

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

**СЕРВИС ИНФОРМИРОВАНИЯ УЧАСТНИКОВ УЧЕБНОГО
ПРОЦЕССА В МЕССЕНДЖЕРЕ TELEGRAM**

Выпускная квалификационная работа
по направлению подготовки 44.03.04 Профессиональное обучение
(по отраслям)
профилю подготовки «Информатика и вычислительная техника»
специализации «Компьютерные технологии»

Идентификационный номер ВКР: 305

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующая кафедрой ИС

_____ Н. С. Толстова

« ____ » _____ 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
СЕРВИС ИНФОРМИРОВАНИЯ УЧАСТНИКОВ УЧЕБНОГО
ПРОЦЕССА В МЕССЕНДЖЕРЕ TELEGRAM

Исполнитель:

обучающийся группы № КТ-412

А. И. Палий

Руководитель:

ст. преподаватель

С. В. Ченушкина

Нормоконтролер:

Б. А. Редькина

Екатеринбург 2017

АННОТАЦИЯ

Выпускная квалификационная работа состоит из серверного приложения и пояснительной записки к нему на 65 страницах, содержащей 44 рисунка и 61 источник литературы.

Ключевые слова: ВУЗ, ИНФОРМИРОВАНИЕ, СЕРВИС, БОТ, РАСПИСАНИЕ, TELLEGRAM.

Палий, А. И. Сервис информирования участников учебного процесса в мессенджере telegram: выпускная квалификационная работа / А. И. Палий; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. информ. систем и технологий. — Екатеринбург, 2017. — 65 с.

В работе рассмотрены виды информирования участников учебного процесса и способы их улучшить.

Целью работы является создание серверного приложения для информирования участников учебного процесса. Для достижения цели были проанализированы методы создания современных серверных приложений и литература на данную тему. На основе сотрудничества с заказчиком был получен виртуальный сервер с доменным именем. После написания приложения, оно было запущено на сервере для массового использования.

СОДЕРЖАНИЕ

Введение.....	5
1 Использование современных средств информирования в высшем учебном заведении.....	7
1.1 Интернет сайт как основной источник информации в высшем учебном заведении.....	7
1.2 Социальные сети как средство общения сотрудников и студентов высшего учебного заведения	10
1.3 Videоканал на YouTube для просмотра видеоконтента высшего учебного заведения	12
1.4 Мобильные приложения для высшего учебного заведения.....	13
1.5 Использования сервиса (бота) в мессенджере Telegram для оперативного информирования	15
1.6 Обзор литературы и интернет-источников	18
1.6.1 Обзор литературы	18
1.6.2 Анализ интернет-источников	23
2 Описание сервиса информирования для мессенджера Telegram.....	29
2.1 Описание целевой аудитории, функции сервиса.....	29
2.2 Анализ исходного кода раздела с расписанием для разработки алгоритма получения данных	30
2.3 Функциональные требования и архитектура ервиса.....	33
2.4 Описание сред и библиотек для реализации сервиса (бота)	35
2.5 Описание работы бота на уровне сервера	38
2.6 Установка и настройка бота для использования в мобильном приложении.....	45
2.7 Возможность просмотра в десктопной версии Telegram.....	54
2.8 Размещение сервиса на сервер Российского государственного профессионально-педагогического университета.....	55

2.9 Описание сопроводительной документации.....	56
Заключение	57
Список использованных источников	58
Приложение	64

ВВЕДЕНИЕ

Сегодня время является одним из наиболее ценных ресурсов не только у работников сферы бизнеса и экономики, где от быстроты реакции зависят прибыль и успех, но и в сфере образования, где преподаватели и студенты ежедневно связаны с проблемами организации и проведения учебных занятий. Тем более, что в настоящее время технические инновации позволяют вышеперечисленной категории лиц носить с собой в карманах достаточно мощные электронные устройства, которые в обыденности воспринимаются, как средства связи.

Почти каждый участник образовательного процесса пользуется различными мессенджерами. Некоторые используют их для общения, связи на расстоянии, обсуждения дня или составления плана коллективных покупок, другие с их помощью договариваются о финансовых сделках или общаются с людьми из других регионов без преград в экономическом плане, так как за пользование мессенджерами, как правило, не взимается плата.

Одним из популярных мессенджеров является Telegram, который открывает перед пользователями еще одну категорию общения внутри мессенджера, где специально разработанные серверные приложения, называемые ботами, отвечают на пользовательские запросы, избавляя их от лишних переходов между приложениями, тем самым обеспечивая экономию времени и удобство работы в привычном окружении.

В Российском государственном профессионально-педагогическом университете учится больше 7 тысяч студентов, большая часть из которых часто используют сайт ВУЗа для просмотра информации о расписании и чтении новостей и объявлений, и для получения обновлений приходится постоянно проверять обновления на сайте.

Для повышения эффективности процесса информирования и расширения виртуального пространства для общения участников образовательного процесса в ВУЗе было решено создать сервис ускоряющий данный обмен.

Таким образом, **объект исследования** – процесс повышения эффективности информирования участников учебного процесса ВУЗа.

Предмет исследования – создание сервиса для информирования и общения участников учебного процесса РГППУ посредством кроссплатформенного мессенджера.

Цель работы: разработать сервис для информирования участников учебного процесса РГППУ посредством кроссплатформенного мессенджера Telegram с сопроводительной учебной документацией.

Для достижения поставленной цели необходимо решить следующие **задачи:**

1. Проанализировать литературу и интернет-источники по использованию современных средств информирования.
2. Проанализировать имеющиеся аналогичные проекты в различных образовательных учреждениях.
3. Подготовить алгоритм работы и архитектуру сервиса в рамках требований мессенджера.
4. Реализовать сервис информирования в выбранной среде программирования.
5. Подготовить сопроводительную учебную документацию по настройке и использования сервиса для администратора.

1 ИСПОЛЬЗОВАНИЕ СОВРЕМЕННЫХ СРЕДСТВ ИНФОРМИРОВАНИЯ В ВЫСШЕМ УЧЕБНОМ ЗАВЕДЕНИИ

1.1 Интернет сайт как основной источник информации в высшем учебном заведении

Первоначальным источником информации о вузе является официальный сайт вуза вся информация должна находиться на официальном сайте вуза. На сайтах вузов размещается информация о структуре университета, истории и информации для абитуриентов, студентов и преподавателей.

Официальный сайт образовательного учреждения является инструментом распространения информации, взаимодействия всех участников образовательного процесса и отражает бренд образовательного учреждения.

От содержания, организационной структуры и функционирования официального сайта зависит не только успех взаимодействия образовательного учреждения с внешним миром, но и управленческие и образовательные процессы, происходящие внутри учебного заведения. В современных условиях все большую популярность набирает новая маркетинговая стратегия для образовательного пространства России — брендинг. Это эффективный механизм формирования и повышения конкурентоспособности образовательного учреждения, создания прочной репутации, которая усиливает доверие сотрудников и потребителей. Задача официального сайта образовательного учреждения состоит в том, чтобы убедить потребителя (общественность) в преимуществах услуг, предоставляемых данным учреждением.

Образовательное учреждение должно выделить свою главную идею и придерживаться ее в своих коммуникациях со всеми целевыми аудиториями. Она должна служить отражением его миссии и воплощаться во всем, что представляет данное учреждение [12].

Так же некоторые сайты университетов позволяют участникам учебного процесса получать информацию об расписании той или иной группы, преподавателя или аудитории (рисунок 1).

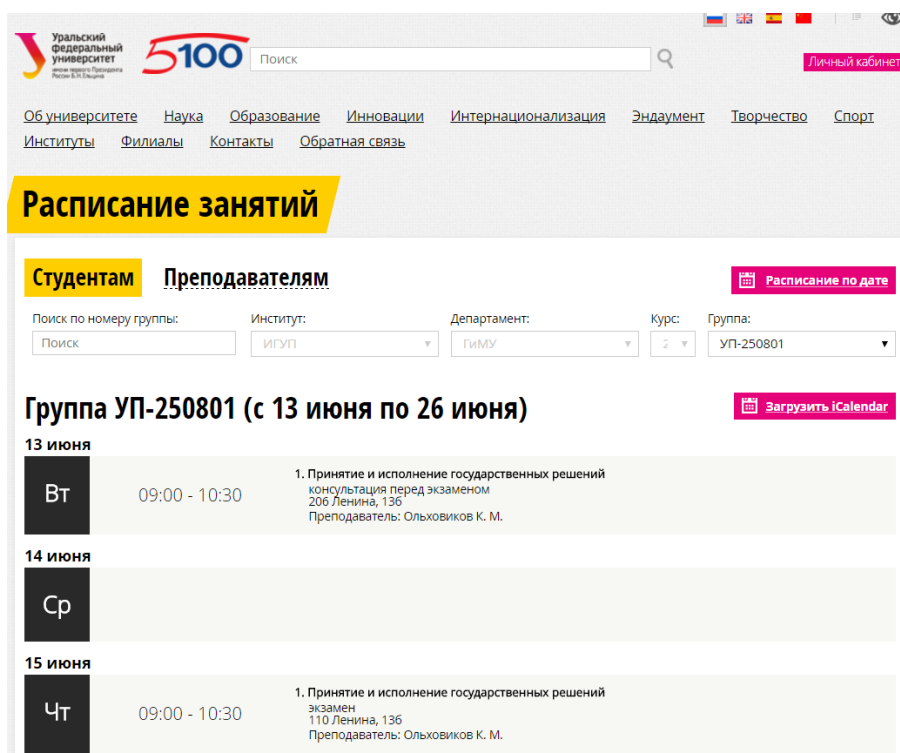


Рисунок 1 — Главная страница сайта Уральского федерального университета им Бориса Ельцина

Структура сайтов образовательных учреждений достаточно однообразна и, как правило, включает следующие разделы.

Главная страница. Дает пользователю представление о структуре сайта. Кроме того, на главной странице размещаются новости, анонсы мероприятий, т.е. та информация, которая может заинтересовать различные категории пользователей (в нашем случае это учащиеся и преподаватели).

Новости (учреждения и непосредственно сайта) могут быть отдельной страницей или совмещаться с другим разделом. Размещать раздел новостей стоит только в том случае, если в нем действительно есть необходимость и уверенность в его будущих обновлениях. При этом устаревшая информация не всегда бесполезна, просто располагаться она должна под другой вывеской, например, «архив».

Об учреждении (структура, история, сертификаты качества, шефы, партнеры, известные выпускники). В некоторых учебных заведениях работают курсы, музеи, центры и т.п., материалы о них чаще всего размещаются в отдельном разделе.

Сведения об образовательной организации. Обязательный пункт согласно требованиям федеральных законов.

Абитуриенту. Этот раздел содержит информацию о вступительных экзаменах в Вузы или Сузы, график работы приемной комиссии, проходной балл и т.п. или информацию о проведении экзаменов, ЕГЭ в школах, гимназиях, лицеях.

Студенту (расписание, образовательная модель, программы обучения, специальности, методики преподавания, методические материалы по предметам). Также в этом разделе, кроме списка учащихся и сведений о них, можно предусмотреть место, где ученики (или студенты) могут представить свои творческие работы. Такие работы могут быть представлены как личные странички учащихся или групп. Также в этом разделе может приводиться расписание занятий, текущая успеваемость.

Преподавателю (информация, сертификаты, аттестация). Сайт также может предоставлять возможность преподавателям размещать свои материалы (вплоть до отдельного раздела по предмету или кафедре).

Контакты. Могут включать в себя почтовый адрес, схему проезда, телефоны, E-mail или форму отправки сообщения для администрации или группы поддержки сайта.

Форум, гостевая книга. Гостевая книга — это место на сайте, где ваши посетители могут оставлять свои сообщения: отзывы на сайт, замечания, пожелания и т. д. Здесь же хранятся и ранее опубликованные сообщения.

В отличие от форума, посетители гостевой книги не могут отвечать на сообщения, оставленные другими посетителями, и тем самым завязывать

дискуссию или полемику по какому-то вопросу. Однако владелец сайта имеет возможность прокомментировать любое из сообщений гостевой книги.

Что касается гостевых книг, то следует учитывать, что только активно обновляемый «живой» сайт пользуется авторитетом и поддержкой учащихся. Если не обеспечено развитие сайта, бессмысленно заводить гостевые книги и прочие интерактивные формы общения. Результат будет отрицательным, т.е. в гостевой книге будет накапливаться различные критические замечания, во все не придающие значимости интернет-ресурсу.

Форум сайта может использоваться как элемент дистанционной поддержки обучения (например, виртуальный консультационный пункт). В таком случае, он должен обязательно обслуживаться ответственными за него (модераторами). Очень важно, чтобы модераторы следили за соблюдением правил, консультировали, удаляли некорректные сообщения, вносили интересные темы для обсуждений и пр.

1.2 Социальные сети как средство общения сотрудников и студентов высшего учебного заведения

Большинство университетов имеют странички в данных социальных ресурсах для привлечения внимания общества и новых абитуриентов к учебному заведению представляемому данной страничкой.

Отдельного рассмотрения требует работа в социальных сетях, так как основная целевая аудитория молодежь и для них актуально получать большую часть информации из сети интернет, а именно из социальных сетей. Поэтому необходимо тщательно налаживать коммуникации именно в социальных сетях.

Социальная сеть — это онлайн-сервис, сайт, позволяющий создавать социальные связи, строить взаимоотношения, распространять информацию и др. Главные особенности социальных сетей [2]:

- большие возможности для обмена самой текстовой и мультимедийной информацией (текстовой, фото, видео; сервисы сообществ и микроблогов; возможность указывать место, отмечать фотографии и т. д.);
- индивидуальные профили страниц с максимальным количеством личной информации (ФИО, вуз, место работы, фото и т. д.);
- добавление в «друзья» в первую очередь по принципу наличия реальной связи (друзья, знакомые, одноклассники или одногруппники, родственники и т. д.) и во вторую — по принципу схожести интересов (группы, сообщества и т. д.) [4].

Контент размещаемый в социальных сетях таких как vk или facebook нацелен на распространение новостей или информирование о каких-либо акциях или событиях, относящихся к вузу. Пример группы приведен на рисунке 2.

