Sustav mobilnog plaćanja za iOS utemljen na Apple Pay usluzi

Mihalj, Ivan

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij
Procesno računarstvo

SUSTAV MOBILNOG PLAĆANJA ZA iOS UTEMELJEN NA APPLE PAY USLUZI

Diplomski rad

Ivan Mihalj

Osijek, 2017.

Sadržaj

1 UVOD

Današnje prometnice prepune su prijevoznih sredstava koje iza sebe ostavljaju kilometre prijeđenih cesta s neiskorištenim slobodnim transportnim mjestom. Razvoj tehnologije pokušava osvijestiti društvo te omogućiti da se na vrlo jednostavan način prijeđeni put iskoristi za transport i dodatnu zaradu. Kako bi se to i ostvarilo, potrebno je kreirati stabilno okruženje koje će riješiti problem digitalizacije transporta i dijeljenja slobodnog transportnog mjesta kojeg je svakodnevno sve više na prometnicama diljem cijeloga svijeta [1] te omogućiti jednostavniji i sigurniji oblik plaćanja za korištenje prijevoznih usluga.

Diplomskim radom rješava se problem praćenja i plaćanja provedenih usluga putem novog *Apple Pay* sustava implementiranog u programsko rješenje mobilne iOS aplikacije. Teorijski i tehnički dijelovi rada opisuju sigurnost sustava i platforme, komunikaciju sa uslugama, pohranjivanje podataka u sigurnosni element, implementaciju, i istraživanje prihvaćanja u društvu putem ankete. Kako bi sigurnost bila još veća unutar aplikacije, implementirano je sigurnosno prijavljivanje, kreiranje i upravljanje korisničkim podacima preko Googleove *Firebase*[1] usluge te unutar aplikacije plaćanje putem biometrijskog čitača otiska prsta[2] (engl. *TouchID*). Praćenje izvršenih transakcija putem *Wallet*[3] *iOS* aplikacije, *Sandbox*[4] testova i analitike sustava *Stripe*[5]. Budući da je aplikacija razvijena za dijeljenje slobodnog transportnog mjesta, unutar aplikacije implementiran je i sustav Google Mape[6] i Google Mjesta[7] koji su korišteni za praćenje korisnika i upravljanje gradovima i adresama na koje pošiljatelj ili vozač želi ići. Zamišljeno je da platforma bude korištena od svih uzrasta te je najveći naglasak stavljen na novu vrstu poslovanja, ekonomiju dijeljenja koja je kroz svoje principe omogućila smislenu implementaciju i korištenje beskontaktnog plaćanja unutar aplikacije.

U uvodnom dijelu rada opisuje se *Apple Pay* sustav te predstavljaju tehnologije pomoću kojih će se doći do implementacije programskog rješenja. Drugo poglavlje sadrži teorijske podloge za analizu sigurnosti *iOS* platforme kao podloge za obavljanje transakcija te sigurnost *Apple Pay* sustava i prijetnji koje mogu ometati ispravan rad. Nastavak rada prikazuje analizu prikupljenih podataka putem ankete provedene u svrhu shvaćanja prihvaćenosti beskontaktnog plaćanja u Republici Hrvatskoj i mogućeg utjecaja na promjene mišljenja korisnika. U trećem poglavlju opisane su komponente razvojne okoline i objašnjen način implementacije programskog rješenja. Četvrto poglavlje predstavlja arhitekturu aplikacije, integraciju usluga te prikazuje i objašnjava korisničko sučelje. Peto poglavlje detaljnije opisuje način razvoja *Apple Pay* okruženja i njegovu implementacija te je analiziran i objašnjen način testiranja istog i donesen zaključak rada.

2 PROBLEMATIKA I IDEJA

Ideja diplomskog rada ispunjava sve kriterije i promjene koje su potrebne u društvu i omogućuje stvaranje novog poslovnog sustava koji će biti iskoristiv i pružiti jačanje cijelog procesa poslovanja te biti koristan i za krajnje korisnike takozvane treće članice[8] (engl. *third-party*), a ne samo za velike tvrtke i njihove poslovne monopole, što znači da neće samo tvrtke profitirati, nego će svaka osoba zasebno u svoje slobodno vrijeme moći postati dio nove ekonomske politike. Od pružatelja do korisnika usluge, [2] (engl. *third-sector*) jasno govori i pokazuje kolike su mogućnosti njegova izvršenja, pogotovo danas kada se na pristupačan način može doći do mehanizama koji će otkriti želje, zahtjeve i potrebe ljudi te ih kroz relativno brzo vremensko razdoblje pretvoriti u uslugu implementirajući ih u on-line i mobilne platforme te biti dio svjetske promjene i dati doprinos digitalizaciji društva.

Projekt se sastoji od platforme koja korisnicima omogućuje dijeljenje vožnji, slobodnog mjesta, i za ljude i za pakete, štedi vrijeme, novac i to sve sa *zelenim* pogledom na svijet [3]. Ljudi koji svakodnevno putuju u jednom smjeru moći će jeftinije prijeći svoj put kroz dijeljenje vožnje s drugima. Slobodna mjesta u prtljažniku popunit će pošiljkom na kojoj će zaraditi, dok će pošiljatelji u isto vrijeme moći pratiti svoje pošiljke i imati bržu dostavu te na osobnoj razini imati povratnu informaciju o svome dostavljaču i implementiranu najsigurniju vrstu plaćanja putem platforme *Apple Pay*.

2.1 Apple Pay

Appleov revolucionarni *NFC*[9] sustav za beskontaktno plaćanje *Apple Pay* razvijen je za *iOS* i *Mac*[10] uređaje, u svrhu provođenja jednostavnijih transakcija putem Appleovih uređaja. Sustav omogućuje sigurnije transakcije unutar *iOS* aplikacija, *watchOS* aplikacija ili putem *Web* sučelja [4].

Korisnici na zabavniji, jednostavniji i sigurniji način mogu obavljati kupovinu, bez gubljenja vremena na unos kartičnih podataka, adrese dostave i osobnih podataka, pri čemu njihovi podaci nisu javno dostupni. Svi osobni podaci pohranjeni su unutar sigurnosnog elementa (engl. *secure Element*)[11] na *iOS* uređaju, koji osim sigurnosti transakcije omogućuje i korištenje sustava u okolini bez pristupa internetu. To znači da u slučaju da je pristup internetu onemogućen ili se uređaj nalazi u zrakoplovnom načinu rada, *iPhone* može izvršiti plaćanje. To je moguće, jer ne mora trenutno komunicirati s udaljenim uslugama budući da se sve nalazi unutar sigurnosnog elementa.

Sigurnost platforme sastoji se od dvije razine sklopovlja (engl. *hardware*) - čitača otiska prsta koji je potreban prilikom provjere (engl. *evaluation*) svake transakcije te sigurnosnog elementa koji čuva financijske i bankovne informacije zaključanima.

Prednosti korištenja ove tehnologije su:

- sigurnost,
- jedno dodirno plaćanje (engl. *one touch payment*),
- jednostavnost i brzina korištenja,
- usklađenost s aplikacijama,
- pohrana podataka na jednom mjestu

Najveći nedostatak platforme je slaba prihvaćenost, kako u trgovačkim lancima, tako i zemljama koji su postali partneri (njih svega 14).

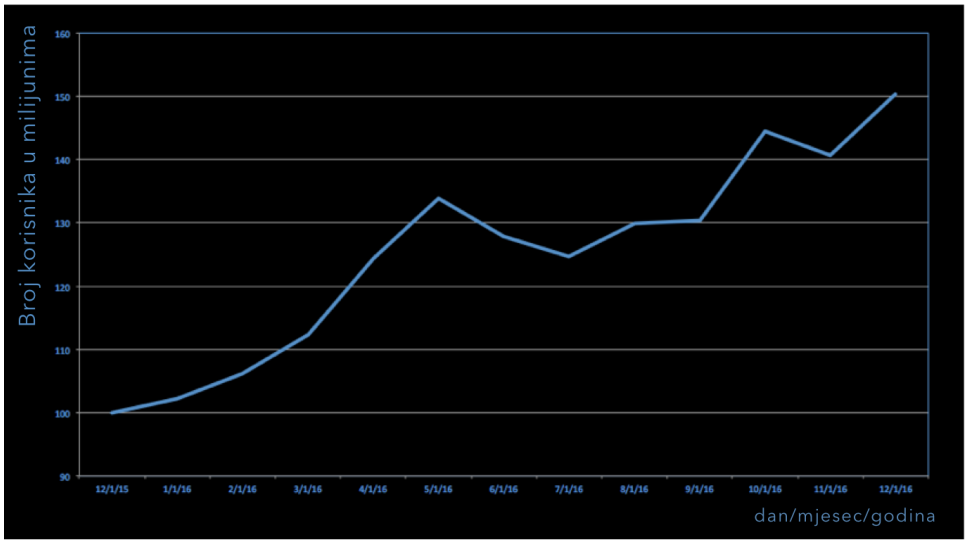
Ukratko, *Apple Pay* predstavlja mobilni način plaćanja koji može biti definiran kao način plaćanja gdje su pametni uređaji korišteni za pokretanje, ovjeru i potvrdu izmjene novčanih vrijednosti u svrhu kupovanja ili plaćanja za usluge ili dobra [5, 6].

2.1.1 Transakcije

Izvršavanje transakcija putem platforme *Apple Pay* za davatelje kartičnih usluga nije besplatno, ovisno na kojem dijelu svijeta se nalaze i ovisno o pravnim regulativama Apple naplaćuje različit postotak za provođenje svake transakcije. Tako je u Americi, prema [19] naknada za provođenje 0,15%, a u državama unutar Europe gdje je platforma odobrena se kreće između 0,2% i 0,3%, ovisno radi li se o kreditnoj ili debitnoj kartici ([7] [8]). Za iste usluge provođenja transakcija i korištenja gotovog sustava, pružatelji kartičnih usluga potražuju između 1% i 3% [9].

Apple je putem svoje platforme u prvoj godini poslovanja, 2015. proveo transakcije u vrijednosti od 10.9 milijardi američkih dolara, što bi s informacijom iz prošlog odlomka uz naknadu 0,15% po izvršenoj transakciji značilo da je uprihodio oko 16 milijuna [10] američkih dolara.

Broj mjesta na kojima je dostupno plaćanje putem *Apple Pay* usluge se podigao s 4% na 35% od ukupnog broja trgovaca u SADu [11]. Prema slici 2.1, jasno je vidljivo kako se broj transakcija obavljenih u razdoblju od prosinca 2015. do prosinca 2016. godine povećao za 50% [12].



Sl. 2.1 *Prikaz rasta korištenja Apple Pay sustava u razdoblju između prosinca 2015 i prosinca 2016. Godine.*

2.1.2 Dostupnost

Stalnim radom na razvoju proizvoda i uređaja koji će podržavati novi sustav (Tab. 2.1) te na širenju tržišta, *Apple* je omogućio svoj sustav korisnicima samo 14 zemalja [13] što je vidljivo na slici 2.2. Kao relativno mlad sustav očigledno je za očekivati da se taj podatak može samo povećavati, a s njim poput domino efekta broj korisnika, transakcija i u konačnici veća zarada kroz sustav.

Tab. 2.1 *Uređaji kompatibilni s Apple Pay sustavom.*

iPhone	iPad	Apple Watch	Mac
iPhone 7	iPad Pro (12.9 - inča)	Apple Watch Series 2	MacBook Pro s TouchID
iPhone 7 Plus	iPad Pro (9.7 - inča)	Apple Watch Series 1	Modeli 2012 i kasnije kompaktni s iPhone ili Apple Watch Touch IDom
iPhone 6s	iPad Air 2	Apple Watch (1. gen)	
iPhone 6s Plus	iPad mini 4		
iPhone 6	iPad mini 3		
iPhone 6 Plus			
iPhone SE			
iPhone 5, iPhone 5s, iPhone 5C *			

*samo u kombinaciji s Apple Watchom

Sl. 2.2 *Države partneri Apple Pay sustava.*

Izlaskom Apple Paya na tržište milijuni ljudi prihvatili su brži i jednostavniji pristup obavljanju transakcija. Plaćanjem u trgovačkim lancima, unutar aplikacija ili kroz web sučelja nije nikad jednostavnije i sigurnije.

Kao što je prikazano na slici 2.3, postoje tri okruženja unutar kojih je podržan *Apple Pay* sustav beskontaktnog plaćanja. Za svaki pojedini oblik potrebno je imati jedan od Appleovih uređaja s podrškom beskontaktnog plaćanja.

Sl. 2.3 *Okruženja unutar kojih je podržan Apple Pay sustav.*

Svaki od četiri navedena uređaja [14] u tablici 2.1 koja podržavaju sustav beskontaktnog plaćanja imaju pravila putem kojih odobravaju transakcije tako da *iPhone* korisnik prilikom plaćanja mora imati prislonjen prst na senzor otiska, sve dok uređaj ne označi provedbu transakcije kao uspješnu. Prethodno je potrebno postaviti vlastite otiske prsta u postavkama [15] *iOS*a koji se poslije mogu koristiti za otključavanje, kupovinu unutar Apple Trgovine mobilnim aplikacijama[12] (engl. *App Store*), plaćati *Apple Payem* ili koristiti unutar drugih aplikacija koje podržavaju čitač otiska prsta.

Plaćanje *Apple Watchem* je zapravo posredničko plaćanje koje ne može funkcionirati ako nije upareno s *iPhone* uređajem. Tek kada je *Apple Watch* uparen, korisnik treba dvostruko pritisnuti rubni upravljački gumb (engl. Side Button) na svom pametnom satu te približiti sat

beskontaktnom čitaču [16]. Plaćanje putem *MacBook* računala može biti samostalno, ako je nova inačica računala koja je izašla krajem 2016. godine ili putem povezanosti računala i *iPhone* uređaja koji podržava *Apple Pay* tako da se narudžba i transakcija izvrši preko računala, a provjera ispravnosti i plaćanje putem otiska prsta preko *iPhone* uređaja. Iznos transakcije koji je odobren bez korištenja dodatnog *PIN*a ovisi o državi u kojoj se koristi sustav, tako je najveća moguća transakcija odobrena putem *Apple Pay*a u Australiji, što iznosi 100 australskih dolara (76 američkih dolara), a najmanja 1000 rubalja (oko 16 američkih dolara) u Rusiji [17].

Plaćanje unutar aplikacija odvija se na isti način kao i plaćanje u trgovinama, samo je prvi puta potrebno ispuniti adresu dostave, ako je proizvod potrebno dostaviti, koja će poslije biti automatski ispunjena ako se korisnik slaže da je adresa za dostavu dobra.

2.2 Korištene programske tehnologije

Ovo potpoglavlje daje u uvid tehnologije korištene u diplomskom radu.

2.2.1 Životni vijek iOS platforme

Appleov operacijski sustav za mobilne uređaje, *iOS* platforma u upotrebi je već 10 godina [18] i iz godine u godinu je nadograđivana i predstavljana putem nove inačice programske podrške. Tako je do ove, 2017. godine plasirano 10 većih inačica te najavljena i nova za rujan iste godine.

Svaka inačica predstavila je barem jednu mogućnost koja je znatno utjecala na napredak platforme, no za ovaj diplomski rad važno je spomenuti iOS 7 i rujan 2013. [19] godine kada je predstavljen novi iPhone 5s i s njime novi način otključavanja iOS uređaja putem biometrijskog otiska prsta, što je preteča ideje za obavljanje transakcija putem mobilnih uređaja.

Sljedeća inačica programske podrške koja je bitna za diplomski rad je iOS 8. Godinu dana nakon plasiranja *TouchID*a na tržište, *Apple* je plasirao inačicu programske podrške s omogućenim beskontaktnim plaćanjem putem *Apple Pay* platforme. Platforma je prvotno bila dostupna samo za *iPhone 6* i *iPhone 6 Plus* uređaje. [20]

S *iOS 9 Apple* je promijenio ime aplikacije *Passbook* u *Wallet* [21], čime se dodatno naglasila važnost digitalne ere pametnih novčanika, u kojoj će korisnici moći spremati kartične podatke u digitalnom obliku, pratiti svoje transakcije, pohranjivati kartice vjernosti, skupljati bodove, ugrađivati kupone za vožnje, odlazak u kino, avionske i autobusne karte te im pristupati direktno putem zaključanog zaslona uređaja ili putem *Wallet* aplikacije.

Operacijski sustav *iOS 10* omogućio je korištenje *Apple Pay* usluge putem *Safari* preglednika Internet veza [22]. Uz *iOS 10*, paralelno je plasirana nova inačica operacijskog sustava *OS X*, unutar koje se po prvi puta omogućilo beskontaktno, odnosno plaćanje otiskom prsta na novim

MacBook Pro [23] uređajima ili u slučaju starih *Mac* računala plaćanje preko uparenih *iPhone* ili *Watch* uređaja.

Nova inačica programske podrške, *iOS 11*, najavljena za jesen ove godine [24] dodatno proširuje *Apple Pay* platformu i omogućuje plaćanje putem usluge *iMessage*[13].

2.2.2 Razvojna okolina

Razvoj aplikacija zahtijeva korištenje *Apple*ovog sklopovlja i programske podrške. Pri razvoju aplikacija na *iOS* platformi za *iOS* uređaj, potrebno je koristiti sljedeće tehnologije:

- *Mac* računalo koje pokreće *Mac OS X*[14] operacijski sustav. Od ove godine razvoj je moguć i na *Linux* platformi
- *Xcode*[15], razvojno sučelje (engl. *IDE - Integrated development environment*) za iOS i ostale aplikacija za Apple platforme
- *iOS* razvojni programski paket (engl. SDK – *Software Development Kit*) koji dolazi ugrađen u Xcode
- *iOS Simulator*[16] ili *iPhone* uređaj
- kada se aplikaciju planira objaviti u *Apple App* trgovini, potrebno je kupiti licencu [25] za uključenje u *Apple* razvojni program [17] (engl. *Apple Developer program*)
- Bez obzira na to što je moguće koristiti *iOS Simulator* za pokretanje aplikacija, prije objave u trgovinu je potrebno aplikaciju testirati i na fizičkom *iPhone* uređaju
- Licencirano razvojno okruženje za dizajniranje korisničkog sučelja

Kod izrade ovog diplomskog rada korištene su i sljedeće tehnologije: *NFC* tehnologija, *Stripe*, *Cocoa Pods*[18], *POS* uređaj[19], *TouchID API*, *Apple Pay API*, interni iPhone senzori, *Sketch*[20] programsko okruženje za razvoj dizajna aplikacije, *Cocoa* i *Cocoa Touch*, *MVC*[21] paradigma razvoja aplikacija (engl. *Model-View-Controller*), baza podataka, *Github*[22], *SourceTree*[23], Web usluga, Internet aplikacijsko programsko sučelje (engl. *Web API)*, *Jenkins*, *Sentry*, *Java SDK*, *GitLab*[24], *Google Firebase*, *Microsoft Office*.

2.2.3 Programski jezik

Swift[25] je novi programski jezik za razvoj *iOS*, *OS X*, *watchOS* i *tvOS* aplikacija koje su izgrađene na najboljim principima *C* i *Objective-C* jezika bez ograničenja na *C* kompatibilnost. *Swift* omogućuje sigurnosne programske paradigme i dodaje moderna svojstva kako bi se programi razvijali jednostavnije i fleksibilnije. U odnosu na prethodni jezik korišten za razvoj *Apple* proizvoda, *Objective-C*, *Swift* pruža bolju sigurnost unosa podataka, sigurnost i performanse [26].

Swift je bio 4 godine u izradi [27]. *Apple* je postavio temelje za *Swift* unapređenjem postojećih prevoditelja, programa za pronalaženje grešaka i razvojnih okvira (engl. *Framework*). Omogućili

su automatsko upravljanje memorijom (engl. *Automatic Reference Counting*) i tako daleko olakšali posao programera. Stog razvojni okvir, napravljen na bazi *Foundation*a i *Cocoa*e razvojnih okvira, je moderniziran i standardiziran. Programski jezik *Objective-C* je unaprijeđen korištenjem blokova, znakovnog područja, modula, omogućujući usvajanje modernih tehnologija bez ikakvih prekida. Sve je to doprinijelo i pomoglo stvaranju jezika budućnosti, na kojem će se temeljiti noviji razvoj Appleovih uređaja.

Swift je vrlo blizak s *Objective-C* programerima, on omogućuje neometan pristup postojećim *Cocoa* razvojnim okvirima te omogućuje neometano pisanje aplikacija povezivanjem oba jezika u jedan. Koristeći zajedničku osnovu, *Swift* uvodi mnoge nove značajke i ujedinjuje proceduralne i objektno orijentirane dijelove jezika.

Swift je potpuno okrenut programerima. Raznolikost platformi na kojima se može razvijati pomoću *Swift*a i vrlo lako učenje tog programskog jezika, razlog je zbog kojeg mnogi studenti i razvojni inženjeri[26] za Apple platformu odlučuju svoj fokus staviti upravo na poboljšavanje svojih vještina u *Swift* programskom jeziku. Dodatnu jačinu pridobivanja interesa razvojnih inženjera Apple je pokušao dobiti otvaranjem i licenciranjem izvornog kôda *Swift* programskog jezika licencom *Open Source* [28], omogućujući tako razvijanje Apple proizvoda ne samo na okruženju koje su sami razvili, nego i na drugim platformama, što je svakako jedan veliki korak za budućnost.

Aplikacije razvijene pomoću *Swifta* mogu se pokretati na *iOS* uređajima počevši od *iOSa 7* pa sve do najnovijeg, a također, na *OS X* uređajima koji imaju inačicu inačicu od *OS X 10.9* na dalje te na novim platformama kao što su *watchOS* i *tvOS*.

Zadnju inačicu *Swift*a moguće je dobiti preuzimanjem najnovije verzije *Xcode*a s *Mac* trgovine aplikacijama ili s *Apple* razvojnog centra (engl. *Apple Developer Centra*). Jednom kada je *Xcode* instaliran, *LLVM* (engl. *low level virtual machine*) prevoditelj za *Swift* i *Objective-C* je automatski instaliran na *Mac* uređaj.

2.3 Apple Wallet

Apple *Wallet* je pred-instalirana aplikacija na *iOS* uređajima koja omogućuje korisnicima digitalni prikaz kartičnih podataka i upravljanje te organizaciju usluga kao što su avionske karte, ulaznice na događanja, poklon kartice, kartice vrijednosti i sve ostale usluge koje razvojni inženjeri ugrade u aplikacije treće strane. Korištenjem *PassKit*[27] okruženja omogućeno je udaljeno dodavanje usluga u *Apple Wallet* te njihovo upravljanje i korištenje putem zaključanog zaslona u vrijeme kada je potrebno upotrijebiti *Pass* uslugu. Osim ručnog upravljanja digitalnim

novčanikom, omogućene su i udaljene akcije te upravljanje i aktualiziranje kartičnih podataka i podataka o uslugama koje su preko kupona prikazane u novčaniku.

Diplomski rad usko koristi aplikaciju *Wallet* preko koje koristi njene obje mogućnosti, mogućnost dodavanja i upravljanja karticama i kartičnog beskontaktnog plaćanja (što radi samo ako se korisnik nalazi u regiji koja ima unaprijed dogovorenu suradnju s Appleom [13]) te mogućnost spremanja kupona unutar aplikacije (odobreno u svim regijama).

Kako bi korisnik *iOS* uređaja neometano koristio usluge plaćanja digitalnim novčanikom, primoran je unijeti svoje kartične podatke unutar *Wallet* aplikacije. Nakon unosa podataka te potvrde valjanosti od strane davatelja kartičnih usluga, digitalni novčanik je spreman za plaćanje Apple Pay uslugom i omogućen mu je pregled izvršenih platnih usluga unutar iste forme. Svaka vožnja ili pošiljka koja se šalje ili prevozi putem *iOS* aplikacije ima implementirane dvije opcije, opciju plaćanja *Apple Pay* uslugom i opciju dodavanja jedinstvenog kôda vožnje u obliku kupona u aplikaciju digitalnog novčanika. Plaćanje unutar aplikacije dodatno ne otvara digitalni novčanik, ali je povezan s njim i u njega sprema podatke o zadnjih 10 transakcija [29]. Spremanje kupona unutar *Wallet* aplikacije odvija se generiranjem jedinstvenog *.pass* kôda koji se jednokratno dodaje u Wallet aplikaciju i može mijenjati i aktualizirati samo preko poslanih obavijesti (engl. *notification*).

2.4 iOS sigurnost

Od početka razvoja *iOS* programske podrške *Apple* je veliku važnost pridavao sigurnosti platforme koju su razvijali. Cilj tvrtke bio je stvoriti najbolju mobilnu platformu, što nije bilo moguće ako se ne posveti dizajniranju arhitekture od jezgre i ako se ne stavi naglasak na osiguranje najsigurnijeg sustava za korisnike. Tako je svaki *iOS* uređaj spoj programskog sklopovlja, programskog okruženja i usluga, dizajniran tako da na najvišoj razini omogućuje sigurnost i transparentnost korištenja uređaja, podataka, mreže, internet usluga i cijelog ekosustava koji se koristi

Kao najsigurnija platforma, arhitekturno je dobro prilagođena korisnicima, gdje korisnici greškom ne mogu nikako promijeniti sigurnosne postavke koje čine programsko okruženje sigurnijim od ostalih okruženja konkurenata i tako narušiti konfiguraciju šifriranih podataka.

Kao što je navedeno, sigurnost *iOS* platforme ne proteže se samo na sigurnost uređaja nego i svih njegovih popratnih segmenata, zaštiti cijelog programskog okruženja, šifriranju podataka, sigurnosti aplikacija, mreže, sigurnosti mobilnih transakcija, internet usluga privatnosti i kontrole nad upravljanjem uređajem u slučaju izvanredne situacije. Svi ti segmenti, bit će objašnjeni detaljnije u poglavljima koja slijede.

2.4.1 Sigurnost programskog okruženja

Sigurnost sustava projektirana je tako da programska podrška i programsko sklopovlje moraju biti sigurni kroz sve temeljne komponente svakog iOS uređaja, što znači da svaka komponenta sustava mora biti provjerena i prihvaćena od sustava. Od početnog pokretanja sustava, svaki korak komunikacije programske podrške i sklopovlja mora biti sinkroniziran u skladu korištenja zajedničkih resursa.

Tako od samog pokretanja iOS sustava, procesi koji se odvijaju u pozadini ne mogu proći bez certifikata, integriteta jezgre i kriptografskih potvrda. Jednom kada su svi zahtjevi uspješno prošli, uređaj je uključen i izvršava programski kod iz memorije spremne samo za čitanje (engl. *Boot ROM*) unutar koje je pohranjen *Apple Root CA* javni ključ i koji potvrđuje ispravnost pokretača operacijskog sustava (engl. *LLB - Low Level Bootloader*) i omogućuje daljnje učitavanje sustava, čime završava prvi ciklus sigurnosne provjere programskog okruženja i započinje sljedeći, *iBoot*, koji svojom validacijom pokreće jezgru *iOS* operacijskog sustava, ali samo u slučaju kada je uređaj na kojem se pokreće sustav izvorni, certificirani *Apple*ov uređaj.

Za uređaje s procesorom počevši od inačice A7 pa nadalje, sigurnosna enklava[28] (engl. *secure enclave - SE*) pomoćni procesor (engl. *coprocessor*) omogućuje sigurnosno pokretanje sustava i ovjeru svih certifikata prilikom podizanja. To radi tako da omogućuje i pohranjuje sve vrste šifriranih podataka koji ne smiju biti dostupni nikome pa ni korisniku unutar čipa i upravlja integritetom zaštite podataka do razine ako je jezgra ugrožena. Komunikacija između SE i procesora aplikacije odvija se preko izolirane razmjene poruka i dijeljene memorije spremnika podataka.

