Mendelova univerzita v Brně
Provozně ekonomická fakulta

Mobilná aplikácia pre elektronickú evidenciu tržieb

Bakalárska práca

Vedúci práce:
Ing. Petr Jedlička, Ph.D.

Peter Osuský

Brno 2017

Rád by som poďakoval pánovi Ing. Petrovi Jedličkovi, Ph.D. za vedenie práce, ochotu a cenné rady pri vývoji aplikácie. Ďakujem firme RedHat za spoluprácu, ochotu a cenné praktické rady pri implementácii aplikácie, ako aj za zapožičanie tlačiarne. Ďakujem tiež mojej rodine a kamarátom za poskytnutú podporu počas písania práce.

Čestné prehlásenie

Prehlasujem, že som prácu: **Mobilná aplikácia pre elektronickú evidenciu tržieb**
vypracoval samostatne a všetky použité zdroje a informácie uvádzam v zozname použitej literatúry. Súhlasím, aby moja práca bola zverejnená v súlade § 47b zákona č. 111/1998 Sb., o vysokých školách v znení neskorších predpisov a v súlade s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*
Som si vedomý, že sa na moju prácu vzťahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzatvorenie licenčnej zmluvy a použitie tejto práce ako školského diela podľa § 60 odst. 1 Autorského zákona.
Ďalej sa zaväzujem, že pred spísaním licenčnej zmluvy o použití diela inou osobou (subjektom) si vyžiadam písomné stanovisko univerzity, že predmetná licenčná zmluva nie je v rozpore s oprávnenými záujmami univerzity a zaväzujem sa uhradiť prípadný príspevok na úhradu nákladov spojených so vznikom diela, a to až do ich skutočnej výšky.

V Brne dňa 14.5. 2017 .....

Abstract

Osuský, P. Mobile application for electronic records of revenues. Bachelor thesis. Brno, 2017.

The thesis is focused on the design and implementation of a mobile application for electronic records of revenues on mobile devices running iOS operation system. Part of the thesis is a description of the electronic records of revenues as well as an overview and comparison of available applications built for this purpose. The main goal is to enable users to electronic records of revenues on iPhone smartphones via an app. The registration will take place in the form of exchange of data messages between the mobile device and the Financial Administration of the Czech Republic and the subsequent printing of receipts. Electronic records of revenues is expected to be more effective in tax administration.

Abstrakt

Osuský, P. Mobilná aplikácia pre elektronickú evidenciu tržieb. Bakalárska práca. Brno, 2017.

Práca je zameraná na návrh a implementáciu mobilnej aplikácie pre elektronickú evidenciu tržieb na mobilných zariadeniach s operačným systémom iOS. Súčasťou práce je popis problematiky elektronickej evidencie tržieb a tiež prehľad a porovnanie dostupných aplikácii pre elektronickú evidenciu tržieb. Hlavným cieľom je cez aplikáciu umožniť užívateľom elektronicky evidovať tržby na smartfónoch iPhone. Evidencia bude prebiehať formou výmeny dátových správ medzi mobilným zariadením a Finančnou správou ČR a následnou tlačou účteniek. Od elektronickej evidencie tržieb sa očakáva efektívnejšia správa daní.

Obsah

1 Úvod

V dnešnej dobe rastie dopyt po mobilných aplikáciách rozširujúcich funkcionalitu smartfónov a tým sa zvyšuje ich využiteľnosť. V 21. storočí sa smer vývoja týchto zariadení zásadne zmenil, už nehovoríme len o prístrojoch na zjednodušenie komunikácie, ale o komplexných zariadeniach slúžiacich takmer na všetko. V obchodoch pre tieto zariadenia sa nachádzajú milióny aplikácií, od hier až po vedecké nástroje. S rýchlym vývojom mobilných aplikácií tiež rastie ich možnosť využitia aj na podnikové účely. Medzi takéto využitie určite patrí aj aplikácia slúžiaca ako pokladňa. Nie každý, kto využíva pri svojej práci pokladňu, je celý deň na jednom mieste.

Práve pre takýchto ľudí by mala slúžiť aplikácia, ktorej vývojom sa budeme zaoberať v tejto práci. Podľa zákona č. 112/2016 Sb., o evidenci tržeb (ďalej len ZoET) totiž každý daňový poplatník bude musieť elektronicky evidovať prijaté platby v hotovosti. Na takýto účel tým pádom potrebuje pokladňu, ktorá umožňuje elektronickú evidenciu tržieb. Prečo by si však mal používateľ kupovať pokladňu, keď mu na to bude stačiť jeho mobilný telefón, ktorý má v tejto dobe skoro každý. Medzi najrozšírenejšie operačné systémy smartfónov sa radí Android a iOS. Android patrí medzi open-source operačné systémy, zatiaľ čo iOS je kompletne uzavretý systém. My sme si zvolili operačný systém iOS, keďže aplikácií pre elektronickú evidenciu tržieb je pre tento systém menej, ako pre Android a naša práca teda bude mať pre ľudí väčší prínos.

1.1 Cieľ práce

Cieľom tejto práce je návrh a implementácia open-source mobilnej aplikácie pre elektronickú evidenciu tržieb na mobilných zariadeniach s operačným systémom iOS. Cieľovou skupinou, pre ktorú aplikácia bude vytvorená sú pre všetky podnikateľské subjekty prechádzajúce na proces elektronickej evidencie tržieb, pre ktorých je povinná podľa ZoET.

Vytvorená aplikácia bude umožňovať nastavenie pokladne, ktoré je nevyhnutné pre úspešnú elektronickú evidenciu tržieb. Ďalej si bude môcť používateľ pridávať jednotlivé ponúkané produkty do listu produktov, aby ich mohol následne pridávať na účtenku. Medzi ďalšie funkcie bude patriť tlač vytvorenej účtenky z mobilného zariadenia na pripojenej tlačiarni.

Za najdôležitejšie výhody vytvorenej aplikácie považujeme jej bezplatnosť, open-source zdrojový kód a jednoduché ovládanie.

2 Literárna rešerš

2.1 Elektronická evidencia tržieb

Elektronická evidencia tržieb je spôsob evidencie tržieb, pri ktorej dochádza k odosielaniu údajov o každej tržbe online na Finančnú správu. Je určená pre všetky podnikateľské subjekty prechádzajúce na proces elektronickej evidencie tržieb, pre ktorých je povinná podľa ZoET. Elektronická evidencia funguje na spôsob výmeny dátových správ. Poplatník odosiela dátovú správu prostredníctvom internetového pripojenia na server Finančnej správy, kde budú informácie spracované a následne bude vygenerovaný unikátny kód. Tento kód je obratom odoslaný obchodníkovi na jeho koncové zariadenie, ktoré kód vytlačí na účtenku. Zákazník si potom môže pomocou tohto kódu overiť na webovej stránke Finančnej správy, či bola tržba zaevidovaná. Neevidujú sa platby bankovým prevodom na účet, inkasom, zloženkou, alebo vkladom hotovosti na účet podnikateľa (Česko, 2016).

2.1.1 Autentizácia a registrácia v EET

Poplatník je povinný pred prijatím prvej evidovanej tržby podať žiadosť o autentizačné údaje do webovej aplikácie pre elektronickú evidenciu tržieb na Daňovom portáli Finančnej správy. Po schválení žiadosti sú mu pridelené autentizačné údaje. Tie sú následne použité pri prihlasovaní na Daňovom portáli, kde poplatník vyplní všetky údaje potrebné k EET. Poplatník potom musí požiadať o vydanie certifikátu, podľa ktorého Finančná správa identifikuje podnikateľa pri zaslaní údajov dátovými správami. Podnikateľ môže používať jeden certifikát na všetkých svojich zariadeniach, alebo môže podľa potreby požiadať aj o väčšie množstvo. Ako posledný krok si musí poplatník stiahnuť aplikáciu na elektronickú evidenciu tržieb do svojho zariadenia, nainštalovať certifikát a môže začať elektronickú evidenciu tržieb (Česko, 2016).

2.1.2 Údaje zasielané finančnej správe (§ 19 ZoET)

Dátová správa vo formáte XML, ktorá je odosielaná na server Finančnej správy musí obsahovať tieto údaje (Česko, 2016):

- daňové identifikačné číslo (DIČ),
- označenie prevádzky, v ktorej je tržba uskutočnená,
- označenie pokladničného zariadenia, na ktorom je tržba zaevidovaná,
- poradové číslo účtenky,
- dátum a čas prijatia tržby, alebo vystavenia účtenky, pokiaľ je vystavená skôr,
- celková čiastka tržby,

- bezpečnostný kód poplatníka (BKP),
- podpisový kód poplatníka (PKP),
- údaj, či je tržba evidovaná v bežnom, alebo zjednodušenom režime.

Za niektorých okolností musia byť v dokumente aj dodatočné údaje (Česko, 2016):

- celková čiastka platieb určených k následnému čerpaniu,
- celková čiastka platieb, ktoré sú následným čerpaním alebo zúčtovaním platby,
- daňové identifikačné číslo poplatníka, ktorý poveril evidovaním tejto tržby poplatníka, ktorý tržbu eviduje,
- základ dane z pridanej hodnoty a daň podľa sadzieb dane z pridanej hodnoty,
- celková čiastka v režimu dane z pridanej hodnoty pre cestovnú službu,
- celková čiastka v režime dane z pridanej hodnoty pre predaj použitého tovaru.