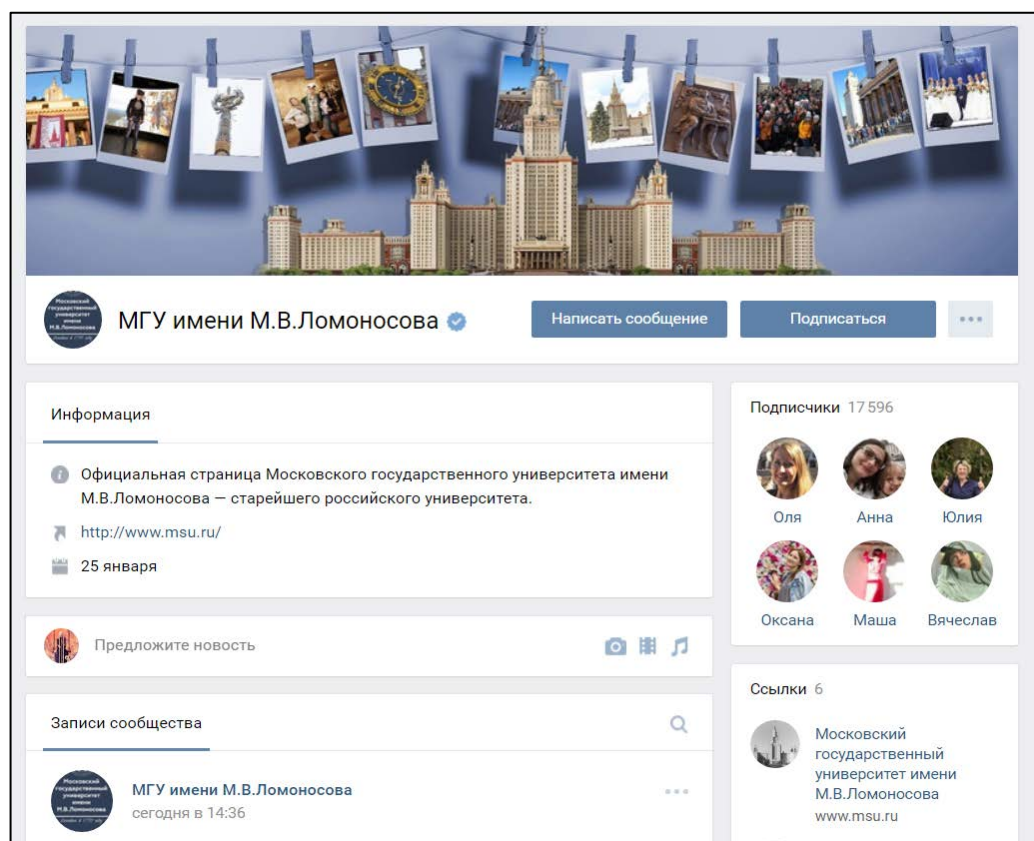


Рисунок 2 — Группа в социальной сети vk.com Московского государственного университета

1.3 Videоканал на YouTube для просмотра видеоконтента высшего учебного заведения

YouTube — видеохостинговая компания, предоставляющая пользователям услуги хранения, доставки и показа видео. Пользователи могут загружать, просматривать, оценивать, комментировать, добавлять в избранное и делиться теми или иными видеозаписями. Благодаря простоте и удобству использования YouTube стал популярнейшим видеохостингом и вторым сайтом в мире по количеству посетителей [20].

YouTube канал содержит в себе краткую информацию о вузе и видеоролики, созданные при поддержке вуза и относящиеся непосредственно к университету, который представляет данный канал.

Примером университета, у которого есть YouTube канал, может быть ЮУрГУ [11]. Данный канал содержит огромное количество видеозаписей, которые раскрывают университет со всевозможных сторон, таких как активность университета в научной или спортивной среде. Пример канала на YouTube приведен на рисунке 3.

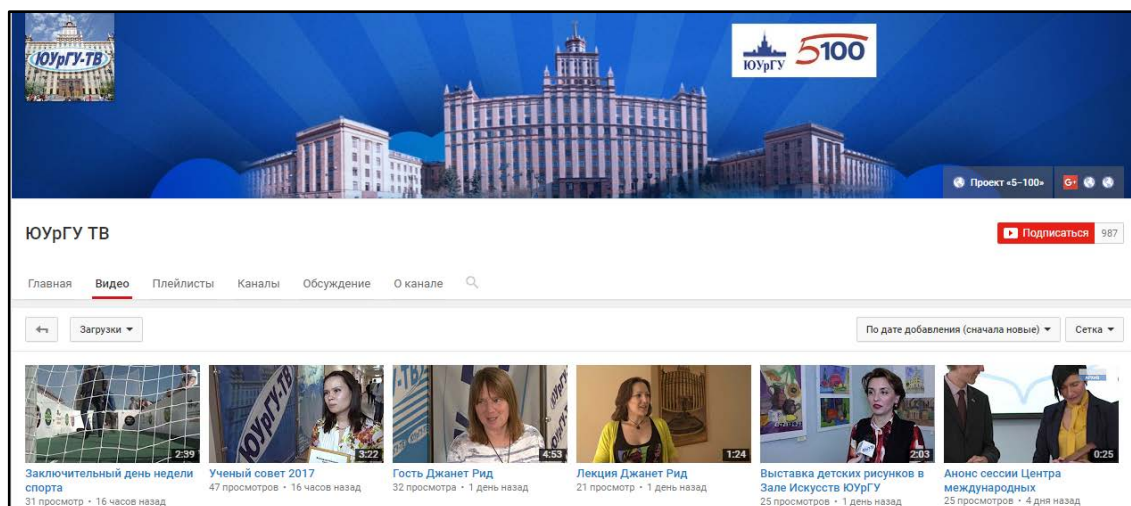


Рисунок 3 — Канал на видеохостинге YouTube Южно-Уральского государственного университета

1.4 Мобильные приложения для высшего учебного заведения

Сегодня каждый человек может бесплатно скачать, установить и настроить инструментарий для разработки мобильных приложений для дальнейшего их написания под выбранную операционную систему, а затем разместить свою работу в специализированных интернет-магазинах с целью полностью платного, бесплатного, а так же freemium скачивания и использования программы.

Приложения должны быть не только информативными, развлекательными и интерактивными, но и быть полезны в практическом или коммерческом смысле. Они должны использовать мало трафика, быть максимально независимыми от сети [18].

Бывают случаи, когда необходимо воспользоваться мобильным Интернетом, поскольку иного доступа к сети нет. Приложения могут сэкономить время и предоставить необходимую информацию, связанную, например, с местом нахождения пользователя в данный момент времени.

Поскольку экраны смартфонов относительно небольшие, самая важная информация о ВУЗе, должна быть заметна сразу же при входе на сайт с мобильного устройства.

Для преподавателей и работников учебных заведений, которым приходится работать с большим числом групп, студентов, коллег, ездить в командировки в различные филиалы и представительства, смартфон или планшет может стать хорошим помощником. Специализированные программы и приложения помогут облегчить рабочий процесс.

Выпускаются всё более новые, исправленные и улучшенные версии операционной системы Android.

Появляются специализированные приложения для преподавателей, которые помогают сэкономить время, упростить и ускорить рабочий процесс.

В наше время множество вузов разрабатывают мобильные приложения которые переносят в себя часть функционала сайта, к таким возможностям

можно отнести: просмотр расписания, просмотр новостей, просмотр различная информации об учебном заведении.

Приложения разрабатываются на две платформы – Android и iOS, которое имеет более широкий функционал, но отличается по дизайну и расписание не отображается в реальном времени. На рисунке 4 приведен снимок экрана, с примером использования приложения «МФТИ» на платформе Android.

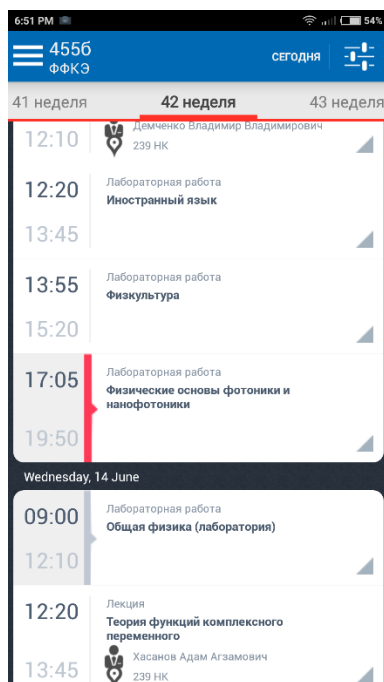


Рисунок 4 — Приложение «МФТИ»

Для преподавателей и работников учебных заведений, которым приходится работать с большим числом групп, студентов, коллег, ездить в командировки в различные филиалы и представительства, приложение может облегчить рабочий процесс и сэкономить время.

Основные преимущества приложения:

1. Мобильность. Нужная информация всегда у вас под рукой.
2. Доступность. Приложение является бесплатным и скачивается с сайта ВУЗа.
3. Обновление. Выходят новые и усовершенствованные версии.

1.5 Использование сервиса (бота) в мессенджере Telegram для оперативного информирования

Telegram — бесплатный кроссплатформенный мессенджер для смартфонов и других устройств, позволяющий обмениваться текстовыми сообщениями и медиафайлами различных форматов [58]. Используются проприетарная серверная часть с закрытым кодом, работающая на мощностях нескольких компаний США и Германии и несколько клиентов с открытым исходным кодом, в том числе под лицензией GNU GPL.

Telegram работает в двух режимах в Десктоп – для настольного компьютера и в качестве мобильного приложения. Пример запущенного приложения на компьютере показан на рисунке 5.



Рисунок 5 — Пример Telegram Desktop

Количество активных пользователей сервиса на февраль 2016 года составляло более 100 млн человек, а количество ежедневно пересылаемых сообщений достигло 10 миллиардов на август 2015.

Telegram, как программа-мессенджер, предназначена для отправки сообщений, никакой ориентации на образовательный процесс не имеет. Однако, в ней, на наш взгляд, имеются широкие возможности для использования этой программы в ВУЗе.

В первую очередь это организация взаимодействия между учащимися при групповой работе, а также при создании проекта, где может потребовать-

ся отправка файлов и фотографий, голосовых заметок, создание групп и каналов (рисунок 6).

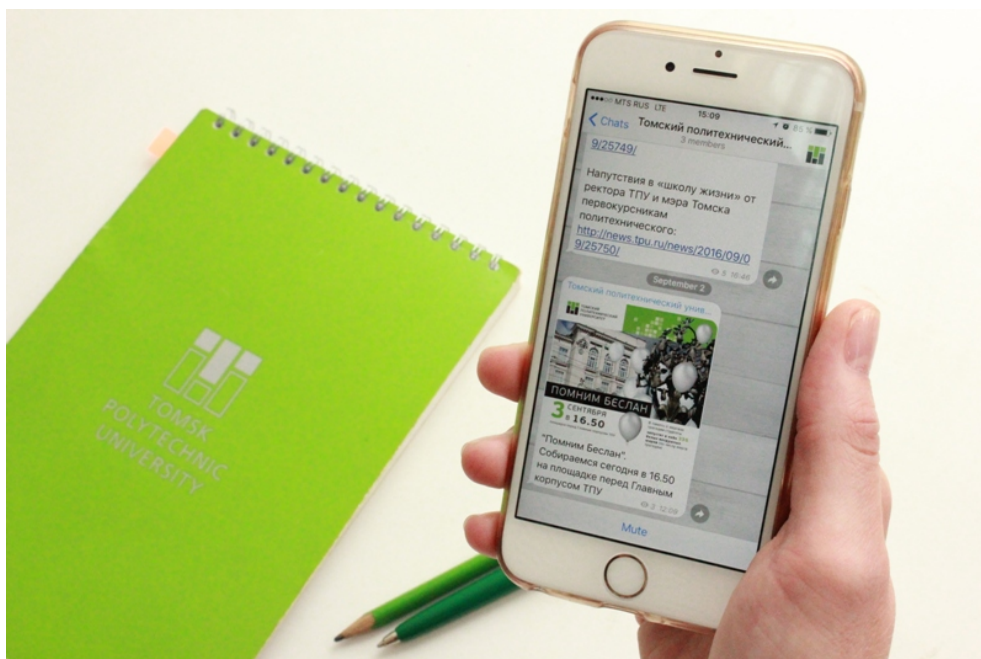


Рисунок 6 — Передача мультимедиа через Telegram

Именно каналы, а также нераспространенная для мессенджеров функция создания ботов делают Telegram весьма функциональной и привлекательной программой для организации процесса обучения. Каналы можно использовать для обмена сразу со многими пользователями. Они выполняют функцию своеобразной массовой рассылки.

Существует большое количество популярных информационных и образовательных каналов, которые могут использоваться как для самообразования, так и для получения информации во время урока или создания проекта. Также каждый учитель (или даже ученик) может создать свой канал, который станет аналогом блога или будет предназначен для представления хода проектной работы всем его участникам. Канал наполняется только сообщениями от узкого круга лиц, поэтому не содержит лишней и отвлекающей информации.

Роботы (или боты) – специальные аккаунты в Telegram, которые могут автоматически обрабатывать и отправлять сообщения и выполнять практически любые задачи, которые только можно придумать.

Боты в Telegram дают огромные возможности при правильном их использовании. Они могут применяться как в групповой работе в общих чатах, так и отдельным пользователем, предлагая и принимая от него данные, а при необходимости и сохраняющие их на специальном сервере. Отличие бота от канала состоит в его интерактивности. Он не просто навязывает определенный контент, а позволяет конструировать запрос.

Популярны боты, сообщающие погоду, курс валют, перевод слов на нужный язык. Кроме этого, существуют боты, имитирующие искусственный интеллект и очень полезны, к примеру, в изучении иностранных языков, поскольку позволяют имитировать общение с реальным человеком. Не менее актуальны боты, проводящие викторины, тесты, устраивающие опросы и голосования.

В мессенджере Telegram боты играют очень важную роль. Они способны быть администраторами в переписках, регулировать сообщения, отправляемые пользователями, в случае, если сообщения содержат в себе аудиозаписи, картинки, стикеры или могут проверять сообщения пользователей на наличие нецензурной лексики. Такие боты используются во многих чатах. Для образовательного учреждения боты создают для получения данных о расписании и чтении новостей и объявлений. Схема условная работы бота представлена на рисунке 7.

