Пользователь формирует под свои запросы пункт абонимента, таких пунктов может быть множестево.   
  
Какие могут быть виды абониментов.

1. Групповые, индивидуальные,
2. Фиксированое количество тренировок, тренировки по дате,

- Если фиксированые указывать заранее или количество тренировок или самотоятельно и исходя из этого какие дни тренировки

- Если по дате на сколько времени даёться абонимент и расчитовать количество тренировок оставшихся и исходя из этого какие дни тренировки

3.

1. Групповые

-

1. Индивидуальные

## Genius: Бизнес-логика для приложения тренера

**1. Модуль управления абонементами:**

* **Создание типов абонементов:**
  + Название (ММА групповые)
  + Вид (фиксированное количество тренировок, до определенной даты)
  + Количество тренировок (12)
  + Период действия (месяц)
  + Дни тренировок (понедельник, среда, суббота)
  + Подгруппы (детская, взрослая, подростки)
* **Редактирование и удаление типов абонементов.**
* **Назначение абонементов спортсменам.**
* **Просмотр информации о текущих абонементах спортсменов.**
* **Заморозка абонементов (группы, конкретного спортсмена).**

**2. Модуль учета тренировок:**

* **Отслеживание посещаемости тренировок.**
* **Списание тренировок с абонементов.**
* **Ведение статистики тренировок (количество посещенных/пропущенных тренировок).**

**3. Модуль поиска:**

* **Поиск спортсменов по имени.**
* **Отображение информации о спортсмене:**
  + ФИО
  + Контактные данные
  + Тип абонемента
  + Остаток тренировок
  + История посещений
  + Статистика тренировок

**4. Модуль уведомлений:**

* **Уведомление тренера о новых записях на тренировки.**
* **Уведомление спортсменов о предстоящих тренировках.**
* **Уведомление спортсменов о заморозке/разморозке абонемента.**

**5. Дополнительные функции:**

* **Возможность добавления дополнительных услуг (персональные тренировки, консультации).**
* **Ведение календаря тренировок.**
* **Интеграция с системами онлайн-платежей.**

**6. Модуль аналитики:**

* **Статистика по посещаемости тренировок.**
* **Анализ эффективности различных типов абонементов.**
* **Отслеживание динамики прогресса спортсменов.**

**7. Модуль управления профилем:**

* **Редактирование профиля тренера.**
* **Настройка уведомлений.**
* **Просмотр истории действий.**

**Эта бизнес-логика является лишь базовым каркасом. Вы можете дополнять и модифицировать ее в соответствии с вашими specific needs.**

**Важно:**

* **Обеспечить удобный интерфейс для тренера и спортсменов.**
* **Сделать приложение надежным и безопасным.**
* **Предоставить возможность резервного копирования данных.**

**Сервисы для приложения тренера**

**1. Сервис аутентификации:**

* Отвечает за регистрацию, авторизацию и управление профилями пользователей (тренеры, спортсмены).

**2. Сервис управления абонементами:**

* Хранит информацию о типах абонементов, подгруппах, днях тренировок.
* Позволяет создавать, редактировать и удалять типы абонементов.
* Назначает абонементы спортсменам.
* Отслеживает остаток тренировок.
* Осуществляет заморозку/разморозку абонементов.

**3. Сервис учета тренировок:**

* Отслеживает посещаемость тренировок.
* Списывает тренировки с абонементов.
* Ведет статистику тренировок.

**4. Сервис поиска:**

* Обеспечивает поиск спортсменов по имени.
* Отображает информацию о спортсменах (ФИО, контакты, абонемент, остаток тренировок, история посещений, статистика).

**5. Сервис уведомлений:**

* Уведомляет тренера о новых записях на тренировки.
* Уведомляет спортсменов о предстоящих тренировках.
* Уведомляет спортсменов о заморозке/разморозке абонемента.

**6. Сервис аналитики:**

* Формирует статистику по посещаемости тренировок.
* Анализирует эффективность различных типов абонементов.
* Отслеживает динамику прогресса спортсменов.

