Mobilní systémy iOS

iOS

Platformě iOS je věnována celá druhá kapitola. Za účelem omezení redundance informací zde proto budou uvedeny pouze základní aspekty této platformy. Operační systém iOS můžeme nalézt na mobilních zařízeních Apple iPhone, iPad a iPod Touch.

Historie iOS

Operační systém iOS byl světu poprvé představen ve verzi 1.0 spolu s prvním komerčně úspěšným „dotykovým" mobilním telefonem iPhone 29. června 2007. Za oběma stojí společnost Apple, která sídlí v Cupertinu (Kalifornie, USA). Dne 11. července 2008 uvádí Apple iPhone 3G a iOS 2.0. Verze 2.0 přináší aplikaci App Store, která umožňuje ze stejnojmeného obchodu stahovat aplikace třetích stran. O rok později přichází Apple s iPhonem 3GS, který obsahuje novou verzi iOS 3. V červnu 2010 je uveden iOS 4, jenž je v říjnu 2011 nahrazen novým iOS 5.

Dne 12. září 2012 představuje Apple iPhone 5 spolu s iOS 6. Zanedlouho, 18. září 2013, byly představeny mobilní telefony iPhone 5C a iPhone 5S, na kterých běžel iOS7. Zatím poslední hlavní verze iOS byla vydána spolu s telefony iPhone 6 a iPhone 6 Plus. Této verzi, která nese označení iOS 8, je věnována celá druhá kapitola této práce.

Systémové aplikace iOS

Součástí iOS jsou také předinstalované systémové aplikace. Uvedu zde ty, které jsou obecně považovány za nejdůležitější:

- Aplikace Telefon – slouží k přijímání a vykonávání telefonátů.
- Aplikace Kontakty – udržuje veškeré kontakty telefonu.
- Aplikace Zprávy – umožňuje přijímání a odesílání zpráv SMS a MMS.
- Aplikace Mail – umožňuje příjem a odesílání e-mailů. V současné době podporuje pro příchozí poštu protokoly IMAP i POP3 a pro odchozí poštu protokol SMTP.
- Aplikace Safari – prohlížeč webových stránek.
- Aplikace App Store – umožňuje stažení aplikací z App Store.
- Aplikace Fotoaparát – slouží k pořizování fotografií a videa prostřednictvím vestavěného fotoaparátu.

Popis platformy iOS

Platforma iOS obdobně jako jakékoli jiné operační systémy tvoří vrstvu mezi hardwarem zařízení a uživatelskými aplikacemi. Aplikace tedy nekomunikují přímo s hardwarovou vrstvou zařízení, ale s předem definovanými rozhraními iOS. To zajišťuje konzistentní prostředí pro běh aplikací a snadnou přenositelnost těchto aplikací napříč různými iOS zařízeními.

Tato kapitola bude pojednávat o aktuální verzi iOS, která nese označení iOS 8. Detailům, jimiž se jednotlivé verze iOS 8 od sebe odlišují nebude věnována pozornost, protože jsou pro potřeby této práce naprosto bezvýznamné.

Architektura iOS je založena na 4 základních vrstvách, přičemž každá vyšší vrstva využívá prostředky nižší vrstvy, na které závisí. Jedná se o následující vrstvy, které jsou seřazeny od nejvyšší po nejnižší:

- Cocoa Touch
- Media
- Core Services
- Core OS

Vrstva Cocoa Touch

Vrstva Cocoa Touch obsahuje klíčové frameworky pro tvorbu iOS aplikací. Tyto frameworky definují nejen, jak aplikace vypadají, ale také zajišťují podporu multitaskingu, dotykového rozhraní, push notifikací a dalších vysokoúrovňových systémových funkcí. Nejdůležitější technologie, které poskytuje Cocoa Touch, jsou uvedeny v následujících odstavcích.

App Extensions

App Extension se poprvé objevily v iOS 8 a jako takové umožňují rozšířit vybrané části sytému. Platforma iOS podporuje App Extensions pro následující části systému, označované rovněž jako body rozšíření:

- Share
  - Sdílení obsahu (například na sociálních sítích).
- Action
  - Provedení nějaké jednoduché akce nad obsahem.
- Widget
  - Umožňuje vykonání jednoduchého úkolu z prostředí notifikačního centra nebo z přehledu „Dnes".
- Photo editing
  - Provedení úprav fotografie v aplikaci „Obrázky".
- Document provider
  - Poskytuje úložiště dokumentů, které může být přístupné i pro jinou aplikaci.
- Custom keyboard
  - Umožňuje využívat klávesnice třetích stran místo původní systémové.

Každý bod rozšíření pro své použití definuje své vlastní API.

Handoff

Jedná se o funkci poprvé představenou v iOS 8 a OS X 10.10. Handoff umožňuje začít vykonávat nějakou činnost v konkrétní aplikaci na jednom zařízení a poté ji dokončit na zařízení jiném. Handoff tedy uchovává aktuální stav rozdělané práce a tento stav, je-li to možné, synchronizuje s dalšími zařízeními.

Document Picker

Document picker umožňuje sdílení dokumentů mezi aplikacemi, které tento UIDocumentPickerViewController implementují. Dovoluje tím tedy „obejít" omezení

sandboxu, kdy každá aplikace má přístup pouze ke své oblasti file systému.

AirDrop

Funkce AirDrop umožňuje uživatelům sdílet fotografie, dokumenty a další druhy dat mezi blízkými zařízeními. Podpora odesílání souborů na jiné zařízení s iOS pomocí Airdrop je implementována ve třídě UIActivityViewController.

TextKit

TextKit je bohatě vybavená množina tříd, které mají na starosti práci s textem a precizní typografii. Použitím TextKit je možné různými způsoby stylovat text do odstavců, sloupců nebo stránek. Umožňuje také rozličné obtékání textu kolem jednotlivých grafických prvků.

UIKit Dynamics

UIKit Dynamics umožňuje aplikaci dynamicky měnit chování objektů, které implementují protokol UIDynamicItem. UIKit Dynamics podporuje následujících 5 druhů chování:

- UIAttachmentBehavior
  - Specifikuje spojení mezi dvěma dynamickými položkami nebo mezi jednou dynamickou položkou a konkrétním bodem. Pokud se pohne jedna položka nebo bod, pohne se současně i druhá položka.
- UICollisionBehavior
  - Umožňuje specifikovat „srážky" mezi jednotlivými dynamickými položkami.
- UIGravityBehavior
  - Specifikuje vektor gravitačního zrychlení, v jehož směru dynamická položka zrychluje do doby, než dojde k její kolizi s jinou dynamickou položkou nebo s hranicí rozsahu pohybu.
- UIPushBehavior
  - Specifikuje vektor kontinuální síly nebo pouze impulzu síly, která působí na dynamickou položku.
- UISnapBehavior
  - Specifikuje rovnovážný bod dynamické položky. Do tohoto bodu se dynamická položka po svém vychýlení vrací s předem nakonfigurovaným efektem.

Multitasking

Z důvodu optimalizace spotřeby jsou aplikace po stisku tlačítka „Home" uspány. Jsou tedy sice stále nahrány v operační paměti zařízení, ale nevykonávají žádný kód. To může být za některých situací problém, proto je pro konkrétní důvody možné z aplikace požádat iOS o čas potřebný k dokončení úkolu nebo nechat vykonávání tohoto úkolu na iOS samotném. Mezi možné způsoby řešení běhu aplikace na pozadí patří:

- Aplikace může iOS požádat o přidělení nezbytně krátkého času na dokončení důležitého úkolu. Během tohoto času aplikace skutečně běží na pozadí. V okamžiku, kdy aplikace úkol dokončí, musí dát vědět operačnímu systému, že ji již může uspat. Pokud vykonávání úkolu trvá příliš dlouho, oznámí iOS aplikaci, že ji za nějaký krátký čas uspí. Po uplynutí tohoto času iOS aplikaci bez dalšího varování uspí.
- Aplikace, která má specifikováno, že umí přehrávat audio, může běžet na pozadí bez omezení.
- Aplikace, která potřebuje na pozadí stáhnout nějaká data, by pro to měla používat objekt třídy NSURLSession. V tomto případě aplikace na pozadí ve skutečnosti neběží. Data totiž pro aplikaci stahuje sám iOS, přičemž aplikace je mezitím uspána.
- Pro aplikace, které požadují po iOS polohu, existují 3 možné způsoby, jak toho docílit:
  - Significant-change – aplikace je vyrozuměna, pokud dojde k výrazné změně polohy. Velikost této změny je možné programově nastavit.
  - Foreground-only – aplikace získává aktuální polohu pouze pokud je v popředí. Toto není možné považovat za případ multitaskingu.
  - Background – aplikace smí získávat data a tedy i běžet na pozadí kontinuálně.

Auto Layout

Auto Layout umožňuje rychlé a efektivní vytváření dynamických uživatelských rozhraní s využitím relativně malého množství kódu. Nahrazuje původně používaný model springs&struts.

Storyboards

Storyboard je doporučený model vytváření uživatelského rozhraní, včetně kompletní hierarchie jednotlivých UIView. Storyboards také definuje přechody mezi jednotlivými zástupci objektů typu UIViewController.

UI State Preservation

UI State Preservation zajišťuje uchování stavu všech aktuálních objektů typu UIView a UIViewController v případě, že aplikace je umístěna do pozadí. Toto zajistí, že v případě nedostatku operační paměti, kdy iOS ukončí naši aplikaci běžící na pozadí, se při dalším startu tato aplikace spustí se stejným rozložením objektů UIView a UIViewController, s jakým ji iOS ukončil.

Apple Push Notification Service

Apple Push Notification Service umožňuje informovat uživatele o jakýchkoli nových informacích a to i v případě, že odpovídající aplikace neběží. Tyto notifikace mimo jiné upozorňují uživatele, že má otevřít danou aplikaci. Pro zasílání push notifikací je potřeba mít vlastní server, který bude komunikovat jak aplikací v zařízení, tak s Apple Push Notification Serverem.

Local Notifications

Jedná se o obdobu push notifikací. Rozdílem je to, že pro zaslání lokální notifikace není potřeba externí server. Jak je z názvu patrné, lokální notifikace může aplikace zaslat jen a pouze na zařízení, na kterém je nainstalována.

Gesture Recognizers

Gesture Recognizers jsou množinou všech tříd odvozených od základní třídy UIGestureRecognizer. Úkolem Gesture Recognizers je detekce a vyhodnocování různých druhů dotyků a dotykových gest.

Standard System View Controllers

Standard System View Controllers je množina systémových „dialogů", které je možné v aplikaci použít pro některé často používané funkce. Příkladem takových dialogů jsou například zástupci následujících:

- Dialog vytvoření e-mailu nebo SMS
- Dialog pro zobrazení nebo úpravu kontaktu
- Dialog pro pořízení fotografie
- Výběr obrázku z galerie obrázků
- Dialog pro natočení videoklipu
- Otevření nebo náhled souboru
- Níže jsou uvedeny Cocoa Touch Frameworky:
  - Address Book UI Framework
  - EventKit UI Framework
  - GameKit Framework
  - iAd Framework
  - MapKit Framework
  - Message UI Framework
  - Notification Center Framework
  - PushKit Framework
  - Twitter Framework
  - UIKit Framework

Vrstva Media

Vrstva Media obsahuje důležité funkce pro práci s grafickými, audio a video technologiemi.

Grafické technologie

Grafické technologie z vrstvy Media jsou navrženy tak, aby bez problému spolupracovaly s architekturou UIKit view. To dává vývojáři na výběr ze dvou možností. Buď použít standardní view, nebo uživatelsky vytvořené view, obojí pomocí jakékoli z níže uvedených technologií:

- UIKit Graphic
  - Definuje tzv. high-level podporu pro kreslení obrázků, Bézierových křivek a animování obsahu jednotlivých view. Zajišťuje rychlé a efektivní vykreslování obrázků a textového obsahu.
- Core Graphic Framework
  - Core Graphic je nativním vykreslovacím enginem pro iOS aplikace. Umožňuje vykreslovat 2D vektorovou a bitmapovou grafiku. Renderování však není tak rychlé jako v případě OpenGL ES.
- Core Animation
  - Jedná se o základní technologii, která optimalizuje vykreslování animací v aplikaci. UIKit view používá tuto technologii při animování defaultně.
- Core Image
  - Core image poskytuje množinu filtrů pro zpracování videa i statických obrázků. Jedná se o nedestruktivní filtry, které zachovávají originální předlohu nezměněnu.
- OpenGL ES a GLKit
  - OpenGL ES je výkonný 2D a 3D vykreslovací engine, který pro vykreslení využívá hardwarovou akceleraci. Pro svou rychlost je využíván především herními vývojáři a aplikacemi, které obsahují renderově náročnou grafiku.
- Metal
  - Metal je rozhraní, které na A7 GPU a všech novějších vyžaduje velice malou režii pro svůj běh. Je proto velice výkonným jak renderovacím tak výpočetním rozhraním.
- TextKit a Core Text
  - TextKit je využíván pro vytváření precizní typografie a správu vykreslování textu obecně.
- Image I/O
  - Image I/O nabízí rozhraní pro čtení a zápis obrázků různých formátů.
- Photos Library
  - Photos Library zajišťuje přístup do uživatelovy galerie obrázků a videí.