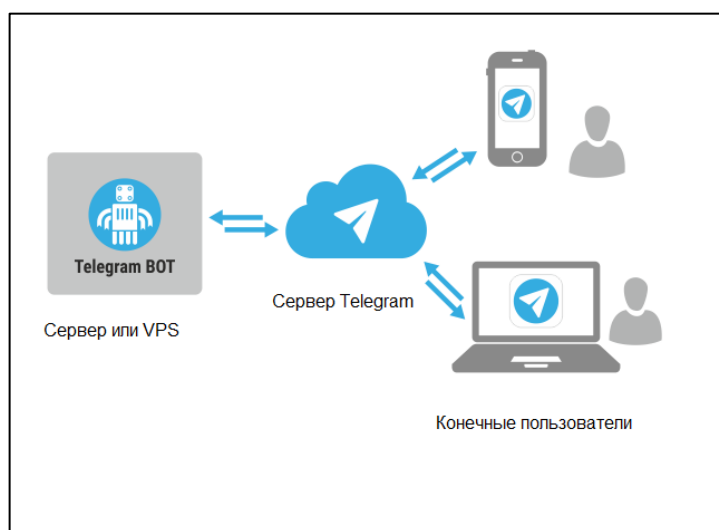


Рисунок 7 — Схема работы Telegram бота

К примеру, в чате «Ubuntu» есть бот-администратор [43], который не позволяет пользователям прикреплять стикеры в сообщения. Есть боты, которые распространяют новости внутри мессенджера на определенные тематические каналы или информируют пользователей о том, что произошло определенное событие, на которое был установлен триггер. Примером такого бота может служить «ru_pythonjobs» [52]. Данный канал под управлением бота распространяет вакансии, содержащие в себе ключевые слова, с сайта hh.ru [37] в мессенджер пользователям. В среде людей, связанных с разработкой или системным администрированием, популярны боты, которые информируют автора данного бота о каких-либо неполадках, что ускоряет реакцию ответственного за ликвидацию неполадок лица. Последний тип ботов, это поисковые боты, например, бот поисковой компании «Yandex» [60]. Данный бот позволяет искать информацию различного характера внутри мессенджера.

Так как telegram является кроссплатформенным приложением, которое распространяется на Android, iOS, Windows Phone, Windows, macOS и Linux, а так же имеет Web клиент, что делает работу одинаковой на всех платформах, вне зависимости от того как это удобно пользователю. Единственным отличием будет то что Версии для клиентов macOS, Windows и Linux не сохраняют историю переписки внутри компьютера. Недостатком это можно считать в редком случае если у пользователя отсутствует интернет.

1.6 Обзор литературы и интернет-источников

1.6.1 Обзор литературы

В отличии от других источников информации, Python имеет небольшую базу книг. Связано это с тем, что разработка и печать книги занимает продолжительное время, за которое некоторые принципы работы в языке мо-

гут поменяться. Будем рассматривать литературу, описывающую язык Python 3.0 и старше.

«A Byte of Python» – это свободная книга по программированию на языке Python. Она может служить учебным пособием или руководством по языку для начинающей аудитории [39].

Данная книга используется в качестве учебного материала в различных учебных заведениях: курс «Программирование на Python» в Гарвардском университете; курс «Введение в разработку приложений» в Бостонском Университете; курс «Обработка геоданных» в Университете штата Мичиган; курс «Основы машинных вычислений» в Калифорнийском Университете в Дэвисе [59].

Книга состоит из 19 глав и 4 приложений. В первых шести главах читателей знакомят с языком: что за язык, когда создан, в чем разница между Python 2 и Python 3, установка и с первым примером программы «Hello, world!».

В последующих главах рассматривается остальной функционал и структура языка. Кроме того, книга используется в лаборатории реактивного движения в рамках проекта «Сеть дальней космической связи» в NASA – национальное управление по воздухоплаванию и исследованию космического пространства [42].

Следующая книга «Изучаем Python» [17], автором которой является ведущий специалист в области обучения языку программирования Python и автором самых ранних и наиболее популярных публикаций по языку. Марк Лутц известен в сообществе пользователей Python своими новаторскими идеями. Книга подходит для углубленного рассмотрения принципов работы языка Python.

Книга разделена на 39 глав и 2 приложения. В ней представлены основные типы объектов в языке Python, порядок их создания и работы с ними, а также функции, как основной процедурный элемент языка. Рассматриваются методы работы с модулями и дополнительными объектно-

ориентированными инструментами языка Python – классами. Включены описания моделей и инструкций обработки исключений, а также обзор инструментов разработки, используемых при создании крупных программ. Каждая глава завершается контрольными вопросами с ответами на закрепление пройденного материала, а каждая часть – упражнениями, решения которых приведены в приложении В.

Марк Пилигрим – разработчик программных средств, сторонник свободного программного обеспечения (СПО), работающий системным архитектором в IBM, а в настоящее время программный разработчик в корпорации Google. Он человек-практик, который в своей рабочей деятельности сталкивается куда больше с практикой, чем с теорией, как Марк Лутц. Поэтому в своей книге – «Погружение в Python3» он делает акцент к практическому применению Python, его реализацию в повседневной жизни. В его творчестве есть другая книга – «Погружение в Python», посвящённая Python 2. Следуя идеалам СПО, распространяет свои книги в свободном доступе, кроме бумажных версий [46].

Книга «Погружение в Python 3» рассказывает о Python 3 и его отличиях от Python 2. По сравнению с «Погружением в Python», она состоит примерно на 20 % из переписанного и на 80 % – из нового материала.

Содержание книги состоит из 17 глав и 4 приложения. В первой половине книги рассказывается о встроенных типах данных, генераторах, регулярных выражений и итераторов.

Особое внимание уделяется итераторам – им в книге посвящены 2 главы. Связано это с тем, что итераторы в Python 3 встречаются везде, и необходимо разбираться в них, потому что множество функции возвращают итераторы, в отличие в Python 2, когда они возвращали списки.

Последующие главы в книге рассказывают больше о работе с программными продуктами и идет уклон на работу в сети: тестирование программного кода, рефакторинга и веб-сервисы.

В Python 3 появляется новый модуль `pickle`, несовместимый с Python 2 – записывает файл с определённым форматом данных. Этому модулю посвящена отдельная тема «Сериализация объектов Python». В главе упоминается JSON – текстовый формат обмена данными, основанный на JavaScript и обычно используемый именно с этим языком, который не поддерживает один тип в Python – `bytes`. Автор показывает, как можно справиться с этим ограничением.

Книга Дэвида Бизли «Python. Подробный справочник» является попыткой создать «выжимку» знаний по языку Python, ухватить саму суть и изложить читателю [2].

Книга компактна, нацелена на суть дела и написана очень доступным языком. Детально обсуждаются не только ядро языка, но и наиболее важные части стандартной библиотеки Python. Дополнительно освещается ряд тем, которые не рассматриваются ни в официальной документации, ни в каких-либо других источниках.

Предлагается практическое знакомство с особенностями Python, включая генераторы, сопрограммы, замыкания, метаклассы и декораторы. Подробно описаны новые модули, имеющие отношение к разработке многозадачных программ, использующих потоки управления и дочерние процессы, а также предназначенные для организации сетевых взаимодействий и работы с системными службами.

В полностью переработанном и обновленном четвертом издании улучшена организация материала, что позволяет еще быстрее находить ответы на вопросы и обеспечивает еще большее удобство работы со справочником. Книга отражает наиболее существенные нововведения в языке и в стандартной библиотеке, появившиеся в Python 2.6 и Python 3.

По уверениям автора, книжка не может являться учебником по языку Python. Написана она для опытных программистов, которые уже имели опыт взаимодействия с другими языками программирования, как C или Java.

В первой части этого справочника кратко рассказывается о языке программирования Python как таковом – описываются операторы, структуры программ, функции, классы и прочее. Имеется пара глав, посвященных среде выполнения, также тестированию, отладке и профилированию

Вторая часть содержит рассказ о стандартной библиотеке Python – встроенные функции, службы Python времени выполнения, работа с текстом и строками, подключение к базе данных (БД). Есть глава, посвященная обработке математических данных, также представлены различные известные алгоритмы и пути упрощения кода. Не забыто многопоточное программирование, а также работа с сетью. Отдельно рассказано про разработку веб-приложений, обработку и представление данных в Интернет.

Разработчик программного обеспечения из городе Сан-Франциско(США) Альберт Свайгарт предоставил книгу собственного написания – «Создание игры с Python & PyGame» [57].

Это издание охватывает библиотеку PyGame с исходным кодом для 11 игр. «Создание игры» была написана как продолжение его книги «Invent с Python». Для знакомства необходимы некоторые основы программирования Python, а с помощью библиотеки PyGame можно расширить свои способности, а именно создания игр или программ с графикой, анимацией и работой со звуком.

Книга разбита на 10 глав, 2 из которых затрагивают установку и основы PyGame, остальное – исходный код для 11 игра. А каждая из этих глав дает полный код для новой игры и учит концепции программирования из примеров. Игры являются клонами классиков, таких как Nibbles, Tetris и другие.

Книга имеет две версии: бумажное и электронное. Электронная версия доступна для свободного скачивания и пользования. Программы с исходным кодом, которые представлены в книге, распространяются свободно, их можно использовать в любых проектах.

Как говорилось ранее, Python используется в различных областях. Например, Rosalind – образовательный ресурс и веб-проект для обучения

биоинформатике с помощью решений задач [54]. Для этого используется язык программирования Python. Основы которого даются на самом ресурсе. Всего за 6 занятий обучающего вводят в синтаксис Python, тем самым показывая низкий порог вхождения и быстроту изучения представленного языка программирования.

1.6.2 Анализ интернет-источников

В глобальной сети существует множество информационных ресурсов по рассматриваемой теме, подавляющее большинство из них в свободном доступе. Рассмотрим некоторые. Стоит отметить, что огромное количество статей, руководств, презентаций и видеоуроков написана на английском языке, что предоставляет определенную преграду для обучающихся, которые имеют небольшие знания по английскому языку.

Стандартная документация по языку находится на главном сайте Python [27]. В целом, к нему обращаются за помощью, как к основному источнику для разрешений затруднений или трудных вопросах.

Это справочное руководство описывает синтаксис и основную семантику языка. Справочник лаконичен, но в тоже время является точным и полным. На сайте расположена несколько документаций для различных версий Python, а именно для версий 2.6, 2.7, 3.2, 3.3, 3.4.1. Каждой рассматриваемому элементу приводится теоритическое объяснение с последующим практическим примером. Единственное затруднение – данные разделы представлены на одном языке, на английском.

Выявленное затруднение обходится переведенной документацией силами добровольных переводчиков «Учебник Python 3.2». Документация была переведена на русский язык со всеми разделами, которые имеются в документации версии 3.2 из официального сайта. Дальнейшее развитие, и корректировка к более последней версии Python, у данного руководства не наблюдается [7].

Сайт «Nabrahabr» является многофункциональным, представляющий собой смешение новостного сайта и коллективного блога, созданный для публикации новостей, аналитических статей, связанных с информационными технологиями, бизнесом и Интернетом. Был основан в июне 2006 года.

В нем есть лента новостей, посвященная языку Python и все что с ним связано. Данный источник может быть оценен как вспомогательный, из него можно узнать различные решения проблемных вопросов.

Подобный сайт, но уже среди англоязычной пользовательской среде, является – reddit, новостной сайт с уклоном в социальный.

Пользователи могут размещать ссылки на какую-либо понравившуюся информацию в Интернете, и с помощью системы голосования они, пользователи, голосуют за интересные сообщения – наиболее популярные из них оказываются на заглавной странице сайта [53]. Выделано отдельный раздел для Python, чтобы пользователи были в курсе событий, связанными с данным языком, не пропускали важнейших новостей. Например, подходит IDLE для имитирования поведения определённого рода жидкости в дипломной работе по физике, или что для этого лучше использовать.

Сайт «ibm.com» является сайтом одной из крупнейших в мире корпорации производителя и поставщика аппаратного и программного обеспечения, а также ИТ-сервисов и консалтинговых услуг [38]. Содержит множество статей на различные темы, связанные с компьютерными технологиями, кроме того и про Python. Статьи рассказывают не только о возможностях языка, но и о различном его применении в компьютерных и научных областях. Однако, этот материал не является учебным материалом в силу разноуровневой сложности статей. В нем можно найти интересные статьи и руководства для разработчиков со стажем.

Присутствуют различные серверы для самостоятельного и дистанционного обучения языку Python. Такие как coursea.org, edx.com, codecademy.com. Все они предоставляют бесплатный доступ к изучению различных курсов. Материал взят из учебных курсов заведений, подробно и понятно объясня-

ются основы Python. Но сложность заключается в том, что все предоставляется на английском языке, требуется знания английского. Для изучения Python'у посвящены некоторые курсы.

На сайте дистанционного обучения «edX» предоставлен электронный курс «Introduction to Computer Science and Programming Using Python» группой преподавателей Массачусетского технологического института (MIT). Университет занимает лидирующие позиции в престижных рейтингах университетов мира, является новатором в областях робототехники и искусственного интеллекта, а его образовательные инженерные программы признаются одними из лучших в мире [31].

Представленный курс является первой частью из двух объединенных курсов: «Computer Science and Programming Using Python» и «Introduction to Computational Thinking and Data Science». Они вместе предназначены для помощи людям, которые не специализируются в областях информатики или программирования, научить писать программы для решения различных задач. Для некоторых курсы будут использованы как базовые для дальнейшего обучения в области информатики.

Курс сосредоточен не на глубине знаний, а на их широте. Цель состоит в том, чтобы предоставить обучаемым кратное представление по разным темам. «Computer Science and Programming Using Python» охватывает различные понятия и правила вычислений в языке программирования Python, некоторые простые алгоритмы, тестирование и отладка, а также неофициальные введение в алгоритмической сложности, в некоторые простые алгоритмы и структуры данных.

На другом сайте дистанционного обучения «Coursera» расположен электронный курс «Введение в интерактивное программирование на языке Python», подготовленный преподавателями университета Райса.

Этот курс разработан, чтобы ввести в основы программирования в Python, научиться основам построения простых интерактивных приложений. Выбор Python пал потому, что является простым в освоении, высокоуровне-

вым языком, который используется во многих вычислительных курсов на Coursera. Разработали свою браузерную среду программирования, чтобы сделать процесс обучения легче. Она позволяет разрабатывать интерактивные приложения куда проще и легче. Эти приложения включают окна, реакции на клавиатуру и мышь. Главный упор сделан на строительстве простых интерактивных игр, таких как Pong, Блэджек и астероидов.