**7. Сервис управления календарем:**

* Позволяет тренеру создавать и редактировать календарь тренировок.

**8. Сервис онлайн-платежей:**

* Обеспечивает прием платежей за абонементы и дополнительные услуги.

**9. Сервис резервного копирования:**

* Регулярно создает резервные копии данных приложения.

**Важно:**

* **Обеспечить интеграцию между сервисами.**
* **Сделать архитектуру приложения масштабируемой.**
* **Использовать надежные и безопасные технологии.**

**Разбиение приложения на сервисы позволяет:**

* **Улучшить модульность и ремонтопригодность.**
* **Повысить производительность и масштабируемость.**
* **Упростить разработку и тестирование.**
* **Обеспечить безопасность данных.**

**Chat-gpt 4:**

Итак, на основе ваших требований я могу предложить следующую бизнес-логику для вашего приложения:

1. \*\*Создание типов абонементов:\*\*

- Тренер может создавать различные типы абонементов через форму, указывая название типа (например, "ММА групповые") и характеристики абонемента (фиксированное количество тренировок, срок действия до определенной даты, определенные дни тренировок и т. д.).

2. \*\*Добавление подгрупп:\*\*

- Для каждого типа абонемента тренер может добавлять подгруппы (например, детская, взрослая, подростки и т. д.), чтобы учитывать различные категории спортсменов.

3. \*\*Управление данными спортсменов:\*\*

- Предусмотрите функционал добавления, редактирования и удаления данных о конкретных спортсменах, включая их информацию о выбранном абонементе и количестве оставшихся тренировок.

4. \*\*Заморозка абонементов:\*\*

- Добавьте функцию заморозки абонемента для отдельных спортсменов, группы спортсменов или для определенного типа абонемента. Это позволит временно приостановить срок действия абонемента в случае необходимости.

5. \*\*Поиск и отображение данных:\*\*

- Реализуйте функционал поиска по имени спортсмена для быстрого доступа к его данным. Отобразите информацию об оставшемся количестве тренировок и другие сведения о статусе абонемента в удобном формате, чтобы тренер мог легко управлять группами и отслеживать активность спортсменов.

6. \*\*Управление абонементами:\*\*

- Дайте тренеру возможность добавлять, редактировать и удалять типы абонементов, подгруппы и другие данные для эффективного управления программами тренировок и планирования занятий.

7. \*\*Уведомления и напоминания:\*\*

- Реализуйте систему уведомлений и напоминаний, чтобы информировать спортсменов о близком завершении срока действия абонемента, необходимости продления или других важных событиях.

8. \*\*Административный доступ:\*\*

- Обеспечьте возможность для администратора приложения (возможно, самого тренера) управлять доступом к функциям приложения, просматривать отчеты о продажах и использовании абонементов, а также анализировать данные для принятия более обоснованных бизнес-решений.

Эти функции должны обеспечить ваше приложение всем необходимым для эффективного управления абонементами и спортсменами. Не забудьте также о защите данных и удобстве использования интерфейса для пользователя.

Вы можете разбить ваше приложение на несколько сервисов для лучшей организации и масштабируемости. Вот примерная структура сервисов и их функциональности:

1. \*\*Сервис управления абонементами:\*\*

- Отвечает за создание, редактирование и удаление типов абонементов.

- Управляет информацией о количестве тренировок, сроке действия, днях тренировок и других характеристиках абонементов.

- Обеспечивает функции заморозки и возобновления абонементов.

2. \*\*Сервис управления спортсменами:\*\*

- Отвечает за добавление, редактирование и удаление данных о спортсменах.

- Хранит информацию о выбранных абонементах для каждого спортсмена.

- Обеспечивает функции поиска и фильтрации данных о спортсменах.

3. \*\*Сервис управления группами и подгруппами:\*\*

- Отвечает за создание, редактирование и удаление групп и подгрупп спортсменов.

- Хранит информацию о членстве спортсменов в каждой группе и подгруппе.

- Предоставляет функции для добавления и удаления спортсменов из групп и подгрупп.