Audio technologie

Audio technologie iOS z vrstvy Media pracují s hardwarovými zvukovými prostředky mobilního zařízení a umožňují aplikacím přehrávat a nahrávat audio ve vysoké kvalitě, zpracovávat formát MIDI a v neposlední řadě také zařizují přístup k systémovým zvukům.

Tyto audio technologie dokáží pracovat s různými formáty pro uchování zvukových „standardní" formáty. Podporované jsou následující:

- AAC
- Apple Lossless (ALAC)
- A-law
- IMA/ADPCM (IMA4)
- Linear PCM
- µ-law
- DVI/Intel IMA ADPCM
- Microsoft GSM 6.10
- AES3-2003

Stejně jako grafické technologie mají i audio technologie několik zástupců:

- Media Player Framework

◦ Umožňuje přístup do uživatelovy knihovny iTunes a přehrávání jak jednotlivých skladeb, tak celých playlistů. Nenabízí však žádnou větší kontrolu nad přehráváním.

- AV Foundation

◦ AV Foundation slouží k pořizování nahrávek a přehrávání audia i videa. Umožňuje velice dobrou kontrolu přehrávání média.

- OpenAL

◦ OpenAl je multiplatformní průmyslový standard pro 3D audio rozhraní. Pro svou vysokou efektivitu bývá používán převážně herními vývojáři.

- Core Audio

◦ Core Audio je soubor frameworků, který nabízí sofistikovaná rozhraní pro nahrávání a přehrávání různých audio formátů a MIDI. Core Audio je doporučeno zkušeným vývojářům, kteří vyžadují vysoký stupeň kontroly nad nahráváním a přehráváním audia.

Video technologie

Video technologie z vrstvy Media zajišťují podporu pro přehrávání jak lokálně umístěných video dat, tak dat, která jsou streamovaná ze sítě internet. U zařízení, obsahujících odpovídající hardware, umožňují tyto technologie též pořizování videonahrávek a jejich případné začlenění do aplikace. Následující seznam uvádí zástupce video technologií z vrstvy Media:

- UIImagePickerController

◦ Tato třída má na starosti interakci s uživatelem v případě, kdy od něj požadujeme získání obrázku nebo videa z jeho knihovny v zařízení, nebo když po něm vyžadujeme pořízení fotografie či videa.

- AVKit Framework

◦ AVKit Framework nabízí jednoduché rozhraní pro přehrávání videa. Podporuje jak přehrávání na celou obrazovku, tak také v režimu okna. Umožňuje pouze základní ovládání přehrávaného videa.

- AV Foundation Framework

◦ AV Foundation Framework nabízí pokročilé rozhraní pro nahrávání a přehrávání videa. Umožňuje s videem vykonávat mimo jiné i velice sofistikované činnosti typu „obraz v obraze".

- Core Media Framework

◦ Definuje tzv. low-level interface a datové typy. Většina aplikací nepotřebuje využívat tento framework přímo.

Platforma iOS podporuje mimo jiné tyto formáty video dat:

- H.264 video do velikosti datového toku 1.5 Mbps, 640x480 pixelů, 30 snímků za sekundu, s audio stopou AAC-LC do velikosti datového toku 160Kbps.
- MPEG-4 video do velikosti datového toku 2.5 Mbps, 640x480 pixelů, 30 snímků za sekundu, s audio stopou AAC-LC do velikosti datového toku 160Kbps.

Frameworky vrstvy Media

- Assets Library Framework
- AV Foundation Framework
  - AVKit Framework
  - Core Audio
    - CoreAudio Framework
    - AudioToolbox Framework
    - AudioUnit Framework
    - CoreMIDI Framework
    - MediaToolbox Framework
  - CoreAudioKit Framework
- Core Graphic Framework
- Core Image Framework
- Core Text Framework
- Core Video Framework
- Game Controller Framework
  - GLKit Framework
- Image I/O Framework
- Media Accessibility Framework
- Media Player Framework
- Metal Framework
- OpenAL Framework
- OpenGL ES Framework
- Photos Framework
- Photos UI Framework
- Quartz Core Framework
- SceneKit Framework
- SpriteKit Framework

Vrstva Core Services

Vrstva Core Services poskytuje aplikacím kromě esenciálních systémových služeb (definice základních datových kontejnerů, práce s řetězci, práce s časem a daty, podpora vláken atd.) také podporu lokace, senzorů pohybu a náklonu, informací o operátorovi, přístupu k událostem v kalendáři, iCloud, sociálních médií a připojení k síti. Níže budou popsány některé zajímavé funkce Core Services.

Peer-to-Peer služby

Peer-to-Peer slouží k navázání připojení k blízkému zařízení pomocí Bluetooth nebo společné Wi-Fi sítě. Většinou se této služby využívá v multiplayerových hrách.

Úložiště iCloud

Úložiště iCloud umožňuje ukládat uživatelovy dokumenty a data na centrální server provozovaný společností Apple. Uživatel k nim poté může přistupovat a editovat je ze všech svých zařízení a to bez nutnosti data explicitně synchronizovat nebo je znovu uploadovat. Výhodou je také dostupnost okamžité zálohy v případě ztráty zařízení.

Existuje několik způsobů, jak využívat iCloud:

- iCloud document storage
  - Používá se pro ukládání uživatelových dokumentů a dat do uživatelova iCloud účtu.
- iCloud key-value data storage
  - Používá se pro ukládání malého objemu dat, který sdílí všechny instance jedné aplikace. Využitelné například jako záloha nastavení aplikace.

Blokové objekty

Blok je speciální druh konstrukce kódu, která má několik nejčastějších použití:

- Jako náhrada za callback.
- Jako náhrada z delegátů a delegátských metod.
- K vykonávání asynchronních úloh.
- Pro implementaci tzv. completion handleru

Ochrana dat

Ochrana dat umožňuje aplikaci označit některé vybrané soubory jako „chráněné". Tyto soubory poté iOS ukládá zašifrované. Pokud je zařízení uzamčeno, nemá k takto ochráněným souborům přístup jak potenciální útočník, tak ani samotná aplikace. V okamžiku odemknutí zařízení je vytvořen klíč pro dešifrování souborů a aplikace může chráněný soubor opět bez problémů číst.

Sdílení souborů

Funkce sdílení souborů dovoluje přistupovat k uživatelským souborům, které ukládá daná aplikace, prostřednictvím aplikace iTunes ve verzi 9.1 nebo vyšší. Uživatel poté může přesouvat soubory ze/do složky dokumentů aplikace na zařízení. Tato funkce však neumožňuje jakékoli sdílení souborů mezi dvěma aplikacemi jednoho zařízení.

Grand Central Dispatch (GCD)

Grand Central Dispatch je technologie sloužící pro optimalizaci chodu aplikací na víceprocesorových zařízeních. Na platformě iOS se poprvé představila v jeho verzi 4.0.

In-App nákupy

In-App nákupy otevírají cestu k nákupům různého prémiového obsahu. Tímto prémiovým obsahem může být například bonusový doplněk ve hře nebo předplatné novin.

In-App nákupy podporují následující modely pořizování prémiového obsahu:

- Consumable products
  - Jedná se o prémiový obsah, který se průběžně spotřebovává. Například nákup hnojiva na virtuální farmu.
- Non-consumable products
  - Jedná se o prémiový obsah, který se nespotřebovává. Příkladem může být například nová postava do hry.
- Auto-renewable subscriptions
  - Jedná se o předplatné prémiového obsahu, které je automaticky obnovováno. Podobný model předplatného může mít třeba elektronická verze novin.
- Non-renewable subscriptions
  - Jedná se o podobný typ jako předchozí příklad. Rozdílem je, že předplatné není automaticky obnovováno.
- Free subscriptions
  - Jedná se o předplatné prémiového obsahu, které je však zcela zdarma. Tento typ předplatného, na rozdíl od předchozích dvou příkladů, nikdy nevyprší.

SQLite

Integrované SQLite dovoluje aplikacím využívat možností databáze bez toho, že by bylo potřeba externího serveru. V iOS aplikaci je možné vytvořit lokální databázový soubor, který bude touto aplikací možné také plně spravovat.

Databáze je navržena pro obecné použití, ale je stále velice dobře optimalizována pro rychlé vyhledávání na výkonově omezeném mobilním zařízení.

Podpora XML

Foundation framework v sobě zahrnuje také třídu NSXMLParser, která pracuje jako regulérní XML parser. Náročnější operace s XML daty je možné provádět pomocí knihovny libxml2, která je také součástí iOS.

Vrstva Core OS

Vrstva Core OS zahrnuje nízkoúrovňové funkce, na kterých je vystaveno mnoho dalších technologií a frameworků. Významné součásti této vrstvy jsou popsány v následujících kapitolách.

Accelerate Framework

Accelerate Framework poskytuje rozličná rozhraní pro výpočty při zpracování digitálních signálů, dále pro výpočty lineární algebry a pro výpočty při transformaci obrázků. Oproti implementaci podobných funkcí vlastními silami má použití Accelerate Framework výhodu v tom, že stejný kód může být velice efektivně vykonáván na různých zařízeních i se zcela odlišnou hardwarovou konfigurací.

Core Bluetooth Framework

Core Blouetooth Framework obsahuje množinu rozhraní, která mohou být využita převážně pro interakci s Bluetooth LE příslušenstvím. Tento framework umožňuje mimo jiné následující činnosti:

- Vyhledávání dostupných Bluetooth zařízení v dosahu a připojení k jednomu vybranému.
- Vytvoření periferie z iOS zařízení schopnou ovládat jiné Bluetooth zařízení.
- Zasílání iBeacon z iOS zařízení.

- Uchování stavu Bluetooth připojení tak, že muže být automaticky obnoveno v případě nového spuštění aplikace.
- Možnost automatické notifikace, která oznámí aplikaci, že je v dosahu Bluetooth nějaké příslušenství.

External Accessory Framework

External Accessory Framewor poskytuje rozhraní pro komunikaci s příslušenstvím, které je připojeno k iOS zařízení. Tento framework umožňuje získat informace o připojeném příslušenství a případně s ním navázat komunikaci. Poté je již možné v aplikaci využívat všechny funkce, které dané příslušenství nabízí.

Generic Security Services Framework

Generic Security Services Framework nabízí množinu rozhraní týkající se bezpečnosti. Základní rozhraní jsou definována v RFC 2743 a RFC 4401. GSS Framework zahrnuje také správu uživatelských přihlašovacích údajů, které sice nejsou zmíněnými standardy specifikovány, ale jsou potřebné pro mnoho druhů aplikací.

Local Authentication Framework

Local Authentication framevork nabízí rozhraní, které umožňuje na podporovaných zařízeních požádat o autentizaci uživatele prostřednictvím Touch ID, tedy za pomoci čtečky otisků prstů integrované do zařízení. Tento framework neumožňuje získat samotný otisk prstu, výsledkem autentizace je pouze pravdivostní hodnota, která vyjadřuje, zda se autentizace zdařila nebo ne.

Network Extension Framework

Network Extensions Framework poskytuje rozhraní pro konfiguraci a kontrolu připojení do tzv. Virtual Private Network (VPN).

Security Framework

Security Framework nabízí jako doplněk ke GGS Frameworku množinu rozhraní týkající se zabezpečení dat, s nimiž aplikace pracuje. Tato rozhraní mají na starosti správu certifikátů, veřejných a privátních klíčů a ověřování důvěryhodnosti. Dále nabízí podporu generování kryptograficky bezpečných pseudonáhodných čísel, úložiště pro certifikáty, privátní klíče a jiná citlivá data, které se v prostředí Apple označuje jako keychain. Security Framework také poskytuje prostředky pro symetrické šifrování, autentizační zprávy typu HMAC a tvorbu hashů.

System

System v sobě zahrnuje kernel, ovladače a nízkoúrovňová UNIX rozhraní. Kernel, který je odvozen od mikrojádra Mach, je zodpovědný za všechny funkce požadované po operačním systému. Jedná se hlavně o následující:

- Správa virtuální paměti
- Správa vláken
- Správa file systému
- Správa síťových připojení
- Správa a zajištění meziprocesové komunikace

Ovladače v této vrstvě zajišťují komunikaci mezi hardwarem a systémovými frameworky. Z bezpečnostních důvodů je přístup k prostředkům kernelu a ovladačů povolen pouze některým systémovým frameworkům a aplikacím.

iOS poskytuje mnoho rozhraní pro přístup k nízkoúrovňovým funkcím operačního systému. Aplikace mohou tyto funkce využívat prostřednictvím knihovny LibSystem. Zmiňovaná rozhraní jsou založena na jazyku C a přináší podporu pro následující:

- POSIX vlákna a Grand Central Dispatch (GCD)
- BSD sokety
- Standardní I/O
- Bonjour a DNS služby
- Alokace paměti
- Jazyková lokalizace a informace o ní
- Matematické výpočty

Podpora 64 bitové HW architektury

Platforma iOS byl původně navrhován tak, aby podporoval binární soubory běžící na zařízeních využívajících 32 bitovou architekturu. S příchodem iOS 7 byla představena 64 bitová architektura. Všechny systémové knihovny jsou tedy uzpůsobeny pro chod na 64 i 32 bitových architekturách. V případě kompilace kódu aplikace jako 64 bitového je možné v některých aplikacích a pouze na některých zařízeních lépe využít hardwarové prostředky. iOS používá model LP64, který umožňuje jednodušší portovatelnost kódu než ostatní modely jako ILP64 nebo LLP64.

