Содержание

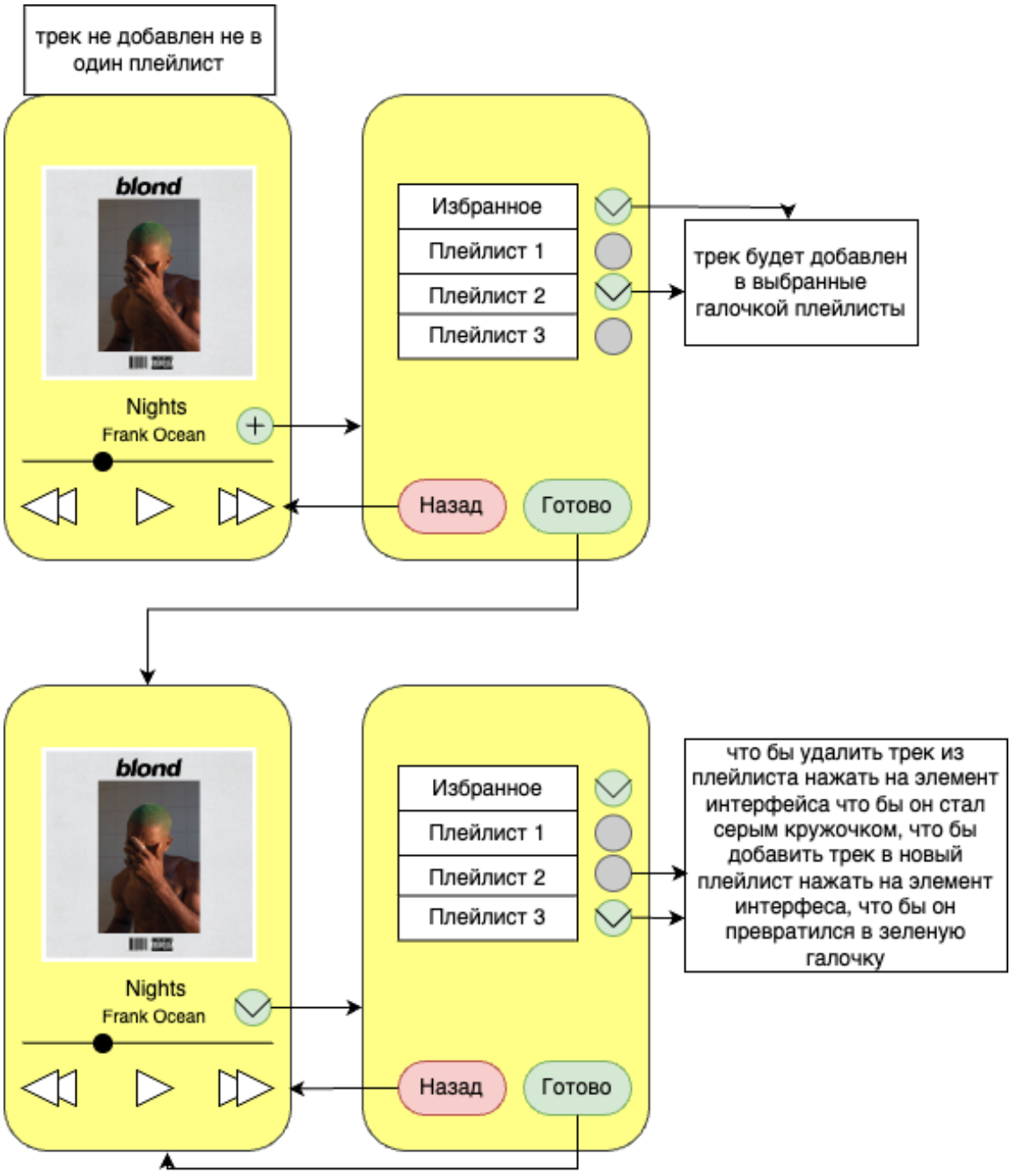
Пакет бизнес требований

1. Фича
Добавление и удаление треков из плей-листов

2. Название продукта
Приложение «музыка»