Nakon podizanja sustava, prolaska certifikacije i pokretanja uređaja, *Apple* je omogućio i dodatnu zaštitu svojih uređaja putem sigurnosne lozinke ili senzora otiska prsta. Svi procesi certifikacije i provjere koji se obavljaju prije pokretanja programske podrške, provjeravaju samo povezanost programskog sklopovlja i programske podrške, sada je na redu korisnik koji svojom interakcijom može omogućiti ili onemogućiti dodatnu zaštitu svome sustavu.

Sva teorijska podloga navedena u poglavlju o sigurnosti *iOS* platforme vrlo je bitna za razumijevanje koliko procesa mora biti uspješno izvršeno prije nego li se dođe do unošenja kartice u sustav *Apple Wallet*, njegovog implementiranja u aplikaciju, otključavanja uređaja, prijavljivanje korisnika u sustav gdje će platforma biti korištena, sve do dolaska do ponovne biometrijske ovjere korisnika unutar same aplikacije prilikom plaćanja *Apple Pay* uslugom.

2.4.2 Ključni čimbenici sigurnosti platforme

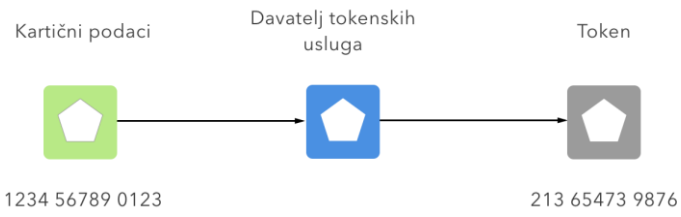
Sigurnost *Apple Pay* platforme može se pokušati usporediti sa sigurnošću transakcija putem plastičnih kartica, premda se može reći kako je *Apple Pay* usluga još sigurnija, budući da ima više razina provjere sigurnosti. *Starpoint* je pisao kako je *Apple* prvi implementirao nove *EMVCo* [29] (engl. *Europay, Visa, Mastercard)* specifikacije tokenizacije što prema njima predstavlja veliki skok iznad *EMV* čipova i unosa *PIN*a i zaključili su kako je *Apple Pay* najsigurnija shema plaćanja na svijetu [30]. Ta sigurnost može se promatrati kroz tri ključna čimbenika [31]:

- Tokenizaciju
- Biometrijska autorizaciju
- Sigurnosni element (engl. *embedded secure element*)

2.4.3 Tokenizacija

Glavnu sigurnost *Apple Pay* usluge predstavlja činjenica da kartični podaci korisnika nisu izravno spremljeni na uređaj niti na jednom od Apple poslužitelja. Ovjera podataka se, prije nego se kartični podaci šifriraju, obavlja putem sheme pružatelja usluga te su nakon toga spremni za šifriranje i zamjenu s jedinstvenim sigurnosnim brojem (engl. *unique Device Account Number*), odnosno tokenom. Token je jedini podatak kojeg Apple sprema na svoj uređaj i to u sigurnosni element kojem nemaju pristup.

Slika 2.4. prikazuje proces stvaranja tokena.



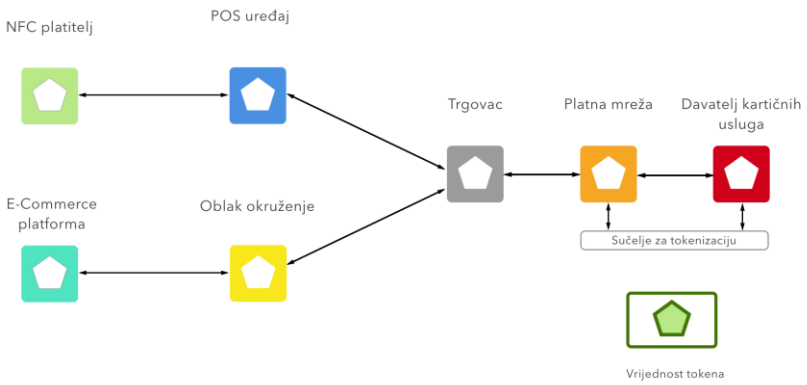
Sl. 2.4 P*roces stvaranja tokena (šifriranje tokena i jedinstvenog broja računa).*

Prilikom izvršavanja transakcije jedinstveni dinamički sigurnosni kôd[30] (engl. *dynamic security code*) poslan je zajedno s brojem uređaja [31] (engl. *device account number*) putem NFC tehnologije. *DSC* je kriptogram koji zamjenjuje statičnu provjeru kartične vrijednosti CVV[32] (engl. *Card verification value*) kôd koji se nalazi na poleđini plastične kartice i jedinstven je za svaku transakciju posebno. To znači kako trgovci prilikom transakcije nemaju nikakav pristup, točnije ne primaju u nikakvom obliku detalje kreditne kartice osobe koja plaća, ali ujedno i onemogućuje dupliciranje tokena i korištenje u zlonamjerne svrhe.

Uz tokenizaciju, *EMV* (opisan u poglavlju 4.1.8) će pridonijeti stvaranju sigurnijeg okoliša prilikom izvršavanja transakcija. Dok je *EMV* dodatna sigurnost na plastičnim karticama, tokenizacija predstavlja sigurnost digitaliziranih transakcija putem mobilnih uređaja ili online putem. Kao što plastična kartica ima samo jedan *EMV* čip, tako i jedan fizički mobilni uređaj ima samo jedan token, ali ako korisnik usluge posjeduje više od jednog uređaja, svaki taj uređaj imat će drugačiji jedinstveni token koji pripada glavnom broju računa (engl. PAN[33] – Primary account number). Za primjer, *MasterCard* omogućuje 9 tokena, dok ih Visa omogućuje 99, što znači da u slučaju *MasterCarda* korisnik može imati 9 jedinstvenih uređaja, a korisnik *Vis*e 99. Osim što je omogućeno imati samo jedan token za jedan fizički uređaj, moguće je imati i poseban token samo za jednog trgovca (engl. *merchant*).

Prednost tokena je što, ako je došlo do ugrožavanja njegove tajnosti, on ne može nikako biti korišten ni na jednom drugom mjestu, nego je izravno vezan uz uređaj na kojem je stvoren. Ako dođe do gubitka iOS uređaja, a budući da se ne koristi originalni broj računa (*PAN*), token uređaja koji je izgubljen na vrlo se jednostavan način može isključiti bez primoranosti pravljenja potpuno nove kartice i korištenja novog *PAN*a.

Način kreiranja tokena u *Apple Pay* okruženju kreće od zahtjeva korisnika kartice, koji putem svog zahtjeva taj isti zahtjev šalje *Apple*u koji ga prosljeđuje određenom davatelju kartičnih usluga, što bi značilo da ukoliko korisnik koristi *Visa* karticu, njegov zahtjev bit će proslijeđen i tokeniziran s *Visa* strane, *MasterCard* od strane *MasterCarda* itd. Generirani token se bazira na *Token BIN* skali ([32], [33]) pojedinog pružatelja kartičnih usluga i onda se šalje ponovno *Apple*u kako bi se mogao pohraniti na njihove uređaje. Pri tome, vidljivo je još jednom kako se broj računa ne sprema nikako na *iOS* uređaje, ali niti u bazu podataka tj. na *Apple* poslužitelje, nego je dostupan samo preko davatelja kartičnih usluga. . Slika 2.5 prikazuje tijek plaćanja s tokenizacijom.



Sl. 2.5 *Tijek plaćanja s tokenizacijom.*

2.4.4 Vrste tokenizacije

Tokenizacija *Card on file* – omogućuje trgovcu promjenu stvarnih kartičnih podataka s platnim tokenima. Transakcijska specifikacija u oblaku računala ili Host Card Emulation (HCE) omogućuje mobilno NFC plaćanje na Android uređajima. Ostali načini su Chase Pay, MCX/CurrentC, te Capital One.

Tokenizaciju je moguće koristiti i na karticama za vjernost, a tu se nudi dosta mogućnosti, kao korištenje tokenizacije kroz kartične usluge, usluge digitalne lisnice ili virtualnih karata i kupona za online kupnju.

Jedini tko ima pristup vrijednosti i povezanosti tokena s pravim *PAN*om je *TSP*[34] (engl. *token service provider*), što znači da sve razine procesa tokenizacije, provedene prije nego li proces dođe do davatelja usluge (TSP) su nedostupne akterima nižeg procesa i teško su dohvatljivi i krivotvoreni. Na taj način *Apple* se ujedno i ograđuje od bilo kake mogućnosti provaljivanja u sustav te je sve odrađeno preko TSPa i Apple ne snosi odgovornost u slučaju napada.

2.4.5 TSP

TSP je glavni dio procesa plaćanja koji stvara platne tokene (surogat podatke) za svaki zahtjev novog tokena. On ima mogućnost upravljanja cijelim platnim ciklusom, njihovim certificiranjem (engl. *credentials*), implementirati tokenizaciju u svrhu smanjivanja prijetnji.

Davatelj tokena omogućuje:

- Tokenizaciju - zamjenu stvarnog primarnog broja kartice sa surogat token brojem
- Detokenizaciju - pretvorbu tokena nazad u primarni kartični broj korištenjem token vrijednosti
- Token vrijednost - uspostavlja i prati platni token i pridružuje ga (engl. *map*) određenom primarnom kartičnom broju
- Upravljanje domenama - omogućuje dodatnu sigurnost tokenizacije, tako da svaki token ima samo određenog trgovca s kojim može surađivati
- Identifikacija i verifikacija - omogućuje da prilikom plaćanja platni token bude proslijeđen do legitimnog, pravog *PAN*a kod davatelja kartičnih usluga
- Proces čišćenja i prekida - omogućuje *Adhoc* detokenizaciju prilikom procesa prekida

2.4.6 Biometrijsko odobravanje korisnika

Bliska povezanost *Apple Pay* usluge s biometrijskom autorizacijom korisnika, dodatno daje na sigurnosti ove platforme [34]. Krađom plastične kartice ili pametnog uređaja na kojem je omogućen način beskontaktnog mobilnog plaćanja neće biti toliko jednostavno pronevjeriti ukradene podatke kao one koji se mogu vidjeti prilikom unosa *PIN*a na jednom od bankomata.

Kada se govori o složenosti krađe podataka, lakše je ukrasti i vidjeti *PIN* kreditne kartice nego to napraviti na biometrijskom senzoru prilikom mobilnog plaćanja**.**

2.4.7 Sigurnosni element

Sigurnosni element je spremnik za sve važne informacije potrebne za izvođenje plaćanja u sigurnosnom okruženju, uzimajući u obzir da vrlo važni podaci kao što su biometrijske šifre otiska prsta ne mogu biti korištene prilikom krađe podataka. Upravo ti biometrijski podaci, spremljeni su sigurno unutar sigurnosnog elementa i uz pomoć načina rada sigurnosnog elementa, tokenizacije kartičnih i transakcijskih podataka te biometrijskom validacijom korisnika, čine Apple Pay jednom od sigurnijih načina plaćanja.

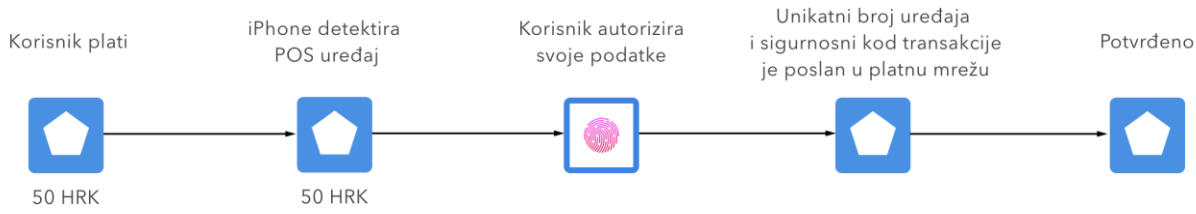
2.4.8 EMV

EMV je standard za kreditne i debitne kartice baziran na kartičnoj čip tehnologiji koji je ime dobio po originalnim tvrtkama koje su ga i razvile (Europay, MasterCard, Visa). *EMV* kartice sadrže ugrađene mikroprocesore koji omogućuju veću rasprostranjenost korištenja te mogu spremati neznatno veću količinu podataka od magnetske trake koja je bila preteča. Kada je kartica korištena na određenom platnom uređaju omogućeno je velikoj količini informacija da budu izmijenjene između kartice, terminala i bankovnog sustava. Mogućnost proširene pohrane omogućila je karticama spremanje podataka kao što su provjera valjanosti podataka (engl. *data authentification*), izvan mrežnu *PIN* validaciju i šifriranje za buduće kartične transakcije te je povećana prevencija napada i krivotvorenja kartica.

2.4.9 Analiza sigurnosti Apple Paya

Plaćanje putem *Apple Paya* izvršava se preko *NFC* tehnologije, koja u slučaju transakcija obavljenih na *iOS* uređajima, ne sprema njihove kartične detalje i ne dijeli ih s prodajnim kućama gdje su transakcije izvršene nego generira nasumični broj uređaja[35] (engl. *Random device number - RDN*) putem kojeg je transakcija izvršena te njena sigurnosti i očuvanje kartičnih podataka korisnika povećana, budući da je generirani broj potpuno drugačiji i nema nikakve veze s brojem kreditne kartice.

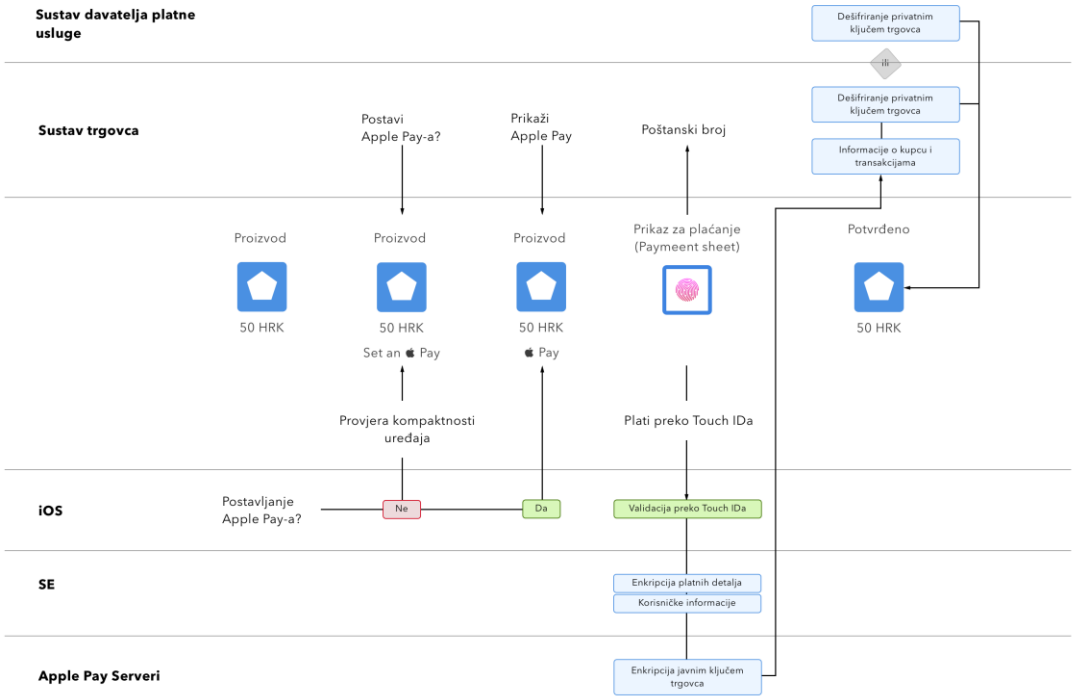
Slika 2.6 prikazuje proces plaćanja na POS uređajima.



Sl. 2.6 *Plaćanje na POS uređajima.*

Budući da se korisničke kartične informacije ne pohranjuju na *Apple* poslužitelje, smanjuje se mogućnost krađe istih podataka putem transakcije s *POS* uređajem. Dok kriminalno ponašanje može biti izvedeno prije, prilikom prvog dodavanja kartičnih podataka unutar *Apple*ove aplikacije za digitalni novčanik. Prema [5], Apple se ograđuje od odgovornosti prilikom unosa kartičnih podataka u njihovu aplikaciju, a banka snosi svu odgovornost prilikom identifikacije (engl. *authentification*) kartice koja je dodana u *Apple*ov sustav.

Slika 2.7 prikazuje proces izvođenja transakcije unutar iOS aplikacije.



Sl. 2.7 *Proces izvođenja transakcije unutar aplikacije.*

2.5 Prijetnje

2.5.1 Lažiranje otiska prsta

Budući da je biometrijska identifikacija nova tehnologija na mobilnim uređajima, hakeri su pronašli način kako probiti u uređaj putem lažiranog otiska. Korisnici svakodnevno dodiruju predmete za koje i ne znaju da su im prijetnja i da netko tko pokušava pronevjeriti njihovu identifikaciju na neki način ipak može doći do njihovog otiska. Tako su uspjeli probiti u *iPhone 5s* putem lažnog otiska prsta [35, 36] koji je kao uzorak pronađen na čaši, što znači u slučaju da korisnik izgubi svoj *iPhone* 5s i ako osoba koja je pronašla zna gdje može doći do otiska prsta osobe koja je izgubila mobitel, postoji način da ta osoba neovlašteno pristupi uređaju. Naravno, senzor otiska prsta na starijim uređajima je dosta sporiji i ima sigurnosna ograničenja operacijskog sustava, što znači da se nakon 5 puta uređaj zaključa i traži unos lozinke.

2.5.2 Korištenje aplikacija trećeg reda

Pristup korisničkim podacima, odnosno *PIN*u *iPhone* uređaja može se ostvariti putem aplikacija za praćenje korisničkog unosa. Osoba koja želi ukrasti lozinku osobe mora imati prvotni pristup uređaju, na njega instalirati zlćudnu aplikaciju i vratiti uređaj nazad korisniku. Pomoću te zlćudne aplikacije i praćenja korisničkog unosa, haker je u stanju pratiti unos i zaslon korisnika *iPhone* uređaja, što znači da je u mogućnosti pronevjeriti njegovu lozinku i unos kartičnih podataka ručnim unosom ili skeniranjem kartice putem kamere.

U slučaju da dođe do krađe *iOS* uređaja ili pronevjere kartičnih podataka putem *iTunes* platforme i *Find My iPhone* okruženja, udaljeno se može blokirati i onemogućiti pristup uređaju. Osim toga, može se obrisati kartica i kartični podaci te onemogućiti korištenje *Apple Pay* sustava. Budući da *Apple Pay* sustav radi i u izvan mrežnom stanju (engl. *offline mode*) korisnicima je omogućeno i izvan mrežno plaćanje jer se komunikacija vrši preko POS uređaja, Appleovih poslužitelja i poslužitelja davatelja kartičnih usluga te nema veze s mrežnim statusom *iOS* uređaja. Ako kradljivac pokuša ukradeni uređaj zaštititi od samozaključavanja putem *Find My iPhone* platforme, on ne može naknadno koristiti kartične podatke i plaćati ukradenim uređajem ako je stvarni vlasnik uređaja udaljeno uklonio karticu.

2.6 Provjera valjanosti kartica pri Apple Pay sustavu

Korisnik *Apple Pay* platforme na *iOS* uređaju može na nekoliko načina dodati i odobriti (engl. *authorise*) karticu na svom uređaju:

- dodavanjem kartice ručno u sustav *Apple Pay*
- dodavanjem putem *iTunes Store* računa
- dodavanje kartice putem bankovne aplikacije

Nakon što je kartica dodana u sustav, Apple prosljeđuje korisničke i kartične podatke do pružatelja kartičnih usluga te im tako omogućuje potpunu slobodu odobravanja, odbijanja i dodavanja kartice budućeg korisnika *Apple Pay* usluge. Da bi podaci uopće došli do pružatelja, *Apple* koristi tri poslužiteljska poziva koji pomoću sigurne šifrirane komunikacije (engl. *SSL – Secure Sockets Layer*) se izmjenjuju između klijenta i poslužitelja:

- provjera obaveznih polja za unos podataka (engl. *required fields*),
- provjera kartice (engl. *check card*) i
- veza i osiguravanje (engl. *link and provision*)

Budući da pokušaja pronevjere podataka ima sve više, detaljni kartični podaci nisu spremljeni na *Apple*ovim poslužiteljima niti njihovim uređajima, nego su samo putem unikatnog broja računa (engl. *unique Device Account Number*), šifrirani i spremljeni u sigurnosni element, tako da im ni

sam *Apple* ne može pristupiti. Zbog njegove unikatnosti i sigurnosti, korištenje *DAN*a se može očekivati i na karticama te na web stranicama, dok je na pametnim uređajima već vrlo rasprostranjen.

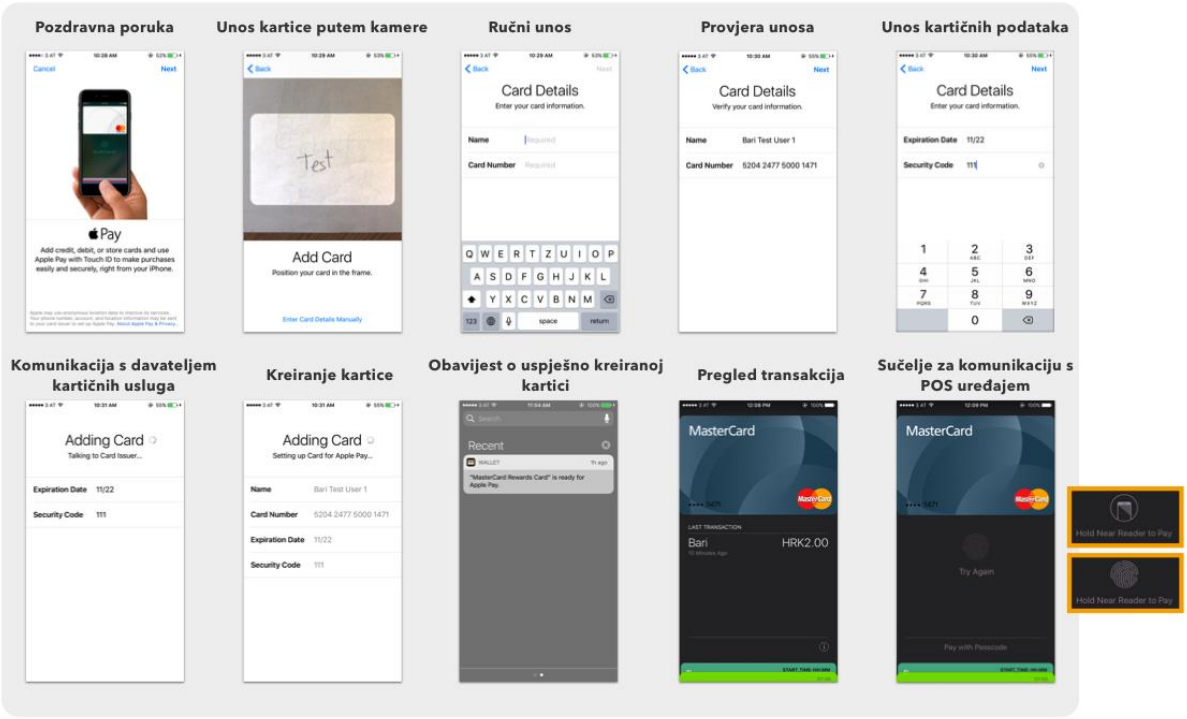
Kartice za korištenje putem *Apple Watch*a ovjeravaju (engl. *verification*) se pomoću *Watch* aplikacije unutar iOS uređaja. Pametni sat mora biti spojen *Bluetooth*om s *iPhone* uređajem i tek tada će se generirati *DAN*, koji je posebno napravljen i odvojen za *Apple Watch* i spremljen unutar sigurnosnog elementa na *Apple Watch* uređaju.

2.6.1 Ručno dodavanje kartice

Proces osiguravanja (engl. *provisioning*) prilikom ručnog dodavanja kartice sastoji se od upisivanja Imena, Broja kreditne kartice, Datuma isteka i CVVa i vidljvio je na slici 2.8.

Navedeni traženi podaci mogu se dodavati unutar postavki *Wallet* ili *Apple Watch* aplikacije i mogu biti upisani ručno ili direktno očitani pomoću *iSight* kamere koja sva polja koja je prepoznala na kartici automatski upiše unutar aplikacije. Skenirana fotografija kartice nikada nije spremljena ni u kakvom digitalnom obliku na *iOS* uređaje. Kada je sve uneseno i prepoznato od strane aplikacije, podaci se šifriraju i šalju na Apple Pay poslužitelj (svi osim CVVa). Ako je na *Apple* poslužitelj vraćen ID s uvjetima i odredbama korištenja (engl. *terms and conditions*), *Apple* prikazuje te uvjete korisniku i tek nakon što je korisnik prihvatio uvjete od davatelja kartičnih usluga, *Apple* šalje *ID i CVV* na proces veze i odobravanja (engl. *Link and Provision*). Tim procesom *Apple* dijeli korisničke podatke (*iTunes* informacije, *App Store* račun, transakcijsku povijest, inačicu uređaja, telefonski broj, ime i programsku inačicu, lokaciju) u svrhu pomoći pri odluci dodavanja korisnika u *Apple Pay* sustav. Kao rezultat *Link* i *Provision* procesa, izvršavaju se dvije stvari, prva, uređaj počinje skidati *.pass* datoteku na uređaj koja predstavlja kreditnu ili debitnu karticu i počinje vezati karticu sa sigurnosnim elementom. *Pass*[36] se sastoji od *URL*a putem kojeg se mogu skinuti kartični ili podaci kontakta. Tako se može saznati je li status personalizacije sigurnosnog elementa završen, koji je status kartice i je li ona suspendirana od strane davatelja kartičnih usluga te je li verifikacija odobrena prije provođenja *Apple Pay* plaćanja.

Slanje dodatnih korisničkih podataka koje posjeduje *Apple* vrlo dobro dođe prilikom provjere korisnika, koja se može izvesti na temelju 4 osnovna statusa imenovanih prema bojama. Zeleni, žuti, narančasti i crveni status provjere. Svaki od statusa provjere ima svoja ograničenja i detalje na temelju kojih davatelji kartičnih usluga mogu uvidjeti stanje korisnika, kada je kreiran Apple ID, kada su zadnji puta promijenjene kartične i adresne postavke, kolika je aktivnost korisnika, je li uređaj bio u izgubljenom stanju i kada, koja je vrijednost uređaja itd.



Sl. 2.8 *Ručno dodavanje kartice.*

2.6.2 Dodavanje putem iTunes računa

Korisnik koji želi dodati karticu putem *iTunes* računa treba samo ponovno upisati svoju *Apple ID* lozinku, nakon koje se obavlja provjera korisnika i omogućuje sinkronizacija korisničkih podataka i pokreće proces provjere kartice (engl. *Check Card*). Ako je kartica valjana i prihvaćena za rad na *Apple Pay* platformi, uređaj prima uvjete i odredbe te šalje *ID* i tajni kôd prema procesu vezivanja i odobravanja.

2.6.3 Dodavanje kartice iz aplikacije davatelja kartičnih usluga

Ovo dodavanje sličo je procesu ručnog dodavanja kartice, samo što se umjesto *CVV*a unosi jednokratna lozinka. Nakon što je aplikacija registrirana za korištenje platforme, generiraju se ključevi za sustav trgovca i za aplikaciju. Ti ključevi, korišteni su za šifriranje kartičnih informacija koji će biti slani na sustav trgovca i tako šifrirani onemogućeni su da budu pročitani od strane *Apple*a.

