Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki

Praca dyplomowa inżynierska

ELEKTRONICZNA LEGITYMACJA STUDENCKA W TELEFONIE

Michał Cieśnik, 117287
Kamil Musiał, 122590
Maciej Śledź, 122492
Rafał Załuski, 122529

Promotor
dr hab. inż. Marek Mika

Opiekun
mgr inż. Marek Gosławski

Poznań, 2018 r.

Miejsce na kartę pracy dyplomowej.

Pragniemy szczególnie podziękować dr. hab. inż. Markowi Mice oraz mgr. inż. Markowi Gosławskiemu za wskazówki udzielone podczas realizacji projektu, jak również Annie Cieśnik za pomoc w koordynacji członków zespołu.

Spis treści

1 Wstęp **1**

1.1 Cel i zakres pracy . . . 1

1.2 Motywacja . . . 1

1.3 Istniejące rozwiązania . . . 2

1.4 Struktura pracy . . . 3

1.5 Podział zadań . . . 3

2 Podstawy teoretyczne **5**

2.1 Karty elektroniczne . . . 5

2.1.1 Historia . . . 5

2.1.2 Typy kart . . . 5

2.1.3 Interfejsy . . . 5

2.1.4 Standardy ISO . . . 6

2.1.5 Komunikacja . . . 6

2.1.6 System plików . . . 6

2.1.7 Reprezentacja danych . . . 7

2.2 NFC . . . 7

2.2.1 Standaryzacja . . . 8

2.2.2 Protokół NFC . . . 8

2.2.3 Tryby pracy . . . 8

Tryb pasywny . . . 8

Tryb aktywny . . . 8

2.3 Elektroniczna Legitymacja Studencka i Elektroniczna Legitymacja Doktoranta . . . 8

2.3.1 Podstawa prawna . . . 8

2.3.2 Struktura danych . . . 9

2.3.3 Dodatkowe aplikacje na karcie . . . 9

Aplikacja biblioteczna . . . 9

Poznańska Elektroniczna Karta Aglomeracyjna . . . 10

Aplikacja płatnicza . . . 10

2.3.4 Bezpieczeństwo . . . 10

2.3.5 Specyfika kart wydawanych w MCPLS . . . 11

3 Specyfikacja wymagań **13**

3.1 Opowieści użytkownika . . . 13

3.2 Specyfikacja wymagań funkcjonalnych . . . 14

3.2.1 Podstawowe funkcjonalności . . . 14

3.2.2 Emulacja . . . 15

3.3 Specyfikacja wymagań pozafunkcjonalnych . . . 15

3.4 Interfejs użytkownika 16

4 Modelowanie biznesowe aplikacji 21

4.1 Aktorzy 21
4.2 Obiekty 21
4.3 Przypadki użycia 22
4.4 Problemy i koncepcja ich rozwiązania 23

5 Projekt i implementacja aplikacji 25

5.1 Opis architektury 25
5.2 Implementacja 25
5.2.1 Narzędzia 25
5.2.2 Technologia 26
5.2.3 Modele kart - plik konfiguracyjny 27
5.2.4 Odczyt danych z kart 28
5.2.5 Sprawdzanie certyfikatów 29
5.2.6 Zapis danych w pamięci telefonu 30
5.2.7 Emulacja kart 30
5.2.8 Interfejs użytkownika 34
5.3 Testy 35
5.3.1 Testy akceptacyjne 35
5.4 Dokumentacja 37

6 Zalecenia wdrożeniowe 39

6.1 Plan wdrożenia oprogramowania 39
6.2 Dalszy rozwój aplikacji 39

7 Podsumowanie 41

Literatura 43

Rozdział 1

Wstęp

1.1 Cel i zakres pracy

Celem pracy jest przygotowanie funkcjonalności związanej z Elektroniczną Legitymacją Studencką (ELS) i Elektroniczną Legitymacją Doktorancką (ELD) dla aplikacji mobilnej uczelni (PPulse) z wykorzystaniem interfejsu NFC. Funkcjonalność ta ma być zaimplementowana w postaci prototypu jako oddzielna aplikacja, której kod w późniejszym czasie zostanie przeniesiony i wdrożony do uczelnianej aplikacji. Projekt wymaga dostosowania się do technologii użytej w PPulse, aby ułatwić późniejsze wdrożenie utworzonego oprogramowania. Wygląd, kolorystyka oraz interfejs użytkownika również muszą zostać przygotowane w stylu nie odbiegającym od aplikacji docelowej. Aplikacja powinna umożliwiać odczyt informacji oraz sprawdzenie ważności ELS/ELD. W przypadku gdy ELS/ELD posiada dodatkowe informacje (identyfikator czytelnika/kod kreskowy, PEKA, certyfikaty itd.), aplikacja również powinna móc je odczytać. Odczytaną zawartość aplikacji ELS/ELD powinno dać się skopiować i zapisać w pamięci telefonu. Następnie zapisaną kartę można emulować wykorzystując interfejs NFC. W tym celu należy przygotować zbiór komend i poleceń na które aplikacja powinna odpowiadać podczas komunikacji z urządzeniem próbującym odczytać emulowane dane.

1.2 Motywacja

Rozwój nauki i technologii jest widoczny gołym okiem w każdej dziedzinie życia. Wystarczy spojrzeć parę lat wstecz. Technologie, o których jeszcze niedawno nawet nie marzono, teraz towarzyszą każdemu na porządku dziennym. Ludzkość idzie w kierunku minimalizacji i automatyzacji. Celem jest ułatwienie sobie życia w maksymalnie możliwy sposób tak, aby obowiązki stały się znacznie mniej niewygodne, a czas przy nich spędzony był znacznie krótszy. Dzięki temu można zyskać więcej czasu wolnego i skupić się na rzeczach przyjemniejszych, oddać się swoim pasjom. XXI wiek należy do ery smartfonów. Aktualnie bardzo rzadko spotyka się osobę bez własnego telefonu. Urządzenia mobile posiadają coraz więcej możliwości i zastępują znaczną część przedmiotów codziennego użytku. Elektroniczna Legitymacja Studencka w telefonie to jeden z projektów, którego zamysłem jest redukcja osobistych dokumentów posiadanych na co dzień przy sobie. Raczej nie zdarza się, aby ktoś wyszedł z domu bez swojego smartfonu. Znacznie częściej zapomina się o takich przedmiotach jak legitymacja.

Elektroniczna Legitymacja Studencka przede wszystkim przechowuje podstawowe dane o studencie, takie jak PESEL, czy numer indeksu. Legitymacja studencka potwierdza status studenta uprawniający do zniżek w wielu miejscach. Skopiowanie plików legitymacji do telefonu może w przyszłości umożliwić korzystanie z uprawnień bez konieczności posiadania przy sobie samej

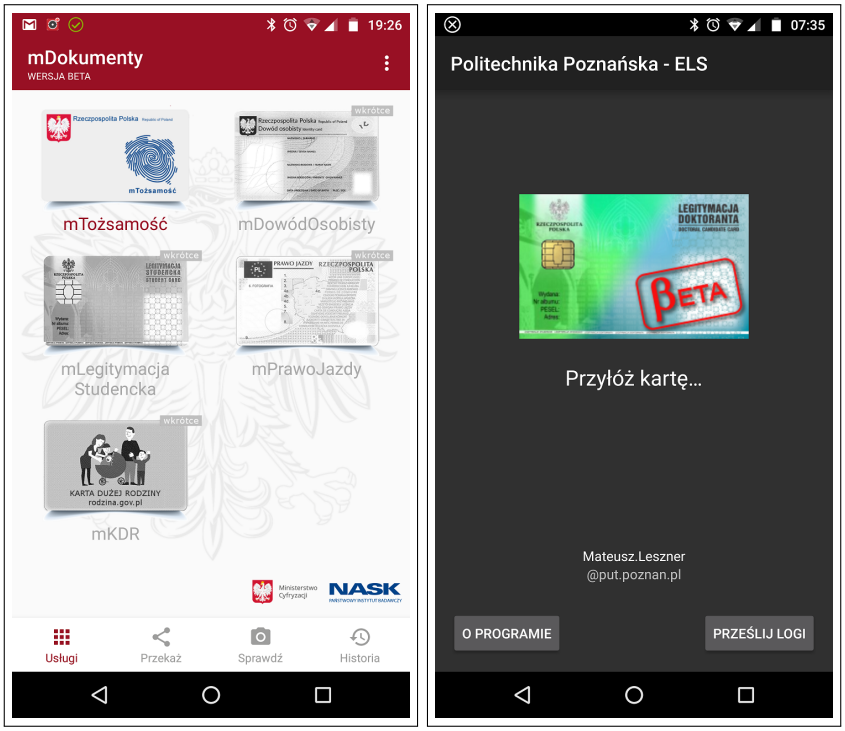
karty. Może to być przydatne dla studentów, którzy mają problem z źapominaniemźabrania ze sobą legitymacji. Dzięki aplikacji mają ją zawsze pod ręką, a być w przyszłości taka funkcja będzie bardziej istotna jeśli tego typu legitymacje zostaną wprowadzone np. do szkół podstawowych. Dzieci zdecydowanie mają większe trudności z pamiętaniem o zabraniu ze sobą legitymacji, a aplikacja w niektórych sytuacjach mogłaby znacznie im pomóc.