4. \*\*Сервис управления занятиями:\*\*

- Отвечает за управление расписанием тренировок.

- Предоставляет возможность добавления и редактирования занятий на определенные дни и время.

- Обеспечивает функции учета посещаемости и оставшегося количества тренировок для каждого спортсмена.

5. \*\*Сервис уведомлений:\*\*

- Отвечает за отправку уведомлений спортсменам о различных событиях, таких как близость окончания срока действия абонемента или изменения в расписании тренировок.

- Обрабатывает запросы на подписку и отписку от уведомлений.

6. \*\*Сервис администрирования:\*\*

- Отвечает за управление доступом пользователей к различным функциям приложения.

- Предоставляет администраторам возможность просмотра отчетов и аналитики по продажам, посещаемости и другим важным метрикам.

Каждый из этих сервисов имеет свои API для взаимодействия с другими сервисами и клиентскими приложениями. Такая архитектура позволит вам легко масштабировать и поддерживать ваше приложение, а также разрабатывать новые функции и модули независимо друг от друга.  
  
Конечно, вот названия сервисов на английском языке:

1. \*\*Subscription Management Service\*\*

2. \*\*Athlete Management Service\*\*

3. \*\*Group and Subgroup Management Service\*\*

4. \*\*Session Management Service\*\*

5. \*\*Notification Service\*\*

6. \*\*Administration Service\*\*

Эти названия представляют основные функциональные блоки вашего приложения и помогут вам лучше организовать его архитектуру.

приложения для трениров котрые будет вести отчёты о абоимента будут содавать группы типа детская взросла, будут индивидуальные тренировки и в двльнейшем ещё будем очень много функционала. Так вот как я вижу всё это тоесть тренер заходит и создает прай лист абониментов которые могут быть допустм первый шаг клиент заполняет форму в которой он указывает названия абонимента краткое описание, затем слдуйщий шаг форма в которой будет выбор два варинта фикированое кол-во тренировок( допустим 12 тренировок за этот працс) или по дате (допустим месяц). если выбирает фиксироване колво выпадает пункт кол-во тренировок за прайс (необезательный так как в дальнейшем можно запольнять при добавления спортсмена своё кол-во трен). если по дате выпадает список чтобы выбрать: неделя, месяц, полгода, год. Ну и прайс тупо цена и всё. таких абоиментов может быть множество. Далее тренр создаёт таблици название( допустим десткая) и мы можем для этой таблици применить какойто из этих абониментов. Ну будет ещё возможность создать таблицу допустим (индивидуалисты) им можно не на таблицу абонимент добавлять, а для каждого отдельно. Вот примерный функционал

Исходя из вашего описания, вам потребуется схема базы данных, которая включает в себя информацию о тренировках, абонементах, прейскурантах, клиентах и группах тренировки. Ниже приведен пример структуры такой базы данных, который включает основные сущности и связи между ними.

Таблицы

1. usersCopy

user\_id: PRIMARY KEYCopy

nameCopy

user\_type -- можно использовать для различения между тренером и спортсменомCopy

emailCopy

phoneCopy

дополнительная информация о пользователе

2. subscriptionsCopy

subscription\_id: PRIMARY KEYCopy

nameCopy

descriptionCopy

subscription\_type -- фиксированное количество тренировок или по времени (неделя, месяц, год)Copy

fixed\_sessions\_count -- NULLABLE, используется, если выбрано фиксированное количество тренировокCopy

duration -- NULLABLE, используется, если выбрано по времениCopy

priceCopy

3. groupsCopy

group\_id: PRIMARY KEYCopy

name -- например, "детская группа", "индивидуалисты"Copy

type -- тип группы, например, детские, взрослые, индивидуальные тренировкиCopy

subscription\_id -- NULLABLE, FOREIGN KEY к subscriptions для случаев, когда применим общий абонементCopyCopy