2.6.4 Dodatne provjere

Davatelj kartičnih usluga može odlučiti želi li dodatne kartične provjere što stvara *Apple Pay* sustav još sigurnijim. U [5] je navedeno da se kartični podaci mogu upisivati nebrojeno puta, bez dodatnog prisilnog prekidanja rada sustava (engl. *brute force*), ova opcija pomaže pružateljima kartičnih usluga dodatnu provjeru prilikom dodavanja kartice. Provjere se mogu obavljati putem

tekstualne poruke, elektroničke pošte, poziva korisničke podrške ili nekom drugom metodom unutar aplikacije trećeg reda, npr. aplikacija banke. Nakon odabira dodatnog načina provjere, kôd se treba unijeti unutar *Wallet* aplikacije u postavkama uređaja. Naravno, dodatna provjera je uvijek dobar korak prema još sigurnijem rješenju sustava, ali ako je sustav hakiran od strane stručnjaka ili mobilni uređaj ukraden, ni dodatna provjera ne može baš puno pomoći jer većinom korisnici primaju istu tu sigurnosnu provjeru na njihove uređaje.

2.6.5 Autorizacija transakcije (plaćanja)

Transakcija unutar *iOS* uređaja potvrđuje se preko sigurnosne enklave (engl. *Sencure Enclave*), koja šalje autorizacijske podatke, unesene od strane korisnika putem biometrijskog senzora otiska prsta ili putem lozinke prema sigurnosnom elementu koji je povezan s NFC upravljačem. Komunikacija te dvije komponente odvija se putem *AES* [37] (engl. *Advanced Encryption Standard*) kako bi se obje strane zaštitile od napada. Iako nisu direktno povezane, veza je vrlo sigurna budući da su povezani dijeljenjim ključem (engl. *shared pairing key*[38]) koji je prvotno generiran unutar enklave (iz njegovog UID [39] i jedinstvenog identifikatora sigurnosnog elementa). Nakon kreiranja, ključ spajanja (engl. *pairing key*) je sigurno prebačen do sigurnosne enklave u sklopovlje sigurnosnog modula (engl. hardware security module HSM[40]) koji ima mogućnost ubaciti ključ spajanja u sigurnosni element.

Nakon korisničkog odobravanja transakcije biometrijskim putem, sigurnosna enklava šalje potpisane i certificirane podatke s detaljima transakcije i otkriva jesu li transakcije napravljene beskontaktnim putem ili unutar aplikacije te ih vežu s nasumičnom vrijednošću odobravanja (engl. *authorisation random*) koja se stvori unutar sigurnosne enklave jednom, na početku, prilikom unosa i odobravanja kreditne kartice i koristi sve dok je *Apple Pay* usluga odobrena i to sve pod posebnim uvjetima šifriranja unutar sigurnosne enklave. Nasumična vrijednost se zatim putem dijeljenog ključa dostavlja do sigurnosnog elementa i bude tamo sve dok se ne promijeni AR vrijednost, nakon čega usluga *Apple Pay* briše sve dotadašnje dodane kartice i označuje ih u sustavu kao obrisane. Briše ih da ne bi došlo do zloupotreebe, a to je kada je AR vrijednost i vrijednost dijeljenog ključa drugačija od vrijednosti postavljenih kada je korisnik dodao kartice u sustav [37]. Takvo ponašanje omogućuje razvojnom okruženju da u sljedećim slučajevima markira kartice kao nevažećima, a ujedno i nauči enklavu kako da postavi kopiju ARa nevažećom:

- Uređaj je vraćen iz sigurnosnog stanja
- Korisnik je obrisao sav sadržaj i postavke
- Lozinka (engl. *passcode*) je isključena

- Korisnik nije prijavljen na *iCloud*
- Lozinka na pametnom satu je onemogućena
- Detekcija zgloba je isključena
- Pametni sat nije uparen s *iPhone* uređajem

Zaključno, prije nego li sigurnosni element odobri aplet [41] (engl. *applet*) za beskontaktno plaćanje mora koristiti dijeljeni ključ i kopiju ARa koju je učitao iz sigurnosne enklave.

2.7 Usporedba s konkurencijom

Glavni konkurenti *Apple Pay* platforme su *Samsung Pay* i *Google Wallet* dok su od ostalih izdvojenih transakcijskih platformi analizirani *Paypal* i *Bitcoin.* Kratka tablica usporedbe *Apple Pay* platforme s konkurencijom nalazi se u tablici 2.2.

Tab. 2.2 *Usporedba s konkurencijom*

Opis	Apple Pay	Samsung Pay	Google Wallet	Paypal	Bitcoin
Dostupnost	Od iPhone 6	Od Samsung Galaxy S6	Bilo koji uređaj s instaliranom aplikacijom	Bilo koji uređaj s aplikacijom	Bilo koji uređaj s aplikacijom
Kako koristiti	Otisak prsta	Otisak prsta	*Tap-to-pay* samo na uređajima s NFC čitačem. Slanje novca putem aplikacije i emaila	Slanje novca putem emaila ili broja mobitela.	Skeniranjem QR kôda
Kako radi	Pomoću NFC radiovalova	NFC ili magnetska polja, na starim POS uređajima	Kao kreditna kartica, registrira se kao korisnik	Preko PayPal mreže	Potpuno neovisan sustav novca
Sigurnost	Najsigurniji. Nema pristupa kartici	Najsigurniji. Nema pristupa kartici	Siguran. Pristup kartici ima samo Google	Siguran. Pristup kartici ima PayPal	Korisnik sam ovisi o sebi
Prednosti	Brzo i jednostavno	Brzo i jednostavno. Radi svugdje	Odlično za slanje novca prijateljima	Odlično za slanje novca prijateljima	Privatnost očuvana. Jednostavno. Za slanje novca prijateljima
Nedostaci	Ne radi svagdje	Opcija magnetskog čitača je loša. Mora se cijelo vrijeme držati iznad čitača	Ne radi svagdje	Samo radi s trgovcima koji su prihvatili PayPal	Jako rijetko prihvaćen

2.8 Statistička obrada podataka

Plasiranje nove tehnologije na tržište zahtijeva prihvaćanje te svijest tog tržišta o novoj tehnologiji koja se pokušava uvesti. Svijest tržišta o novoj tehnologiji stvara najveće probleme

pri prihvaćanju iste, što u konačnici zna dovesti da prilagodba na novu tehnologiju traje znatno duže nego implementacija iste.

Bankovni sustavi novog doba imaju problem s privlačenjem korisnika na nove tehnologije, zbog čega u početku znaju dosta gubiti na vremenu i novcu prilikom pokušavanja dolaska do što većeg broja korisnika. Tako i sigurno u slučaju beskontaktnih transakcija banke i trgovine, a ujedno i tvrtke koje se bave razvojem programske podrške će imati veliki zalogaj i težak posao prilikom pokušaja i pripreme te učenja korisnika o novim beskontaktnim, mobilnim plaćanjima.

Za potrebe rada izrađena je jedinstvena, anonimna anketa gdje su pitanja bila u vezi mobilnih transakcija, njenoj budućnosti, implementaciji i uopće svijesti ljudi o mobilnim transakcijama u Hrvatskoj. Svrha ankete je dobiti uvid u razinu prihvaćanja ovakve nove tehnologije te stav o novim i modernim načinima plaćanja. Anketa je izrađena pomoću *Google Forms*[42] platforme.

Kao priprema za pisanje diplomskog rada i sastavljanja te analiziranja ankete, proučeno je nekoliko dokumenata, među kojima je i jedan od Hrvatske narodne banke, koji daje jasan uvid u platne kartice i kartične transakcije te na vrlo jednostavan i detaljan način prikazuje statistike platnog prometa [38]. Kako bi analiza bila detaljnija, fokus je stavljen na digitalni dio gdje su prikazani podaci za beskontaktne platne kartice, mobilna plaćanja i broj izdanih računa.

Beskontaktna platna kartica je kartica koja omogućuje obavljanje normalnih transakcija (kontaktnim putem) i novi način beskontaktnih transakcija (beskontaktnim kartičnim putem) putem čipa ili magnetske trake. Plaćanje na beskontaktnom EFTPOS[43] uređaju odvijaju se beskontaktno približavanjem ili prislanjanjem beskontatkne kartice uređaju koji u sebi ima ugrađene čitače digitalnog zapisa s kartica. Osim beskontaktnog plaćanja, takva vrsta platne kartice zadržava i sve funkcionalnosti kontaktnih platnih kartica. Tablica 2.3 prikazuje broj izdanih platnih kartica u Republici Hrvatskoj.

Tab. 2.3 *Broj kontaktnih i beskontaktnih platnih kartica izdanih u Republici Hrvatskoj. Prema* [38].

Vrsta platne kartice	Kontaktna	Beskontaktna	Ukupno
Debitna platna kartica	5.306.872	1.572.509	6.879.381
Kreditna platna kartica	1.752.718	106.894	1.859.612
Ukupno	7.059.590	1.679.403	8.738.993

Od ukupnog broja platnih kartica iz analize HNBa na dan 31. prosinca 2015. onih koje imaju mogućnost beskontaktnog plaćanja je 1.679.403 ili 19% dok kontaktnih ima znatno više, 7.059590, što je 81%.

Digitalizacija mobilnih transakcija ne može se odviti bez korištenja mobilnih uređaja. Broj mobilnih telefonskih linija u Hrvatskoj prema [39] prikazan slikom 2.9. je 4.78 milijuna, a

korisnika interneta 3.17 milijuna, što govori da je digitalizacija već započela i da postoje preduvjeti za stvaranja okoliša unutar kojeg bi se moglo koristiti i implementirati mobilna plaćanja. Taj poprilično velik broj mobilnih uređaja prikazuje dosta slabu količinu obavljenih kartičnih platnih transakcija putem njih. Svega negdje oko 1000 kartičnih platnih transakcija, što je nešto oko 0.84 milijuna kuna [38].

Sl. 2.9 *Broj stanovnika, korisnika interneta i pretplatnika[44] mobilnih operatera.*

Količina obavljenih mobilnih transakcija mora porasti s vremenom u Hrvatskoj, a porast mobilnih transakcija dovodi do ubrzanog života i smanjenja troškova izdavanja računa. U Hrvatskoj je u 2015. godini izdano 2.327 milijuna računa s ukupnom vrijednosti od 154.571 milijun kuna. Od ukupne vrijednosti računa, njih 88,5% plaćeno je gotovim novcem.

2.8.1 Anketa - Spremnost društva na beskontaktna plaćanja putem pametnih uređaja

Prilikom kreiranja ankete i anketiranja korisnika korišteno je nekoliko teorija i modela istraživanja preuzetih od Sajida i Haddara [40] od kojih je važno izdvojiti dva modela:

Model prihvaćanja tehnologije[45] (engl. *Tehnology Acceptance Model*) je Davisov model [41] informacijskih sustava koji pokazuje prihvaćanje određene tehnologije, u ovom slučaju *NFC* plaćanja. Budući da se *NFC* mobilno plaćanje još uvijek ne koristi u velikoj mjeri u Republici Hrvatskoj, vrlo je važno provesti istraživanje spremnosti društva na beskontaktna plaćanja putem pametnih uređaja kako bi se dobila povratna informacija od korisnika (Sl. 2.6).

Procjena korisnosti[46] (engl. *perceived usefulness – PU*) – opisuje stupanj u kojem osoba vjeruje da će se njena učinkovitost povećati koristeći određeni sustav.

Procjena jednostavnosti korištenja[47] (engl. *perceived ease-of-use – PEOU*) [42] – stupanj prema kojem osoba vjeruje kako će korištenje određenog sustava biti bez napora, tj. jednostavnije.

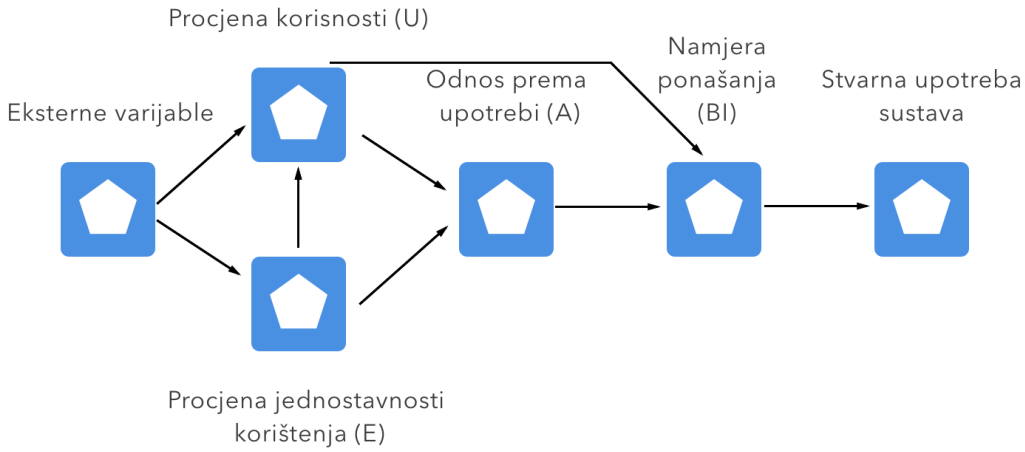
- Drugi oblik istraživanja je model upravljanja promjenama[48], tj. model promjene mišljenja [43] (*engl. change management model*). Model se sastoji od tri razine koje su korištene

kako bi se testirali korisnici i vidjela njihova spremnost na promjenu tradicionalnog načina plaćanja

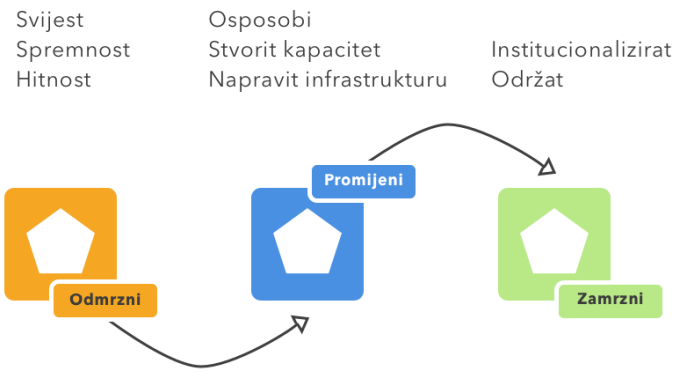
Ovaj model se smatra bitnim unutar domene upravljanja promjenama i ima tri glavne razine koja ga opisuju:

- Odmrzni (engl. *unfreeze*) – Priprema korisnika za promjenu i lagano prisiljavanje na promjene te isto tako uspostavljanje potrebnog okoliša za promjene koje se žele napraviti
- Promjena ili tranzicija (engl. *change*) – Tranzicija korisnika s jednog na drugi oblik ponašanja i korištenja. Ovaj dio uključuje smanjenje potencijalnog otpora prilikom promjene
- Zamrzni (engl. *refreeze*) – uspostaviti stabilnost jednom kada je promjena uvedena, provest promjenu i promicanje prihvaćanja promjene kao nove norme u sustavu

Opisani modeli prikazani su slikom 2.10. i 2.11.



Sl. 2.10 *Model prihvaćanja tehnologije (TAM).*



Sl. 2.11 *Model promjene mišljenja.*

Pretpostavke koje su postavljene prije istraživanja:

H1: Glavne prednosti za usvajanje *NFC* tehnologije bit će jednostavnost korištenja i korisnost sustava

H2: Svijest o *NFC* plaćanju će biti srednje vrijednosti

H3: Korisnici su spremni za promjene i zamjenu kartičnog plaćanja s beskontaktnim mobilnim

H4: Plaćanje kreditnim karticama uzet će veći postotak

H5: Korisnici vjeruju u *NFC* plaćanje

H6: Bojaznost korisnika je da će izgubiti podatke i biti još više praćeni od banaka

H7: Korisnici su spremni promijeniti banku ako u njoj imaju mogućnost mobilnog plaćanja

H8: Stanovnici Republike Hrvatske spremni su na promjenu načina plaćanja i digitalizaciju

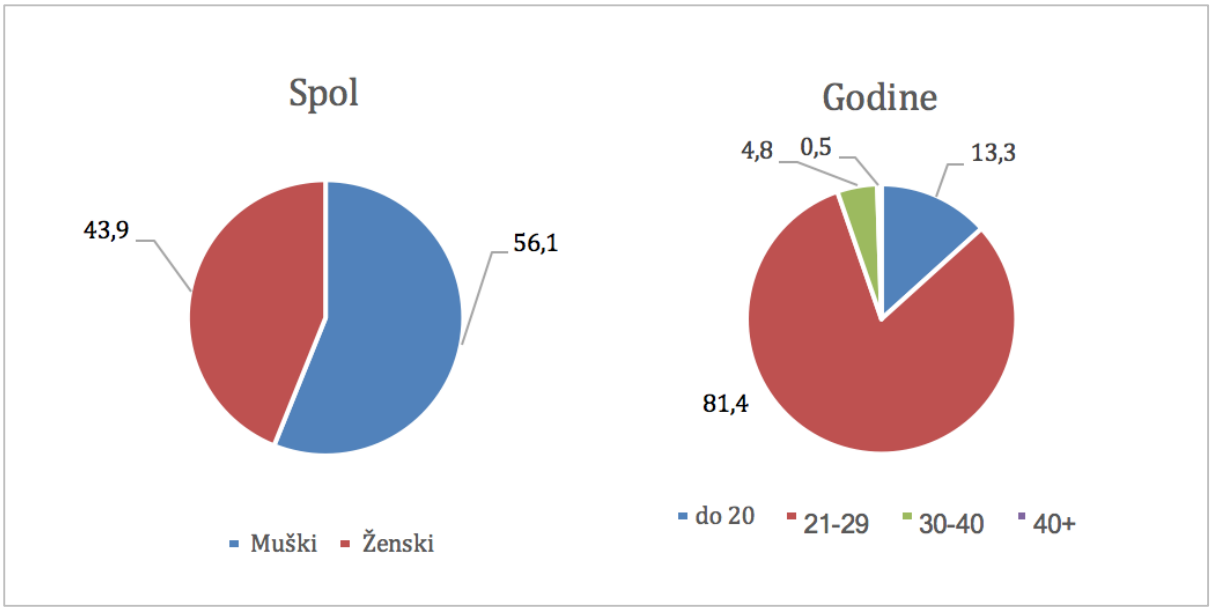
Istraživanje je provedeno pomoću *Google* obrasca za ankete [44] i provoditelj istraživanja nije upotrebljavao niti spremao korisničke podatke anketiranih osoba. Anketa je jedinstvena i samo se jednom mogla ispuniti nakon prijave u *Google* sustav što znači da je svaka osoba, tj. korisnički račun, mogao samo jednom ispuniti anketu.

Postavljeno je 14 pitanja koja su pokrivala više polja i tema, od osobnih podataka kao što su, spol, godine i socioekonomski status te obrazovanje, do svijesti o *NFC* plaćanjima, samoj tehnologiji te budućnosti Hrvatske države. Anketa je objavljena na *Facebook* grupama, poslana poznanicima i prijateljima te je prikupljeno 189 odgovora. Anketa je pisana na hrvatskom jeziku i nije predviđeno da ju popunjavaju ljudi koji ne žive na teritoriju Republike Hrvatske i ne razumiju hrvatski jezik jer se odnosi na digitalizaciju Republike Hrvatske i demografsku analizu.

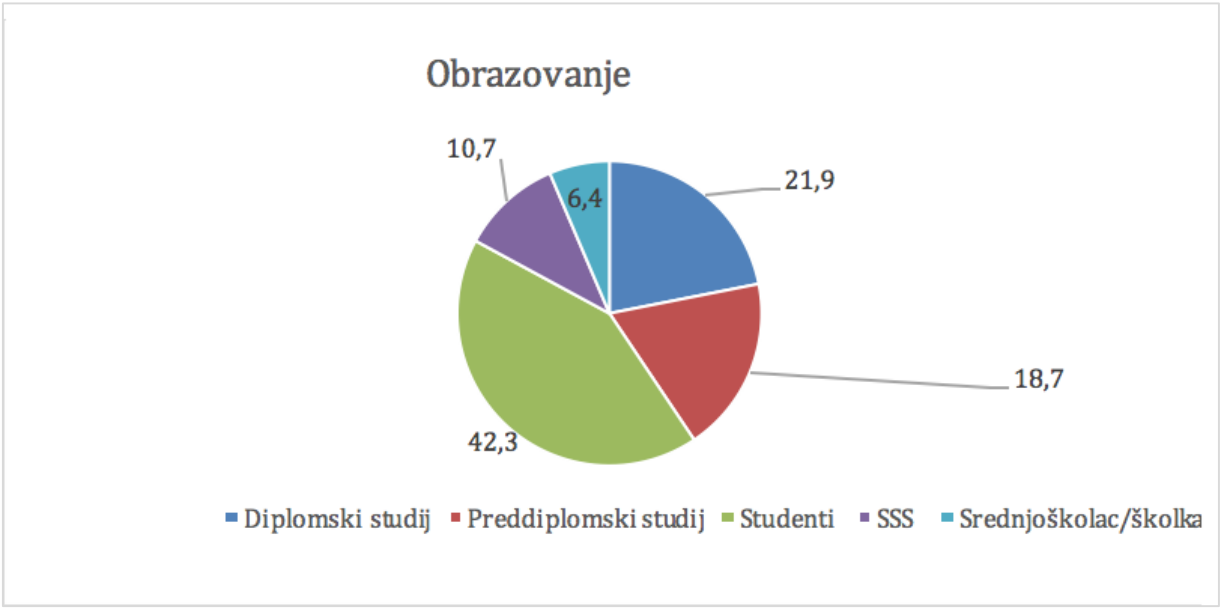
2.8.2 Brojčani prikaz rezultata ankete

P1 - Opća demografija

Sljedećih navedenih pet grafova na slikama 2.12 do 2.14. prikazuju opće demografske podatke ispitanika. Graf ”Spol” prikazuje spol osoba koje su ispunile anketu, graf ”Dob” njihovu dob, a treći stupanj trenutnog obrazovanja (Graf ”Obrazovanje”).

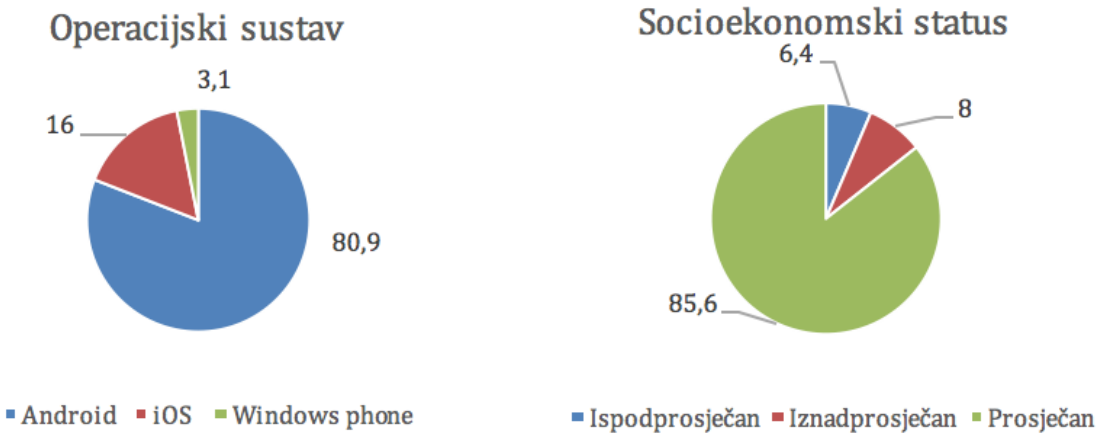


Sl. 2.12 *Grafički prikaz spola i dobi ispitanika.*



Sl. 2.13 *Grafički prikaz trenutnog ili završenog obrazovanja ispitanika.*

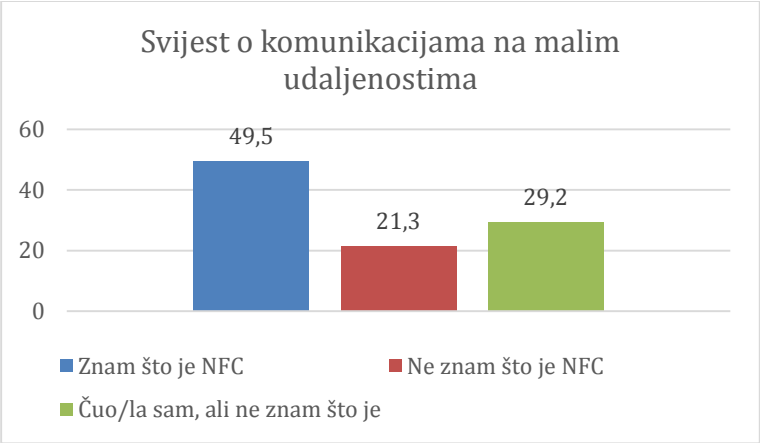
Preostala dva grafa na slici 2.14 prikazuju financijsku situaciju ispitanika, tako da graf 4 (”Operacijski sustav”) prikazuje odnos mobilnih operacijskih sustava korištenih od strane ispitanika, dok zadnji graf (”Socioekonomski status”) prikazuje njihov socio-ekonomski status.



Sl. 2.14 *Grafički prikaz socioekonomskog statusa ispitanika i pametnog telefona kojeg koristi.*

P2 - Svijest o komunikacijama na malim udaljenostima

Prvo konkretno pitanje o temi rada provjerava svijest ispitanih o *NFC* tehnologiji. Istraživanje je pokazalo kako 49,5% ispitanih zna što je *NFC*, dok njih 21,3% nikad nije čulo za *NFC*, što je značajan broj, a 29,2% ispitanika je čulo no ne znaju točno čemu služi. Prilagodba nove tehnologije će biti teža, budući da 50,5% ispitanika nije dobro informirano. Grafički prikaz rezultata svjesti o komunikacijama na malim udaljenostima prikazan je slikom 2.15.



Sl. 2.15 *Grafički prikaz svijesti o komunikacijama na malim udaljenostima.*

P3 - Jednostavnost korištenja i korisnost beskontaktnog mobilnog plaćanja

Pitanje 3 provjerava što ispitanici misle i kako zamišljaju transakcije u budućnosti te kako ih žele izvršavati - kartičnim putem, gotovinskim plaćanjem, putem mobilnih uređaja ili općenito beskontaktnom tehnologijom. Kao korisnik beskontaktnog plaćanja u budućnosti se vidi 49,5% ispitanih, što je jednako broju ispitanika koji su odmah znali o čemu je riječ na pitanje o *NFC*

tehnologiji. Novčanim putem još uvijek želi plaćati 29,3% ispitanih, dok se za kreditne kartice opredijelilo 21,3% ispitanih. U konačnici, to znači da bi svaki drugi ispitani koristio prednosti i usluge mobilnog plaćanja bez ikakve dodatne prisile da ga koristi, dok za 21,3% ispitanih je potrebno prilagoditi određene reklame i informirati ih zašto je dobro koristiti novu platnu tehnologiju, a poseban fokus na informiranje o novoj tehnologiji treba usmjeriti prema 29,3% ispitanih, koji nisu izrazili želju za korištenjem digitalnog načina plaćanja.

P4 - Davanje prednosti NFC mobilnim uređajima ispred plaćanja kreditnim karticama ili novčanicama

Ovo pitanje provjerava koliki postotak ljudi će odmah, bez imalo nagovaranja promijeniti svoje mišljenje, budući da je pitanje zapravo istog principa kao i prošlo, samo je ispitanicima naveden primjer kioska, gužve na njemu, točnije navedena im je situacija kada je idealno platiti beskontaktnim, bržim putem kako se ne bi stvarale gužve. Rezultat je da je 10,6% ljudi promijenilo svoje mišljenje i ipak se odlučilo za plaćanje putem mobilnih uređaja, što je već olakšalo put pri integraciji te vrste plaćanja i ostao je manji postotak koji je potrebno učiti o novoj tehnologiji i njenim prednostima.

