|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по итоговому проекту**

по дисциплине «Технологии разработки программных приложений»

**Тема практической работы:** «Разработка бот-сервиса»

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИНБО-05-19 | Паламарчук Я.В. |
| **Проверил:** | Торхов А.Е. |

2021 г.

**СОДЕРЖАНИЕ**

[Часть 1. О проекте 3](#_Toc75451671)

[Часть 2. Макет и интерфейс бота 4](#_Toc75451672)

[Часть 3. Разработка проекта 6](#_Toc75451673)

[Часть 3. Развертывание c помощью Docker 11](#_Toc75451674)

[Ссылка на репозиторий с проектом: 11](#_Toc75451675)

**Часть 1. О проекте**

В жизни студента важную роль играет грамотное планированное своих дел, а также своевременное получение актуальной учебной информации. И для оптимизации данных задач мы решили разработать ассистент-бота в популярном мессенджере Telegram, который будет помогать студентам РТУ МИРЭА в учебном процессе

*Ассистент РТУ МИРЭА - Telegram bot с полезной и актуальной учебной информацией для студентов.*

**User Story**

Как студент, я хотел бы иметь возможность получать информацию о расписании своей группы как на сегодняшний день, так и на неделю в целом. Расписание должно быть всегда актуальным, чтобы не попадать в неудобные ситуации, а при обновлении расписания группы получать уведомление от бота.

**Начальный список функций:**

- выбор группы обучающегося;

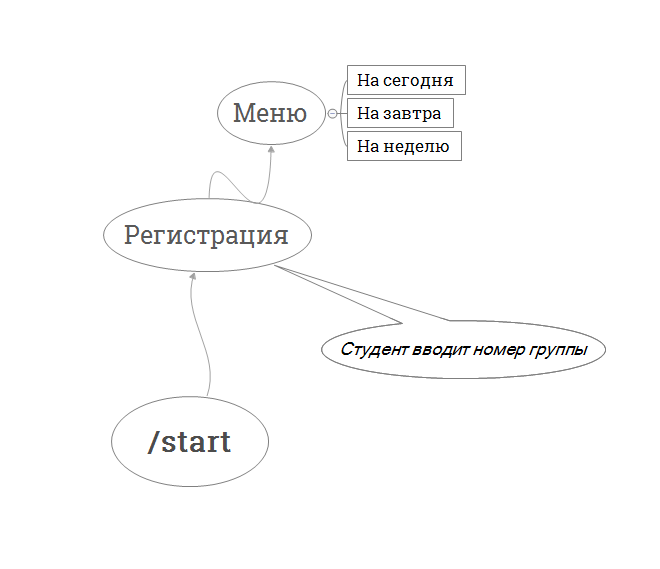
- вывод текущего расписания на сегодня, завтра и на неделю;

- уведомление об обновлении расписания.

