

VotesApp (Android-клиент)

Функциональность

1. Регистрация (требуется код, который отправляется на электронный адрес)
2. Авторизация
3. Просмотр голосований
4. Возможность проголосовать
5. Возможность предложить голосование

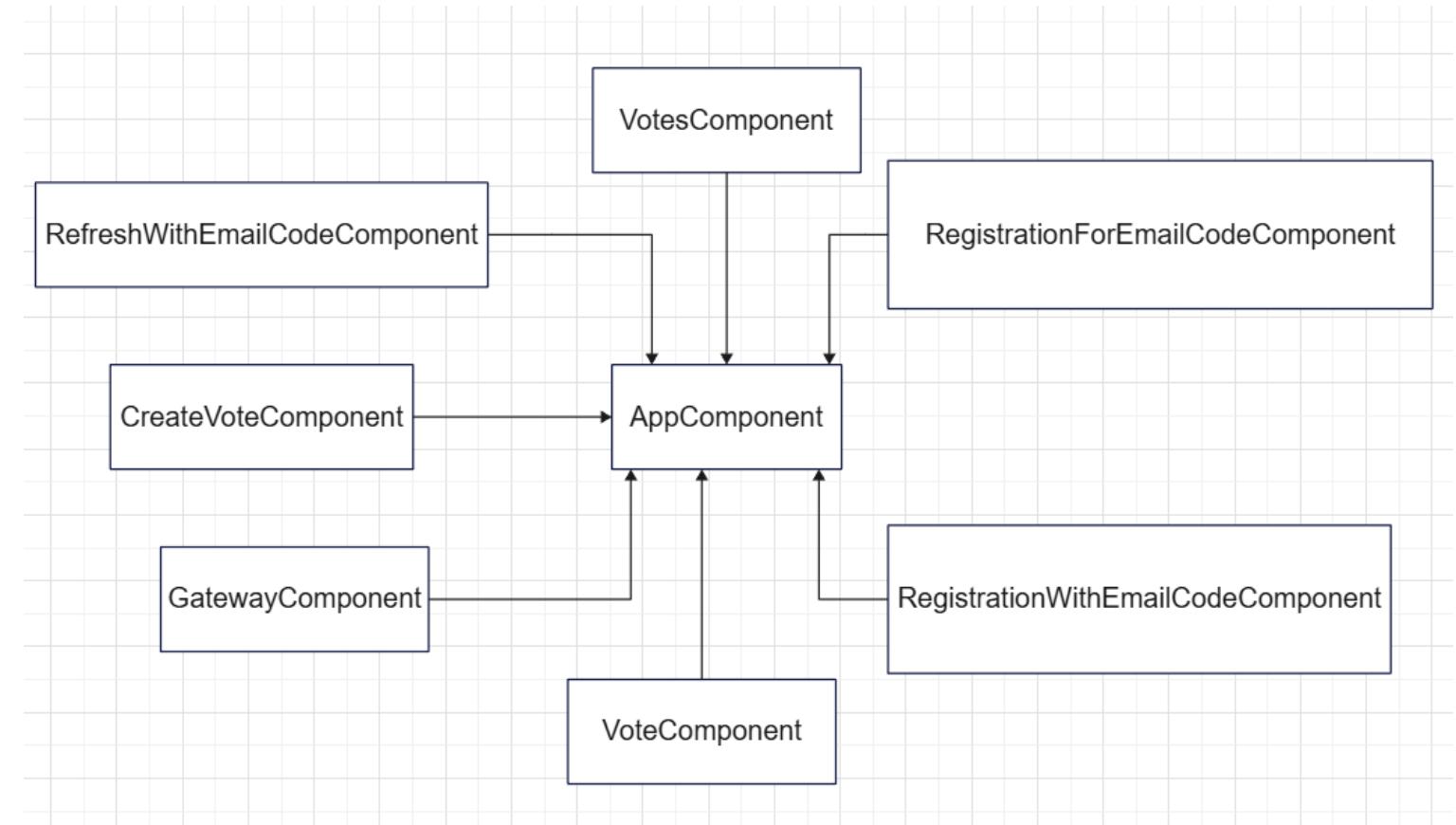


Многомодульная архитектура в VotesApp

```
> app
> common
< feature
    > create-vote
    > gateway
    > refresh-with-email-code
    > registration-for-email-code
    > registration-with-email-code
    > vote
    > votes
```

Организация DI в VotesApp (1)

```
> app
> common
< feature
  > create-vote
  > gateway
  > refresh-with-email-code
  > registration-for-email-code
  > registration-with-email-code
  > vote
  > votes
```



Организация DI в VotesApp (2)

AppComponent предоставляет другим компонентам следующие зависимости:

1. ApiService (для сетевых запросов),
2. LocalDataStorage (для хранения access и refresh токенов, а также доступа к ним),
3. Диспетчеры корутин.