Ponadto jednym z założeń Elektronicznej Legitymacji Studenckiej w telefonie jest możliwość emulowania karty, czyli emulacji danych na niej zawartych wykorzystując do tego interfejs NFC urządzenia mobilnego. Pierwsza myśl, jaka przychodzi do głowy to wykorzystywanie smartfonu zbliżeniowo np. w komunikacji miejskiej czy na parkingu uczelnianym jako zastępnik zwykłej karty. Wystarczy przyłożyć telefon do czytnika, aby ten odczytał to jak oryginał. Byłoby to bardzo ciekawe oraz wygodne zastosowanie aplikacji. Projekt jest interesujący i może znaleźć zastosowanie w praktyce.

1.3 Istniejące rozwiązania

Projekt Ministerstwa Cyfryzacji o nazwie mDokumenty ma na celu ograniczenie potrzeby korzystania z tradycyjnej postaci dokumentów takich jak dowód osobisty, prawo jazdy, legitymacja studencka, legitymacja szkolna czy karta dużej rodziny. Korzystanie z mDokumentów wymaga zainstalowania aplikacji mObywatel, dostępnej na systemy Android i iOS. Obecnie (kwiecień 2018 r.) dostępna jest jedynie usługa mTożsamość, która służy do identyfikacji na linii obywatel–obywatel, lub obywatel–przedsiębiorca. W październiku 2018r. planowane jest uruchomienie usługi legitymacji szkolnej i legitymacji studenckiej w formie mobilnych dokumentów.

Istnieje również aplikacja ELS PP dostępna na smartfony z systemem Android, przygotowana przez Mateusza Lesznera, absolwenta Wydziału Informatyki Politechniki Poznańskiej. Aplikacja umożliwia odczyt danych z elektronicznych legitymacji studenckich i elektronicznych legitymacji doktoranckich korzystając z interfejsu NFC.



Rysunek 1.1: Aplikacja mObywatel i ELS PP (zrzuty ekranu)

1.4 Struktura pracy

Pierwszy rozdział jest wstępem pracy inżynierskiej. Zawiera cel jaki powinien zostać uzyskany w przypadku zrealizowania projektu oraz motywację jego podjęcia. Przedstawiony został w nim również podział zadań zespołu. W rozdziale drugim opisane są podstawy teoretyczne niezbędne do realizacji projektu. Zapoznanie się z nimi jest potrzebne do pełnego zrozumienia kolejnych rozdziałów pracy. Rozdział trzeci został poświęcony został specyfikacji wymagań funkcjonalnych oraz pozafunkcjonalnych aplikacji ELS w Telefonie. Rozdział czwarty zawiera modelowanie biznesowe aplikacji. Opisuje docelowych użytkowników systemu oraz powiązane z użytkowaniem aplikacji urządzenia. Ponadto przedstawione są w nim przypadki użycia aplikacji oraz problemy napotkane przy realizacji projektu wraz z koncepcjami ich rozwiązania. W rozdziale piątym przedstawiony został projekt aplikacji zawierający opis architektury, implementacji i testów oraz dokumentację. Rozdział szósty dotyczy planu wdrożenia utworzonego oprogramowania oraz rozwoju i możliwych zastosowań aplikacji w przyszłości. Rozdział siódmy jest podsumowaniem wyników pracy.

1.5 Podział zadań

Złożoność projektu pozwoliła na rozdzielenie pracy pomiędzy członków zespołu i zrównoleglenie pracy.

- Za zaimplementowanie odczytu kart oraz sprawdzanie certyfikatów był odpowiedzialny Kamil Musiał.
- Zapisywanie oraz odczytywanie kart z pamięci telefonu, a także przeprowadzanie testów należało do Macieja Śledzia.
- Napisaniem funkcji emulacji kart zajął się Michał Cieśnik.
- Za przygotowanie interfejsu użytkownika był odpowiedzialny Rafał Załuski.

Rozdział 2

Podstawy teoretyczne

2.1 Karty elektroniczne

2.1.1 Historia

Technologia kart elektronicznych powstała z rozwinięcia kart z paskiem magnetycznym, które były wykorzystywane jako karty płatnicze oraz karty telefoniczne. Karty z układem elektronicznym oferują lepsze bezpieczeństwo przechowywanych danych oraz dają możliwość lokalnej weryfikacji PINu bez oczekiwania na połączenie z bankiem.

Przedstawienie historii rozwoju i wprowadzenia technologii kart elektronicznych na rynek wykracza poza zakres tej pracy. Informacje na ten temat można znaleźć w literaturze przedmiotu[6].

2.1.2 Typy kart

Ze względu na rodzaj zastosowanego układu elektronicznego wyróżniamy dwa rodzaje kart[5]:

- Karty pamięciowe
  Karty tego typu umożliwiają bezpośredni dostęp do układu pamięci. Dostęp do obszarów pamięci może być chroniony przez układ logiczny. Pomimo ograniczonej funkcjonalności karty tego typu znajdują zastosowanie jako karty telefoniczne. Zaletą tych kart jest niski koszt produkcji. Mniejsze zapotrzebowanie na prąd umożliwia działanie pamięciowych kart bezstykowych w większej odległości od terminala.

- Karty procesorowe
  Karty procesorowe posiadają mikroprocesor, który pośredniczy w dostępie do pamięci i umożliwia realizację bardziej złożonej logiki aplikacji. We współczesnych kartach procesorowych wiele aplikacji może współistnieć na jednej karcie pod nadzorem systemu operacyjnego. Zazwyczaj istnieje możliwość wgrania dodatkowych aplikacji na kartę już po zakończeniu personalizacji. Głównym ograniczeniem dla nowych zastosowań kart procesorowych jest obecnie ograniczona prędkość transmisji danych.

2.1.3 Interfejsy

Układ elektroniczny może korzystać z następujących interfejsów:

- Interfejs stykowy — karta wyposażona jest w styki, dzięki którym możliwa jest komunikacja z kartą.

– Interfejs bezstykowy — nie wymaga elektrycznego połączenia między kartą a terminalem. Zasilanie układu elektronicznego i wymiana danych następują na zasadzie sprzężenia indukcyjnego obwodu anteny karty oraz obwodu anteny terminala.

2.1.4 Standardy ISO

ISO/IEC 7816 i ISO/IEC 14443 to zbiór międzynarodowych standardów związanych z kartami elektronicznymi. Standardy są zarządzane wspólnie przez Międzynarodową Organizację Normalizacyjną (ISO) oraz Międzynarodową Komisję Elektrotechniczną [2, 3].

Część pierwsza normy o numerze 7816 definiuje fizyczne parametry materiału, z jakiego zbudowane są karty, głównie przez odniesienie do normy określającej sposób badania wytrzymałości kart. W części drugiej opisano umiejscowienie punktów styku mikroprocesora karty z czytnikiem na karcie, jak i minimalne wymiary tych styków. Część trzecia definiuje zasady komunikacji pomiędzy czytnikiem a kartą na poziomie elektrycznym, a czwarta formy przechowywania danych na kartach oraz sposoby dostępu do nich i manipulowania nimi.

ISO/IEC 14443 to norma opisująca elektroniczne karty zbliżeniowe. Część pierwsza dotyczy fizycznych parametrów kart zbliżeniowych. Druga część opisuje wykorzystanie pola magnetycznego do transmisji danych i zasilania. Część trzecia definiuje procedury wywołania protokołu oraz metody zapobiegania kolizji. W części czwartej opisany jest sam protokół transmisji. Określone są dwa typy transmisji: Typ A oraz Typ B.

Różnice między komunikacją stykową a bezstykową istnieją głównie na poziomie warstwy fizycznej.