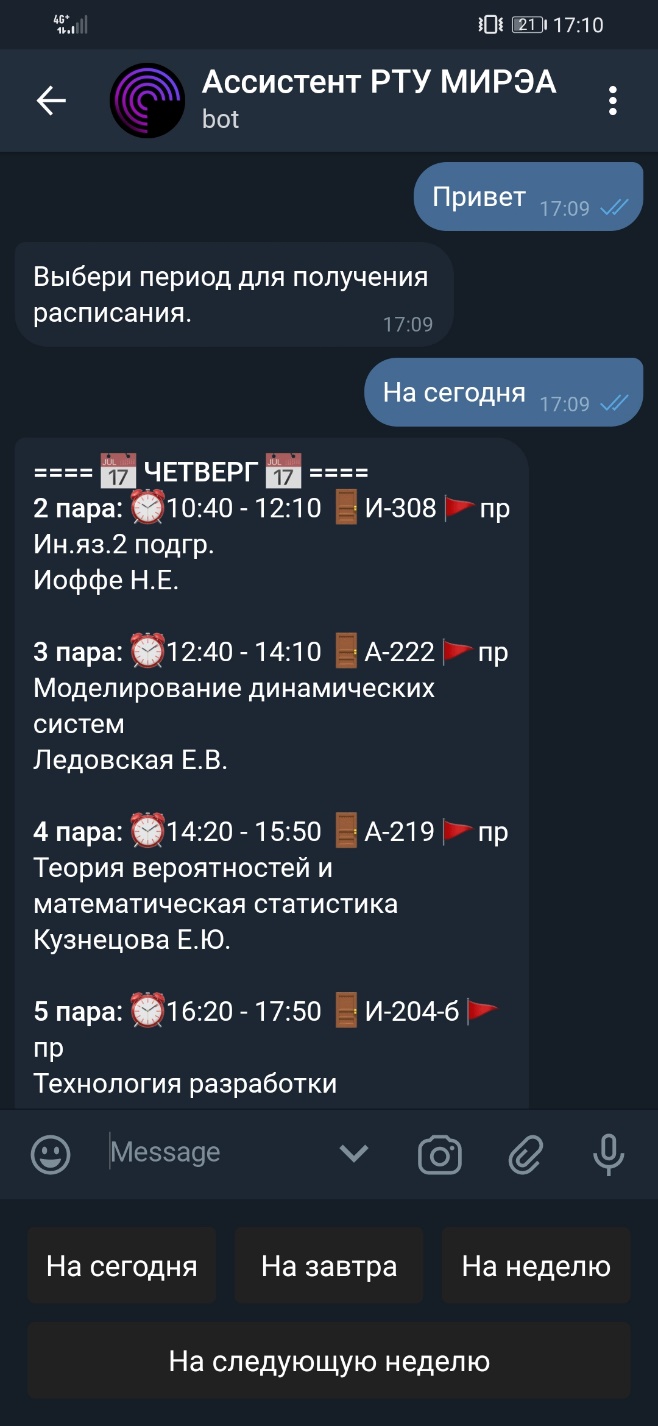
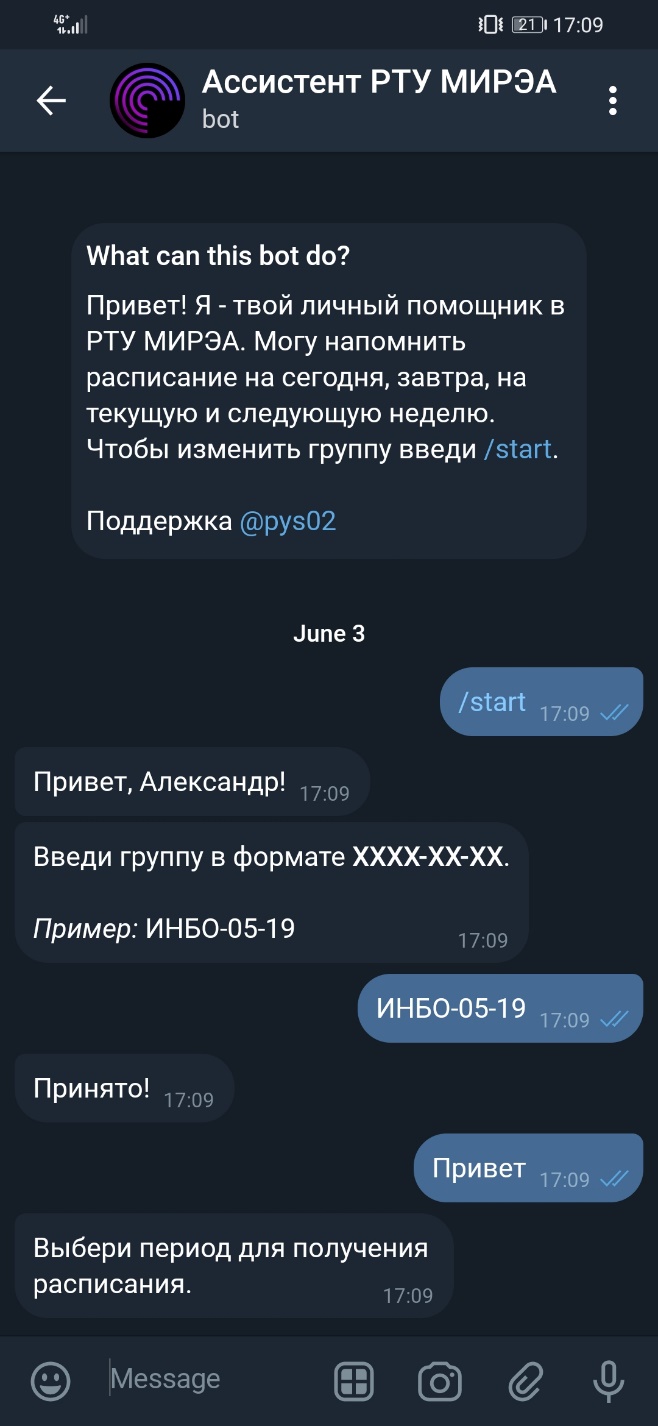
**Часть 2. Макет и интерфейс бота**