P5 - Spremnost ispitanih na beskontaktna plaćanja u budućnosti

Pitanje P5 vrlo je blisko s prošla dva pitanja, ali prikazuje koliki dodatni postotak ispitanih smatra kako će ipak u budućnosti biti spremno za beskontaktna mobilna plaćanja, što se može iskoristiti u svrhu analize uspješnost i iskorištenost beskontaktne platne tehnologije na području Republike Hrvatske. U odnosu na prošlo pitanje, gdje je rezultat porastao za 10,6%, u ovom pitanju 72,3% ispitanih odgovorilo je kako je spremno na beskontaktna plaćanja što je još jedan dodatni porast od 12,2% u razmaku od samo dva pitanja. Na temelju ovog pitanja, moguće su daljnje, detaljnije analize korištenja sustava i optimistično se može zaključiti da je i Republika Hrvatska spremna za promjene platnih usluga.

Osim što se povećao broj onih koji bi mogli biti spremni i koristiti beskontaktna plaćanja, smanjio se broj onih koji su rekli kako žele koristiti novčanice ili kartice. Taj broj je sada 27,7%, od kojih je 12,8% reklo odgovorilo ”NE” dok je njih 14,9% odgovorilo ”nisam spreman”. U daljnjim psihološkim analizama, ova dva odgovora mogu biti dodatno analizirana, ali je za sada dovoljno napomenuti kako je broj onih koji još uvijek nisu spremni 27,7%.

P6 - Korisnik vjeruje u tehnologiju beskontaktnog plaćanja

Na ovo pitanje 75,5% ispitanih vjeruje u svijetlu budućnost plaćanja beskontaknim putem, dok njih 24,5% nema dobro mišljenje. Zaključno, 7-8 ljudi od 10 ispitanih ima povjerenja u beskontaktnu tehnologiju. Taj broj je dosta zadovoljavajući i daje na kraju ovog istraživanja

dosta pozitivno razmišljanje o tehnologiji beskontaktnog plaćanja i njenoj implementaciji u Republici Hrvatskoj.

P7 – Prešao/la bih u banku koja nudi mogućnost plaćanja putem pametnih uređaja

Koliko tehnologija upravlja korisnicima, a da oni to zapravo nisu ni svjesni bilo je pitano u sljedeća dva pitanja, gdje je 41% ispitanih odgovorilo kako bi sigurno promijenili banku ako ta banka nudi novi oblik plaćanja putem pametnih uređaja, njih 38,3% još uvijek nije sigurno što bi napravilo u tom trenutku, dok je 20,7% ispitanih reklo kako to ne bi napravilo. Nakon ovog pitanja jasno je vidljivo kako se tržišno poslovanje može povećati razvojem tehnologije i ulaganjem u nju, gdje postoji dosta velik postotak korisnika koji bi promijenili banku samo kako bi mogli uživati u novim tehnologijama.

P8 - Koristio/la bih samo trgovačke lance koji nude mogućnost plaćanja putem pametnih uređaja

Na ovo pitanje od ukupnih ispitanika, 13,8% osoba bi koristilo samo trgovačke lance koji im omogućuju programe vjernosti te beskontaktno plaćanje putem mobilnih uređaja u suradnji s bankama, 45,7% ne mari koju tehnologiju trgovački lanac koristi, dok 40,4% ispitanih misli kako to nije bitno i kako to ne bi napravili.

P9 – Koliko često kupujem

Ovo pitanje postavljeno je kako bi se analizirala zarada koju je moguće dobiti kroz svaku provedenu transakciju. 56,4% korisnika kaže kako kupuje na dnevnoj, 36,2% na tjednoj i 7,4% na mjesečnoj bazi.

P10 - Budućnost beskontaktnih transakcija u Hrvatskoj

Zadnje pitanje bilo je u vezi uvođenja novog mobilnog platnog sustava direktno u Republici Hrvatskoj te odnos između razmišljanja ispitanika i spremnosti države da kroz tehnologiju i kroz društveni napredak to i ostvari. U rezultatima, 55,3% korisnika kaže kako je Republika Hrvatska potpuno spremna uvesti beskontaktna mobilna plaćanja, dok njih 44,7% misli kako se još uvijek treba ulagati u obrazovanje korisnika i infrastrukturu te da društvo nije spremno za digitalnu revoluciju.

U anketi je postavljeno 14 pitanja od kojih je pet analizirano u pod poglavlju P1, te uz P8 i P9 neće biti korišteno za pregled analize, što znači da su potpoglavlja od P2 do P7 i poglavlje P10 uključena.

Tablica 2.4. prikazuje sažete rezultate ankete.

Tab. 2.4 *Sažeti rezultati ankete.*

Broj pitanja	Rezime rezultata		
	Testirano	**Pozitivno%**	**Negativno%**
P2	Svijest	49,50%	50,50%
P3	Korisnost/Jednostavnost	49,50%	50,50%
P4	Korisnost/Jednostavnost	60,10%	39,90%
P5	Spremnost na promjenu	72,30%	27,70%
P6	Povjerenje	75,50%	24,50%
P7	Spremnost na promjenu	41,00%	59,00%
P10	Spremnost na promjenu	55,30%	44,70%
P (Ukupni[49])	**U (Zbroj)**	**403,20**	**296,80**

Iz tablice 2.4. analizom ukupnih pozitivnih i negativnih postotaka izračunata je srednja vrijednost svih odgovora na pitanja i ustvrđeno je kako se 6 od 10 ispitanih slaže da društvu trebaju promjene i da će koristit mobilne beskontaktne transakcije. Zaključno, Republika Hrvatska prema ispitanicima iz ove ankete ima pozitivnu budućnost korištenja mobilnih beskontaktnih transakcija, samo treba uložiti u edukaciju korisnika i njihovo upoznavanje s novom tehnologijom i novim, bržim načinom života.

Koristeći se teorijskim okvirima kao metodom istraživanja i empirijskim (iskustvenim) podacima na pitanjima P3 i P4, ispitana je korisnost i jednostavnost sustava te mogućnost promijene razmišljanja na temelju iskustvenog pitanja. Zbog toga se pitanja P3 i P4 mogu dodatno analizirati i razdvojiti te se dobiveni rezultat mjeri u 54,80% ((49,5+60,1)/=54,8) u pozitivnom i 45,20% ((50,50+39,9)/2=45,2) u negativnom smjeru.

Lewinov sustav upravljanja promjenama [43] korišten je kako bi se uvidjela spremnost korisnika na promjenu. Kao rezultat ove analize uzeta su u obzir dva pitanja, P5 i P10, te su na temelju njih doneseni zaključci, prema istim podacima iz tablice 2.4. Iskustvena analiza, prikazuje još veći pozitivni postotak korisnika i iznosi 63,80%, dok je postotak negativnih 36,20%. Zadnje, iskustveno istraživanje donijelo je značajne rezultate, budući da se Republika Hrvatska nalazi u trenutku kada bi se trebala tehnološki još više uzdići i promijeniti način razmišljanja i poslovanja, uključiti u digitalno doba i pokušati iskoristiti sve prednosti tehnologije, kako bi se mogla dalje razvijati kao globalni konkurent i dodatno zarađivati na svojim digitalnim proizvodima. Lewinova iskustvena analiza daje veći postotak od prvotnog pitanja svijesti o *NFC* usluzi, što je zanimljiv faktor ako se želi još detaljnije analizirati kombinacija odgovora.

Nakon provedene temeljite ankete i analize različitim metodama može se reći kako je ispitani dio stanovništva Republike Hrvatske spreman na beskontaktni način plaćanja te kako postoji mogućnost da se i ostali nezainteresirani stanovnici putem iskustvene analize i primjera pridobiju

za sudjelovanje u digitalnoj transformaciji bankarstva. Digitalnom promjenom stanovnicima se omogućuje uživanje potpuno drugačijih proizvoda i usluga te ih se potiče na samostalni rad i daje mogućnost novog načina poslovanja. Obrazovanjem i upoznavanjem stanovnika s osnovnim tehnološkim aspektima pružaju im se mogućnosti zarade, uštede vremena i novca te jedan potpuno drugi pristup i pogled na svijet. U današnjem društvu, takav oblik informiranja stanovnika vrlo lako se može iskoristiti u ekonomiji dijeljenja, koja je budućnost poslovanja i sve više se teži njoj.

3 KOMPONENTE RAZVOJNE OKOLINE

U ovom poglavlju navedene su i detaljnije opisane komponente i način rada komponenti potrebnih za razvoj tehničkog dijela diplomskog rada.

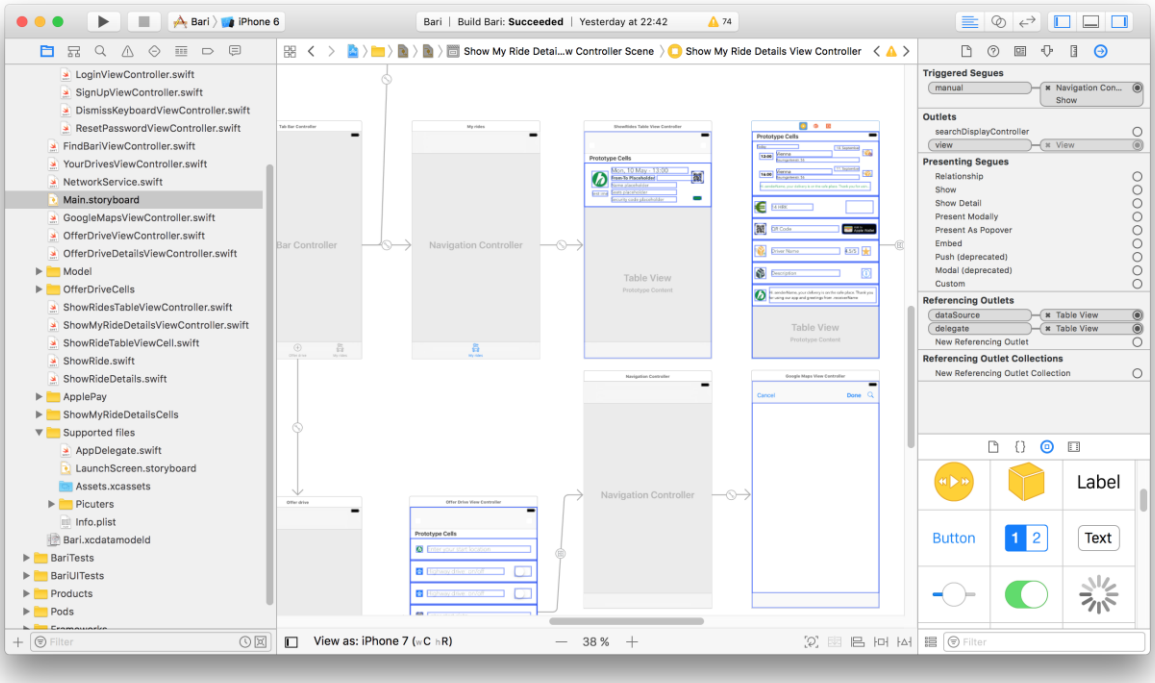
3.1 Razvojna okolina

3.1.1 Xcode

Razvojno okruženje (engl. *Integrated developer environment*) za razvoj programske podrške za *Apple* platformu naziva se *Xcode*. Pomoću njega razvojni inženjeri mogu objaviti svoj kôd za jednu od navedenih platformi: *iPhone, iPad, Mac, Apple Watch,* i *Apple TV*. *Xcode* se ne naplaćuje dodatno, besplatni operacijski sustav koji dolazi uz standardne vrste *Apple*ovih računala. Svakom novom inačicom *Apple* omogućuje pregršt novih funkcionalnosti i mogućnosti koje omogućuju jednostavniji razvoj aplikacija za *Apple* platforme te jedno potpuno novo korisničko iskustvo. Uzevši u obzir da je razvoj *iOS* aplikacija predviđen na *Mac* računalima s kompaktnim sklopovljem i programskom podrškom, govori da je *Xcode* jedan nevjerojatan spoj, središte *Apple* razvoja, koji je čvrsto integriran s *Cocoa* i *Cocoa Touch* razvojnim okvirima što za rezultat daje najbolje aplikacije na tržištu.

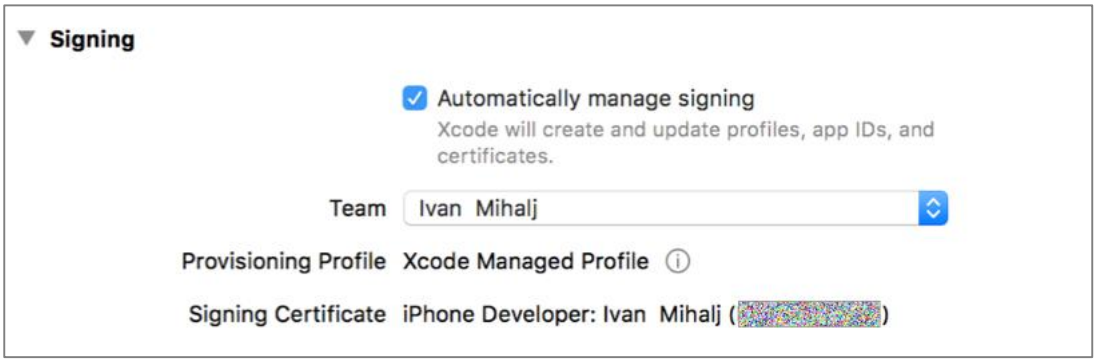
Xcode čine sljedeće komponente: *Source editor, Asset Catalog, Assistant editor, Open Quickly, Version Editor, Open GL Frame Capture, Interface Builder Built In, Complete Documentation, iOS Simulator, Live Issues, Integrated Build System, Fix-it, Compilers, Quick Help, Graphical Debugger, XCTest Framework, Continuous Integration, Static Analysis, Instruments, UNIX Tools, Compilers, Script Languages* i puno drugih.

Na slici 3.1 se vidi kako izgleda *Xcode* korisničko sučelje te je prikazano nekoliko zaslona iz aplikacije. S lijeve strane sučelja nalazi se navigacijski dio zaslona unutar kojeg su spremljene sve klase i prateći dokumenti (engl. *supporting files*), u sredini je dio razvojnog okruženja u kojem razvojni inženjer trenutno radi, u ovom slučaju dio za razvoj korisničkog sučelja aplikacije (engl. *Storyboard*)[50]. S desne strane okruženja *Xcode* nalaze se usluge i podrške koje razvojni inženjer koristi prilikom razvoja aplikacije. Pri dnu zaslona nalazi se dodatno korisničko sučelje za nadgledanje grešaka (engl. *Debug area*) koje se prikazuje pokretanjem aplikacije. Svaki opisani dio može se uključiti ili isključiti ovisno u kojem trenutku je potreban razvojnom inženjeru te osloboditi zaslon računala za bolji ugođaj pisanja programskog kôda.



Sl. 3.1 *Xcode korisničko sučelje.*

Na slici 3.2 prikazano je sučelje za unos certifikata programa razvojnog inženjera bez kojeg se ne može zatražiti niti jedan drugi *Apple*ov certifikat i nastaviti razvijati aplikacija za potrebe ovog diplomskog rada.

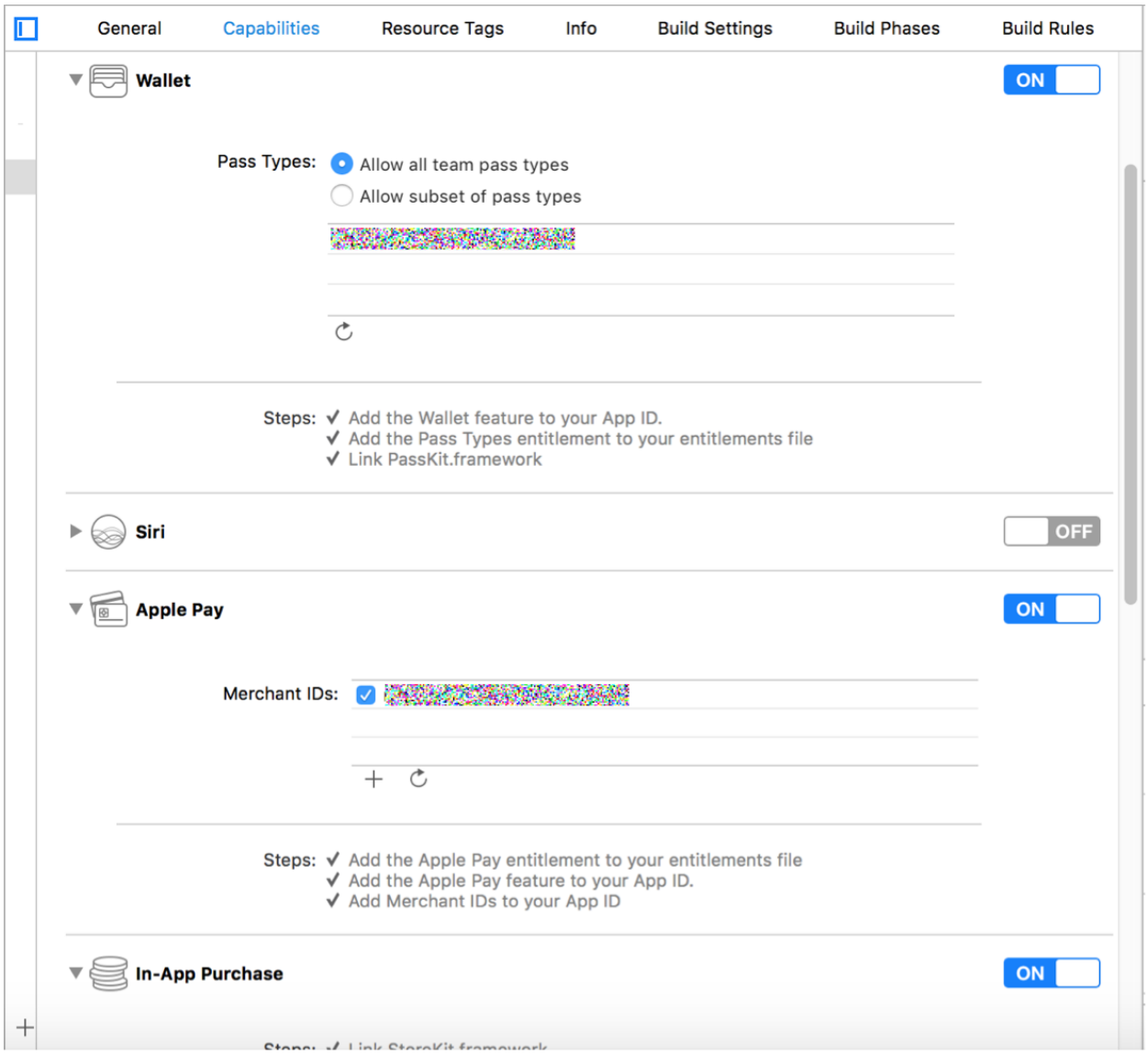


Sl. 3.2 *Xcode razvojni certifikat.*

Slika 3.3. prikazuje odobravanje certifikata za razvoj diplomskog rada s *Apple* strane unutar *Xcode* razvojnog okruženja. Certifikati odobreni za razvoj su:

- *Apple Wallet* certifikat
- *Apple Pay* certifikat
- Certifikat za plaćanje unutar aplikacija

Više o certifikaciji spomenuto je u poglavlju 3.1.5.



Sl. 3.3 *Certifikati Xcode Wallet, Apple Pay i In-App Purchase.*

3.1.2 Simulator

Kako bi se omogućilo svakom početniku razvoj sustava bez dodatnog troška za *iOS* uređaje, a razvojnim inženjerima pružila jednostavnija i brža interakcija na različitim veličinama zaslona, *Apple* ima *Simulator* koji unutar *Xcode* okruženja pruža zamjenu za fizički uređaj. Pokrećući *Simulator* i testirajući kôd na njemu, on se ponaša poput standardne *Mac* aplikacije i koristi resurse dostupne u *Mac* računalu - središnju jedinicu za obradu (*CPU*), mrežu i memoriju. Za svaki početak, *Simulator* je vrlo dobro okruženje za testiranje, dok svi budući, malo ozbiljniji projekti, bi ipak trebali biti testirani na fizičkom uređaju, kako za testiranje prilikom razvoja, tako i za završno ili *sprint*[51] testiranje proizvoda u fazi prije isporuke ili završetka određenog dijela ciklusa razvoja proizvoda. Osim testiranja mobilnih aplikacija, *Simulator* pruža podršku i simulaciju *watchOS* te *tvOS* uređaja. Svaki simulirani uređaj i njegova popratna programska

podrška imaju svoj važeći operacijski sustav, neovisan o drugom, s vlastitim postavkama i podacima koji postoje i na fizičkim uređajima.

Kao što je i navedeno, *Simulator* ima svoje prednosti i nedostatke, vrlo je koristan prilikom testiranja korisničkog sučelja koje se na vrlo jednostavan način mogu obavljati interakcijom miša, no već kod testiranja ozbiljnijih i kompleksnijih programskih rješenja, koji moraju imati popratnu podršku i sklopovlja na kojem se nalaze, *Simulator* postaje ograničen i nedovoljan. Razlike u sklopovskim i *API* sastavnicama *Simulatora* su:

- Kretanje – akcelerometar i žiroskop
- Audio i video ulaz su nepodržani – kamera i mikrofon
- Senzor blizine (engl. *proximity sensor*)
- Barometar
- Senzor svjetla
- Obavijesti unutar uređaja (engl. *Push notification*)
- Sigurnosna upozorenja za pristup lokaciji, kameri, kontaktima, kalendaru, podsjetniku
- *UIBackgroundModes* (Govori da aplikacija treba nastaviti radit u pozadini)
- Početi raditi dokument, e-mail ili poruku na jednom Apple uređaju, a nastaviti na drugom (engl. *handoff support*)
- Reprodukcija multimedije

Svaki novi *iOS* uređaj nosi neko tehnološko unaprjeđenje što ponekad znači i nove vrste senzora. Neke od najzanimljivijih senzora na trenutno dva najnovija *iOS* uređaja su senzor blizine (engl. *proximity*), svjetla, akcelerometar, žiroskop, kompas, barometar, komunikacije bliskog polja (engl. *NFC, Near Field Communication*) za plaćanje, senzor otiska prsta, senzor pritiska.

3.1.3 GitLab

Razvoj ozbiljne agilne (engl. *agile*) [45] programske podrške zahtijeva i određene sustave za spremanje različitih inačica kôda. Tako je za potrebe ovog diplomskog rada korišten besplatni sustav za praćenje i razvoj više-verzijskog kôda, *GitLab* [46].

GitLab je web bazirani sustav praćenja promjena u inačicama kôda (engl. *version control system*[52]) koji omogućuje koordinaciju i suradnju više osoba na jednom projektu, te tako olakšava proces razvoja programskog rješenja. Razvoj i pohranjivanje kôda na *GitLab* poslužitelje koristio se za svaku veću značajku sustava, što znači da je svaka značajka imala svoju posebnu granu pohrane koja se nakon završetka razvoja zatvorila i spojila (engl. *merge*) s glavnom granom projekta.

Osim za spremanje inačica kôda aplikacije, *GitLab* je korišten i za stvaranje gotovih inačica *iOS* aplikacije. Spremanjem više inačica gotove aplikacije, ručno i automatsko testiranje aplikacije postaje lakše i omogućuje detekciju problema prije samog kraja razvojnog ciklusa.

3.1.4 SourceTree

SourceTree je grafičko korisničko sučelje koje se koristi za rad s *Git* programskim rješenjima u svrhu upravljanja repozitorijem nastalom lokalno na računalu i njegovom sinkronizacijom s web klijentima koji pohranjuju izmjene kôda.

3.1.5 Firebase

Firebase je *Google*ovo razvojno okruženje za pomoć pri izradi web i mobilnih rješenja. Sve potrebno za razvoj programskog rješenja nalazi se na jednom mjestu u oblaku računala. Pohrana podataka u stvarnom vremenu, sustav za praćenje rušenja aplikacije (engl. *crash reporting*), kreiranje i upravljanje korisnicima, udaljena memorija na oblak računalu, smještaj i održavanje internetske stranice (engl. *hosting*), testiranje programskog rješenja i mjerenje učinkovitosti aplikacije (engl. *performance monitoring*), analitika, poruke u stvarnom vremenu te oglašavanje i mnoge druge opcije.

Za potrebe diplomskog rada, *Firebase* sustav korišten je za sigurnosno kreiranje računa korisnika, njegovu registraciju, prijavu i poništavanje lozinke u stvarnom vremenu te za dohvaćanje trenutnog korisnika prijavljenog u sustav i izvještavanje pozadinskog usluga prilikom dohvaćanja detalja potrebnih za neometani rad aplikacije.

3.1.6 Postman

Postman [47] je programsko rješenje za imitaciju klijentskih zadaća prema poslužitelju. Sučelje koje šalje zahtjeve prije nego li se oni implementiraju unutar aplikacije i spoje s pozadinskim uslugama. Ovakva vrsta usluga ima jedinstveni *URL* s kojeg se pristupa udaljenim podacima putem *HTTP* protokola. Svaki poziv jedinstvenog *URL*a ima jednu od akcija koju izvršava, GET, POST, PUT, DELETE.

3.1.7 Sketch

Sketch je profesionalno razvojno okruženje za razvoj korisničkog sučelja na *Mac* uređajima. Osim razvojnog okruženja na *Mac* uređajima, *Sketch* pruža i *iOS* mobilnu aplikaciju za razvoj i testiranje korisničkog sučelja, kako bi se u stvarnom vremenu vidio odziv kreiranih korisničkih sučelja za *iOS* uređaje direktno na *iPhone* uređaju.

3.1.8 Certifikacija i plasiranje proizvoda

Zahtijevanje, preuzimanje i korištenje *Apple*ovih certifikata nije moguće bez učlanjenja u Appleov program razvojnih inženjera (engl. *Apple Developer Program membership)* i posjedovanja *Apple*ovog *ID*a, stoga je prvi korak, a ujedno i prvi certifikat potreban za rad s *Apple*ovom programskom podrškom učlanjenje u program razvojnih inženjera. Program se sastoji od dvije vrste certifikata, certifikat za razvojne inženjere i razvojni certifikat za poduzeća. Razlika u cijeni je 200 dolara, gdje će razvojni inženjer za osnovnu, privatnu inačicu morati platiti 99 američkih dolara godišnje, a poduzeće za isto vremensko razdoblje 299 američkih dolara.

Certifikati korišteni prilikom izrade diplomskog rada vidljivi na slici 3.4 su:

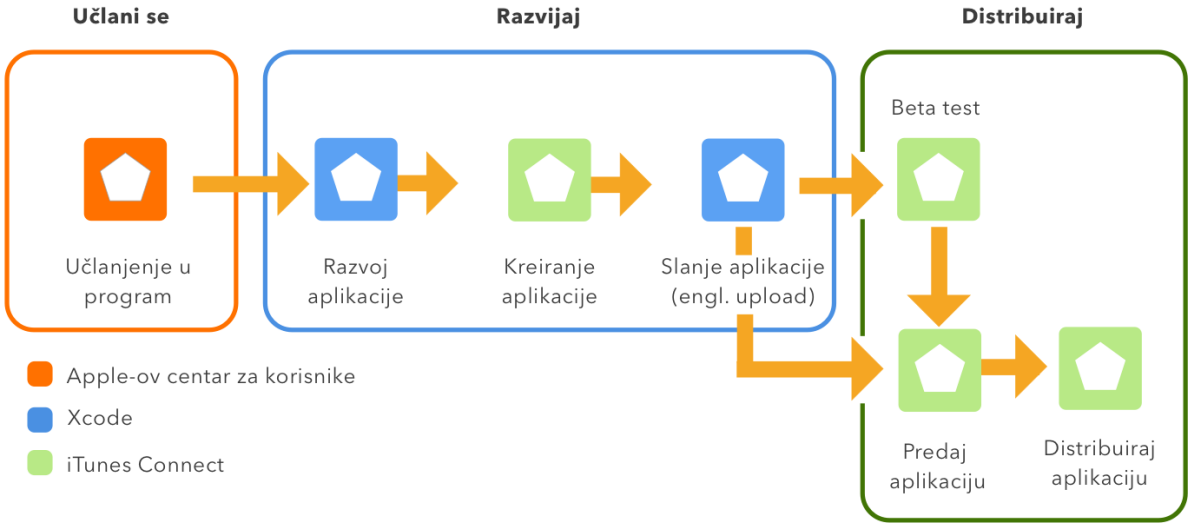
- Razvojni certifikat (engl. *iOS development certificate*) [48]
- Certifikat identiteta trgovca (engl. *Merchant Identity Certificate*) [49]
- *Apple Pay* certifikat
- *Stripe* certifikat

Ostali certifikati koji nisu korišteni u trenutnoj inačici programske inačice aplikacije, a potrebni su za daljnji razvoj aplikacije su:

- Certifikat za slanje obavijesti (engl. *Apple Push Notification Service Certificate*)
- Distribucijski certifikat (engl. *iOS Distribution Certificate – App Store*)
- Distribucijski certifikat za internu upotrebu (engl. *iOS Distribution Certificate, in-house*)

Sl. 3.4 *iOS certifikati.*

Nakon prikupljanja svih potrebnih certifikata za razvoj i objavljivanje *iOS* aplikacije, aplikaciju je potrebno programski razviti, testirati i na kraju plasirati na tržište putem trgovine aplikacijama. Slika 3.5 prikazuje proces od učlanjenja, razvoja, testiranja, do predaje i distribucije aplikacije.



