**Техническое задание на разработку Чат-бота Telegram для конвертации расписания любой группы РГСУ в файл календаря**

Утверждено:

Генеральный директор

ООО «Исследовательская группа «Омнибус»,

Золотова С. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_

15 февраля 2022 год

Содержание

[1. Введение 4](#_Toc88922759)

[1.1. Наименование разработки 4](#_Toc88922760)

[1.2. Краткая характеристика разработки 4](#_Toc88922761)

[2. Основания для разработки 5](#_Toc88922762)

[2.1. Заказчик 5](#_Toc88922763)

[2.2. Исполнитель 5](#_Toc88922764)

[3. Назначение разработки 6](#_Toc88922765)

[3.1. Основные плюсы разработки 6](#_Toc88922766)

[3.2. Основные преимущества разработки 6](#_Toc88922767)

[4. Требования к ПО 7](#_Toc88922768)

[4.1. Пользовательский интерфейс 7](#_Toc88922769)

[4.2. Платформа разработки ПО 7](#_Toc88922770)

[4.3. Инструментальная часть разработки ПО 7](#_Toc88922771)

[5. Требования к программной документации 8](#_Toc88922772)

[5.1. Документация для пользователей 8](#_Toc88922773)

[5.2. Документация для администратора 8](#_Toc88922774)

[6. Технико-экономические показатели 9](#_Toc88922775)

[6.1. Обоснования использования ЯП 9](#_Toc88922776)

[6.2. Обоснования использования IDE 9](#_Toc88922777)

[6.3. Обоснования использования Redis 9](#_Toc88922778)

[6.4. Стоимость разработки 9](#_Toc88922779)

[7. Стадии разработки 10](#_Toc88922780)

[7.1. (Опционально) Нулевая стадия – Разработка шаблона модуля 10](#_Toc88922781)

[7.2. Первая стадия – Модуль парсера 10](#_Toc88922782)

[7.3. Вторая стадия – Модуль БД 10](#_Toc88922783)

[7.4. Третья стадия – Модуль календаря 11](#_Toc88922784)

[7.5. Четвертая стадия – Модуль рассылки почты 11](#_Toc88922785)

[7.6. Пятая стадия – Модуль обнаружения обновлений расписания 11](#_Toc88922786)

[7.7. Шестая стадия – Разработка модуля приложения 11](#_Toc88922787)

[7.8. Седьмая стадия – Разработка Чат-Бота Telegram 12](#_Toc88922788)

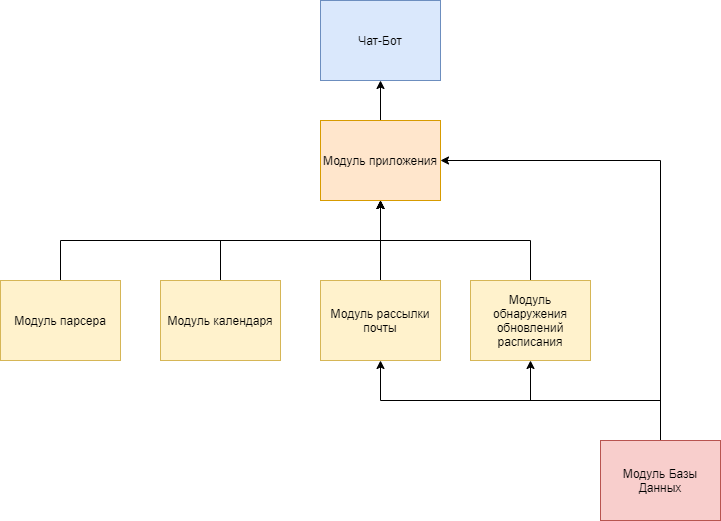
[7.9. Восьмая стадия – Реализация автоматического тестирования 12](#_Toc88922789)

[7.10. Девятая стадия – Тестирование на устройствах пользователей 12](#_Toc88922790)

[8. Тестирование программного комплекса 13](#_Toc88922791)

[8.1. Автоматическое тестирование ПО 13](#_Toc88922792)

[8.2. Тестирование на устройствах пользователей 13](#_Toc88922793)