2.1.5 Komunikacja

Karta elektroniczna nie posiada własnego źródła zasilania. Układ elektroniczny rozpoczyna działanie w momencie kontaktu styków karty ze stykami terminala lub zbliżenia karty do terminala w przypadku kart z interfejsem bezstykowym. Następuje wtedy automatyczny reset oraz wysłanie ATR (ang. *Answer to Reset* - odpowiedź na reset). W komunikacji bezstykowej podobną funkcję pełni ATS (ang. *Answer to Select* - odpowiedź na wybranie). Terminal analizuje otrzymany ATR lub ATS, który zawiera informacje o parametrach transmisji, a następnie wysyła pierwsze polecenie. Po otrzymaniu polecenia karta wykonuje zadaną operację i wysyła odpowiedź do terminala. Wysłanie następnego polecenia może nastąpić dopiero po odebraniu odpowiedzi na poprzednie polecenie. Polecenia, jak i odpowiedzi są przesyłane w strukturze APDU (*Application Protocol Data Unit*), która jest przedstawiona w tablicy 2.1. Znaczenie poszczególnych pól struktury zostało opisane w standardzie ISO/IEC 7816-4[2].

Standard ISO/IEC 7816-4 nie narzuca zbioru poleceń, które powinny być obsługiwane przez kartę elektroniczną.

2.1.6 System plików

Istnieją dwa typy plików: pliki dedykowane (DF) oraz pliki elementarne (EF). Pliki dedykowane mogą zawierać dane lub zbiór plików podrzędnych. DF może być katalogiem aplikacji, wówczas jego nazwa jest identyfikatorem aplikacji (AID *application identifier*). Pliki elementarne zawierają wyłącznie dane i nie mogą mieć przypisanych plików podrzędnych.

Możliwe są dwa sposoby logicznej organizacji plików:

– Pliki dedykowane tworzą strukturę katalogową z pojedynczym katalogiem głównym (MF – *master file*),

Tablica 2.1: Para polecenie – odpowiedź

Pole	Opis	Liczba bajtów
Nagłówek polecenia	Bajt klasy CLA	1
	Bajt instrukcji INS	1
	Bajty parametrów P1–P2	2
Pole L_c	Brak tego pola oznacza $N_c = 0$, obecność pola oznacza $N_c > 0$	0, 1 lub 3
Pole danych polecenia	Pominięte dla $N_c = 0$, ciąg N_c bajtów dla $N_c > 0$	N_c
Pole L_e	Brak tego pola oznacza $N_e = 0$, obecność pola oznacza $N_e > 0$	0, 1, 2 lub 3
Pole danych odpowiedzi	Pominięte dla $N_c = 0$, ciąg N_c bajtów dla $N_c > 0$	N_r (nie więcej niż N_e)
Zakończenie odpowiedzi	Bajty statusu SW1–SW2	2

– Wszystkie DF aplikacji są równorzędne bez widocznej hierarchii katalogów.

Wykonywanie operacji na pliku wymaga, aby plik był wcześniej wybrany. Wybór pliku może być jawny, poprzez wykonanie polecenia SELECT z 16 bitowym identyfikatorem (FID *file identifier*). Identyfikatory te nie są unikalne globalnie, a jedynie w ramach katalogu. Pliki w jednym katalogu mają FID różne od siebie oraz nadrzędnego katalogu. Niektóre polecenia (na przykład READ BINARY) umożliwiają wybór pliku elementarnego w bieżącym katalogu przez podanie jako parametru krótkiego, 5 bitowego, identyfikatora pliku (SFI – *short file identifier*). Katalog główny jest wybrany bezpośrednio po resecie karty i może być w dowolnym momencie wybrany ponownie przez jego identyfikator `0x3f00`. Wybranie konkretnego pliku następuje z reguły przez wybór poszczególnych katalogów w ścieżce.

Pliki elementarne mogą być przezroczyste, lub z narzuconą strukturą:

– Liniowa stała — składa się z rekordów o stałej długości. Dostęp do rekordów jest swobodny.

– Liniowa zmienna — z rekordami o zmiennej długości.

– Cykliczna — rekordy o stałej długości ułożone w listę cykliczną. Dostęp do rekordów jest sekwencyjny.

2.1.7 Reprezentacja danych

Dane na ELS i ELD zapisane są w notacji ASN.1 (ang. *Abstract Syntax Notation One*) i zakodowane w formacie DER (ang. *Distinguished Encoding Rules*)[16].

Obiekty ASN.1 są kodowane w strukturze TLV, gdzie T (ang. *tag*) jest identyfikatorem typu, L (ang. *length*) oznacza długość a V (ang. *value*) to same dane obiektu. Przewagą tego rodzaju kodowania nad statyczną strukturą binarną jest częściowa przezroczystość. Struktura może być poszerzana o dodatkowe pola bez zrywania kompatybilności ze starszymi systemami.

2.2 NFC

NFC (ang. *Near Field Communication*) to technologia komunikacji bezprzewodowej krótkiego zasięgu, stworzona głównie z myślą o zastosowaniach w telefonach komórkowych. Oferuje połą-

czenie możliwości kart elektronicznych i technologii sieciowych takich jak Bluetooth i Wi–Fi[7]. Komunikacja wymaga zbliżenia dwóch urządzeń NFC na odległość kilku centymetrów. Krótki zasięg ogranicza możliwość podsłuchania transmisji, zwiększając bezpieczeństwo. NFC umożliwia wymianę danych między dwoma urządzeniami mobilnymi, jak również odczyt tagów RFID oraz elektronicznych kart identyfikacyjnych.

2.2.1 Standaryzacja

NFC Forum jest organizacją non-profit utworzoną przez firmy NXP (wówczas Philips), Nokia i'Sony w 2004 roku[6]. Celem NFC Forum jest przygotowanie jednolitej specyfikacji standardu NFC i przewodniczenie w rozwoju technologii[8]. W 2006 r. organizacja opublikowała zarys architektury technologii NFC. Określone zostały 4 typy tagów NFC. Typ 4 jest w pełni zgodny ze standardem ISO/IEC 14443.

Pełna specyfikacja NFC jest zawarta w standardzie ISO/IEC 18092.

2.2.2 Protokół NFC

Protokół NFC jest podobny do protokołu bezstykowych kart elektronicznych. Istotną różnicą jest brak podziału na urządzenia typu master i slave. Komunikacja jest możliwa w trybie półdupleks, czyli interfejs może pełnić w danym momencie funkcję nadawczą lub odbiorczą, ale nie obie jednocześnie.

Występuje rozróżnienie między inicjatorem (ang. *initiator*) i interfejsem docelowym (ang. *target*). Urządzenie rozpoczynające komunikacje staje się inicjatorem. Inicjator pełni funkcję podobną do terminala w systemach kart zbliżeniowych, kontrolując wymianę danych. Drugi z interfejsów automatycznie przyjmuje rolę interfejsu docelowego. Taki podział obowiązuje na cały czas trwania połączenia.

2.2.3 Tryby pracy

Tryb pasywny

Komunikacja w trybie pasywnym przebiega podobnie, jak w systemach RFID i bezstykowych kart elektronicznych. Interfejs inicjujący (ang. *initiator*) stale wytwarza pole magnetyczne. Interfejs docelowy odpowiada modulując to pole i może czerpać z niego energię elektryczną do zasilania układu.

Tryb aktywny

W trybie aktywnym obydwa urządzenia naprzemiennie generują pole magnetyczne. Po zakończeniu nadawania, interfejs przestaje generować pole magnetyczne.

2.3 Elektroniczna Legitymacja Studencka i Elektroniczna Legitymacja Doktoranta

2.3.1 Podstawa prawna

Dokumentem regulującym elektroniczne legitymacje studenckie wydawane przez uczelnie w Polsce jest Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 16 września 2016 w sprawie dokumentacji przebiegu studiów[9]. Dla legitymacji doktoranta jest to Rozporządzenie Ministra

Nauki i Szkolnictwa Wyższego z dnia 9 sierpnia 2017 r. w sprawie studiów doktoranckich i stypendiów doktoranckich[10]. Legitymacja studencka może być wydana w formie legitymacji papierowej lub legitymacji elektronicznej, natomiast legitymacja doktoranta musi być legitymacją elektroniczną. Rozporządzenia wymagają, aby karty posiadały interfejs stykowy określony w normach ISO/IEC 7816-2 i ISO/IEC 7816-3.

2.3.2 Struktura danych

Rozporządzenia określają format plików dla legitymacji studenckiej i doktoranta. Struktura plików aplikacji ELS jest przedstawiona na rysunku 2.1. Dane zapisane są w plikach odpowiednio `DF.SELS/EF.ELS` i `DF.SELD/EF.ELD` w strukturze opisanej w tablicy 2.2.

Format jest jednakowy binarnie z różnie nazwanymi, ale analogicznymi polami.

Dane podpisane są certyfikatem kwalifikowanym, który zapisany jest w pliku `EF.CERT`.