Sl. 3.5 *Ciklus distribuiranja aplikacije.*

3.1.9 Biblioteke

Razvoj programske podrške znatno je olakšan mogućnošću korištenja biblioteka za razvoj programskog kôda, gdje je razvojnom inženjeru omogućeno korištenje gotovih klasa i objekata ili njihovu promjenu gdje će kôd biblioteke biti baza za neku novu vrstu implementacije.

U razvoj *iOS* aplikacije za potrebe diplomskog rada korištene su sljedeće biblioteke:

- *Foundation, UIKit*
- *Firebase, FirebaseAuth*
- *Google Mape, Google Mjesta*
- *PassKit*
- *Cocoa i Cocoa Touch*
- *Almofire*

3.1.10 Touch ID

Touch ID je sustav koji korisniku omogućuje sigurnosni pristup svom uređaju na jednostavniji i brži način. Tehnologija funkcionira na načelu prepoznavanja otiska prsta iz svih kutova, pri čemu je potrebno samo u postavkama kreirati otisak za svaki prst koji korisnik želi koristiti. Daljnjim će korištenjem *iOS* programska podrška putem umjetne inteligencije naučiti sve varijante i kutove pokušaja otključavanja te proširiti mapu koja pohranjuje sve uspješne

kombinacije otključavanja. Ta mapa može sadržavati do maksimalno pet različitih prstiju. Mogućnost da se nasumičnim odabirom uređaj otključa otiskom prsta neke druge osobe je 1:50000, ali je i tu postavljena blokada. Stoga nakon pet neuspjelih očitavanja otiska korisnik mora upisati svoju lozinku.

Korištenjem tehnologije biometrijskog prepoznavanja otiska prsta, korisnicima osim najsigurnijeg načina zaključavanja/otključavanja uređaja omogućuje i dodatnu sigurnost kreiranjem kompliciranije lozinke za otključavanje uređaja, budući da lozinka vrlo vjerojatno neće biti više korištena toliko često, može se staviti teža kombinacija ili čak uključiti i specijalne znakove i slova u lozinku, što će u konačnici od *iOS* uređaja napraviti još sigurniju napravu. Postavljanje lozinke moguće je bez *Touch ID* provjere, ali obrnuti slučaj nije moguć i u nekim slučajevima lozinka je još uvijek korištena ispred senzora otiska prsta:

- Pri ponovnom pokretanju uređaja
- Uređaj nije otključan više od 48 sati
- Lozinka nije korištena za otključavanje u zadnjih 6 dana i *Touch ID* nije otključao uređaj u zadnjih 8 sati
- Uređaj je primio udaljenu komandu za zaključavanje
- Nakon pet neuspješnih pokušaja *Touch ID* otključavanja
- Prilikom postavljanja novog prsta u *Touch ID* sustav

Korištenjem *Touch ID*a kao primarne metode otključavanja *iOS* uređaja, postoji još par pravila kada je uređaj moguće otključati. Prilikom korištenja lozinke, pritiskom na gumb za zaključavanje uređaja, moguće je narediti da sustav još neko vrijeme ne traži lozinku prilikom povratka u sučelje uređaja, u slučaju da korisnik učestalo koristi programsko okruženje – kod *touch ID* senzora to nije moguće, prilikom zaključavanja uređaja, obavezna je ponovna provjera korisnika putem senzora.

Touch ID senzor reagira samo ako je sklopovlje osjetilo otisak prsta, tada okida imaginarni spremnik kombinacija otiska prsta kako bi skenirao i provjerio valjanost otiska te ga šalje na završno skeniranje u sigurnosnu enklavu. Sken otiska prsta je prvotno spremljen u šifriranu memoriju unutar sigurnosne enklave, koja ga za vrijeme provjere analizira i provjerava sa spremljenim kutovima otiska prsta. Rezultati trenutnog mapiranja i pretrage otiska prsta, spremljen je samo unutar enklave i nikom drugom nije javno dostupan, što znači da može biti samo očitan na uređaju gdje je pohranjen, a *Apple*, njegovi pozadinski usluga, *iCloud* ili *iTunes*, o tome nemaju nikakvih informacija. Ti rezultati ako je otisak koji je unesen jednak jednom od spremljenih otisaka, preko sigurnosne enklave stvaraju ključ za razmotavanje (engl. *unwrap*) zaštićenih podataka i uređaj je otključan.

Uz sve značajke koje *TouchID* pruža, u diplomskom radu je bitan zbog toga što se preko njega obavlja validacija korisnika prilikom obavljanja svake nove transakcije unutar *Apple*ovih digitalnih trgovina (*iTunes*, *App Store*, *iBook Store*), prilikom kupovanja dobara u fizičkim trgovinama, putem Internet kupnje ili unutar aplikacija.

3.1.11 Izvanmrežne transakcije

Apple Pay sustav može djelovati u izvanmrežnom i u zrakoplovnom okruženju. Razlog funkcioniranja platforme u okolišu bez pristupa Internetske veze je taj što prilikom kreiranja transakcije *iOS* uređaj nema nikakvog dodatnog spajanja sa poslužiteljima davatelja kartičnih usluga ili s *Apple* transakcijskim poslužiteljima. Spajanje na oba navedena poslužitelja odvija se samo prilikom prvog kreiranja kartice unutar *Apple Wallet* okruženja.

Plaćanje u trgovačkim lancima, za korisnike *iOS* uređaja je omogućeno i u izvanmrežnom jer u tom trenutku *iOS* uređaj ima ponašanje kao kreditna kartica te služi samo za ovjeru i provjeru valjanosti kartičnih podataka. (i putem *NFC* čipa i otiska prsta šalje token na POS uređaj)

Ako *Apple Pay* račun nije unaprijed konfiguriran, sustav neće funkcionirati, što znači da ni s internetom, a ni bez njega korisnik neće moći izvršiti transakciju.

Izvršavanje transakcija unutar *iOS* aplikacije razvijene u diplomskom radu omogućeno je samo uz prisustvo Internet veze, budući da se aplikacija trećeg reda direktno spaja s klijentom za pružanje kartičnih usluga *Stripe* i s pozadinskim uslugama razvijenim za potrebe diplomskog rada. Ako korisnik nema pristup internetu, neće se moći prijaviti u sustav niti će moći dohvaćati, spremati, koristiti podatke unutar aplikacije pa tako ni izvršavati plaćanja.

3.1.12 NFC

Uvođenjem kartičnog plaćanja, sve ostale platne tehnologije (korištenje novca) su se polako smanjivale, a novi način plaćanja je bio prikladniji, jednostavniji i ono što je najbitnije vremenski efikasniji. Ideja o ugradnji magnetne trake na karticu te korištenje *PIN*a kako bi se transakcija uspješno evaluirala, dovela je do novih spoznaja kako transakcije mogu funkcionirati. Sve navedeno bila je preteča zamislima da i mobilne uređaje, tzv. pametne telefone pretvore u digitalne lisnice, budući da su postali neizostavni dio ljudske svakodnevnice.

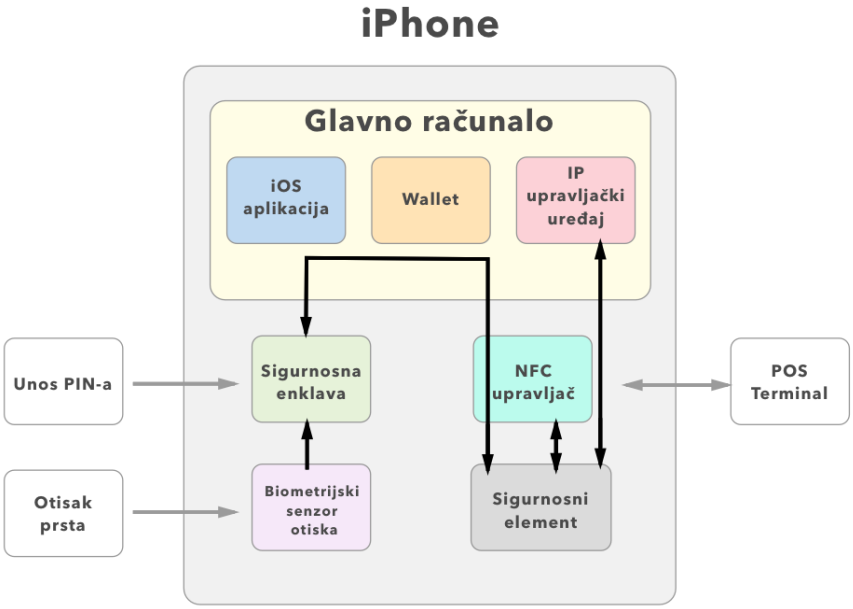
Zamisao o digitaliziranju lisnice putem pametnih telefona mogla je biti i ostvarena je koristeći *NFC* tehnologiju. *NFC* tehnologija je bazirana na *RFID* (engl. *Radio Frequency Identification*) komunikaciji, koja omogućuje direktnu komunikaciju mobilnog uređaja i *NFC* baziranog terminala, tako da mobilni uređaj mora biti blizu senzora blizine (engl. *proximity sensor*) terminala na kojem se želi izvršiti transakcija.

3.1.13 NFC i Apple Pay

Budući da sigurnosni element sadrži sve važne podatke za provođenje sigurne transakcije, direktna komunikacija između njega i *POS* uređaja nije sigurnosno dobra tako da se ta komunikacija odvija u suradnji s *NFC* upravljačem. Proces izvršavanja transakcije nakon što je korisnik ovjeren putem biometrijskog senzora ili lozinke na svom iPhone uređaju ili pak dvostrukim klikom na svom *Apple Watch* uređaju ako ga posjeduje, odvija se kao što je rečeno preko sigurnosnog elementa koji ne šalje transakcijsku ovjeru nigdje dalje od *NFC* upravljača u slučaju komunikacije s *POS* uređajem, dok u slučaju autorizacije unutar aplikacija ili plaćanja preko weba je proslijeđen do aplikacijskog procesora, ali tek nakon što je prvo šifriran preko sigurnosnog elementa unutar *Apple Pay poslužitelja*.

Pasivni RFID šalje informacije samo u slučaju kada je *NFC* čitač to zatražio od njih te im za to ne treba nikakvo dodatno napajanje budući da se oni napajaju pomoću elektromagnetskih valova nastalih od samog *NFC* čitača. (npr. beskontaktne bankovne kartice).

Aktivni *RFID* sustav koristi uređaje koji imaju stalni pristup baterijskom napajanju i neprestano šire vlastite signale. Primjer aktivnog *NFC* čitača je ”*beacon*” koji prati lokaciju u stvarnom vremenu. Uređaji s implementiranim aktivnim *RFID* sustavom imaju dugoročniji vijek, ali su dosta skuplji od pasivnih [50]. Slika 3.6. prikazuje iPhone komponente Apple Pay sustava.



Sl. 3.6 *Unutarnje i vanjske iPhone komponente potrebne za korištenje Apple Pay sustava ma* **[51]**

4 STRUKTURA PROGRAMSKOG RJEŠENJA

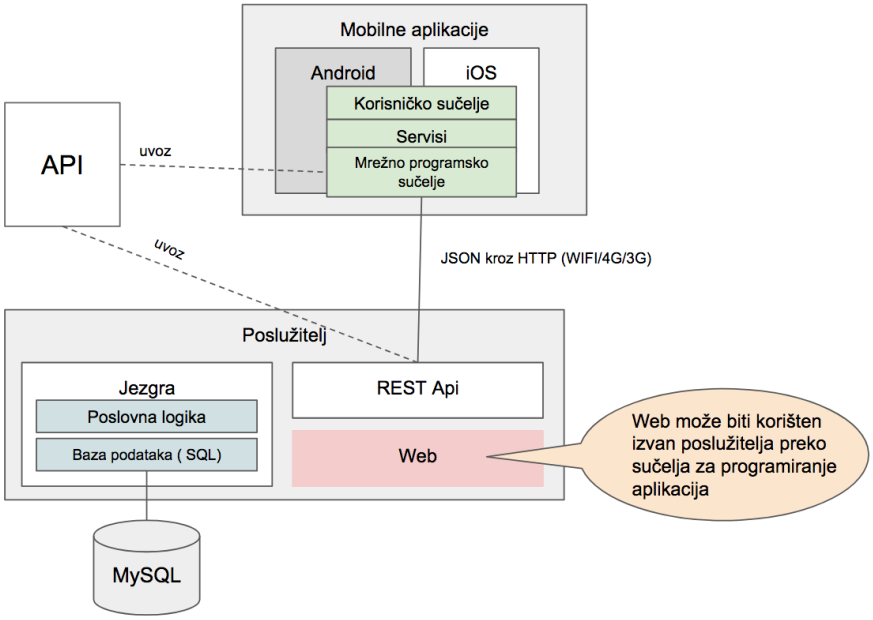
Programsko rješenje diplomskog rada bazirano je na razvoju *iOS* aplikacije za dijeljenje transportnog mjesta te naplaćivanje usluge za istog. Razvoj složenog sustava ne može zadovoljiti uvjete kvalitetnog razvoja programske podrške ako se cijeli sustav bazira samo na *iOS* platformi. Zbog toga su za ovaj diplomski rad korišteni i dodatni sustavi koji pomažu stvaranju nadogradive (unaprjeđene) i dinamične programske podrške i omogućuje brži i kvalitetniji rad aplikacije.

4.1 Arhitektura aplikacije

Arhitektura sustava sastoji se od slojeva koji su osmišljeni da se u budućnosti mogu nadograditi i proširiti na ostale platforme kao što su Web sučelja, desktop aplikacije, *Windows-Phone* ili *Android* aplikacije. Osim nadogradivosti na druge platforme, sustav je realiziran da dinamički, udaljeno upravlja sadržajem unutar aplikacije. Kvalitetan razvoj aplikacije ne dopušta kreiranje sporih i preopterećenih programskih rješenja. Zbog toga se svi podaci koji mogu pohranjuju izvan aplikacije budući da se jedino tako brzo i učinkovito mogu mijenjati sadržaji u aplikaciji.
Pohrana podataka vrši se dijelom na *Firebase* poslužitelju, a dijelom u bazi podataka kreiranoj na osobnom poslužitelju za razvoj ovog projekta, naravno postoje i dijelovi podataka koji se pohranjuju unutar aplikacije, ali njihova veličina je zanemariva. Svi važni podaci prikazani u aplikaciji, mogu se naknadno i udaljeno, putem web korisničkih sučelja mijenjati i ponovnim pokretanjem *iOS* aplikacije ažurirati i prikazati kao novi sadržaj.
Na slici 4.1 prikazana je arhitektura kompletnog sustava za transport, gdje se vide svi slojevi arhitekture razvijeni u diplomskom radu i slojevi arhitekture na koje će se ovaj projekt naknadno nadograđivati. Bitno je napomenuti kako su glavna sastavnica arhitekture sustava: *iOS* aplikacija te popratni *API* i baza podataka.

4.1.1 MVC programska paradigma

Razvojem načina pisanja programske podrške, aplikacije postaju sve zahtjevnije i prepune svojstvima kojima treba opreznije i sigurnije upravljati i znati koristiti ako se želi omogućiti normalan rad aplikaciji. Organizacija kôda i način razvoja mobilnih aplikacija (engl. *state-the-art*) vrlo je bitna stavka pri razvoju, stoga se razvila paradigma pisanja aplikacija kako bi se to i ostvarilo. *MVC* programska paradigma [52], omogućila je razvoj organiziranijih, urednijih i kvalitetnijih aplikacija kojima je lakše upravljati i nadograđivati ih kroz komunikaciju između tri vrste objekata u svrhu boljeg iskorištavanja kôda, njegove ponovne upotrebe, bolje implementacije u sustav te bržeg, kvalitetnijeg i sigurnijeg izvođenja.



Sl. 4.1 *Arhitektura kompletnog programskog rješenja*

4.1.2 Model

Model predstavlja podatke i upravlja logikom aplikacije. Budući da je model zapravo mozak aplikacije posvećen jednom određenom zadatku, isti taj model može biti iskorišten za drugi, sličan zadatak, bez da se mora ponovno pisati isti kôd. Model nema nikakve izravne veze ni kontakta s objektom izgleda (engl. *View*) koji prikazuje njegove podatke i omogućuje korisnikovu interakciju s njima. Promjene unutar model objekta su naravno moguće i česte, ali njihovim daljnjim korištenjem unutar aplikacije model ne upravlja. Za svu komunikaciju i promjene korisničkog sučelja zadužen je upravljač (engl. *controller*), koji je poveznica između modela i objekta izgleda.

4.1.3 Objekt izgleda

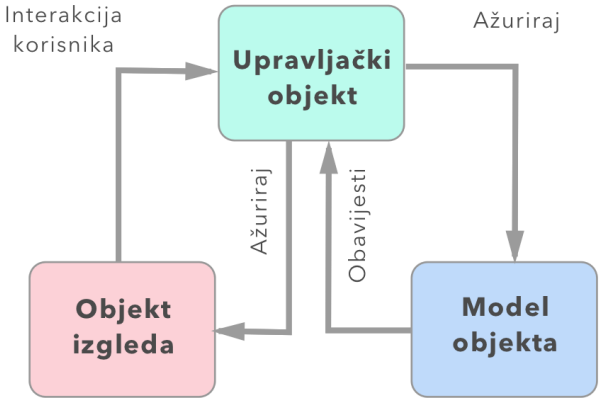
Nakon uspješno kreiranih podataka i logike aplikacije preko upravljačkog objekta ti isti podaci poslani su prema objektu izgleda aplikacije koji ih kreira na zaslonu uređaja. Sve vidljivo na zaslonu uređaja naziva se objektom izgleda i pruža korisniku interakciju s uređajem.

Izvršavanjem određenih akcija, korisniku je omogućeno upravljanje načinom prikaza podataka na zaslonu. Od promjene pozicije unutar aplikacije do interakcije s podacima pohranjenima u modelu. Izravna komunikacija između modela i objekta izgleda ne postoji, sve aktualne izmjene nastale na upravljačkom objektu, tek su omogućene za prikaz nakon što ih upravljački objekt dostavi objektu izgleda aplikacije.

U konačnici, objekt izgleda je korisničko sučelje aplikacije, sve što korisnik vidi na zaslonu i s čime može upravljati (tekstualna polja, gumbi…)

4.1.4 Upravljački objekt

Upravljački objekt direktna je poveznica između podataka i korisničkog sučelja u kojem će podaci biti prikazani. Svi podaci pohranjeni u modelu koji neposredno komuniciraju s određenim objektom izgleda unutar aplikacije prikazani su na zaslon uređaja putem upravljačkog objekta. Osim prikaza istih podataka i njihove pohrane, upravljač ima mogućnost osvježavanja korisničkog sučelja s novim podacima dospjelim s poslužitelja na model kojim komuniciraju. Upravljač osim komunikacijom upravlja i životnim vijekom trajanja određenog modela i upravlja načinom njegovog prikazivanja i sakrivanja sa zaslona.



Sl. 4.2 *MVC programska paradigma.*

4.1.5 Podatkvna veza

Kreiranje zahtjeva za spajanje s mrežom odvija se preko *HTTP* zahtjeva. Ako poveznica (engl. *Link*) kojoj se želi pristupiti nema kriptiranu komunikaciju i sigurnu identifikaciju web poslužitelja, pristup preko *HTTP* zahtjeva se prvo treba odobriti u ”*Info.plist*” dokumentu *Xcode* okruženja. Odobravanje se odvija tako da se doda nova ključ vrijednost (engl. *key*) pritiskom na plus znak unutar dokumenta, u tražilici upiše ”*App Transport Security Settings*” i postavi vrijednost ”*Allow Arbitrary Loads*” na istinitu (engl. *true*). Nakon postavljanja pristupa podacima preko manje sigurnog zahtjeva, aplikacija je spremna za korištenje metoda za dodavanje, dohvaćanje, promjenu i brisanje podataka.

Svi zahtjevi korišteni u diplomskom radu (Tab. 4.1) preneseni su putem *JSON* (engl. *JavaScript Object Notation*) transportnog formata. *JSON* je tekstualni standard za razmjenu podataka izveden iz *JavaScript* jezika, koji je u zadnje vrijeme zamijenio *XML*. Primjer *JSON*

transportnog oblika vidljiv je na slici 4.3. Opis i imena metoda korištenih u diplomskom radu prikazani su u tablici 4.1


```
{
“rides": [
   {
     "rideId": "3",
     "driverId": "222",
     "driverName": “Ivan”,
     "driverSurname": “Mihalj”,
     "ridePrice": “20”,
     "deliverPrice": “15”,
     "startPlaceDetailed": “Đakovo, Ante Starčevića 196b”,
     "toPlaceDetailed": “Osijek, Kampus”
     "highway": false,
     "roundTrip": true,
     "startDateAndTime": "2017-18-06T18:25:43.511Z",
     "estimatedEndDateAndTime": "2017-18-46T18:25:43.511Z",
     "freePeopleSpace": "3",
     "smokeFree": false,
   } ],
 "sortKey": "sortKey",
 "sortType": "sortType",
 "index": 1,
 "batchSize": 10
}
```


Sl. 4.3 *Prikaz JSON GET zahtjeva*

Tab. 4.1 *HTTP metode* **[53]**.

Metoda	Značenje
GET	Dohvaćanje podataka
POST	Dodavanje podataka
PUT	Promjena podataka
DELETE	Brisanje podataka

Klase za dohvaćanje podatka korištenih u radu su *URLRequest* i *URLSession*, dok je za pretvorbu iz *JSON* u *Foundation* format, te obratno korištena *JSONSerialisation* klasa.

4.1.6 Osvježavanje aplikacije

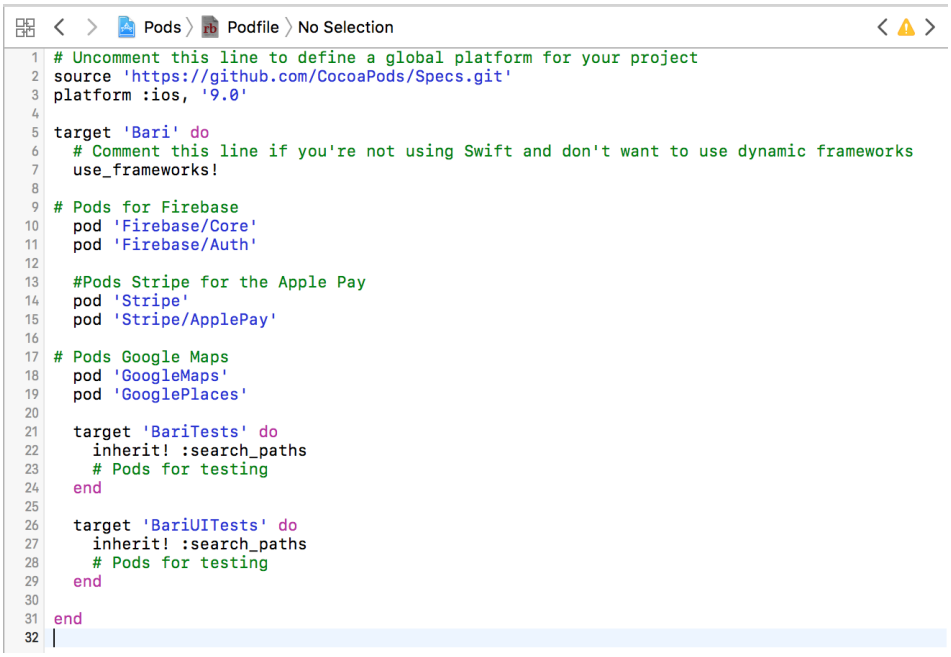
Budući da se radi s dinamičkim podacima unutar programskog rješenja, potrebno je te podatke pravovaljano ažurirati. Način ažuriranja podataka koji se pojavljuju na određenom zaslonu aplikacije odvija se najčešće na sporednoj niti, dok se osvježavanje zaslona uvijek odvija na glavnoj. Osvježavanje implementirano u diplomskom radu izvršava se prilikom novog pokretanja aplikacije ili odabiranja određenog zaslona. U slučaju potrebe traženja novih

podataka, aplikacija se osvježava u trenutku odabira parametara unutar tražilice te prikazuje najsvježije informacije. Osim programskog osvježavanja, aplikacija se može osvježiti i direktnom interakcijom korisnika po standardnom *iOS* načelu, ako korisnik klizne prstom od vrha zaslona prema dolje.

4.1.7 Pods integracija

Cocoapods [54] je upravitelj ovisnostima (engl. *dependency manager*) o drugim bibliotekama korištenim unutar *Xcode* razvojnog okruženja. Koristi se pri integriranju biblioteka ili radnog okruženja (engl. *framework*) unutar projekta bez razmišljanja o načinu na koji se okruženje treba postaviti.

Kako bi se *Xcode* okruženje pripremilo za korištenje vanjskog kôda, umjesto kreiranja novog *Xcode* projekta, treba se kreirati *Xcode Workspace* koji omogućuje korištenje *Cocoapods* biblioteka i povezivanje među projektima ako se nalaze u istom *Workspace* okruženju. Primjer kôda korištenja *Cocoapods*a vidljiv je na slici 4.4.



Sl. 4.4 *Cocoapods file unutar Xcode okruženja.*

4.1.8 Firebase integracija

*Google*ovo okruženje za upravljanje i pohranu podataka na oblaku računala korišteno je radi sigurnosnog prijavljivanja u aplikaciju i upravljanja korisničkim podacima.

Spajanje usluga s *iOS* platformom izvršava se preko *Firebase* upravljačke ploče [55], gdje je kreiran jedinstveni račun i novi projekt aplikacije. Nakon kreiranja aplikacije potrebno je unijeti

jedinstveni *Bunde ID* preko kojeg će aplikacija biti ovjerena te će biti omogućen pristup konfiguracijskom "*GoogleService-Info.plist*" dokumentu bez kojeg se ne mogu koristiti *Google*ove biblioteke. Nakon preuzimanja *.plist* dokumenta i njegove integracije u *Xcode* projekt, dodatno je potrebno instalirati *Firebase pod* dokument i omogućiti dijeljenu *Firebase* aplikacijsku instancu unutar konfiguracijske *AppDelegate.swift* klase.

4.1.9 Integracija Google Mapa

Integracija *Google* Mapa implementirana je na sličan način kao i integracija *Firebase* okruženja, samo se kreiranje i odobravanje aplikacije odvijalo unutar *Google API* okruženja. Nakon kreirane aplikacije, potrebno je dodati *API* ključ u *iOS* aplikaciju, te uvesti *Google* Mape i *Google* Mjesta u projekt te u klase unutar kojih se opcija koristi.