2.1.3 Výmena správ medzi poplatníkom a správcom dane

V dátovej správe sa nachádzajú všetky povinné údaje, definované v predchádzajúcej kapitole. Z údajov DIČ poplatníka, identifikácia prevádzky, identifikácie pokladne, poradové číslo účtenky, dátum a čas prijatia tržby, celková čiastka tržby sa vytvorí elektronický podpis. Z neho sa vytvorí odtlačok pomocou algoritmu SHA-256, ktorý je následne podpísaný algoritmom RSA-2048. Takto vytvorený elektronický podpis tvorí podpisový kód poplatníka (ďalej len PKP). Pre úplnosť dátovej správy sa musí z PKP vytvoriť ďalší odtlačok pomocou SHA1. Tento odtlačok predstavuje bezpečnostný kód poplatníka (ďalej len BKP).

Takto podpísaná dátová správa sa zabalí do formátu SOAP, opäť sa elektronicky podpíše a protokolom HTTPS je odoslaná do webovej služby správu dane. Po prijatí správy od poplatníka sa tržbe pridelí unikátny kód, tzv. fiškálny identifikačný kód (ďalej len FIK) a správca ho odošle v ďalšej dátovej správe poplatníkovi (Česko, 2016).

2.1.4 Režimy evidencie

Tržby je možné elektronicky evidovať v troch rôznych režimoch. Hovoríme o evidencii tržieb v bežnom režime (§ 18ZoET) – „on-line“, evidencii tržieb pri výpadku spojenia (§ 22 ZoET) a evidencii tržieb v zjednodušenom režime (§ 23 ZoET) – „off-line“.

Evidencia tržieb v bežnom režime (§ 18 ZoET) – „on-line“ Podnikateľ je povinný zaslať dátovú správu s potrebnými údajmi o tržbe správcovi dane a vytlačiť zákazníkovi účtenku najneskôr v okamihu uskutočnenia evidovanej tržby.

Za okamih uskutočnenia evidovanej tržby sa považuje okamih prijatia evidovanej tržby, alebo okamih vydania príkazu k jej realizácii, pokiaľ tento okamih nastal skôr.

Podnikateľ má tiež možnosť zaevidovať tržbu v predstihu a účtenku vydať neskôr pri prijatí platby, ale najneskôr pri jej predaní. Túto možnosť môže využiť napríklad pri rozvoze jedla na objednávku. Podnikateľ zaeviduje platbu už v reštaurácii a účtenku vydá zákazníkovi až pri preberaní objednaného jedla. Zákon o evidencii tržieb neupravuje formát ani spôsob predávania účtenky. Podnikateľ môže zákazníkovi poskytnúť účtenku v tlačenej podobe, elektronicky (emailom), alebo iným spôsobom umožňujúcim zákazníkovi s účtenkou zaobchádzať (Česko, 2016).

Postup evidencie v bežnom režime (Česko, 2016):

- podnikateľ zašle dátovú správu o transakcii Finančnej správe,
- systém finančnej správy posiela odpoveď s FIK,
- podnikateľ vystaví zákazníkovi účtenku,
- zákazník prevezme účtenku,
- zákazník si platnosť svojej účtenky môže overiť prostredníctvom webovej aplikácie Finančnej správy.

Evidencia tržieb pri výpadku spojenia (§ 22 ZoET) V prípade technického problému, dočasného výpadku internetového pripojenia, počas ktorého sa nepodarí nadviazať spojenie s nastavenej medznej dobe (zo zákona viac ako dve sekundy), nemusí pokladničné zariadenie čakať na odozvu serveru Finančnej správy. Podnikateľ v tomto prípade vystaví zákazníkovi účtenku bez fiškálneho identifikačného kódu. Namiesto neho bude na účtenke vytlačený podpisový kód poplatníka.

Doba odozvy (§ 21 ZoET) je časový úsek medzi pokusom o odoslanie údajov o zaevidovanej tržbe a prijatím fiškálneho identifikačného kódu späť na pokladničné zariadenie podnikateľa. Medznú dobu by mal podnikateľ nastaviť podľa kvality internetového pripojenia tak, aby bola väčšina tržieb zaevidovaná on-line. Akonáhle sa spojenie obnoví, podnikateľ dátovú správu opäť odošle, najneskôr však do 48 hodín od zrealizovania evidovanej tržby (Česko, 2016).

Postup evidovania tržieb pri výpadku spojenia (Česko, 2016):

- podnikateľ zašle dátovú správu o transakcii Finančnej správe,
- systém Finančnej správy nepošle odpoveď s fiškálnym identifikačným kódom,
- podnikateľ vystaví zákazníkovi účtenku, namiesto FIK však obsahuje PKP,
- po obnove spojenia odošle podnikateľ opätovne dátovú správu,
- systém Finančnej správy posiela odpoveď aj s FIK.

Evidencia tržieb v zjednodušenom režime (§ 23 ZoET) „off-line“ Tento režim slúži podnikateľom, ktorí nemajú možnosť mať stále internetové pripojenie,

alebo majú výnimku zo zákona. Jedná sa o evidenciu tržieb uskutočnených v pravidelnej hromadnej doprave osôb (§ 10 ZoET), alebo tržby, ktoré povolil evidovať v zjednodušenom režime správca dane na podnet podnikateľa (§ 11 ZoET).

Ak je tržba evidovaná v zjednodušenom režime, podnikateľ nie je povinný na účtenke udávať FIK, namiesto neho účtenka musí obsahovať PKP. Zároveň je podnikateľ nútený tieto tržby odoslať správcovi dane najneskôr do 5 dní od zaevidovania tržby (Česko, 2016).

Postup evidovania tržieb pri využívaní zjednodušeného režimu (Česko, 2016):

- podnikateľ vystaví zákazníkovi účtenku, namiesto FIK je uvedený PKP,
- do piatich dní podnikateľ zašle elektronicky evidované tržby,
- finančná správa zasiela potvrdenie o prijatých dátových správach.

2.1.5 Povinné údaje uvádzané na účtenke (§ 20 ZoET)

Nielen dátová správa má svoje povinné položky, účtenka musí taktiež obsahovať niekoľko povinných údajov. V prípade, že má podnikateľ možnosť neuvádzať na účtenke FIK, je povinný uviesť PKP (Česko, 2016).

Povinné údaje na účtenke (Česko, 2016):

- fiškálny identifikačný kód (FIK),
- daňové identifikačné číslo (DIČ) poplatníka,
- označenie prevádzky, v ktorej bola tržba uskutočnená,
- označenie pokladničného zariadenia, na ktorom bola tržba zaevidovaná,
- poradové číslo účtenky,
- dátum a čas prijatia tržby,
- celková čiastka tržby,
- bezpečnostný kód poplatníka (BKP),
- informácia, či bola tržba evidovaná v bežnom, alebo zjednodušenom režime.

2.1.6 Existujúce riešenia

V prieskume existujúcich riešení sme sa zamerali na mobilné aplikácie pre operačný systém iOS poskytujúce elektronickú evidenciu tržieb. Niektoré aplikácie sú však spoplatnené a na ich reálne vyskúšanie sme nemali finančné prostriedky. Tieto aplikácie budeme hodnotiť podľa informácií, ktoré uvádzajú ich poskytovatelia na svojich webových stránkach.

ePokladna od spoločnosti Vodafone ponúka veľmi komplexné riešenie pre EET. Je možné ju využívať na smartfóne, ale aj tablete. Podporuje aj anglický a vietnamský jazyk, taktiež automaticky zálohuje vystavené účtenky, či nastavenia do cloudového úložiska. Veľkou výhodou je aj práca v offline móde ak je zariadenie na určitý čas odpojené od internetu, po opätovnom pripojení budú všetky tržby dodatočne zaevidované.

Spoločnosť ponúka 3 rôzne balíčky. Základný ponúka všetky vyššie spomínané vlastnosti. Stredný balíček ponúka naviac napríklad 1000 obľúbených položiek pre rýchly predaj, či hromadné nahrávanie položiek z počítača. Najvyšší balíček obsahuje správu zliav, skladových zásob, dodávateľov, či inventúru. Cena za najvyšší balíček bola ku dňu 8.3.2017 347Kč, najlacnejší balíček stál 149Kč s DPH mesačne. (Vodafone, 2017).

eKasa je aplikácia pre elektronickú evidenciu tržieb od spoločnosti O2. Je rozdelená do 2 balíčkov. O2 Mobilná eKasa Air je určená pre telefóny s Androidom 5.0 a vyššie, verzia O2 eKasa Air je určená pre tablety s uhlopriečkou aspoň 7“. Výhodou je, že používateľ nemusí platiť zriaďovaciu cenu, súčasťou balíčku je totiž len samotná aplikácia. Mesačné poplatky sú u prvej varianty 349 Kč a u druhej 449 Kč bez DPH. Aplikácia tiež podporuje anglický a vietnamský jazyk. Veľkou nevýhodou je však podpora zariadení len s operačným systémom Android (O2, 2017).

