Programátorská dokumentace

Úvod

Aplikace bitcoinová peněženka slouží pro správu deterministicky vygenerovaných privátních klíčů z ‘Master key’, neboli seedu, které jsou přiřazeny k vytvořenému účtu. Aplikace umožňuje jak zobrazení jednotlivých mincí, příjem mincí, zobrazení **LEGACY** adres, tak i jejich odeslání na jinou adresu. Peněženku lze jak importovat/exportovat pomocí 12ti slovného seedu nebo read-only pomocí Xpub, tak i vytvořit novou. Transakce jsou do sítě odesílány za pomocí RPC klienta běžícím na full-nodu.

Frameworky

.NET Framework
CommunityToolkit.MVVM
AvaloniaUI

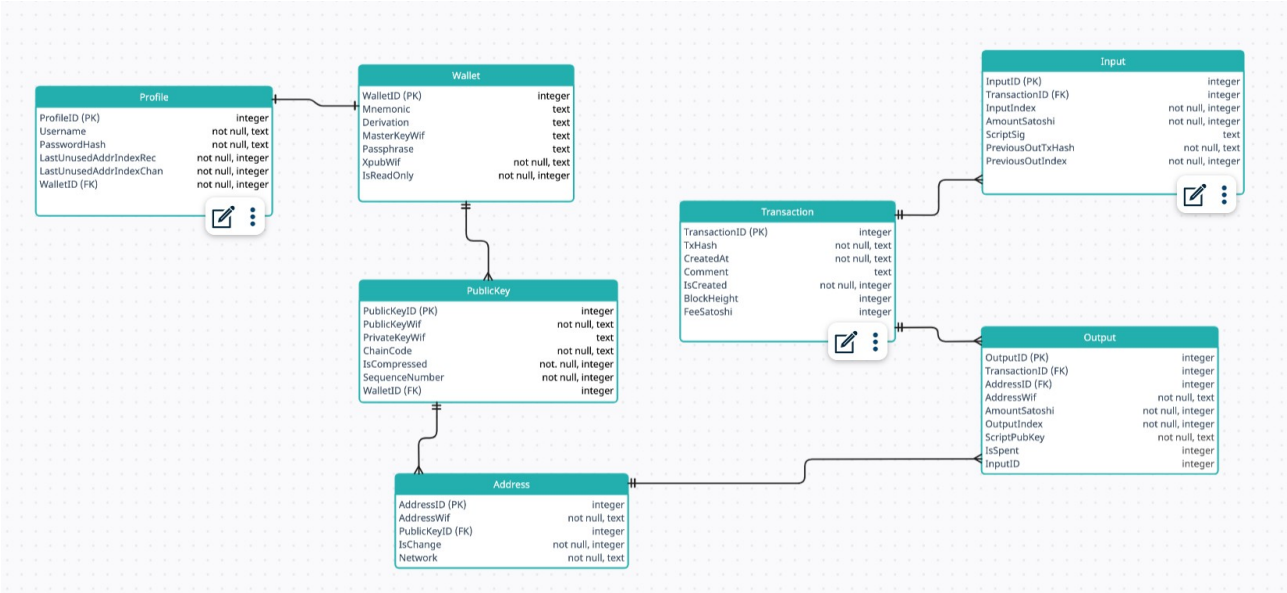
Knihovny

NBitcoin
QRCoder
EFCore – SQLite

Architektura

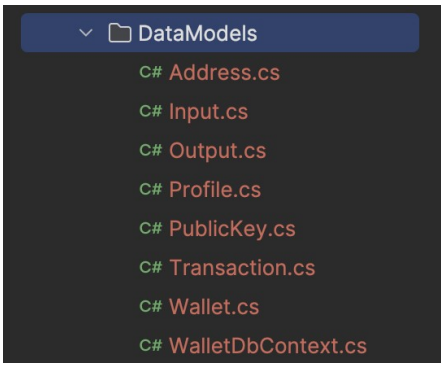
Doporučuji se seznámit s UI, viz uživatelská dokumentace. Jednotlivé metody obsluhující uživatelské rozhraní jsou uloženy ve ViewModels záložce, kde je přiměřená komentářová dokumentace.

