Числовой | Внешний (FK) |
| ID статья | Числовой | Внешний (FK) |
| ID статья short | Числовой | Внешний (FK) |
| Конец периода | Дата и время |  |

Таблица 15. Сущность «Пожелания»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID пожелания | Числовой | Первичный (PK) |
| ID пользователя | Перечисление | Внешний (FK) |
| ID группы | Перечисление | Внешний (FK) |
| Текст | Текстовый |  |
| Дата пожелания | Дата и время |  |

Таблица 16. Сущность «Статья»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID статья | Числовой | Первичный (PK) |
| ID статья short | Числовой | Внешний (FK) |
| ID рубрика | Числовой | Внешний (FK) |
| ID категория | Числовой | Внешний (FK) |
| Название | Текстовый |  |
| Тэг статья | Текстовый |  |
| Текст | Текстовый |  |
| Актуальность статьи | Числовой |  |
| Посещаемость статьи | Числовой |  |
| Полная ссылка | Текстовый |  |

Таблица 17. Сущность «Группа»

|  |  |  |
| --- | --- | --- |
| Логическое имя атрибута | Тип данных | Ключ |
| ID группы | Числовой | Первичный (PK) |
| Права | Текстовый |  |

В диаграмме «сущность-связь» представлено 13 сущностей: «пользователь», «личная подборка», «аннотация статьи», «категория», «статистические данные», «исследование», «общая подборка», «рубрика», «пожелания», «статья», «опрос», «выбор», «группа».

Пользователь – содержит информацию о фамилии, имени, отчестве, телефоне, поле, e-mail, ВУЗе, курсе пользователя.

Личная подборка – содержит информацию о дате подборки, и о том, что в эту подборку входит (статьи).

Аннотация статьи – содержит информацию о названии аннотации, тексте, и о том, какая статья к ней относится, а также о том, к какой категории и рубрике она относится.

Категория – содержит информацию о названии категории, теге, актуальности и посещаемости категории.

Статистические данные – содержит информацию о статистических данных пользователя по посещения страниц с категориями, рубриками и статьями.

Исследование - содержит информацию о цели, тексте, результатах и дате исследования.

Общая подборка - содержит информацию о причастности к исследованию, количестве статей в подборке, тексте, начале и конце периода для подборки, отношение к тем или иным рубрикам, категориям и статьям.

Рубрика – содержит информацию о названии рубрики, теге, актуальности и посещаемости рубрики, а также о том, к какой категории она относится

Пожелания – содержит информацию о тексте, дате пожелания, а также о пользователе, которому оно принадлежит.

Статья - содержит информацию о названии статьи, тексте, посещаемости и актуальности, о полной ссылке на статью, а также о том, к какой категории и рубрике она относится.

Опрос – содержит информацию о том, какие вопросы включены в экспертную систему.

Выбор – содержит выборы ответов на каждый вопрос в экспертной системе.

Группа – содержит информацию о пренадлежности пользователя к опреденной группе.

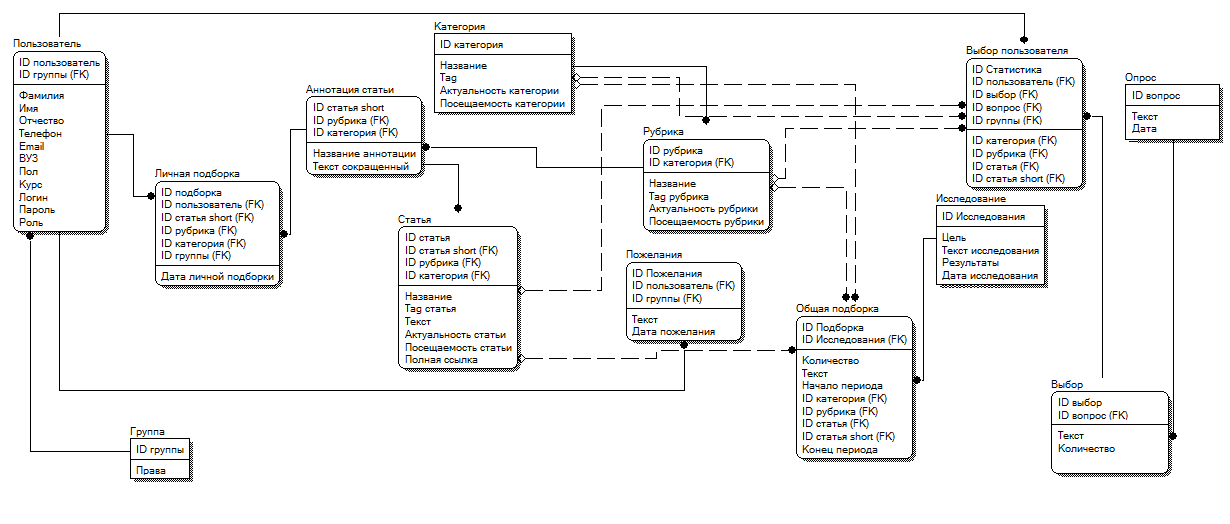


Рисунок – инфологическая модель процесса

#### ***1.2.6.3. Даталогическая модель***

Построение даталогической модели основано на разработанной ранее инфологической модели.

При отображении инфологической модели в даталогическую, сущности соответствуют коллекциям, атрибуты — полям документов.

Таблица Таблица app\_user

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID пользователь | Id\_user | INT | 11 | True | PK |
| ID группы | Id\_group | INT | 11 | True | FK |
| Фамилия | Surname | VARCHAR | 30 | True |  |
| Имя | Name\_user | VARCHAR | 30 | True |  |
| Отчество | Fam\_name | VARCHAR | 30 | True |  |
| Телефон | Telephone | INT | 11 | True |  |
| Email | Email | VARCHAR | 254 | True |  |
| ВУЗ | University | VARCHAR | 50 | True |  |
| Пол | Gender | VARCHAR | 30 | True |  |
| Курс | Course | INT | 11 | True |  |
| Логин | Login | VARCHAR | 50 | True |  |
| Пароль | Password | VARCHAR | 30 | True |  |
| Роль | role | VARCHAR | 30 | True |  |

Таблица Таблица app\_selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID Подборки | Id\_sel | INT | 11 | True | PK |
| ID Пользователя | Id\_user+id | INT | 11 | True | FK |
| ID статья short | Id\_ann\_id | INT | 11 | True | FK |
| ID рубрика | Id\_rub\_id | INT | 11 | True | FK |
| ID категория | Id\_cat\_id | INT | 11 | True | FK |
| ID группы | Id\_group\_id | INT | 11 | True | FK |
| Дата личной подборки | Date\_sel | Datetime |  |  |  |

Таблица Таблица app\_annotation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID статья short | Id\_ann | INT | 11 | True | PK |
| ID рубрика | Id\_rub\_id | INT | 11 | True | FK |
| ID категория | Id\_cat\_id | INT | 11 | True | FK |
| Название аннотации | Name\_ann | TEXT |  | True |  |
| Текст сокращенный | Text\_ann | TEXT |  | True |  |
| Дата публикации | Date\_ann | Datetime |  |  |  |

Таблица Таблица app\_category

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID категория | Id\_cat | INT | 11 | True | PK |
| Название | Name\_cat | VARCHAR | 20 | True |  |
| Тэг | Tag\_cat | VARCHAR | 20 | True |  |
| Актуальность категории | Actual\_cat | VARCHAR | 20 | True |  |
| Посещаемость категории | Going\_cat | VARCHAR | 20 | True |  |

Таблица app\_rubrica

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID рубрика | Id\_rub | INT | 11 | True | PK |
| ID категория | Id\_cat | INT | 11 | True | FK |
| Тэг рубрика | Tag\_rub | VARCHAR | 20 | True |  |
| Актуальность рубрики | Actual\_rub | VARCHAR | 20 | True |  |
| Посещаемость рубрики | Going\_rub | VARCHAR | 20 | True |  |

Таблица Таблица app\_poll

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID вопрос | Id\_poll | INT | 11 | True | PK |
| Текст | Text\_poll | VARCHAR | 200 | True |  |
| Дата | Date\_poll | Datetime |  | True |  |