Ďalšou výbornou aplikáciou je **LILKA** od spoločnosti ArrowSys. Je možné ju využívať na operačnom systéme Android, iOS, aj Windows Phone. Zriaďovacia cena je 899 Kč bez DPH a mesačné poplatky sú nulové. Cenová hladina teda zaraďuje tento produkt medzi tie najlacnejšie. Aplikácia pritom ponúka nadväznosť na skladové a účtovné systémy. Výhodou je aj ľubovoľné rozšírenie o periférie ako čítačka čiarových kódov, alebo zásuvka na hotovosť. O univerzálnosti aplikácie svedčí aj fakt, že je možné používať ju na PC, tablete aj smartfóne.

Túto aplikáciu sme mali možnosť odskúšať aj na mobilnom telefóne, keďže je za určitých podmienok ponúkaná zdarma. Aplikácia obsahuje podobné funkcie ako vyššie dve spomínané varianty. Výhodou je možnosť evidovať vo všetkých režimoch elektronickej evidencie tržieb, prehľadná história tržieb, alebo možnosť uzavretia kasy. Sporným bodom je však zložitosť aplikácie, keďže menej zdatný používateľ nemusí vedieť ako ju obsluhovať. Ako najväčšie mínus by sme určite zvolili neprehľadnosť a jej občasné zasekávanie (ArrowSys, 2017).

Zaujímavé riešenie ponúka aplikácia **MAXiPokladna**. V najnižšom balíčku je úplne zadarmo, podmienkou je však evidovať menej ako 50 účteniek mesačne. Tento produkt je veľmi vhodný pre remeselníkov a drobných živnostníkov. Aplikácia podporuje rôzne druhy platieb, napr. hotovostná, kreditnou kartou, stravnými lístkami, či šekom. Veľmi dobrou funkciou je aj rozdelenie sortimentu do vlastných skupín. MAXiPokladna tiež dokáže zdieľať jednu tlačiareň medzi viacerými smartfónmi. Ak používateľ prekročí limit 50 zaevidovaných platieb, mesačne zaplatí 99 Kč. Medzi plusy tohto riešenia patrí aj mesačná platba len za mesiace, v ktorých bola vykonaná elektronická evidencia tržieb (Mobile APP, 2017).

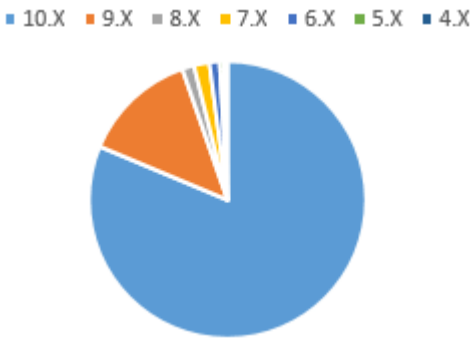
2.2 Operačný systém iOS

iOS, predtým nazývaný iPhone OS, je mobilný operačný systém vytvorený a vyvíjaný spoločnosťou Apple Inc. exkluzívne pre ich hardvér. Konkrétne sa jedná o mobilné zariadenia iPhone, iPad a iPod Touch. Predstavený bol v roku 2007 na konferencii MacWorld spolu s telefónom iPhone. Po Androide je to druhý najpopulárnejší mobilný operačný systém.

Je založený na operačnom systéme Darwin s jadrom XNU a patrí do kategórie unix-like systémov. Je zložený zo štyroch vrstiev: Core OS, Core Services, Media a Cocoa Touch. Aplikácie teda nepracujú s hardvérom priamo, namiesto toho komunikujú s množinou systémových rozhraní. To umožňuje vytváranie aplikácií pracujúcich konzistentne aj na zariadeniach s odlišným hardvérom. Systém je napísaný v programovacích jazykoch C, C++, Objective-C a Swift. Je plne uzavretý, čiže zdrojové kódy nie sú verejne dostupné. Vďaka tomu je iOS považovaný za bezpečný operačný systém (Wikipedia, 2017).

2.2.1 Fragmentácia platformy

Veľkým plusom zariadení s operačným systémom iOS je prechod na nové verzie. Až 81 percent zariadení má nainštalovanú najnovšiu verziu 10, tým pádom je riešenie spätnej kompatibility so staršími verziami takmer nepotrebné. iOS 9 používa stále 13,4 percenta zariadení, a verzie 8-4 sú nainštalované celkovo len na 5,3 percent zariadení (Smith, 2017).



Obrázok 1: Fragmentácia platformy iOS. Zdroj: (Smith, 2017)

2.2.2 Architektúra

Cocoa Touch

Táto časť systému je zložená z najdôležitejších aplikačných rámcov pre vývoj aplikácií. Vo vrstve nájdeme infraštruktúru pre grafické rozhranie a interakciu s používateľom. Taktiež sa stará o multitasking, ktorý bol do iOS zavedený od verzie 4. Táto technológia zabezpečuje chod aplikácií aj na pozadí. Bez multitaskingu sa aplikácie po stlačení domovského tlačidla ukončili.

Vrstva tiež podporuje funkcie na šifrovanie dát. Tie sa využívajú v aplikáciách, ktoré ukladajú citlivé dáta. Následne je takýto súbor označený ako chránený a uložený v zašifrovanej forme. Ak je zariadenie uzamknuté, súbor je nedostupný, dešifrovaný bude až po odomknutí zariadenia. Cocoa Touch má na starosti aj notifikácie. Tento nástroj informuje používateľa o udalostiach aj keď nie je daná aplikácia spustená. Súčasťou vrstvy je aj podpora externých zobrazovacích zariadení. Aplikácia môže byť otvorená na externom zariadení, zatiaľ čo v druhom okne je zobrazený iný obsah, alebo je využitá funkcia zrkadlenia, kedy je rovnaký obsah zobrazený na primárnom aj sekundárnom zariadení (Mark a LaMarche, 2010).

Media

Vrstva Media obsahuje grafické, zvukové a video technológie používané na implementáciu multimédií v aplikácii. Laicky povedané, technológie v tejto vrstve zabezpečia, aby aplikácia vyzerala a znela dobre. iOS podporuje napríklad tieto audio a video formáty: AAC, ALAC, MP3, IMA4, MP4. Súčasťou tejto vrstvy je aj technológia AirPlay, ktorá umožňuje streamovanie audio a video obsahu na Apple TV, alebo prehrávanie hudby na reproduktoroch tretej strany (Mark a LaMarche, 2010).

Core services

Ako už názov napovedá, táto vrstva obsahuje dôležité systémové služby pre aplikácie. Tieto služby poskytujú hlave rozhrania Core Foundation a Foundation. Core services tiež umožňuje využívanie polohy, služby iCloud, alebo sociálnych sietí. Nemenej dôležitou súčasťou je aj služba Peer-to-Peer (klient-klient), spolupracujúca s technológiou Bluetooth. Táto časť systému je primárne využívaná v hrách, no môže byť použitá aj na zahájenie komunikácie medzi zariadeniami nachádzajúcimi sa v blízkosti. Core services obsahuje podporu zdieľania súborov medzi aplikáciou iTunes od verzie 9.1. Dôležitým komponentom tejto vrstvy je podpora formátu XML, ktorý bude vo vlastnej práci často využívaný (Mark a LaMarche, 2010).

Core OS

Táto vrstva obsahuje nízko-úrovňové funkcie, na ktorých je postavená väčšina ostatných technológií. Ak sa teda vyskytne situácia vyžadujúca komunikáciu s externým hardvérom, alebo osobitné riešenie bezpečnosti, využijú sa aplikačné rámce obsiahnuté vo vrstve Core OS. Ďalšou časťou je framework Accelerate, obsahujúci rozhrania pre spracovanie digitálneho signálu. Pri používaní tohto aplikačného rámca nemusí programátor riešiť efektívny chod aplikácie na odlišnom hardvéri, o to sa totiž stará framework. Funkciou Core OS je aj riešenie podpory pre externé

zariadenia cez funkcie aplikačného rámca External Accesory. Tieto zariadenia sú so smartfónom prepojené cez 30 pinový dock konektor, alebo technológiou bluetooth. Framework následne zabezpečuje výmenu informácií, či riadenie takéhoto príslušenstva pomocou príkazov, ktoré sú podporované samotným externým zariadením (Mark a LaMarche, 2010).

2.3 Vývoj aplikácií

V poslednej dobe vlastní mobilné zariadenie takmer každý človek na svete. Vďaka tomuto faktu napreduje vývoj týchto zariadení míľovými krokmi. Medzi najväčších hráčov na trhu sa radí Apple s ich iOS, Google so systémom Android. Microsoft sa s ich Windows Phone snažil taktiež preraziť, no projekt nenaplnil ciele a v priebehu roka 2017 bol oznámený koniec platformy.