3. User story
Как пользователь приложения «Музыка», я хочу иметь возможность добавлять и удалять треки из своих плей-листов, что бы удобнее управлять своей музыкальной коллекцией

4. Макет



5. Use case

Заголовок	Удаление и добавление трека в разные плей-листы
Акторы	Пользователь
Предусловие	Пользователь зарегистрирован и авторизован в приложении, у пользователя есть несколько созданных плей-листов, пользователь находится на экране проигрывание трека. Пользователю доступен элемент добавления/удаления трека.
Ограничения	Экран добавления трека в плей-лист отображает только 5 плей-листов
Триггер	Пользователь нажимает на элемент управления треком
Основной сценарий 1	1. Система определила что трек уже добавлен в какой то плейлист отобразила это элементом управления(он в галочке)(экран 3) 2. Клиент тапает на элемент управления 3. Система отображает плей-листы пользователя(зеленой галочкой отмечены плейлисты в которые добавлен трек, серыми кружочками в которые можно добавить)(экран 4) 4. Пользователь тапнул на плейлист в который трек еще не добавлен 5. Система отобразила элемент интерфейса рядом с плей-листом на который тапнул пользователь зеленой галочкой 6. Пользователь тапнул кнопку "Готово" 7. Система вернула пользователя на экран воспроизведения трека. **Критерий успеха:** трек добавлен в выбранный плейлист
Альтернативный сценарий 1	1. Система определила что трек еще не добавлен ни в один плей-лист отобразила это элементом управления(он в плюсике)(экран 1) 2. Клиент тапает на элемент управления 3. Система отображает плей-листы пользователя в которые можно добавить трек (экран 2) 4. Пользователь тапнул на 2 плей-листа 5. Система отобразила элемент интерфейса рядом с плей-листами на который тапнул пользователь зеленой галочкой 6. Пользователь тапнул кнопку "Готово" 7. Система вернула пользователя на экран воспроизведения трека. **Критерий успеха:** трек добавлен в выбранные плейлисты
Альтернативный сценарий 2	4а. На 4 шаге основного сценария пользователь тапнул на плейлист в который трек уже добавлен. 5а. Система отобразила элемент интерфейса рядом с плей-листом на который тапнул пользователь серым кружочком(проинформировала о том что трек будет удален из этого плей-листа). 6а. Пользователь тапнул кнопку "Готово". 7а. Система вернула пользователя на экран воспроизведения трека. **Критерий успеха:** трек удален из выбранного плей-листа
Альтернативный сценарий 3	4б. На 4 шаге основного сценария пользователь тапнул на плейлист в который трек уже добавлен и на плейлист в в который трек еще не добавлен. 5б. Система отобразила элемент интерфейса рядом с плей-листом, в который трек был добавлен, серым кружочком(проинформировала о том что трек будет удален из этого плей-листа) и рядом с плей-листом, в котором трека небыло, зеленой галочкой(трек будет добавлен) 6б. Пользователь тапнул кнопку "Готово". 7б. Система вернула пользователя на экран воспроизведения трека. **Критерий успеха:** трек удален из выбранного плей-листа и добавлен в другой выбранный плейлист

Пакет функциональных требований

1. Архитектура

A - фронтед, мобильное приложение

B - бекенд/сервер

C - Реаляционная база данных, PG

1 - Протокол взаимодействий HTTPS

2 - Протокол взаимодействий TCP/IP

2. Модель данных

Обьект User

Родительская сущность	Атрибут	Описание
User	-	Обьект пользователя, который имеет атрибуты и ссылки на другие объекты
	user_login	Имя пользователя
	email	Электронная почта
	playlist	Плей-листы пользователя(Ссылка на сущность **playlist**)

