

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина» (УрФУ)

Институт радиоэлектроники и информационных технологий - РТФ

ОТЧЕТ

о проектной работе

по теме: Автоматизация общения с абитуриентами

по дисциплине: Проектный практикум

Команда: Improve Yourself

Екатеринбург

2021

СОДЕРЖАНИЕ

[Введение 3](#_Toc73665135)

[Команда 4](#_Toc73665136)

[Целевая аудитория 5](#_Toc73665137)

[Календарный план проекта 6](#_Toc73665138)

[Определение проблемы 10](#_Toc73665139)

[Подходы к решению проблемы 11](#_Toc73665140)

[Анализ аналогов 12](#_Toc73665141)

[Требования к продукту и MVP 13](#_Toc73665142)

[Стек для разработки 14](#_Toc73665143)

[Прототипирование 15](#_Toc73665144)

[Разработка системы 16](#_Toc73665145)

[Заключение 17](#_Toc73665146)

Введение

В настоящее время поступление в университет происходит удаленно, из-за этого особенно актуальной является проблема поиска информации о вузе перед подачей документов.

Однако наданный момент у Уральского Федерального Университета нет достаточно удобной системы поиска нужной информации, касающейся поступления и информации о вузе в целом, отсутствует удобное разбиение часто задаваемых вопросов по категориям. Стоит отметить, что существуют сайты, позволяющие создать своего чат-бота или FAQ-страницы для необходимых задач, но они имеют ряд весомых недостатков: отсутствие гибкости решения, а также полная зависимость от работы чужого сервиса.

Цель этого проекта создать специализированный ресурс, предоставляющий удобный поиск необходимой информации с приятным внешним видом. Для этого необходимо разработать сайт со всем необходимым функционалом и разделением вопросов на категории, дизайн которого будет похож на дизайн официального сайта УрФУ. Должны быть реализованы возможность редактирования контента и инструментарий для поиска информации на сайте. К тому же, нужно разработать чат-бота в социальной сети ВКонтакте, который отвечал бы на вопросы абитуриентов заранее приготовленными ответами, и, в случае отсутствия данного вопроса в базе, обращаться к сотруднику УрФУ для решения проблем пользователя.

Команда

* Николаев Илья Александрович РИ-100021 – Тимлид
* Япрынцев Артём Максимович РИ-100021 – Программист
* Кучеренко Алексей Викторович РИ-100014 – Тестировщик

Целевая аудитория

Суть данного проекта реализовать систему удобной организации информации об Уральском Федеральном Университете. Таким образом, целевой аудиторией данного проекта являются люди, преимущественно 16-20-летнего возраста, желающие узнать больше об этом учебном заведении, но в большей части это будут абитуриенты университета, вне зависимости от пола, направления подготовки. С помощью разработанных специализированных ресурсов они смогут узнать о всех тонкостях поступления, обучения, проживания и других аспектах обучения в университете. Пользователь сможет это сделать при помощи своего мобильного телефона или персонального компьютера, в любое свободное время, не дожидаясь начала рабочего дня сотрудников медиа-штата университета.