V práci sme sa zaoberali vývojom aplikácie na operačný systém iOS. Na to potrebuje vývojár zariadenie so systémom OS X a vývojársky nástroj na to určený. V prvom rade sa však vývojár musí rozhodnúť, s akým programovacím jazykom bude pracovať. Pre iOS sú k dispozícii jazyky Objective-C a Swift. Lepšou voľbou sa javí druhý menovaný, keďže je to nový moderný jazyk a je doporučený aj spoločnosťou Apple. Dá sa teda očakávať, že jeho vývoj bude pokračovať a Objective-C bude podporovaný čoraz menej. Na čo však vývojár nesmie zabudnúť, sú nové verzie Swift, ktoré so sebou prinášajú neustále obmieňajúce sa funkcie, čo má za následok potrebu starať sa o existujúce aplikácie a ich úpravu na nové verzie.

V súčasnosti existujú dva nástroje pre programovanie na iOS v jazyku Swift, jedná sa o AppCode od českej spoločnosti JetBrains (JetBrains, 2017), ktorý je však nutné zakúpiť, alebo Xcode, ktorý je zdarma dostupný na AppStore a ponúka ho samotná spoločnosť Apple. Posledným krokom je výber vývojárskeho programu, Apple ich ponúka niekoľko. Jeden z nich je zadarmo a na jeho používanie stačí AppleID, nevýhodou však je nemožnosť ponúkať vytvorené aplikácie na AppStore. Ostatné sú za poplatok od 99 do 299 dolárov (Apple, 2017a).

2.3.1 Swift

Programovací jazyk Swift bol predstavený v roku 2014 na konferencii WWDC. V súčasnosti sa používa už jeho tretia verzia. Tento jazyk poskytuje alternatívu k Objective-C. Swift by mal byť modernejší a syntakticky jednoduchší ako Objective-C. Nepoužíva ukazovatele a zamedzuje tvorbu nebezpečného kódu. Premenné sú totiž inicializované ešte pred ich použitím, alebo polia nemôžu pretiecť.

Ďalším bezpečnostným ukazovateľom je aj fakt, že objekty v jazyku Swift nemôžu nikdy nadobudnúť hodnotu nil. Kompilátor v takomto prípade zastaví činnosť. V niektorých triedach však vývojár musí pracovať aj s objektami ktoré majú hodnotu nil, v tomto prípade sa používa takzvaná funkcia optionals, ktorá povoľuje objektu nadobudnúť aj takúto hodnotu. Zároveň však upozorňuje vývojára, aby s takýmto objektom pracoval opatrnejšie. Takto označené objekty majú za svojim

názvom uvedený znak otáznik. Zaujímavosťou taktiež je, že za koncom príkazu nie je nutné písať bodkočiarku (Apple, 2017c).

2.3.2 Xcode

Jedným z mála vývojárskych prostredí na tvorbu aplikácií pre operačný systém iOS je v dnešnej dobre Xcode. Výstupom môžu byť aplikácie pre zariadenia Mac, iPhone, iPad, Apple TV, či Apple Watch. Prostredie je dostupné v obchode App Store a prvá verzia bola vydaná v roku 2003, dnes je dostupná už v poradí ôsma verzia. Xcode podporuje jazyky C, C++, Objective-C, Objective-C++, Java, Swift a mnoho ďalších.

Nástroj sa skladá z niekoľkých oddelených súčastí. Hlavnou je integrované vývojárske prostredie Xcode. To obsahuje veľkú časť vývojárskej dokumentácie a vstavaný Interface Builder používaný na konštruovanie grafického používateľského rozhrania. Prostredie zabezpečuje vývojárom jednotný pracovný postup pre vytváranie grafického rozhrania, kódovanie, testovanie a ladenie. Na písanie kódu sa používa editačná oblasť. Navigačná oblasť obsahuje štruktúru jednotlivých súborov a oblasť s nástrojmi poskytuje prácu s triedami Cocoa Touch, alebo atribútmi. Súčasťou prostredia je v neposlednej rade aj debugger a konzola. Nevýhodou však je možnosť použitia Xcode len na zariadeniach s OS X, iné nie sú podporované (Apple, 2017d).

2.3.3 iOS simulátor

Na simuláciu vytvorenej aplikácie môže vývojár používať náhradu za reálne zariadenie. Tento nástroj je súčasťou prostredia Xcode a umožňuje skúšať aplikáciu na virtuálnych zariadeniach s operačným systémom iOS, watchOS, či tvOS. Funkcionalita takýchto zariadení je však značne obmedzená, ako napríklad používanie vnútornej pamäte. Pre základné funkcie je však postačujúci. Aplikácia sa však na reálnom zariadení môže správať inak, je preto veľmi vhodné simulátor používať len v nevyhnutných prípadoch. Na druhú stranu však vývojár môže používať tento nástroj aj bez nutnosti vytvorenia vývojárskej licencie (Knott, 2014).

2.4 Základné knižnice

2.4.1 Používateľské rozhranie

Apple vo svojej dokumentácii ponúka zásady a smernice, ktorých by sa mal vývojár pri navrhovaní aplikácie držať. Jedná sa o obecné vzory pre navrhovanie používateľského rozhrania a framework *Cocoa Touch*, ktorý pod sebou združuje čiastkové frameworky starajúce sa o toto rozhranie. *Cocoa Touch* napríklad poskytuje aplikácii multi-tasking, či rozpoznávanie gest (Knott, 2014).

2.4.2 Ukladanie dát

Predpokladáme, že aplikácia určená k elektronickej evidencii tržieb bude musieť byť schopná ukladať dáta rôzneho druhu. Jednať sa môže o základné nastavenia, alebo používateľské dáta. Druhá menovaná skupina dát môže mať v určitých prípadoch väčší objem a na takéto záležitosti je vhodné používať framework *Core Data*. Ten ukladá údaje do vytvorenej databázy, ktorú si vývojár napred musí navrhnúť. Pre ukladanie používateľských dát je vhodné použiť triedu *UserDefaults*, ktorá je súčasťou knižnice *Foundation*. Dáta v tejto triede sa ukladajú a načítavajú pomocou takzvaných kľúčov (Knott, 2014).

2.4.3 Bluetooth

Keďže aplikácia bude fungovať ako virtuálna pokladnica, bude potrebné, aby mohla spolupracovať s tlačiarenským zariadením poskytujúcim pripojenie cez bluetooth. Vzájomné prepojenie zariadení zabezpečuje knižnica *Core Bluetooth*, ktorá bude v aplikácii potrebná (Knott, 2014).

2.4.4 Správa závislostí

Keďže aplikácia bude vyžadovať aj knižnice tretích strán, vývojár bude musieť použiť jeden z nástrojov na správu závislostí. V súčasnosti sú najpoužívanejšie nástroje na takúto úlohu *CocoaPods* a *Carthage* (CocoaPods, 2017).

2.4.5 XML

Elektronická evidencia tržieb používa na prenos dátových správ medzi používateľom a serverom finančnej správy dokumenty vo formáte XML. V aplikácii bude teda nevyhnuté použiť knižnicu na vytváranie a parsovanie tohto formátu. Swift obsahuje framework Foundation, ktorý má v sebe obsiahnutý nástroj na prácu s formátom XML. Tento nástroj je však dostupný len pre operačný systém OS X a je teda nemožné ho použiť na iOS (Knott, 2014).

Z tohto dôvodu je nutné použiť knižnicu tretej strany. Alternatívy sú *KissXML* (Hanson, 2017), alebo *AEXML* (Tadić, 2017), ktoré sú dostupné pod voľnou licenciou. Po preštudovaní dokumentácie sa môže používateľ rozhodnúť, ktorú knižnicu si vyberie. Výhodou KissXML je väčšia komplexnosť ako u *AEXML*, ak by vývojár potreboval zložitejšie funkcie na prácu s formátom XML, pravdepodobne si vyberie *KissXML*.

2.4.6 Šifrovanie

Ďalšou povinnosťou pri elektronickej evidencii tržieb je šifrovanie dát. Šifrovanie a podpisovanie treba vykonávať pomocou algoritmov SHA256, RSA-2048, SHA1 a privátneho kľúču, ktorý je obsiahnutý v certifikáte od finančnej správy (Česko, 2016).

Swift síce obsahuje knižnicu na tieto bezpečnostné funkcie, jedná sa o framework Security. Ako však aj v prípade podpory XML formátu, aj tento framework je dostupný len pre operačný systém *OS X* (Mathias a Gallagher, 2015). V prípade, ktorý bude riešený v tejto práci je ale využívaný *iOS* a pre správny chod aplikácie bude nutné využiť knižnice tretej strany.

Našťastie však takýchto riešení nie je málo a vývojár si môže vybrať hneď z niekoľkých knižníc s voľnou licenciou. Ak však vývojár zoberie do úvahy všetky potrebné algoritmy a dokumentáciu k ponúkaným riešeniam, ostanú mu dve rozumné varianty, *SwiftyRSA* (Scoop Technologies, 2017) a *CryptoSwift* (Krzyżanowski, 2017).