4.2 Korisničko sučelje mobilne aplikacije

Aplikacija za *iOS* uređaje razvijena u diplomskom radu odgovara prikazu na više dimenzija zaslona *iPhone* uređaja. Implementiranje sučelja osjetljivog na promjene veličine zaslona znatno je olakšano korištenjem opcije automatskog rasporeda (engl. *Auto Layout* [53]) i opcije ograničavanja proširenja objekata (engl. *constraint*) unutar *Xcode* okruženja. Korištenjem tih mogućnosti razvojni inženjer ne mora razvijati korisničko sučelje za svaki zaslon zasebno nego napravi jednu klasu korisničkog sučelja na kojoj postavi opcije automatskog rasporeda i ograničavanja te one samostalno prilagođavaju raspored i veličine ovisno na kojem zaslonu se pojave. Kreiranje korisničkog sučelja implementirano je putem *Storyboard* rješenja za kreiranje vizualnih objekata. Rotacija zaslona diplomskog rada je isključena, što znači kako je samo odobren uspravni način prikaza aplikacije (engl. *portrait mode*). Podržani uređaji su *iPhone 5, 5s, 6, 6s, 6s Plus, 7 i 7 Plus*.

4.2.1 Razvojni vodič korisničkog sučelja

Dio korisničkog sučelja mobilne aplikacije unutar kojeg je implementirana *Apple Pay* usluga ima dodatne kriterije strogoće od strane *Apple*ovog tima. Budući da grafička implementacija *Apple Pay* usluge ne zahtijeva previše iskustva grafičkog dizajna, a i kako bi se ta implementacija olakšala *Apple* je postavio implementacijske standarde [56] koji omogućuju korisnicima jednostavnije i lakše korištenje na svim uređajima. Svi *Apple Pay* gumbi moraju biti implementirani na isti način kako bi ugođaj kupovine bio bolji i jednostavnost pronalaženja gumba lakša.

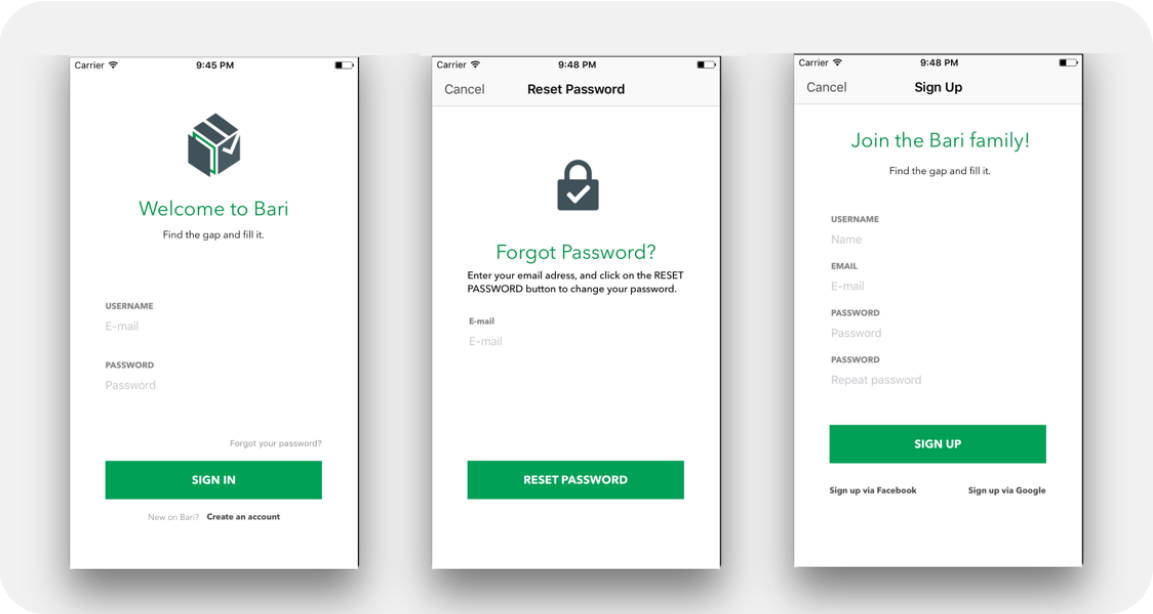
Prilikom izrade diplomskog rada korištena su 3 oblika *Apple Pay* gumba. Gumb u slučaju kada korisnik stanuje u regiji unutar koje nije odobreno korištenje sustava (bijeli gumb s crnim

obrubom), crni gumb s tekstom ”*Set up ・ Pay*” u slučaju kada se korisnik nalazi u odobrenoj regiji, ali još nije postavio svoju karticu, te gumb ”*Buy with ・ Pay*” kada su i korisnik i uređaj zadovoljili sve uvjete za korištenje *Apple Pay* okruženja. Prikaz 3 različita gumba mogu se vidjeti na slici 4.5.

Sl. 4.5 *Apple Pay gumbi.*

4.2.2 Scenariji korištenja

Slika 4.6. prikazuje 3 sučelja koja su korisniku dostupna prilikom pokretanja aplikacije. Prvo sučelje s lijeva predstavlja obrazac za prijavu u sustav. Osim prijave u sustav vidljive su poveznice do sučelja za kreiranja novog računa (prvi zaslon s desne strane) i sučelja zaboravljene lozinke. Unosom korisničkih podataka aplikacija provjerava njihovu ispravnost (format elektronične pošte) i jesu li podaci uopće uneseni. Ako su podaci uneseni ispravno, korisnik se preusmjerava na prvo sučelje sa slike 4.7, a ako su podaci neispravni, aplikacija prikazuje poruku pogreške u obliku obavijesti (engl. *Notification*) upozorenja objekta klase *AlertView* unutar aplikacije.

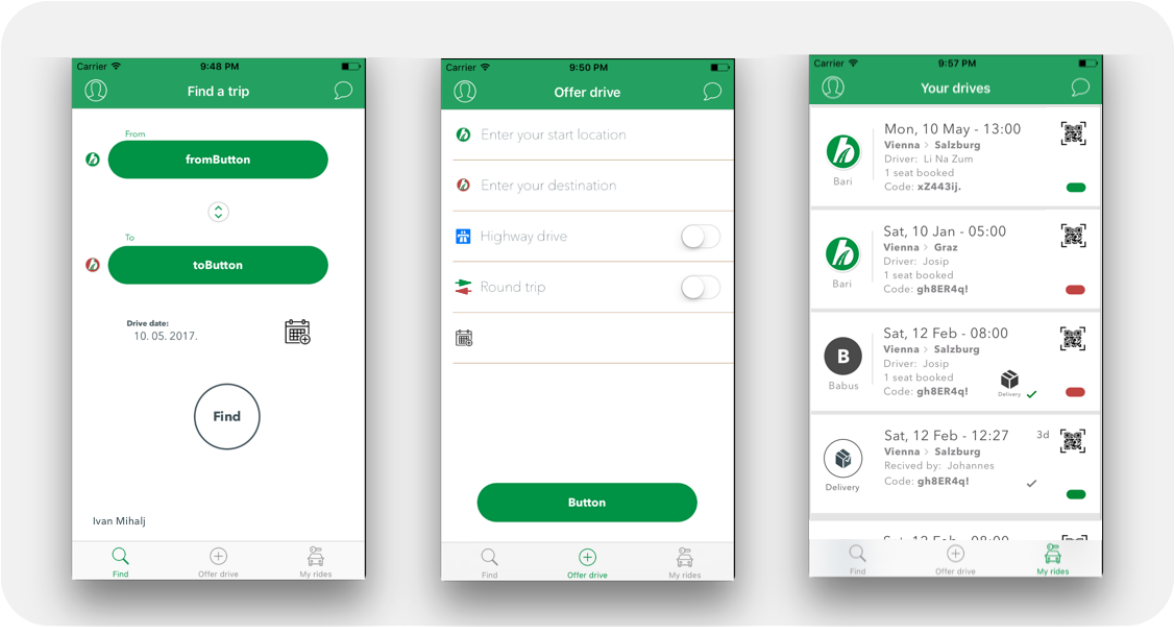


Sl. 4.6 *Zaslon prijave, poništavanja lozinke i registracije u sustav.*

Korisnik koji je zaboravio svoju lozinku, može na jednostavan način pritisnuti poveznicu "*Forgot your password?"* i tako se preusmjeriti na zaslon za ponovno postavljanje lozinke. Sve što je potrebno je unijeti adresu elektroničke pošte i poveznica za poništavanje lozinke bit će poslana na tu poštu.

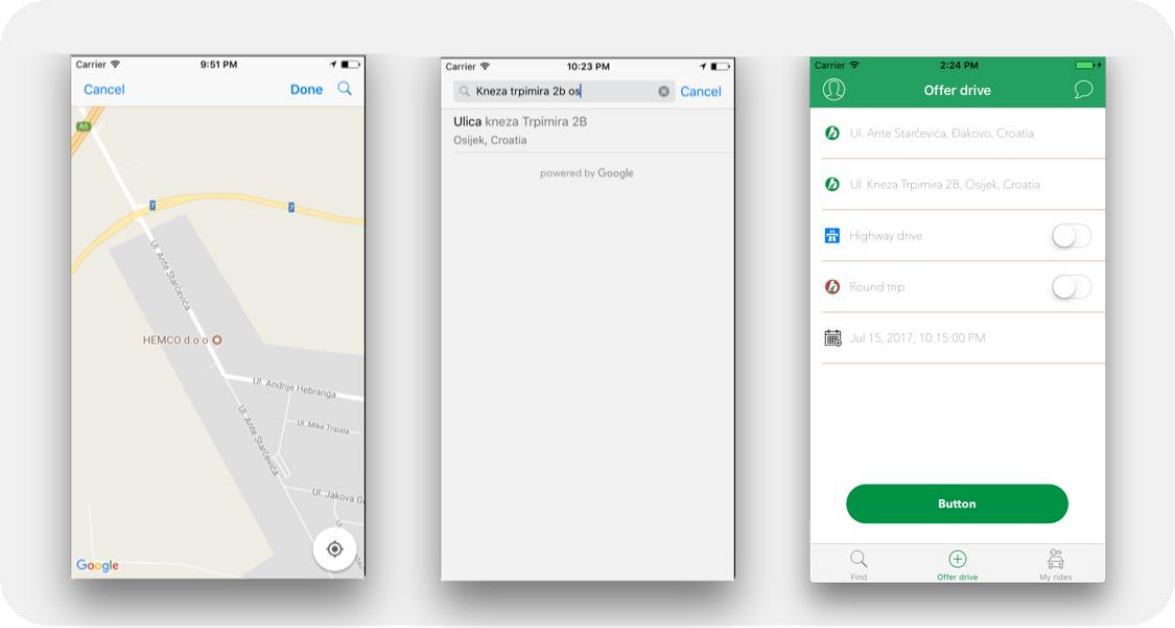
Zadnje sučelje sa slike 4.6 omogućuje svakom novom korisniku jednostavno kreiranje računa i pristupanje aplikaciji.

Nakon kreiranja računa i prijave u sustav, korisniku je putem izbornika s odjeljcima *UITabBarController*[54] omogućen pristup k 3 glavna sučelja koji predstavljaju osnovne značajke aplikacije. Prvi zaslon s lijeve strane omogućuje korisniku potragu za vožnjom ili dostavom u određeno vrijeme, na određenom mjestu, dok treći prikazuje dosadašnje korisnikove vožnje. Drugi zaslon prikazuje način na koji korisnik može objaviti vožnju koja će kasnije biti prikazana pretragom putem prvog zaslona. Na slici 4.7 prikazan je način na koji korisnik može objaviti ili pretraživati vožnju, a slika 4.8 detaljnije prikazuje kako ta pretraga radi.



Sl. 4.7 *Glavni zasloni aplikacije.*

Putem *Google* Mapa, *Google* Mjesta, te automatske dopune traženog sadržaja (engl. *autocomplete*) korisniku je omogućeno upisivanje mjesta polazišta i mjesta odredišta te kao što je vidljivo na zadnjoj slici s desna, prikaz rezultata unutar tekstualnog polja. Osim unosa gradova, korisnik mora unijeti i vrijeme polaska, a i vrijeme povratka u slučaju da putuje u oba smjera.



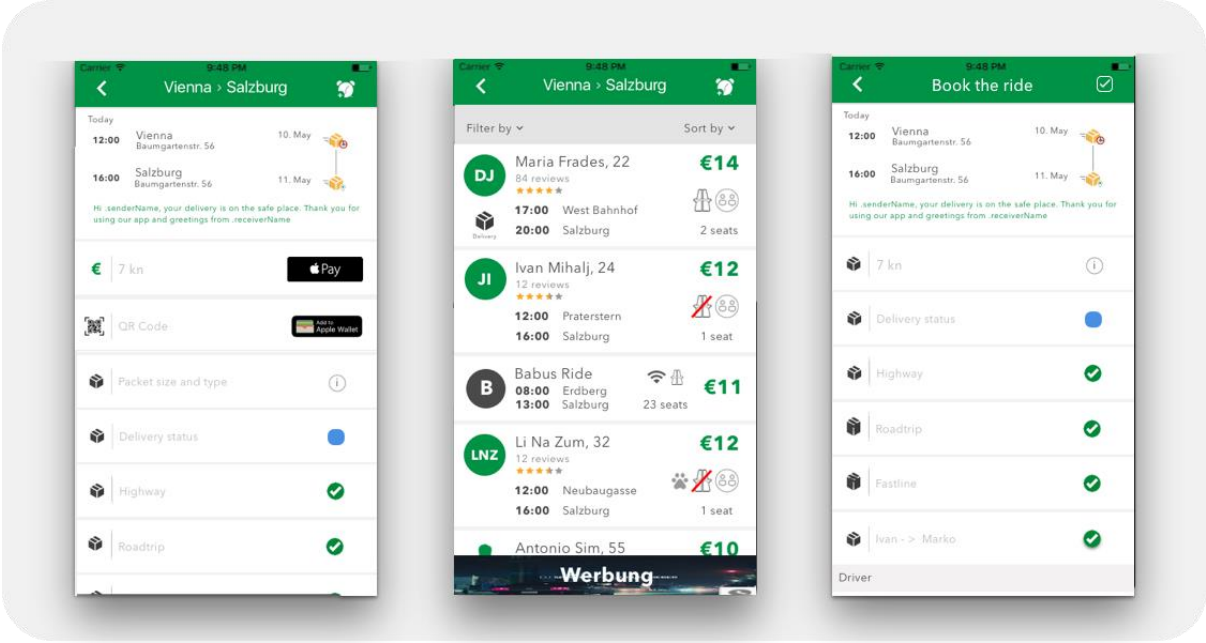
Sl. 4.8 *Zaslon odabira mjesta pomoću Google Mapa i Google Mjesta.*

Na sljedećoj slici (Sl. 4.9.) prikazani su detaljni zasloni redom: detaljan zaslon svake pojedine vožnje gdje su prisutni svi najvažniji podaci o jednoj ruti, detaljan zaslon korisničke pretrage

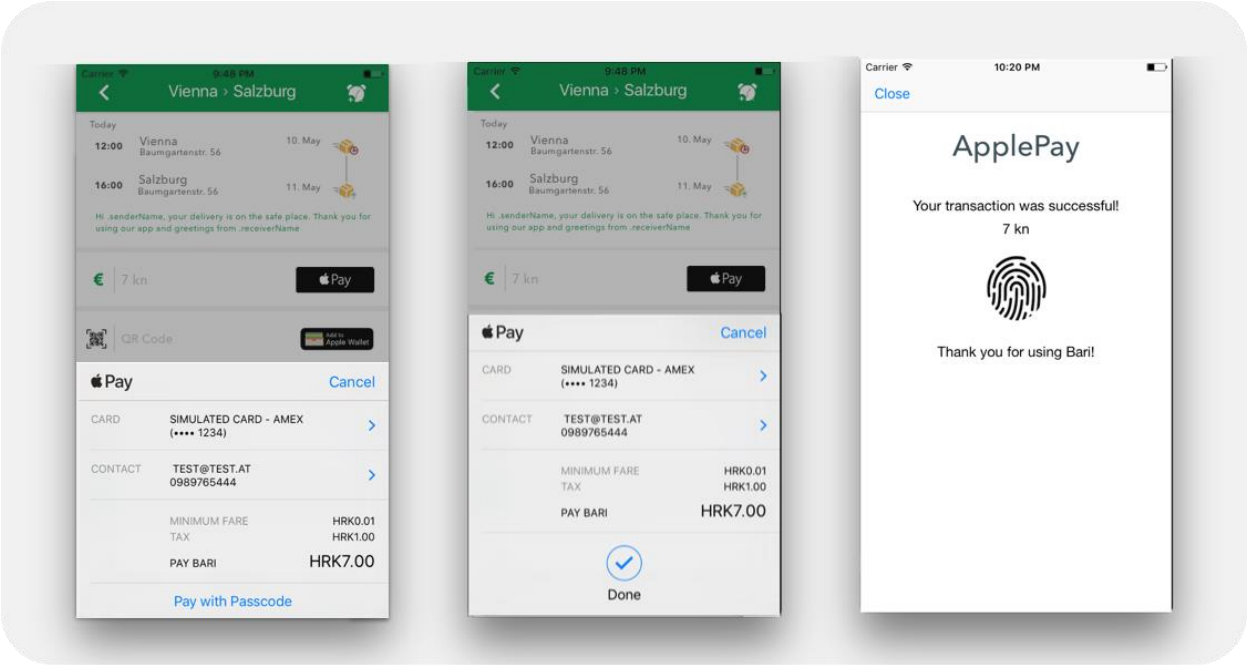
gdje korisnik može vidjeti sve vozače koji idu u njegovom smjeru te detaljan zaslon korisničke pretrage gdje su vidljivi dodatni podaci o pojedinoj vožnji i prikazana mogućnost rezerviranja vožnje.

Slika 4.10 prikazuje proces plaćanja unutar detaljnog zaslona jedne vožnje. Korisnik prvo pritisne gumb ”*Apple Pay*” nakon čega mu se prikazuje dodatni dio zaslona s detaljima transakcije. Pritiskom na ”*Pay with Passcode*” korisniku se prikazuje mogućnost plaćanja putem biometrijskog senzora otiska prsta. Nakon prislanjanja prsta biometrijskom čitaču, aplikacija je zabilježila uspješno obavljenu transakciju i pokazala transakcijske detalje na dodatnom zaslonu.

Za slučajeve kada *Apple Pay* nije postavljen, *iOS* aplikacija preusmjeri korisnika na postavljanje računa. A ako korisnik nema mogućnost korištenja usluge, onda se prikaže poruka s greškom (”Nalazite se u regiji unutar koje nije moguće koristiti Apple Pay uslugu.”).

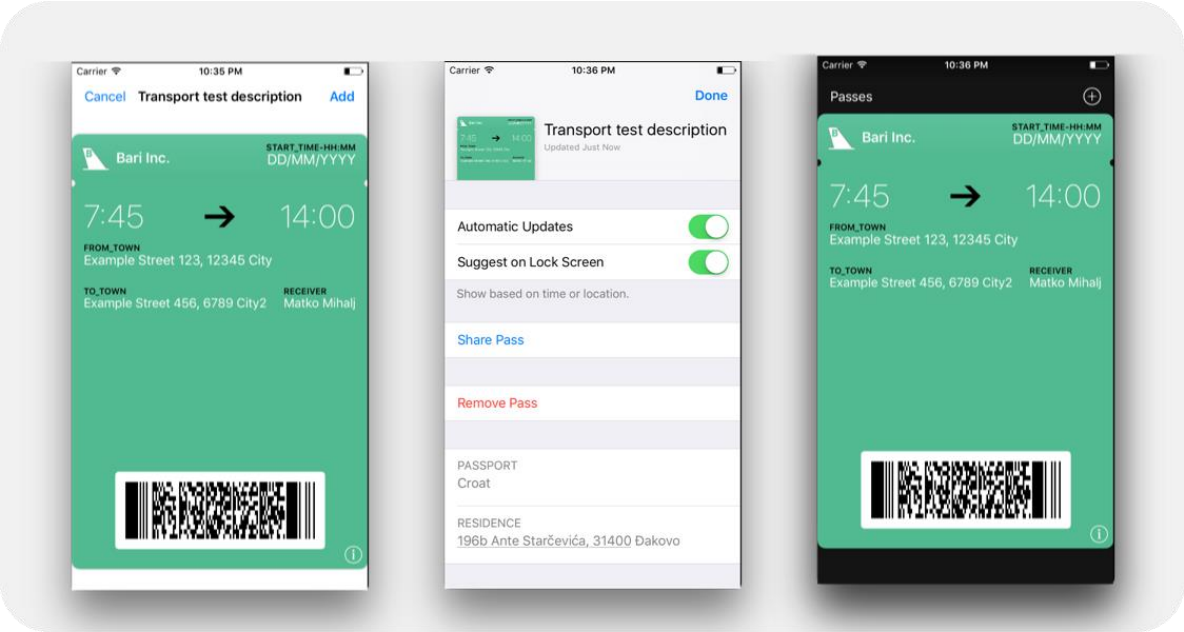


Sl. 4.9 *Detalj zasloni.*



Sl. 4.10 *Zasloni uspješne Apple Pay transakcije.*

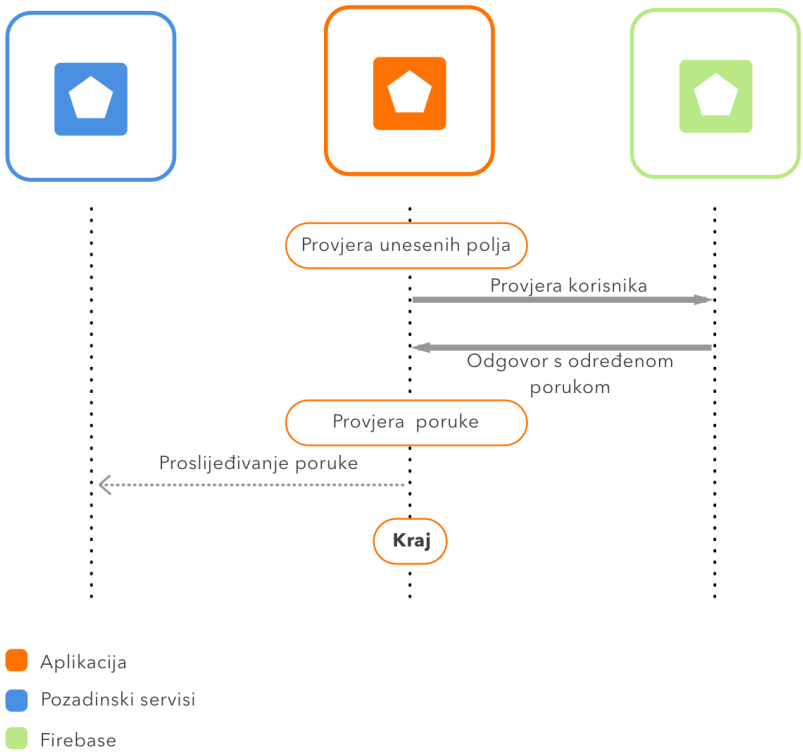
Osim plaćanja unutar aplikacije korisnik može koristiti opciju dodavanja vožnje putem kupon usluge unutar *Apple Wallet*a. Ta opcija omogućuje korisniku brži i jednostavniji pristup svojoj vožnji i upravljanje istom. Potrebno je samo pritisnuti na gumb ”*Add to Apple Wallet*” i otvorit će se dodatni zaslon s pitanjem o dodavanju kupona unutar *Apple Wallet* okruženja. Nakon pritiska na dodaj, kupon se integrira u novo okruženje sa svim podacima koji su dohvaćeni iz jedinstvenog *JSON* dokumenta o vožnji. Kreiranje kupona prikazano je na slici 4.11.



Sl. 4.11 *Zasloni kreiranja i dodavanja kupona u Apple Wallet.*

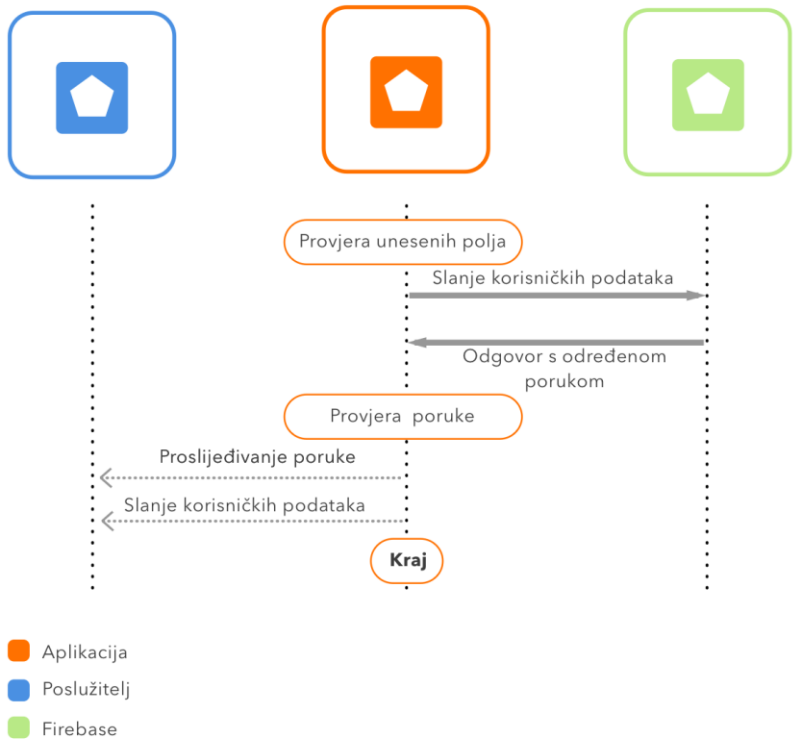
4.3 Dijagrami komunikacije aplikacije s udaljenim uslugama

Komunikacija prilikom prijave korisnika u sustav prikazana je slikom 4.12, slikom 4.13 prikazana je registracija korisnika. Slika 4.14 prikazuje način promjene lozinke poslane na klijent elektroničke pošte unutar *iOS* uređaja i zadnja slika (Sl. 4.15) prikazuje jedan proces slanja i primanja podataka s i na poslužitelj.