Таблица Таблица app\_choice

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID выбор | Id\_choice | INT | 11 | True | PK |
| ID вопрос | Poll\_id | INT | 11 | True | FK |
| Текст | Text\_choice | VARCHAR | 200 | True |  |
| Количество | votes | INT | 11 | True |  |

Таблица Таблица Reseach

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID исследование | Id\_res | INT | 11 | True | PK |
| Цель | Aim\_res | VARCHAR | 200 | True |  |
| Текст исследования | Text\_res | TEXT |  | True |  |
| Даты исследования | Date\_res | Datetime |  |  |  |

Таблица Таблица app\_collection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID подборка | Id\_col | INT | 11 | True | PK |
| ID исследования | Id\_res | INT | 11 | True | FK |
| Текст | Text\_col | TEXT |  | True |  |
| Начало периода | Date\_start | Datetime |  | True |  |
| ID категория | Id\_cat\_id | INT | 11 | True | FK |
| ID рубрика | Id\_rub\_id | INT | 11 | True | FK |
| ID статья | Id\_iss\_id | INT | 11 | True | FK |
| ID статья short | Id\_ann\_id | INT | 11 | True | FK |
| Конец периода | Date\_end | Datetime |  | True |  |

Таблица Таблица app\_write

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID пожелания | Id\_wr | INT | 11 | True | PK |
| ID пользователя | Id\_user | INT | 11 | True | FK |
| ID группы | Id\_group | INT | 11 | True | FK |
| Текст | Text\_wr | TEXT |  | True |  |
| Дата пожелания | Date\_wr | Date | 11 | True |  |

Таблица Таблица app\_issue

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID статья | Id\_iss | INT | 11 | True | PK |
| ID статья short | Id\_ann | INT | 11 | True | FK |
| ID рубрика | Id\_rub | INT | 11 | True | FK |
| ID категория | Id\_cat | INT | 11 | True | FK |
| Название | Name\_iss | VARCHAR | 30 | True |  |
| Тэг статья | Tag\_iss | VARCHAR | 30 | True |  |
| Текст | Text\_iss | TEXT |  | True |  |
| Актуальность статьи | Actual\_iss | VARCHAR | 30 | True |  |
| Посещаемость статьи | Going\_iss | VARCHAR | 30 | True |  |
| Полная ссылка | url\_iss | VARCHAR | 30 | True |  |

Таблица Таблица app\_group

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Логическое имя атрибута | Поле | Тип данных | Длина поля | NN (Not Null) | Тип ключа |
| ID группы | Id\_group | INT |  | True | PK |
| Права | prava | VARCHAR | 200 | True |  |

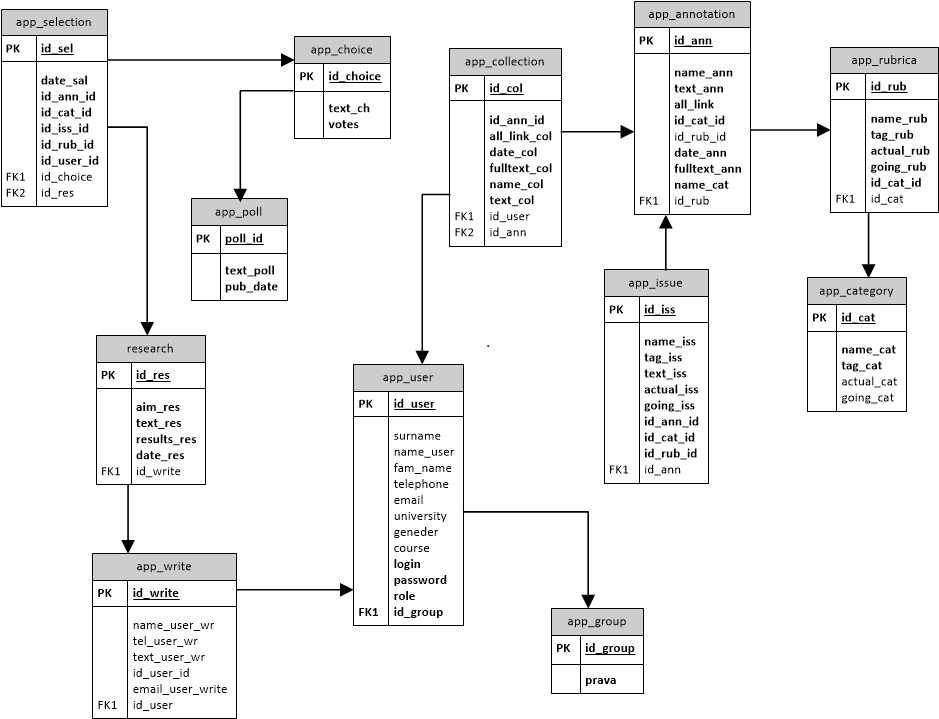


Рисунок – даталогическая модель базы данных

### 1.2.7. Выбор программных средств

**Операционная система**

Все компоненты системы являются кроссплатформенными. В процессе разработки использовался компьютер под управлением ОС Windows 10.

Для пользователей веб-сервиса ИСППОС выбор ОС не принципиален, главное, чтобы на пользовательском устройстве был браузер с поддержкой языков HTML5, CSS3, JS5.

**Программное обеспечение**

Для работы клиентской части веб-приложения на компьютере, ноутбуке/нетбуке пользователя требуется:

1. Любая ОС, поддерживающая работу с браузером, который поддерживает HTML5, CSS3, JS5;
2. Один из перечисленных веб-браузеров:

- Internet Explorer 11.0 и выше,

- Mozilla Firefox 36.0 и выше,

- Google Chrome 42.0 и выше,

- Opera 30.0 и выше,

- Safari 4.0 и выше.

3. Для работы серверной части требуется:

- Облачный сервер

### 1.2.8. Выбор аппаратных средств

* Для пользователей ИСППОС достаточны компьютеры со следующими минимальными характеристиками:
* Процессор с частотой не менее 1 ГГц;
* Размер оперативной памяти не менее 512 МБ;
* Видеоадаптер и монитор, способные обеспечить графический режим не менее 1024 x 768 точек с 16 битной цветопередачей;
* Система устройства хранения данных не менее 1 ГБ;
* Наличие манипулятора «мышь» или другого указывающего устройства;
* Наличие клавиатуры или её экранного аналога;
* Обеспечение доступа в интернет.

### 1.2.9. Структура программного изделия

В рамках работы было разработано веб-приложение. В общем виде структуру проекта можно представить так (в скобках указаны рабочие названия составляющих):

**Проект DjangoWebProject4:**

1. Окружение Python
2. Папка app (файлы приложения):
   * 1. Папка migrations (файлы миграций базы данных)
     2. Папка static (статические файлы приложения)
        + Папка app (js и css файл проекта)
        + Папка images (графические файлы проекта)
     3. Папка templates (html файлы проекта и файлы конфигурации)
        + Папка app (html файлы проекта)
        + Файлы конфигурации
3. Папка DjangoWebProject4 (файлы проекта)
4. Файл базы данных
5. Файл конфигурации прложения и проекта
6. Файл с необходимыми компонентами для установки

**1.2.9.1. Окружение Python**

Виртуальное окружение помогает изолировать зависимости Python/Django для каждого отдельного проекта. Это значит, что изменения одного сайта никогда не затронут другие сайты.

Для разработки данного приложение, необходимо было установить следующие компоненты виртуального окружения:

- пакет azure

- пакет Django

- пакет mysql

- пакет pillow

- пакет pip

- пакет setuptools

**1.2.9.2. Папка app**

Данная папка отвечает за файлы, относящиеся к конфигурации приложения. Она подразделяется на 3 составляющие: migrations, static и templates.

**Migrations**

Папка migrations содержит все миграции, который были проведены в базу данных.

**Static**

Папка static разделяется на две составляющие: app и images. В папке app содержатся файлы формального языка описания внешнего вида документа, написанного с использованием языка разметки (css файлы - папка contents); файлы, содержащие шрифты (папка fonts); файлы, содержащие javascripts (папка scripts). Папка images содержит графические файлы приложения.