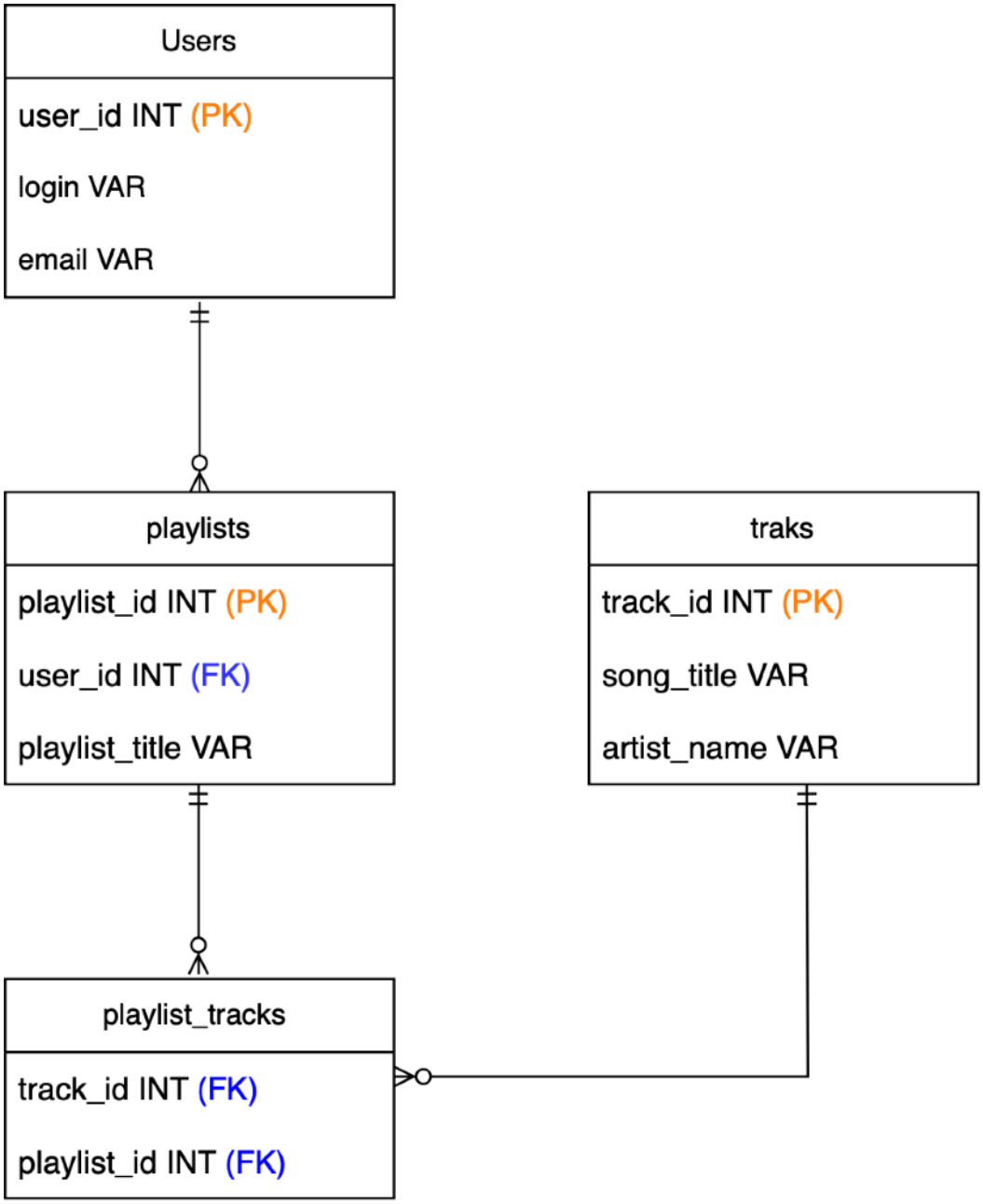
Обект Playlist

Родительская сущность	Атрибут	Описание
Playlist	-	Обьект пользователя, который имеет атрибуты и ссылки на другие объекты
	playlist_name	Название плейлиста
	User	Создатель плейлиста(Ссылка на сущность **user**)
	track	Треки плей-листа(Ссылка на сущность **track**)

Обект track

Родительская сущность	Атрибут	Описание
track	-	Обьект пользователя, который имеет атрибуты и ссылки на другие объекты
	track_name	Название трека
	artist	Исполнитель
	playlist	Плейлисты в которые входит трек(Ссылка на сущность **playlist**)