Тестирование

1. Об используемом для тестирования фреймворке:
<https://developer.android.com/develop/ui/compose/testing>
2. Для тестирования используются специальные реализации интерфейсов для работы с сетью и файловой системой. Перед тестированием следует **закомментировать** аннотации **ApiServiceTestQualifier** и **LocalDataStorageTestQualifier** в **AppComponent**, а также **раскомментировать** аннотации **LocalDataStorageQualifier** и **ApiServiceQualifier** в классах **LocalStorageModule** и **NetworkModule** соответственно. Перед обычным запуском следует выполнить операцию, обратную к тому, что указано в предыдущем предложении.

Проблемы

В Android-приложении проекта VotesApp проблемы делятся на две категории:

1. Проблемы, связанные с тем, что можно улучшить само Android-приложение при неизменности внешнего API, предоставляемого серверной частью;
2. Проблемы, связанные с ограниченной функциональностью, обусловленные возможностями серверной части.

Проблема 1

Для некоторых сетевых запросов нужен access токен. Каждый раз он достаётся из файловой системы, что может замедлять получение результата. Для ускорения можно хранить актуальное значение токена в оперативной памяти.

```
override fun getVotesFlow(): Flow<VotesState> {
    return flow {
        emission.collect {
            try {
                emit( value = VotesState.Loading)
                emit(
                    value = VotesState.Success(
                        votes = apiService.getVotes(
                            page = 1,
                            token = localDataStorage.readAccessTokenPreferences()?.token
                                ?: throw RuntimeException( message = "Access token must be non-nullble!")
                        ).titles
                    )
                )
            } catch (_: Exception) {
                emit( value = VotesState.Failure)
            }
        }
    }
}
```

Проблема 2

Поддерживается только одно устройство для аккаунта. Одно из решений проблемы состоит в том, чтобы допустить хранение нескольких (но не более некоторого количества) refresh токенов для одного пользователя на стороне сервера.

```
@Transactional
fun refreshToken(refreshTokenWithBearer: String): RefreshTokenResult {
    val currentRefreshToken = refreshTokenWithBearer.extractToken()
    val username = jwtGenerator.extractUsername(currentRefreshToken)
    val encoded =
        refreshTokenRepository.findByUsername(username) ?: return RefreshTokenResult.RefreshTokenNotFound
    if (!passwordEncoder.matches(currentRefreshToken, encoded.refreshToken)) {
        return RefreshTokenResult.IncorrectRefreshToken
    }
    val (token, refreshToken) = generateTokens(username)
    refreshTokenRepository.deleteByUsername(username)
    refreshTokenRepository.save(
        RefreshToken(
            refreshToken = passwordEncoder.encode(refreshToken),
            username = username
        )
    )
    return RefreshTokenResult.Success(
        token = token,
        refreshToken = refreshToken
    )
}
```

Проблема 3

Имеется ряд ограничений функциональности, в том числе:

1. Нельзя отменить голос,
2. Нельзя выбрать несколько вариантов в голосовании,
3. Нет личного кабинета пользователя, в котором могла бы появляться информация о принятии или об отклонении его голосования;
4. Нет возможности выйти, удалив на стороне сервера refresh токен;
5. Нет возможности удалить аккаунт.

Проблема 4

Пользователь может два раза предложить одно и то же голосование. Тогда, если голосование будет принято, при попытке получить информацию о голосовании произойдёт аварийная остановка работы Android-приложения, потому что будет нарушен unique constraint для ключа LazyColumn. Вероятно, для решения проблемы достаточно добавить unique constraint для пары (title, variant) на стороне сервера.

```
items(count = vote.size, key = { vote[it].variant }) { index ->
```

```
create table if not exists votes (
    id bigserial primary key,
    title varchar(512) not null,
    variant varchar(128) not null
);
```