Rysunek 2.1: Struktura plików aplikacji ELS

```
DF.SELS.....................D6 16 00 00 30 01 01
└─ EF.CERT..................................00 01
   └─ EF.ELS................................00 02
```

Tablica 2.2: Struktura danych w legitymacjach studenckiej i doktoranta

ELS	ELD	typ
wersja		INTEGER
numerSeryjnyUkladu		PrintableString (SIZE (8..16))
nazwaUczelni	nazwaUczelni-lubJednostkinaukowej	UTF8String (SIZE (1..128))
nazwiskoStudenta	nazwiskoDoktoranta	SEQUENCE OF UTF8String (SIZE (1..28))
imionaStudenta	imionaDoktoranta	SEQUENCE OF UTF8String (SIZE (1..24))
numerAlbumu	numerLegitymacji	PrintableString (SIZE (1..16))
numerEdycji		PrintableString (SIZE (1))
numerPesel		PrintableString (SIZE (11))
dataWaznosci		GeneralizedTime

2.3.3 Dodatkowe aplikacje na karcie

Karta procesorowa może być dynamicznie rozszerzana przez instalowanie dodatkowych aplikacji. Elektroniczne legitymacje studenckie i doktoranckie mogą być wykorzystywane jednocześnie jako karta biblioteczna, karta komunikacji miejskiej lub karta płatnicza.

Aplikacja biblioteczna

Dane czytelnika przechowywane są w ramach osobnej aplikacji bibliotecznej — ELIB[14]. Plik `ELIB.ID` (rysunek 2.2) zawiera identyfikatory czytelnika w różnych bibliotekach i sieciach bibliotek. W pliku `ELIB.EVENT` zapisane jest do 10 ostatnich zdażeń — wypożyczeń lub zwrotów. Ponadto w pliku `ELIB.LOCK` przechowywane są dane dla każdego z kont czytelnika:

- data ostatniej wizyty w bibliotece,
- data ważności konta,
- liczba wypożyczonych książek,
- blokady — nieoddanie książek w terminie, zniszczenie lub zgubienie książek, nieuregulowane opłaty i kary.

Podczas personalizacji legitymacji poza danymi na rewersie karty nadrukowywany jest też identyfikator czytelnika w formie kodu kreskowego.

Obecnie wykorzystywana jest uproszczona wersja aplikacji bibliotecznej, która realizuje dedykowany dla niej zbiór poleceń i odpowiedzi APDU, a ograniczona struktura plikowa sprowadzona została wyłączenie do wewnętrznego sposobu przechowywania danych i nie jest dostępna dla użytkowników.

Rysunek 2.2: Struktura plików aplikacji bibliotecznej

```
DF.ELIB.....................D6 16 00 00 40 01 01
 ├─ EF.CONFIG...............................00 01
 ├─ EF.ID ..................................01 00
 ├─ EF.EVENT................................02 00
 └─ EF.LOCK ................................03 00
```

Poznańska Elektroniczna Karta Aglomeracyjna

Nowsze modele legitymacji (wydane po sierpniu 2013 roku) posiadają interfejs bezprzewodowy zgodny z Poznańską Elektroniczną Kartą Aglomeracyjną (PEKA). Dla tych legitymacji istnieje możliwość korzystania z nich jako nośnika ulgowych biletów okresowych lub tPortmonetki do opłacania pojedynczych przejazdów.

Aplikacja płatnicza

W ostatnich latach MCPLS nawiązało współpracę z bankami, dzięki czemu legitymacja studencka i doktorancka mogła posiadać funkcję płatniczą realizowana za pomocą aplikacji płatniczej dostarczonej przez jedną z międzynarodowych organizacji płatniczych: Visa lub MasterCard.

2.3.4 Bezpieczeństwo

Zabezpieczenie danych na karcie ELS/ELD odbywa się przede wszystkim poprzez podpisanie danych podpisem kwalifikowanym. Dane studenta/doktoranta znajdują w podpisanej kluczem publicznym wystawcy legitymacji wiadomości zgodnej z PKCS #7. Kluczem publiczny wystawcy legitymacji, który także znajduje się na karcie, występuje w formie certyfikatu X.509. Ten certyfikat w polu „właściciel certyfikatu” w atrybucie „nazwa powszechna” zawiera sformułowanie „osoba upoważniona do wystawiania legitymacji studenckiej” lub odpowiednio „osoba upoważniona do wystawiania legitymacji doktoranta”. Gwarantuje to cyfrowe podpisywanie legitymacji tylko przez upoważnione osoby. Certyfikat jest podpisany przez pośredni urząd certyfikacji wydający certyfikaty kwalifikowane, którego certyfikat jest z kolei podpisany przez Narodowe Centrum Certyfikacji (NCCert). Taka ścieżka certyfikacji od podpisanych danych do zaufanego certyfikatu NCCert zapewnia poprawność zapisanych danych.

Jednym z pól znajdujących się w strukturze podpisywanych danych jest numer seryjny karty inteligentnej. Celem tego pola jest zapobiegnięcie sytuacji w której dane z karty są przegrywane na inną. Później przy ewentualnej weryfikacji tożsamości miałoby to zapobiec kradzieży tożsamości lub wyłudzeniu zniżki studenckiej. Sprawdzenie polega na odczycie numeru seryjnego karty w sposób niezależny od aplikacji ELD/ELD, bezpośrednio z systemu karty. Niestety pobieranie numeru seryjnego jest zależne od modelu karty, co utrudnia stworzenie uniwersalnego systemu sprawdzającego. Ponadto teoretycznie możliwe jest stworzenie karty, która jest w stanie emulować dowolne dane, w tym także numer seryjny. Aktualnie nie istnieje na rynku takie rozwiązanie, ale w momencie gdy system elektronicznej legitymacji się rozpowszechni, możliwe jest, że powstanie. Z tych powodów ELS w Telefonie nie sprawdza numeru seryjnego karty.

Inną kwestią związaną z bezpieczeństwem jest zabezpieczenie zapisywanych danych przez ELS w Telefonie. Możliwym sposobem dokonania tego jest zaszyfrowanie tych danych hasłem podawanym przez użytkownika. Jest to sposób, który zapewnia bezpieczeństwo danych nawet w sytuacji, gdy telefon wejdzie w posiadanie osób niepożądanych z możliwością swobodnego dostępu do pamięci urządzenia. Ten sposób został wykorzystany w systemie mDokumenty Ministerstwa Cyfryzacji. Ze względu na uciążliwość dla użytkownika polegającą na potrzebie podawania hasła przy każdym uruchomieniu aplikacji oraz na trwałej utracie danych w przypadku zapomnienia hasła, ELS w Telefonie nie szyfruje zapisywanych danych. Jednak jest to rozwiązanie nad którym należy się zastanowić w aplikacji docelowej.

2.3.5 Specyfika kart wydawanych w MCPLS

Jednostką odpowiedzialną za personalizację ELS wydawanych przez Politechnikę Poznańską jest Międzyuczelniane Centrum Personalizacji Legitymacji Studenckiej (MCPLS). Personalizacja polega zarówno na nadrukowaniu danych studenta na blankiet jak i zapisaniu w układzie elektronicznym danych dotyczących uczelni i studenta[13].

Oprócz wymaganego interfejsu stykowego, legitymacje personalizowane w MCPLS wyposażone są w interfejs bezstykowy. Podstawą realizacji projektu jest możliwość odczytu danych aplikacji ELS i ELD poprzez interfejs bezstykowy.

Rozdział 3

Specyfikacja wymagań

W rozdziale tym rozważone zostaną wszelkie *wymagania funkcjonalne* oraz wymagania pozafunkcjonalne związane z budową aplikacji na smartfony. W pierwszej kolejności przeanalizowane będą potrzeby głównego użytkownika systemu w postaci *opowieści użytkownika* (ang. *user stories*). Na tej podstawie zostaną sformułowane wymagania. Ostatecznie pojawi się zarys oczekiwanego interfejsu użytkownika i w jednym miejscu zostaną zebrane wszelkie oczekiwania wobec projektu, które ostatecznie będzie można skonfrontować z powstałą implementacją.

3.1 Opowieści użytkownika

- Jako użytkownik aplikacji (student/doktorant), chciałbym mieć możliwość zapisu odczytanych danych z legitymacji w pamięci telefonu, abym nie musiał ponawiać ich odczytu za każdym razem, gdy zamierzam odwołać się do nich.

- Jako student/doktorant, chciałbym mieć możliwość odczytu danych z legitymacji i ich wyświetlenia za pomocą telefonu, aby w łatwy i szybki sposób dotrzeć do potrzebnych mi informacji.