2.4.7 Zhrnutie

Podľa zistených informácii sme sa rozhodli využiť vývojové prostredie *Xcode*, keďže je vyvíjaný spoločnosťou Apple a ponúka najväčší komfort pri vývoji aplikácií pre operačný systém iOS. Medzi programovacími jazykmi sme si zvolili jazyk *Swift*, keďže je oproti *Objective-C* novší, bezpečnejší a spoločnosť Apple ho v poslednej dobe tlačí do popredia. Na tvorbu a prácu s formátom XML sme si vybrali knižnicu *KissXML*, keďže ponúka ako jediná všetky neskôr potrebné funkcie. Na šifrovanie dát zvolíme knižnicu *SwiftyRSA*, pretože oproti jedinej rozumnej variante v podobe knižnice *CryptoSwift* ponúka prvá menovaná aj šifrovanie pomocou algoritmu RSA-2048, ktorý bude v našej aplikácii potrebný.

3 Metodika

Vďaka dostatočnej rešerši bolo možné pristúpiť k výberu najvhodnejších metód pre splnenie zadanej úlohy. Dôležitým krokom bol výber vývojového prostredia. Kvôli komplexnosti a dostupnosti sme si vybrali prostredie *Xcode* v aktuálnej verzii 8. Druhou nutnou podmienkou ešte pred začatím implementácie bol výber programovacieho jazyka. Spoločnosť Apple začala dávať do popredia jazyk *Swift*, ktorý je modernejší a bezpečnejší ako alternatíva v podobe *Objective-C*. Na základe týchto faktov sme si vybrali *Swift* vo verzii 3.0.

3.1 Grafické rozhranie

Ešte pred samotným vyvíjaním aplikácie si musíme rozmyslieť, čo bude používateľ od aplikácie vyžadovať a akým spôsobom mu dokážeme jeho prácu uľahčiť. Dôležitým faktorom je aj interakcia a vzhľad, ktorý zaujme už na prvý pohľad. Až po vyjasnení týchto náležitostí môže vývojár začať s vývojom grafického prostredia. Apple na svojich vývojárskych stránkach ponúka moderné riešenia na tvorbu grafických používateľských rozhraní a tie si v tejto kapitole rozoberieme. Spoločnosť dbá hlavne na tieto tri požiadavky (Apple, 2017b):

- Jasnosť – v celom systéme je text čitateľný vo všetkých veľkostiach, ikony sú presné, jasné a ozdoby nerušia používateľa. Dôležitý obsah je jemne zvýraznený a ponúka interaktivitu.
- Zrozumiteľnosť – plynulý pohyb a ostrosť napomáha používateľom porozumieť a pracovať s obsahom. Obsah vyplňuje celú obrazovku, rámy a tiene udržiavajú aplikáciu vzdušnú a zároveň zabezpečujú, že je obsah prvoradý.
- Hĺbka – výrazné vizuálne vrstvy a realistický pohyb sprostredkováva hierarchiu a uľahčuje porozumenie aplikácie. Dotyk umožňuje prístup k funkciám a ďalšiemu obsahu bez straty kontextu. Ako používateľ prechádza obsahom, prechody poskytujú dojem hĺbky.

Z týchto odporúčaní vyplýva na čo by si mal vývojár dať pozor, aké techniky použiť a akým sa naopak vyhnúť. Apple odporúča vyhnúť sa farebným prechodom, či tieňom a vyzdvihuje jednoduchý dizajn. Taktiež si myslí, že je vhodné používať dostatočný odstup medzi tlačidlami, aby bolo zamedzené nechceným kliknutiam.

Dôležitým bodom je aj spôsob prechodov medzi jednotlivými oknami aplikácie. Používateľovi by mali hierarchickým spôsobom ujasňovať, kde sa práve nachádza. Vhodným spôsobom treba narábať aj s gestami, tie môžu byť napríklad použité pre prácu s položkami v liste produktov a na účtenke.

Aplikácia bude používaná na zariadeniach s rozličnou veľkosťou a rozlíšením obrazovky a s tým je potrebné prispôsobiť rozmiestnenie jednotlivých grafických prvkov. Slúžiť nám na to budú nástroje vo vývojovom prostredí *Xcode*, ktoré takéto funkcie zabezpečujú.

3.2 Tvorba XML dokumentu

Údaje o tržbách budú ukladané do dátových správ vo formáte XML, ktoré je nutné nejakým spôsobom vytvoriť. Na to nám poslúži knižnica *KissXML*. Tá totiž zahrňuje všetky funkcie, ktoré pri tomto kroku budeme vyžadovať. Štruktúra XML dokumentu je uvádzaná v dokumentoch k elektronickej evidencii tržieb.

V elemente envelope musí byť uvedený XML signature a certifikát, ktorý bol poplatníkovi pridelený a pomocou ktorého bol XML signature vytvorený. V elemente body sú vypísané údaje o tržbe a poplatníkovi. Niektoré spomínané elementy je však potrebné pred odoslaním tržby zašifrovať a podpísať, tejto činnosti sa budeme venovať v ďalšej podkapitole.

3.3 Šifrovanie a podpisovanie dát

Niektoré položky obsiahnuté v XML dokumente musia byť kvôli bezpečnosti zašifrované. To budú zabezpečovať algoritmy SHA256, SHA1 a RSA-2048. V rešerši sme zistili, že pre systém iOS nie sú vytvorené funkcie na šifrovanie a tak sme boli nútení použiť knižnicu tretej strany. Rozhodli sme sa pre *SwiftyRSA*, keďže alternatíva *CryptoSwift* (Krzyżanowski, 2017) neponúkala šifrovanie pomocou RSA-2048.

Na pripojenie *SwiftyRSA* do projektu v *Xcode* bude využitý nástroj na správu závislostí *CocoaPods*.

3.4 Komunikácia s Finančnou správou

Elektronická evidencia tržieb je uskutočňovaná pomocou výmeny dátových správ medzi poplatníkom a serverom Finančnej správy. V správach sú obsiahnuté všetky náležitosti vyplývajúce zo zákona. Odosielanie a prijímanie správ je vykonávané pomocou protokolu SOAP, ktorý zabezpečuje výmenu XML správ prostredníctvom HTTP.

Tvorba XML správ je riešená pomocou knižnice *KissXML* (Hanson, 2017) spomínanej v predchádzajúcej podkapitole. Vytvorený XML dokument je následne pripravený na odoslanie na server Finančnej správy. V tomto momente bude vytvorený SOAP request, ktorý odošle dátovú správu. V prípade splnenia všetkých podmienok udávaných pre elektronickú evidenciu tržieb bude poplatníkovi zaslaná SOAP response, taktiež v XML formáte. Z tejto správy bude nutné prebrať FIK a BKP. V prípade nedostupnosti servera Finančnej správy, alebo inej technickej chyby je FIK nahradený BKP.

4 Riešenie

4.1 Rozbor konceptu

Základná funkcia aplikácie Pokladnik spočíva vo vytváraní účteniek z ponúkaných produktov a následným elektronickým evidovaním tržby. Používateľ bude musieť najskôr nastaviť svoju pokladňu, aby dátové správy odosielané na server finančnej správy obsahovali všetky náležitosti, ako je napríklad DIČ poplatníka, identifikátor pokladne, alebo prevádzky. Ďalším krokom je vyhľadanie bluetooth tlačiarne, ktorá bude slúžiť na tlač účteniek.

Následne si používateľ v jednoduchom používateľskom prostredí bude musieť vytvoriť ponúkané produkty. Tie budú následne podľa potreby pridávané na účtenku. Ak bude účtenka dokončená, používateľ vyvolá tlač účtenky. Produkty obsiahnuté na tejto účtenke budú z virtuálnej účtenky v aplikácii odstránené a používateľ môže vytvoriť novú účtenku.

4.2 Návrh

Pred samotným implementovaním mobilnej aplikácie sme si museli určiť, pre minimálne akú verziu operačného systému iOS bude vyvíjaná. Podľa fragmentácie platformy, ktorej sme sa venovali v rešerši, sme určili, že posledné tri verzie predstavujú až približne 96 percentný podiel zo všetkých zariadení. Pre aplikáciu Pokladnik sme teda určili minimálnu verziu systému 8.0. Toto rozhodnutie zdôvodňujeme faktom, že sme sa rozhodli používať najnovšiu verziu programovacieho jazyku Swift 3.0, ktorého podpora siaha taktiež len po verziu 8.0.

4.3 Štruktúra aplikácie

Pre vývoj aplikácií na iOS sa používa architektúra MVC, čiže Model, View a Controller a tomu bude uspôsobená aj štruktúra aplikácie. Tá sa skladá z dvoch hlavných štruktúr:

- Pokladnik
- Pods

Zložka Pokladnik je rozdelená na niekoľko ďalších štruktúr:

- Model
- View
- Controller
- Database
- Others

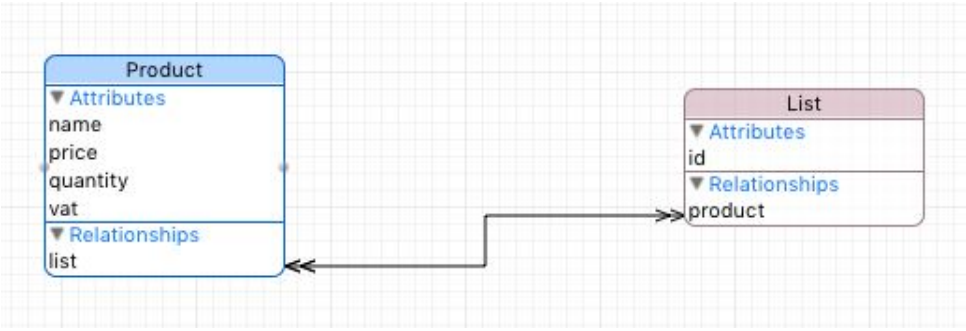
Súbory v podštruktúre v názvom *Model* sa starajú o hlavnú funkčnosť aplikácie a sú nezávislé na ostatných komponentoch, ako to určuje MVC. Súbory v zložke *Controller* poskytujú prepojenie medzi používateľským rozhraním aplikácie a funkciami, ktoré sú naimplementované v modeloch. V štruktúre *View* je obsiahnutý súbor na vykresľovanie vzhľadu aplikácie.