Рекомендуемая подготовка обучающихся для изучению курса особой не требуется. Лишь для тех, кто заинтересован в некоторой подготовке для более легкого изучения курса, рекомендуется самостоятельно пройти какой-либо обучающий сайт, например, codecademy.com.

Программировать может каждый – так гласит кредо веб-школы «Codecademy», которая работает с абсолютными новичками. Бесплатная регистрация, «медали» за каждый пройденный уровень и простое объяснение функции и переменных – все это обеспечило популярность веб-школы. На данном сервере изучают языки как JavaScript, Ruby, Python, язык гипертекста HTML и таблиц стилей CSS [34].

Курс обучения состоит из 12 разделов, каждый из которых состоит из несколько заданий. Сами задания состоят из маленьких упражнений: подсчитать длину имени, присвоить переменную, превратить буквы в заглавные. Каждое действие сопровождается автоматическим приободрением и инструкцией о том, что делать дальше. С помощью сервиса «Course Creator», которая позволяет каждому желающему писать упражнения. Можно сделать свой собственный курс или добавлять задания в уже существующие разделы основной программы. Создатели веб-школы считают, что максимальная интерактивность, работа «шаг за шагом» — наиважнейший подход обучения программированию.

Бесплатное дистанционное обучение в Национальном Открытом Университете «ИНТУИТ» [19] – это удобный способ получения знаний, которые помогут получить новую работу, занять более высокую должность, получить

повышение в квалификации. На данном ресурсе имеются три курса, связанных с Python.

Электронный курс «Язык программирования Python», подготовленный Романом Сузи, изучает сценарный язык программирования, его основные библиотеки и некоторые приложения. Курс выбран по причине того, что Python один из бурно развивающихся и популярных в настоящее время сценарных языков программирования [29].

Курс состоит из 14 лекции, 14 теста и завершается экзаменом. Весь материал написан доступно и сопровождается рисунками, схемами для большего понимания материала.

Следующие два курса разработаны Албертом Свайгартом. Это «Разработка компьютерных игр с помощью Python и Pygame» и «Разработка компьютерных игр на языке Python».

«Разработка компьютерных игр на языке Python» – электронный курс как научиться программировать собственные компьютерные игры. Этот курс будет полезен для начинающих, кто хочет проявить себя в компьютерном программировании, даже не имея никакого опыта в программировании или по разработке компьютерных программ [14].

Курс разбит на 20 лекции, первые 4 посвящены синтаксису Python. Каждая из остальных лекции является игрой, исходный код которой представлен в начале. Идет объяснение каждого шага. На протяжении всего курса будет 17 контрольных точек в виде тестов, в конце курса экзамен. Имеются 5 разделов с дополнительными материалами.

Следующий электронный курс «Разработка компьютерных игр с помощью Python и PyGame» является также по разработке графических компьютерных игр, но уже с подключением программной библиотеки PyGame. PyGame позволяет легко создавать программы с 2D-графики [15].

Курс начинается с краткого введения в библиотеку PyGame и функции, которые она имеют. На это все отведено 2 лекции, остальные 8 посвящены

играм. За весь курс обучающие напишут 8 контрольных тестовых заданий. В конце курса идет экзамен, представлен он в виде итогового теста.

Ресурс «Youtube» – предоставляющий услуги размещения видео на сервере. Позволяет добавлять, просматривать и комментировать видеозаписи, добавленные пользователями сайта [61].

На этом ресурсе можно обнаружить огромное количество видеороликов от уровня простого любителя до профессионального видео. Имеются обучающее видео по языку Python. Это видео можно разделить на два больших раздела: либо это объяснение чего-либо (например, плюсы и минусы языка или видео с конференций, таких как PyCon), либо это полноценные обучающие курсы (например, курс является введением в язык программирования Python от Computer Science Center).

Рассмотрим некоторые курсы. На ресурсе имеется страница MIT OpenCourseWare с различными курсами. Имеется также и курс «Introduction to Computer Science and Programming», о котором говорилось выше. Но здесь только видеолекции, без контроля усвоения знания.

На «Youtube» предоставлены уроки от компания Google в рамках Google Code University. Они отмечаются высоким качеством [33].

Курс включает в себя шесть видеолекции, где проводится обучение новичков и подробно рассказывается о разных особенностях и отличий Python. Время видеолекций занимает до 50 минут. С уроками еще идут задачи, которые можно скачать самостоятельно. Это простые задачи, идеально подходящее для изучения Python программистами, имеющим небольшой опыт работы с другими языками программирования.

2 ОПИСАНИЕ СЕРВИСА ИНФОРМИРОВАНИЯ ДЛЯ МЕССЕНДЖЕРА TELEGRAM

2.1 Описание целевой аудитории, функции сервиса

Сервис ориентирован на преподавателей и работников высших учебных заведений и выполняется по заказу Российского государственного профессионально-педагогического университета.

Задача сервиса – облегчить процесс информирования преподавателей и студентов, помочь держать необходимую информацию всегда под рукой.

Сайт РГППУ содержит в себе огромный набор информации, зачастую не нужный в повседневной жизни участников учебного процесса, но есть элементы которые необходимы почти каждый день, такие как расписание, «Таймлайн» или «Мои документы».

Зачастую приходится искать тот или иной ресурс на сайте, что бы перейти к нему. Из-за большой загруженности сайта различными ссылками данный поиск не удобен. Так же сайт не адаптирован под мобильные платформы, что делает работу с мобильного устройства неудобной.

По этой причине было решено вынести в бота такие элементы как:

1. Расписание звонков.
2. Расположение корпусов.
3. Часто используемые ссылки участниками учебного процесса:
 - «Таймлайн»;
 - «Электронная библиотека»;
 - «Мои документы»;
 - «Об университете».
4. Расписание занятий.

2.2 Анализ исходного кода раздела с расписанием для разработки алгоритма получения данных

Для работы сервиса необходимо определить данные, с которыми оно будет работать. Было установлено, что с сайта РГППУ необходимо получить список всех преподавателей и групп, а также разработать алгоритм получения расписания с сайта.

В первую очередь информацию о расписании занятий. РГППУ [25] имеет более удобную систему для информирования участников учебного процесса о расписании, так как на данном ресурсе позволяет фильтрация расписания по преподавателям и аудиториям. В расписании очного отделения (рисунок 8) любой участник образовательного процесса может посмотреть расписание любой из групп очного отделения или преподавателя, но в разделе расписания заочного отделения у преподавателя можно посмотреть, как занятия очного отделения, так и заочного.



Рисунок 8 — Страница раздела расписание сайта Российского государственного профессионально-педагогического университета

На сайте, при просмотре расписания, вне зависимости от того к какому подразделению оно относится, сегодняшний день подсвечивается отличным от основной таблицы цветом, что помогает его найти на экране. Данный пример отображен на рисунке 9. Но данный сайт ориентирован на версию

полноценного браузера, тогда как многие участники образовательного процесса используют мобильные устройства для просмотра расписания.

АТ-402

Предыдущий период 27.02.2017 - 12.03.2017	Выбранный период 13.03.2017 - 26.03.2017	Следующий период 27.03.2017 - 09.04.2017
ПОНЕДЕЛЬНИК		
13.03.2017		20.03.2017
1. 08:00 -		1. 08:00 -
2. 09:45 -		2. 09:45 -
3. 11:30 Автоматизированные системы управления на автомобильном транспорте (лекция) 0-122 Кухаренко А.И.		3. 11:30 -
		4. 13:45 -
4. 13:45 Автоматизированные системы управления на автомобильном транспорте (лаб. работа) 0-122 Кухаренко А.И.		5. 15:30 -
		6. 17:15 -
5. 15:30 Транспортное право (лекция) 0-113 Шайдурова Т.Ю.		7. 19:00 -
6. 17:15 -		
7. 19:00 -		
ВТОРНИК		
14.03.2017		21.03.2017
1. 08:00 -		1. 08:00 -
2. 09:45 -		2. 09:45 -
3. 11:30 Техническая эксплуатация автомобилей (лекция) 0-110 Лялин К.В.		3. 11:30 Техническая эксплуатация автомобилей (лекция) 0-110 Лялин К.В.
4. 13:45 Техническая эксплуатация автомобилей (практика) 0-110 Лялин К.В.		4. 13:45 Техническая эксплуатация автомобилей (лаб. работа) 0-110 Лялин К.В.
5. 15:30 Система автоматизированного проектирования в автомобилестроении (лекция) 0-110 Лялин К.В.		5. 15:30 Система автоматизированного проектирования в автомобилестроении (практика) 0-110 Лялин К.В.
6. 17:15 -		6. 17:15 -
7. 19:00 -		7. 19:00 -
СРЕДА		
15.03.2017		22.03.2017
1. 08:00 -		1. 08:00 -
2. 09:45 -		2. 09:45 -
3. 11:30 Транспортное право (практика) 2-523 Шайдурова Т.Ю.		3. 11:30 -
4. 13:45 История мировой автомобилизации (лекция) 0-110 Эльяш Н.Н.		4. 13:45 История мировой автомобилизации (лекция) 0-110 Эльяш Н.Н.

Рисунок 9 — Пример расписания одной из групп

Так как мобильная версия сайта находится в разработке и не предоставляет пользователю удобный интерфейс для выбора расписания, а само расписание с мобильного устройства почти каждый раз будет обновляться, что будет отнимать время участника учебного процесса. С учетом данных факторов работа является актуальной.

Так же поиск некоторой информации на официальном сайте вуза не всегда удобна, не важно с мобильного устройства или с компьютера. К таким ресурсам можно отнести электронную библиотеку, внутреннюю систему «Таймлайн» или информацию об образовательном учреждении.

Изначально существовало два способа получения информации о расписании:

- «парсинг» существующих страниц;
- вывод информации через базу данных расписание.

Сначала использовалась **технология «парсинга»**.

Расписание занятий на сайте университета находится в открытом доступе, что позволяет любому посетителю просматривать его. Для просмотра расписания отдельного объекта формируется GET-запрос, в котором указывается тип расписания и идентификационный номер объекта. GET-запрос - HTTP запрос, содержание которого формируется в адресной строке.

Для формирования расписания используется 2 типа GET-запросов:

- «v_gru» — расписание групп;
- «v_rgr» — расписание преподавателей.

Расписание преподавателей имеет расширенную версию, формирующуюся для заочного отделения, поэтому расписание преподавателей было решено получать именно таким способом.

Все идентификационные номера хранятся в тегах «select», которые позволяют пользователю в графическом режиме выбирать необходимое ему поле. В случае с серверным приложением выбор проводится внутри кода программы. Лучшим вариантом для быстрой работы бота является создание списка, содержащего внутри себя все номера и соответствующие им обозначения, и сохранение этого списка внутри отдельной базы данных.

Для формирования списка необходимо изучить HTML-страницу с расписанием. Опытным путем было установлено, что внутри тегов «select» имеются теги «option», внутри которых есть свойство «value», отвечающее за номер объекта, а содержимое данных тегов является названием группы или именем преподавателя. Необходимо будет добавить в модуль «parser.py» функции для получения данных списков, так как этот модуль отвечает за взаимодействие серверного приложения с сайтом университета.

Следующей задачей является анализ таблицы с расписанием. Это необходимо для разработки алгоритма, который будет передавать данные о выбранном расписании в серверное приложение. Всё расписание хранится в таблице с «id» = «timetable_tab9». В результате анализа был разработан алгоритм, позволяющий получать расписание по необходимой дате.

Данный алгоритм будет занесен в модуль «parser.py». Были написаны 3 функции, которые запрашивают расписание и выбирают из него определенный день или дни. Данными функциями являются:

- «get_schedule_today» — получение расписания на сегодня;
- «get_schedule_tomorrow» — получение расписания на завтра;
- «get_schedule_week» — получения расписания на неделю, начиная с сегодняшнего дня.

Во втором случае – **использование базы** данных напрямую через запросы.

В случае с расписанием, то лучше всего использовать для более быстрой работы базу данных, в которой хранится расписание. В РГППУ база данных которая содержит в себе расписание поставляется системой «Галактика». Для составления запросов было необходимо изучить строение базы данных и определить необходимые таблицы и поля в этих таблицах, которые используются для расписания.

В случае с каждым типом расписания необходимо написать функции которые будут приводить расписание из типа данных «словарь» в «строку», которая будет содержать внутри себя оформление и информацию о дне, за который отвечает данное расписание.

2.3 Функциональные требования и архитектура ервиса

Основной функционал Telegram-бота, в рамках данной работы, заключается в удобном и быстром получении информации участниками образовательного процесса. Для того, чтобы бот смог удовлетворять данному

условию, необходимо написать модели базы данных и хранить данные о пользователе внутри этой базы данных. Это позволяет участникам образовательного процесса получать их расписание в одно нажатие по команде.

Конечный пользователь взаимодействует с ботом через сервер Telegram, который перенаправляет данные на сервер приложения, где в блоке «Бизнес-логика» указаны модули, необходимые для работы приложения. Все это взаимодействует с базой данных напрямую или косвенно. Также присутствует односторонняя связь с сайтом университета РГППУ для получения расписания. Архитектура сервиса представлена на рисунке 10.