3. ERD-диаграмма



4. Sequence-диаграмма

1	Пользователь на интерфейсе мобильного приложения выбирает плей-листы в которые будет добавлен трек и нажимает кнопку "Готово"
2	Мобильное приложение отправляет http-запрос на backend на добавление трека в выбранный список плей-листов
3	backend отправляет sql запрос в базу данных на изменение данных
4	Data base возвращает ответ о том что данные успешно изменены
5	Backend отправляет на Mobile app json файл с ответом
6	Mobile app изменяет кнопку "плюсик" на "галочку" и возвращает пользователя на экран прослушивания музыки

5. REST. Табличный вид

POST/playlists_tracks

Пользователь добавляет трек в выбранные плейлисты

Requset				
Название парамтера	Тип данных	Описание	Обязательность	находится в
user_id	int	Уникальный идентификатор пользователя	да	body
track_id	int	Уникальный идентификатор трека	да	body
arr_playlist_id	array of int	Массив идентификаторов плей-листов, в которые будет добавлен трек	да	body

Response 200 OK

Сервер возвращает ответ об успешном добавлении трека в плейлисты

Response				
Название парамтера		Тип данных	Описание	Обязательность
add_playlist_obj_arr		array of object	Массив объектов, где объект состоит из (playlist_id, code)	да
	playlist_id	int	Уникальный идентификатор плей-листа	да
	code	int	Код сообщения, добавился ли трек в данный плей-лист(1 - добавился, 2-ошибка)	да

GET/playlists/{user_id}

Пользователь запрашивает список своих плей-листов

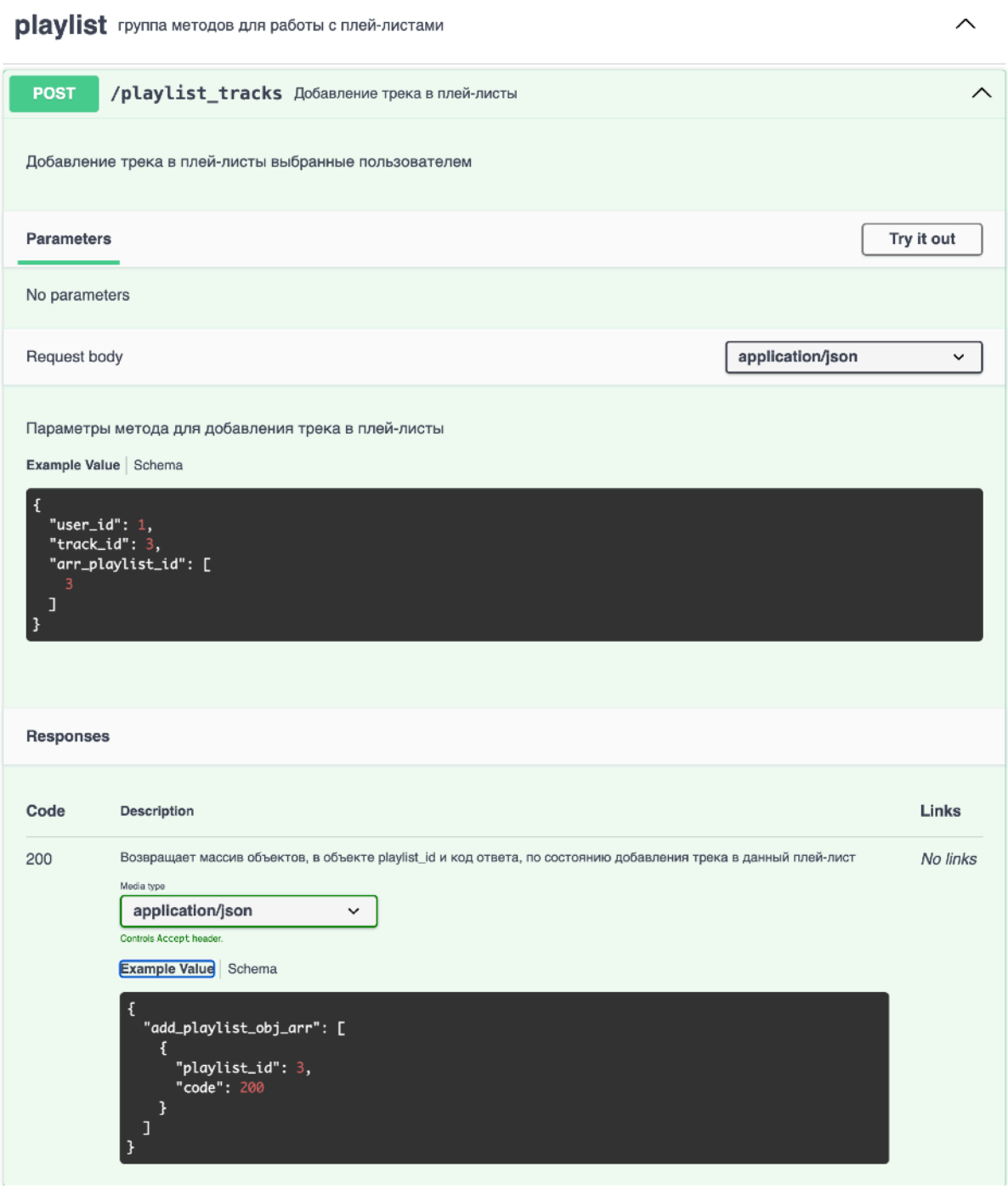
Request				
Название парамтера	Тип данных	Описание	Обязательность	находится в
user_id	int	Уникальный идентификатор пользователя	да	path