- Jako student/doktorant, chciałbym mieć możliwość emulowania mojej legitymacji przy użyciu smartfonu, abym miał możliwość używania telefonu jako zamiennik karty, gdy nie mam jej przy sobie.

- Jako student/doktorant, chciałbym mieć możliwość wyświetlenia numeru karty bibliotecznej w formie kodu kreskowego w telefonie, aby mógł skorzystać z biblioteki bez potrzeby posiadania legitymacji przy sobie.

- Jako student/doktorant – użytkownik aplikacji, chciałbym mieć możliwość usunięcia zapisanej karty z pamięci telefonu, które nie są mi już potrzebne, aby mógł odzyskać wolne miejsce w pamięci telefonu.

- Jako student/doktorant, chciałbym żeby aplikacja sprawdzała podczas odczytu czy dane są podpisane cyfrowo przez osobę do tego upoważnioną, abym miał pewność, że są prawdziwe.

- Jako osoba weryfikująca poprawność emulowanych danych, chciałbym, aby emulowane dane odczytane z aplikacji ELS/ELD nie różniły się niczym od tych, które dostarcza karta, abym mógł je swobodnie odczytywać.

3.2 Specyfikacja wymagań funkcjonalnych

Wymagania funkcjonalne to dogłębny opis wszystkich funkcji aplikacji oraz jej zachowań i odpowiedzi na poszczególne sytuacje. Dzięki rozpatrzeniu wstępnych wytycznych dotyczących zadań jakie miała spełniać aplikacja, ustalono jak dokładnie powinna wyglądać jej interakcja z użytkownikiem oraz jakie funkcjonalności będą dla niego dostępne. Ponadto, zaznajomienie się z tematem pracy pozwoliło na wcześniejsze wykrycie problemów, które można było napotkać w dalszej fazy projektu i zapobiec ich wystąpieniu.

3.2.1 Podstawowe funkcjonalności

Pierwszą i zasadniczą funkcją Elektronicznej Legitymacji Studenckiej (ELS) lub też Elektronicznej Legitymacji Doktoranckiej (ELD) w telefonie, bez której nie ma dostępu do pozostałych, powinna być możliwość odczytu danych z kart elektronicznych przy użyciu interfejsu NFC urządzenia mobilnego.

Po przyłożeniu karty, aplikacja powinna odczytać i wyświetlić odpowiednie informacje w zależności od jej profilu.

Jeśli przyłożona karta zostanie rozpoznana jako obsługiwany typ legitymacji studenckiej lub doktoranta, aplikacja powinna wyświetlić:

- typ legitymacji (legitymacja studencka/doktoranta),
- nazwę uczelni lub jednostki naukowej,
- nazwisko i imiona studenta/doktoranta,
- numer albumu/legitymacji,
- numer edycji (który jest związany z duplikatem/edycją legitymacji o tym samym numerze albumu w zakresie danej uczelni),
- numer PESEL,
- datę ważności legitymacji.

Ponadto, karta powinna być sprawdzona pod kątem zainstalowanych innych aplikacji, np: aplikacji komunikacji miejskiej PEKA oraz posiadania aplikacji jElib[14] (w tym przypadku przyda się opcja wyświetlania numeru karty bibliotecznej oraz tego numeru w postaci kodu kreskowego).

Po udanym odczycie karty, powinna istnieć możliwość zapisu odczytanych z niej danych w pamięci telefonu, aby w każdej chwili można było do nich powrócić bez konieczności ponownego odczytu karty.

Użytkownik powinien mieć dostęp do menadżera kart, w którym to będzie miał możliwość wglądu we wcześniej zapisane karty, a także możliwość ich usunięcia lub też dodania nowych (w razie potrzeby).

Aplikacja powinna sprawdzać, czy dane zapisane na karcie są podpisane podpisem elektronicznym i czy certyfikat znajdujący się na karcie do niego pasuje. Ponadto powinna być możliwość weryfikacji poprawności/ważności użytego podpisu względem CA (oraz uprawnień osoby, która dokonała tego podpisu).

3.2.2 Emulacja

Kluczową funkcją aplikacji w ramach tego projektu jest możliwość emulacji legitymacji ELS/ELD za pomocą telefonu.

W związku z tym, użytkownik (student/doktorant) powinien mieć możliwość wyboru jednej z uprzednio zapisanych kart w pamięci telefonu, której dane powinny być emulowane za pomocą interfejsu NFC urządzenia mobilnego.

Powinny istnieć 3 tryby emulacji karty:

- emulacja tylko podczas otwartego okna aplikacji,
- emulacja na określony okres czasu ustalony przez użytkownika,
- emulacja w trybie ciągłym (również gdy aplikacja jest wyłączona).

Emulacja karty przez interfejs NFC powinna być rozpoznawana przez czytniki kart odpowiednio jako legitymacja studencka/doktoranta. Nie spodziewamy się tutaj błędu ze względu na nośnik danych legitymacji (karta vs telefon).

3.3 Specyfikacja wymagań pozafunkcjonalnych

Oprócz wymagań funkcjonalnych bardzo ważnym aspektem tego projektu są wymagania pozafunkcjonalne. Wymagania te odnoszą się przede wszystkim do interfejsu użytkownika, jego wyglądu i charakterystyki. Ponadto definiują inne aspekty, nie związane bezpośrednio z funkcjonalnością aplikacji, takie jak: docelowe systemy, czy obsługiwane typy kart. Regulują kwestie dotyczące bezpieczeństwa, wydajności i niezawodności produktu. Wymagania pozafunkcjonalne mają na celu dopilnowanie, aby korzystanie z systemu odbywało się w sposób przyjemny, intuicyjny i bezproblemowy dla użytkownika.

Elektroniczna Legitymacja Studencka/Doktoranta w telefonie jest z założenia aplikacją, która w późniejszym etapie rozwoju projektu powinna być zintegrowana z oficjalną aplikacją Politechniki Poznańskiej *PPulse*, dlatego też wygląd oraz kolorystyka ELS/ELD w telefonie, nie powinna odbiegać od tych, które zostały użyte w aplikacji PPulse.

Według początkowych założeń projektu aplikacja miała być zaimplementowana na wszystkie mobilne systemy operacyjne na których jest aplikacja PPulse, tj. system Android, iOS oraz Windows Phone. Jednak iOS nie udostępnia deweloperom aplikacji użytkownika dostępu do protokołu NFC do łączenia się z kartami inteligentnymi. Wersja iOS 11 wprowadziła zmianę pozwalającą na odczytywanie danych z tagów NFC [17], ale nie jest to wystarczające dla potrzeb projektu. Aktualnie nie ma żadnych informacji na temat pełnego udostępnienia NFC w iOS. Z kolei Windows Phone ma niewielki udział w rynku smartfonów z funkcją NFC oraz utracił oficjalne wsparcie Microsoftu w 2017 roku [18]. Ostatecznie przyjęto więc aplikacja powinna działać na smartfonach z systemem Android.

Odczyt oraz emulacja powinny obejmować karty dualne, które zostały spersonalizowane w Międzyuczelnianym Centrum Personalizacji Legitymacji Studenckiej. Możliwość odczytu i emulacji powinna obejmować następujące modele kart:

- Gemalto Optelio (wersje R5, R6 i R7)
- Oberthur Cosmo 7
- Gisecke&Devrient

Rysunek 3.1: Aplikacja PPulse Politechniki Poznańskiej (zrzut ekranu)

Aplikacja powinna być otwarta na zmiany wprowadzane na poziomie nowych typów kart, dzięki trzymaniu informacji na temat struktury danych i sposobu przechowywania danych na kartach w edytowalnym pliku konfiguracyjnym.

3.4 Interfejs użytkownika

Na podstawie zebranych wymagań funkcjonalnych i niefunkcjonalnych udało się zrealizować projekt, który spełnia większość wymagań opisanych powyżej.

Szata graficzna projektu jest zbliżona do kolorystyki użytej w ramach aplikacji *PPulse* Politechniki Poznańskiej. Moduł elektronicznej legitymacji studenckiej/doktoranta na telefonie nie został jeszcze zintegrowany z aplikacją *PPulse*, natomiast funkcjonuje on jako odrębna aplikacja, którą można zainstalować na telefonie z systemem Android i obsługą interfejsu NFC.

