

# JOIN MONEY APP

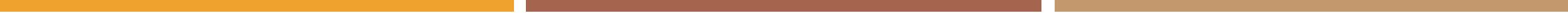
СЮКОВ СЕРГЕЙ

12.06.2017



# СОДЕРЖАНИЕ

- Постановка задачи, возможности и цели приложения
- Структура приложения и решения



# ПОСТАНОВКА ЗАДАЧИ, ВОЗМОЖНОСТИ И UI ПРИЛОЖЕНИЯ



# ПОСТАНОВКА ЗАДАЧИ

- В маленькой компании работают 5 человек. Два учредителя и трое наемных работников. Компания на ендв схеме налогообложения, поэтому нет необходимости в налоговой отчетности за каждую денежную операцию. В денежном обороте принимают участие все сотрудники (прием предоплаты, закупка материалов, оплата накладных расходов, прием пост оплаты и т.д.).
- Учет движения денежных средств осуществляется через одним из учредителей, остальные присылают ему все свои расходы и доходы по смс, а он вносит их в используемую им программу для учета расходов.
- Проблемы:
  - Второй учредитель имеет не постоянный доступ к данным программы.
  - Каждый из наемных сотрудников не знает о своем текущем балансе.
  - Трудоемкий и неудобный процесс для занимающегося внесением информации в систему учета.

# ПОСТАНОВКА ЗАДАЧИ

- **Внесение данных**
  - Вносить основные операции (доход, расход, перевод), с указанием счетов, категорий, тегов.
  - Просмотр остатка денег на счетах.
  - Просмотр детализации по счетам, категориям, тегам.
  - Синхронизация данных между устройствами.
- **Многопользовательский режим**
  - Подключение к одному аккаунту нескольких пользователей.
  - Закрепление за пользователем счета.
- **Выставление уровня доступа пользователя. Выявлено три роли:**
  - Администратор – может видеть и редактировать все записи.
  - Пользователь – может просматривать все записи, но не редактировать. Может добавлять заметки (текст и фото) для последующей обработки их администратором.
  - Ограниченный пользователь – может просматривать операции только по своему счету (без редактирования). Может видеть баланс своего счета. Может добавлять заметки.

# АНАЛИЗ СУЩЕСТВУЮЩИХ ПРЕДЛОЖЕНИЙ

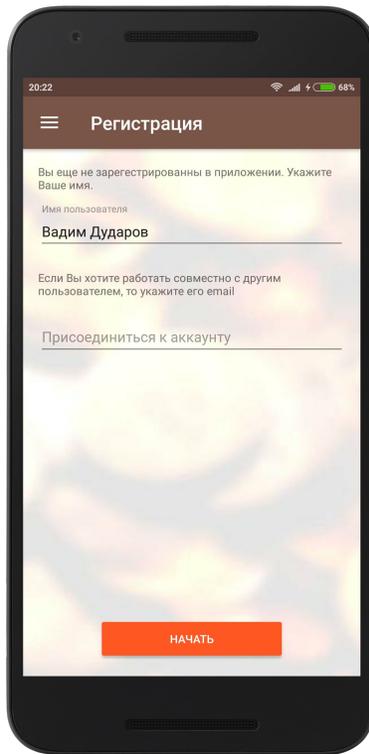
Приложение	Синхронизация	Несколько пользователей на один аккаунт	Распределение ролей между пользователями
Дребеденьги	+	-	-
TeamSpender	+	-	-
Дзен-мани	+	+	-
Monefy	+ -	-	-
Мои финансы	+	+	-
AndroMoney	+	-	-
Журнал расходов	+	-	-
Personal Finance	-	-	-
CoinKeeper	+	-	-