Návrh a implementace aplikace pro iOS

Tato kapitola bude pojednávat o iOS aplikaci FormApps Mobile, která slouží primárně pro podporu elektronického podepisování webových formulářů společnosti Software602 a.s., přičemž obecně do těchto formulářů přináší další funkce a rozšiřuje tak možnosti použití těchto formulářů.

Analýza problému

Formuláře Software602 jsou webové formuláře založené na moderních variantách technologií HTML 5 a AJAX, které dále komunikují se serverovou stranou, později v textu označovanou jako backend, případně pouze jako „server", pokud to bude z kontextu pochopitelné. Tyto webové formuláře mají převážně za úkol napomáhat v řízení vnitrofiremních procesů, dále při opatřování dokumentů elektronickým podpisem a jejich následné dlouhodobé archivaci.

Vzhledem k povaze formuláře je ve velké části případů možné jej vyplnit přímo v jakémkoliv internetovém prohlížeči, a to dokonce i v jeho „mobilních" ekvivalentech. Přestože je již většina webových technologií velice sofistikovaná, jsou zde stále určité činnosti, které se pomocí samostatného webového formuláře nedají vykonávat na všech platformách bez problémů (nejvíce postižené jsou v tomto smyslu právě platformy mobilní).

Těmito činnostmi je myšleno například právě připojení elektronického podpisu, odeslaní dat z backendu na jiný server (zvláště pokud je pro tento jiný server nutná autentizace, ať již základní basic, nebo pomocí klientského certifikátu, NTLM) a konkrétně na platformě iOS je to pro absenci uživatelsky přístupného file systému zrovna vkládání příloh do webových formulářů. Tato chybějící funkčnost, která je pro celkový chod Software602 ekosystému naprosto esenciální, byla hlavní motivací pro vytvoření mobilní aplikace, jež by měla tyto nedostatky řešit.

Popis aplikace

Aplikace Software602 FormApps Mobile je mobilní aplikace pro platformy iOS, Android a Windows Phone (zde budeme vždy uvažovat pouze aplikaci pro iOS), která dokáže zcela plnohodnotně pracovat s výše zmíněnými webovými formuláři Software602. Doplňuje tedy standardní funkce webového prohlížeče o následující možnosti: podepsání (i vizuální) lokálního PDF dokumentu, podepsání vzdáleného dokumentu na serveru, podepsání vyplněného formuláře, časové razítkování lokálních PDF dokumentů, časové razítkování vzdálených dokumentů na serveru, připojení vlastnoručního podpisu k lokálnímu PDF dokumentu, přeposlání dat z backendu na jakýkoli jiný vzdálený server, který může vyžadovat autentizaci (basic, klientským certifikátem, NTLM), vložení vyfocené fotografie do formuláře, načtení dat z čárkového nebo QR kódu do formuláře, vkládání příloh do formuláře, úložiště dokumentů včetně podpory Google Drive, generování certifikátu s využitím SecuStamp CA, generování komerčních a kvalifikovaných (osobních i systémových) certifikátů PostSignum.

V následujících kapitolách bude stručně vysvětleno, jak jsou vlastně jednotlivé funkce implementovány. A to jak z pohledu koncového uživatele, tak z technologického pohledu.

Popis architektury aplikace FormApps Mobile pro iOS

Aplikace je z velké části napsaná v programovacím jazyce Objective-C++. Kód aplikace, který přímo pracuje s knihovnami třetích stran, napsanými v čistém jazyce C, je také psán v jazyce C, případně v jazyce C++. To je případ zejména práce s knihovnou OpenSSL, jež zajišťuje kryptografické funkce a dále práce s knihovnou libjpeg, která má na starosti transformaci JPEG obrázku při převodu obrázku do formátu PDF.

Základem celé aplikace je webový prohlížeč, který slouží k práci s webovými formuláři. Toto „jádro" je postaveno na systémové komponentě UIWebView. Ta má na starosti vykreslování veškerých html stránek. Pro komunikaci ve směru webový formulář -> aplikace je využita implementace vlastní metody:

- webView:shouldStartLoadWithRequest:navigationType:

definované protokolem UIWebViewDelegate. Veškeré požadavky na stažení elementu webové stránky tedy procházejí touto funkcí. Pokud chce webový formulář nějakou součinnost od aplikace, vytvoří požadavek na stažení elementu, který obsahuje přesně definovaný řetězec. Tento řetězec se skládá z kontrolního řetězce oznamujícího aplikaci, že si má ze serveru stáhnout požadavek k vykonání, a ze samotné URL, na které se nachází PKCS7 specifikující požadavek, který je potřeba vykonat. Tímto požadavkem se myslí například podepsání formuláře, vložení přílohy do formuláře, vzdálené podepsání na serveru atd.

U každého staženého PKCS7 požadavku je před jeho případným vykonáním ověřeno, zda u něj nedošlo k porušení integrity a zda je podepsán správným certifikátem. Tím je významně omezena míra potenciálního zneužití.

Výsledek vykonání požadavku je poté vrácen na server a webový formulář je o tom obeznámen voláním jeho konkrétní javascriptové funkce. Toto volání se provádí pomocí metody

- stringByEvaluatingJavaScriptFromString:

objektu třídy UIWebView, který zobrazuje právě zpracovávaný webový formulář.

Popis základních funkcí aplikace FormApps Mobile

FormApps Mobile nabízí veliké množství funkcí. Ze všech si však pro potřeby této práce vyberu ty nejdůležitější a popíši je jak z pohledu běžného uživatele, tak technologicky.

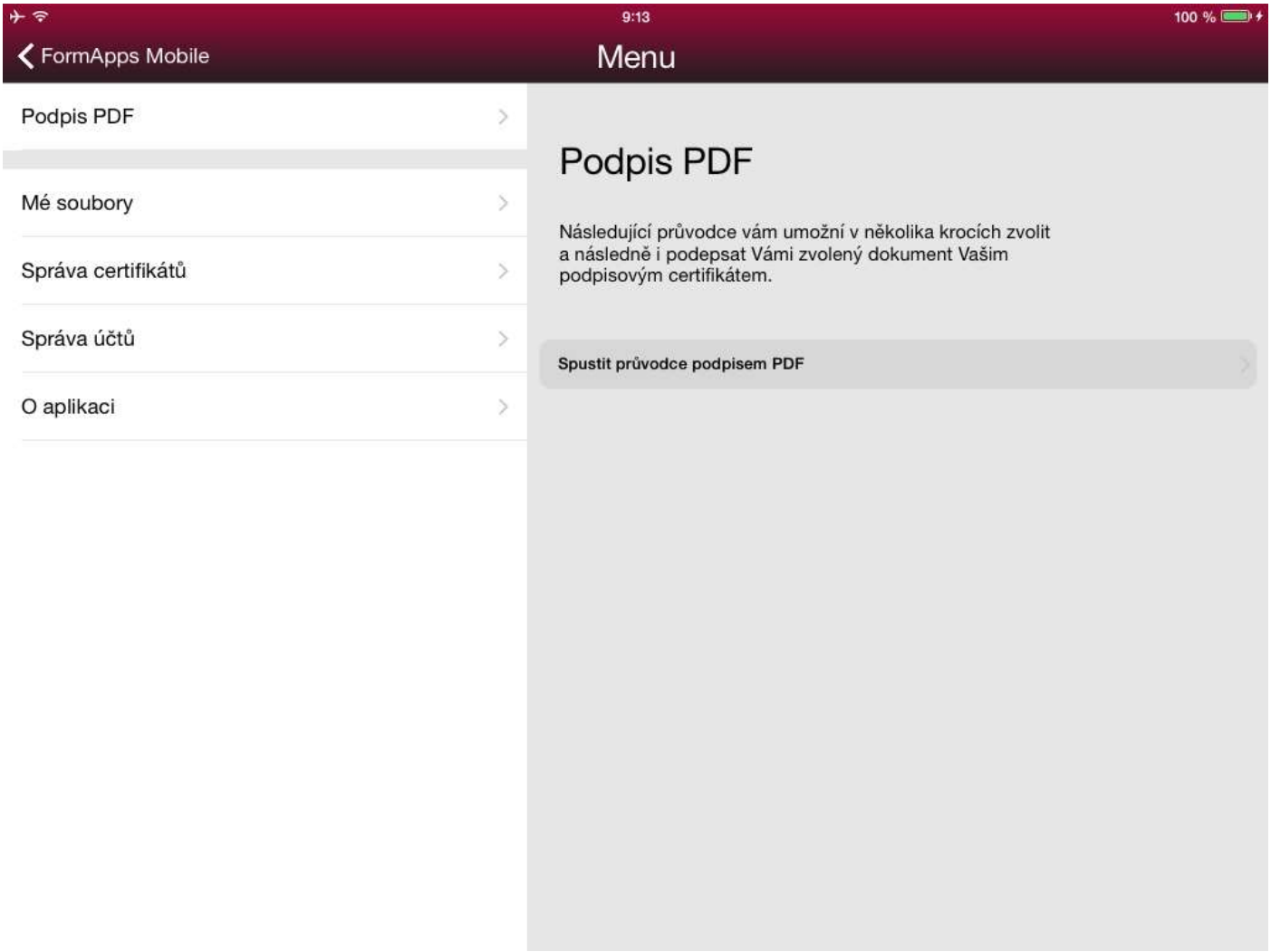
Podepsání lokálního dokumentu

Aplikace FormApps Mobile je schopna v současné chvíli lokálně podepsat pouze dokumenty typu PDF (pro podepisování ostatních dokumentů jako například *.docx, *.xlsx, *pptx, *.odt je určena aplikace Software602 Signer). Podepsáním lokálního dokumentu je myšleno připojení elektronického podpisu k PDF dokumentu, který se nachází v úložišti aplikace FormApps Mobile, nebo který se nachází v úložišti Google Drive, asociované s aplikací FormApps Mobile.

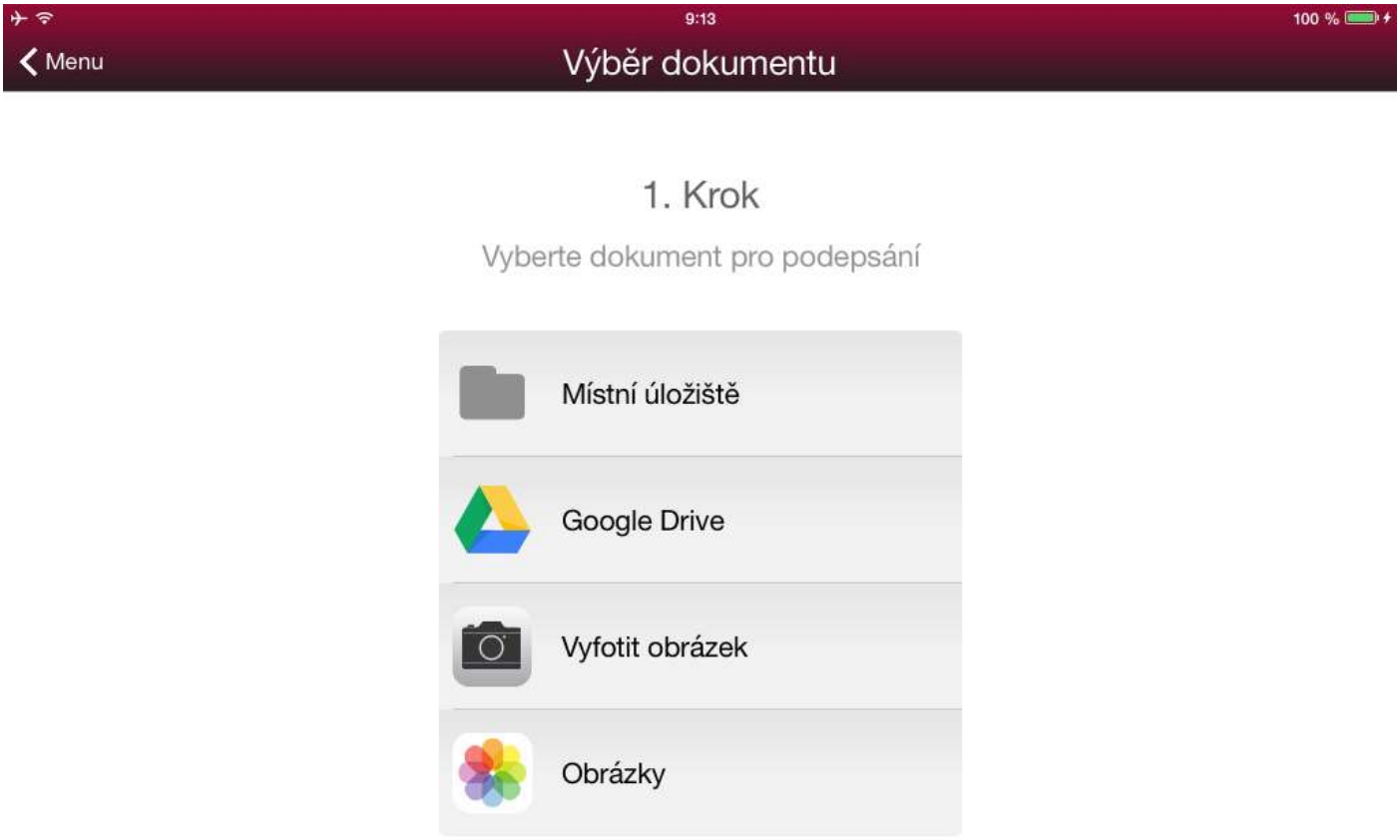
Vizuální podpis v dokumentu graficky znázorňuje, kdy a kdo daný dokument podepsal. Uživatel je před samotným podpisem vyzván k vybrání obdélníkové části v dokumentu, kam bude následně vizuální podpis umístěn.