Календарный план проекта

Название проекта: 1С21S.B.01. Автоматизация общения с абитуриентами

Руководитель проекта: Сергеева Любовь Владимировна

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Ответственный** | **Длительность** | **Дата начала** | **Временные рамки проекта** | | | |
| 1-2 нед | 3-4 нед | 5-7 нед | 8-9 нед |
| ***Анализ*** | | | | | | | | |
| *1.1* | *Определение проблемы* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.2* | *Выявление целевой аудитории* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.3* | *Конкретизация проблемы* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.4* | *Подходы к решению проблемы* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.5* | *Анализ аналогов* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.6* | *Определение платформы и стека для продукта* | Николаев И.А. | 2 недели | 01.04.21 |  |  |  |  |
| *1.7* | *Формулирование требований к MVP продукта* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.8* | *Определение платформы и стека для MVP* | Николаев И.А. | 2 недели | 01.04.21 |  |  |  |  |
| *1.9* | *Формулировка цели* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.10* | *Формулирование требований к продукту* | Кучеренко А.В. | 2 недели | 01.04.21 |  |  |  |  |
| *1.11* | *Определение задач* | Николаев И.А. | 2 недели | 01.04.21 |  |  |  |  |
| ***Проектирование*** | | | | | | | | |
| *2.1* | *Архитектура системы (компоненты, модули системы)* | Николаев И.А. | 2 недели | 16.04.21 |  |  |  |  |
| *2.2* | *Разработка сценариев использования системы* | Николаев И.А. | 2 недели | 16.04.21 |  |  |  |  |
| ***Разработка*** | | | | | | | | |
| *3.1* | *Написание кода* |  | 3 недели | 02.05.21 |  |  |  |  |
| *3.1.1* | *Дизайн-макет* | Николаев И.А. | 1 недели | 02.05.21 |  |  |  |  |
| *3.1.2* | *Десктоп версия* | Николаев И.А. | 2 недели | 02.05.21 |  |  |  |  |
| *3.1.3* | *Мобильная версия* | Николаев И.А. | 2 недели | 02.05.21 |  |  |  |  |
| *3.1.4* | *Тестирование сайта* | Кучеренко А.В. | 2 недели | 02.05.21 |  |  |  |  |
| *3.2* | *Тестирование приложения* |  | 3 недели | 09.05.21 |  |  |  |  |
| *3.2.1* | *Проверка правописания* | Япрынцев А.М. | 1 недели | 02.05.21 |  |  |  |  |
| *3.2.2* | *Приведение слов к начальной форме* | Япрынцев А.М. | 1 недели | 02.05.21 |  |  |  |  |
| *3.2.3* | *Удаление слов не несущих смысла* | Япрынцев А.М. | 2 недели | 02-05-21 |  |  |  |  |
| *3.2.4* | *Синонимайзер* | Япрынцев А.М. | 1 недели | 09.05.21 |  |  |  |  |
| *3.2.5* | *Алгоритм нечеткого поиска* | Япрынцев А.М. | 2 недели | 09.05.21 |  |  |  |  |
| *3.2.6* | *Логирование всех введенных в поиск запросов* | Япрынцев А.М. | 2 недели | 09.05.21 |  |  |  |  |
| *3.2.7* | *Внедрение на сайт* | Николаев И.А. | 1 недели | 16.05.21 |  |  |  |  |
| *3.2.8* | *Тестирование поисковой системы, выявление уязвимостей и доработка словарей* | Кучеренко А.В. | 3 недели | 02.05.21 |  |  |  |  |
| *3.3* | *Создание админки* |  | 3 недели | 02.05.21 |  |  |  |  |
| *3.3.1* | *Аутентификация пользователя* | Николаев И.А. | 1 недели | 02.05.21 |  |  |  |  |
| *3.3.2* | *Создание базовых страниц и шаблонных таблиц* | Николаев И.А. | 1 недели | 02.05.21 |  |  |  |  |
| *3.3.3* | *Затачивание каждой страницы под конкретные требования (Дашборд, Категории, QA, Control Version System, Synonyms, Extra Words, Preview)* | Николаев И.А. | 1 недели | 09.05.21 |  |  |  |  |
| *3.3.4* | *Система сортировки записей для настройки популярности* | Николаев И.А. | 1 недели | 16.05.21 |  |  |  |  |
| *3.3.5* | *Настройка WYSIWYG – редактора* | Николаев И.А. | 1 недели | 16.05.21 |  |  |  |  |
| *3.3.6* | *Перезапись статичных страниц* | Николаев И.А. | 1 недели | 16.05.21 |  |  |  |  |
| *3.3.7* | *Тестирование и выявление уязвимостей админки* | Кучеренко А.В. | 1 недели | 16.05.21 |  |  |  |  |
| ***Внедрение*** | | | | | | | | |
| *4.1* | *Оформление MVP* | Вся команда | 2 недели | 24.05.21 |  |  |  |  |
| *4.2* | *Внедрение MVP* | Вся команда | 2 недели | 24.05.21 |  |  |  |  |
| *4.3* | *Написание отчета* | Вся команда | 2 недели | 24.05.21 |  |  |  |  |
| *4.4* | *Оформление презентации* | Вся команда | 2 недели | 24.05.21 |  |  |  |  |
| *4.5* | *Защита проекта* | Вся команда | 1 неделя | 07.06.21 |  |  |  |  |

Определение проблемы

Тысячи выпускников ежегодно сталкиваются с такими вопросами: Как поступить? Куда поступать? Чем один вуз лучше другого? Какое направление выбрать? Чтобы получить ответы и выбрать правильный путь в своё светлое будущее, абитуриенты ищут информацию о различных учебных учреждениях, начинают писать в медиацентры разных вузов, из-за чего работникам приходится справляться с большим объёмом работы. При этом выпускники проживают в различных регионах нашей страны и забывают про различия в часовых поясах, они ждут быстрого и развёрнутого ответа, но в силу длительности рабочего дня сотрудники не могут ответить на некоторые вопросы.