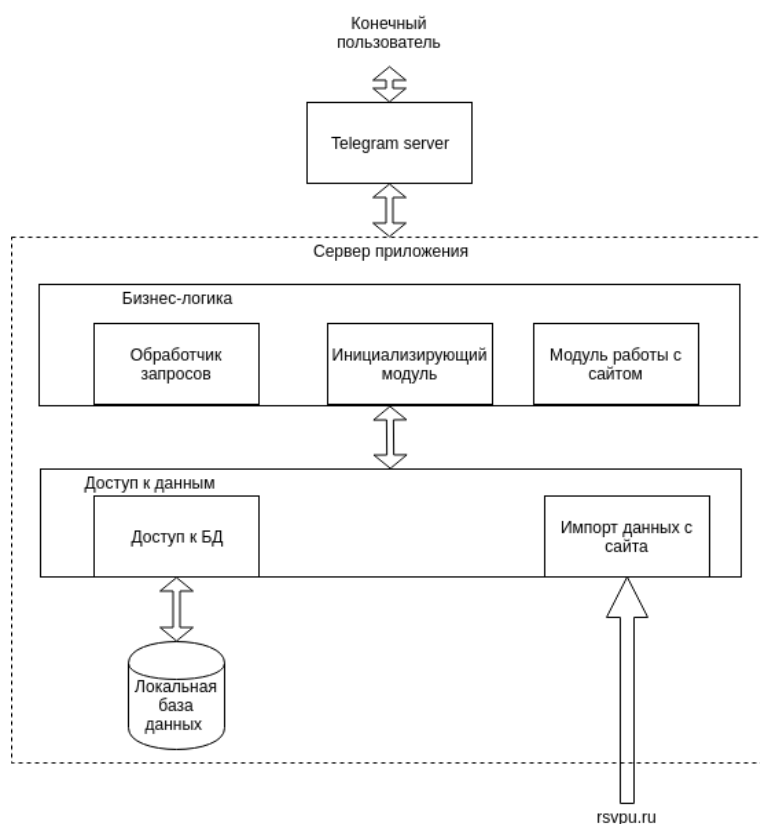


Рисунок 10 — Архитектура приложения

Дополнительный функционал бота расширяем путем добавления пользователю нового статуса и прописыванию функционала внутри статуса, тем самым повышается устойчивость приложения к различным сбоям. Дополнительными функциями были добавлены возможности получения расписания звонков, расположения корпусов и полезных ссылок на интернет ресурсы связанные с университетом.

2.4 Описание сред и библиотек для реализации сервиса (бота)

Создание любого бота начинается с написания Telegram-боту с именем «BotFather» [32]. Набор команд данного бота позволяет создать бота и провести все настройки, касающиеся его работы. Создание бота начинается с команды «/newbot», которая предлагает пользователю ввести название бота. После создания пользователю дается токен, с помощью которого автор бота может получать сообщения бота и отправлять сообщения от лица бота.

Затем можно изменять настройки бота, вводя команды. Ниже приведены примеры команд используемых для настройки и редактирования бота:

- «/setname» — установка имени бота;
- «/setdescription» — установка описания бота;
- «/setabouttext» — установка информации о боте;
- «/setuserpic» — установка аватара бота;
- «/setcommands» — установка команд бота;
- «/setinline» — переключение «инлайн» режима;
- «/setinlinegeo» — переключение «инлайн» запроса локации;
- «/token» — сгенерировать токен авторизации;
- «/revoke» — аннулировать токен бота;
- «/setjoingroups» — разрешение присоединения к группам;
- «/setprivacy» — настройки приватности.

Для написания бота необходимо выбрать язык программирования и библиотеку для данного языка. Для решения поставленных задач был выбран язык Python версии 3.5 [50], отличающийся большим количеством библиотек и высокой скоростью разработки. Программы, написанные на языке Python, легко запустить на сервере, потому что на сервере предустановлен интерпретатор языка Python 3.5.

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и

читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python (рисунок 11) поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).



Рисунок 11 — Логотип языка Python поверх исходного кода

Для работы с ботом было выбрано PyTelegramBotAPI [49], данный программный интерфейс приложения позволяет создавать обработчики событий в зависимости от вводимых пользователем данных.

Пример простого «эхо»-бота представлен на рисунке 12.

```
1 import telebot
2
3 bot = telebot.TeleBot("TOKEN")
4 @bot.message_handler(func=lambda m: True)
5 def echo_all(message):
6     bot.send_message(message.chat.id, message.text)
7
8 if __name__ == '__main__':
9     bot.polling(none_stop=True)
10
```

Рисунок 12 — Листинг простого «эхо» бота

Бот из примера будет отвечать пользователю таким же сообщением, которое пользователь отправил боту.

Для взаимодействия с сайтами внутри программ необходимо из кода программы сделать запрос по url-ссылке и работать с загруженным текстом, как с HTML-документом. Самым сложным вариантом выполнения данной работы является использование библиотеки `urllib.request` [56], и работа с полученным HTML-текстом, как с обычным текстом, но недостатком данной библиотеки является то, что необходимо написать свои собственные обработчики для каждой ситуации, что не является хорошим тоном в среде программистов на языке Python. В официальной документации [56] рекомендуется использовать библиотеку `requests`, предоставляющую высокоуровневый интерфейс работы с url-ссылками, что необходимо в рамках данной работы, так как для расписания формируется url-ссылка, включающая в себя GET-запрос.

Для высокоуровневой обработки HTML-кода была выбрана библиотека «BeautifulSoup4» [30]. Данная библиотека позволяет работать с HTML-тегами, как с отдельными объектами, что повышает качество кода и делает его более логичным при прочтении.

Выбранные библиотеки хорошо подходят для разработки модуля «`parser.py`», который будет отвечать за получение расписания с сайта университета и доставку его в серверное приложение, отвечающее за работу бота.

В качестве среды разработки была выбрана среда Visual Studio Code — кроссплатформенный редактор исходного кода, поддерживающий базовые возможности интегрированной среды разработки, созданный в Microsoft. Позиционируется как «легкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Visual Studio Code распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, и доступен в версиях для платформ Windows, Linux и OS X.

Для размещения был выбран веб сервер `nginx`, который по статистике Netcraft, держал 29% самых нагруженных сервисов.

Для отслеживания статуса и автоматического запуска приложения при неполадках сервера была выбрана программа Supervisor, предоставляемый из стандартных репозиториях Ubuntu Server.

Так как основной нашего веб приложения является библиотека Flask, которая базируется на Web Server Gateway Interface (WSGI), то для передачи данных рекомендуется между приложением и сервером рекомендуется использовать библиотеку, которая будет осуществлять передачу данных. Для этого была выбрана библиотека Gunicorn, как одна из самых активно-развивающихся библиотек для WSGI сервера.

2.5 Описание работы бота на уровне сервера

Серверное приложение, отвечающее за работу бота, использует библиотеку PyTelegramBotAPI. Основной модуль, именуемый «telegram_bot.py», принимает пользовательские запросы и, в зависимости от состояния пользователя и содержания, отправляет ему ответ.

Для каждого состояния пользователя был сделан отдельный обработчик, который получает пользователя из базы данных и генерирует нужный ответ. Бот является устойчивым к ситуациям, когда расписание группы находится в редактировании, а на сайте не доступна ссылка, сформированная для данной группы или преподавателя.

Бот разработан в двух вариантах «Парсинг» расписания. Основной модуль управляющий логикой поведения бота, делает запрос в модуль «parser.py», который методом «парсинга» получает необходимые данные с сайта. В функции основного модуля входят:

- отправка запроса на создание пользователя в модуль user;
- обработка сообщений пользователя;
- отправка сообщений пользователю;
- поиск возможных значений;
- изменение состояния пользователя;

- подготовка расписания к отправке пользователю.

Telegram имеет клиенты как для веб-браузера, различных мобильных устройств, на платформах, таких как Android, iOS, Windows phone, так и для операционных систем Windows, MacOS или семейства Linux. Поведение бота не отличается от клиентской оболочки пользователя, поэтому он всегда будет вести себя одинаково.

Внутри бота для удобства пользователей, были установлены подсказки к командам, выполняющим основные действия бота, каждое действие бот комментирует. Данные команды изображены на рисунке 13.

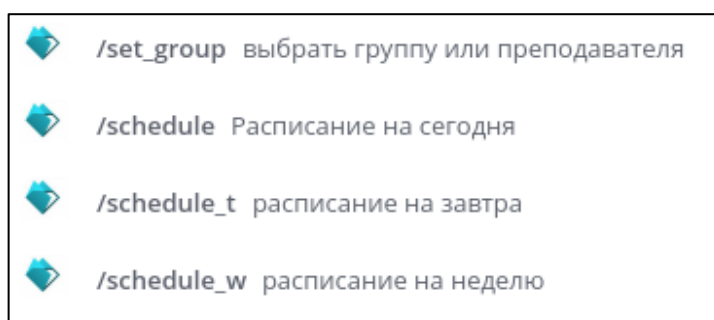


Рисунок 13 — Пример команд используемых в боте

Второй вариант – работа с базой расписания для этого был создан специальный модуль «shchedule_db», который отвечает за логику работы основного модуля telegram_bot с базой данных хранящей в себе расписание и список групп, преподавателей и аудиторий.

В окончательном варианте структура проекта состояла из следующих модулей:

- Telegram_bot;
- Models;
- Schedule_db;
- User;
- Gunicorn;
- State;
- Config;
- Contents.

Структура проекта открытого в редакторе изображена на рисунке 14.



Рисунок 14 — Открытый проект в редакторе кода

Все настройки подключения бота к серверу базу данных, токен самого бота или доменное имя сервера, на котором находится бот, прописаны в модуле config. Пример содержания модуля config изображен на рисунке 15.

```
1  """this module contain configure data to this bot"""
2  from collections import namedtuple
3  from peewee import SqliteDatabase
4
5  TOKEN = '379933414:AAH9ZrLoDZ5qGW0TNbvAC6_qcS6RZkx3_Bk'
6
7  _ScheduleDB = namedtuple('ScheduleDB', ['server', 'user', 'pwd', 'db'])
8
9  SCHEDULEDB = _ScheduleDB('192.168.222.4',
10 |                                     'galaxy_reader',
11 |                                     '*****',
12 |                                     'galaxy_2016_2')
13
14  LOCALBASE = SqliteDatabase('base.db')
15
16
17  HOST = 'tg.rsvpu.ru'
18
19
20
```

Рисунок 15 — Модуль config

Пример запроса расписания группы изображен на рисунке 16 Данный запрос вызывает функцию, которая обращается к базе данных, и возвращает словарь, ключом которого является время пары, а элементами связанными с данным ключом являются кортежи, с учетом выбранной даты и группы.

```
63 def _do_query(query) -> list:
64     cursor = _connect.cursor(as_dict=True)
65     cursor.execute(query)
66     return cursor.fetchall()
67
68
69 def schedule_group_query(group_id: int, day) -> dict:
70     """return schedule for entered group and selected day"""
71     group = _do_query(sql.select_group_name.format(id=group_id))
72     try:
73         if group: # if database have this group
74             pairs = _do_query(sql.schedule_group.format(date=day, id=group_id))
75             schedule = {}
76             for pair in pair_time:
77                 schedule[pair] = tuple(s for s in pairs
78                                     if s['start_time'] == pair)
79             return schedule
80         else:
81             raise DatabaseError
82             # todo error when group field is empty
83     except DatabaseError:
84         # todo make exception to db error
85         pass
86
```

Рисунок 16 — Часть модуля schedule_db

Модуль state содержит в себе все состояния пользователей, из данного модуля другие модули могут получить информацию о том, в каком состоянии находится сейчас пользователь. В зависимости от состояния пользователя, реакция того или иного модуля будет отличаться (рисунок 17).

```
1 states = {'StartMenu': 0,
2           'Menu': 1,
3           'Set_sub_schedule': 2,
4           'Settings': 3,
5           'Timeline_login': 4,
6           'Get_self_schedule_date': 5,
7           'Get_timetable': 6,
8           'Get_search_schedule_step1': 7,
9           'Get_search_schedule_step2': 9,
10          'Get_academic_buildings': 8}
11
```

Рисунок 17 — Модуль state

Модуль отвечающий за работу пользователя называется user. Он изолирует работу ORM внутри класса, где прописаны методы, которые позво-

ляют получить значение поля или установить другое, без дополнительных действий, которые обычно необходимы при работе с ORM такие как сохранение данных в базу или получение элементов находящихся внутри json поля (рисунок 18).

```
8 class ScheduleType:
9     Teacher = 0
10    Group = 1
11    Classroom = 2
12
13
14 class User:
15     def __init__(self, message=None, query=None):
16         if message:
17             self.chat_id = message.chat.id
18             self.user_id = message.from_user.id
19             self.user, self.new = models.User.get_or_create(user_id=self.user_id,
20                                                         chat_id=self.chat_id)
21         elif query:
22             self.user_id = query.from_user.id
23             self.user = models.User.get(user_id=self.user_id)
24         else:
25             raise ValueError
26
27     def __increment_query(self):
28         self.user.number_of_queries = self.user.number_of_queries+1
29
30     def __update_last_action(self):
31         self.user.last_action = datetime.datetime.now()
32
```

Рисунок 18 — Элемент модуля user

Модуль model содержит в себе модель пользователя, которая создается в базу данных и через которую идет обращение к базе данных, без использования structured query language(SQL) запросов, а используя функции внутри языка программирования на котором ведется разработка. Модель пользователя в готовом продукте изображена на рисунке 19.

```
15 class User(Model):
16     """this model contain user data"""
17     state = IntegerField(default=0)
18     state_data = CharField(default='{}')
19     sub_news = IntegerField(default=0)
20     sub_schedule = CharField(null=True)
21     timeline_login = CharField(null=True)
22     chat_id = IntegerField()
23     user_id = IntegerField(null=False, primary_key=True)
24     last_action = DateField(null=True)
25     number_of_queries = IntegerField(default=0)
26
27     class Meta:
28         database = LOCALBASE
```

Рисунок 19 — Модель пользователя

Модуль contents содержит в себе строковые данные, которые используются внутри бота, для кастомных клавиатур или ответов, которые присылает бот пользователю, так же тут содержится расписание, которое серверное

Данный модуль отвечает за подключение бота к серверу telegram. Этот модуль отвечает за приветствие пользователя и всю работу с пользователем. Подготовку сообщений к отправке и отправка этих сообщений. Данный модуль реализует событийно-ориентированную архитектуру. Данная архитектура заключается в изменении состояния пользователя при определенных событиях и отдельной реакции на каждое сообщение, в зависимости от состояния пользователя. Это позволяет разрабатывать отдельные модули для приложения, не обращая внимания на предыдущие модули, так как их работа не взаимосвязана и это не приводит к сбою внутри приложения. Так же этот модуль ведет логирование о поступившем сообщении и о состоянии выполнения тех или иных задач. Элементы главного меню из описанного модуля изображены на рисунке 22.

```
@bot.message_handler(func=compare_state(state.states['Menu']),
                      content_types=['text'])
def main_menu(message):
    """
    Main menu
    """
    usr = user.User(message)
    if message.text == TODAY:
        text = get_self_schedule(usr, schedule_db.Days.today())
        bot.send_message(usr.chat_id, text, parse_mode='MARKDOWN')
    elif message.text == TOMORROW:
        text = get_self_schedule(usr, schedule_db.Days.tomorrow())
        bot.send_message(usr.chat_id, text, parse_mode='MARKDOWN')
```

