Цифровая олимпиада «Волга–IT`20»

Дисциплина «Мобильная разработка (iOS, Android, Flutter)»
Задание отборочного этапа

Задание

Написать мобильный клиент для гитхаба.
Приложение должно загружать данные с API github.com (документация https://developer.github.com/v3)

1. Задание выполняется последовательно, каждый следующий шаг увеличивает функционал на основе предыдущих шагов.
2. Чем больше пунктов будет сделано, тем выше оценка.
3. Дизайн всех элементов на свое усмотрение, чем презентабельнее результат, тем лучше. Можно придерживаться гайдов Material Design или Apple Human Interface Guidelines
4. Чистота и простота кода важна в первую очередь
5. Если присланное приложение не будет компилироваться и запускаться на актуальных на момент сдачи версиях IDE и фреймворков - все пункты считаются не выполненными
6. Отдельным плюсом будет предоставление результата в качестве ссылки на git репозиторий. Каждый шаг выполнения желательно оформлять в отдельный коммит
7. Багов и крашей в приложении быть не должно.

ВАЖНО! Количество неавторизованных запросов к API Github ограничено 60-ю в час (https://developer.github.com/v3/#rate-limiting), после привышения на любой запрос с того же IP будет возвращаться ошибка. Самый простой способ это обойти - добавить авторизацию через Basic Auth. Для этого в начале каждого URL нужно добавить логин и пароль, например вместо

- https://api.github.com/orgs/github/repos

будет

- https://YOUR_USERNAME:YOUR_USER_PASSWORD@api.github.com/orgs/github/repos

В качестве плюса будет засчитываться использование более продвинутых и безопасных способов авторизации

1) Главный экран приложения - список репозиториев запрошенных по урлу https://api.github.com/repositories, для получения дополнительной информации о репозитории нужно делать отдельный запрос для каждого репозитория на урл https://api.github.com/repos/<REPO_full_name>
Каждый репозиторий должен содержать:
- название
- описание (должно отображаться полностью, ячейки должны растягиваться соответствующим образом)
- язык программирования, если есть
- количество форков
- количество звезд
- имя автора и фото (должно быть круглым)

2) Открытие детальной информации о репозитории
По нажатию на элемент списка должен открываться отдельный экран с детальной информацией о репозитории
Можно отобразить всю информацию, которую получится достать.
Обязательно должны отображаться:
- Название
- Информация об авторе (имя и аватар)
- Последние 10 коммитов
    - описание (сообщение) коммита

- дата
- имя и аватар автора

3) Pull-to-refresh на экране списка репозиториев
Возможность перезагрузить список за счет использования контрола pull-to-refresh
разрешается использовать стороннюю либу

4) Бесконечный скролл на экране списка репозиториев
Найти метод в апи для получения данных пачками (pagination) и реализовать возможность дозагружать следующую пачку данных в момент когда мы доскролили до низа списка репозиториев
Загрузка большого количества элементов не должна влиять на производительность приложения

5) Сделать 2 список репозиториев
Реализовать второй экран с тем же интерфейсом (использовать те же UI компоненты, которые были реализованы на первых шагах) и наполнить другими данными (сделать запрос репозиториев с url https://api.github.com/orgs/github/repos)
Переключение между экранами должно быть сделано с помощью табов внизу экрана

6) Сделать сохранение репозиториев в избранное
На первом экране со списком репозиториев сделать возможность добавить репозиторий в избранное (сохранить в локальном хранилище - на свое усмотрение).
На втором экране со списком репозиториев отображать репозитории загруженные из локального хранилища (должно работать без доступа в интернет)
Репозиторий должен добавляться и удаляться из избранного.
Элемент списка должен сдвигаться вбок отображая кнопку с добавлением/удалением в избранное (по аналогии с https://i.stack.imgur.com/hXG0t.png)

7) Добавить анимацию загрузки данных.
В момент когда мы попадаем на экраны списка репозиториев или детальной страницы и данные еще не загрузились - отображать экран с анимированным элементом - на свой выбор