Android-приложение “Bitcoin-кошелёк”

Описание задачи

Разработайте минималистичное Android-приложение на языке Kotlin для сети Bitcoin Testnet, с помощью которого пользователь сможет отправить введенную сумму биткоинов на указанный биткоин-адрес.

Основной интерфейс приложения должен состоять как минимум из следующих элементов:

1) Заголовок экрана - Bitcoin wallet;
2) Подзаголовок - Balance. Текущий баланс адреса, установленного в качестве **источника** биткоинов для отправки на указанный пользователем адрес. На этот адрес должна быть заранее переведена необходимая для тестов сумма биткоинов;
3) Поле ввода - Amount to send. Сумму в биткоинах, которую необходимо отправить пользователю.
4) Поле ввода - Address to send. Биткоин-адрес, на который пользователю необходимо отправить биткоины.
5) Кнопка - Send. По тапу на кнопку Send формируется и отправляется биткоин-транзакция.

Оформление приложения
Интерфейс приложения должен быть аккуратно сверстан в соответствии с Material Design 2 или Material Design 3.

После успешной отправки средств необходимо любым способом выводить сообщение об успехе содержащее следующие элементы:

1) Заголовок - Your funds have been sent!
2) Текст - Your transaction ID is af8ee4b868fee…59d469a9b4. По клику на ID транзакции должен открыться web-браузер и осуществлен переход на страницу биткоин-транзакции в обозревателе блоков https://blockstream.info/.
3) Кнопка - Send more. По тапу на кнопку Send more пользователь должен быть возвращен на основной экран приложения с актуализированным балансом кошелька.

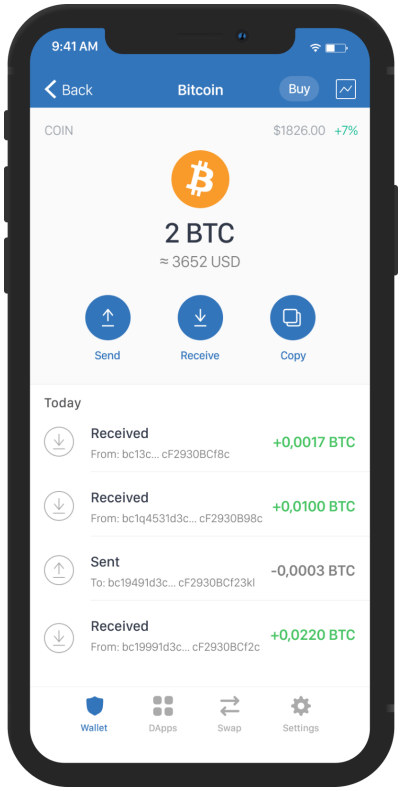
В случае неуспеха пользователю должна выводиться соответствующая ошибка. Обработка ошибок не регламентируется и реализуется на усмотрение кандидата.

Перед отправкой средств приложение должно проверять, достаточно ли средств для отправки (включая комиссию майнерам).

Комиссия майнерам должна составлять 0.000001 **tBTC**.

Опциональные задания на бонусные баллы (можно выполнить любое количество из нижеперечисленных пунктов):

- Рассчитывайте комиссию майнерами пропорционально размеру (в байтах) получившейся транзакции (майнеры устанавливают стоимость обработки за один байт транзакции - чем больше в байтах размер транзакции, тем больше стоимость транзакции)
- Реализуйте отображение истории отправленных и полученных транзакций. Подразумевается отображение транзакций в виде списка: Индикация получения либо отправки, ID транзакции, время, сумма в tBTC.
  Пример такой реализации:



Внешние ресурсы: “необходимо и достаточно”

Если понадобятся тестовые BTC, то их можно бесплатно получить набрав в Google “Bitcoin testnet faucets”.

При возникновении сложностей с получением tBTC следует сообщить об этом и мы пришлём 0.05 **tBTC** на указанный вами адрес.

Для тестирования транзакций в качестве **Bitcoin кошелька для сети Bitcoin Testnet** рекомендуем использовать Electrum, запустив его с флагом --testnet:
https://electrum.org/#download

Для **формирования BTC транзакции** используйте библиотеку bitcoinj.
Она предоставляет **класс Transaction** для работы с транзакциями, а также классы работы с приватными ключами (ECKey/DumpedPrivateKey).

С деталями создания биткоин-транзакции можно ознакомиться в разделе документации Working with transactions.

Входы транзакции можно сформировать с помощью **метода Transaction.addSignedInput()**, используя **классы TransactionOutPoint** и **Script**.
Выходы транзакции можно сформировать с помощью **метода Transaction.addOutput()**.

Для выполнения задания класс **не следует использовать WalletAppKit** т.к предполагается самостоятельное создание ключа, формирование и подписание транзакций средствами bitcoinj, а также получение непотраченных выходов транзакций (UTXO) через веб-сервис.

Для исследования hex транзакций можно использовать
https://live.blockcypher.com/btc/decodetx/.

Для **бродкаста транзакций** (и получения данных по транзакциям, уже включенным в блокчейн) используйте Blockstream:
https://github.com/Blockstream/esplora/blob/master/API.md

Подсказки:

- Посмотрите, что такое UTXO (unspend transaction outputs). Проще всего сделать так, чтобы каждая очередная транзакция в качестве входов использовала *все* UTXO, соответствующие вашему адресу обменника.
- Адрес разрабатываемого “кошелька” в любой момент может быть пополнен из другого кошелька. Транзакция пополнения в качестве *одного* из выходов будет содержать адрес, сгенерированный скриптом (другие выходы - “сдача”). Для создания расходной транзакции вам потребуется определить, какой выход каждой из транзакций пополнения ведёт к вашему адресу обменника (см. документацию по ссылкам выше). В API blockstream есть способ получить выходные адреса bitcoin-транзакции.

Ликбез по блокчейну (опуская детали)

Блокчейн состоит из цепочки транзакций.

Для совершения операции “перевода” с одного адреса на другой создаётся транзакция, “входом” (input) которой является первый адрес, а “выходом” второй. Транзакция может содержать несколько входов и несколько выходов.

Разница (в сумме “монет”) между всеми выходами и всеми входами определяет комиссию майнеров; если комиссия нулевая, транзакция не будет обработана майнерами - стоит оставить им некоторую плату.

Для того, чтобы потратить только часть “монет” с некоторого адреса, в качестве дополнительного выхода транзакции устанавливается адрес, находящийся под контролем отправителя. Иначе говоря, “сдача” с операции в явном виде должна быть перечислена себе.