Бот работает через мессенджер Telegram, в котором управление ассистентом реализовывается как через ввод заготовленных команд, так и через нажатие всплывающих кнопок. Второй вариант является более понятным для пользователя, поэтому он будет предпочтительнее при реализации нашего интерфейса.

*Диаграмма состояний бота*

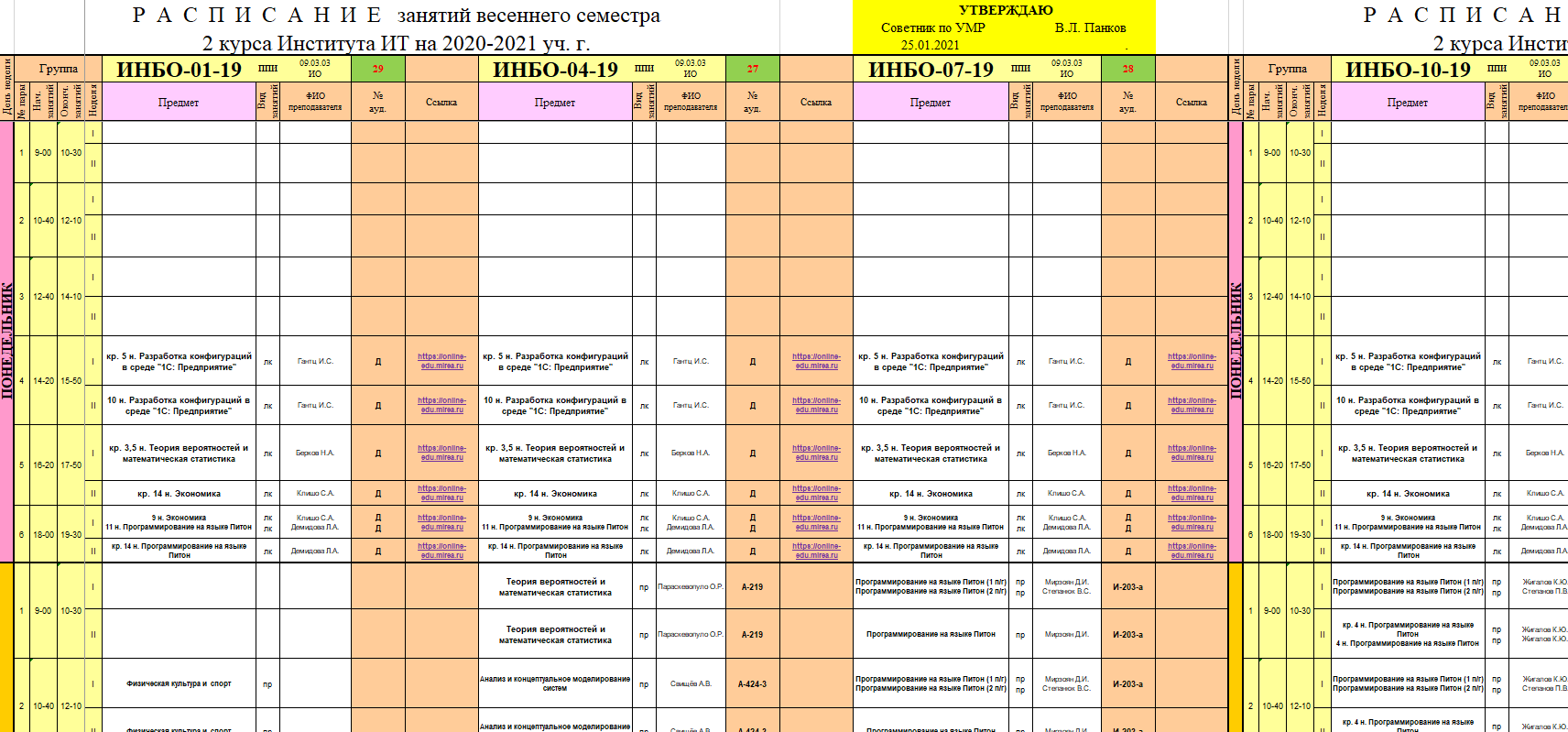


*Интерфейс бота*



**Часть 3. Разработка проекта**

Для ознакомления студентов на сайте РТУ МИРЭА расписание представлено множеством xlsx файлов следующего вида:



Пример файла расписания

Такие файлы весят очень много, в них неудобно ориентироваться, а их загрузка занимает продолжительное время ввиду сложности разметки. Для получения данных в удобном для работы виде нам необходимо проделать следующие действия:

1. Проанализировать сайт, извлечь ссылки на файлы с расписанием.
2. Скачать таблицы с данными.
3. Извлечь из файлов данные, поместить их в хранилище для бота.

Алгоритм работы бота представлен ниже:

Приветствие пользователя

Запрос группы

Отчет о наличии расписания для выбранной группы

Ожидание запроса