Для того чтобы упростить поиск информации для выпускников и снизить нагрузку на сотрудников медиацентра, необходимо разработать веб-ресурс, позволяющий за относительно небольшой промежуток времени, ознакомить абитуриента с правилами приема.

Подходы к решению проблемы

Для решения поставленной проблемы заказчиком было предложено решить проблему одним из трех способов: создать игру, которая бы позволила познакомить пользователя с правилами поступления, веб-сайт или бота. Проанализировав все варианты, мы пришли к выводу, что: для разработки игры потребуется более углубленные знания ряда программ. К тому же, чтобы пользователь мог поиграть в игру, ему необходимо будет скачать её, на что пойдёт не каждый пользователь интернета, также свою роль играет тот факт, что в таком случае придется адаптировать игру под Android и iPhone одновременно. Если делать игру внутри браузера, то потребовался бы детальный разбор ряда программных обеспечений, что привело к осложнению решения поставленной задачи. Поэтому наша команда, проанализировав все аспекты других подходов, начала разрабатывать бота и веб-сайт, которые работали бы в паре. Таким образом, пользователь сможет удобно найти нужную информацию внутри мессенджера или с помощью официального сайта Уральского Федерального Университета.

Анализ аналогов

У Уральского Федерального Университета уже была страница на официальном портале, которая выполняла роль информирования поступающих о правилах приема. К ее минусам можно отнести то, что внутри этой страницы не было разделения часто задаваемых вопросов на категории, все вопросы располагались на одной странице. Полный список выложить не представлялось возможным, иначе результат – захламление информации о поступлении.

Но, помимо этого, существуют несколько сервисов-конкурентов, позволяющих создать своего чат-бота или FAQ-страницы для необходимых задач, такие как: «Botmother», «FAQ Bot», «Tilda».

«Botmother» и «FAQ Bot» позволяют создать чат-бота, однако решение не будет обладать необходимой гибкостью, ведь помимо ответов на вопросы, абитуриентов также необходимо ознакомить со всеми правилами приема, т.е. нужен открытый список частых вопросов, чего данные сервисы не предоставляют.

«Tilda» является конструктором сайтов для пользователей без опыта программного создания сайтов. Она не использует базу данных для хранения информации о контенте в блоках, а значит правки придется вносить прямо в визуальном редакторе. Это отсутствие динамичности элементов может плохо сказаться на визуальном отображении элементов сайта в будущем, усложнить сортировку вопросов по популярности

Таким образом, мы не нашли сервис позволяющий без проблем решить поставленную задачу

Требования к продукту и MVP

MVP

1. Возможность добавления, изменения, удаления вопросов, категорий админом – пользовательское требование. Система должна иметь возможность редактирования контента в базе данных посредством обработки введенных в формы данных – функциональное требование
2. Возможность сортировки вопросов, категорий – пользовательское требование. Сбор данных о порядке сортировки посредством JavaScript с передачей через обработчика в базу данных - функциональное требование
3. Возможность присвоения к категориям цвет - пользовательское требование. Использование WYSIWYG-редактора - функциональное требование
4. Возможность присвоения к каждой категории своей иконки - пользовательское требование. Использование Font Awesome в качестве иконочного шрифта. Вставка иконки через заполнение ее названия в форме - функциональное требование
5. Хранение всей редактируемой информации внутри БД – нефункциональное требование
6. Дизайн сайта должен соответствовать дизайну УрФУ – пользовательское требование