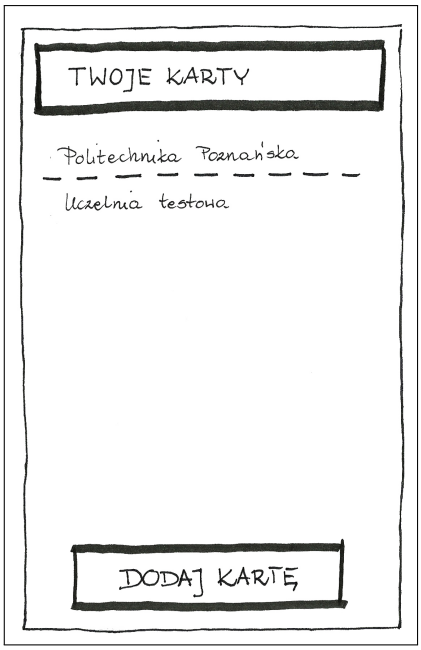
Brak dostępu do NFC zostanie wykryty przez aplikację i zostanie wyświetlony komunikat błędu.

Po zainstalowaniu aplikacji na telefonie z wymaganymi zasobami, zostanie wyświetlony ekran z listą zapisanych legitymacji na telefonie. Na początku jest ona oczywiście pusta.

Jest możliwość zapisania na telefonie więcej niż jednej legitymacji (przy założeniu, że student może równolegle dokształcać się na więcej niż jednej uczelni). Taki przypadek użycia został przedstawiony na poniższym ekranie (Rysunek 3.2).

W przypadku, gdy lista kart (legitymacji) jest pusta, lub też znajdują się już na niej jakieś elementy, jest możliwość dodania kolejnej karty do listy wybierając przycisk *Dodaj kartę* na dole ekranu. Okno z odczytem karty pozostaje puste do czasu udanego odczytu karty. Można tę akcję przerwać, używając podstawowego przycisku powrotu do poprzedniego ekranu. Poprawny odczyt danych z testowej karty (testowej legitymacji studenckiej) został przedstawiony na Rysunku 3.3.

Jeśli z listy kart (legitymacji studenckich/doktoranta) zapisanych na telefonie, wybierzemy jedną z kart, wówczas zostaną wyświetlone szczegółowe dane z tej karty. Rysunek 3.4 przedstawia



Rysunek 3.2: Aplikacja ELS – lista zapisanych kart na telefonie (zrzut ekranu)

Rysunek 3.3: Aplikacja ELS – odczyt danych z legitymacji studenckiej (zrzut ekranu)

widok danych szczegółowych karty dla karty testowej, natomiast kolejny rysunek 3.5 zawiera już dane szczegółowe prawdziwej legitymacji studenckiej. Można zauważyć m.in. brak dodatkowych aplikacji komunikacji miejskiej typu PEKA (*Nie zainstalowano*), brak numeru eLib (*Nie zainstalowano*), pierwszą edycję karty (*A*). Natomiast na kolejnym rysunku 3.5 mam już podgląd na dane użytkownika karty z zainstalowaną aplikacją PEKA, dostępnym numerem eLib, zbliżającym się terminem ważności karty, drugą edycją legitymacji itd.

Po wybraniu karty z listy jest możliwość jej usunięcia. Jest też opcja dodania tej samej ponownie. Niestety aktualny problem jest taki, że 2 legitymacje tej samej uczelni są nierozróżnialne

na liście kart, przez co można omylnie wybrać do emulacji niepoprawną kartę kub usunąć niewłaściwą kartę z telefonu. Nie mniej jednak, nie jest to typowy przypadek użycia, ponieważ nie spodziewamy się, aby student posiadał 2 legitymacje tej samej uczelni.

Rysunek 3.4: Aplikacja ELS – szczegóły dotyczące wybranej legitymacji zapisanej na telefonie (zrzut ekranu – karta testowa)

Rysunek 3.5: Aplikacja ELS – szczegóły dotyczące wybranej legitymacji zapisanej na telefonie (zrzut ekranu – karta PP)

Interfejs użytkownika wydaje się być prosty i intuicyjny. Błędne akcje są rozpoznawane i sygnalizowane użytkownikowi odpowiednimi komunikatami o błędzie.

W przypadku dalszego rozwoju aplikacji można wziąć pod uwagę nowe funkcjonalności takie jak komunikat o zbliżającym się terminie ważności karty, sugerowanie wstąpienia do dziekanatu

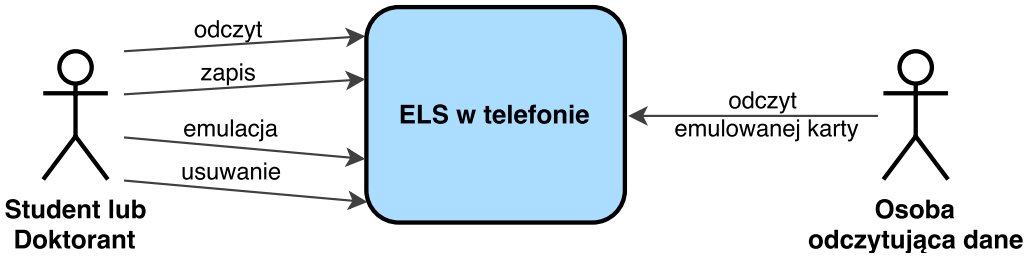
w godzinach pracy, gdyby była dostępna lokalizacja telefonu, aktualny czas i informacje o pracy dziekanatu. Pomysłów na dalszy rozwój tej aplikacji lub też powiązanych aplikacji jest wiele, dlatego też zostaną one zebrane w oddzielnym rozdziale dotyczącym wniosków i zaleceń wdrożeniowych.

Rozdział 4

Modelowanie biznesowe aplikacji

4.1 Aktorzy

Projekt elektronicznej legitymacji studenckiej/doktoranckiej na urządzeniu mobilnym przewiduje zasadniczo dwóch aktorów (Rysunek 4.1). Pierwszy z nich to główny użytkownik systemu, czyli student lub doktorant. Natomiast drugi aktor, to osoba odczytująca lub też weryfikująca dane z emulowanej legitymacji studenckiej/doktoranta.



Rysunek 4.1: Interakcje aktorów z projektowanym systemem

4.2 Obiekty

Projekt systemu przewiduje następujące obiekty, które będą brane pod uwagę w trakcie interakcji aktorów ze systemem:

- **Karta elektroniczna** – Mamy tutaj na myśli nośnik danych w postaci plastikowej karty z wbudowanymi układami scalonymi.
- **Profil karty** – Jest to zbiór aplikacji znajdujących się na karcie elektronicznej, determinujący jej charakter.
- **Aplikacja** – Jest to oprogramowanie znajdujący się na karcie elektronicznej, odpowiadający za jej zachowanie.
- **Urządzenie mobilne** – Rozważamy tutaj przenośne urządzenie elektroniczne działające bez konieczności utrzymywania połączenia z siecią. Służy do przesyłania i odbierania danych.
- **Czytnik** – Jest to urządzenie elektroniczne służące jako podstawowe narzędzie do odczytu danych z kart elektronicznych. W ramach projektu będzie ono mogło odczytać również dane emulowanej legitymacji studenckiej ze smartfonu.

4.3 Przypadki użycia

Podstawowy zestaw przypadków użycia naszego projektu został przeanalizowany dla następujących sytuacji:

- zapis danych z karty w telefonie,
- wyświetlenie danych zapisanej już karty,
- emulacja wybranej karty,
- usunięcie wybranej karty,
- wyświetlenie numeru karty bibliotecznej z wybranej karty w formie kodu kreskowego.

Pełen opis przypadków użycia został przedstawiony poniżej (Tabele 4.1–4.6)

Tablica 4.1: **Przypadek użycia UC01**: Odczyt i wyświetlenie danych z karty

Aktorzy: Student/doktorant
Scenariusz główny 1. Aplikacja wyświetla menu główne z zapisanymi kartami oraz przyciskiem dodania nowej karty 2. Użytkownik wciska przycisk dodania nowej karty 3. Aplikacja wyświetla prośbę o zbliżeniu karty 4. Użytkownik zbliża kartę do telefonu 5. Aplikacja odczytuje dane z karty 6. Aplikacja wyświetla odczytane dane 7. Aplikacja wyświetla informację na temat podpisu cyfrowego
Rozszerzenia 5a Aplikacja nie potrafi odczytać danych z karty 5a1. Aplikacja wyświetla komunikat z przyczyną błędu odczytu 5b W telefonie nie ma certyfikatu pośredniego 5b1. Aplikacja wyświetla okno z informacją o braku certyfikatu pośredniego w telefonie oraz prośbą o umieszczenie go w odpowiednim folderze 5b2. Użytkownik potwierdza zapoznanie się z informacją 5b3. Aplikacja kontynuuje od punktu 6