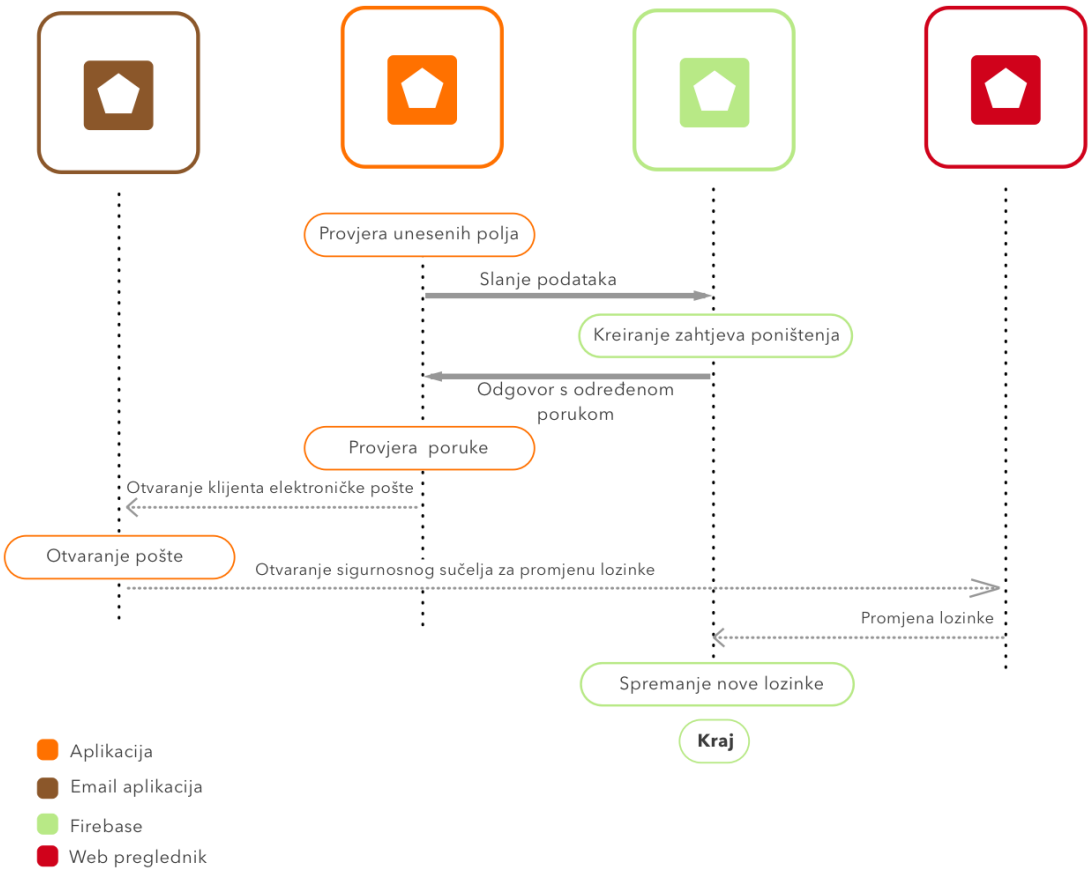
Sl. 4.12 *Prijava korisnika u sustav.*

Svaki korisnik koji nema kreiran račun za korištenje aplikacije može isti napraviti na zaslonu za registraciju. Proces registracije vidljiv je na slici 4.13. a odvija se unošenjem osobnih podataka, nakon čega se ti podaci provjeravaju na *Firebase* poslužitelju. *Firebase* šalje odgovor putem poruke valjanosti te aplikacija prikazuje poruku i kreira novi korisnički račun. Nakon kreiranja računa novonastali korisnički podaci šalju se i do internih poslužitelja aplikacije.



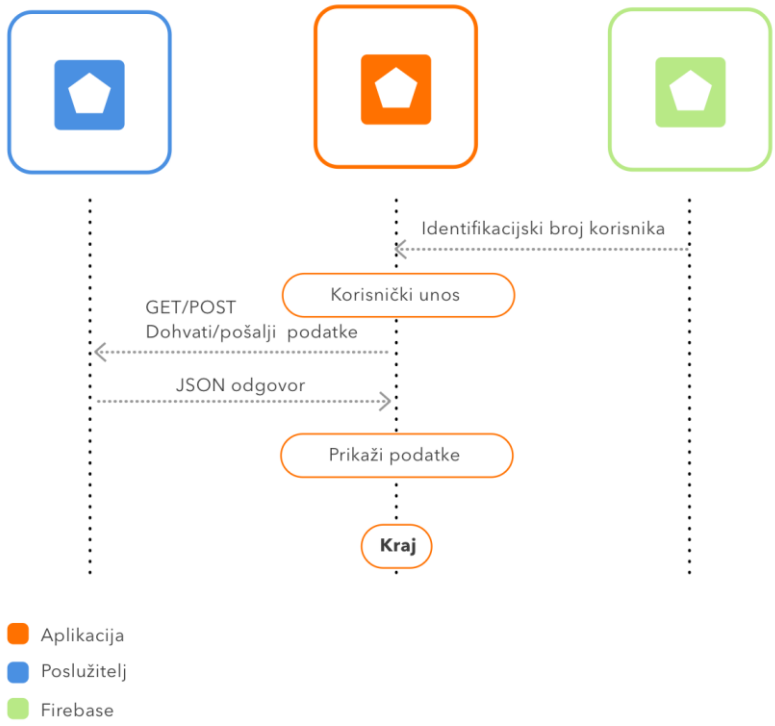
Sl. 4.13 *Registracija korisnika u sustav.*

Promjena korisničke lozinke odvija se interakcijom aplikacije razvijene diplomskim radom, *Firebase* okruženja, web preglednika i udaljenog poslužitelja aplikacije. Tijek poništavanja lozinke prikazan je slikom 4.14.



Sl. 4.14 *Promjena lozinke.*

Komunikacija između klijent aplikacije i udaljenih poslužitelja prikazana je slikom 4.15.



Sl. 4.15 *Komunikacija s poslužiteljem u svrhu dohvaćanja i slanja podataka.*

5 PROGRAMSKO RJEŠENJE I ANALIZA IMPLEMENTACIJE APPLE PAY SUSTAVA

Nakon uspješno kreiranih i uvezenih certifikata nabrojanih u poglavlju 3.1.1 i 3.1.8 te odobravanja *Apple Pay* i *Apple Wallet* usluge unutar *Xcode* razvojnog okruženja, okruženje za razvoj beskontaktnog mobilnog plaćanja je konfigurirano i spremno za razvoj aplikacije.

5.1 Programsko rješenje

5.1.1 Apple Pay gumb

Prvi sljedeći korak je provjera dostupnosti usluge na korisnikovom pametnom uređaju te nakon toga provjera valjanosti unesene kartice, ako ju korisnik ima. Kratki dio kôda koji predstavlja tu provjeru vidljiv je na slici 5.1.

```
class func applePayStatus() -> (canMakePayments: Bool, canSetupCards: Bool) {
    return ( PKPaymentAuthorizationController.canMakePayments(),
             PKPaymentAuthorizationController.canMakePayments(usingNetworks:
                 supportedNetworks));
}
```

Sl. 5.1 *Provjera dostupnosti Apple Pay usluge i provjera valjanosti kartice.*

Slika 5.2 prikazuje podržane davatelje usluga koji moraju biti direktno implementirani unutar aplikacije i njihova izmjena nije moguća direktnom promjenom na udaljenom poslužitelju.

```
static let supportedNetworks: [PKPaymentNetwork] =
    [
    .amex,
    .discover,
    .masterCard,
    .visa
]
```

Sl. 5.2 *Podržani davatelji usluga.*

Nakon uspješne provjere svih podataka potrebnih za nastavak transakcije, programski kôd kreira platni zahtjev u obliku *PKPaymentRequest* objekta. *PKPaymentRequest* objekt predstavlja iznos i podatke koji se potražuju od korisnika prilikom korištenja usluge. Na slici 5.3 vidljivo je koji su detalji korišteni.

PaymentSummaryItems objekt služi za izračunavanje krajnje cijene usluge ili proizvoda iz svih podataka od osnovne cijene, poreza na cijenu pa do popusta ako ga ima. *MerchantCapabilities* služi za odobravanje 3D sigurnosnog protokola[55]. Obavezna polja prilikom unosa osobnih podataka implementirana su pomoću *RequiredShippingAdressFields objekta.* Objekt koji prikazuje koje vrste kartica su podržane unutar aplikacije je *supportedNetworks*.

```
let paymentRequest = PKPaymentRequest()

paymentRequest.paymentSummaryItems = paymentSummaryItems
paymentRequest.merchantIdentifier = [redacted]
paymentRequest.merchantCapabilities = .capability3DS
paymentRequest.countryCode = "HR"
paymentRequest.currencyCode = "HRK"
paymentRequest.requiredShippingAddressFields = [.phone, .email]
paymentRequest.supportedNetworks = PaymentHandler.supportedNetworks
```

Sl. 5.3 *Obavezni detalji platne transakcije.*

Kreiranje *paymentSummaryItems* niza i pregled vrijednosti koje se prikazuju kupcu prilikom kreiranja nove transakcije prikazane su dijelom kôda sa slike 5.4. Sustav omogućuje korištenje negativne vrijednosti, ali pod uvjetom da rezultat mora biti veći od nula.

```
let fare = PKPaymentSummaryItem(label: "Minimum Fare", amount: NSDecimalNumber(string:
    minimumFare), type: .final)
let tax = PKPaymentSummaryItem(label: "Tax", amount: NSDecimalNumber(string: taxFare),
    type: .final)
let discount = PKPaymentSummaryItem(label:
    "Discount", amount: NSDecimalNumber(string: discountAmount), type: .final)
let total = PKPaymentSummaryItem(label: "Bari", amount: NSDecimalNumber(string: totalFare),
    type: .final)

paymentSummaryItems = [fare, tax, discount, total]
```

Sl. 5.4 *Iznos kojeg korisnik plaća.*

Korištenje *Apple Pay* usluge omogućeno je od *iOS*a 10, a prikaz dodatnog korisničkog sučelja s detaljima *Apple Pay* transakcije implementirano je preko kôda sa slike 5.5.

```
paymentController = PKPaymentAuthorizationController(paymentRequest: paymentRequest)
paymentController?.delegate = self
paymentController?.present(completion: { (presented: Bool) in
    if presented {
        NSLog("Presented payment controller")
    } else {
        NSLog("Failed to present payment controller")
        self.completionHandler!(false)
    }
})
```

Sl. 5.5 *Prikaz Apple Pay zaslona.*

Nakon provjere korisnika i odobravanja transakcije biometrijskim putem, delegat transakcije je pozvan s transakcijskim tokenom koji je korišten za odobravanje transakcije.

Tijek jedne *Apple Pay* transakcije:

- Kreiran je *PKPaymentAuthorizationController*
- Korisnik odobrava transakciju putem biometrijskog senzora, ili izravno putem lozinke

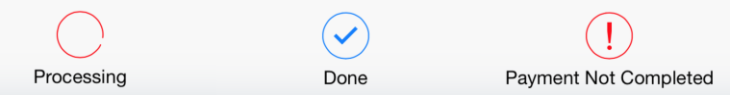
- Ćelija (engl. *cell*) ”*Pay with Passcode*” sa slike 4.10 se transformira u indikator učitavanja (engl. *loading spinner*) s tekstom ”*Processing*”
- Delegat prima potvrdu plaćanja sa slike 5.6 i odvija se asinkrona komunikacija s poslužiteljem i transakcijskim procesorom u svrhu naplaćivanja detalja s primjera kôda na slici 5.4. Nakon izvršavanja transakcije, poziva se *completion* parametar koji prikazuje uspješnost transakcije
- Uspješnost transakcije prikazuje se putem indikatora sa slike 5.8, ovisno je li transakcija obavljena uspješno ili je transakcija neuspješna
- Nakon uspješne transakcije njen prikaz je vidljiv u *Apple Wallet* aplikaciji i putem *Stripe* okruženja za testiranje transakcija
- Delegat prima informacije o izvršenosti transakcije putem *didAuthorizePayment* (Sl. 5.7) i poziva *controller.dismiss* za opozivanje zaslona za *Apple Pay* transakciju

```
@available(iOS 10.0, *)
extension PaymentHandler: PKPaymentAuthorizationControllerDelegate {
    func paymentAuthorizationController(_ controller: PKPaymentAuthorizationController,
        didAuthorizePayment payment: PKPayment, completion: @escaping
        (PKPaymentAuthorizationStatus) -> Void) {


        if payment.shippingContact?.emailAddress == nil || payment.shippingContact?.phoneNumber ==
            nil {
            paymentStatus = .invalidShippingContact
        } else {

            paymentStatus = .success

        }

        completion(paymentStatus)
    }
```

Sl. 5.6 *Potvrda plaćanja.*

```
func paymentAuthorizationControllerDidFinish(_ controller: PKPaymentAuthorizationController) {
    controller.dismiss {
        DispatchQueue.main.async {
            if self.paymentStatus == .success {
                self.completionHandler!(true)
            } else {
                self.completionHandler!(false)
            }
        }
    }

}
```

Sl. 5.7 *Završetak transakcije.*



Sl. 5.8 *Odaziv transakcije*

Kreiranje *Apple Pay* gumba ne odvija se putem korisničkog sučelja nego se izravno kreira unutar kôda. Gumb ima više načina prikaza, a oni korišteni u diplomskom radu prikazani na slici 4.5 kreiraju se putem kôda sa slike 5.9. U slučaju kada korisnik ima mogućnost korištenja usluge, prilikom pritiska na gumb poziva se kreiranje crnog kupi gumba putem funkcije *touchUpInside*. Kada korisnik ima mogućnost korištenja usluge, ali nije kreirao karticu unutar *Apple Wallet* okruženja, tada se korisniku prikaže crni gumb s logotipom. Ako korisnik pritisne na takav gumb bit će preusmjeren na konfiguraciju postavljanja nove kartice. U slučaju kada se korisnik nalazi u regiji koja još uvijek nije u partnerskim državama za uslugu, gumb će biti bijele boje i pritiskom na njega će biti prikazana poruka pogreške.

```
let result = PaymentHandler.applePayStatus()

var applePayButton: UIButton?

if result.canMakePayments {
    applePayButton = PKPaymentButton(type: .buy, style: .black)
    applePayButton?.addTarget(self, action: #selector(ShowMyRideDetailsViewController.
        applePayButtonPressed), for: .touchUpInside)
} else if result.canSetupCards {
    applePayButton = PKPaymentButton(type: .setUp, style: .black)
    applePayButton?.addTarget(self, action: #selector(ShowMyRideDetailsViewController.
        setupPressed), for: .touchUpInside)
} else {
    applePayButton = PKPaymentButton(type: .plain, style: .whiteOutline)
    applePayButton?.addTarget(self, action: #selector(ShowMyRideDetailsViewController.
        applePayNotAllowed), for: .touchUpInside)
}

if applePayButton != nil {
    applePayButton!.autoresizingMask = [.flexibleLeftMargin, .flexibleRightMargin]

    cell2.buyWithApplePay.addSubview(applePayButton!)

}
```

Sl. 5.9 *Kreiranje i pozivanje funkcije putem gumba.*

Kreiranje gumba i popunjavanje podataka nakon pritiska na isti odvija se pomoću 3 različite klase. Klase ”*ShowMyRidesAmountTableViewCell.swift”* koja predstavlja model ćelije unutar koje se nalazi gumb, ”*ShowMyRideDetailsViewController.swift*” klasa koja poziva kreiranje dinamičke ćelije, pozicionira gumb te dohvaća podatke iz ”*ShowMyRidesAmountTableViewCell*” modela i iz *PaymentHandler.swif* klase unutar koje se nalaze svi važni podaci o usluzi te pomoću detalja o dostupnosti i statusu *Apple Pay* usluge kreira određeni gumb prema slici 4.5. Dohvaćanje cijene s poslužitelja vrši se putem *JSON* dokumenta i *HTTP* zahtjeva koji dohvaća cijelu *JSON* datoteku (*URLRequest* i *URLSession*), pretvara ju u oblik razumljiv razvoju *iOS* aplikacija (*JSONSerialisation,* pretvorba iz JSON u Foundation format*)* te ga šalje u upravljačku klasu i izmjenjuje transakcijske podatke sa slike 5.4 za svaku vožnju posebno. Osim korištenja

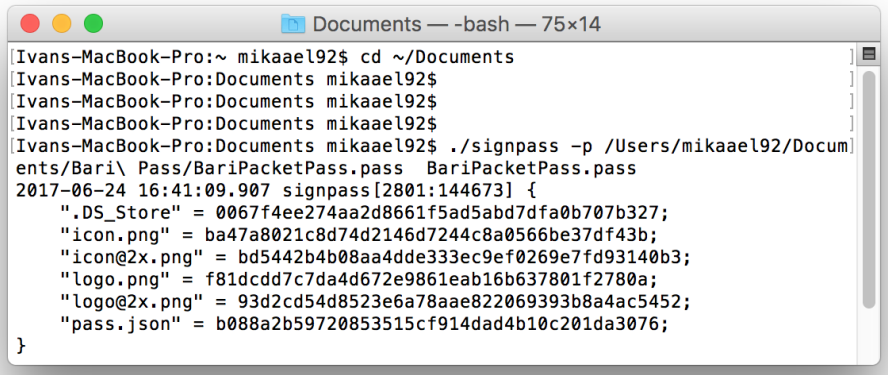
osnovnih iOS biblioteka prilikom komunikacije sa poslužiteljem, za razvoj diplomskog rada korištena je i *Alamofire* [57] biblioteka.

5.1.2 Kreiranje kupona i dodavanje u Apple Wallet

Kuponi unutar *iOS* okruženja kreiraju se kao paketi koji sadrže *pass.json* dokument unutar kojeg će biti pohranjene sve informacije potrebne za kreiranje kupona.

Da bi razvojni inženjer mogao koristiti kupon potrebno je kreirati novu datoteku u već postojećoj projektnoj datoteci. Nakon kreiranja datoteke potrebno je preuzeti konfiguraciju s *Apple*ovog poslužitelja [58], otvoriti ju unutar novog *Xcode* projekta i pomoću nje kreirati *signpass* dokument pomoću kojeg će se kasnije kreirati kupon. *Signpass* dokument potrebno je prebaciti u već prije kreiranu datoteku unutar projektnog direktorija te izmijeniti sadržaj *JSON* dokumenta i pohraniti vlastite podatke. Za prikaz željenih, prilagođenih (engl. *custom*) podataka unutar kupona još je potrebno unutar istog *JSON* dokumenta (Sl. 5.11) podesiti identifikaciju tima koja se nalazi u detaljima *merchant* certifikata unutar *Keychain Access* okruženja na *Mac* računalu te identifikaciju kupona koja se nalazi unutar *Xcode* okruženja prikazanog slikom 3.3.

Kada su svi podaci valjano popunjeni, potrebno je preko komandne ploče pristupiti kreiranoj datoteci i kreirati novi kupon (Sl. 5.10). Nakon uspješno završenog procesa kreiranja kupona, potrebno je kupon ugraditi u Xcode projekt.



Sl. 5.10 *Kreiranje kupona putem komandne ploče*

```
{
    ...
    "passTypeIdentifier" : "
your pass type identifier
",
    "teamIdentifier" : "
your Team ID
",
    ...
}
```

Sl. 5.11 *Postavljanje identifikacije kupona i tima.*

Kreirani kupon nalazi se unutar *Xcode* okruženja i omogućeno je upravljanje i korištenje istog. Kôd sa slike 5.12 prikazuje kreiranje korisničkog sučelja za dodavanje kupona (Sl. 4.11) u *Apple Wallet* pomoću imena ”*BariPacketPass*” u slučaju da je *PKPass* biblioteka dostupna.

```
func addToAppleWalletButtonTapped(at index: IndexPath) {
    print("button tapped at index:\(index)")
    if PKPassLibrary.isPassLibraryAvailable() {
        let passData = NSData(contentsOfFile: Bundle.main.path(forResource: "BariPacketPass",
            ofType: "pkpass")!)!
        var error: NSError?
        let pass = PKPass(data: passData as Data, error: &error)

        if error == nil {

            let vc = PKAddPassesViewController(pass: pass)
            self.present(vc, animated: true, completion: nil)

        }
    }

}
```

Sl. 5.12 *Kreiranje kupona.*

Budući da se gumb za dodavanje kupona unutar *Apple Wallet* okruženja nalazi unutar ćelije u tabličnom prikazu (engl. *Table view*), za kreiranje gumba korištene su 2 različite klase. Jedna unutar koje su model gumba i vizualni objekti unutar kojih će biti prikazani podaci i druga unutar koje se prikupljaju podaci s poslužitelja za određenu ćeliju te funkcija koja kreira novi zaslon za dodavanje kupona u *Apple Walet* okruženje. Komunikacija između te 2 klase odvija se preko delegat metode (Sl. 5.13) koja preuzima funkciju iz jedne klase i izvršava ju unutar druge prilikom pritiska gumba (Sl. 5.14).

```
protocol TestAddToAppleWalletDelegate {
    func addToAppleWalletButtonTapped(at index: IndexPath)
}
```

Sl. 5.13 *Funkcija za komuniciranje između različitih upravljačkih objekata.*

```
@IBOutlet weak var addToAppleWalletButton: UIButton!
var indexPath: IndexPath!


@IBAction func addToAppleWalletButtonAction(_ sender: UIButton) {

    self.delegate.addToAppleWalletButtonTapped(at: indexPath)

}
```

Sl. 5.14 *Pozivanje funkcije iz druge klase i kreiranje gumba.*

5.2 Integracija sa Stripe-om

Osim kreiranja certifikata bilo je potrebno implementirati *Stripe SDK* unutar *Xcode* projekta, što je riješeno Pomoću *cocoa pods*a te je sljedeći korak bio dodavanje osobnog *Stripe API* ključa unutar *AppDelegate.swift* klase (Sl. 5.15).

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
    [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.

    //FIRApp.configure()
    FirebaseApp.configure()
    GMSServices.provideAPIKey("[redacted]")
    GMSPlacesClient.provideAPIKey("[redacted]")
    STPPaymentConfiguration.shared().publishableKey = "[redacted]"

    return true
}
```

Sl. 5.15 *Implementacija Stripe okruženja.*

Budući da je *Stripe* okruženje razvijeno u *Objective-C* jeziku bilo je potrebno kreirati *Bridging-Header* klasu i unutar nje uvesti *Stripe* i *Stripe Apple Pay* zaglavlje (engl. *header*). Kreiranje *Bridging-Header* klase prikazano je slikom 5.16.