Z technologického hlediska je využitím privátního klíče a jemu odpovídajícího certifikátu vytvořen podepsaný dokument splňující normu PDF/A. Tento dokument může být během podepsání opatřen také časovým razítkem. Vytvoření časového razítka je popsáno v samostatném odstavci týkajícím se časových razítek.

Proces podepsání je z uživatelského hlediska krok za krokem znázorněn níže.



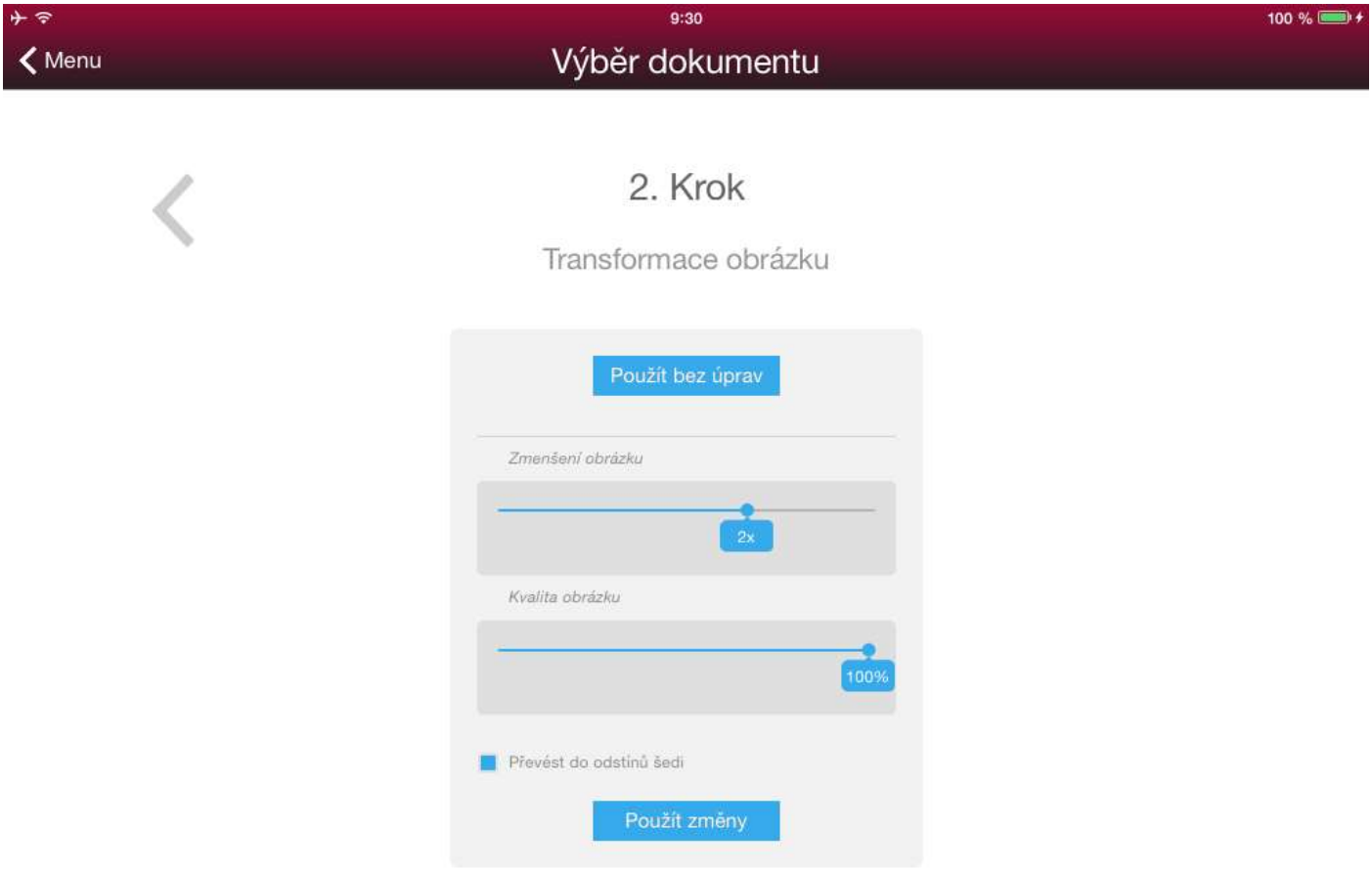
Obrázek 2: Dialog "Menu" s vybranou volbou "Podpis PDF", zdroj: Autor



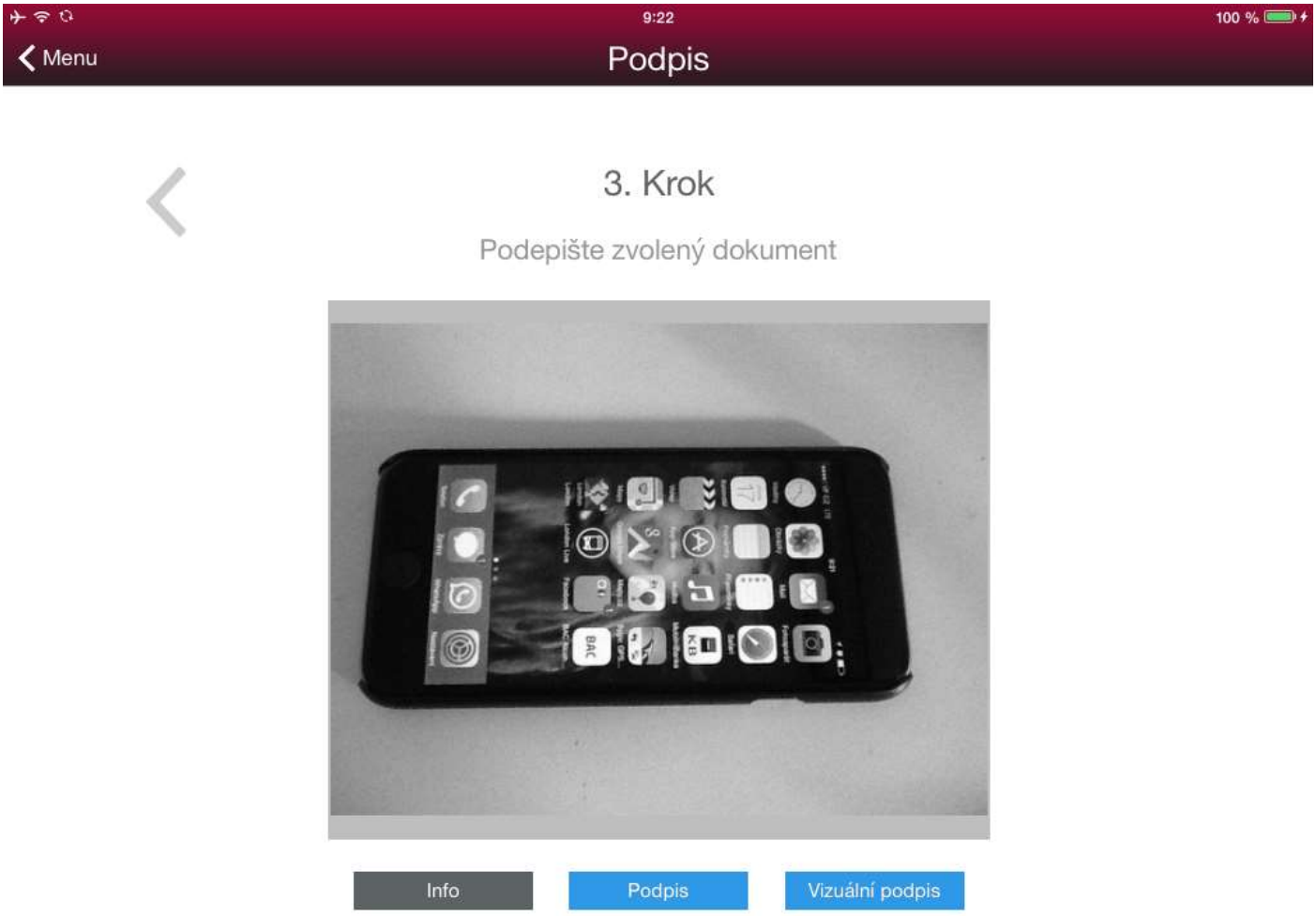
Obrázek 3: Dialog výběru dokumentu, zdroj: Autor

Obrázek 4: Zobrazený UIImagePickerController po vybrání možnosti "Vyfotit obrázek", zdroj: Autor

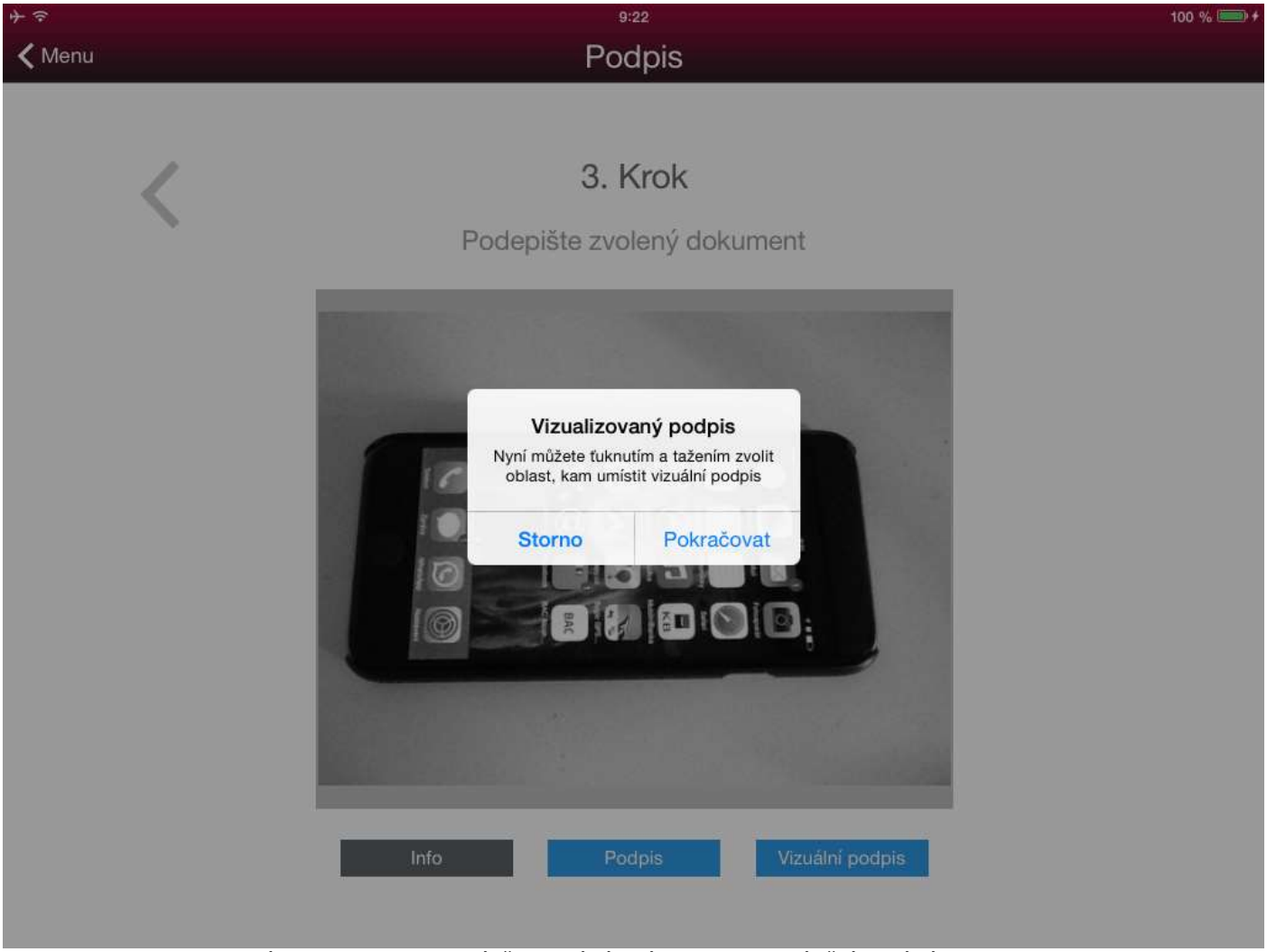
Obrázek 5: UIImagePickerController po pořízení fotografie, zdroj: Autor