Рисунок 22 — Обработчик главного меню

Для корректной работы всех модулей необходимо было составить список всех использованных библиотек и занести их в файл requirements.txt с указанием версий использованных библиотек. Пример данного файла изображен на рисунке 23.

```
1  pymssql==2.1.3
2  flask==0.12.1
3  peewee==2.9.2
4  pytelegrambotapi==2.3.2
5  beautifulsoup4==4.6.0
```

Рисунок 23 — Содержание requirements.txt

После написания всех модулей необходимо настроить веб сервер, чтобы работа приложения была более стабильной и не занимала системные пор-

ты серверной машины. С учетом использования стандарта WSGI библиотекой flask на базе которой работает локальный веб сервер приложения, необходимо было настроить автозапуск самого приложения при старте системы и отслеживания работы приложения. Для этого была использована стандартная утилита linux называемая supervisor. Первый конфигурационный файл имеющий название gunicorn.conf отвечает за старт приложения, содержание данного файла приведено на рисунке 24 Второй конфигурационный файл отвечает за работу самой утилиты supervisor и переадресацию данных от nginx сервера в написанное серверное приложение. Содержание второго файла приведено на рисунке 25.

```
1 [program:gunicorn]
2 command=/opt/venvs/informer_bot/bin/gunicorn wsgi:app -b 127.0.0.1:5000 -w 1 --
3 directory=/opt/projects/rsvpu_informer_bot/
4 user=antonpaly
5 redirect_stderr=True
6 stdout_logfile=/tmp/gunicorn.log
7 stderr_logfile=/tmp/gunicorn_err.log
8 autostart=true
9 autorestart=true
10 startsecs=10
11 stopwaitsecs=10
12 priority=999
13
```

Рисунок 24 — Содержание gunicorn.conf

```
1 [unix_http_server]
2 file=/tmp/telgram_bot_supervisord.sock
3 [supervisord]
4 logfile=/tmp/telgram_bot_supervisord.log
5 logfile_maxbytes=50MB
6 logfile_backups=10
7 loglevel=debug
8 pidfile=/tmp/telgram_bot_supervisord.pid
9 nodaemon=false
10 minfds=1024
11 minprocs=200
12 [rpcinterface:supervisor]
13 supervisor.rpcinterface_factory = supervisor.rpcinterface:make_main_rpcinterface
14 [supervisorctl]
15 serverurl=unix:///tmp/telgram_bot_supervisord.sock
16 [include]
17 files = /opt/projects/rsvpu_informer_bot/gunicorn.conf
18
```

Рисунок 25 — Содержание supervisor.conf

2.6 Установка и настройка бота для использования в мобильном приложении

При переходе по ссылке на бота будет выполнена команда «/start», которая занесет пользователя в базу данных. Дальше пользователь может рабо-

тать с ботом по своему усмотрению. Пример начала работы с ботом изображен на рисунке 36.

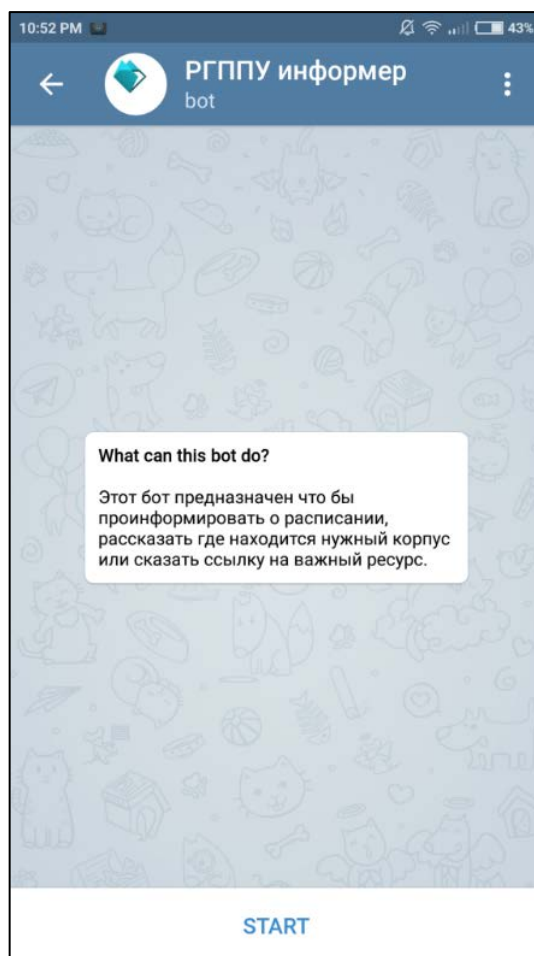


Рисунок 26 — Приветственное сообщение

В приветственном сообщении пользователю приходит сообщение, к которому прикреплена ссылка, в которой описаны основные функции которые есть в боте, пример открытой ссылки внутри telegram изображен на рисунке 28 Для оформления подписки на необходимое расписание, нужно выбрать элемент меню «Подписаться на расписание».

Пример меню бота после нажатия кнопки «START» изображен на рисунке 27. Дальше бот предложит выбрать тип расписания, группу или преподавателя. Последним шагом будет ввод данных для подписки, если будет много совпадений с полями базы данных, бот предложит пользователю выбор, рисунок 30.

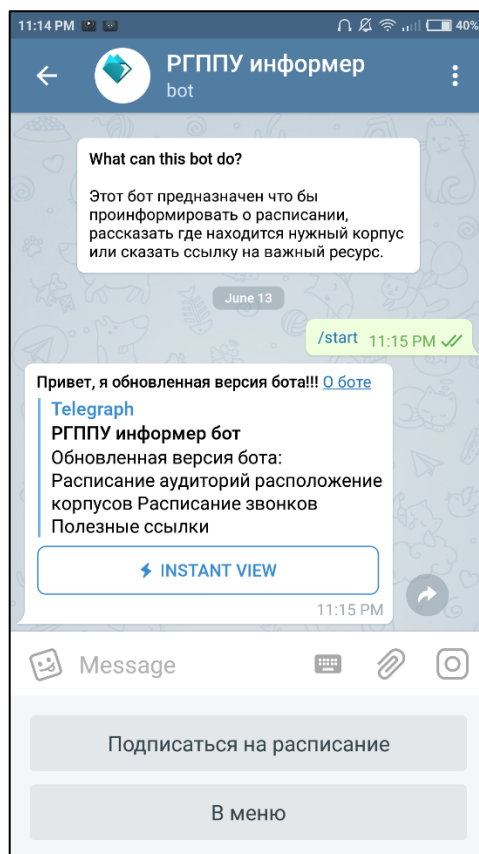


Рисунок 27 — Ссылка на Instant view с описанием возможностей бота

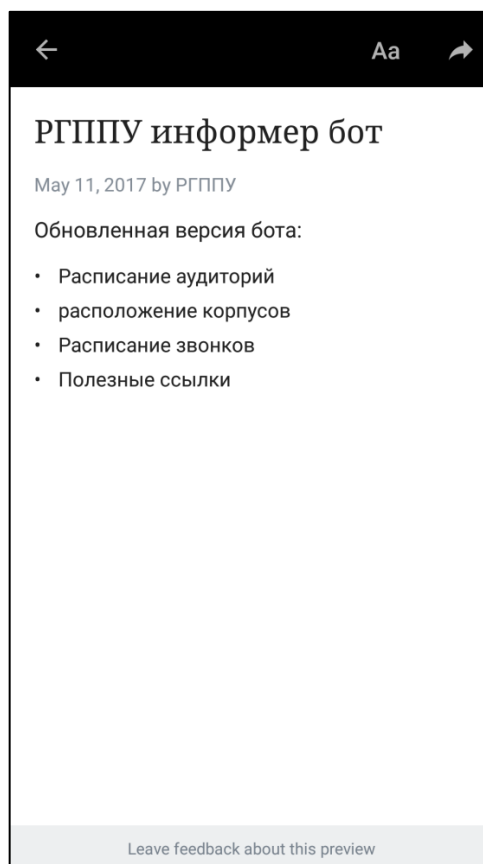


Рисунок 28 — Instant view открытый с телефона

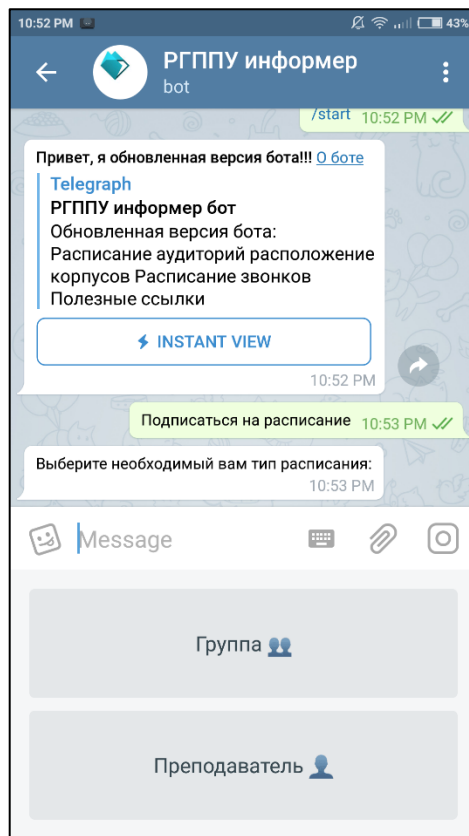


Рисунок 29 — Выбор типа подписки

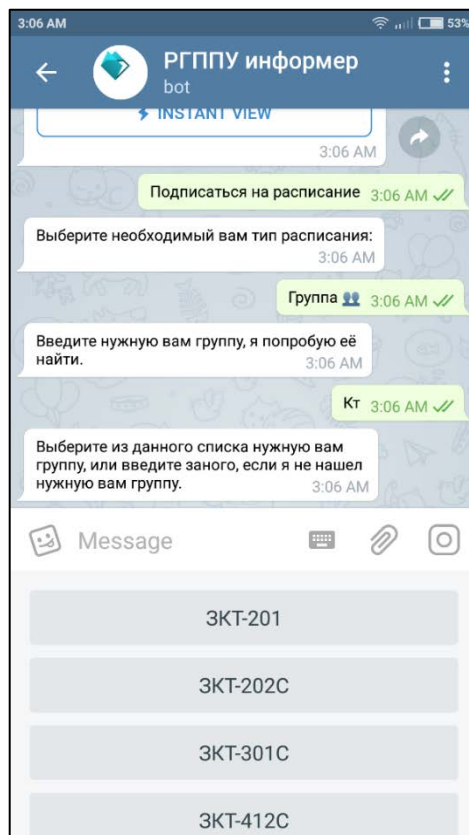


Рисунок 30 — Реализация поиска

После выполнения подписки пользователю открывается основное меню, по которому он может переходить, нажимая на кнопки, которые отображаются у него вместо клавиатуры. Пример получения расписания на неделю по оформленной подписке приведен на рисунке 31. Все элементы меню предоставленные пользователю функционируют и выполняют действия, в соответствии с их названиями.

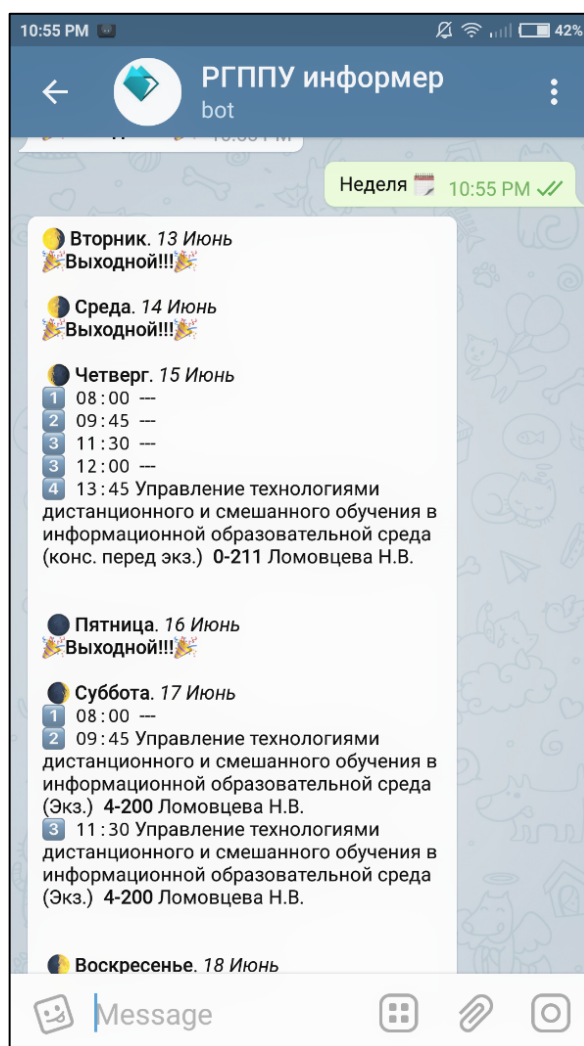


Рисунок 31 — Пример получения расписания на неделю

При нажатии на кнопку «Расписание звонков», пользователю предлагается выбрать расписание для каких курсов пользователь желает получить расписание. Пример описанного меню изображен на рисунке 32. В зависимости от того какой элемент выберет пользователь, бот отправит расписание для выбранных курсов. Примеры расписания звонков приведены на рисунке 33.