Отправить расписание на сегодня

Отправить расписание на текущую неделю

Отправить расписание на завтра

Отправить расписание на следующую неделю

На неделю

На сегодня

На завтра

На следующую неделю

Алгоритм работы бота

Для извлечения ссылок с сайта МИРЭА и предобработки данных из xlsx таблиц воспользуемся готовым инструментарием языка Python, а непосредственно приложение чат-бота реализуем средствами языка R. Среда разработки RStudio представляет обширный функционал для разработки на языках R и Python, а также позволяет с легкостью осуществлять их взаимную интеграцию. В качестве буфера памяти для хранения данных о расписании для бота будем использовать встраиваемую СУБД SQLite.

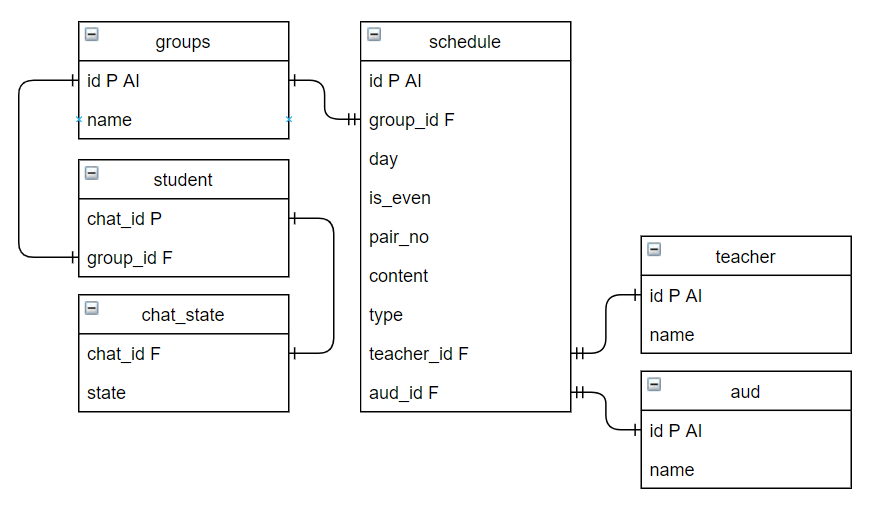
Для анализа сайта воспользуемся библиотекой BeautifulSoup4 языка Python. Выполним поиск по странице всех ссылок на файлы, оканчивающиеся расширением «.xlsx». Получим список ссылок на файлы. Затем, с помощью модуля requests скачаем интересующие нас таблицы. После этого извлечем данные из скачанных таблиц в JSON-файл, что облегчит последующую загрузку данных в СУБД.