1. Введение
   1. Наименование разработки
      1. Чат-бота Telegram для конвертации расписания любой группы РГСУ в файл календаря (далее, ПО)
   2. Краткая характеристика разработки
      1. Данное ПО предназначено для оптимизации работы студентов с изменениями расписания РГСУ и более удобной организации своего времени посредством использования календарей.
2. Основания для разработки
   1. Заказчик
      1. ПО разрабатывается для Студентов и Преподавателей РГСУ.
   2. Исполнитель
      1. Чайковский Никита Олегович.
3. Назначение разработки
   1. Основные плюсы разработки
      1. ПО позволяет сэкономить время при внесении расписания в различные календари в т.ч и Google Calendar.
   2. Основные преимущества разработки
      1. ПО спроектировано с помощью модульной архитектуры, что позволяет использовать ПО частично в других проектах.
      2. ПО следит за обновлениями календаря что бы студент был всегда в курсе новейшего расписания.
      3. Ведение календаря позволяет эффективно разгрузить мозг так как необходимость помнить такую информацию как: время, кабинет, предмет, время на перемену, и прочие данные доступные из расписания — пропадает.
4. Требования к ПО
   1. Пользовательский интерфейс
      1. Пользовательский интерфейс основан на Telegram Bot API
   2. Платформа разработки ПО
      1. Любая платформа позволяющая запустить Telegram, а именно: Web, Android, iOS, Windows Phone, PC, Linux
   3. Инструментальная часть разработки ПО
      1. Языки программирования (далее, ЯП): HTML, CSS, JavaScript, TypeScript 4.4.x
      2. Пакетный менеджер: pnpm
      3. Серверная часть: Node.js 16.13.x
      4. Инфраструктура: Docker, docker-compose
      5. База данных: Redis
      6. Выделенный сервер: DigitalOcean Droplet
      7. IDE: VS Code
      8. Unit-тестирование: Jest
      9. Подход к разработке: Моно-репозиторий с использованием Lerna
      10. Форматирование и линтинг: Eslint, Prettier
      11. Стабильность рабочего пространства: Volta
      12. CI/CD: Github Actions
      13. Сторонние библиотеки: cheerio, got, telegraf, ical-generator
5. Требования к программной документации
   1. Документация для пользователей
      * 1. Как начать работу с Чат-ботом?  
           Найдите при помощи поиска в Telegram бота @rsuu-timetable-bot
        2. Как скачать расписание?  
           Наберите /start, после бот предложит вам выбрать факультет и группу и предложит скачать файл расписание
        3. Как подписаться на обновления расписания?  
           После заполнения факультета и группы нажмите на кнопку «Подписаться»
        4. Как отписаться от обновлений расписания?  
           После заполнения факультета и группы нажмите на кнопку «Отписаться»
        5. Как сменить факультет или группу?  
           После заполнения факультета и группы нажмите на кнопку «Изменить»
   2. Документация для администратора
      1. Схема зависимости модулей:
      2. 

|  |  |  |
| --- | --- | --- |
| **API Модуля парсера** | | |
| **Функция** | **Параметры** | **Описание** |
| getPage: string | faculty?: number; group?: string | Получает страницу расписания РГСУ, опциональные параметры выставляют соответственные Query Para |
| getFaculties: {text: string; value: string} |  | Получает список доступных факультетов |
| getGroups: {text: string; value: string} | faculty: number | Получает список доступных групп факультета |
| getTimetable: Timetable | faculty: number; group: string | Получает расписание |



|  |  |  |
| --- | --- | --- |
| **API Модуля календаря** | | |
| **Функция** | **Параметры** | **Описание** |
| generateCalendar: boolean | timetable: Timetable; path: string | Генерирует файл календаря по указанному пути |



|  |  |  |
| --- | --- | --- |
| **API Модуля рассылки почты** | | |
| **Функция** | **Параметры** | **Описание** |
| sendMailTo: boolean | email: string; file: PathLike | Отправляет указанный файл на указанную почту |



|  |  |  |
| --- | --- | --- |
| **API Модуля обнаружения обновления расписания** | | |
| **Функция** | **Параметры** | **Описание** |
| detectTimetableChange: boolean | newTimetable: Timetable; oldTimetable: Timetable | Проверяет есть ли изменения в расписании |