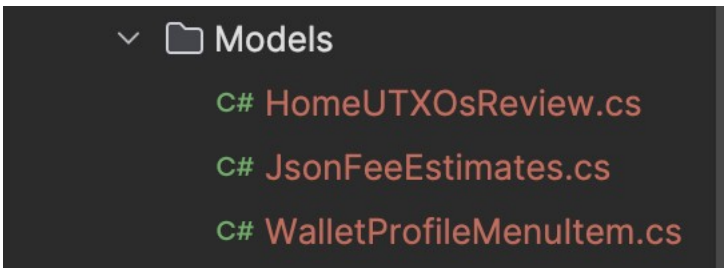
Jedná se o aplikaci napsanou pomocí AvaloniaUI frameworku, který zabezpečuje interakci s uživatelským rozhraním aplikace. Byl použita architektura MVVM (Model-View-ViewModel), která odděluje aplikační grafické rozhraní od jednotlivých fukcí a bussiness logiky pomocí data bindings a SQLite databáze jako modelu, viz Image1.



Modely

Modely aplikace se rozdělují do dvou skupin, datových (scaffold sqlite databáze za pomocí efcore balíčku, viz Image2) a prezentačních (tabulková prezentace dat pro uživatele, viz Image3)





Data Models.* slouží jako API mezi SQLite databází WalletDb.db a ViewModely aplikace.

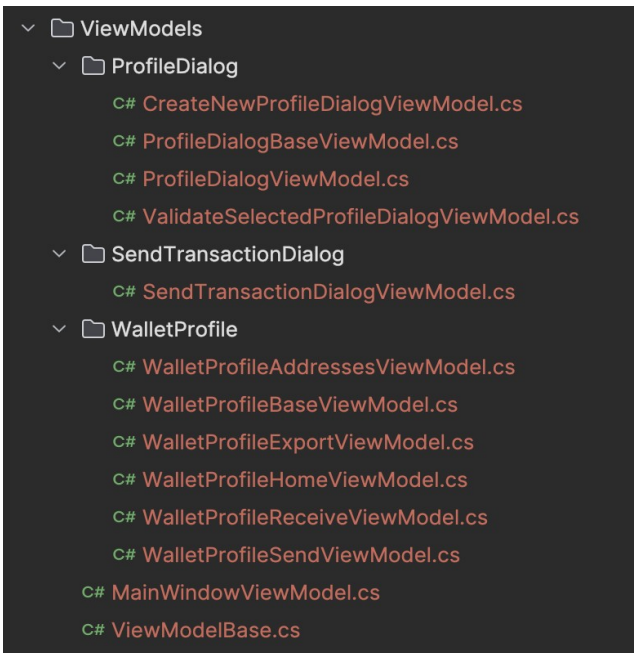
Models.HomeUTXOsReview slouží pro prezntaci jednotlivých mincí, které jsou uloženy v ObservableCollection, která bere data z SQLite databáze, viz WalletProfileHomeViewModel. Data pak jsou ve View prezentovány DataGridem s nastaveným bindigem na ItemSource.

JsonFeeEstimates je API pro data z uri "https://bitcoiner.live/api/fees/estimates/latest", která přijdou v podobě JSONu, a jsou deserializována do tohoto modelu.

WalletProfileMenuItem slouží pro prezentaci položek menu, která jsou uchována v ObservableCollection WalletProfileMenuItemsList, která má uložené obsahy menu pro danou záložku. Data pak jso prezentována ListBoxem, viz WalletProfileBaseView.

ViewModely a Views

Views a ViewModels jsou spolu úzce spjaty, každý View má i svůj ViewModel s výjimkou ProfileDialogBaseViewModel a WalletProfileBaseViewModel a ViewModelBase, které shrnují obecné metody a přístupy pro následné ViewModely v hierarchii, viz Image4.



ProfileDialog
ProfileDialog se řeší přihlášení či vytvoření profilu-peněženky pro uživatele.

ProfileDialogBaseViewModel implementuje otevření dialogového okna pro otevření profilu či tvorby nového a také implementuje funkci SHA256, pomocí níž se ukládají hesla do databáze.

ProfileDialogViewModel funguje jako rozcestník mezi vytvořením nového profilu a otevřením již existujícího, který je vyzobrazen v listboxu. Implementuje tedy za pomoci CommunityToolkitu.MVVM základní interakci mezi uživatelem a UI aplikace a validátory.

ProfileDialogValidateSelectedViewModel reprezentuje validátor vloženého hesla pro vybraný existující profil v sqlite databázi. Uživatel je připuštěn dále, pokud zadá správné heslo.