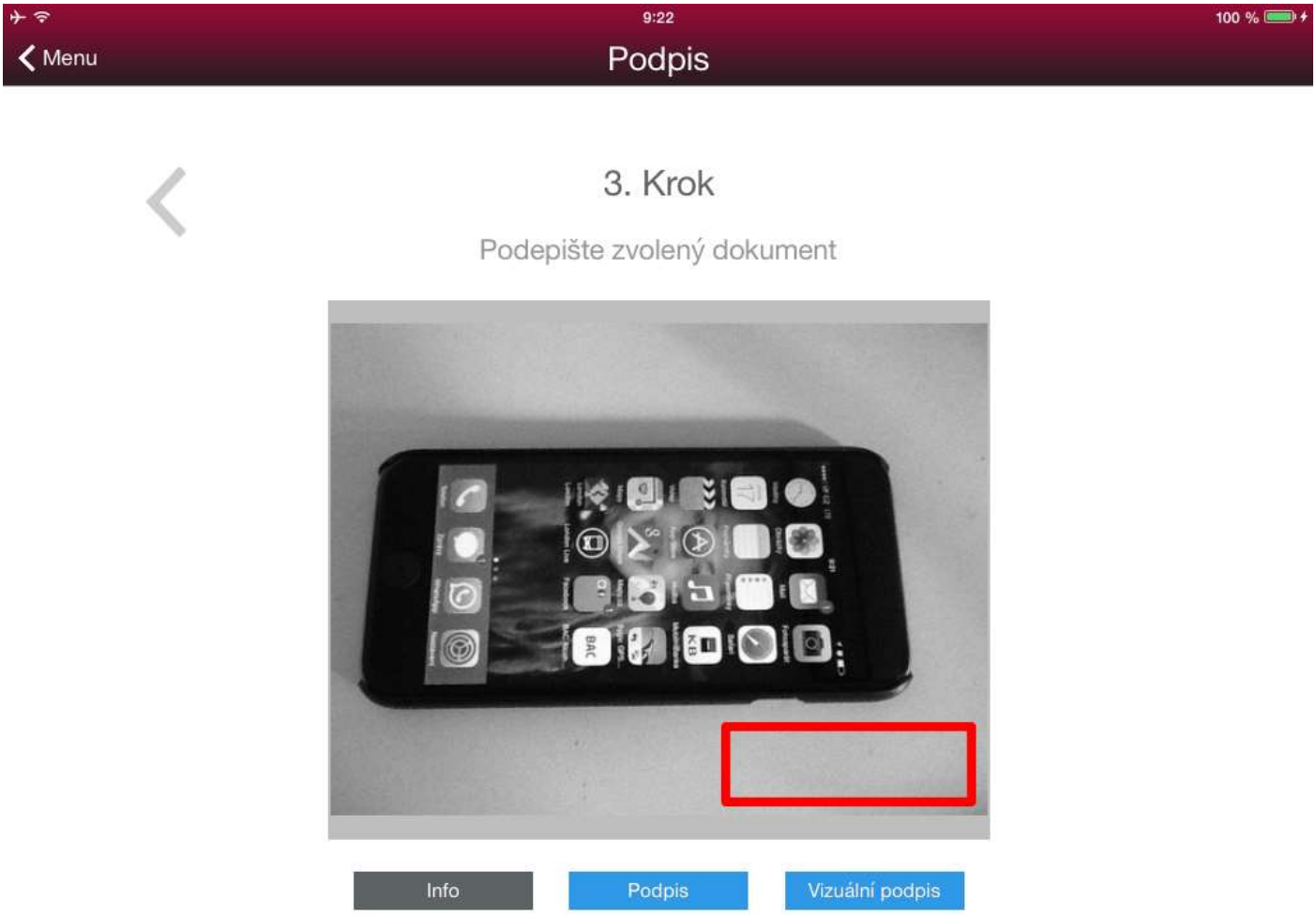
Obrázek 6: Dialog pro nastavení kvality obrázku pro konverzi do formátu PDF, zdroj: Autor



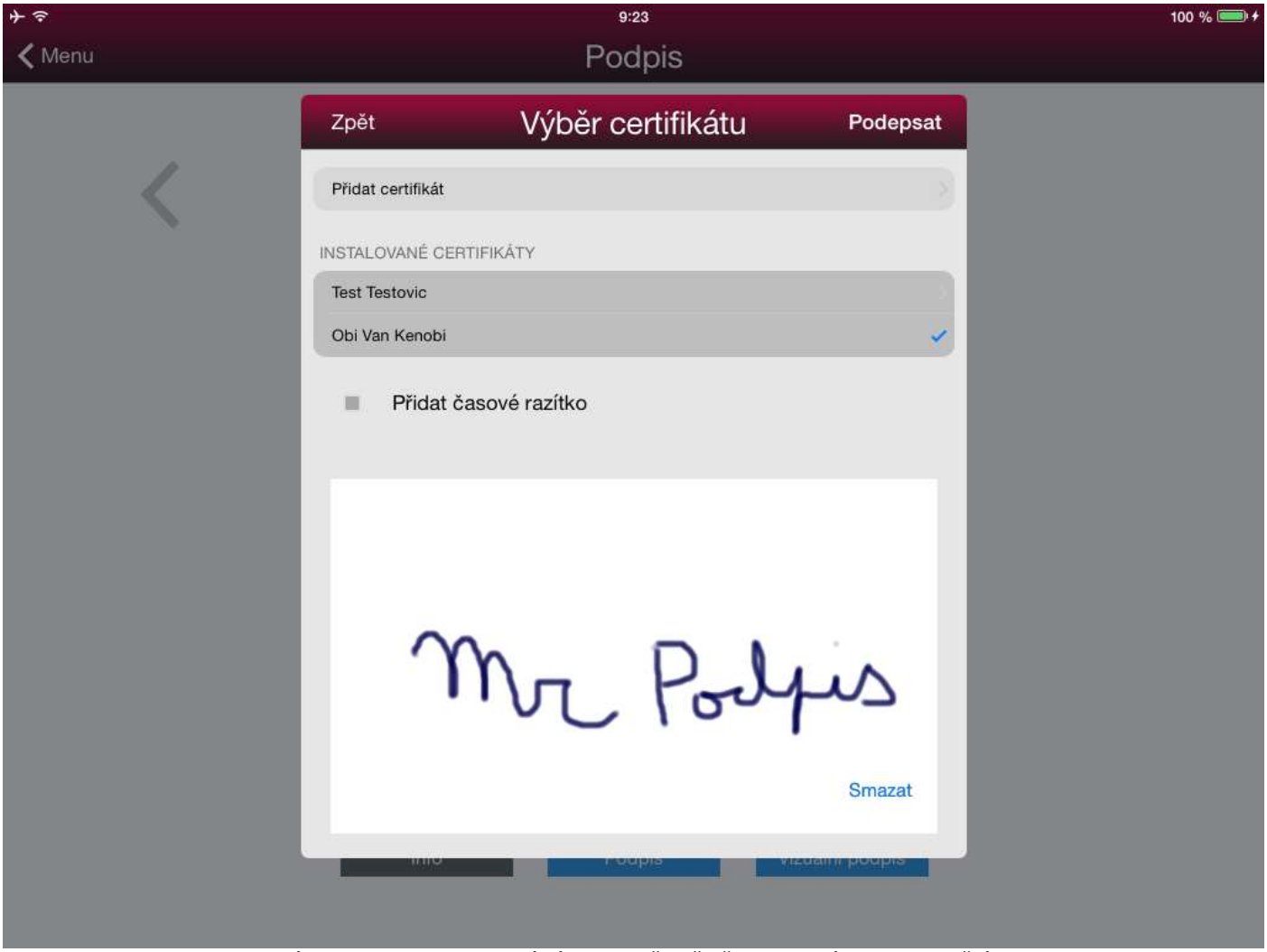
Obrázek 7: Náhled vytvořeného PDF dokumentu s možnostmi "Info", "Podpis" a "Vizuální podpis", zdroj: Autor



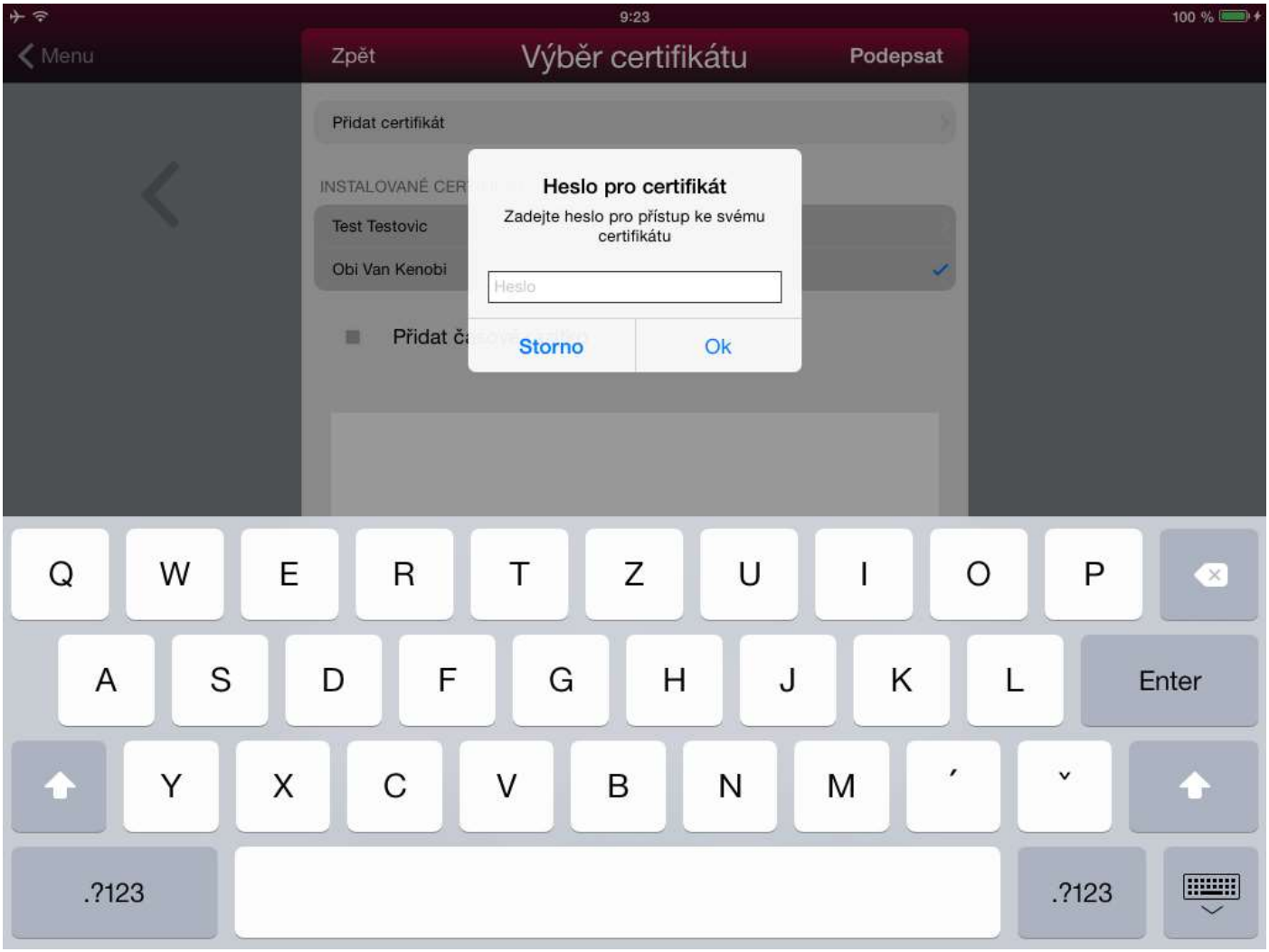
Obrázek 8: Informace o výběru obdélníkové oblasti pro umístění vizuálního podpisu zobrazená po zvolení volby "Vizuální podpis", zdroj: Autor