Zložka s aplikáciou taktiež obsahuje databázu, ktorá je vytvorená pre ukladanie dát o produktoch uložených v aplikácii. Triedy poskytujúce kontrolu nad databázou sa nachádzajú v zložke *Database*. Súčasťou aplikácie je aj súbor s koncovkou .app, ten reprezentuje samotnú aplikáciu. Ďalšími súbormi sú frameworky v zložke *Pods*, ktoré sme do aplikácie vložili pomocou správy závislostí *CocoaPods*.

4.4 Databáza

V databázi *Core Data* sú uložené údaje o produktoch uložených v aplikácii a ich naviazanie na vytvorenú účtenku. Keďže z dôvodu vyvinutia jednoduchej aplikácie bude aj databáza vynikať jednoduchosťou. V aplikácii nebude zavedený archív účteniek, pretože túto službu poskytuje samotná Finančná správa na svojom webovom portáli.

Databáza sa skladá z dvoch tried *List* a *Product*. *List* reprezentuje vytvorenú účtenku a *Product* popisuje jednotlivé produkty. Pri produkte musíme ukladať informácie o jeho názve, cene, DPH a počte kusov. Objekt *List* obsahuje len identifikátor.



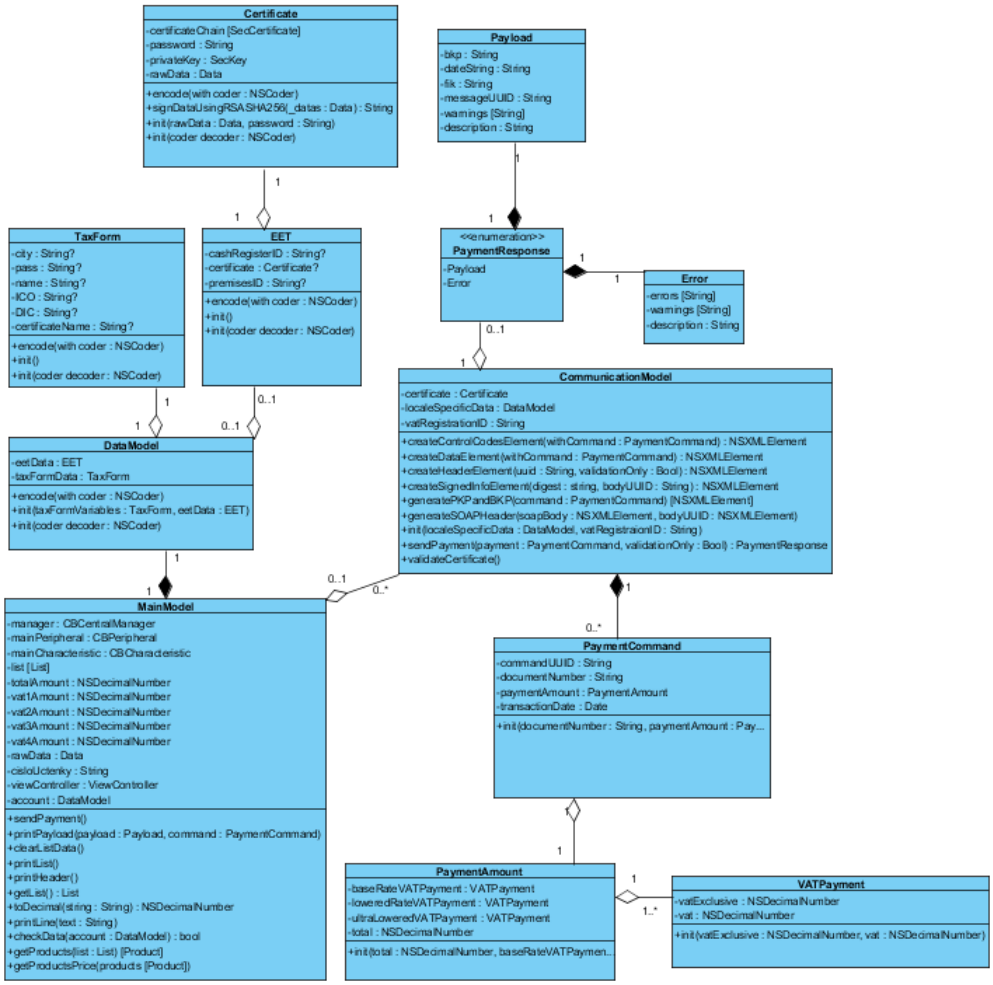
Obrázok 2: Diagram objektov v úložisku

Jednotlivé objekty v databáze sú tvorené dedením z triedy *NSManagedObject*. Tieto kúsky kódov boli automaticky vygenerované po namodelovaní databáze. Nasledujúca ukážka kódu patrí objektu *Product*, ktorý sme si predstavili v predchádzajúcom odstavci. Ten obsahuje spomínané atribúty ako aj funkcie pre pridanie a odobratie produktu z účtenky.

4.5 Model aplikácie

Aplikácia bola navrhovaná podľa MVC, takže obsahuje niekoľko tried, ktoré vytvárajú základný model aplikácie, ten je znázornený v UML diagrame.

- MainModel: jadro aplikácie, ktoré obsluhuje ostatné komponenty modelu
- CommunicationModel: vytváranie XML dokumentu a komunikácia s Finančnou správou
- DataModel: obsahuje informácie o nastaveniach pokladne a certifikáte z Finančnej správy
- TaxForm: vytvára objekt s údajmi o poplatníkovi
- EET: obsahuje objekt s certifikátom a údaje o pokladni a prevádzke
- Certificate: objekt obsahujúci všetky potrebné dáta o certifikáte
- PaymentResponse: odpoveď zo serveru Finančnej správy, obsahuje Payload alebo Error
- PaymentCommand: trieda obsahuje údaje o tržbe (suma, dátum, číslo účtenky)



Obrázok 3: UML diagram modelu aplikácie

Hlavnou triedou modelu je *MainModel*. Ten spravuje a vytvára ostatné triedy a stará sa o celkovú funkčnosť aplikácie. Preberá informácie o pripojenej tlačiarni, používateľských dátach a dátach o tržbe vytvorené v triede *CommunicationModel*. Po úspešnom zaevidovaní tržby sú tieto dáta posielané do pripojenej tlačiarne.

Druhou najdôležitejšou triedou modelu je *CommunicationModel*. Hlavnou úlohou tejto triedy je vytvorenie XML elementov. Vytvorené elementy sú potom spojené a vložené do XML dokumentu. Takto pripravené dáta sú následne vložené do funkcie na prenos SOAP správ medzi aplikáciou a serverom Finančnej správy. Po odoslaní dátovej správy o tržbe vráti server odpoveď s identifikačnými kódmi tržby, alebo chybové kódy v prípade, ak bola tržba nesprávne zaevidovaná.

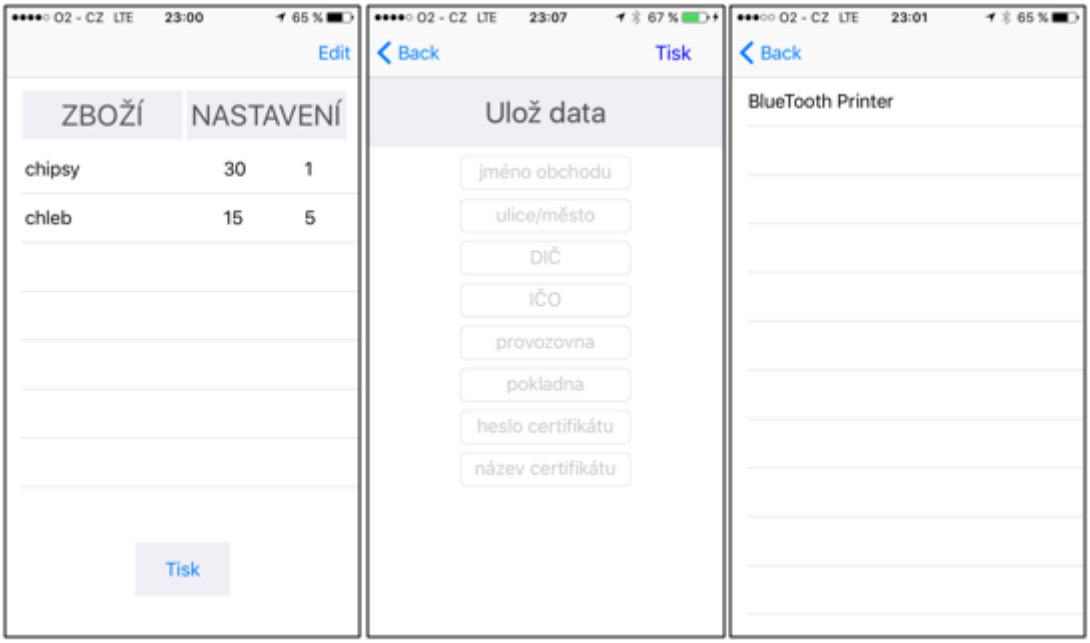
Trieda *DataModel* sa už podľa názvu zaoberá funkciami na správu dát. Obsahuje všetky informácie o prevádzke a taktiež má za úlohu ukladať dáta o certifikáte, či hesle k certifikátu. Dôležitou funkciou je aj podpisovanie dát pomocou algoritmu RSA-2048, či načítanie certifikátu a privátneho kľúču zo zariadenia.

```
public func signDataUsingRSASHA256(_ datas: Data) throws -> String {

            let privateKey2 = try! PrivateKey(reference: privateKey)
            let clear = try ClearMessage(data: datas)
            let signature = try clear.signed(with: privateKey2,
                digestType: .sha256)

            let data = signature.data
            let base64String = signature.base64String

            return base64String
}
```