|  |  |  |
| --- | --- | --- |
| **API Модуля БД** | | |
| **Функция** | **Параметры** | **Описание** |
| getUser: User | userId: string | Получает пользователя |
| setUser: User | userId: string; payload: User; | Создает пользователя |
| updateUser: User | userId: string; payload: Partial<User>; | Обновляет пользователя |
| deleteUser: User | userId: string | Удаляет пользователя |
| getCalendar: Calendar | faculty: number; group: string | Получает календарь |
| setCalendar: Calendar | faculty: number; group: string;  payload: Calendar | Создает календарь |
| updateCalendar: Calendar | faculty: number; group: string;  payload: Partial<Calendar> | Обновляет календарь |
| deleteCalendar: Calendar | faculty: number; group: string | Удаляет календарь |
| getMailingList: MailingList | faculty: number; group: string | Получить список рассылки |
| setMailingList: MailingList | faculty: number; group: string; payload: MailingList | Установить список рассылки |
| updateMailingList: MailingList | faculty: number; group: string; payload: Partial<MailingList> | Обновить список рассылки |
| removeMailingList: MailingList | faculty: number; group: string | Удалить список рассылки |
| removeMailingListItem: MailingList | faculty: number; group: string; payload: string | Удалить элемент из списка рассылки |
| setMailingListItem: MailingList | faculty: number; group: string;  payload: string | Добавить элемент в конец списка рассылки |



|  |  |  |
| --- | --- | --- |
| **API Модуля приложения** | | |
| **Функция** | **Параметры** | **Описание** |
| createUser: User | userId: string; payload: User | Создать пользователя |
| updateUser: User | userId: string; payload: Partial<User> | Обновить пользователя |
| deleteUser: User | userId: string | Удалить пользователя |
| getTimetable: Timetable | faculty: string; group: string | Получить расписание |
| getCalendar: Calendar | faculty: string; group: string | Получить календарь |
| createCalendar: Calendar | Faculty: string; group: string; payload: Calendar | Создать календарь |
| updateCalender: Calendar | faculty: string; group: string; payload: Partial<Calendar> | Обновить календарь |
| setupUpdateCalendarJobs: void |  | Запустить цикл проверки обновлений календаря |
| setupMailingListJobs: void |  | Запустить цикл рассылки почтовых обновлений |
| setupUpdateCacheJob: void |  | Запуск цикла догрузки и обновления кэша календарей |
| addMailingListItem: boolean | email: string | Добавить почту в список рассылки |
| removeMailingListItem: boolean | email: string | Удалить почту из списка рассылки |