```
//#ifndef Bari_Bridging_Header_h
//#define Bari_Bridging_Header_h

#import <Stripe/Stripe.h>
#import <Stripe/Stripe+ApplePay.h>

//#endif /* Bari_Bridging_Header_h */
```

Sl. 5.16 *Stripe zaglavlje*

5.3 Testiranje implementiranog sustava

5.3.1 Kreiranje Sandbox testnog računa

Stvaranje okruženja pogodnog za testiranje *Apple Pay* sustava zahtjeva prolaženje i stvaranje nekoliko sigurnosnih koraka raspisanih od strane *Apple*a. S Apple razvojnim certifikatom potrebno je kreirati testni *Sandbox* račun [59] s valjanom adresom elektroničke pošte na koju će biti poslana poveznica za ovjeru kreiranog testnog računa. Prilikom potvrde računa potrebno je dokazati kako je adresa korisničke pošte spojena s originalnim *Apple* razvojnim računom valjana kako bi se proces mogao dalje nastaviti. Budući da se u Hrvatskoj Apple Pay usluga ne može još službeno koristiti, potrebno je kreirati račun za neku od zemalja gdje je trenutno odobreno implementiranje platforme. Za potrebne ovog diplomskog rada kreiran je *iCloud* i *App Store* račun koji je namijenjen korištenju na američkom tržištu, što je moralo biti jasno navedeno prilikom kreiranja testnog *Sandbox* računa [60].

Nakon uspješno kreiranog testnog okruženja prema američkim standardima i prihvaćanju uvjeta i odredbi korištenja kako testnog sustava tako i same *Apple Pay* platforme, bilo je potrebno odjaviti se s osobnog *iCloud* računa na osobnom *iPhone* uređaju i prijaviti se na novi testni *Sandbox* račun. Sama prijava još uvijek ne omogućuje potrebno okruženje za testiranje sklopovlja i programske podrške nego je potrebno promijeniti regiju u kojoj se uređaj nalazi na regiju koja podržava navedene usluge (isto na američku).

Nakon postavljanja testnog računa, ovjere testera, sklopovlja, regije, odobreno je testiranje implementirane aplikacije i provjera valjanosti *Apple Pay* usluge i njene implementacije. Unutar *Apple Wallet* aplikacije, nakon uspješne registracije američkog računa i prebacivanjem na američku regiju, *Apple Pay* opcija se automatski dodaje i pritom omogućuje unošenja kartice. Treba paziti kako se ne bi unijeli podaci valjane karticu i na tako izravno naplaćivale testne transakcije s nje.

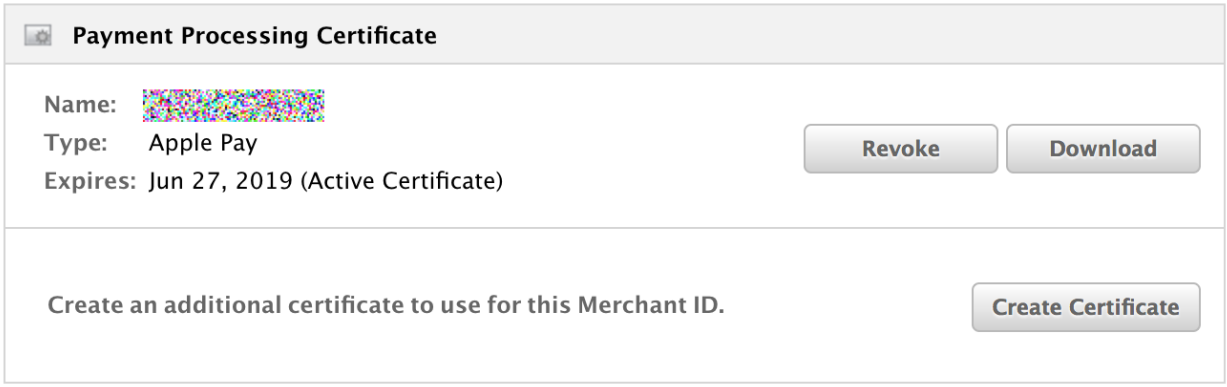
Klikom na *Apple Pay* otvara se sučelje za podešavanje platforme beskontaktnog ili plaćanja biometrijskim putem. Karticu je moguće dodati skeniranjem njenih podataka kroz *Apple*ov sustav za prepoznavanje podataka sa slike koji automatski popunjava podatke i olakšava korisniku onaj dio kada možda ne zna gdje koji podatak s kartice treba biti upisan, no budući da je korištena testna kartica i testni podaci dostupni na službenim *Apple*ovim stranicama taj korak nije bio potreban. *Wallet* aplikacija je preskakanjem skeniranja kartice omogućila ručno unošenje podataka. Bilo je potrebno unijeti ime vlasnika kartice, broj kartice (*PAN*), nakon toga je slijedila ovjera unesenih podataka (Sl. 2.8). Kartični podaci su ispravno uneseni i ovjereni te je slijedilo unošenje kartičnih detalja, datum valjanosti i sigurnosni *CVM* kôd. Sljedeći korak je provjera

unesenih podataka s kojom *Apple* nema ništa nego se direktno ovjerava i kreira od strane pružatelja kartičnih usluga kako bi se Apple ogradio od pohrane podataka na svojem uređaju i njihovim upravljanjem. Uspješnom ovjerom preko poslužitelja davatelja kartičnih usluga kreira se testna kartica i *Apple Pay* sučelje je spremno za korištenje unutar mobilne aplikacije.

Prilikom pokretanja aplikacije i izvršavanja testa, aplikacija je javljala grešku kako aplikacija nije valjana za korištenje *Apple Pay* usluge. Osim što je kreiran *iCloud* račun pa zatim testni *Sandbox* račun, testna kartica i prolazak kroz postupak ovjere, još uvijek nije bilo moguće testirati aplikaciju. Sljedeće što je bilo potrebno napraviti je ovjeriti certifikat od pružatelja transakcijskih usluga.

Za pružatelja je korišteno *Stripe* okruženje unutar kojeg je kreiran certifikat za *Apple Pay* usluge te se naknadno pomoću tog istog certifikata kreirao još jedan certifikat za izvršavanje transakcije unutar već kreiranog *merchant* certifikata.

Novokreirani *Apple* certifikat sa slike 5.17. potom je dodan u *Stripe* okruženje (*apple_pay.cer*) te je omogućeno kreiranje završnog testnog tajnog ključa (Sl. 5.18.) koji će biti implementiran unutar aplikacije i pomoću kojeg će se moći koristiti testne transakcije umjesto pravih koje bi se direktno naplaćivale s pravog računa.



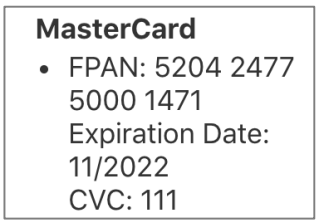
Sl. 5.17 *Certifikat za izvršavanje transakcije u Apple okruženju za certifikate.*

Sl. 5.18 *Potvrđeni certifikat u Stripe okruženju.*

Završna implementacija za kreiranje testnog okruženja je slanje autorizacijskog tokena na *Stripe* sustav i njegovo izvršavanje, a to se izvršava na sljedeći način:

- Slanje autorizacijskog tokena *PKPayment* na dešifriranje u *Stripeov* sustav, ako je uspješno, stripe vraća svoj jedinstveni token
- Slanje *Stripe* testnog tokena na vlastiti poslužitelj kako bi naš pozadinski sustav znao za narudžbu i izvršio svaki sljedeći potreban proces
- Slanje *Stripe* tokena nazad u *Stripe* sustav kako bi se izvršile transakcije

Testni kartični podaci korišteni u diplomskom radu su prikazani na slici 5.19.



Sl. 5.19 *Testni kartični podaci*

5.3.2 Testiranje sustava na Simulatoru

Testiranje *Apple Pay* transakcija unutar *Simulator*a omogućilo je samo provjeru korisničkog sučelja i korake koje aplikacije izvodi prilikom plaćanja. Plaćanje putem biometrijskog otiska prsta nije omogućeno, budući da se to ne može simulirati preko otiska te je svaka transakcija pritiskom na gumb ”*Pay with Passcode*” provedena i izvršena bez obzira na to što *Simulator* ima nedostatak *Touch ID*a. Iako prilikom plaćanja preko Simulatora aplikacija dohvaća transakcijski token potrebno je daljnje testiranje obaviti na fizičkom uređaju.

5.4 Sentry sustav za nadgledanje valjanosti programskog rješenja

Putem *Sentry* [61] programske podrške unutar aplikacije implementirana je još jedna dodatna razina kvalitete koja omogućuje pronalaženje grešaka u kôdu čim se one dogode, drugim riječima Sentry je sustav za provjeru rušenja aplikacije u realnom vremenu, omogućujući tako razvojnim inženjerima lakše pronalaženje greške i brzo reagiranje prilikom njenog uklanjanja.
Sentry sustav koristi za testiranje rušenja aplikacije u produkcijskom okruženju, što znači da se aplikacija prvo treba kreirati i tek onda će doći obavijesti o greškama unutar *Sentry* sustava i direktno na elektroničku poštu razvojnih inženjera.

5.5 Daljnji razvoj aplikacije

Aplikacija za transport i plaćanje transportnih usluga je aplikacija koja može promijeniti način na koji njeni korisnici zarađuju i provode vrijeme u svojim vozilima i na prometnicama. . Daljnji

razvoj ove aplikacije bit će orijentiran prema personaliziranom razvoju za tvrtke i omogućit će nove mogućnosti unutar trenutne inačice aplikacije. Moguća daljnja nadogradnja u vidu obavijesti (engl. *push notifications*), kreiranja automatske pohrane podataka i korištenja tekstualnih kratica za dohvaćanje podataka s poslužitelja, moguće je dodati vlastiti način prijave u sustav i nadograditi sustav s mogućnošću plaćanja putem web sučelja, budući da se planira napraviti i web aplikacija. Paralelno s *iOS* aplikacijom moguće je razviti *Android* aplikaciju unutar koje će biti omogućeno plaćanje putem *Android* operacijskog sustava.

Aplikacija će se objaviti na trgovinama aplikacijama krajem godine i bit će pokriveno 5 glavnih već ispitanih tržišta te će se aplikacijom pokušati pokrit i humanitarni karakter.

6 ZAKLJUČAK

U diplomskom radu osmišljen je i programski realiziran sustav za beskontaktni način plaćanja putem *iOS* aplikacije. Kako je cilj bio razviti *iOS* aplikaciju pomoću koje će korisnici moći dijeliti slobodna mjesta i slobodan prostor te ujedno na sigurniji način plaćati unutar aplikacije za potrebe rada vrlo je važno bilo dobiti povratnu informaciju od budućih korisnika. Putem provedene ankete i detaljne analize utvrđeno je kako je 58% ispitanih spremno za korištenje beskontaktnog načina plaćanja.

Beskontaktni način plaćanja razvijen je putem nove *Apple Pay* tehnologije koja omogućuje dosad najsigurniji oblik digitalnog plaćanja zbog čvrste povezanosti programske podrške i popratnog sklopovlja. Plaćanje putem biometrijskog otiska prsta ili predodređene lozinke na uređaju puno je sigurniji od svih ostalih načina provođenja transakcija jer ima više slojeva sigurnosti ugrađenih u sustav. Aplikacija osim sigurnosne provjere korisnika putem otiska prsta ima implementiranu i sigurnosnu prijavu u aplikaciju putem *Google*ove platforme u oblaku računala te tako omogućuje dodatnu sigurnost aplikacije. Razvoj kvalitetne aplikacije s mogućnošću transakcijskog poslovanja dodatno je olakšan korištenjem *MVC* programske paradigme putem koje je aplikacija dodatno organizirana i omogućeno olakšano praćenje promjena.

Prednosti platforme nad konkurentskim platformama je što se kartični i transakcijski podaci ne spremaju na *Apple*ove poslužitelje i što je platforma sinergija sklopovlja i programske podrške razvijene od iste tvrtke te je korištenje resursa znatno bolje i sigurnije. Još jedna prednost implementiranog programskog rješenja je ta što se svi ostali podaci spremaju na poslužitelj te je na jednostavniji način omogućeno udaljeno proširivanje paketa pohrane što u suprotnom ne bi bilo moguće bez ažuriranja aplikacije jer bi se ti podaci morali implementirati izravno unutar aplikacije.

S druge strane, problem ovog načina plaćanja je što još uvijek nije odobren u Republici Hrvatskoj i što će vrijeme njegove integracije sigurno potrajati neko vrijeme, a to će dodatno otežati način izvršavanja plaćanja unutar aplikacije. Zaključno, programsko rješenje diplomskog rada trenutno je najsigurniji oblik beskontaktnog plaćanja koji zajedno s razvijenom *iOS* aplikacijom teži ka digitalizaciji društva te je implementirano tako da se u budućnosti vrlo lako može proširiti i na ostale platforme.

7 LITERATURA

[1] European Union, EU Transport in figures, 2016. Dostupno: https://ec.europa.eu/transport/sites/transport/files/pocketbook2016.pdf. [Pristup: 1. 2. 2017.].

[2] Z. Georgios, D. Proserpio, J.W. Byers. The rise of the sharing economy: Estimating the impact of Airbnb on the hotel industry. Journal of Marketing Research (2014)

[3] UN, Green economy, 2017. Dostupno: https://www.unep.org/greeneconomy/. [Pristup: 2. 10. 2017.].

[4] S. O'Kane, Person-to-person payments and an 'Apple Cash Card', 6. 5. 2017. [Mrežno]. Dostupno: https://www.theverge.com/2017/6/5/15731034/apple-pay-iphone-payments-venmo-update-announced-wwdc-2017. [Pristup: 6. 6. 2017].

[5] Jawale, Ashay S., and Joo S. Park. A Security Analysis on Apple Pay. Intelligence and Security Informatics Conference (EISIC), 2016 European. IEEE, 2016. .

[6] Leavitt, Neal. Are mobile payments ready to cash in yet?. Computer 45.9 (2012)

[7] A. F. a. D. T. Martin Arnold, UK banks put squeeze on Apple Pay fees, 14 6 2015. [Mrežno]. Dostupno: https://www.ft.com/content/02287f44-2a3d-11e5-8613-e7aedbb7bdb7. [Pristup: 3. 23. 2017].

[8] Eru, Dostupno: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2015.123.01.0001.01.ENG.

[9] J. W. a. E. A. M. Siegel, Early days, but Apple Pay struggles outside U.S., 2 6 2016. [Mrežno]. Dostupno: http://www.reuters.com/article/us-apple-pay-idUSKCN0YN61U). [Pristup: 20. 3. 2017.].

[10] J. W. a. E. A. M. Siegel, 2 6 2016. Dostupno: http://www.reuters.com/article/us-apple-pay-idUSKCN0YN61U. [Pristup: 20. 3. 2017.].

[11] L. Rao, Apple Pay Now Accounts for Three-Fourths of U.S. Contactless Payments, 26 7 2016. Dostupno: http://fortune.com/2016/07/26/apple-pay-contactless/. [Pristup: 22. 3. 2017].

[12] TXN, Apple Pay on the Rise, Dostupno: http://blog.txn.com/apple-pay-on-the-rise/. [Pristup: 7. 2. 2017.].

[13] Apple, iOS Feature Availability, Dostupno: http://www.apple.com/ios/feature-availability/#apple-pay. [Pristup: 11. 5. 2017.].

[14] Apple, Apple Pay is compatible with these devices, Dostupno: https://support.apple.com/sv-se/KM207105. [Pristup: 17. 6. 2017.].

[15] Apple, Use Touch ID on iPhone and iPad, Dostupno: https://support.apple.com/en-us/HT201371. [Pristup: 4. 4. 2017.].

[16] Apple, Using Apple Pay in stores, and within apps and websites, Dostupno: https://support.apple.com/en-us/HT201239. [Pristup: 20. 5. 2017.].

[17] Apple, About limits when using Apple Pay in stores, Dostupno: https://support.apple.com/en-us/HT207435. [Pristup: 22. 6. 2017.].

[18] D. Deino, I was there for the first iPhone release 10 years ago — here's what it was like, Dostupno: http://www.businessinsider.de/first-iphone-release-10-years-ago-2017-5?r=US&IR=T. [Pristup: 30. 5. 2017.].

[19] Apple, iOS 7 With Completely Redesigned User Interface & Great New Features Dostupno: https://www.apple.com/newsroom/2013/09/10iOS-7-With-Completely-

Redesigned-User-Interface-Great-New-Features-Dostupno-September-18/. [Pristup: 4 4 2017].

[20] M. Swider, iOS 8 features and updates, Dostupno: http://www.techradar.com/news/phone-and-communications/mobile-phones/ios-8-10-things-we-want-to-see-1166133. [Pristup: 4. 4. 2017.].

[21] J. Crook, Apple Rebrands Passbook To Wallet, Dostupno: https://techcrunch.com/2015/06/08/apple-rebrands-passbook-to-wallet/. [Pristup: 3. 3. 2017.].

[22] Apple, iOS 10, Dostupno: https://www.apple.com/ios/ios-10/. [Pristup: 15. 9. 2016.].

[23] Apple, Use Touch ID on MacBook Pro, [Pristup: 7. 6. 2016.].

[24] M. Swider, iOS 11 release date, news and features, Dostupno: http://www.techradar.com/news/ios-11-release-date-news-and-features. [Pristup: 27. 5. 2017.].

[25] Apple, Apple Developer Program, Dostupno: https://developer.apple.com/programs/. [Pristup: 8. 8. 2016.].

[26] P. Solt, Dostupno: http://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html. [Pristup: 15. 12. 2016.].

[27] A.J. Wagner, , G. Scalzo, J. Hoffman. Swift: Developing iOS Applications. Packt Publishing Ltd, 2016. APA.

[28] Apple, Swift, Dostupno: https://swift.org/. [Pristup: 2. 3. 2017.].

[29] Apple, About Apple Pay Dostupno: https://support.apple.com/en-us/HT201469. [Pristup: 12. 15. 2016.].

[30] Delivering Value, iPhone 6 – Quick Thoughts, Dostupno: http://blog.starpointllp.com/blog/?p=3883. [Pristup: 5. 2. 2017.].

[31] BellID, What is Apple Pay & how does it work? .

[32] BinDB LLC, Bin List (Binlist) & Bin Ranges, Dostupno: https://www.bindb.com/bin-list.html. [Pristup: 4. 4. 2017.].

[33] L. Gulsvig, MasterCard BIN Range Coming October 2016, Dostupno: https://www.forte.net/blog/mastercard-bin-range-coming/. [Pristup: 3. 22. 2017.].

[34] MLA Bhagavatula, Chandrasekhar, et al. Biometric authentication on iphone and android: Usability, perceptions, and influences on adoption. Proc. USEC (2015)

[35] Ratha, Nalini K., Jonathan H. Connell, and S. Pankanti. Big data approach to biometric-based identity analytics. IBM Journal of Research and Development (2015)

[36] PYMNTS, Apple Pay's Low-Tech Security Problem, Dostupno: http://www.pymnts.com/apple-pay-tracker/2016/apple-pays-low-tech-security-problem/. [Pristup: 4. 4. 2017].

[37] Apple, iOS Security, Dostupno: https://www.apple.com/business/docs/iOS_Security_Guide.pdf. [Pristup: 4. 4. 2017.].

[38] Hrvatska narodna banka, Platne kartice i kartične transakcije (statistika platnog prometa), Direkcija za izdavačku djelatnost Trg hrvatskih velikana 3, 10002 Zagreb, Zagreb, 2015..

[39] S. Kemp, 2017 DIGITAL YEARBOOK: DIGITAL DATA FOR EVERY COUNTRY IN THE WORLD, Dostupno: https://wearesocial.com/special-reports/2017-digital-yearbook-digital-data-every-country-world. [Pristup: 05. 10. 2017.].

[40] M. Sajid, Ozaire, and M. Haddara. NFC mobile payments: Are we ready for them?. SAI Computing Conference (SAI), 2016. IEEE, 2016. .

[41] F.D. Davis, A technology acceptance model for empirically testing new end-user information systems: Theory and results. Diss. Massachusetts Institute of Technology, 1985. APA .

[42] Davis, Fred D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS quarterly (1989): 319-340. APA.

[43] Lewin, Kurt. Frontiers in group dynamics: Concept, method and reality in social science; social equilibria and social change. Human relations 1.1 (1947)

[44] I. Mihalj, Spremnost društva na bezkontaktna plaćanja putem pametnih uređaja, Dostupno: https://goo.gl/forms/ObdBykX34OZrbDvF3. [Pristup: 3. 5. 2017.].

[45] Cohen, David, M. Lindvall, and P. Costa. Agile software development. DACS SOAR Report 11 (2003).

[46] D. Z. a. V. Sizov, The platform for modern developers, Dostupno: https://gitlab.com. [Pristup: 7. 1. 2017.].

[47] Postdot Technologies, Inc, Developing APIs is hard Postman makes it easy, Dostupno: https://www.getpostman.com/. [Pristup: 4. 4. 2017.].

[48] Apple, Apple Developer Program, Dostupno: https://developer.apple.com/programs/enroll/. [Pristup: 5. 8. 2016.].

[49] Apple, Apple PKI, 2017. Dostupno: https://www.apple.com/certificateauthority/. [Pristup: 4. 4. 2017.].

[50] S. Smiley, Active RFID vs. Passive RFID: What's the Difference?, Dostupno: http://blog.atlasrfidstore.com/active-rfid-vs-passive-rfid. [Pristup: 4. 4. 2017.].

[51] Margraf, Marian, S. Lange, and F. Otterbein. Security Evaluation of Apple Pay at Point-of-Sale Terminals. Next Generation Mobile Applications, Security and Technologies (NGMAST), 2016 10th International Conference on. IEEE, 2016. APA .

[52] Apple, Cocoa Core Competencies, Dostupno: https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html. [Pristup: 1. 2. 2017.].

[53] RestApiTutorial, Using HTTP Methods for RESTful Services, Dostupno: http://www.restapitutorial.com/lessons/httpmethods.html. [Pristup: 14. 4. 2017.].

[54] E. Durán, Get on with building your app, not duplicating code., Dostupno: https://cocoapods.org/. [Pristup: 7 11 2016].

[55] Google Developers, Everything you need to build and grow your app, all in one place., Dostupno: https://console.firebase.google.com/. [Pristup: 1. 2. 2017.].

[56] Apple, Human Interface Guidelines iOS, Dostupno: https://developer.apple.com/ios/human-interface-guidelines/technologies/apple-pay/. [Pristup: 12. 5. 2017.].

[57] J. Shier, Elegant HTTP Networking in Swift, Dostupno: https://github.com/Alamofire/Alamofire. [Pristup: 7. 6. 2017.].

[58] Apple, Wallet Passes Companion Files, Dostupno: https://developer.apple.com/services-account/download?path=/iOS/Wallet_Support_Materials/WalletCompanionFiles.zip. [Pristup: 2. 5. 2017.].

[59] Apple, iTunes Connect Developer Guide, Dostupno: https://developer.apple.com/library/content/documentation/LanguagesUtilities/Concept

ual/iTunesConnect_Guide/Chapters/SettingUpUserAccounts.html. [Pristup: 27. 3. 2017.].

[60] Apple , Apple Pay Sandbox Testing, Dostupno: https://developer.apple.com/support/apple-pay-sandbox/. [Pristup: 10. 4. 2017.].

[61] Functional Software, Inc, Track errors in every language, framework, and library., 2017. [Mrežno]. Dostupno: https://sentry.io/welcome/. [Pristup: 4. 20. 2017.].

[62] I. Walter. Steve jobs. JC Lattès, 2011. APA.

[63] Davis, Fred D. "Perceived usefulness, perceived ease of use, and user acceptance of information technology." MIS quarterly (1989)

[64] K. Fiveash, Apple Pay is a tidy payday for Apple with 0.15% cut, sources say, Dostupno: http://www.theregister.co.uk/2014/09/13/apple_to_get_15_cents_for_every_100_dollar_payment_on_its_pay_service_says_ft. [Pristup: 14. 12. 2016].

[65] European Commission, Collaborative economy, Dostupno: http://ec.europa.eu/growth/single-market/services/collaborative-economy_hr. [Pristup: 10. 1. 2107.].

[66] TXN, Apple Pay on the Rise, Dostupno: http://blog.txn.com/apple-pay-on-the-rise/. [Pristup: 7. 2. 2017].

[67] TXN, Apple Pay on the Rise, Dostupno: http://blog.txn.com/apple-pay-on-the-rise/. [Pristup: 7. 2. 2017.].

[68] Bohemian BV, Sketch Education Store, Dostupno: https://www.sketchapp.com/store/edu/. [Pristup: 1. 2. 2017.].

[69] European Union, EU Transport in figures, Dostupno: https://ec.europa.eu/transport/sites/transport/files/pocketbook2016.pdf. [Pristup: 1. 2. 2016.].

[70] B. Voo, Digital Wallets – 10 Mobile Payment Systems To Take You There, Dostupno: http://www.hongkiat.com/blog/digital-wallets/. [Pristup: 1. 3. 2017.].

[71] J. M. a. E. Lavandera, Why big companies buy, sell your data, Dostupno: http://edition.cnn.com/2012/08/23/tech/web/big-data-acxiom/index.html. [Pristup: 10. 3. 2017.].

[72] S. Kemp, DIGITAL IN 2017: GLOBAL OVERVIEW, Dostupno: https://wearesocial.com/special-reports/digital-in-2017-global-overview. [Pristup: 20. 3. 2017.].

[73] D. Chaffey, Global social media research summary 2017, Dostupno: http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/. [Pristup: 2. 5. 2017].

[74] R. A. Buchanan, History of technology, Dostupno: https://www.britannica.com/technology/history-of-technology. [Pristup: 5. 3. 2017].

8 SAŽETAK

U ovom radu osmišljen je i programski realiziran sustav za beskontaktni način plaćanja putem *iOS* aplikacije. Cilj rada bio je implementirati beskontaktni način plaćanja što je ostvareno unutar *iOS* aplikacije za dijeljenje prijevoznog mjesta. Aplikacija je razvijena pomoću programskog jezika *Swift* i programske paradigme *MVC*. Korištene su dodatne biblioteke za beskontaktni način plaćanja putem biometrijskog senzora, spajanje na *Apple Pay* usluge, pohrana podataka unutar sigurnosnog elementa, certificiranje putem razvojnog okruženja Stripe i pohrana transportnih kupona unutar aplikacije *Apple Wallet*. Za prijenos podataka između aplikacije i poslužitelja korišten je *JSON* format podataka. Dodatno su još korištene biblioteke za rad s *Google* mapama i sustav oblaka računala *Firebase* te je sve praćeno putem *Sandbox* i *Sentry* testnog razvojnog okruženja. Aplikacija je korisnicima omogućila traženje slobodnog prijevoznog mjesta, a osobama koje nude vožnju omogućila je dodatni oblik zarade, jednostavniji način obavljanja transakcija i pronalaženje korisnika na jednom mjestu.

Ključne riječi: *Apple Pay*, dijeljena ekonomija, paketi, transport, sigurne transakcije.

9 ABSTRACT

A system for contactless payment via iOS application has been designed and programmed in this paper. The purpose of this paper was to implement a contactless payment method which was accomplished within the iOS application for shared transport. The application has been developed by using the Swift programming language and the MVC paradigm. Additional libraries for contactless payment via touch ID, Apple Pay services, data storage inside the secure element, certification via Stripe software platform, and storage of transport passes within the Apple Wallet application have been used. The JSON data format has been used to transfer data between the application and the server. Additional libraries have been used for Google Maps and Firebase Cloud based service and they have all been tracked through the Sandbox and Sentry testing software platform. The application has enabled users to search for an available shared transport, and to people who are offering the ride a form of additional earnings and a simpler way to perform transactions and finding users in one place.

Key words: Apple Pay, packages, transport, secure transactions, sharing economy.

10 ŽIVOTOPIS

Ivan Mihalj rođen je 10.05.1992. godine u Đakovu. Završetkom Gimnazije upisuje preddiplomski studij na Elektrotehničkom fakultetu u Osijeku. Tijekom studija obnaša funkciju potpredsjednik *IEEE* ogranka unutar kojeg s kolegama osniva prvi Studentski radio za kojeg dobiva pismenu pohvalu dekana. Nakon povratka s prvog *Erasmus* semestra u Bremenu aktivno se uključuje u rad studentskih udruga na fakultetu te biva izabran u Studentski zbor Elektrotehničkog fakulteta gdje zajedno s kolegama postiže odlične rezultate i pomaže ulasku Elektrotehničkog fakulteta u Osijeku u organizatore Elektrijade. Zadnju godinu studija biva prepoznat od uprave Fakulteta i Sveučilišta te postaje studentski predstavnik u Europskom parlamentu na godinu dana te odlazi na poduku u Brisel. Osim bavljenja tehnologijom zanima ga poduzetništvo i osvaja treće mjesto na sveučilišnom studentskom poslovnom natjecanju te aktivno nastavlja raditi na proučavanju ekonomskih procesa gdje se stvara ideja o aplikaciji za transport ljudi i dobara putem sustava dijeljene ekonomije. Ivan prepoznaje potrebe najmlađih te u sklopu programa ”*Hour of Code*” posjećuje osnovne i srednje škole i podučava učenike o važnosti pisanja kôda. Nakon završenih 5 godina studija u roku, upisuje apsolventsku godinu radi odlaska na stručnu praksu u Austriju gdje se bavi razvojem i testiranjem *iOS* aplikacija te proučavanjem poduzetništva.

Ivan Mihalj

11 PRILOZI

Prilog 1. Diplomski rad, pdf i doc inačica

Prilog 2. Izvorni kôd mobilne aplikacije

Prilog 3. Grafički rezultati provedene ankete

POPIS POJMOVA

[1] Firebase – mobilna platforma koja olakšava kreiranje mobilnih aplikacija
[2] Biometrijski čitač otiska prsta – uređaj koji prepoznaje otisak prsta i omogućava otključavanje uređaja ili izvršavanje transakcije
[3] Wallet – Digitalni novčanik
[4] Sandbox – Okruženje unutar kojeg se mogu kreirati testni računi za testiranje Apple Pay platforme
[5] Stripe – Sustav preko kojeg se mogu izvršavati i pratiti transakcije
[6] Google Mape – Digitalne karte na pametnom uređaju
[7] Google Mjesta – Pretraga mjesta unutar digitalnih karti
[8] Treće članice – Osobe koje indirektno utječu na poslovanje sustava
[9] NFC – Komunikacija na malim udaljenostima
[10] Mac – Računalo razvijeno od tvrtke Apple
[11] Sigurnosni element – Mikro čip koji može pohraniti važne podatke i pokretati sigurnosne aplikacije poput onih za razmjenu transakcija
[12] Apple App Store – Trgovina mobilnim aplikacijama
[13] iMessages – Platforma za razmjenu poruka između korisnika Apple operacijskih sustava
[14] OS X – Operacijski sustav za Apple računala
[15] Xcode – Razvojno okruženje za kreiranje programske podrške za Apple uređaje
[16] iOS Simulator – Simulacija iPhonea
[17] Apple razvojni program – Program za razvojne inženjere koji žele razvijati za Apple platforme
[18] Cocoa pods – Upravitelj ovisnostima o drugim bibliotekama korištenim unutar *Xcode* razvojnog okruženja
[19] POS – Uređaj za kreiranje transakcija u trgovačkim lancima
[20] Sketch – Programska podrška za razvoj korisničkog sučelja
[21] MVC – Programska paradigma
[22] Github – Sustav verzioniranja kôda
[23] SourceTree – Korisničko sučelje za verzioniranje kôda
[24] GitLab – Sustav verzioniranja kôda
[25] Swift – Appleov novi programski jezik
[26] Razvojni inženjer – Osoba koja se bavi razvojem programske podrške
[27] PassKit – Biblioteka korištena za razvoj Apple Pay sustava
[28] Sigurnosna enklava – Dio A7 čipa gdje su spremljeni šifrirani podaci za otisak prsta i unutar koje se odvija proces Touch IDa
[29] EMVCo – Europay, Visa, Mastercard
[30] Dinamički sigurnosni kôd – Kôdza sigurnosnu ovjeru transakcija
[31] Broj uređaja – Jedinstveni broj iPhone uređaja
[32] CVV – Sigurnosni kôdkartice
[33] PAN – Primarni broj računa
[34] TSP – Davatelj usluga tokena
[35] RDN – Sigurnosni broj koji zamjenjuje stvarni broj kartice
[36] Pass - Kupon
[37] AES – Američki državni standard za šifriranje
[38] Shared pairing key – Dijeljeni ključ uparivanja
[39] UID – Jedinstveni identifikator (engl. *Unique Identifier*)
[40] HSM – Uređaj koji štiti i upravlja digitalnim ključem te omogućava strogo šifriranje
[41] Applet – Mali dio programa
[42] Google Forms – Sustav u oblaku računala za analizu rezultata

[43] EFTPOS – Uređaj namijenjen bezgotovinskom plaćanju pomoću kojeg se transakcije provode električnim putem
[44] Pretplatnik - Osoba koja je u zadnjih 90 dana aktivno koristila mobilni uređaj ili uplatila bon
[45] Model prihvaćanja tehnologije – Metoda analiziranja prihvaćenosti tehnologije u društvu
[46] Procjena korisnosti – Metoda procjene korisnosti određene tehnologije preko mjerenja učinkovitosti nakon korištenja programske podrške
[47] Procjena jednostavnosti korištenja - Stupanj prema kojem osoba vjeruje kako će korištenje određenog sustava biti jednostavnije
[48] Model upravljanja promjenama – Model kojemu je cilj promjena mišljenja korisnika
[49] P – Ukupan zbroj pozitivnih i negativnih postotaka odgovora radi dobivanja ukupne srednje vrijednosti rezultata
[50] Storyboard – Naziv za objekt korisničkog sučelja kreiran unutar Xcode okruženja
[51] Sprint – Dio Agilnog razvoja programske podrške koji predstavlja vremenski period kada određeni zadatak treba biti implementiran, testiran i spreman za pregled
[52] Version control system – Sustav upravljanja inačicom programske inačice
[53] Auto Layout – Način automatskog proširivanja i poslagivanja objekta unutar korisničkog sučelja
[54] Tab Bar Controller – Način implementacije korisničkog sučelja unutar iOS platforme
[55] 3D Sigurnosni protokol – Protokol za Internet plaćanje putem kreditnih i debitnih kartica