**Templates**

Данная папка содержит html страницы, которые вызывает приложение. Html страницы лежат в папке app и содержат шаблон, в котором указываются формы, представления и данные, которые надо отобразить. Данные шаблоны являются частью принципа Django приложений DRY (don’t repeat yourself) :

* *Annotations.html*

Отвечает за отображение списка статей в каждой категории и рубрике.

* *Annotationdetails.html*

Отвечает за отображение каждой отдельной статьи.

* *Collections.html*

Отвечает за отображение индивидуальной и общей подборки для пользователей.

* *Contact.html*

Отвечает за отображение формы обратной связи.

* *Editprofile.html*

Отвечает за отображение формы редактирования профиля.

* *Layout.html*

Главная страница, которая содержит в себе все основные особенности приложения (стили, шапка, подвал).

* *Login.html*

Отвечает за авторизацию.

* *Poll.html*

Отвечает за список вопросов из экспертной системы.

* *Profile.html*

Отвечает за профиль пользователя.

* *Register.html*

Отвечает за регистрацию.

* *Results.html*

Отвечает за результаты опроса

**Также папка templates содержит файлы конфигурации приложений:**

**-** *\_init\_.py*

Отвечает за сборку проекта.

* *Admin.py*

Содержит конфигурации, отвечающие за администрирование приложения.

* *Forms.py*

Содержит формы отображения данных (форма аутендификации, форма регистрации, форма обратной связи, форма профиля пользователя, форма редактирования профиля пользователя)

* *Models.py*

Содержит классы моделей данных. В структуре приложений Django класс, это таблица в базе данных – class Poll, class Choice, class UserChoice, class User, class Category, class Rubrica, class Annotation, class Issue, class Collection, class Selection, class Write, class Group, class Reseach.

* *Tests.py*

Содержит файлы обработки для экспертной системы.

* *Urls.py*

Содержит ссылки на страницы, “urls”, которые вызывают нужные представления и наследование ссылок из urls,py из приложения.

* *views.py*

Содержит представления приложения PollListView, PollDetailView, PollResultsView, layout2, write, contactform, contact, vote, deletevotes, seed, annotations, annotationdetails, get\_prefs, collections, collectiondetails, shots, create, profile, editprofile, register, UserFormView.

**1.2.9.3. Папка DjangoWebProject4**

**В данной папке содержатся четыре файла .py:**

* *\_init\_.py*

Отвечает за сборку проекта.

* *Settings.py*

Содержит необходимые настройки проекта (шаблоны, языки, установленные приложения, классы, информацию о базе данных)

* *Urls.py*

Содержит ссылки на страницы, “urls”, которые вызывают нужные представления и наследуются в urls,py приложения.

* *Wsgi.py*

Конфигурирует настройки проекта.

Структурная схема системы в целом представлена на рисунке ниже.



Рис. 4. Структурная схема системы

Будет другая картинка

### 1.2.10. Алгоритм экспертной системы

В разработанном приложение экспертная система анализирует предпочтения пользователей и организует индивидуальную подборку для каждого зарегестрировавшегося и прошедшего опрос.

ТУТ БУДЕТ АЛГОРИТМ

# 2. Технологическая часть

## 2.1. Граф диалога

Изначально пользователь попадает на главную страницу приложения. Пройдя авторизацию или регистрацию, пользователю необходимо пройти опрос для получения индивидуальной рекоммендательной подборки статей. Так как это веб-приложение (сайт), то в пункты, которые обозначены на главной странице (1.1 – 1.10) можно попасть из любой точки приложения.

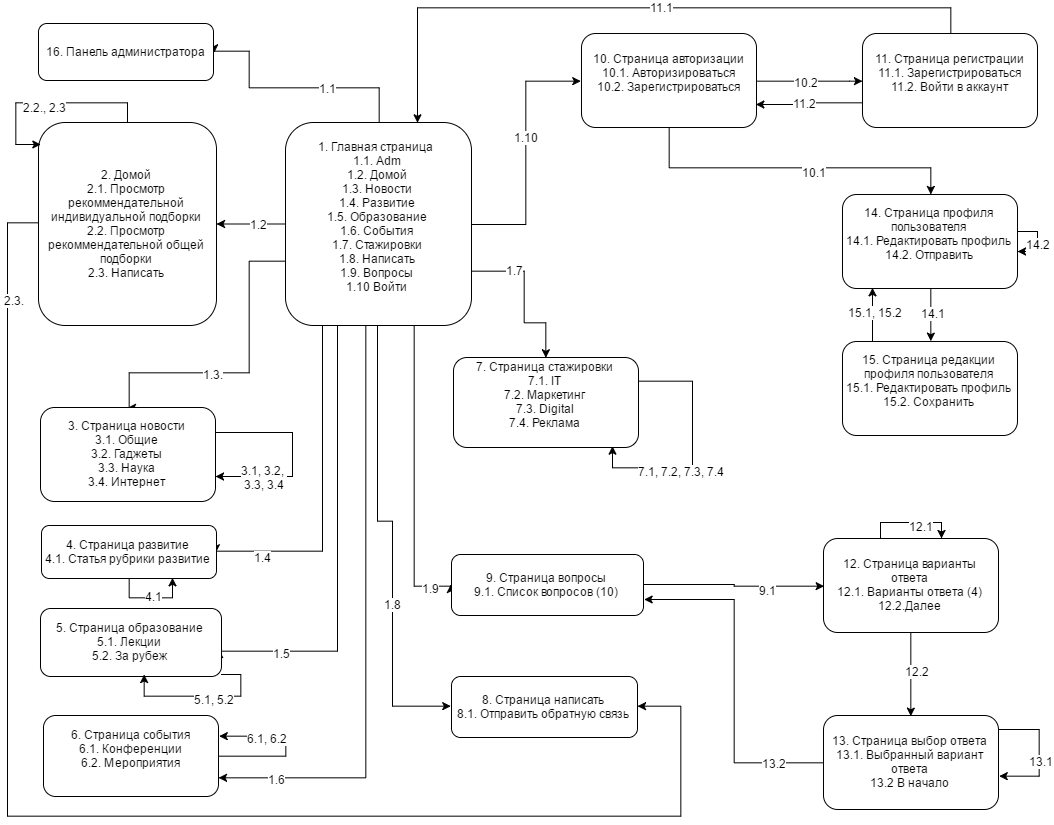


Рис. 10. Граф диалога

## 2.2. Разработка интерфейса взаимодействия с пользователем

Интерфейс взаимодействия с пользователем представляет собой набор веб-страниц. Главной страницей является приветственная страница пользователя, на которой у зарегистрировавшегося и прошедшего опрос пользователя есть возможность ознакомится с индивидуальной рекоммендательной подборкой, а так же общей подборкой для всех пользователей. Кроме этой страницы разработаны страницы авторизации, регистрации, личного кабинета, редактирования личного кабинета, опроса, вариантов ответа на опрос, результатов опроса, отображения статей, обратной связи.

Для отображения информации используется фреймворк для web-разработки Bootstrap и jQuery — библиотека JavaScript.