Určitá časť zdrojového kódu týchto modelov bola prebraná z existujúceho projektu na elektronickú evidenciu tržieb pre zariadenia s operačným systémom *OS X* od používateľa CharlieMonroe (Charlie Monroe Software, 2017a). Mnoho funkcií pripravených na *OS X* však nie je možné použiť pre operačný systém *iOS*. Hovoríme o knižniciach na prácu s XML formátom, či bezpečnostné funkcie používané na šifrovanie dát.

4.6 View

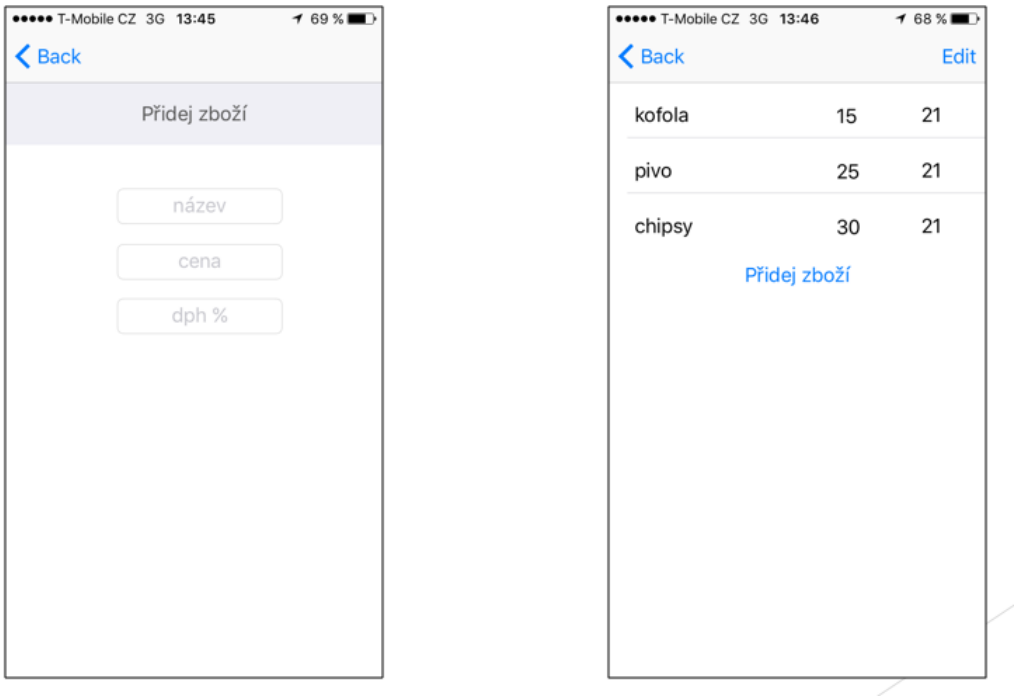
O View sa v *Xcode* stará súbor *Main.Storyboard*. V ňom sme si navrhli všetky obrazovky využívané v aplikácii a prechody medzi nimi. Hlavná obrazovka je *MainView*, tá obsahuje niekoľko tlačidiel a tabuľku. Tlačidlá *Nastavení* a *Zboží* slúžia na prechod do ďalších obrazoviek s nastaveniami a správou produktov. V tabuľke sú zobrazené produkty pridané na účtenku spolu s ich cenou a počtom kusov. Tlačidlo *Tisk* slúži na odoslanie tržby a následnú tlač účtenky.



Obrázok 4: Základná obrazovka, nastavenia a obrazovka s výberom bluetooth zariadenia

Obrazovka Settings slúži na nastavenie aplikácie. Obsahuje textové polia, do ktorých používateľ zadá názov obchodu, adresu sídla, IČO, DIČ, id pokladne, id prevádzky, heslo certifikátu a názov certifikátu. Tlačidlo v ľavej hornej časti slúži na vyhľadanie a následné pripojenie bluetooth tlačiarne.

V obrazovke *ProductsView* sa nachádza tabuľka všetkých ponúkaných produktov. Ak chce používateľ pridať ďalší produkt do tejto tabuľky, po kliknutí na tlačidlo *Přidej zboží* sa aplikácia presunie na obrazovku *ProductAddView*, kde je nutné vyplniť všetky údaje o produkte a následným stlačením tlačidla *Přidej zboží* sa produkt vloží do tabuľky produktov a do databáze.



Obrázok 5: Vytváranie produktov a ich výpis

Prechody medzi jednotlivými pohľadmi spravujú takzvané segue. Aplikácia má celkovo 4 takéto prechody:

- Hlavná obrazovka -> Nastavenia
- Hlavná obrazovka -> Produkty
- Nastavenia -> Výber bluetooth zariadenia
- Produkty -> Vytvorenie nového produktu

Ak sa používateľ chce vrátiť v aplikácii späť, použije tlačidlo *Back* v hornej časti obrazovky, kde sa nachádza navigation bar. Animácie prechodov podporujú hierarchiu a dávajú používateľovi najavo, kde práve sa v aplikácii nachádza.

Aplikácia podporuje aj gestá, tie sú využívané v dvoch prípadoch. Pre vymazávanie produktov z účtenky, alebo zoznamu produktov stačí posunúť prstom doľava po konkrétnom produkte a objaví sa tlačidlo *Delete*.

4.7 Controller

Pre návrh používateľského rozhrania sme použili controllery a nástroj StoryBoard. Controllery podľa MVC slúžia na prepojenie modelu a view. Inak povedané spájajú funkcie aplikácie s vizuálnymi prvkami. Celkovo aplikácia obsahuje sedem controllerov. O úvodnú obrazovku sa stará *ViewController*, obrazovku s nastaveniami obsluhuje *SettingsViewController*, vyhľadávanie bluetooth zariadení sa vykresľuje pomocou *ScanTableViewController*. Obrazovka s výpisom produktov je riadená triedou *ProductViewController* a o vytváranie produktov sa stará *ProductAddController*.

Obrazovky obsahujúce tabuľku musia mať ďalší controller na správu tabuľkových buniek. Hovoríme o *ListCellController* a *ProductCellController*. Tieto triedy vykresľujú bunky a spravujú ich dáta.

Obrázok 6: UML diagram controllerov

5 Diskusia

5.1 Testovanie aplikácie

Pred uverejnením aplikácie v obchode AppStore je nutné ju otestovať. V priebehu vývoja sa nám podarilo ošetriť chybové stavy, ktoré boli objavené počas testovania. Hovoríme o týchto chybových stavoch:

- Nevyplnené údaje o pokladni
  Pre úspešné zaevidovanie tržby je potrebné mať vyplnené všetky údaje v nastaveniach. V prípade pokusu o odoslanie tržby s nevyplnenými údajmi je používateľ informovaný o tejto skutočnosti a platba nie je odoslaná.
- Neplatné dáta
  V nastaveniach musí používateľ zadať svoje DIČ, ktoré je potrebné na vytvorenie dátovej správy. Ak používateľ zadá DIČ, pre ktoré nebol vytvorený použitý certifikát, server Finančnej správy vráti chybové hlásenie.
- Neexistujúci certifikát
  V nastaveniach aplikácie je taktiež nutné zadať názov naimportovaného certifikátu. Ak sa takýto súbor v zariadení nenájde, opäť nie je možné odoslať dátovú správu a používateľ je o tom informovaný
- Nedostupné internetové pripojenie
  Zákon o elektronickej evidencii tržieb umožňuje pri určitých výnimkách pracovať aj v režime bez internetového pripojenia. V našej aplikácii však takáto funkcionalita nie je zahrnutá a preto je používateľ pri výpadku internetového pripojenia informovaný o chybe.
- Nedostupná tlačiareň
  Na tlač účteniek je využívaná bluetooth tlačiareň. Tú je nutné pripojiť v nastaveniach. Ak tlačiareň nie je pripojená, používateľ je nútený ju pripojiť, inak nebude tržba zrealizovaná.
- Ostatné chyby
  Ani otestovaná aplikácia nemusí byť bezchybná. Pri dlhšom používaní môžu byť objavené aj doposiaľ nezachytené chyby počas testovania.