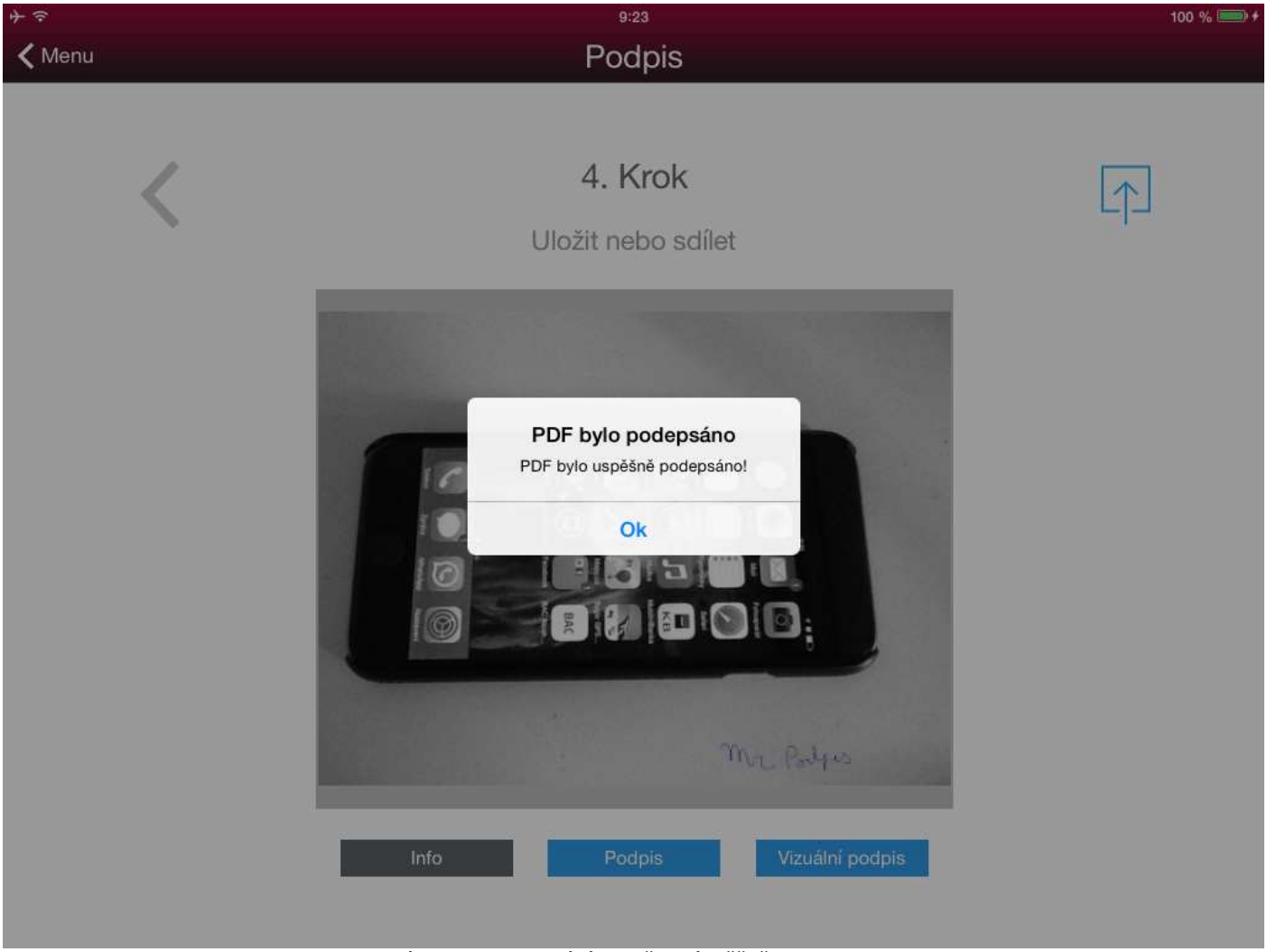
Obrázek 9: Po výběru oblasti pro umístění vizuálního podpisu, zdroj: Autor



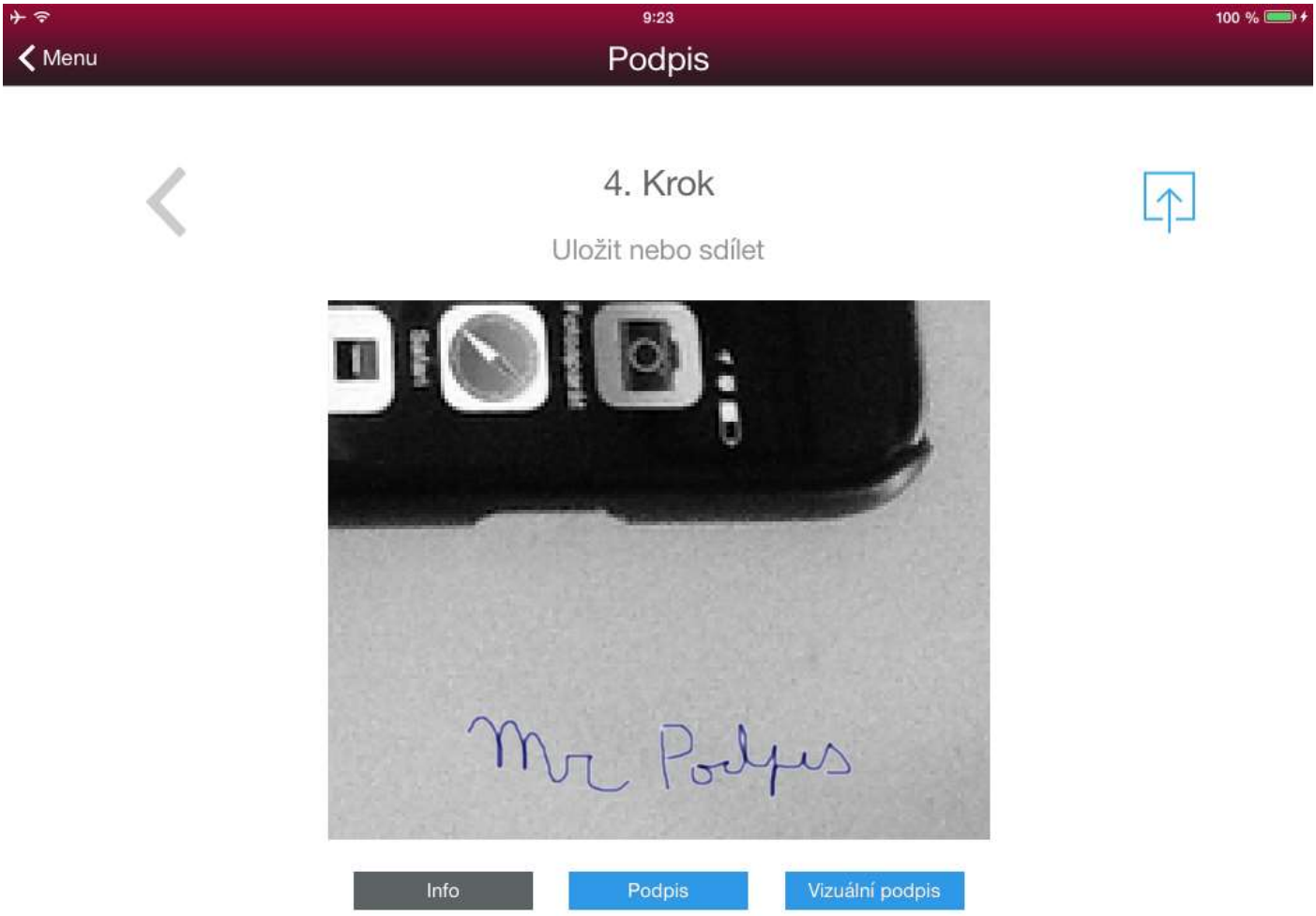
Obrázek 10: Dialog pro vizuální podpis včetně již provedeného vlastnoručního podpisu, zdroj: Autor



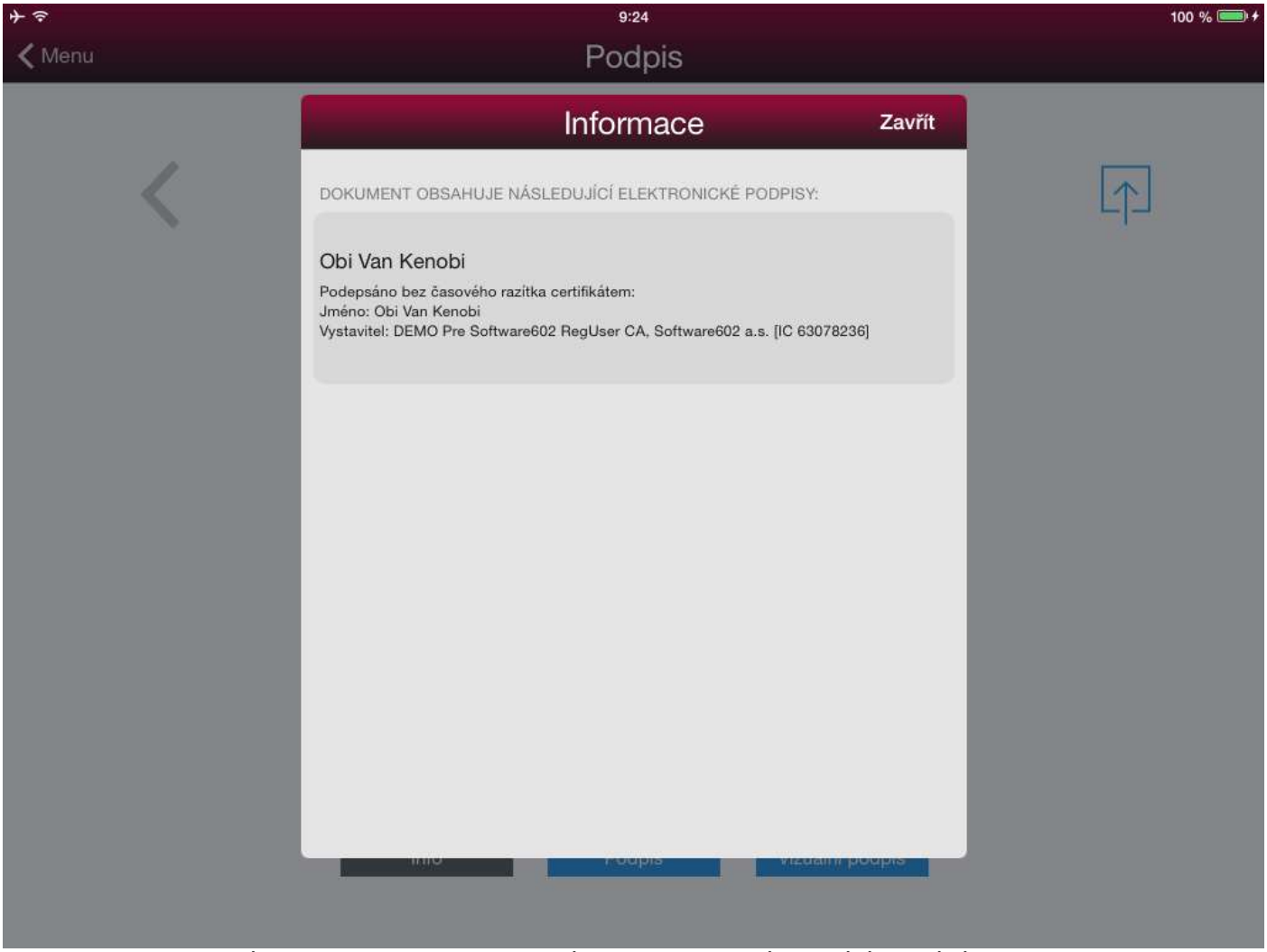
Obrázek 11: Výzva na zadání hesla pro přístup k privátnímu klíči, zdroj: Autor