# UI

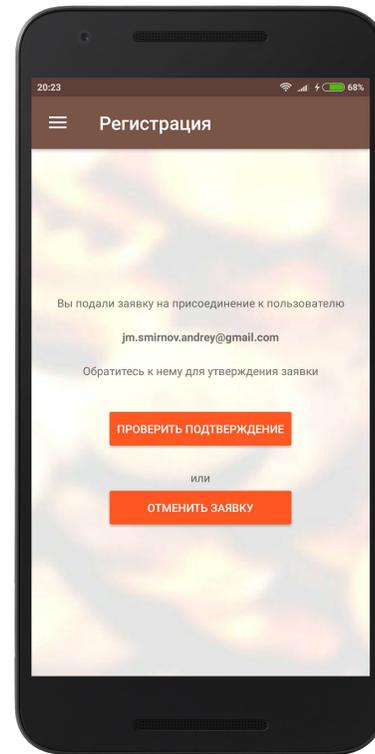
Вход в приложение  
через Google аккаунт



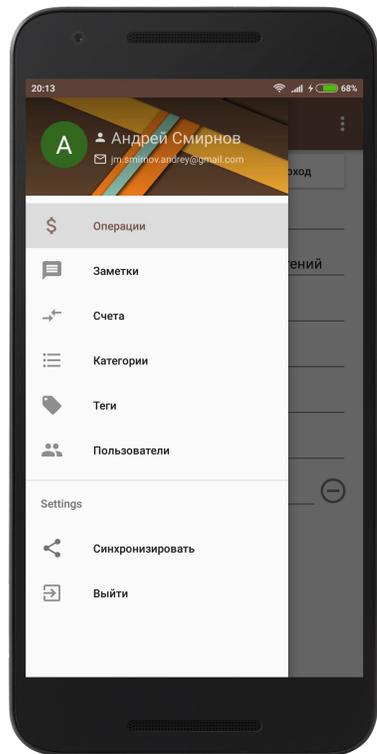
Регистрация. Возможность  
присоединиться к существующему  
аккаунту



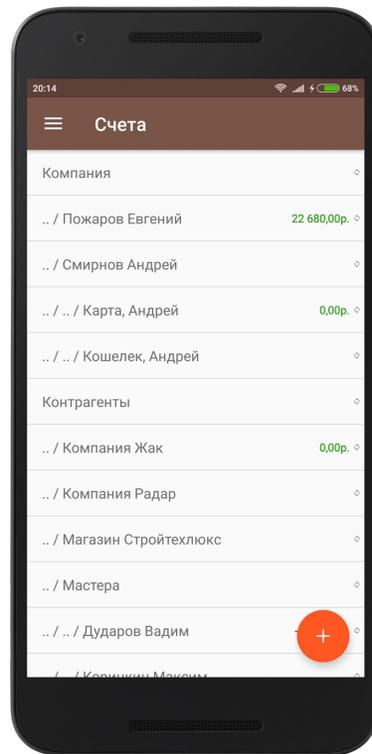
Ожидание подтверждения при  
запросе на присоединение к  
существующему аккаунту



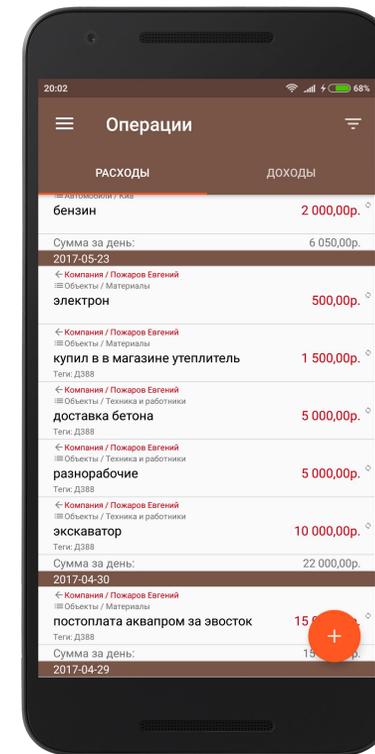
## Меню (при входе под администратором)