ПРОДУКТ

1. Возможность добавления, изменения, удаления синонимов (синонимичных слов), слов из «белого» (аббревиатуры) и «черного» (слова не несущие смысловой нагрузки) списков админом – пользовательское требование. Система должна иметь возможность редактирования контента в базе данных посредством обработки введенных в формы данных – функциональное требование
2. Возможность поиска по базе вопросов - пользовательское требование. Исправление орфографии, лемматизация текста, фильтрование по «черному» и «белому» спискам, замена синонимов и поиск по базе с помощью алгоритма Левенштейна - функциональное требование
3. Отображение пользовательских вопросов, заданных в боте на отдельной странице админ-панели - пользовательское требование. Запись пользовательских вопросов, заданных в боте в отдельную таблицу в БД - функциональное требование
4. Возможность добавление, редактирования, деактивации, активации, удаления, резервной смены пароля, разграничение прав доступа пользователей - пользовательское требование. Система должна иметь возможность редактирования пользовательских данных в БД посредством обработки введенных в формы данных, обеспечивая безопасность, запрещая дублирование логинов – функциональное требование

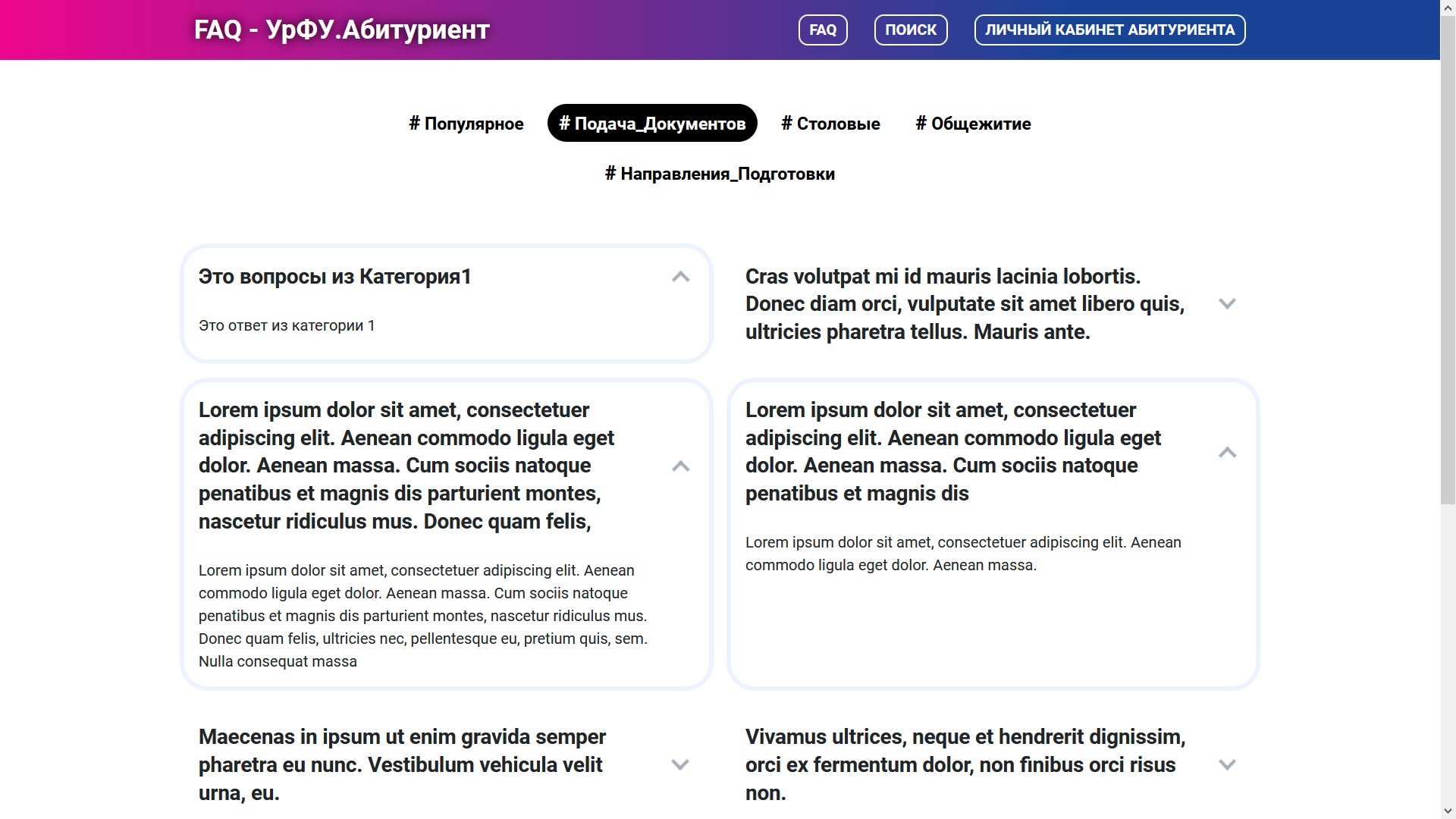
Стек для разработки

* Flask (фреймворк Python) – является основой проекта. Само приложение состоит из 3 частей: веб-сайт, бот, админ-панель. Также дополнительно используются его модули:
  + Flask-Classy – для разделения частей приложения на отдельные классы
  + Flask-Login – предоставляет функционал для авторизации пользователя
  + Flask-SQLAlchemy – работа с БД, безболезненный переход между разными типами БД
  + Flask-WTF – создание форм
* Jinja2 – шаблонизатор, используемый в Flask
* Дополнительные библиотеки для Python:
  + pyenchant – проверка орфографии и исправление слов
  + fuzzywuzzy – многофункциональная библиотека для нечётного сравнения строк
  + pymorphy2 – приведение слов к начальной форме
* Для создания сайта использовали стандартные средства web-разработки (HTML, CSS, JS). А также:
  + Bootstrap – библиотека стилей
  + Reboot – часть Bootstrap – приведение элементов сайта к одному внешнему виду
  + jQuery – фреймворк для упрощения работы с JS
  + Font Awesome – набор иконок
  + a-color-picker – WYSIWYG-редактор для выбора цвета
  + Masonry – библиотека для выстраивания элементов сайта по принципу «вставать, куда встается»
  + SortableJS – UI сортировщик
* VK API – для работы бота в социальной сети ВКонтакте
* SQLite – база данных, можно легко перейти на MySQL или PostgreSQL

Прототипирование

Начальные требования – сделать сайт с возможностью переключения по категориям и бота, отсылающего пользователя к этому сайту.

Наш прототип – статичный сайт с данным дизайном:

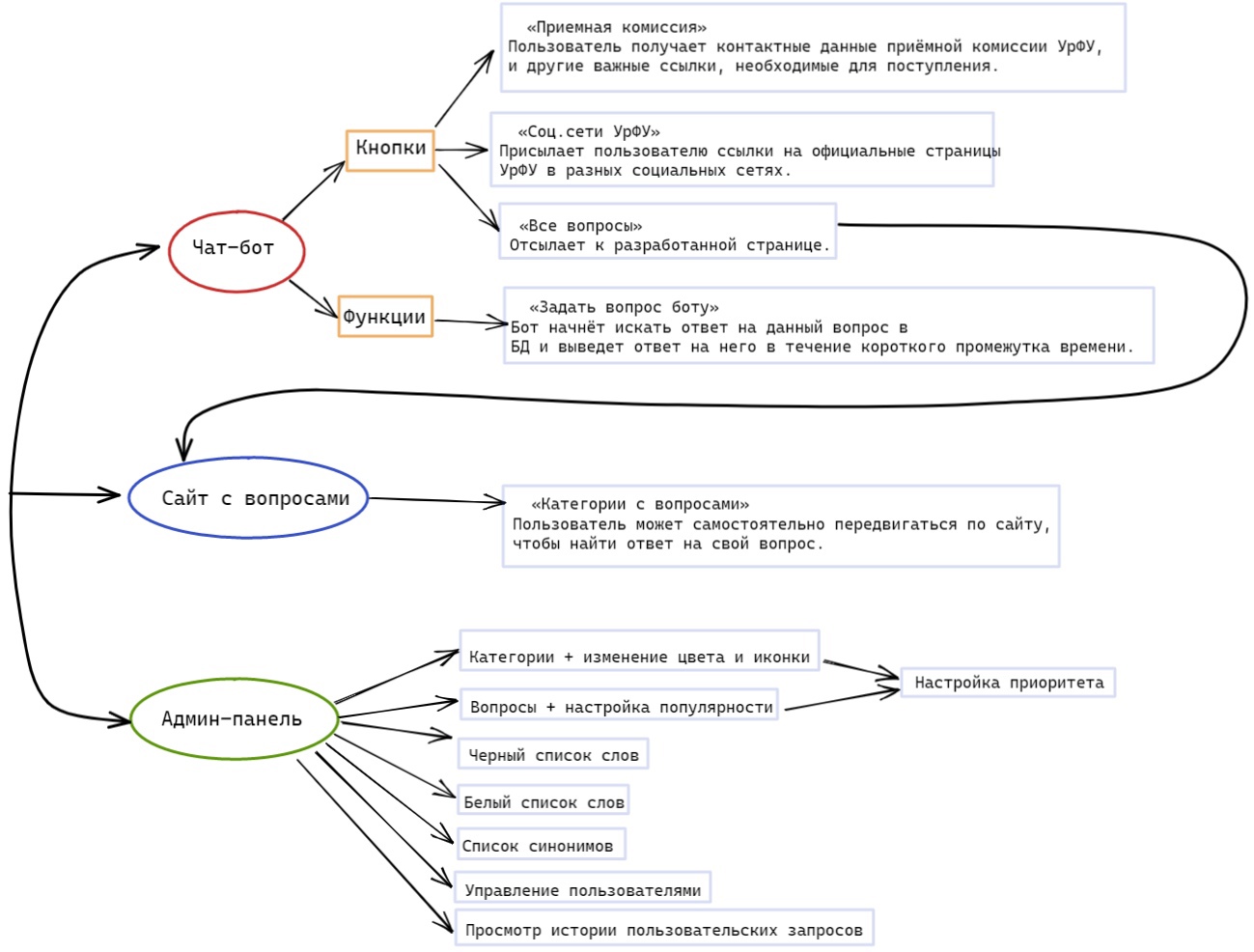


Он уже имел необходимые категории, кликаемые вопросы с раскрывающимися ответами, но заказчику не понравился внешний вид, и он был переделан. В дальнейшем прототип изменялся внешне и появлялась возможность редактировать контент.

Прототип бота на момент запуску имел только одну функцию – присылать пользователю при нажатии на кнопки важную информацию об УрФУ и ссылку на данный сайт. После встречи с заказчиком добавились новые функции, например: ответ на поставленный вопрос.

Разработка системы

Проект состоит из трёх основных частей – чат-бот, сайт с категориями вопросов и база данных.



Заключение

Конечный продукт полностью соответствует поставленным задачам, а именно: внутри социальной сети ВКонтакте внедрен бот, у которого пользователь может узнать ответ на свой вопрос, если он есть в базе данных, получить важные ссылки необходимые для поступления или другие официальные страницы УрФУ в различных соц. сетях; дизайн сайта полностью соответствует желаемому дизайну заказчика, все вопросы рассортированы по своим категориям для удобного поиска нужно информации; реализована админ-панель для сотрудников медиа-центра университета, с помощью неё они могут редактировать контент веб-сайта и просматривать запросы абитуриентов, настраивать поиск.

Для дальнейшей корректной работы системы нужно периодически пополнять её базу данных актуальными вопросами, а также вносить правки в настройки поиска. Важно следить за пользователями админ-панели, чтобы не допустить нарушения безопасности базы данных.

СПИСОК ЛИТЕРАТУРЫ

1. Чат бот для ВКонтакте на Python на Callback API [Электронный ресурс]. Статьи написания чат-бот ВКонтакте. Режим доступа: https://habr.com/ru/post/326898/, свободный. Дата обращения: 05.05.2021.
2. Нечёткое сравнение текстов на Python с Fuzzywuzzy [Электронный ресурс]. Статья написания кода для работы системы нечёткого поиска. Режим доступа: https://tirinox.ru/fuzzywuzzy-python/, свободный. Дата обращения: 23.04.2021.
3. Как проверить правописание с помощью Python pyenchant [Электронный ресурс]. Статья работе библиотеки enchant. Режим доступа: https://vc.ru/dev/174452-kak-proverit-pravopisanie-s-pomoshch-python-pyenchant, свободный. Дата обращения: 25.03.2021.
4. Введение во Flask [Электронный ресурс]. Работа с Flask. Режим доступа: https://pythonru.com/uroki/1-vvedenie-vo-flask, свободный. Дата обращения: 15.05.2021.
5. Сгенерировать пароль JS[Электронный ресурс]. Как разработать генератор паролей для админ-панели. Режим доступа: https://snipp.ru/jquery/generate-password-js, свободный. Дата обращения: 15.05.2021.