### 2.2.1. Экранные формы веб-сервиса

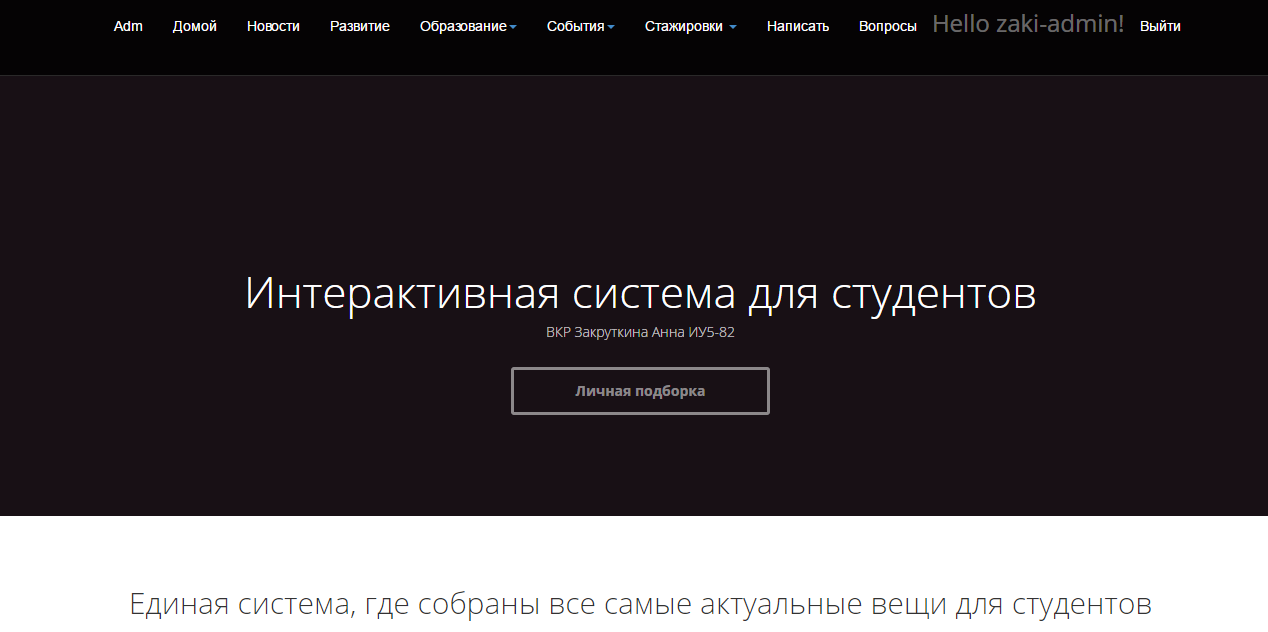


Рис. Главная страница

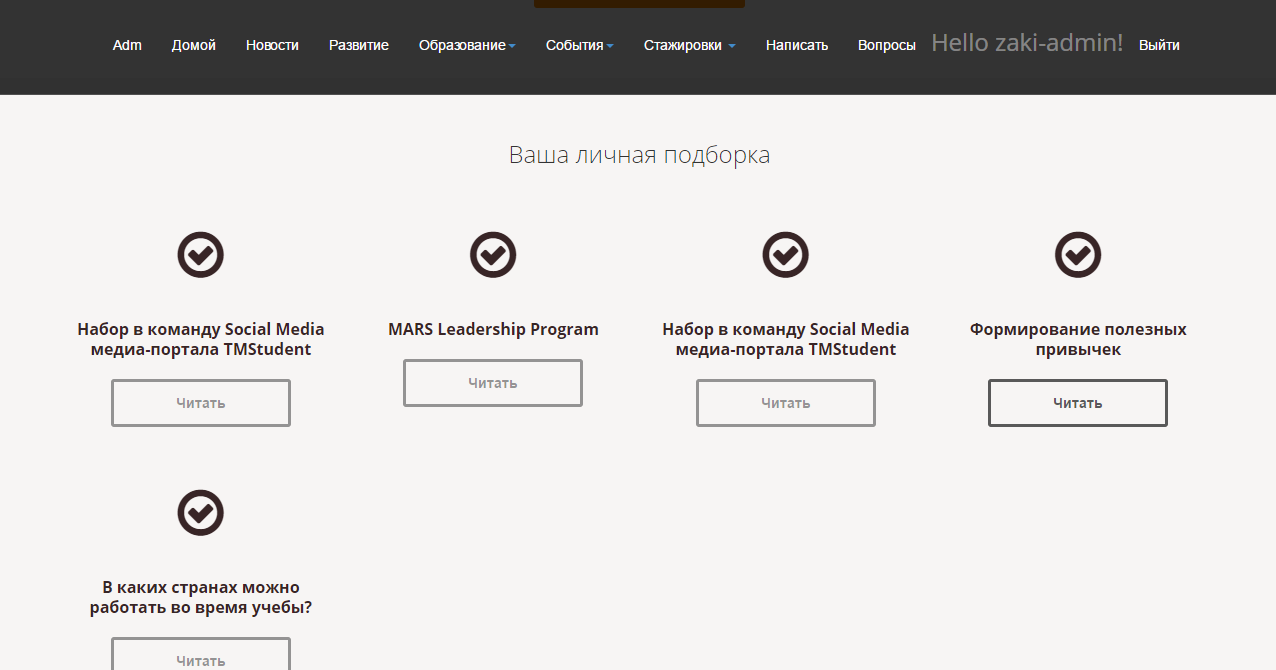


Рис. Индивидуальная рекоммендательная подборка

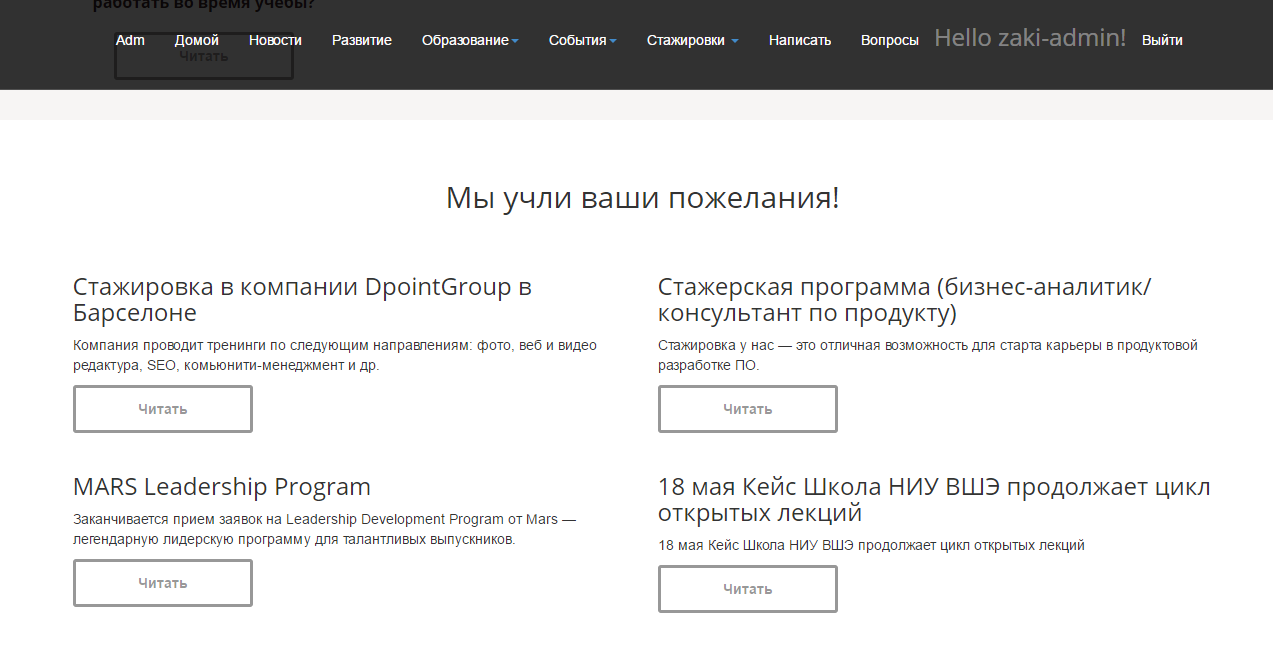


Рис. Общая рекоммендательная подборка

Страница с отображением Яндекс.Новостей включает в себя несколько внутренних рубрик, по которым можно перемещаться за счет сделанных на странице кнопок.