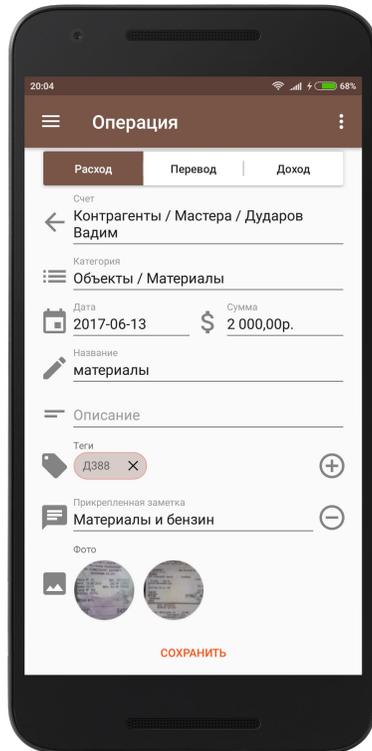
## Список счетов с балансом по каждому



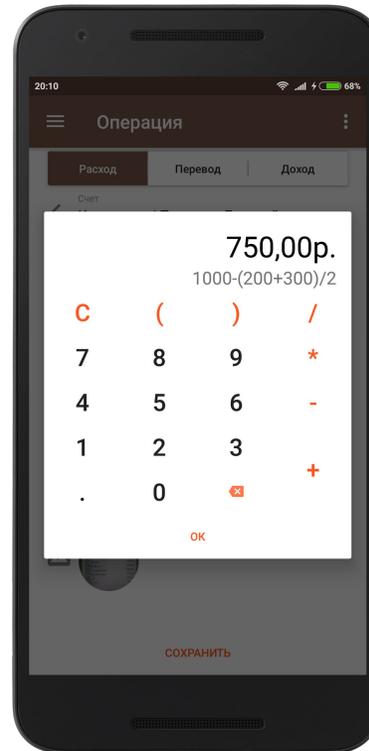
## Список операций



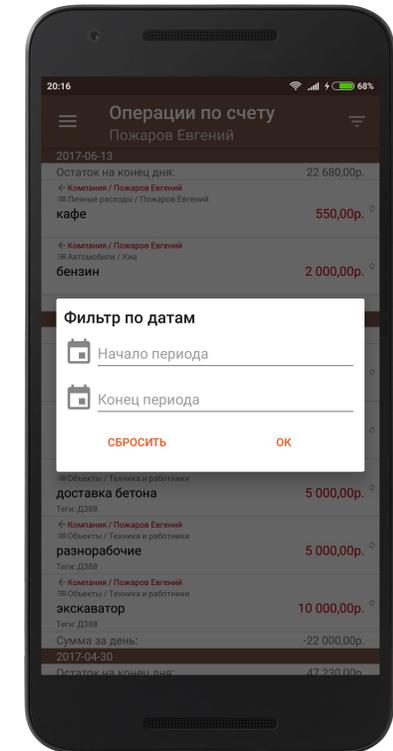
## Редактирование операции



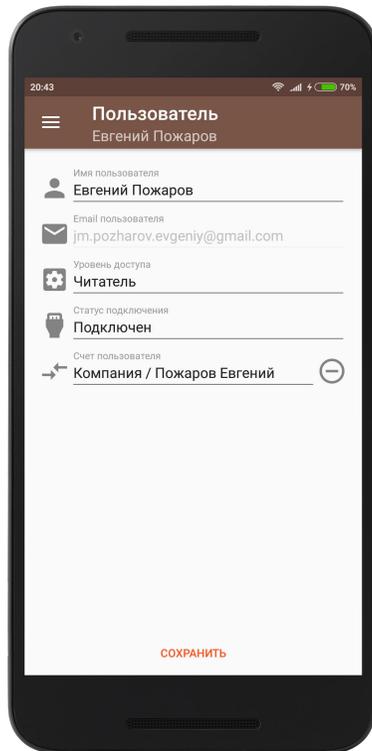
## Окно ввода суммы операции



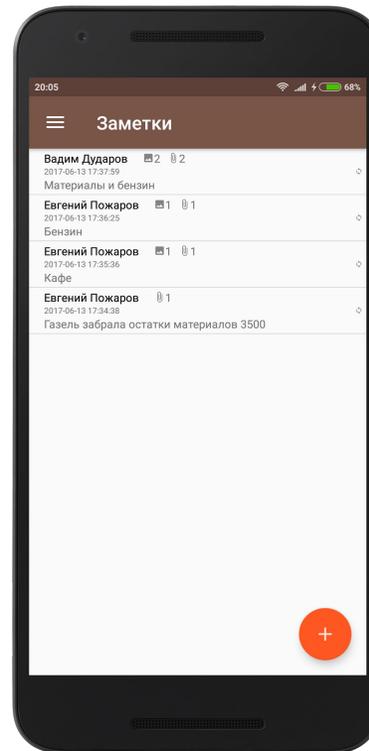