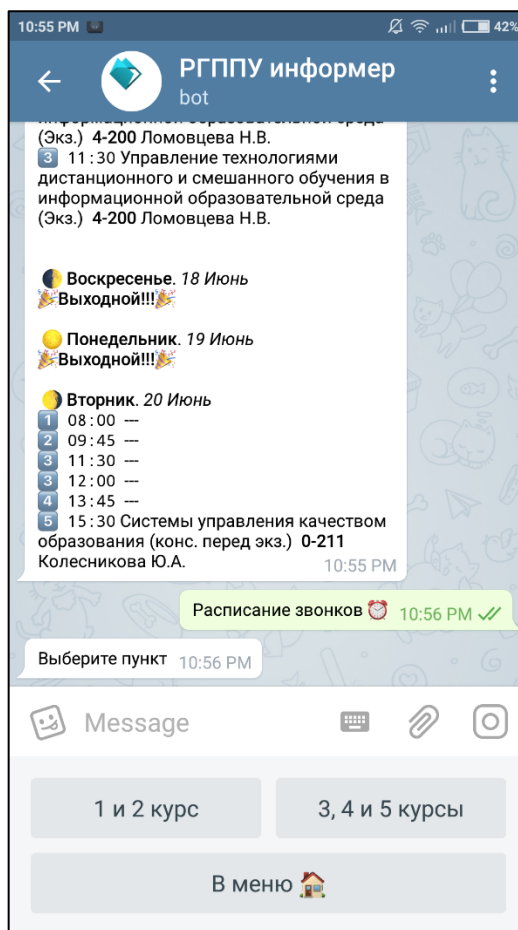


Рисунок 32 — Выбор расписания звонков

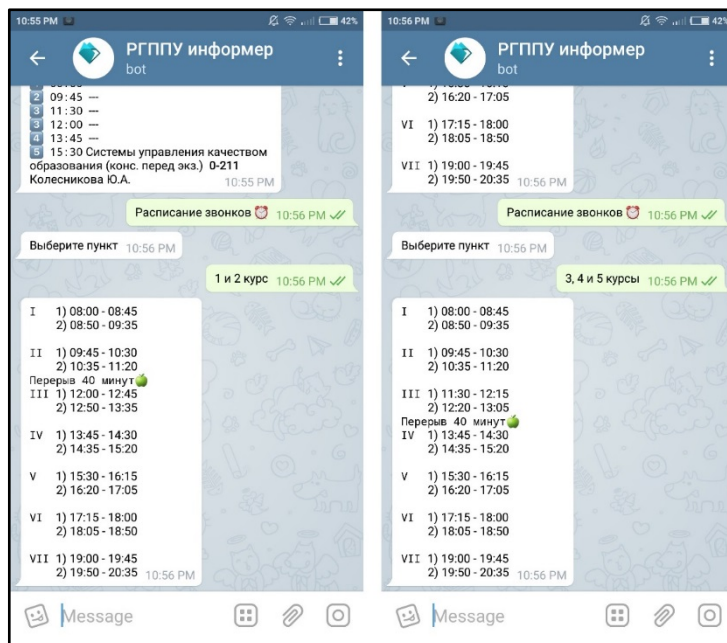


Рисунок 33 — Примеры расписания звонков

В главном меню есть пункт «Важные ссылки» в котором, пользователю отправляется сообщение, внутри которого находятся гиперссылки с подписями, на какой из ресурсов учебного заведения они ведут. На рисунке 34

изображено сообщение, которое пользователь получит при использовании данного элемента меню.

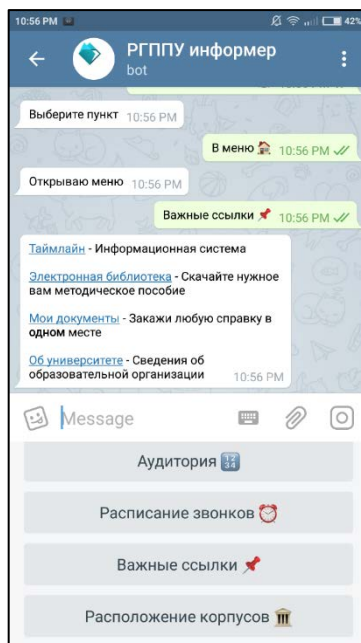


Рисунок 34 — Важные ссылки

Пункт меню «Расположение корпусов» позволяет пользователю узнать о расположении того или иного корпуса, пользователь может получить данные о нескольких корпусах и затем перейти в меню, с помощью кнопки расположенной в самом низу списка. Пример использования данной команды приведен на рисунке 35.

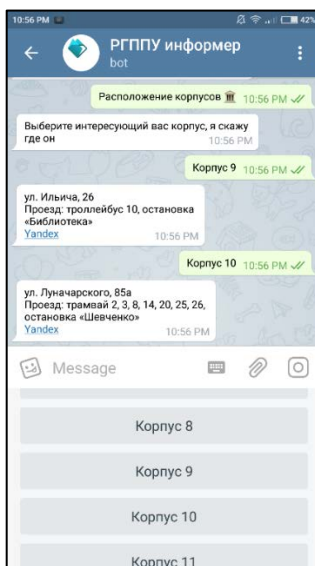


Рисунок 35 — Расположение корпусов

Для изменения настроек подписки, пользователь может перейти в раздел «настройки» изображенном на рисунке 36. В данном разделе выбрать

«подписка на расписание» и там следуя шагам, которые были выполнены пользователем при оформлении подписки в первый раз, настраивается расписание. Пример изменения подписки пошагово изображен на рисунках 37 38 и 39 вместе с отображением работы поиска.

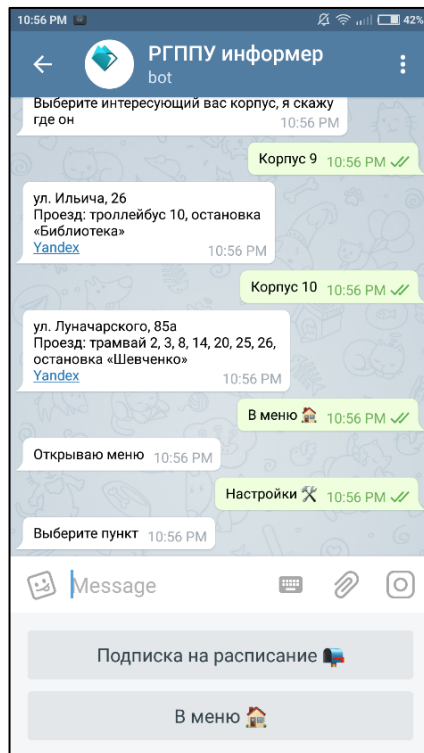


Рисунок 36 — Меню настроек

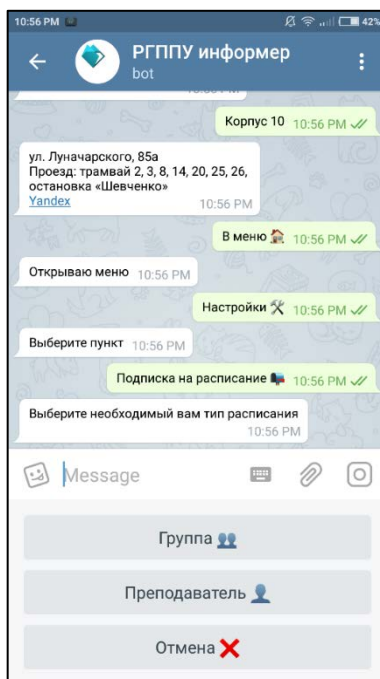


Рисунок 37 — Меню изменения подписки на расписание

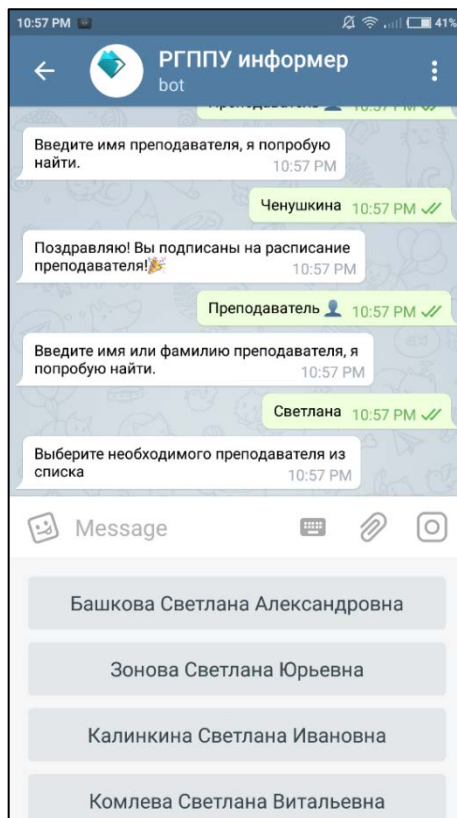


Рисунок 38 — Пример поиска преподавателей внутри бота

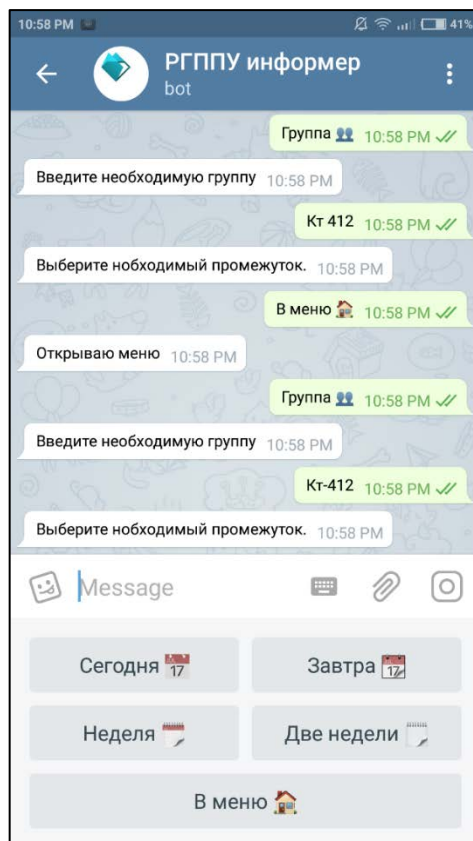


Рисунок 39 — Поиск групп внутри бота

2.7 Возможность просмотра в десктопной версии Telegram

Так как telegram разрабатывался под управлением одного человека, версии различных клиентов имеют минимальные различия, таким образом работа с серверным приложением внутри телефонной версии и версии на настольном компьютере не имеют отличий.

Для работы с ботом на компьютере в Windows, необходимо скачать и установить приложение Telegram Desktop, если до этого не было работы с ботом, его можно найти, введя в поиск @rsvpu_bot и выбрать «РГППУ информер». В случае если работа с ботом была до этого, достаточно найти в истории сообщений переписку с ботом и продолжить работу.

На рисунке 40 приведен пример работы с данным ботом на операционной системе Windows 7.

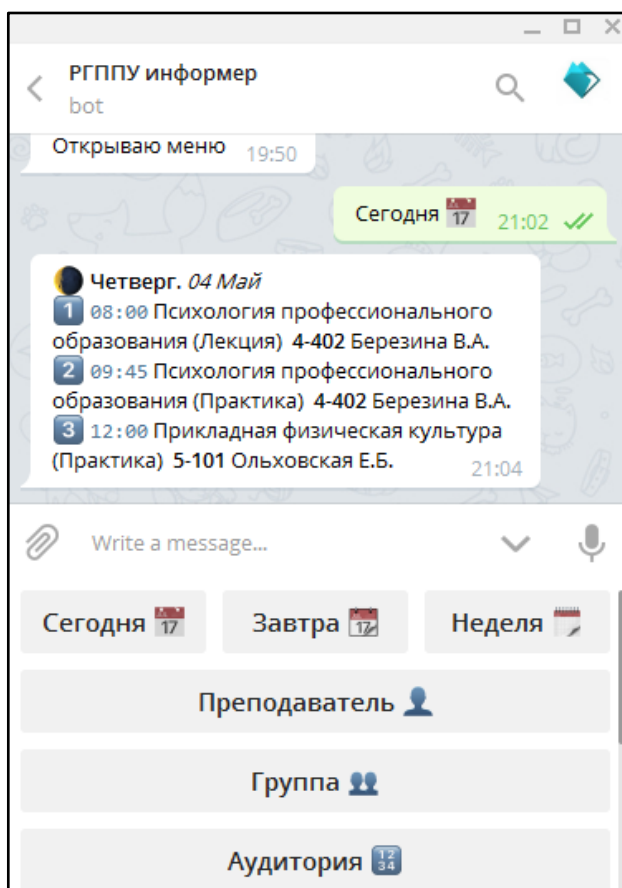


Рисунок 40 — Получение расписание на десктопной версии

2.8 Размещение сервиса на сервер Российского государственного профессионально-педагогического университета

Для размещения сервиса отделом администрирования компьютерной сети университета был подготовлен сервер Ubuntu Server 16.04 LTS

После чего он был конфигурационный файл nginx был настроен таким образом, чтобы веб сервер работал по протоколу HTTPS. Данная конфигурация отображена на рисунке 41. Данная конфигурация пересылает все получаемые данные на внутренний 5000 порт сервера, который прослушивается серверным приложением, которое управляет ботом.

```
server {
    listen 80;
    listen 443 ssl;

    server_name = tg.rsvpu.ru www.tg.rsvpu.ru;

    ssl_certificate /etc/letsencrypt/live/tg.rsvpu.ru/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/tg.rsvpu.ru/privkey.pem;

    location / {
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Scheme $scheme;
        proxy_pass http://localhost:5000/;
    }
}
```

Рисунок 41 — Конфигурация nginx

Так же отделом администрирования компьютерных сетей был выделен поддомен tg.rsvpu.ru от домена rsvpu.ru, на который был получен сертификат безопасности, для реализации HTTPS подключения. Сведения о подключении показываю, что с настройками nginx данное подключение стало безопасным (рисунок 42).

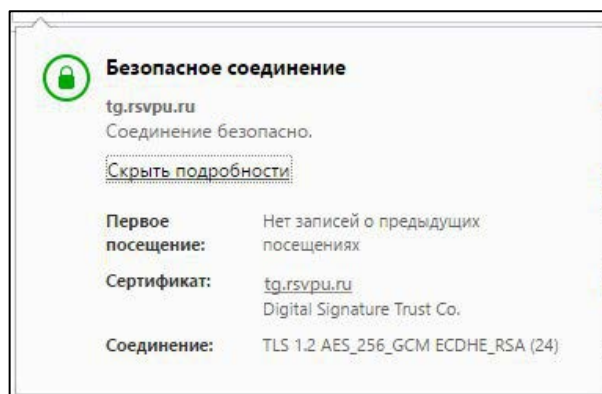


Рисунок 42 — Сведения о подключении к домену tg.rsvpu.ru

Данное серверное приложение было апробировано на студентах группы КТ-412 и некоторых преподавателях, было получено много положительных отзывов.

2.9 Описание сопроводительной документации

Был подготовлен документ, в котором даны подробные шаги по настройке модулей, изменение которых необходимо с течением времени, база данных расписания обновляется раз в полгода или возможно поступят предложения по изменению ответов бота или добавлению нового корпуса. Часть документа изображена на рисунке 43.