Testovanie bolo vykonávané na zariadeniach iPhone 5s a iPhone 6 s operačným systémom 10.3 a uhlopriečkami obrazoviek 4 a 4,7“. Tieto zariadenia môžu využívať aplikáciu naplno. Aplikácia nepadala a pracovala rýchlo. Na simulátore sme si vyskúšali ako sa bude vykresľovať dizajn na zariadeniach s najväčšou uhlopriečkou 5,5“, keďže takéto zariadenie sme nemali k dispozícii. Jedna z vecí, ktorú sa nám nepodarilo otestovať, je funkčnosť ostatných bluetooth tlačiarní, keďže sme mali k dispozícii len jednu, čo je veľká škoda.

5.2 Praktické využitie aplikácie

Navrhnutá aplikácia môže nájsť uplatnenie hlavne u poplatníkov, ktorí vyžadujú jednoduchú aplikáciu na elektronickú evidenciu tržieb bez zbytočných funkcií a za málo peňazí. Aplikácia je totiž open-source a tým pádom je dostupná zadarmo. Hlavným a aj jediným využitím aplikácie bude elektronická evidencia tržieb, vyplývajúca ako povinnosť zo zákona. Tento zákon platí aj pre malých podnikateľov, ktorí dostávajú hotovosť len niekoľko krát do týždňa a rozsiahla aplikácia by bola pre nich zbytočná.

Veľmi vhodnou by bola aj pre starších používateľov, ktorí nemajú s používaním podobných aplikácií žiadne skúsenosti. Aplikácia má zároveň aj dostatočnú veľkosť písma a tlačidiel, tým pádom by bolo ovládanie ešte jednoduchšie. Hlavná cieľová skupina zákazníkov zrejme budú používatelia, ktorí sú stále v pohybe a mobilný telefón s malou tlačiarňou by pre nich bolo veľmi vhodné riešenie pre elektronickú evidenciu tržieb.

5.3 Návrhy na vylepšenie

Návrhov na vylepšenie by sme mali viac. Ostáva si však položiť otázku, či by aplikácia nezačala strácať svoju jednoduchosť, ktorá bola jej hlavnou prednosťou oproti ostatným aplikáciám. Medzi zaujímavé možnosti vylepšenia by sme však určite zaradili možnosť pracovať bez internetového pripojenia.

Skvelou funkciou by taktiež bola možnosť posielať účtenky zákazníkovi emailom, keďže môže nastať technická chyba na tlačiarni a tým pádom by poplatník nemohol prijať platbu v hotovosti.

Používatelia by určite uvítali aj lepšiu správu produktov v produktovom liste, keďže nemajú možnosť vyhľadávania podľa názvu produktu, alebo rozdelenie produktov do skupín.

Ako posledný návrh na zlepšenie by používatelia určite privítali aj verziu pre operačný systém Android. Trúfame si však povedať, že najlepšie návrhy na vylepšenie by aj tak napadli samotných používateľov aplikácie v priebehu jej využívania.

6 Záver

Cieľom tejto práce bolo navrhnúť a implementovať mobilnú aplikáciu pre elektronickú evidenciu tržieb pre operačný systém iOS. Tento cieľ sa nám, podarilo splniť. Väčšina aplikácií pre operačný systém iOS vyvinutých pre tento účel obsahuje všakovaké funkcie, ktoré používateľovi neraz sťažujú prácu s nimi, pretože je zložité orientovať sa v tak zložitej aplikácii.

Aplikácia bude využívaná na zariadeniach iPhone rôznych verzií. Nutnosť je však mať v ňom nainštalovaný systém iOS verzie 8.0 a vyššie. Predispozíciou je nutnosť zakúpiť si bluetooth tlačiareň, na ktorej budú tlačené účtenky.

Po vykonaní rešerše a vyskúšaní existujúcich riešení sme si stanovili čiastkový cieľ vyvinúť jednoduchú aplikáciu, aby ju dokázali ovládať aj starší používatelia. To sa nám podarilo splniť, keďže je aplikácia zložená len z celkovo piatich pohľadov, ktoré si používateľ dokáže osvojiť behom pár minút. Po nastavení aplikácie a pridaní produktov do produktového listu bude používateľovi napokon stačiť používať len dva pohľady. Okrem jednoduchosti má aplikácia voči konkurencii navrch aj v čitateľnosti a prehľadnosti textu.

Podľa rešerše sme taktiež postupovali pri výbere jednotlivých frameworkov tretích strán potrebných pre dokončenie jednotlivých cieľov. Jedná sa o *KissXML* (Hanson, 2017) používaný na tvorbu XML dokumentu odosielaného na server Finančnej správy ako aj parsovanie odpovedi zo serveru, v ktorej sa nachádzajú potrebné údaje o tržbe, ktoré je nutné z dokumentu prebrať. Veľmi dôležitou knižnicou bola aj *SwiftyRSA* (Scoop Technologies, 2017), ktorá bola využitá na šifrovanie a podpisovanie niektorých elementov XML dokumentu. Taktiež sme využívali framework *XUCore* (Charlie Monroe Software, 2017b), ktorý obsahuje funkciu na stiahnutie dokumentov, v našom prípade to bola odpoveď zo serveru Finančnej správy. Na pripojenie niektorých knižníc tretích strán sme používali nástroj na správu závislosti CocoaPods (CocoaPods, 2017).

7 Referencie

Apple Inc. *Apple Developer Program* [online]. [cit. 2017-05-13]. Dostupné z: https://developer.apple.com/programs/.

Apple Inc. *iOS Human Interface Guidelines* [online]. [cit. 2017-03-13]. Dostupné z: https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/.

Apple Inc. *Swift* [online]. [cit. 2017-05-11]. Dostupné z: https://developer.apple.com/swift/.

Apple Inc. *Xcode* [online]. [cit. 2017-03-13]. Dostupné z: https://developer.apple.com/xcode/.

ArrowSys s.r.o. *EET Pokladna LILKA* [online]. [cit. 2017-03-11]. Dostupné z: https://itunes.apple.com/cz/app/eet-pokladna-lilka/id1098126251.

Česko *Zákon č. 112/2016 Sb., o evidenci tržeb.* [online]. 2016. Dostupné z: http://www.sbirka.cz/POSL4TYD/NOVE/16-112.htm.

Charlie Monroe Software *SwiftEET* [online]. [cit. 2017-05-13]. Dostupné z: https://github.com/charlieMonroe/SwiftEET.

Charlie Monroe Software *XUCore* [online]. [cit. 2017-05-13]. Dostupné z: https://github.com/charlieMonroe/XUCore.

CocoaPods [online]. [cit. 2017-03-20]. Dostupné z: https://cocoapods.org/.

David, S. *iOS version stats* [online]. [cit. 2017-05-13]. Dostupné z: https://david-smith.org/iosversionstats/.

Hanson, R. *KissXML* [online]. [cit. 2017-04-12]. Dostupné z: https://github.com/robbiehanson/KissXML.

JetBrains s.r.o *AppCode* [online]. [cit. 2017-04-13]. Dostupné z: https://www.jetbrains.com/objc/.

Knott, M. *Beginning Xcode Swift Edition.* Berkeley: Apress., 2014. ISBN 978-1-4842-0539-6.

Krzyżanowski, M. *CryptoSwift* [online]. [cit. 2017-04-12]. Dostupné z: https://github.com/krzyzanowskim/CryptoSwift.

Mark, D., LaMarche, J. *iPhone SKD Průvodce vývojem aplikací pro iPhone a iPod touch.* 1. vyd. Brno: Computer Press, a.s., 2010. ISBN 978-80-251-2820-6.

Mathias, M., Gallagher, J. *Swift programming: the Big Nerd Ranch guide.* Atlanta, GA: Big Nerd Ranch, 2015. ISBN 978-0134398013.

Mobile APP s.r.o. *MAXiPokladna* [online]. [cit. 2017-03-13]. Dostupné z: https://www.maxipokladna.cz/.

O2 Czech Republic a.s. *eKasa* [online]. [cit. 2017-03-19]. Dostupné z: https://www.o2.cz/podnikatel/ekasa-aplikace/.

Scoop Technologies, Inc. *SwiftyRSA* [online]. [cit. 2017-03-10]. Dostupné z: https://github.com/TakeScoop/SwiftyRSA.

Tadić, M. *AEXML* [online]. [cit. 2017-03-15]. Dostupné z: https://github.com/robbiehanson/KissXML.

Vodafone Czech Republic a.s. *ePokladna* [online]. [cit. 2017-03-19]. Dostupné z: https://www.vodafone.cz/podnikatele/specialni-sluzby/epokladna/.

Wikipedia *iOS* [online]. [cit. 2017-05-13]. Dostupné z: https://sk.wikipedia.org/wiki/IOS.

Prílohy

A CD s aplikáciou

Na priloženom CD je k dispozícii projekt vytvorený v Xcode a testovací certifikát vydaný Finančnou správou.