## Окно установок фильтра по датам



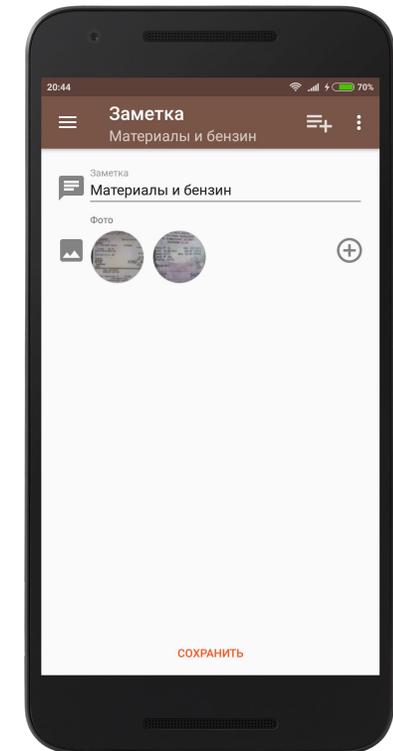
## Редактирование пользователя



## Список заметок



## Редактирование заметки

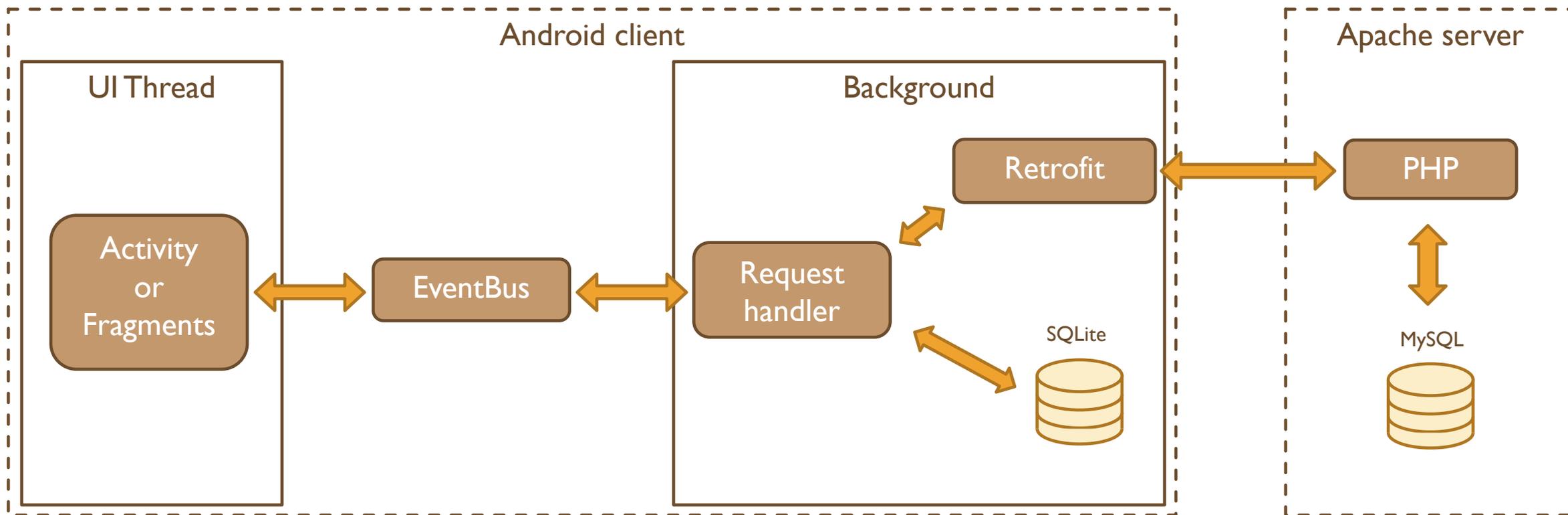




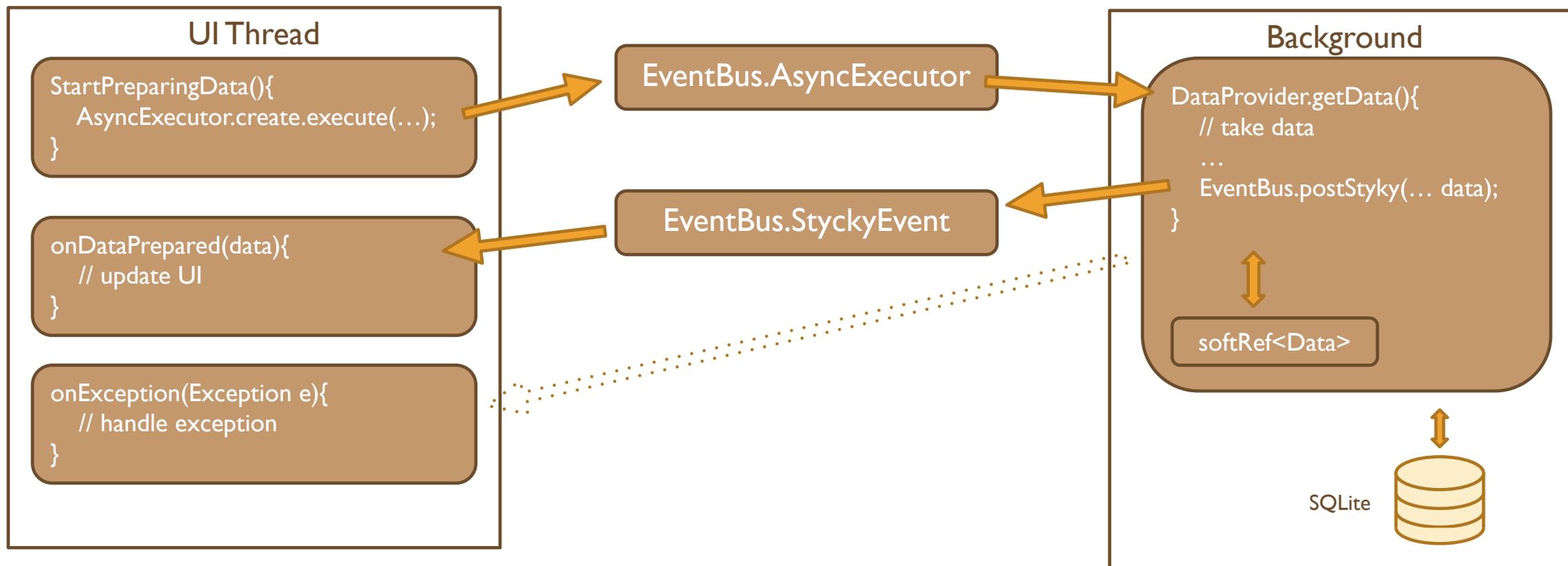
# РЕШЕНИЯ



# СХЕМА СЕТЕВЫХ ЗАПРОСОВ



# СХЕМА ЗАПРОСОВ К БАЗЕ ДАННЫХ



# РЕШЕНИЯ

## Apache + PHP + MySQL

Рассматриваемые альтернативы:

- Firebase real-time database

- + обновления в реальном времени

- + меньше кода

- NoSQL база данных. В приложении используются запросы с JOIN, который отсутствует в Firebase.