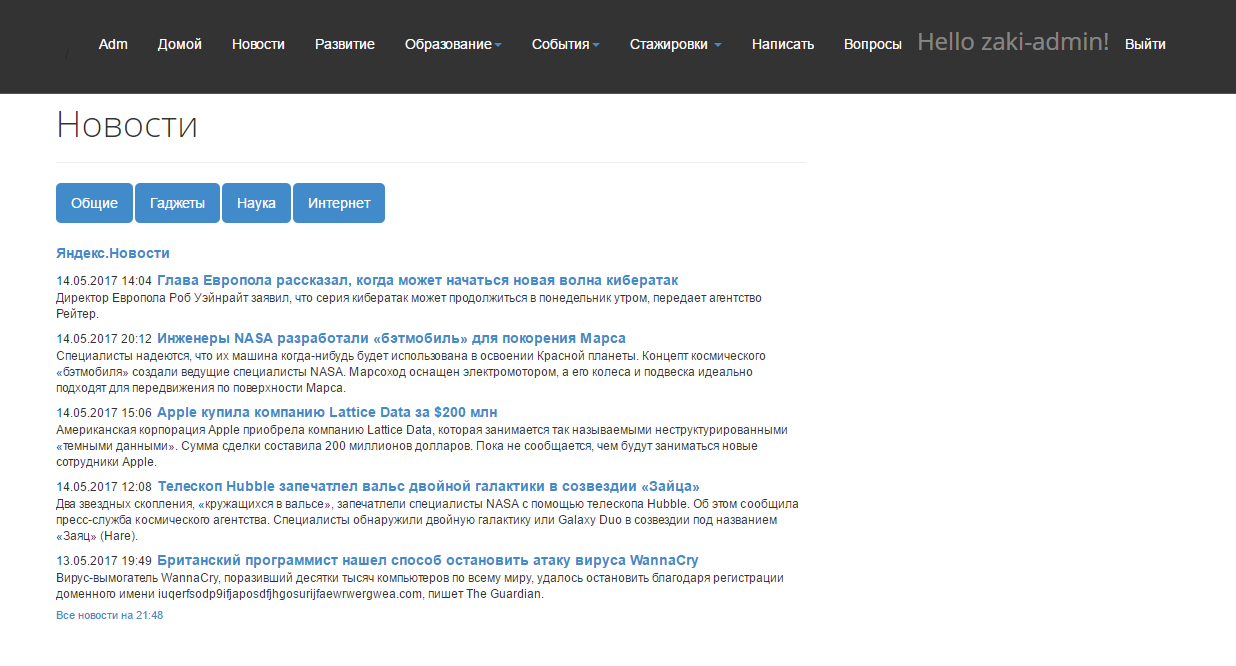


Рис. Страница с интегрированными новостями с Яндекс.ру

Страница с отображением статей по разным категориям включает в себя отображение статей данной категории и отображение общей подборки для пользователей (справа).

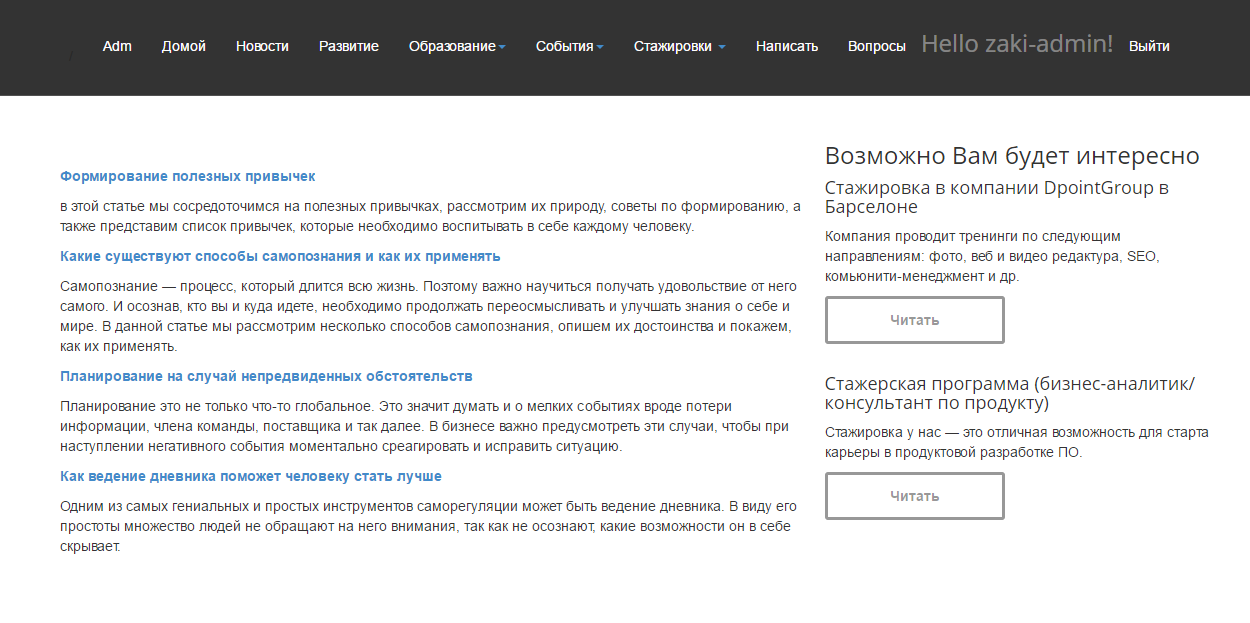


Рис. Страница со статьями

Страница с отображением стажировок включает в себя несколько внутренних рубрик, по которым можно перемещаться за счет сделанных на странице кнопок.

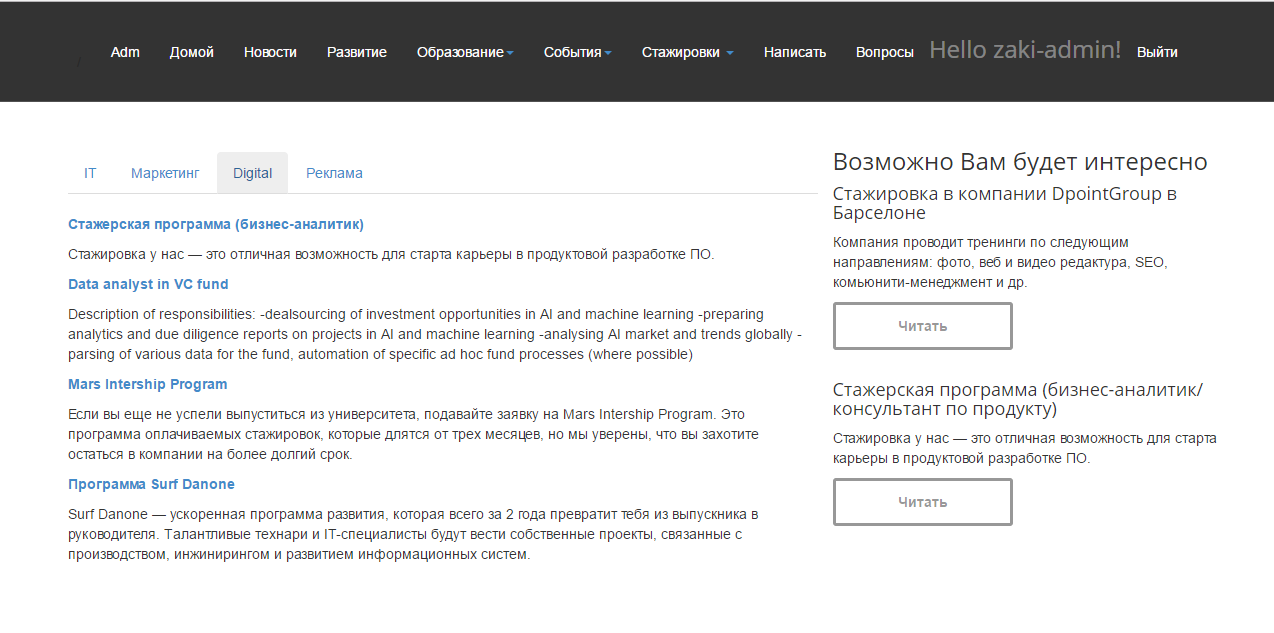


Рис. Страница со статьями, категория «стажировки»