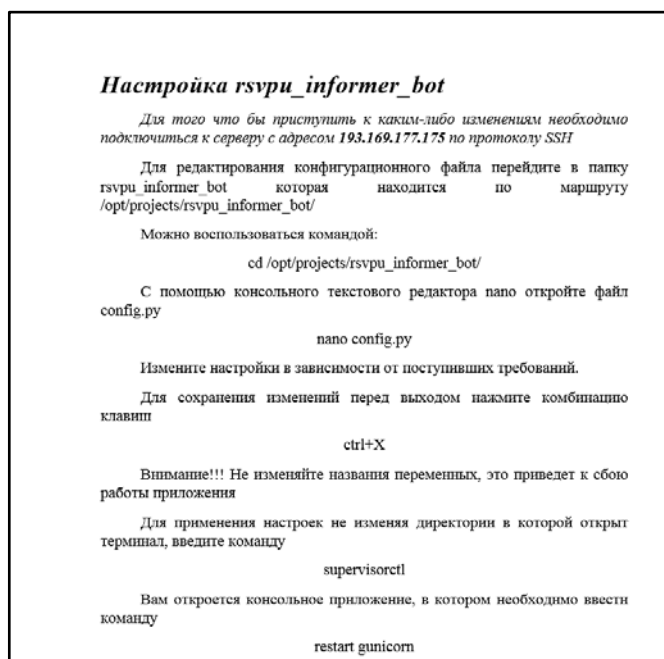


Рисунок 43 — Сопроводительная документация бота

ЗАКЛЮЧЕНИЕ

В рамках данной работы было разработано серверное приложение, состоящее из 3 модулей. Основной модуль способен работать независимо от остальных, тем самым, разработанный продукт можно распространять в рамках других учебных заведений от начального образования до высшего, путем замены модуля получения расписания РГППУ на модуль получения расписания иного учебного заведения и подгонку расписания под входные данные основного модуля.

В результате работы были разработаны:

1. Telegram-бот и проведена его настройка.
2. Архитектура серверного приложения.
3. Интерфейс для работы пользователя в Telegram.
4. Модель базы данных.
5. Модуль получения расписания из базы данных РГППУ.
6. Основной модуль, обрабатывающий события на сервере.
7. Разворачивание сервера.

В результате проделанной работы были решены следующие задачи:

Проанализирована литература и интернет-источники по использованию современных средств информирования.

Проанализированы имеющиеся аналогичные проекты в различных образовательных учреждениях.

Подготовлен алгоритм работы и архитектура сервиса в рамках требований мессенджера.

Реализован сервис информирования в выбранной среде программирования.

Подготовлена сопроводительная учебную документацию по настройке и использования сервиса для администратора.

Задачи решены, цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Библиотека «PyGame» [Электронный ресурс]. – Режим доступа: <http://www.pygame.org/news.html> (дата обращения: 5.06.17).
2. Бизли Д. Python. Подробный справочник [Текст] / Д. Бизли. – пер. А. Киселева – СПб: Символ-Плюс, 2013. – 864 с.
3. Васенин Д. Н. Социальные сети как эффективный инструмент повышения имиджа вуза [Электронный ресурс] / Д. Н. Васенин // Молодежный научный форум: Общественные и экономические науки: электр. сб. ст. по материалам XXIII студ. междунар. заочной науч.-практ. конф. — М.: МЦНО. — 2015. — № 4(23). — Режим доступа: [https://nauchforum.ru/archive/MNF_social/4\(23\).pdf](https://nauchforum.ru/archive/MNF_social/4(23).pdf) (дата обращения: 8.05.17).
4. Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/%CF%EВ%Е0%F2%F4%EE%F0%EC%E5%F0> (дата обращения: 5.06.17).
5. Группа РГППУ [Электронный ресурс]. – Режим доступа: <https://vk.com/rsvpu> (дата обращения: 18.05.17).
6. Документация Flask [Электронный ресурс]. – Режим доступа: <http://flask.pocoo.org> (дата обращения: 18.05.17).
7. Дрейк Ф. Л. Учебник Python 3.2 [Электронный ресурс]. – Режим доступа: http://ru.wikibooks.org/wiki/Учебник_Python_3.1 (дата обращения: 05.06.17).
8. ИДО РУДН [Электронный ресурс]. – Режим доступа: <http://www.ido.rudn.ru/open/multimedia/gloss.htm> (дата обращения: 05.06.17).
9. Использование мобильных телефонов в разных странах [Электронный ресурс]. – Режим доступа: <http://www.cossa.ru/152/37433/> (дата обращения: 15.05.17).

10. Использование мобильных устройств [Электронный ресурс]. – Режим доступа: <http://www.wi-life.ru/stati/wi-fi/marketingovye-stati-2/mobile-devices-use-aruba-research-results> (дата обращения: 15.05.17).

11. Канал ЮУрГУ [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/channel/UCNPVbFncaCIur8t3O4RmxHA> (дата обращения: 18.05.17).

12. Кургузов В. А. Роль официального сайта образовательного учреждения в формировании информационно-управленческой среды лица [Электронный ресурс]. – Режим доступа: <http://nsportal.ru/shkola/administrirovaniye-shkoly/library/rol-ofitsialnogo-saita-obrazovatel'nogo-uchrezhdeniya-v-formi> (дата обращения: 05.02.17).

13. Курс «Child Nutrition and Cooking» [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/learn/childnutrition> (дата обращения: 05.06.17).

14. Курс «Разработка компьютерных игр с помощью Python и Pygame» [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/3730/972/info> (дата обращения: 05.06.17).

15. Курс «Разработка компьютерных игр с помощью Python и Pygame» [Электронный ресурс]. – Режим доступа: http://www.intuit.ru/studies/professional_skill_improvements/11206/info (дата обращения: 05.06.17).

16. Лицензия GNU GPL [Электронный ресурс]. – Режим доступа: <http://www.fsf.org/news/agplv3-pr> (дата обращения: 05.06.17).

17. Лутц М. Изучаем Python [Текст] / М. Лутц. – СПб. Символ-Плюс, 2013. – 1280 с.

18. Мобильное приложение, что такое, определение, новости, статьи, видео [Электронный ресурс]. – Режим доступа: <https://indicator.ru/tags/mobilnoe-prilozhenie/> (дата обращения: 15.05.17).

19. Национальный открытый университет ИНТУИТ [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/> (дата обращения: 05.06.17).

20. О YouTube [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/YouTube> (дата обращения: 04.06.17).
21. Проект «Coursera» [Электронный ресурс]. – Режим доступа: <https://www.coursera.org/> (дата обращения: 05.06.17).
22. Проект «Python» на Github [Электронный ресурс]. – Режим доступа: <https://github.com/python> (дата обращения: 05.06.17).
23. Проект edX на GitHub [Электронный ресурс]. – Режим доступа: <https://github.com/edx/edx-platform> (дата обращения: 05.06.17).
24. Рабочая программа дисциплины «Интеллектуальные информационные системы» [Текст] / И. А. Сулова – Екатеринбург: ФГАОУ ВПО РГПШУ, 2011. – 11 с.
25. Расписание занятий РГПШУ [Электронный ресурс]. – Режим доступа: <http://www.rsvpu.ru/raspisanie-zanyatij-ochное-otdelenie/> (дата обращения: 09.02.17).
26. Сайт СПбГУ [Электронный ресурс]. – Режим доступа: <http://spbu.ru/news-spsu/18926-spbgu-i-coursera-inc-dogovarivayutsya-o-sotrudnichestve> (дата обращения: 05.06.17).
27. Сайт языка программирования «Python» [Электронный ресурс]. – Режим доступа: <http://python.org> (дата обращения: 05.06.17).
28. СПбГТИ [Электронный ресурс]. – Режим доступа: <http://technology.edu.ru/studentu/schedule.html> (дата обращения: 18.05.17).
29. Сузи Р. Язык программирования Python: [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/49/49/info> (дата обращения: 05.06.17).
30. Beautiful Soup Documentation [Электронный ресурс]. – Режим доступа: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (дата обращения: 14.02.17).
31. Bettex Morgan «MIT undergraduate engineering again ranked No. 1» [Электронный ресурс]. – Режим доступа: <http://newsoffice.mit.edu/20-10/undergraduate-rankings> (дата обращения: 05.06.17).

32. BotFather [Электронный ресурс]. – Режим доступа: <https://t.me/botfather> (дата обращения: 09.02.17).
33. Brody Hartley «Google’s Python Lessons are Awesome» [Электронный ресурс]. – Режим доступа: <http://blog.hartleybrody.com/google-python/> (дата обращения: 05.06.17).
34. Codecademy [Электронный ресурс]. – Режим доступа: codecademy.com (дата обращения: 05.06.17).
35. Coursera «Специальности на Coursera» [Электронный ресурс]. – Режим доступа: www.coursera.org/specializations (дата обращения: 05.06.17).
36. Coursera Blog «Introducing Coursera Learning Hubs: Global Participation, Local Access and Support!» [Электронный ресурс]. – Режим доступа: <http://blog.coursera.org/post/65596539008/introducing-coursera-learning-hubs-global> (дата обращения: 05.06.17).
37. HeadHunter [Электронный ресурс]. – Режим доступа: <https://hh.ru/> (дата обращения: 09.02.17).
38. IBM [Электронный ресурс]. – Режим доступа: <http://www.ibm.com/ru/ru/> (дата обращения: 05.06.17).
39. Lulu [Электронный ресурс]. – Режим доступа: <http://www.lulu.com/shop/swaroop-c-h/a-byte-of-python/paperback/product-211429-68.html> (дата обращения: 05.06.17).
40. Moodle [Электронный ресурс]. – Режим доступа: <https://moodle.org/> (дата обращения: 05.06.17).
41. Mssql server [Электронный ресурс]. – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2016> (дата обращения: 04.06.17).
42. NASA Deep Space Network [Электронный ресурс]. – Режим доступа: <http://deepspace.jpl.nasa.gov/dsn/> (дата обращения: 05.06.17).
43. NoStickerBot [Электронный ресурс]. – Режим доступа: https://t.me/-ru_ubuntu (дата обращения: 09.02.17).
44. Openssl [Электронный ресурс]. – Режим доступа: <https://www.openssl.org> (дата обращения: 18.05.17).

45. Peewee [Электронный ресурс]. – Режим доступа: <http://docs.peewee-orm.com/en/latest/> (дата обращения: 12.02.17).
46. Pilgrim M. Dive into Python 3 [Текст] / М. Pilgrim – NY: Apress, 2013 – 360 с.
47. Plaat Aske «MTD(f) algorithm» [Электронный ресурс]. – Режим доступа: <https://askeplaat.wordpress.com/534-2/mtdf-algorithm/> (дата обращения: 05.06.17).
48. Pymssql [Электронный ресурс]. – Режим доступа: <http://pymssql.org/-/en/stable/> (дата обращения: 18.05.17).
49. PyTelegramBotAPI [Электронный ресурс]. – Режим доступа: <https://git-hub.com/eternnoir/pyTelegramBotAPI> (дата обращения: 23.03.17).
50. Python [Электронный ресурс]. – Режим доступа: <https://www.python.org/> (дата обращения: 08.02.17).
51. Python for scientists [Электронный ресурс]. – Режим доступа: <https://sites.google.com/site/pythonforscientists/python-vs-matlab> (дата обращения: 5.06.17).
52. PythonWorkBot [Электронный ресурс]. – Режим доступа: https://t.me/ru_pythonjobs (дата обращения: 09.02.17).
53. Reddit [Электронный ресурс]. – Режим доступа: <http://www.reddit.com/> (дата обращения: 05.06.17).
54. Rosalind [Электронный ресурс]. – Режим доступа: <http://rosalind.info/problems/locations/> (дата обращения: 05.06.17).
55. Round Catherine «The Best MOOC Provider: A Review of Coursera, Udacity and edX» [Электронный ресурс]. – Режим доступа: <http://www.skilledup.com/blog/the-best-mooc-provider-a-review-of-coursera-udacity-and-edx> (дата обращения: 05.06.17).
56. RSVPUScheduleBot [Электронный ресурс]. – Режим доступа: https://telegram.me/rsvpu_bot (дата обращения: 15.03.17).

57. Sweigart A. «Making Games with Python & Pygame» [Электронный ресурс]. – Режим доступа: <http://inventwithpython.com/makinggames.pdf> (дата обращения: 05.06.17).

58. Telegram (мессенджер) [Электронный ресурс]. – Режим доступа - [https://ru.wikipedia.org/wiki/Telegram_\(мессенджер\)](https://ru.wikipedia.org/wiki/Telegram_(мессенджер)) (дата обращения: 04.05.17).

59. Wombat.org.ua [Электронный ресурс]. – Режим доступа: <http://wombat.org.ua/AByteOfPython/#id22> (дата обращения: 05.06.17).

60. Yandex-бот [Электронный ресурс]. – Режим доступа: <https://t.me/ya> (дата обращения: 11.02.17).

61. Youtube [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/> (дата обращения: 05.06.17).

ПРИЛОЖЕНИЕ

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
направление 44.03.04 Профессиональное обучение (по отраслям)
профиль «Информатика и вычислительная техника»
профилизация «Компьютерные технологии»

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Н. С. Толстова

« _____ » _____ 2017 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

студента 4 курса курса, группы КТ-412 Паляя Антона Игоревича

1. Тема Сервис информирования участников учебного процесса в мессенджере Telegram утверждена распоряжением по институту от 07.02.2017 г. № 73.

2. Руководитель С.В. Ченушкина, ст. преподаватель каф. ИС

3. Место преддипломной практики кафедра ИС РГППУ

4. Исходные данные к ВКР Бизли Д. «Python. Подробный справочник», Дрейк Ф. Л. «Учебник Python 3.2», Лутц М. «Изучаем Python», Сузи Р. «Язык программирования Python», Пилгрим М. «Dive into Python 3».

5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)

Проанализировать литературу и интернет-источники по использованию современных средств информирования.

Проанализировать имеющиеся аналогичные проекты в различных образовательных учреждениях.

Подготовить алгоритм работы и архитектуру сервиса в рамках требований мессенджера.

Реализовать сервис информирования в выбранной среде программирования.

Подготовить сопроводительную учебную документацию по настройке и использованию сервиса для администратора

6. Перечень демонстрационных материалов

презентация, выполненная средствами Microsoft PowerPoint

бТ7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа дипломной работы	Срок выполнения этапа	Процент выполнения ВКР	Отметка руководителя о выполнении
-------	-------------------------------------	-----------------------	------------------------	-----------------------------------

