ОГЛАВЛЕНИЕ

стр.

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 2](#_Toc59792719)

[ВВЕДЕНИЕ 2](#_Toc59792720)

[1. ОБЗОРНАЯ ЧАСТЬ 4](#_Toc59792721)

[1.1. Описание предметной области 4](#_Toc59792722)

[1.1.1. Описание серверной архитектуры 4](#_Toc59792723)

[1.1.2. Описание модели внедрения программных продуктов Agile Scrum 6](#_Toc59792724)

[1.1.3. Схема ручного сопровождения API-документации 6](#_Toc59792725)

[1.2. Исследование существующих технологий 6](#_Toc59792726)

[1.2.1. Перечень функций, подлежащих автоматизации 6](#_Toc59792727)

[1.2.2. Выбор и обоснование критериев качества 6](#_Toc59792728)

[1.2.3. Анализ аналогов и прототипов 6](#_Toc59792729)

[1.2.4. Сравнение аналогов и прототипов 7](#_Toc59792730)

[2. РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 7](#_Toc59792731)

[2.1. Определение требований к системе 7](#_Toc59792732)

[2.2. Разработка структуры автоматизированной системы 7](#_Toc59792733)

[2.3. Разработка структуры интерфейса взаимодействия пользователя с системой 7](#_Toc59792734)

[2.4. Разработка алгоритмов программных модулей 7](#_Toc59792735)

[2.5. Разработка плана проведения тестирования 7](#_Toc59792736)

[3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ 7](#_Toc59792737)

[3.1. Реализация разработанных алгоритмов 8](#_Toc59792738)

[3.2. Тестирование и отладка системы 8](#_Toc59792739)

[3.3. Руководство пользователя 8](#_Toc59792740)

[ЗАКЛЮЧЕНИЕ 8](#_Toc59792741)

[СПИСОК ЛИТЕРАТУРЫ 8](#_Toc59792742)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1. REST API – это набор правил, по которым следует обращаться к серверу для отправки или получения данных.
2. Пользователь – живой человек (определение человека по Платону: «двуногое без перьев»)
3. Клиент – любое приложение которое делает запросы на сервер. Например, в роли клиента может выступать веб браузер, когда пользователь открывает веб-сайт.
4. АС – автоматизированная система
5. ПО – Программное обеспечение
6. API-документация – это техническая документация, в которой фиксируются инструкции о том, как использовать программное API.
7. UI – пользовательский интерфейс.
8. База данных (БД) – это совокупность систематизированных особым образом данных, находящаяся в памяти вычислительной системы. Для работы с БД используются специальные средства – системы управления базами данных (СУБД).

# ВВЕДЕНИЕ

На сегодняшний день большинство крупных IT компаний для взаимодействия сервера и клиента используют REST API [1].

Компании вроде Яндекса, Google и т.п. Предоставляют открытые API методы своих сервисов чтобы разработчики могли интегрироваться с ними.

Например, при получении данных о пользователе, информация о котором храниться в БД (базе данных) на сервере необходимо указать путь до сервера (URI), идентификатор пользователя (ID) и метод (Method) по которому сервер поймет, что нужно сделать с ресурсом, в данном случае вернуть информацию о пользователе. Данный процесс «общения» клиента и сервера, представлен на рисунке 1.1.



Рисунок 1.1 – Схема получения информации о пользователе

Помимо написания самих API методов необходимо написание подробной документации по ним, поскольку без нее попросту не удастся воспользоваться методом. А также не менее важно поддерживать документацию в актуальном состоянии поскольку если документация будет неправильная или устаревшая, то велика вероятность ошибок и в конечном итоге может сказывается на качестве и стоимости продуктов.

Поэтому написание API-документации очень важная и актуальная тема.

Компания ООО «ФорМакс» разрабатывает продукт Fonmix, серверная часть которого полностью базируется на технологии REST API, т.е. общение любого пользователя с сервером Fonmix осуществляется через REST API.

Основными клиентами [3] для сервера Fonmix являются:

1. Веб-сайт fonmix.ru – представляет собой веб интерфейс, в котором пользователи [2] могут управлять музыкой в своих заведения: создавать плейлисты, составлять музыкальное расписание, добавлять рекламу в перерывах между песнями и т.п.
2. FM.Player – кроссплатформенный медиапроигрыватель разрабатываемый также в компании ООО «ФорМакс», с помощью которого воспроизводиться медиа контент правообладателей.
3. Правообладатель – это исполнитель и изготовитель фонограмм, с которым заключается договор о дистрибуции контента и предоставлении отчетов об использовании.

---

Целью данной работы является создание системы автоматического сопровождения API-документации, позволяющей ускорить и повысить качество разработки. В соответствии с поставленной целью, работа над АС (автоматизированной системой) была разделена на несколько этапов, в рамках которых решались следующие задачи:

* анализ предметной области
* обзор и сравнение современных технологий по сопровождению API-документации
* выделение перечня функций, подлежащих автоматизации
* выфвыф

# ОБЗОРНАЯ ЧАСТЬ

## **Описание предметной области**

### Описание серверной архитектуры

Серверная часть проекта Fonmix на разделена на микросервисы.

Микросервисная архитектура[[1]](#footnote-1) – вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей – микросервисов.

Основными микросервисами являются:

* FM.Core – Основное ядро. Количество API методов 253
* FM.CRM – выф. Количество API методов 153
* FM.ID – dsa. Количество API методов 23
* FM.Notify – dsa. Количество API методов 34
* FM.Store – dsa. Количество API методов 15
* FM.Media – dsa. Количество API методов 36

Итого, общее количество методов 514

Схема взаимодействия клиентов и сервера представлена на рисунке 1.2.



Рисунок 1.2 – Схема взаимодействие клиентов и сервера

---

Исходя из этого можно сделать вывод что сопровождение API-документации в ручном режиме может занимать очень много времени.

---

### Описание модели внедрения программных продуктов Agile Scrum

Текст

### Схема ручного сопровождения API-документации

Текст

## **Исследование существующих технологий**

### Перечень функций, подлежащих автоматизации

Текст

### Выбор и обоснование критериев качества

Для проведения сравнительного анализа аналогов и прототипов выбраны следующие критерии:

1. Трудозатраты на изучение технологии
2. Потребность в дополнительном ПО

### Анализ аналогов и прототипов

Текст

### Сравнение аналогов и прототипов

Текст

# РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ



## **Определение требований к системе**

Текст

## **Разработка структуры автоматизированной системы**

Текст

## **Разработка структуры интерфейса взаимодействия пользователя с системой**

Текст

## **Разработка алгоритмов программных модулей**

Текст

## **Разработка плана проведения тестирования**

Текст

# ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

## **Реализация разработанных алгоритмов**

Текст

## **Тестирование и отладка системы**

Текст

## **Руководство пользователя**

Текст

# ЗАКЛЮЧЕНИЕ

Что в итоге получилось.

# СПИСОК ЛИТЕРАТУРЫ

1. Python Documentation – [Электронный ресурс]: <https://docs.python.org/3/>   
   (Дата обращения: 28.09.2020)
2. Bootstrap Documentation – [Электронный ресурс]:  
   <https://getbootstrap.com/docs/3.3/> (Дата обращения: 28.09.2020)
3. PostgreSQL Database Documentation – [Электронный ресурс]: <https://www.postgresql.org/docs/> (Дата обращения: 28.09.2020)
4. Скотт Б., Нейл Т. Проектирование веб-интерфейсов. – СПб.: Символ-Плюс, 2010. – 352 с.

1. https://ru.wikipedia.org/wiki/Микросервисная\_архитектура [↑](#footnote-ref-1)