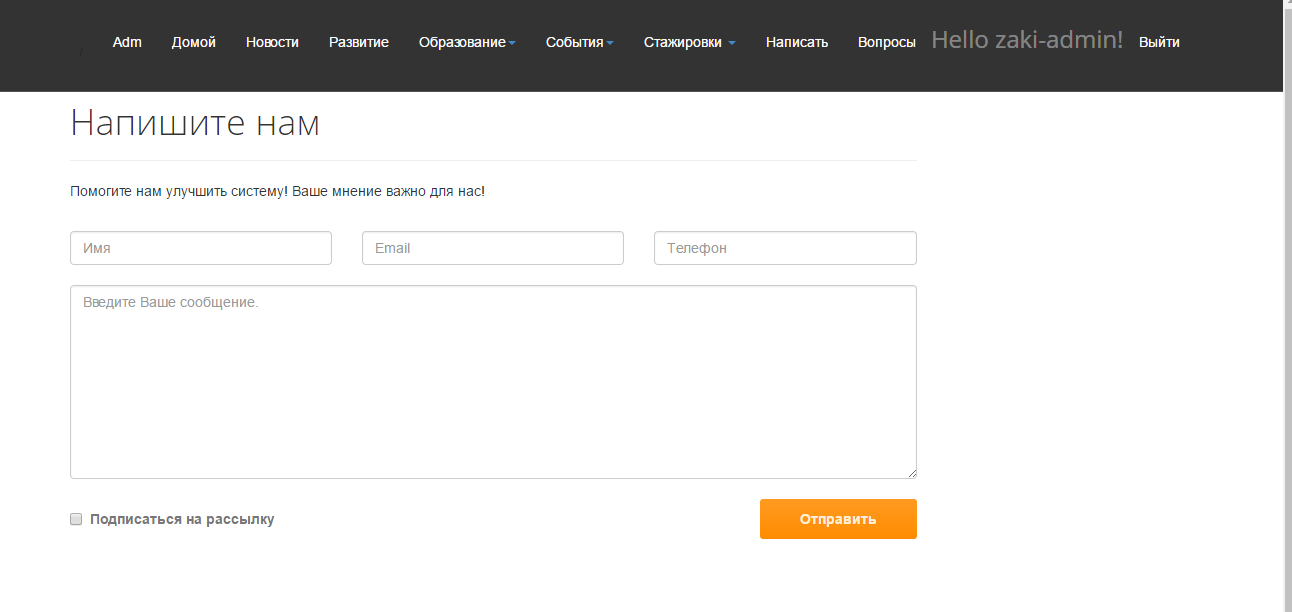


Рис.Страница «обратная связь»