Tablica 4.2: **Przypadek użycia UC02**: Zapis karty w pamięci telefonu

Warunki początkowe: Aplikacja wyświetla dane odczytanej karty. (UC01) **Aktorzy**: Student/doktorant
Scenariusz główny 1. Aplikacja wyświetla przycisk do zapisu karty 2. Użytkownik wciska przycisk zapisu karty 3. Aplikacja zapisuje kartę w pamięci telefonu
Rozszerzenia 3a Aplikacja nie może zapisać karty 3a1. Aplikacja wyświetla przyczynę odmowy zapisu

Tablica 4.3: **Przypadek użycia UC03**: Wyświetlenie danych zapisanej karty

Aktorzy: Student/doktorant
Scenariusz główny 1. Aplikacja wyświetla menu główne z listą zapisanych kart 2. Użytkownik wybiera kartę, której informacje chce wyświetlić 3. Aplikacja wyświetla informację na temat podpisu cyfrowego 4. Aplikacja wyświetla numer karty bibliotecznej w formie kodu kreskowego

Tablica 4.4: **Przypadek użycia UC04**: Włączenie emulacji danych zapisanej karty

Aktorzy: Student/doktorant, Osoba odczytująca emulowaną legitymację
Warunki początkowe: Aplikacja wyświetla dane zapisanej karty. (UC03)
Scenariusz główny 1. Aplikacja wyświetla przycisk emulacji karty 2. Użytkownik wciska przycisk emulacji karty 3. Aplikacja emuluje dane karty

Tablica 4.5: **Przypadek użycia UC05**: Wyłączenie emulacji danych zapisanej karty

Aktorzy: Student/doktorant
Warunki początkowe: Aplikacja emuluje dane zapisanej karty. (UC04)
Scenariusz główny 1. Aplikacja wyświetla przycisk wyłączenia emulacji karty 2. Użytkownik wciska przycisk wyłączenie emulacji karty 3. Aplikacja przestaje emulować kartę

Tablica 4.6: **Przypadek użycia UC06**: Usunięcie zapisanej karty z pamięci telefonu

Aktorzy: Student/doktorant
Warunki początkowe: Aplikacja wyświetla dane zapisanej karty. (UC03)
Scenariusz główny 1. Aplikacja wyświetla przycisk usunięcia karty 2. Użytkownik naciska przycisk usunięcia karty 3. Aplikacja usuwa kartę z pamięci telefonu 4. Aplikacja wraca do menu głównego

4.4 Problemy i koncepcja ich rozwiązania

Podczas analizy wymagań funkcjonalnych i niefunkcjonalnych, doboru narzędzi, rozwoju oprogramowania i jego testowania napotkaliśmy na następujące problemy. Niektóre z nich udało się wcześnie wykryć już na etapie planowania i uniknąć kosztownych czasowo zmian w projekcie.

Jednym z podstawowych problemów podczas tworzenia aplikacji była kwestia sposobu odczytu danych z kart niezależnie od istniejących już rodzajów kart oraz tych wprowadzanych w najbliższej przyszłości. Pod uwagę należy wziąć także ciągły rozwój technologiczny. W bardzo krótkim czasie można spodziewać się nowych, lepszych i szybszych systemów dostosowanych do rosnących potrzeb użytkowników. Aktualnie na rynku istnieje już wiele typów kart elektronicznych, a każda z nich może posiadać swój własny sposób implementacji dostępu do danych. Mamy tutaj na myśli m.in. komendę SELECT, która wymaga innych parametrów P1 i P2 w zależności od rodzaju karty, aby odpytać kartę o ten sam zakres danych. Karty w odpowiedzi na te zapytania mogą zwracać także

odmienne zakresy danych, np. bloki zamiast bajtów.

Należy pamiętać, że cały czas powstają nowe rodzaje kart, a istniejące obecnie i powszechnie używane rodzaje kart mogą zostać zastąpione nowymi. Rodzi to problem konieczności dostosowywania projektu do potencjalnych zmian wprowadzanych na poziomie kart w późniejszym terminie. Bez tego, dodanie obsługi nowych kart wiązałaby się z koniecznością modyfikacji kodu źródłowego programu oraz jego rekompilacji w celu utrzymania stabilności i aktualności systemu. Aby rozwiązać ten problem, informacje na temat aktualnego sposobu przechowywania danych dla poszczególnych rodzajów kart zostały wyciągnięte poza kod źródłowy aplikacji do zewnętrznych plików konfiguracyjnych typu *.xml. Zgodnie z obranym podejściem podczas próby odczytu jakiejkolwiek karty, w pierwszej kolejności nastąpi określenie typu odczytywanej karty, a następnie w drugim kroku odczytu obsługiwanych konfiguracji z pliku *.xml. Na tym etapie zostaną wyciągnięte informacje na temat sposobu przechowywania danych na karcie (formatu danych) i aplikacja będzie w stanie przystąpić do odczytu interesujących nas danych z karty. Dzięki temu prostemu i powszechnie stosowanemu podejściu będziemy mogli obsługiwać nowe rodzaje kart w trakcie pracy systemu bez konieczności zmian w kodzie źródłowym aplikacji.

Kolejnym problemem, jaki należy rozwiązać jest utrudniony dostęp do plików na urządzeniu mobilnym. Należy pamiętać, że zwykły użytkownik nie ma uprawnień do wyświetlania, ani edytowania plików systemowych. Tak więc rozważany plik konfiguracyjny musi być ulokowane w katalogu "Dokumenty", który jest dostępny dla użytkownika z poziomu urządzenia. Bez tego, (pośrednio) użytkownik nie miałby możliwości zmiany konfiguracji aplikacji, co jest konieczne dla utrzymania systemu w przyszłości.

Rozdział 5

Projekt i implementacja aplikacji

5.1 Opis architektury

Aplikacja podzielona jest na dwa projekty: projekt współdzielony (PCL) oraz projekt na platformę Android. Podział powstał ze względu na możliwą przyszłą implementację projektu na innych platformach. Z tego powodu aplikacja została tak zaprojektowana, aby jak największa część kodu znalazła się w części współdzielonej.

Diagram ważniejszych klas aplikacji klas jest przedstawiony na rysunku 5.1. Wszystkie klasy w diagramie klas z wyjątkiem klas z przestrzeni nazw Droid są z projektu PCL. Przestrzeń nazw Services zawiera interfejsy, które pozwalają projektowi współdzielonemu na dostęp do funkcjonalności zależnych od platformy. Przestrzeń nazw ViewModel zawiera klasy będące modelami widoku ekranów aplikacji. W przestrzeni nazw Model znajdują się klasy będące modelami do komunikacji pomiędzy różnymi częściami aplikacji. Przestrzeń nazw Helper zawiera klasy będące jądrem aplikacji oraz klasy pomocnicze.

Aby zacząć oczekiwanie telefonu na odczyt karty wywoływana jest metoda INFCReader.StartReading(). Aby wyłączyć oczekiwanie wywoływana jest metoda INFCReader.StopReading().

Kiedy karta zostanie połączona z telefonem, należy wywołać metodę CardReadingController.ProcessConnectedCard(). Jako argumenty przekazywane są bajty historyczne (służące do rozpoznania modelu karty) oraz obiekt klasy implementującej interfejs IConnectedCard. Metoda SendCommand tego interfejsu służy do wysyłania komend APDU do karty. Metoda zwraca odpowiedź APDU. (Klasa implementująca interfejs IConnectedCard znajduje się w projekcie zależnym od platformy.) Metoda ProcessConnectedCard pobiera typ (model) karty z metody CardTypeSelector::FromHistoricalBytes, a następnie przekazuje ten typ i obiekt typu IConnectedCard do obiektu typu CardReader, dzięki któremu pobiera zawartość plików z karty.