code: 200 OK

Сервер возвращает плей-листы пользователя

Response					
Название парамтера		Тип данных	Описание	Обязательность	находится в
arr_playlist_obj		array of object	Массив объектов, где объект состоит из (playlist_id, playlist_title)	да	body
	playlist_id	int	Уникальный идентификатор плей-листа	да	body
	playlist_title	string	Название плей-листа	да	body
code		string	Код сообщения	да	body

6. REST. Swagger



GET `/playlists/{user_id}` Получить список плей-листов

Получить список плей-листов по уникальному идентификатору пользователя

Parameters Try it out

Name	Description
user_id * required `string` *(path)*	уникальномый идентификатор пользователя user_id

Responses

Code	Description	Links
200	Возвращает массив объектов, где в один объект входит playlist_id и playlist_title соответсвенно Media type application/json Controls Accept header. **Example Value** | Schema	*No links*

```
{
  "arr_playlist_obj": [
    {
      "playlist_id": 1,
      "playlist_title": "favorite"
    }
  ],
  "code": 200
}
```

Schemas

playlist_traks_request {
- user_id* > [...]
- track_id* > [...]
- arr_playlist_id* > [...]

}

playlist_traks_response {
- add_playlist_obj_arr* > [...]

}

get_playlists_response {
- arr_playlist_obj* > [...]
- code* > [...]

}

Критерии приемки

Кейс 1

Функциональность: Удаление и добавление трека в выбранные плей-листы
Дано: Пользователь нажимает на кнопку "Добавление трека в плей-листы". Трек уже добавлен в два плей-листа, остальные плей-листы не имеют данного трека. Пользователь попадает на экран выбора плей-листов, где отображаются галочки для плей-листов, в которые трек уже добавлен, а остальные плей-листы отмечены пустыми чекбоксами.
Когда: Пользователь не меняет расположение галочек и нажимает «готово»
Тогда: Система вернет пользователя на экран прослушивание и составы плей-листов не изменяться

Кейс 2

Функциональность: Удаление и добавление трека в выбранные плей-листы
Дано: У пользователя нет ранее созданных плей-листов кроме «Избранное». Пользователь нажимает на кнопку добавление трека в плей-листы(трек еще не добавлен ни в один плей-лист) и оказывается на экране выбора плей-листов где отображается один плей-лист «Избранное».
Когда: Пользователь отметил на интерфейсе галочкой плей-лист «Избранное»
Тогда: Система добавит трек в плей-лист избранное

Нефункциональные требования

НФТ относятся к атрибутам качества системы, которые определяют, как она работает, а не что она делает.

Пример

Требования надежности:
1. Система должна быть доступна 99% времени.
Требования производительности:
1. Страница выбора плей-листов должна открываться не более 2 секунд.
2. Запрос плей-листов пользователя GET /playlists/{user_id} должен выдерживать нагрузку 1 rps.