**ОГЛАВЛЕНИЕ**

стр.

[ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 2](#_Toc58776524)

[ВВЕДЕНИЕ 2](#_Toc58776525)

[1. ОБЗОРНАЯ ЧАСТЬ 2](#_Toc58776526)

[1.1. Описание предметной области 2](#_Toc58776527)

[1.1.1. Основные элементы и схема взаимодействия 2](#_Toc58776528)

[1.1.2. Разработка API и написание API-документации 5](#_Toc58776529)

[1.2. Анализ технологий и возможных средств решения проблемы 5](#_Toc58776530)

[1.3. Выбор средств и технологий 6](#_Toc58776531)

[2. РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ 6](#_Toc58776532)

[3.1. Определение требований к системе 6](#_Toc58776534)

[3.2. Разработка структуры автоматизированной системы 6](#_Toc58776535)

[3.3. Разработка структуры интерфейса взаимодействия пользователя с системой 6](#_Toc58776536)

[3.4. Разработка алгоритмов программных модулей 6](#_Toc58776537)

[3.5. Разработка плана проведения тестирования 7](#_Toc58776538)

[3. ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ 7](#_Toc58776539)

[3.1. Реализация разработанных алгоритмов 7](#_Toc58776540)

[3.2. Тестирование и отладка системы 7](#_Toc58776541)

[3.3. Руководство пользователя 7](#_Toc58776542)

[ЗАКЛЮЧЕНИЕ 7](#_Toc58776543)

[СПИСОК ЛИТЕРАТУРЫ 7](#_Toc58776544)

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

АС – автоматизированная система.

API (Application Programming Interface) – это набор правил по которым

API-документация – это техническая документация, в которой фиксируются инструкции о том, как использовать программное API.

UI – пользовательский интерфейс.

# ВВЕДЕНИЕ

Текст введения. Тут надо что-то сказать о API документации, для кого она нужна, кем пишется и т.п.

# ОБЗОРНАЯ ЧАСТЬ

## **Описание предметной области**

### Основные элементы и схема взаимодействия

Компания ООО «ФорМакс» разрабатывает продукт (не совсем уверен что слово продукт подходит) Fonmix.

Fonmix – это SaaS[[1]](#footnote-1) платформа для аудио и видео оформления общественных заведений. Схема взаимодействия представлена на рисунке 1.1.

Основные элементы платформы являются:

1. Веб-сайт fonmix.ru – представляет собой веб интерфейс, в котором пользователи 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 физическое или юридическое лицо, с которым заключается договор о дистрибуции его контента и предоставлении отчетов об использовании.
4. Сервер Fonmix – набор веб серверов принимающие HTTP-запросы от клиентов и выдающий HTTP-ответы.

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. (стоит это оставить или убрать?)

Правообладатель (а нужно ли о нем что-то писать?)



Рисунок 1.2 – Схема взаимодействие клиентов и сервера

Для взаимодействия всех клиентов и серверной частей используется REST API.

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 1.2.



Рисунок 1.2 – Схема получения информации о пользователе

### Разработка API и написание API-документации

Для разработки продукта Fonmix используется Scrum[[3]](#footnote-3) методология разработки ПО основоная на Agile[[4]](#footnote-4).

Схема разработки (одной итерации) с точки зрения бэкенд разработчика нового функционала представлена на рисунке 1.3.

TODO: расписать этапы

1. Составление требований о разработке нового функционала
2. Написание кода (API)
3. Документация API
4. Передача в тестирование (в редких случаях покрытие автотестами ☹ )

По окончанию работы над API методом или набором методов, бэкенд разработчик или технический писатель, обязан предоставить подробную документацию о том, как пользоваться методом, какие параметры он принимает, по какому пути будет доступен метод, а также примеры запросов и ответов от сервера. Соответствующий документ должен располагаться в единой справочной системе компании.

## **Анализ технологий и возможных средств решения проблемы**

Тут перечень аналогов, как другие компании решают автоматизацию ведения документации.

## **Выбор средств и технологий**

Тут текст о том, что какое решение в итоге выбрано + аргументировать почему

# РАСЧЕТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ



## **Определение требований к системе**

Текст (без понятия о чем писать…)

## **Разработка структуры автоматизированной системы**

Тут текст о…

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

Тут о том, как разрабатывался интерфейс? (вообще это будет консольная утилита и сомневаюсь, что интерфейс вообще понадобится) однако можно рассказать о том, как устроен интерфейс в постмане и как вообще с ним работать (+5-7 листов)

## **Разработка алгоритмов программных модулей**

Тут можно написать (возможно добавить блок-схему) о том, как выглядит структура данных в постмане, о том какие данные пригодятся и как их достать.

+ можно описать как будет производится авторизация в конфлюенсе

## **Разработка плана проведения тестирования**

Тут о том, как проводить тестирование ???

# ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

## **Реализация разработанных алгоритмов**

Тут текст о том как работает скрипт. Как он по структуре из постмана рекурсивно выбирает информацию чтобы потом сделать страницу в конфлюенсе.

## **Тестирование и отладка системы**

Тут как проверял и отлаживал программу. Возможно напишу про дебаггер в perl

## **Руководство пользователя**

TODO: почитать как правильно составлять руководство пользователя

# 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)
3. https://ru.wikipedia.org/wiki/Scrum [↑](#footnote-ref-3)
4. https://ru.wikipedia.org/wiki/Agile [↑](#footnote-ref-4)