

# ОПЫТ СОЗДАНИЯ ПЕРВООЧЕРЕДНЫХ ФУНКЦИЙ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ СЭД ТПУ

В.В. Леухин, А.А. Цибульников  
Научный руководитель Чудинов И.Л.  
Томский политехнический университет  
vv118@tpu.ru

## Введение

Целесообразность использования мобильных устройств участниками систем электронного документооборота (СЭД) в настоящее время не вызывает сомнений. Очевидно также и то, что создание мобильных приложений для СЭД связано с трудностями, обусловленными следующими причинами:

- Большое число процедур (функций) которые желательно реализовать в мобильном приложении
- Разнообразие операционных систем и ограничения имеющихся у них инструментальных средств программирования приложений
- Необходимость интеграции мобильного приложения с программным обеспечением СЭД

Обоснование состава функций, которые целесообразно и возможно реализовать в мобильном приложении СЭД Томского политехнического университета и инструментария программирования мобильных приложений (фреймворков) для двух наиболее распространенных операционных систем Android и IOS приведено в представленном на настоящей конференции докладе «Выбор программных средств для создания мобильного приложения в системе электронного документооборота» [1].

## Особенности разработки кроссплатформенного мобильного приложения

Для разработки кроссплатформенного приложения был выбран фреймворк React Native.

Так как приложение разрабатывается на две ОС Android и IOS, то одним из критериев в реализации графического интерфейса является максимальное соблюдение дизайна нативного интерфейса каждой ОС.

Для удобства и скорости разработки графического интерфейса была использована сторонняя библиотека NativeBase [2]. Хотя данная библиотека и существенно расширяет стандартный список графических компонентов, представленных самим фреймворком, то как правило этого недостаточно для специфических задач, поэтому и не обходится без написания собственных компонентов.

Основным достоинством этой графической библиотеки является соблюдение нативного дизайна компонентов в каждой ОС.

Один и тот же графический компонент отображается по-разному согласно дизайну той или иной ОС. Поэтому, разрабатываемое приложение использует свойства нативного дизайна и максимально гармонирует с ОС, создавая впечатление

законченности (качества проработывания дизайна).

Хоть кроссплатформенная разработка подразумевает, что один и тот же написанный код при помощи интерпретаторов используется в разных ОС, случаются и свои трудности. Основные трудности, встречаемые при разработке графического интерфейса в кроссплатформенном приложении, заключаются в некорректном отображении на экране компонентов или в отсутствии универсальных функций выполняющее одно и то же действие на разных ОС.

Для устранения данной проблемы было принято решение создавать отдельно под каждую ОС, те блоки кода, которые выполняются некорректно.

## Результаты реализации мобильного приложения.

После запуска приложения отображается главный экран (рисунок 1) приложения, на котором в нижнем меню доступны следующие действия:

- Переход на экран профиля пользователя
- Открыть боковое меню «Deawer» приложения.

Боковое меню «Deawer» (рисунок 2) приложения можно открыть с любого экрана, если потянуть его с левой стороны экрана вправо.

На экране профиля (рисунок 3) пользователя можно перейти на экран авторизации (кнопка «выход»).

На экране «Документы» (рисунок 4) можно перейти на экран быстрого выбора документов (кнопка меню), либо листать вкладки или экраны, смещая вправо и влево.

Так же доступно нижнее меню, с которого можно вернуться на главный экран, открыть боковое меню или зайти в профиль.

На экране поручения (рисунок 5) можно выбрать два типа поручений:

- Я поручил
- Мне поручили

В зависимости от типа поручений изменяется количество отображаемых вкладок и отображаемая информация.



Рис. 1. Главный экран

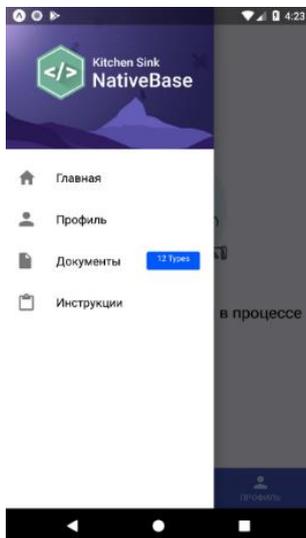


Рис. 2. Компонент Drawer

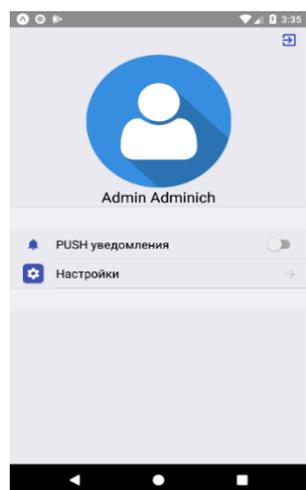


Рис. 3. Экран профиля пользователя

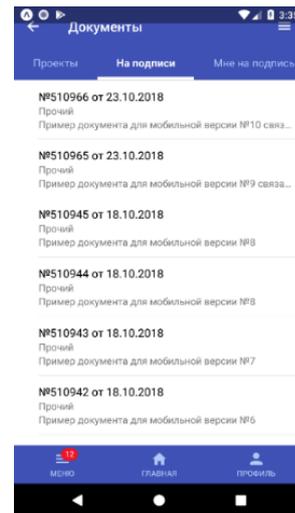


Рис. 4. Документы на подпись



Рис. 5. Экран инструкций

### Заключение

В результате проделанной работы был реализован основной «скелет» графического интерфейса реализуемого приложения состоящих из 5 главных экранов, хранящих в себе большое множество компонентов. А также написана библиотека, отвечающая для обращения на сервер, получения и обработка ответа с сервера. Данное приложение было протестировано на двух ОС для исправления ошибок в отображении и функционировании графического интерфейса.

### Список использованных источников

1. NativeBase// [Электронный ресурс]. – URL: <https://docs.nativebase.io>. (Дата обращения 20.09.2018).