**ОГЛАВЛЕНИЕ**

стр.

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 2](#_Toc59553296)

[ВВЕДЕНИЕ 2](#_Toc59553297)

[1. ОБЗОРНАЯ ЧАСТЬ 2](#_Toc59553298)

[1.1. Описание предметной области 2](#_Toc59553299)

[1.1.1. Описание серверной архитектуры 2](#_Toc59553300)

[1.1.2. Описание модели внедрения программных продуктов Agile Scrum 2](#_Toc59553301)

[1.1.3. Схема ручного сопровождения API-документации 2](#_Toc59553302)

[1.2. Исследование существующих технологий 2](#_Toc59553303)

[1.2.1. Перечень функций, подлежащих автоматизации 2](#_Toc59553304)

[1.2.2. Выбор и обоснование критериев качества 3](#_Toc59553305)

[1.2.3. Анализ аналогов и прототипов 3](#_Toc59553306)

[1.2.4. Сравнение аналогов и прототипов 3](#_Toc59553307)

[2. РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 3](#_Toc59553308)

[2.1. Определение требований к системе 3](#_Toc59553309)

[2.2. Разработка структуры автоматизированной системы 3](#_Toc59553310)

[2.3. Разработка структуры интерфейса взаимодействия пользователя с системой 3](#_Toc59553311)

[2.4. Разработка алгоритмов программных модулей 3](#_Toc59553312)

[2.5. Разработка плана проведения тестирования 4](#_Toc59553313)

[3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ 4](#_Toc59553314)

[3.1. Реализация разработанных алгоритмов 4](#_Toc59553315)

[3.2. Тестирование и отладка системы 4](#_Toc59553316)

[3.3. Руководство пользователя 4](#_Toc59553317)

[ЗАКЛЮЧЕНИЕ 4](#_Toc59553318)

[СПИСОК ЛИТЕРАТУРЫ 4](#_Toc59553319)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1. Пользователь – живой человек (определение человека по Платону: «двуногое без перьев»)

# ВВЕДЕНИЕ

---

Компания ООО «ФорМакс» разрабатывает продукт Fonmix.

Fonmix – это SaaS[[1]](#footnote-1) платформа для аудио и видео оформления общественных заведениях.

Пользователь регистрируется на сайте, оформляет договор с ООО «ФорМакс», настраивает музыкальное расписание, скачивает и устанавливает плеер себе на устройство либо приобретает FM.Box – сетевой медиаплеер с установленным приложением FM.Player. После чего может легально пользоваться музыкальным сопровождением.

Помимо воспроизведения музыки, Fonmix собирает информацию о том где и когда воспроизводился тот или иной трек чтобы

---

REST API – это набор правил, по которым следует обращаться к серверу за получением информации. Данный метод отправки и получения данных базируется на протоколе HTTP[[2]](#footnote-2).

---

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

---

Например, при получении данных о пользователе, информация о котором храниться в БД (базе данных) на сервере необходимо:

1. Указать путь до сервера (URI)
2. Авторизоваться на сервере
3. Передать идентификатор пользователя (ID)

Данный процесс «общения» клиента и сервера, представлен на рисунке 1.1.

---

---

Поэтому написание API-документации очень важная и актуальная тема. Чем качественнее и точнее она будет, тем …

---

Архитектура сервера спроектирована таким образом, что каждый клиент взаимодействует с определенной частью сервиса.

---

Соответственно, помимо написания кода API метода, в обязанности бэкенд разработчика входит написание и поддержание в актуальном состоянии API-документации.

Поскольку написание документации занимает 1/3 всей разработки, было принято решение автоматизировать данный процесс.

---

Написание подробной документации как пользоваться API методами входит в обязанности любого бэкенд разработчика и/или технического писателя.

---

Соответственно написание и поддержание в актуальном состоянии документации как пользоваться API методами для всех клиентов входит в обязанности любого бэкенд разработчика и/или технического писателя.

---

# ОБЗОРНАЯ ЧАСТЬ

## **Описание предметной области**

### Описание серверной архитектуры

Текст

### 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

Текст

### Анализ аналогов и прототипов

### Postman

---

Основным понятия, которые оперирует Postman являются Collection (коллекции) на верхнем уровне и Request (запрос) на нижнем. Коллекции предназначены для группировки запросов по проектам.

---

---

Возможности Postman достаточно обширны и выходят за рамки данной дипломной работы.

---

### Ручной метод сопровождения API-документации

---

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

---

### Сравнение аналогов и прототипов

Соответствие рассматриваемых аналогов указанным критерием представлено в таблице 1.1. В каждой ячейке стоит соответствие критерия и степень качества критерия:

* A – Отлично
* B – Очень хорошо
* C – Хорошо
* D – Удовлетворительно
* E – Плохо
* F – Очень плохо

Таблица 1.1. – Качественные характеристики аналогов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Swagger | Blueprint | RAML | Ручное сопровождение документации | Postman |
| Трудозатраты на изучение технологии | E |  |  | C | D |
| Потребность в дополнительном ПО | F |  |  | B | B |
| Настраиваемость системы | D |  |  | A | C |
| Время, затрачиваемое на сопровождение документации | E |  |  | F | B |
| Публикация документации в ЕСС | C |  |  | A | A |
|  |  |  |  |  |  |

# 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://en.wikipedia.org/wiki/Software\_as\_a\_service [↑](#footnote-ref-1)
2. https://ru.wikipedia.org/wiki/HTTP [↑](#footnote-ref-2)