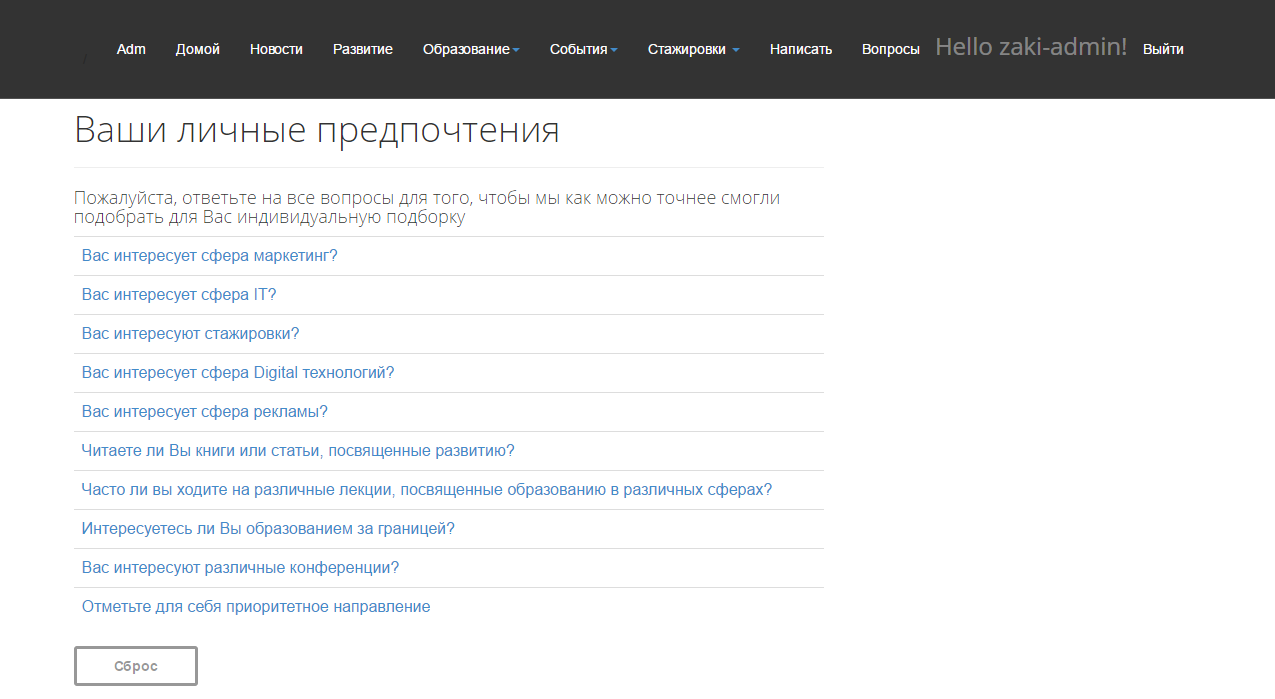


Рис. Страница с опросом

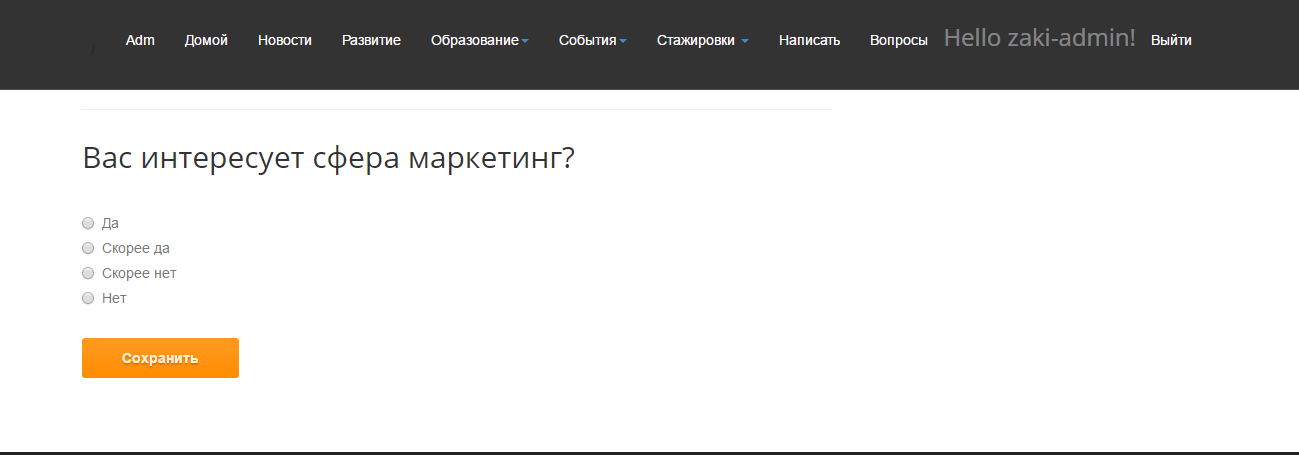


Рис. Страница с выбором вариантов ответа

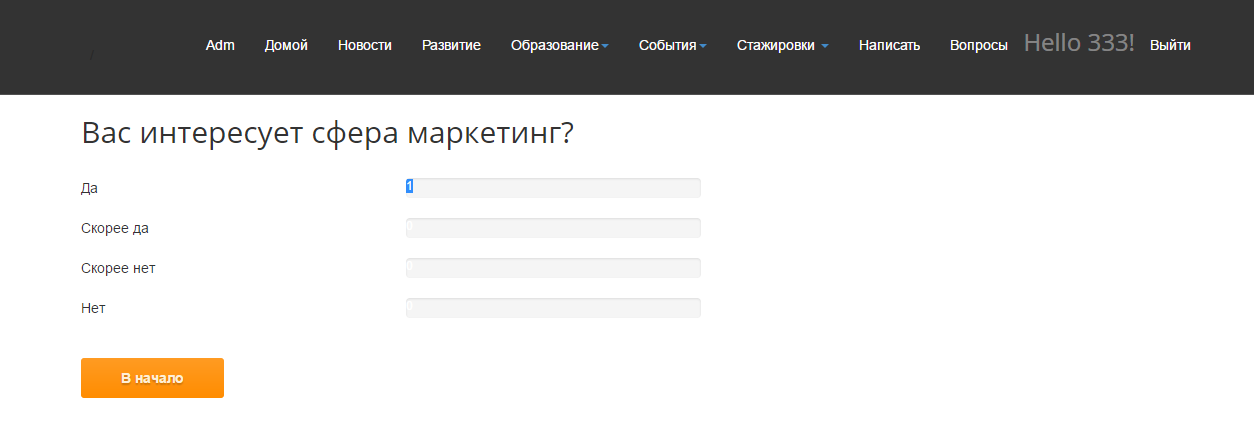


Рис. Страница с выбранным ответом



Рис. Страница личного кабинета пользователя

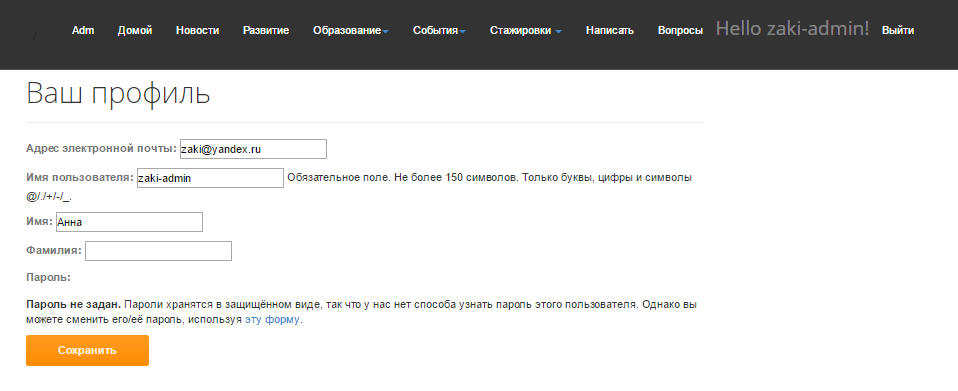


Рис. Страница редактирования информации пользователя

# 3. Исследовательская часть

## Выбор способа развертывания приложения на облачной платформе Azure

**Цель исследования:** необходимо оценить, какая из предоставляемых технологий будет наиболее полно соответствовать требованиям проекта.

**Описание критериев**

Таблица 17. Описание критериев

|  |  |  |
| --- | --- | --- |
| **Критерий** | **Обозначение** | **Описание** |
| Производительность (запросов в секунду) | К1 | Среднее количество запросов, обслуживаемых сервером в секунду |
| Объем используемой оперативной памяти (МБ) | К2 | Количество используемой оперативной памяти во время работы сервера |
| Трудоёмкость разработки (месяцев) | К3 | Ожидаемое время, которое уйдёт на разработку при использовании данной технологии |
| Процента загрузки процессора (%) | К4 | Процент загрузки процессора |
| Качество документации  (балл) | К5 | Доступность, понятность документации и наличие примеров |
| Простота обработки JSON (балл) | К6 | Простота обработки данных в формате JavaScript Object Notation |
| Простота разворачивания (балл) | К7 | Простота запуска сервера на реальном оборудовании |

**Описание вариантов**

Таблица 18. Описание вариантов

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Обозначение** | **Описание** |
| NodeJS (JavaScript) | В1 |  |
| Python/Django | В2 |  |
| PHP | В3 |  |

**Перевод качественных критериев в количественные**

Таблица 19. Перевод качественных критериев в количественные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Оценка** | **Качественная оценка** | **Качество документации** | **Простота обработки JSON** | **Простота разворачивания** |
| 5 | Очень просто | Очень простая документация, есть примеры | Встроенная поддержка на уровне языка | Установка и настройка не требуются |
| 4 | Просто | Простая документация, есть примеры | Сериализация и десериализация выполняются автоматически | Требуется минимальная установка ПО |
| 3 | Удовлетвори-тельно | Документация существует, но недостаточно полна или мало примеры | Сериализация и десериализация возможна средствами библиотеки | Требуется установка и настройка ПО |
| 2 | Сложно | Неполная документация или отсутствие примеров | Встроенных средств для работы с JSON нет | Требуется перекомпиляция на стороне сервера |
| 1 | Очень сложно | Документация отсутствует, примеры отсутствуют | Обработка JSON невозможно или крайне затруднительна | Требуется доработка и компиляция под конкретный сервер |

**Оценка важности критериев**

*1. Метод парного сравнения*

Таблица 20. Метод парного сравнения критериев

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерии** | **K1** | **K2** | **K3** | **K4** | **K5** | **K6** | **K7** |  |  |
| **K1** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.04 |
| **K2** | 1 | 1 | 0 | 0.5 | 0.5 | 0 | 0.5 | 3.5 | 0.13 |
| **K3** | 1 | 1 | 1 | 0.5 | 1 | 0.5 | 1 | 6 | 0.21 |
| **K4** | 1 | 0.5 | 0.5 | 1 | 0.5 | 0.5 | 1 | 5 | 0.18 |
| **K5** | 1 | 0.5 | 0 | 0.5 | 1 | 0.5 | 1 | 4.5 | 0.16 |
| **K6** | 1 | 1 | 0.5 | 0.5 | 0.5 | 1 | 0.5 | 5 | 0.18 |
| **K7** | 1 | 0.5 | 0 | 0 | 0 | 0.5 | 1 | 3 | 0.10 |

Итоговая сумма:

Вычисляем веса критериев:

*2. Метод базового критерия*

В качестве базового критерия возьмем наименее значимый критерий К1

*K1 = X*

*K2 = 2X*

*K3 = 4X*

*K4 = 2X*

*K5 = 3X*

*K6 = 4X*

*K7 = 2X*

*18 X = 1 => X = 0.06*

Для определения весов критериев возьмем среднее значение для каждого веса, полученное двумя методами, тогда:

Итог:

**Количественные и качественные характеристики рассматриваемых вариантов**

Таблица 21. Характеристики рассматриваемых вариантов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерии** | **B1 - NodeJS** | **B2 – Python/dajngo** | **B3 - PHP** | **Вес** |
| К1 | 430 | 1800 | 1700 | 0.04 |
| К2 | 110 | 36 | 50 | 0.13 |
| К3 | 1 | 2 | 3 | 0.23 |
| К4 | 9 | 8 | 3 | 0.15 |
| К5 | 5 | 4 | 4 | 0.16 |
| К6 | 5 | 3 | 3 | 0.19 |
| К7 | 4 | 4 | 2 | 0.11 |

Все оценки производительности выполнялись на одном компьютере.

Для нормирования значений вариантов по критериям «чем больше, тем лучше» (критерии К1, К5, К6, К7) воспользуемся формулой:

Для нормирования значений вариантов по критериям «чем меньше, тем лучше» (критерии К2, К3, K4) воспользуемся формулой:

**Нормирование**

Таблица 22. Нормированная таблица характеристик

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий** | **B1 - NodeJS** | **B2 – Python/dajngo** | **B3 - PHP** | **Вес** |
| K1 | 0.24 | 1 | 0.95 | 0.04 |
| K2 | 0.32 | 1 | 0.72 | 0.13 |
| K3 | 1 | 0.5 | 0.33 | 0.23 |
| K4 | 0.33 | 0.38 | 1 | 0.15 |
| K5 | 1 | 0.8 | 0.8 | 0.16 |
| K6 | 1 | 0.6 | 0.6 | 0.19 |
| K7 | 1 | 1 | 0.5 | 0.11 |

**1. Выбор лучшего варианта методом максимума взвешенной суммы**

Наилучшим вариантом по методу максимума взвешенной суммы является вариант №1 – ***NodeJS***.

**2. Выбор лучшего варианта методом близости к идеалу**

Наилучшим вариантом является вариант № 2 – ***Python/Django***

**3. Выбор лучшего варианта методом аналитической иерархии**

*m = 7 => R = 1.32*

Критерии расположены в предположительном порядке уменьшения весового коэффициента.