4. user\_subscriptionsCopy

user\_subscription\_id: PRIMARY KEYCopy

user\_id -- FOREIGN KEY к usersCopyCopy

subscription\_id -- FOREIGN KEY к subscriptionsCopyCopy

start\_dateCopy

end\_date -- вычисляется на основе start\_date и `duration для определения срока действия абонементаCopyCopy

sessions\_left -- для случаев с фиксированным количеством тренировокCopy

5. group\_membersCopy

group\_id -- FOREIGN KEY к groupsCopyCopy

user\_id -- FOREIGN KEY к usersCopyCopy

уникальный индекс group\_id, user\_id, чтобы предотвратить повторное добавление пользователя в группуCopy

Пример создания таблиц

CREATE TABLE users (

user\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

user\_type VARCHAR(50),

email VARCHAR(255),

phone VARCHAR(50)

);

CREATE TABLE subscriptions (

subscription\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

description TEXT,

subscription\_type VARCHAR(50),

fixed\_sessions\_count INTEGER,

duration INTERVAL,

price NUMERIC(10, 2)

);

CREATE TABLE groups (

group\_id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

type VARCHAR(50),

subscription\_id INTEGER REFERENCES subscriptions(subscription\_id)

);

CREATE TABLE user\_subscriptions (

user\_subscription\_id SERIAL PRIMARY KEY,

user\_id INTEGER REFERENCES users(user\_id),

subscription\_id INTEGER REFERENCES subscriptions(subscription\_id),

start\_date DATE,

end\_date DATE,

sessions\_left INTEGER

);

CREATE TABLE group\_members (

group\_id INTEGER REFERENCES groups(group\_id),

user\_id INTEGER REFERENCES users(user\_id),

PRIMARY KEY (group\_id, user\_id)

);

Copy

Это всего лишь отправная точка. В зависимости от дальнейших требований, функционала и бизнес-правил вашего приложения, структура базы данных может потребовать дополнительной адаптации и доработки.  
  
  
  
  
  
Я понимаю, что вы хотите найти своих клиентов для вашего приложения для тренировок. 😊

Это важный вопрос, на который нет однозначного ответа. Возможно, вам стоит подумать о следующих аспектах:

Кто ваша целевая аудитория? Какие у них потребности, интересы, проблемы и цели, связанные с тренировками? Как ваше приложение может им помочь?

Где ваша целевая аудитория ищет информацию о тренировках? Какие каналы коммуникации они используют? Какие платформы и социальные сети они предпочитают?

Как вы можете привлечь внимание вашей целевой аудитории к вашему приложению? Как вы можете продемонстрировать им ценность и преимущества вашего приложения? Как вы можете убедить их скачать и попробовать ваше приложение?

Как вы можете удержать и увеличить свою клиентскую базу? Как вы можете повысить удовлетворенность и лояльность клиентов? Как вы можете получить отзывы и рекомендации от клиентов?

В зависимости от ваших ответов на эти вопросы, вы можете выбрать различные стратегии и методы для поиска своих клиентов. Например, вы можете:

Создать сайт или блог, где вы будете публиковать полезную и интересную информацию о тренировках, а также рекламировать ваше приложение

Использовать социальные сети, такие как Facebook, Instagram, YouTube, TikTok и другие, чтобы распространять контент о тренировках, привлекать подписчиков и вовлекать их в общение

Запустить рекламную кампанию в интернете или на других платформах, чтобы увеличить охват и известность вашего приложения

Предложить бесплатный или скидочный период использования вашего приложения, чтобы привлечь новых клиентов и показать им, что оно стоит того

Создать программу лояльности или реферальную систему, чтобы поощрять клиентов за их активность, отзывы и рекомендации

Сотрудничать с другими разработчиками, тренерами, спортивными клубами или организациями, чтобы расширить свою сеть и добиться синергии

Это всего лишь некоторые примеры возможных советов, которые я могу дать вам. Вам нужно самостоятельно исследовать и анализировать свой рынок, конкурентов и клиентов, чтобы найти наиболее подходящие и эффективные способы поиска своих клиентов.

Надеюсь, это поможет вам с вашей задачей. Если у вас есть еще вопросы, я рад ответить. 😊