Извлечение данных происходит следующим образом. Для удобной работы с .xlsx файлами подключим библиотеку openpyxl. Затем пройдемся по второй строке каждой таблицы, пока не найдем соответствие регулярному выражению:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Если найдено соответствие, то значит, что мы находимся в ряду с расписанием. Следовательно, в 72 ячейках ниже (6 дней недели, 6 пар, 2 варианта на четную и нечетную недели) содержится информация о занятиях, в столбце правее – тип занятия (практика или лекция), двумя столбцами правее – ФИО преподавателя, тремя – номер аудитории.

Создадим встраиваемую базу данных «bot.db» под управлением SQLite. Структура этой базы представлена ниже:



*Структура базы данных bot.db*

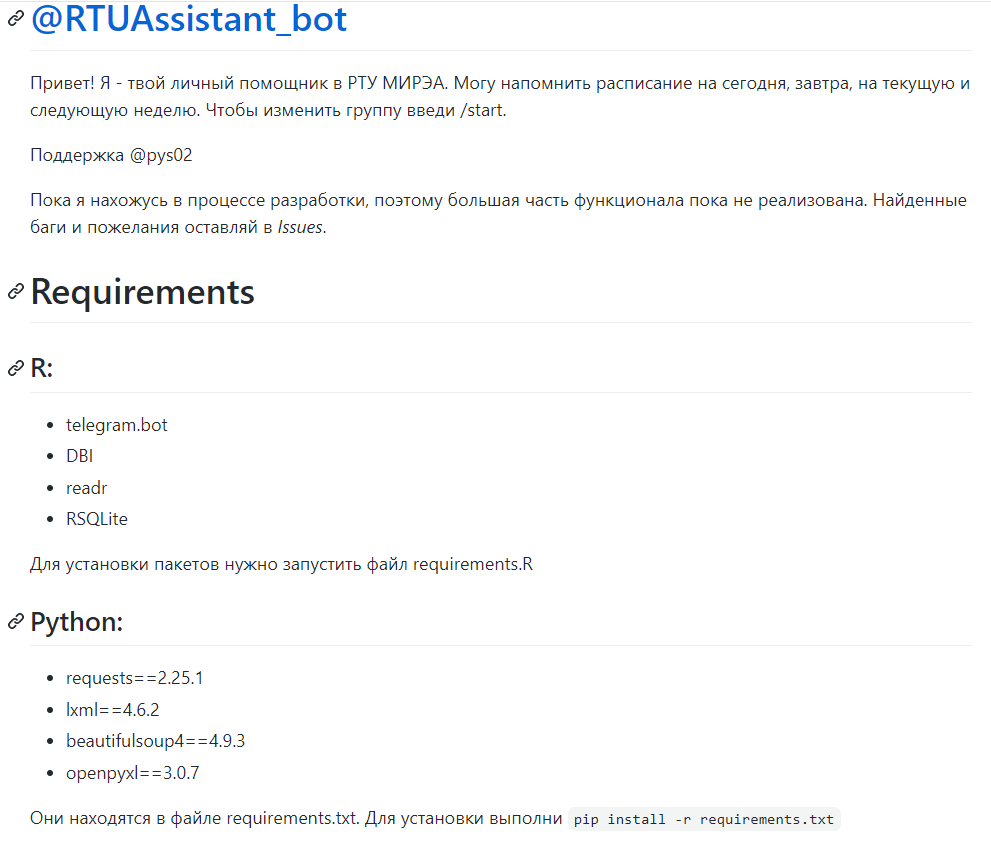
Заполним созданную базу данных данными из JSON файла средствами языка R. Для этого подключим библиотеки DBI и RSQLite. Добавим в таблицы groups, teacher и aud соответствующую информацию из JSON файла с помощью метода dbExecute. Затем так же заполним таблицу schedule.