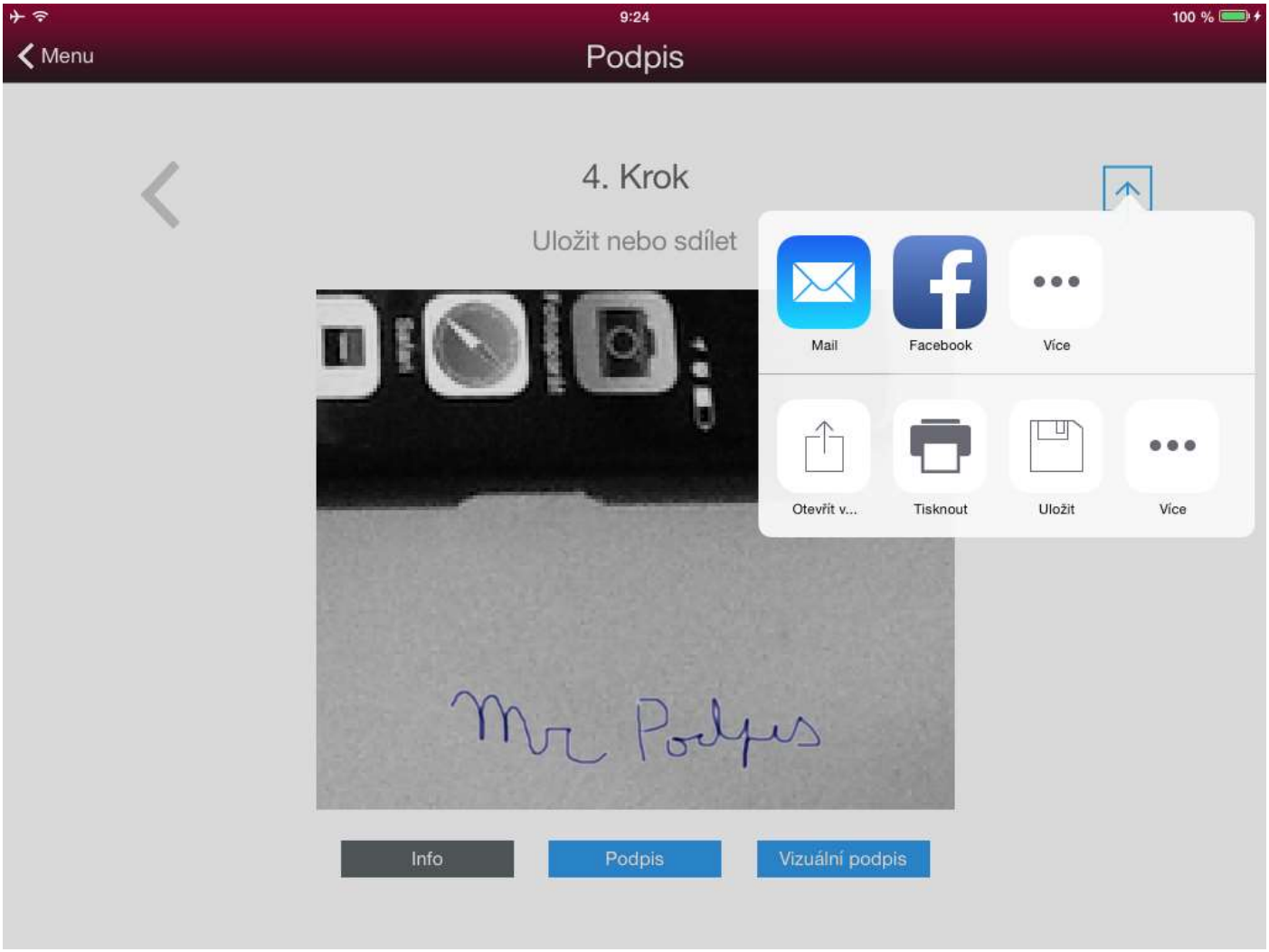
Obrázek 12: Podepsání proběhlo úspěšně, zdroj: Autor



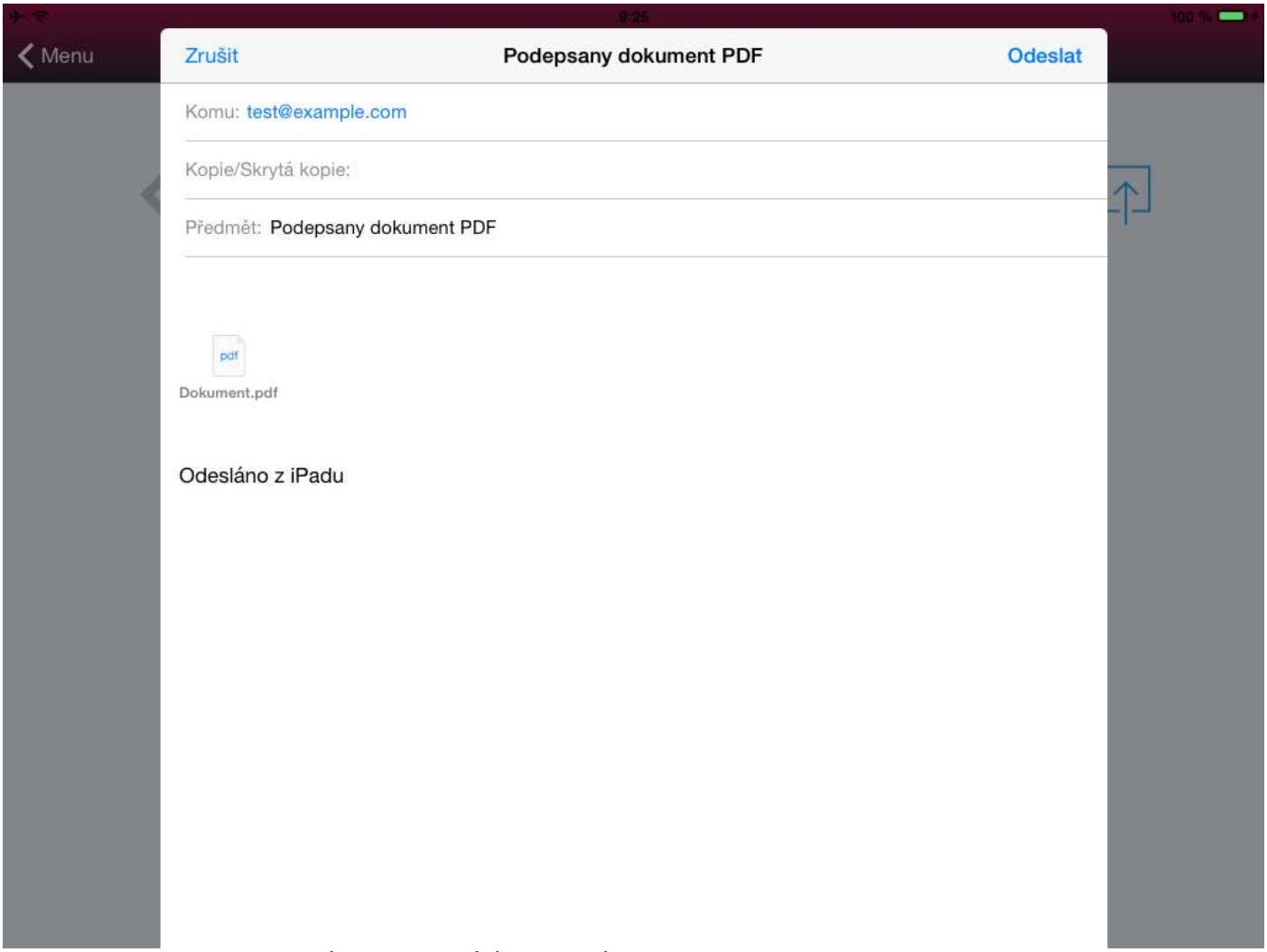
Obrázek 13: Detail vlastnoručního podpisu, zdroj: Autor



Obrázek 14: Informace o elektronickém podpisu, na který je navázán vizuální (v tomto případě dokonce vlastnoruční) podpis, zdroj: Autor



Obrázek 15: Znázornění možností sdílení dokumentu, zdroj: Autor



Obrázek 16: Odeslání podepsaného dokumentu e-mailem, zdroj: Autor

Podepsání vzdáleného dokumentu

Aplikace FormApps Mobile dokáže vzdáleně podepsat jakýkoliv elektronický dokument, který má pro to podporu na serveru. Toto vzdálené podepsání je možné vykonat pouze ve spolupráci s webovým formulářem (který je napojen na backend). Webový formulář si v reakci na uživatelovu akci, jako je například kliknutí na tlačítko, vyžádá po aplikaci elektronický podpis určitých dat. Tato data mohou mít dva možné tvary:

První možností je, že data pro podepsání jsou již přímo hashem dané části dokumentu (obecně je možné podepsat třeba pouze některé konkrétní části dokumentu, ne úplně celý). Tento hash je v drtivé většině SHA1 nebo nějaký zástupce rodiny SHA2, tedy obvykle SHA256. V případě, že data pro podepsání jsou již přímo hashem, dále se z nich hash nedělá a jsou pouze pomocí šifrovacích algoritmů RSA nebo DSA podepsána. Jedná se tedy pouze o „surový" podpis, který není zabalen do žádné obálky typu PKCS7. Je však převeden do formátu base64, zabalen do SOAP obálky a takto poslán webové službě na backendu.

Druhou možností je, že ze serveru přijde nějaký fragment dat. Z těchto dat je poté potřeba vytvořit hash. Podporované hashovací funkce jsou SHA1 a všichni zástupci z rodiny SHA2. Vytvořený hash se poté podepíše buď pomocí RSA nebo DSA a opět jako „surový" podpis je převeden do formátu base64, zabalen do SOAP obálky a poslán webové službě na backendu. Na serveru se poté „surový" podpis připojí k dokumentu podle normy, která je pro daný typ dokumentu specifická (PDF/A, XAdES, PAdES, CAdES, BES/EPES). Podepsání se provádí uživatelovým privátním klíčem, který je spolu s odpovídajícím certifikátem uložen v keychainu aplikace.

Podepsání vyplněného formuláře

Webový formulář jako takový podepsat nelze, protože se jedná o kombinaci HTML 5, AJAXu a dalších elementů jako třeba obrázků. Lze však úspěšně podepsat xml, které vzniklo transformací dat tohoto formuláře. Toto xml může být nejen elektronicky podepsáno tak, jak je specifikováno normou XMLDSig, ale může být také dlouhodobě věrohodné, pokud se pro podpis zvolí nějaká vyšší forma specifikace XAdES, například XAdES-A. Podepsání webového formuláře probíhá tedy následovně: V prvním kroku převede backend data z formuláře do odpovídajícího XML. Toto XML může být poté kdykoli backendem převedeno zpět do webového formuláře. V druhém kroku je aplikaci FormApps Mobile odeslána série hashu, které mají být podepsány. Ve třetím kroku jsou podepsané hashe (popis podepsání hashů byl zmíněn výše) zaslány zpět na server. V posledním čtvrtém kroku jsou podepsané hashe backendem vloženy do xml a je vytvořen dokument podle podle normy XAdES. Její stupeň záleží na nastavení serveru a na jeho propojení s certifikační autoritou.

Časové razítkování lokálních dokumentů

Časové razítko je zjednodušeně řečeno elektronický podpis časového údaje a samotného dokumentu. Tento elektronický podpis, včetně zmíněného časového údaje,

poskytuje konkrétní certifikační autorita. Časové razítko slouží jako důkaz toho, že daný dokument v daném čase v dané podobě existoval. Důvěryhodnost časového razítka závisí převážně na důvěryhodnosti zvolené certifikační autority a na samotném algoritmu elektronického podpisu.

Časové razítko je v aplikaci FormApps Mobile možné lokálně připojit pouze k dokumentům typu PDF. Technicky je připojení časového razítka prováděno následovně: Pomocí OpenSSL je vytvořena žádost o časové razítko. Ta je poté převedena do formátu base64, vložena do SOAP obálky a následně odeslána webové službě SecuStamp, která zde plní funkci certifikační autority. Po provedení basic autentizace, která může získat přihlašovací údaje z nastavení v aplikaci, je žádost zpracována a do aplikace je vráceno časové razítko. Toto časové razítko je poté spolu s elektronickým podpisem (časové razítko je vázáno na podepsání dokumentu uživatelovým certifikátem) vloženo do PDF dokumentu.

Časové razítkování vzdálených dokumentů na serveru

Vzdálené připojení časového razítka je v aplikaci možné pro všechny typy dokumentů, které na to mají podporu na straně serveru. Samotné časové razítkování probíhá následovně: V prvním kroku si server vyžádá časové razítko pro hash, který předá aplikaci. V druhém kroku aplikace získá časové razítko přesně tak, jak je uvedeno v předchozím odstavci. Ve třetím kroku zasílá aplikace časové razítko zpět na server. V posledním kroku server připojí časové razítko k dokumentu tak, jak je specifikováno v normě pro daný typ dokumentu (PDF/A, XAdES-T, aj.).

Připojení vlastnoručního podpisu k lokálnímu dokumentu

Připojení vlastnoručního podpisu je možné opět pouze pro dokument typu PDF, a to jen v případě, že je zvolen vizuální podpis. Vlastnoruční podpis je implementován pomocí OpenGL a systémových prostředků detekujících polohu doteku prstu na zobrazovacím zařízení. Vlastnoruční podpis je poté spolu s regulérním elektronickým podpisem umístěn do dokumentu jako vizuální podpis. Z uživatelského pohledu je postup podepsání analogický s podepsáním lokálního dokumentu PDF. Proces podepsání je graficky znázorněn v kapitole „Podepsání lokálního dokumentu".

Přeposlání dat z backendu na jakýkoli jiný vzdálený server

Přeposlání dat z backendu na jakýkoli jiný vzdálený server se využívá například při potřebě odeslat data z formuláře do datové schránky. Technologicky není problém tato data poslat z backendu přímo na daný server (v tomto případě do datové schránky), ale uživatel by musel backendu svěřit své přístupové údaje, nebo dokonce svůj certifikát, který pro přístup do datové schránky využívá. To by samozřejmě mohlo být bráno jako potenciální riziko zneužití. Z tohoto důvodu jsou data z webového formuláře nejprve zasílána do aplikace FormApps Mobile a poté jsou bez jakéhokoli čtení nebo uložení přeposílána na cílový server. Cílový server může vyžadovat autentizaci. V tuto chvíli jsou v aplikaci podporovány tři druhy autentizace: basic, klientským certifikátem a NTLM. Veškerá komunikace probíhá prostřednictvím zabezpečeného protokolu https.

Aplikace FormApps Mobile má možnost uložit až dvě kombinace přihlašovacího jména a odpovídajícího hesla pro přístup do datové schránky. Pokud je má uživatel uloženy, nemusí je již při posílání webového formuláře zadávat. Přihlašovací údaje jsou uchovávány v keychainu aplikace.