1. Технико-экономические показатели
   1. Обоснования использования ЯП
      1. TypeScript в связке с Node.js предлагает быстрый и комфортный цикл разработки приложений.
   2. Обоснования использования IDE
      1. VS Code имеет первостепенную поддержку TypeScript, а это основная технология ПО.
   3. Обоснования использования Redis
      1. БД будет использована для хранения профилей пользователей. БД была выбрана за её простоту использования, скорость, и в связи с тем, что нам не нужен функционал полноценной БД.
   4. Стоимость разработки
      1. Общее ориентировочное время разработки: 178 часов
      2. Общая ориентировочная цена разработки: 70 000 руб.
      3. Поэтапное время разработки:
         1. Разработка шаблона модуля – 4 часа
         2. Модуль парсера – 12 часов
         3. Модуль БД – 20 часов
         4. Модуль календаря – 6 часов
         5. Модуль рассылки почты – 8 часов
         6. Модуль приложения – 40 часов
         7. Разработка Чат-Бота Telegram – 40 часов
         8. Реализация автоматического тестирования – 12 часов
         9. Тестирование на устройствах пользователей – 48 часов
2. Стадии разработки
   1. (Опционально) Нулевая стадия – Разработка шаблона модуля
      1. Необходимо создать общий шаблон модуля, использующий описанные выше технологии, стадия опциональная, но она сэкономит время.
   2. Первая стадия – Модуль парсера
      1. Разработка парсера сайта расписания занятий [РГСУ](https://rgsu.net/for-students/timetable/timetable.htm).
      2. Модуль должен реализовать следующие варианты использования (далее, ВИ):
         1. Получение списка факультетов, при этом текст вида «--выбрать» должен быть исключен.
         2. Получение списка групп на основе выбранного факультета
         3. Получения расписания, используя Query Params необходимо получить чётную и нечетную недели.
         4. Получение расписания текущего дня, дату, статус чётности или нечётности недели, время.
      3. Каждое расписание состоит из чётной и нечётной недели в каждой из которых находится список дней: Понедельник, Вторник, Среда, Четверг, Пятница, Суббота.
      4. Каждый день состоит из списка лекций, каждая лекция содержит: время начала, время окончания, аудитория, предмет, дни проведения предмета (если есть, иначе предполагается что предмет проводится каждый день), тип предмета (лекция, практическое занятие, лабораторная работа), преподаватель, регалии преподавателя (например: д.ф.н).
   3. Вторая стадия – Модуль БД
      1. Модуль БД предоставляет контейнер с необходимой конфигурацией Redis, а также необходимые абстракции для работы с данными.
      2. Модуль должен предоставлять уровень абстракции над базой данных, а также конфигурацию.
      3. Необходимо учесть следующие ВИ:
         1. Использования для хранения профилей пользователей. Ключом является Telegram User Id, а в качестве значения Хэш хранящий факультет, группу, и опциональный адрес почты.
         2. Использование для хранения кэша файлов календарей. Ключом является факультет и группа через знак «|», а в качестве значения Хэш хранящий ссылку на файл и дату последнего обновления
         3. Использование для хранения списка рассылки обновлений расписания. Ключом является факультет и группа через знак «|», а в качестве значения Множество, содержащее почтовые адреса отправки.
   4. Третья стадия – Модуль календаря
      1. Разработанный модуль должен генерировать календарь из данных расписания.

Календарь генерируется с помощью библиотеки ical-generator.

* 1. Четвертая стадия – Модуль рассылки почты

Задачей модуля является отправка почтового шаблона содержащего сообщение об обновлении с прикрепленным файлом календаря используя сервера яндекс.почты.

* 1. Пятая стадия – Модуль обнаружения обновлений расписания
     1. Задача модуля проверять сайт на обновления спустя некоторое время, при наличии обновлений необходимо через механизм Pub/Sub выбросить уведомление, содержащее текущую дату, факультет.
  2. Шестая стадия – Разработка модуля приложения
     1. Модуль приложения связывает Модуль парсера, Модуль календаря, Модуль БД, Модуль рассылки почты, Модуль обнаружения обновления расписания.
     2. Модуль реализует следующие ВИ:
        1. Сохранение и обновление данных пользователя, а также получение файла рассылки на основе Id пользователя.
        2. Получение файла календаря попутно сохранив его используя Модуль БД.
        3. Создание и обновление подписок на обновления расписания
        4. Получение уведомлений из Модуля обнаружения обновлений расписания и реагирование на них путём сохранения нового расписания и использование Модуля рассылки почты для отправки обновлений.
  3. Седьмая стадия – Разработка Чат-Бота Telegram
     1. Модуль реализует следующие ВИ:
        1. Дать возможность пользователю выбрать свой факультет и группу
        2. Дать возможность пользователю подписаться на обновления расписания
        3. Дать возможность пользователю отписаться от обновлений расписания
        4. Дать возможность пользователю изменить свой факультет или группу
        5. Дать возможность пользователю видеть в интерфейсе расписание текущего дня, статус чётности или нечётности недели, дату, день недели.
        6. Дать возможность пользователю мануально скачать последнюю доступную версию расписания, при этом кэш расписания может обновиться.
  4. Восьмая стадия – Реализация автоматического тестирования
     1. Написание автоматических Unit-тестов для модулей.
  5. Девятая стадия – Тестирование на устройствах пользователей

1. Тестирование программного комплекса
   1. Автоматическое тестирование ПО
      1. Unit-тестирование с использованием Jest.
   2. Тестирование на устройствах пользователей
      1. Приложение не было протестировано.

Приложение 1

«Чат-бот Telegram для конвертации расписания любой группы РГСУ в файл календаря»