Таблица 23. Нормированная таблица характеристик

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Критерии** | **K3** | **K6** | **K5** | **K4** | **K2** | **K7** | **K1** |  |  |
| K3 | 1 | 2 | 2 | 5 | 5 | 7 | 8 | 3.43 | 0.33 |
| K6 | 1/2 | 1 | 3 | 4 | 5 | 8 | 8 | 2.95 | 0.28 |
| K5 | 1/2 | 1/3 | 1 | 2 | 3 | 7 | 5 | 1.66 | 0.16 |
| K4 | 1/5 | 1/4 | 1/2 | 1 | 2 | 7 | 8 | 1.16 | 0.11 |
| K2 | 1/5 | 1/5 | 1/3 | 1/2 | 1 | 7 | 8 | 0.87 | 0.08 |
| K7 | 1/7 | 1/8 | 1/7 | 1/7 | 1/7 | 1 | 2 | 0.27 | 0.03 |
| K1 | 1/8 | 1/8 | 1/5 | 1/8 | 1/8 | 1/2 | 1 | 0.22 | 0.02 |

Оценка согласованности ОС = 0.08 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К1.

Таблица 24. Сравнение вариантов по критерию К1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В3** | **В2** | **В1** |  |  |
| **В3** | 1 | 2 | 3 | 1.817 | 0.528 |
| **В2** | 1/2 | 1 | 3 | 1.145 | 0.333 |
| **В1** | 1/3 | 1/3 | 1 | 0.481 | 0.14 |

Оценка согласованности ОС = 0.05 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К2.

Таблица 25. Сравнение вариантов по критерию К2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В3** | **В1** | **В2** |  |  |
| **В3** | 1 | 1 | 2 | 1.26 | 0.413 |
| **В1** | 1 | 1 | 1 | 1.0 | 0.327 |
| **В2** | 1/2 | 1 | 1 | 0.794 | 0.26 |

Оценка согласованности ОС = 0.05 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К3.

Таблица 26. Сравнение вариантов по критерию К3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В1** | **В2** | **В3** |  |  |
| **В1** | 1 | 5 | 6 | 3.107 | 0.717 |
| **В2** | 1/5 | 1 | 3 | 0.843 | 0.195 |
| **В3** | 1/6 | 1/3 | 1 | 0.382 | 0.088 |

Оценка согласованности ОС = 0.08 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К4.

Таблица 27. Сравнение вариантов по критерию К4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В3** | **В1** | **В2** |  |  |
| **В3** | 1 | 2 | 4 | 2.0 | 0.547 |
| **В1** | 1/2 | 1 | 4 | 1.26 | 0.345 |
| **В2** | 1/4 | 1/4 | 1 | 0.397 | 0.109 |

Оценка согласованности ОС = 0.05 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К5.

Таблица 28. Сравнение вариантов по критерию К5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В1** | **В2** | **В3** |  |  |
| **В1** | 1 | 3 | 5 | 2.466 | 0.627 |
| **В2** | 1/3 | 1 | 4 | 1.101 | 0.28 |
| **В3** | 1/5 | 1/4 | 1 | 0.368 | 0.094 |

Оценка согласованности ОС = 0.08 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К6.

Таблица 29. Сравнение вариантов по критерию К6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В1** | **В2** | **В3** |  |  |
| **В1** | 1 | 5 | 5 | 2.924 | 0.709 |
| **В2** | 1/5 | 1 | 2 | 0.737 | 0.179 |
| **В3** | 1/5 | 1/2 | 1 | 0.464 | 0.112 |

Оценка согласованности ОС = 0.05 < 0.1 => матрица согласована.

Сравнение вариантов по критерию К7.

Таблица 30. Сравнение вариантов по критерию К7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Варианты** | **В1** | **В2** | **В3** |  |  |
| **В1** | 1 | 3 | 7 | 2.759 | 0.649 |
| **В2** | 1/3 | 1 | 5 | 1.186 | 0.279 |
| **В3** | 1/7 | 1/5 | 1 | 0.306 | 0.072 |

Оценка согласованности ОС = 0.06 < 0.1 => матрица согласована.

Выбор лучшего варианта:

Наилучшим вариантом является вариант №2 – ***Python/Django***.

**4. Выбор лучшего варианта методом ранжирования Борда**

Таблица 31. Метод ранжирования Борда

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий** | | **B1 - NodeJS** | **B2 – Python/dajngo** | **B3 - PHP** |
| Производительность (запросов в секунду) | K1 | 3 | 2 | 1 |
| Объем используемой оперативной памяти (МБ) | K2 | 2 | 3 | 1 |
| Трудоёмкость разработки (месяцев) | K3 | 1 | 2 | 3 |
| Процента загрузки процессора (%) | K4 | 2 | 3 | 1 |
| Качество документации  (балл) | K5 | 1 | 2 | 3 |
| Простота обработки JSON (балл) | K6 | 1 | 2.5 | 2.5 |
| Простота разворачивания (балл) | K7 | 1 | 2 | 3 |
|  |  | 11 | 16.5 | 14.5 |

Наилучшим вариантом является вариант №2 – ***Python/Django***.

**Результаты исследования**

Таблица 32. Результаты исследования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Метод** | **Расчёт** | **B1** | **B2** | **B3** | **Предпочтение** |
| Метод максимума взвешенной суммы |  | 0.79 | 0.69 | 0.65 |  |
| Метод близости к идеалу |  | 0.38 | 0.39 | 0.42 |  |
| Метод аналитической иерархии |  | 0.62 | 0.21 | 0.18 |  |
| Метод ранжирования Борда |  | 11.00 | 16.50 | 14.50 |  |

**Вывод:** Было использовано четыре метода выбора лучшего варианта веб-серверных технологий для использования в выпускном проекте. Все четыре метода показали одинаковый результат в качестве лучшего №2 – ***Python/Django***.. Таким образом, можно сделать вывод, что эта технология действительно является оптимальной и отвечающей всем необходимым требованиям проекта.

# Заключение

При разработке интерактивной системы поддержки процесса обучения студентов с использованием облачных технологий достигнуты следующие результаты:

* + 1. изучена предметная область
    2. изучена облать работы с облачными платформами
    3. сформулированы требования к разрабатываемой программе;
    4. спроектирована структура базы данных;
    5. разработана архитектура системы;
    6. разработана структура модулей веб-интерфейса;
    7. описан процесс проектирования интерфейса программы;
    8. разработан и написан комплект технической документации;
    9. разработана графическая часть конструкторской документации.

Полученный информационно-программный продукт имеет возможность дальнейшего расширения и усовершенствования: имеется возможность добавления новых модулей и функциональных возможностей без серьезного вмешательства в уже разработанные модули.

# Список литературы

1. ГОСТ 19.201-78. ЕСПД. Техническое задание. Требования к содержанию и оформлению.
2. ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению.
3. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению.
4. ГОСТ 7.32—2001 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.
5. Приказ № 1047 (от 5 сентября 2013 г.) «Об утверждении Порядка формирования федерального перечня учебников, рекомендуемых к использованию при реализации имеющих государственную аккредитацию образовательных программ начального общего, основного общего, среднего общего образования».
6. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 03.12.2012) // «Российская газета», N 256, 31.12.2001
7. Федеральный закон от 24.07.2009 N 212-ФЗ (ред. от 25.12.2012) «О страховых взносах в Пенсионный фонд Российской Федерации, Фонд социального страхования Российской Федерации, Федеральный фонд обязательного медицинского страхования» // «Собрание законодательства РФ», 27.07.2009, N 30, ст. 3738
8. Методические указания по выполнению выпускной работы бакалавра по специальности «Информатика и вычислительная техника» для студентов кафедры «Системы обработки информации и управления» / Кузовлев В.И.,– М.: МГТУ им. Н.Э. Баумана, 2017 г.
9. Интернет-справочник по языку HTML. URL: <http://www.htmlbook.ru>
10. Портал аналитической информации, справочные материалы по разработкам баз данных (разработка инфологических и даталогических моделей). URL: <http://www.citforum.ru>
11. Руководство по внедрению и использованию Twitter Bootstrap. URL: <http://getbootstrap.com/>
12. Документация к Django.

#### Приложение А. Иллюстрации для доклада