ProfileDialogCreateNewViewModel implementuje vytvoření či import peněženky pro nově vytvořený profil. Tento ViewModel úzce spolupracuje s vázaným View, pomocí data bindings, kdy se povoluje a zakazuje zápis do určitých polí dle vybrané konfiugrace import/export, default/readonly. Uživatel je připuštěn dále až po automatickém zvalidování vybraných polí viz uživatelská dokumentace. **Peněženka je vytvořena/importována s LEGACY P2PKH adresami**. Po procesu vytvoření profilu je v databázi uložena jak peněženka tak profil k ní vázaný.

Validním zadáním údajů se přidá do MainWindowViewModel _tabs nová položka, která je dostupná pro uživatele.

WalletProfile
WalletProfile se zabývá interakcí uživatele s konkrétní peněženkou uložnou v databázi.

WalletProfileBaseViewModel abstrahuje kontent momentálně otevřeného obsahu vybrané záložky v menu. Zároveň implementuje otevření ProfileDialogu popsaného výše pro otevření nové záložky profilu, či zavření momentální záložky.

WalletProfileHomeViewModel je základní obrazovka pro vybranou peněženku, kde se nachází souhrn dat, mincí a jejich refresh z bitcoinvé sítě za pomoci RPC klienta. ViewModel implementuje tři hlavní fuknce tlačítek **RefreshUsingRPCAsync** (komunikace s RPC ohledně aktuálně držených UTXOs na adresách patřících peněžence a jejich zobrazení v listboxu) **OpenNewExportWindow** (otevření okna zobrazující data pro export aktuální peněženky) a metody ohledně získání dat kurzů z uri „https://api.coingate.com/v2/rates/merchant/BTC/USD“.

WalletProfileExportViewModel je obrazovka prezentující databázová data pro export peněženky (seed, derivation path, xpub, passphrase).

WalletProfileAddressViewModel je obsah prezentující adresy patřící aktuální peněžence. Data jsou pomocí LINQu a EFCore vrácena z databáze.

WalletProfileReceiveViewModel slouží pro příjem mincí na zobrazenou adresu, kterou uživatel může měnit za pomocí tlačítka s nímž vygeneruje novou adresu, aktualizuje v databázi. Zároveň je k dané adrese generován QR kód, který zapouzdřuje kromě adresy i vloženou částku a zprávu měnící se instantně.

WalletProfileSendViewModel implementuje modifikovatelnou tvorbu transakce (**NENÍ ZOBRAZOVANÉ PRO READ-ONLY PENĚŽENKY, viz WalletProfileBaseViewModel**) na **1 zadanou adresu**. Aplikace v ihned validuje zadané vstupy a dává uživateli zpětnou vazbu za pomoci proměnných IsError a IsLoadingErrorRPC. Zároveň se při inicializaci asynchronně stahuje

z uri "https://bitcoiner.live/api/fees/estimates/latest" nápověda - hodnota velikosti poplatků těžařům. Uživatel si také může vybrat mezi minimálním množstvím vybraných UTXOs pro platbu (tedy se vybírá částka z největších mincí, dokud není naplněna požadovaná hodnota) nebo maximalizací UTXOs (skládáme nejmenší mince dokopy) s čehož plyne větší velikost transakce a tedy větší poplatky těžaři. Hlavním výstupem je tlačítko pro vytvoření podepsané transakce, při zadání validních údajů, čímž se otevře okno SendTransactionDialogViewModel se shrnutím podepsané transakce a možnosti ji vyslat do bitcoinové sítě přes nastaveného RPC klienta.

SendTransactionDialog
SendTransactionDialog je okno shrnující data o vytvořené transakci, které umožňuje za pomocí předaného RPC klienta broadcastovat transakci do sítě.

SendTransactionDialogViewModel pretentuje data o vytvořené transakci, při špatné konfiguraci či použití neaktuálních (utracených) mincí se zobrazí korespondující chybová hláška. Tlačítkem Cancel dojde ke zrušení podepsané transakce, při odeslání transakce do sítě, se vybrané vstupy transakce označí v databázi za utracené a nebudou již zobrazovány. Pokud jde o odeslání mincí na moji adresu, je nutné počkat, než se transakce zařadí za pomocí těžaře do blockchainu a potom v menu home refreshovat mince pro danou peněženku.