

Документация по работе над приложением EventHub

Основная цель: сделать iOS-приложение, отвечающее требованиям задания, в установленные сроки.

Важные примечания:

- задание определено на платформе DevRush, сроки на разработку: 17 - 30.11.2024;
- UI должен быть адаптивным ко всем моделям iPhone;
- локализация приложения: английский язык либо русский язык, если он является системным на устройстве пользователя;
- поддержку светлой и темной темы сделаем, если в пределах установленных сроков останется время после выполнения требований задания;
- код должен предусматривать обработку ошибок;
- при взаимодействии с сетью стремимся получать минимально достаточные данные.

Взаимодействие в команде:

- общение, обмен референсами, ссылками на обучающие материалы проводим в группе команды в Telegram, слияние кода – в репозитории на GitHub, взаимодействие с системой контроля версий – с помощью git-клиента Fork, планирование и отражение выполнения задач в рамках работы команды над приложением – на доске в GitHub Projects;
- на регулярной основе проводим общекомандные встречи в видеочате группы в Telegram (анонсы даты, времени и вопросов к обсуждению делает тимлидер), каждому в команде предоставлена возможность начинать видеочат для проведения парных кодинг-сессий, решения индивидуальных и общих проблем в разработке;
- если в ходе разработки возникают сложности, которые не удалось решить самостоятельно, сразу же спрашиваем в группе команды в Telegram. Знаем, как помочь другому, - помогаем, потому что потенциал команды значительно больше потенциала каждого по отдельности. Не нужно переходить в личную переписку. Если возникает объективная невозможность выполнить задачу в срок, сообщаем тимлидеру, не пропадаем.

Работа с репозиторием GitHub:

Нейминг веток, создаваемых членами команды:

- nick/feature, например: marat/mainView (где marat – ник, mainView – название разрабатываемой функциональности, соответствующей одной задаче на доске или несколькими взаимосвязанным задачам);

Нейминг коммитов:

- [FEAT]: some descriptive, например: [FEAT]: add main view and components (где [FEAT] – категория коммита, add main view and components – краткое описание выполненной работы. Желательно писать на английском, но допустимо и на русском, главное, чтобы получалось кратко, но понятно, при этом не только для себя, но и для остальных. Подробности можно указать в описании к коммиту);

Категории коммитов:

- [FEAT] – создание какой-то определенной функциональности: нового компонента, добавление логики (к этой категории относится подавляющее большинство коммитов);
- [FIX] – фикс багов;
- [REFACTORING] – рефакторинг кода, не влияющий на логику в целом. Форматирование кода, избавление сору/paste, подчистка или написание комментариев;
- [CHORE] – добавление ReadMe, иконки и т.п.

Коммитим только код, который собирается!

Перед созданием ветки стягиваем себе актуальный develop (нужно выполнить pull)

Чтобы код верстки был читаемым, разбиваем его на компоненты. Компоненты, которые используются на других экранах в приложении, лучше выносить в отдельный файл, а не просто копировать код. Если компонент используется только на данном экране, его логично оставить здесь же. Однако, если компонент очень объемный и есть желание разгрузить файл для лучшей читаемости, можно вынести и его.

По гиту нам нужно уметь:

- **Clone** - клонировать проект на компьютер;
- **Branch** - создать ветку;
- **Stage** - застейджить только необходимые изменения;
- **Commit** – закоммитить;
- **Commit --amend** - закоммитить, переписав предыдущий коммит;
- **Push** - отправить ваши сохраненные изменения на удаленную ветку;
- **Pull Request** - сделать пулл-реквест;
- **Merge** - слить ветки;
- **Pull** - стянуть актуальное состояние ветки себе.

Опционально:

- **Stash**;
- **Rebase**;
- **Cherry picking**.

Защита веток main и develop:

Слияние кода, написанного каждым в своей ветке, будет происходить в ветке origin/develop. Ветка origin/main предназначена только для релизных версий кода. По ходу выполнения текущих задач в origin/main ничего не добавляем! В нее будем переносить

готовый результат из develop на самом финише. Ветка origin/develop – является веткой по умолчанию, если иное специально не определено тимлидером в группе в Telegram.

На ветке origin/develop установлена защита, которая требует:

- создать pull request каждый раз перед тем, как смёрджить в origin/develop (напрямую смёрджить не получится);
- во время создания pull request-а указать других членов команды, чтобы получить review и approval кода (можно всех, можно только некоторых);
- чтобы открылась возможность смёрджить, надо получить как минимум 1 approval от членов команды (от любого из указанных вами при создании pull request-а).

Сразу, как создали pull request, пишем об этом в чат группы в Telegram, чтобы привлечь внимание остальных на то, что нужно посмотреть и утвердить этот код. Это сократит время ожидания вами и достижения командой общего результата.

Сразу, как обновили origin/develop, пишем об этом в чат группы в Telegram, чтобы остальные смогли себе подкачать актуальную версию develop. Это уменьшит вероятность мёрдж-конфликтов при последующих слияниях.

Мы стремимся сообща выполнить и базовую, и продвинутую части задания, а также, если в пределах установленных сроков останется время, сделать приложение более удобным, а код – качественным.