# РЕШЕНИЯ

## EventBus + Retrofit

Рассматриваемые альтернативы:

- **Retrofit (асинхронные запросы)**
  - не следит за состоянием фрагмента/активности
- **Loaders + Retrofit**
  - более сложная реализация
- **AsyncTask + Retrofit**
  - не следит за состоянием фрагмента/активности

# РЕШЕНИЯ

## Retrofit

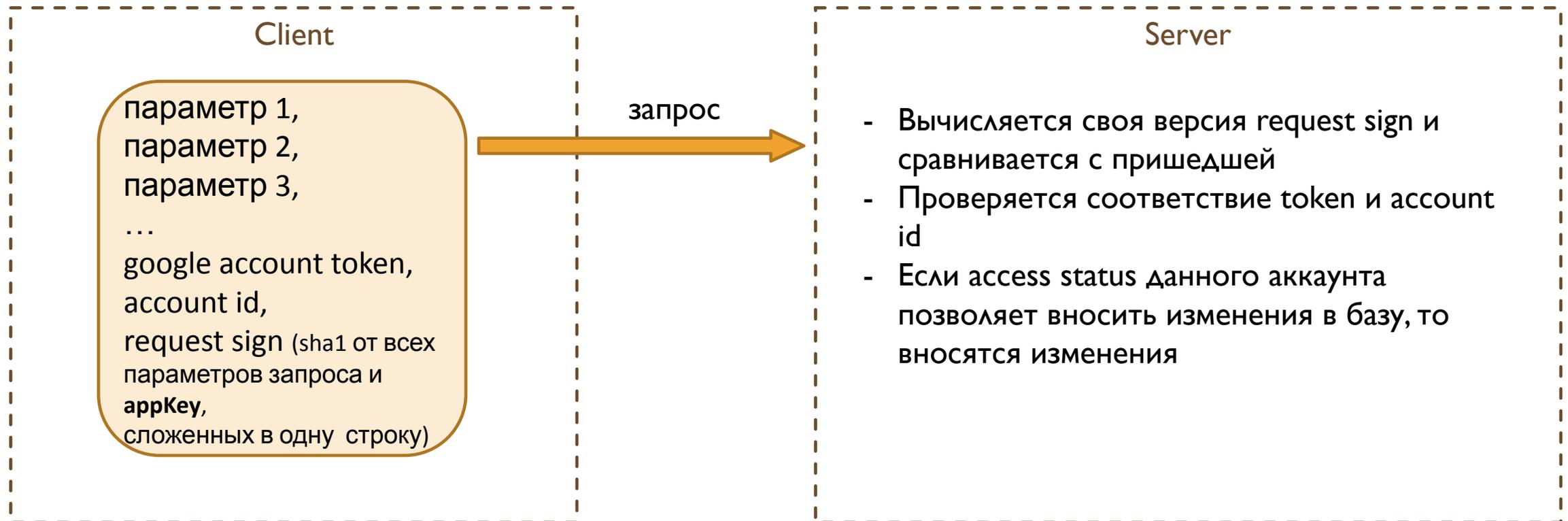
Рассматриваемые альтернативы:

- Volley
  - нет синхронных запросов, а асинхронные не отслеживают жизненный цикл фрагментов.
- HttpURLConnection
  - Нужно вручную парсить json.

# РЕШЕНИЯ

- Для кеширования запросов к sqlite базе данных, был создан класс `DataProvider`. Это singleton, через который все фрагменты запрашивают данные из базы.
- В этом классе результаты всех используемых запросов к базе сохраняются в soft reference.
- Если данные и условия запроса не были изменены, то в следующий `DataProvider` сразу возвращает данные из соответствующей soft reference (или если система их удалила, то снова идет в sqlite)
- При изменении/добавлении данных, вызываем у `DataProvider` метод обнуляющий соответствующую soft reference.

# БЕЗОПАСНОСТЬ





Спасибо за внимание!