Vyfocení fotografie do formuláře

Webový formulář si může jako přílohu vynutit mimo jiné i pořízení fotografie fotoaparátem zařízení. Princip je analogický jako v případě vložení regulérní přílohy z lokálního úložiště nebo z Google Drive. Vyfocenou fotografii je však také možné převést do formátu PDF. Při této konverzi lze nastavit několik parametrů ovlivňujících výslednou kvalitu a velikost jak samotného obrázku, tak samozřejmě i výsledného PDF. Jsou to: výsledná velikost obrázku, výsledná kvalita obrázku, možnost převedení barevného obrázku do stupňů šedi. Tyto transformace (tedy transformace nad obrázkem, ne samotnou transformaci obrázku do PDF) provádí v aplikaci knihovna libjpeg.

Načtení 1D nebo 2D kódu do formuláře

Formulář může také požádat aplikaci o data, která jsou zakódována v 1D nebo 2D kódu. Načtení se provádí fotoaparátem zařízení. Aplikace v současné době plně podporuje 1D kódy ve specifikaci EAN-13 a 2D kódy ve specifikaci ISO/IEC 18004:2006.

Tato funkce je samozřejmě dostupná pouze na zařízeních, která mají k dispozici fotoaparát (toto omezení se dotýká například zařízení iPad 1, které není vybaveno fotoaparátem).

Vkládání příloh a úložiště dokumentů

Vzhledem k tomu, že iOS z principu nepodporuje „standardní" filesystem, a tedy ani práci se soubory ve svém mobilním prohlížeči Safari, bylo potřeba tuto funkčnost implementovat ve FormApps Mobile. FormApps Mobile proto obsahuje prohlížeč souborů a umožňuje jakýkoli soubor, ke kterému má přístup, vložit do webového formuláře, který tuto funkci podporuje. Lokální soubory jsou ukládány do lokálního úložiště aplikace, ke kterému je přístup z aplikace iTunes.

Soubory v úložišti Google Drive jsou viditelné až po povolení a přihlášení do Google Drive v nastavení aplikace. Autentizace ke Google Drive je prováděna pomocí OAuth2.

Vkládání příloh z technologického hlediska probíhá následovně: Nejprve si webový formulář vyžádá vložení přílohy. V reakci na to se uživateli zobrazí dialog, umožňující vybrat konkrétní soubor. Tento soubor je poté převeden do formátu base64, zabalen do SOAP obálky a zaslán zpět na backend.

Prohlížeč souborů navíc také podporuje tzv. „long tap" na jednotlivé soubory. Po provedení „long tapu" jsou uživateli k dispozici další možnosti práce s dokumenty, jako například odeslání vybraného dokumentu pomocí e-mailu nebo jeho smazání.

Generování certifikátu s využitím SecuStamp CA

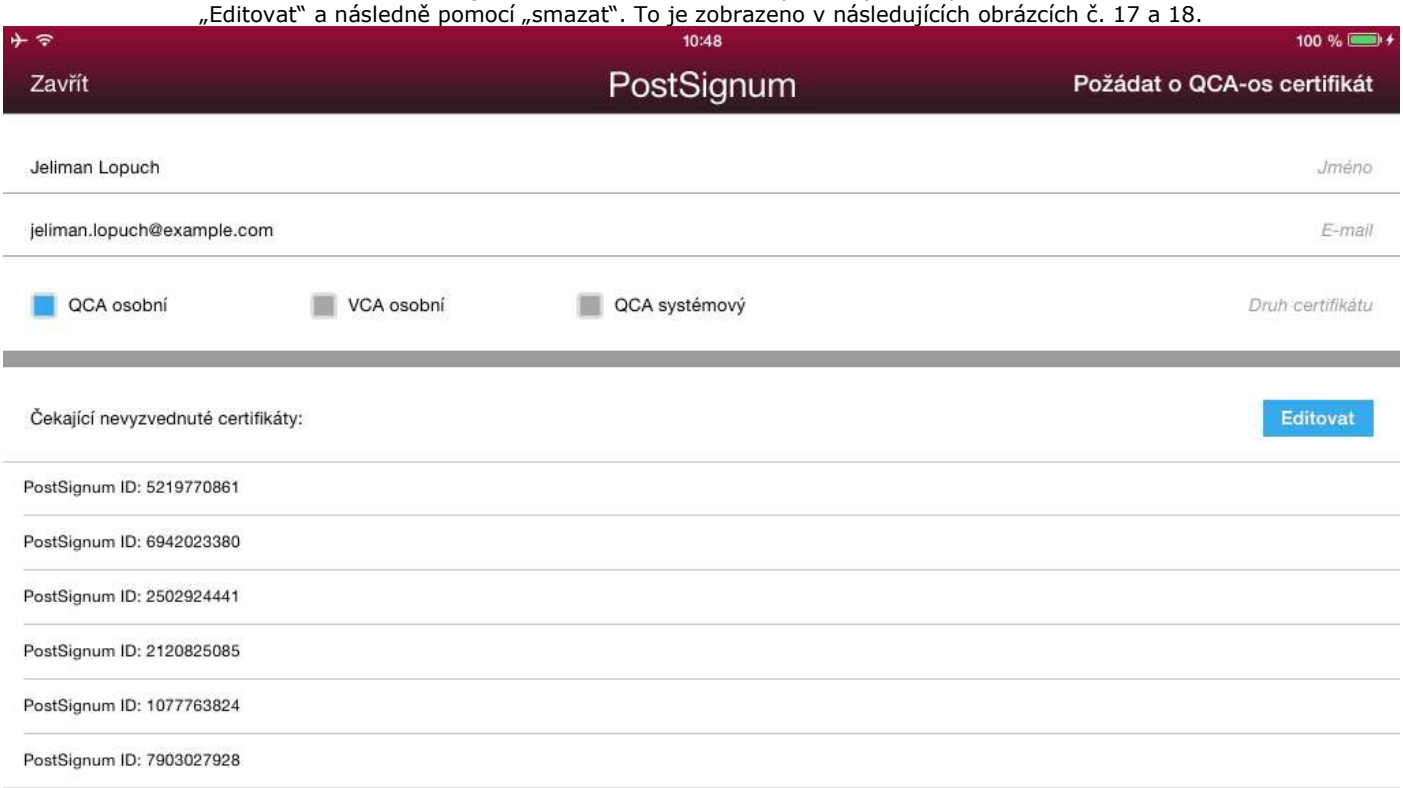
Aplikace FormApps Mobile umožňuje generování certifikátů prostřednictvím certifikační autority SecuStamp CA. Technologicky je celý proces následovný: v prvním

kroku je pomocí OpenSSL vygenerována dvojice privátní-veřejný klíč. Ve druhém kroku je vytvořena PKCS10 žádost o certifikát, která je podepsána právě tímto privátním klíčem. PKCS10 žádost je poté převedena do formátu base64, zabalena do SOAP obálky a poslána webové službě SecuStamp. Webová služba po zpracování požadavku vystaví certifikát a pošle jej zpět do aplikace. Aplikace se zeptá na heslo pro uložení privátního klíče. SHA256 hashem z tohoto hesla zašifruje privátní klíč pomocí AES 256 v módu OFB a spolu se získaným certifikátem jej uloží do keichainu aplikace.

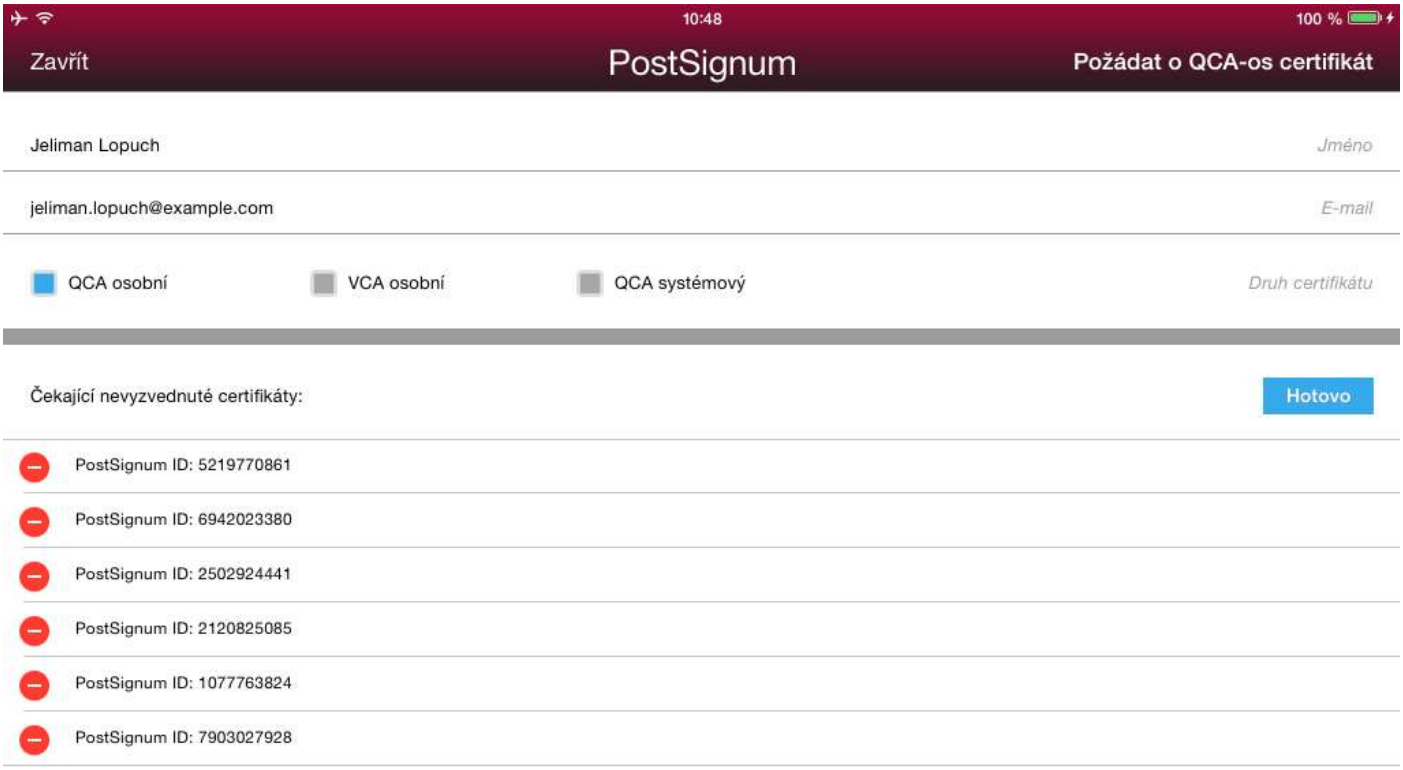
Generování komerčních a kvalifikovaných (osobních i systémových) certifikátů PostSignum.

Aplikace FormApps Mobile umožňuje také generování certifikátů od certifikační autority PostSignum. A to nejen komerčních, ale dokonce také kvalifikovaných. Postup generování certifikátů PostSignum je následovný: V prvním kroku je pomocí OpenSSL vygenerována dvojice privátní-veřejný klíč. Ve druhém kroku je vytvořena PKCS10 žádost o certifikát. Tato žádost je převedena do base64 formátu, vložena do SOAP obálky a odeslána na server PostSignum. Ten v případě úspěchu vrací ID žádosti. Toto ID žádosti se v aplikaci interně páruje s privátním klíčem a s SHA1 hashem veřejného klíče (často také nazývaného SKID). S tímto ID žádosti je potřeba se osobně dostavit na pobočku České pošty. Zde po zkontrolování totožnosti pracovníkem pošty nastaví tento v systému příznak, že certifikát je možné vydat. Po replikaci záznamů v databázi PostSignum, trvající nejdéle 3 minuty od schválení požadavku o certifikát, je již možné certifikát pomocí aplikace vyzvednout. Vyzvednutí samotné probíhá následovně: Aplikace při každém svém spuštění a při každém zobrazení dialogu „PostSignum" kontaktuje server PostSignum a zjišťuje, zda je již připraven ke stažení certifikát, ke kterému je v aplikaci odpovídající SKID. Aplikace tedy pošle na server PostSignum seznam všech SKID, které má k dispozici. Počet „čekajících" certifikátů a tedy celkový seznam SKID není nijak aplikačně omezen. Pokud byl některý SKID serverem vyhodnocen jako „připravený", zasílá server zpět aplikaci již přímo vydaný certifikát. Aplikace se poté uživatele zeptá na heslo pro privátní klíč a poté jej spolu s přijatým certifikátem uloží naprosto shodným způsobem, jako v případě generování certifikátu prostřednictvím SecuStamp CA.

Libovolnou „čekající" žádost může uživatel kdykoliv vymazat pomocí tlačítka „Editovat" a následně pomocí „smazat". To je zobrazeno v následujících obrázcích č. 17 a 18.



Obrázek 17: Dialog "PostSignum" se seznamem žádostí o certifikát, které čekají na vyřízení, zdroj: Autor



Obrázek 18: Dialog "PostSignum" po stisku tlačítka "Editovat", které umožňuje smazat vybrané žádosti o certifikát, zdroj: Autor