Предварительно получим token бота у @BotFather в Telegram, проведем предварительную настройку, придумаем название и добавим фотографию на аватар. После чего приступим к разработке.

Для разработки telegram-бота на языке R воспользуемся библиотекой telegram.bot. Бот является конечным автоматом, переключающимся между различными заранее заданными состояниями. Для удобства разработки создадим 5 файлов: bot\_methods.R, содержащий непосредственно функционал бота для общения с пользователем; msg\_filters.R, содержащий фильтры сообщений; handlers.R, содержащий перехватчики классов MessageHandler и CommandHandler, управляющие поведением бота; db\_methods.R, содержащий методы бота для обращения к базе данных; и короткий файл bot.R, в котором объявлены различные константы, подключены библиотеки и описанные выше файлы. В последнем файле инициализируется и запускается бот.

Так как наш бот пишется на двух языках программирования: R и Python, то необходимо проработать сборку для данных языков. Для R мы скачиваем необходимые библиотеки, находящиеся в файле requirements.R. Для Python же мы формируем файл requirements.txt со списком зависимостей и уже в дальнейшем подключаем setuptools для сборки пакета с проектом.

Также важной частью является написание файла README.md с общим описанием проекта, опишите зависимости проекта и команду для его запуска.

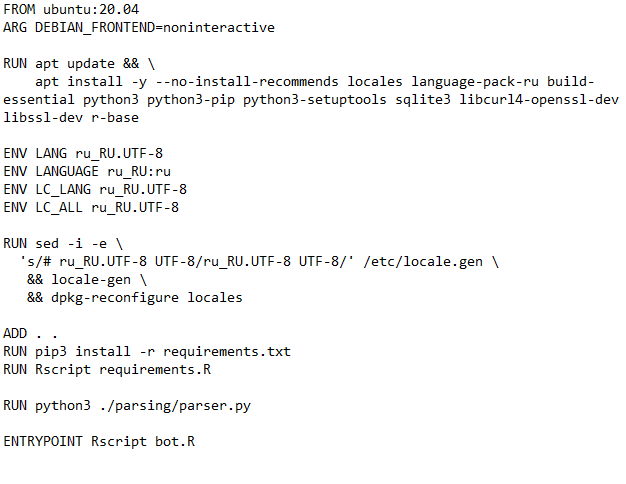


*Файл README.md*

Теперь, имея рабочую сборку нашего проекта, можем продолжать обновлять код и дополнять его новыми методами и функциями. При этом нельзя забывать и об использовании системы контроля версий git, которая также повышает удобство разработки.

**Часть 3. Развертывание c помощью Docker**

Для развертывания чат-бота была выбрана платформа Heroku, позволяющая бесплатно развертывать небольшие приложения на своих серверах. Для развертывания был создан следующий докерфайл:



*Листинг Dockerfile*

После написания докерфайла было развернуто приложение на платформе Heroku. Ссылка на приложение https://rtu-assistant-bot.herokuapp.com/

**Ссылка на репозиторий с проектом:**

https://github.com/palyaros02/RTUAssistant\_bot