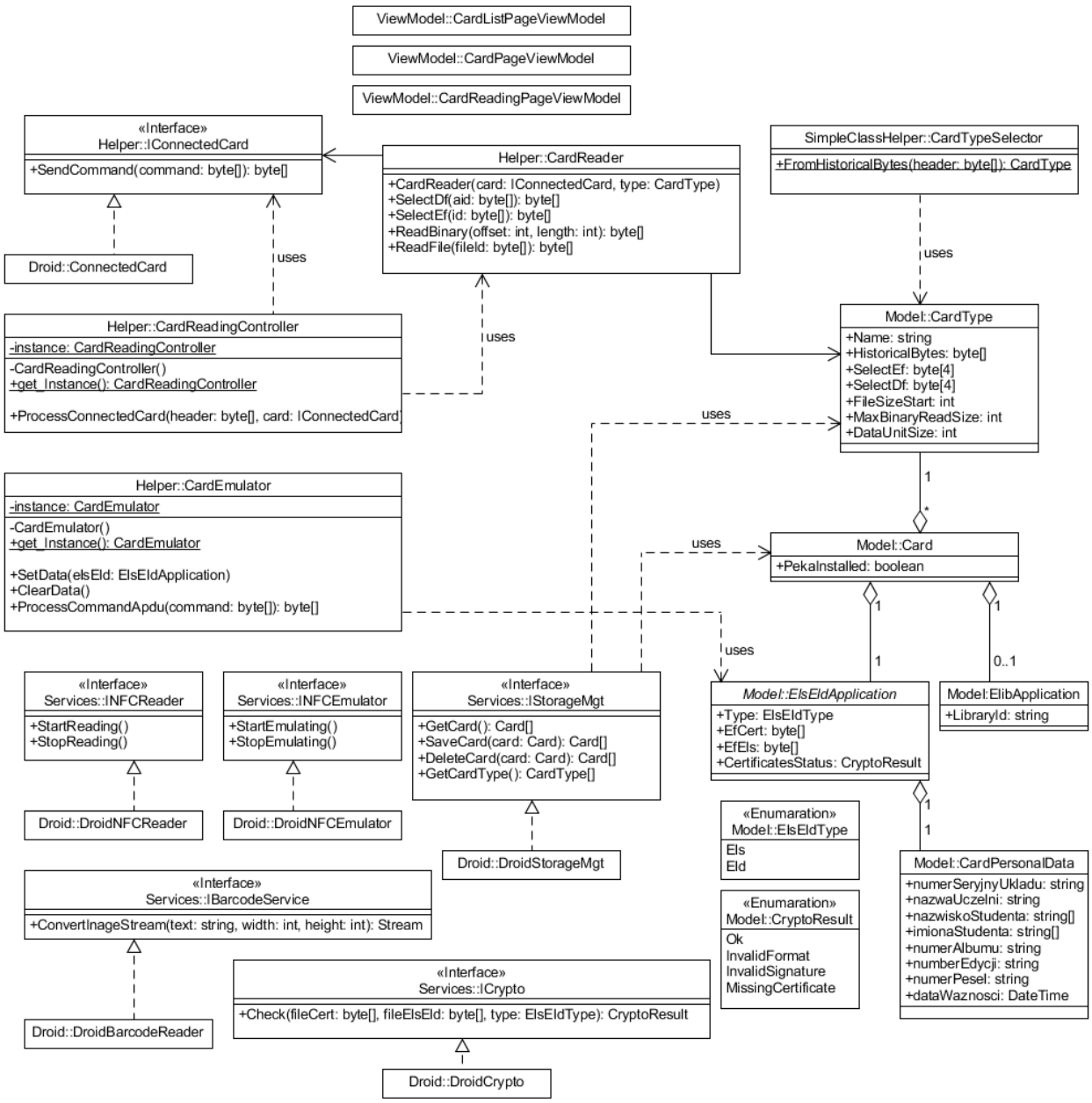
Analogicznie do odczytu emulacja jest włączana i wyłączana przez metody INFCEmulator.

Kiedy telefon zostanie zbliżony do czytnika i otrzyma komendę APDU, ta komenda jest przekazywana do metody CardEmulator.ProcessCommandApdu(), która zwraca odpowiedź do wysłania. Metody CardEmulator.SetData() i CardEmulator.ClearData() są wykorzystywane do ustalenia karty do emulowania.

5.2 Implementacja

5.2.1 Narzędzia

Użyto systemu zarządzania projektami GitLab. Podczas programowania korzystano ze środowiska Microsoft Visual Studio 2017 oraz platformy deweloperskiej Xamarin Forms, umożliwiającej



Rysunek 5.1: Diagram klas zaprojektowany na potrzeby odczytu danych z legitymacji oraz jej emulacji

tworzenie natywnych aplikacji mobilnych dla systemów iOS, Android i Windows. Testy zostały przeprowadzone na urządzeniach z systemem Android np. Samsung Galaxy Note 3 oraz czytniku kart Omnikey CardMan 5121.

5.2.2 Technologia

Aplikacja została zaimplementowana w języku programowania C# oraz frameworku .NET w wersji 4.5. Użycie tej technologii oraz środowiska pozwoliło na stworzenie jednego projektu, który może być uruchomiony na urządzeniach mobilnych z różnymi systemami operacyjnymi. Pliki konfiguracyjne oraz dane są sformatowane do struktury JSON i zapisywane w pamięci telefonu w formacie tekstowym.

W projekcie zostały wykorzystane następujące biblioteki:

Asn1Net.Reader W celu odczytu danych ASN.1 w kodowaniu BER została zastosowana biblioteka Asn1Net.Reader. Jest ona licencjonowana pod Apache License 2.0. Z tego powodu wymagane jest umieszczenie w projekcie kopii tej licencji.

Problemem z tą biblioteką było to, że nie jest ona w stanie przetworzyć strumieni danych, które zawierają więcej niż tylko jedną strukturę ANS.1. Tak jest w przypadku plików EF.CERT i EF.SELS/EF.SELD na niektórych modelach kart, które nie pozwalają na zmianę rozmiaru pliku. W takich sytuacjach karta zawiera pliki o stałym rozmiarze, większym od przechowywanych danych. Zastosowane rozwiązanie polega na wykorzystaniu biblioteki najpierw do odczytania pierwszego nagłówka. Następnie plik jest ograniczany do rozmiaru zależnego od nagłówka i biblioteka jest stosowana po raz drugi, do przetworzenia wszystkich danych.

Android.Nfc.CardEmulators.HostApduService HostApduService to klasa usługi z platformy Xamarin Forms, którą można rozszerzyć, aby emulować zapisaną kartę korzystając z interfejsu NFC. Usługę należy zadeklarować w pliku XML, w którym znajdują się również identyfikatory aplikacji (ang. *Application Identifier – AID*), które obsługuje. W trakcie działania aplikacji najpierw zostaje wybrany odpowiedni AID metody APDU, a następnie jest on przekazany do funkcji HostApduService.ProcessCommandApdu(Byte[],Bundle).

5.2.3 Modele kart - plik konfiguracyjny

W celu obsługi różnych modeli kart stosowany jest plik konfiguracyjny. Niektóre modele wymagają do działania odpowiedniego wpisu w tym pliku, a niektóre działają na wartościach domyślnych. Plik konfiguracyjny może być dostarczony przez użytkownika. Aplikacja szuka go w folderze pobranych plików (`Downloads`) na ścieżce `PUT_CardTypes/CardTypeConfig.txt`. Folder pobranych plików został wybrany jako miejsce pobierania pliku od użytkownika, ponieważ jest to miejsce do którego użytkownik i aplikacja mają łatwy, bezproblemowy dostęp. W przypadku braku pliku w danej lokalizacji wykorzystywany jest domyślny plik osadzony w aplikacji.

Plik jest w formacie JSON i zawiera tablicę obiektów reprezentujących modele kart. Przykładowy plik konfiguracyjny jest przedstawiony na listingu 5.1. Obiekt modelu karty może zawierać następujące pola:

- $Name$: Nazwa modelu karty – Niewykorzystywana.
- $HistoricalBytes$: Bajty historyczne – Wymagane aby rozpoznać model karty z ATR.
- $FileSizeStart$: Początek rozmiaru pliku – Obecność tego pola oznacza, że dla danego modelu rozmiar pliku w odpowiedzi na wybór pliku znajduje się na bajtach $FileSizeStart$ i $FileSizeStart + 1$. Brak tego pola oznacza obecność rozmiaru pliku zgodną ze standardem [2].
- $MaxBinaryReadSize$: Maksymalny rozmiar danych odczytywanych w pojedynczym poleceniu BINARY READ. Brak tego pola oznacza używanie domyślnej wartości 230.
- $DataUnitSize$: Rozmiar jednostki danych. Brak tego pola oznacza używanie domyślnej wartości 1.
- $SelectEf$: Wybór pliku dedykowanego – Tablica 4 bajtów oznaczająca nagłówek polecenia SELECT do wyboru aplikacji.

Listing 5.1: Przykładowy plik konfiguracyjny

```
[
   {
      "Name":"Gemalto Optelio R5",
      "HistoricalBytes":[
         0x80,
         0x65,
         0xA2,
         0x01,
         0x01,
         0x01,
         0x3D,
         0x72,
         0xD6,
         0x41
      ],
      "FileSizeStart":"8",
      "MaxBinaryReadSize":"248",
      "DataUnitSize":"4",
      "SelectEf":[
         0x00,
         0xA4,
         0x02,
         0x00
      ],
      "SelectDf":[
         0x00,
         0xA4,
         0x04,
         0x00
      ]
   }
]
```

– *SelectDf*: Wybór pliku elementarnego – Tablica 4 bajtów oznaczająca nagłówek polecenia SELECT do wyboru pliku.

5.2.4 Odczyt danych z kart

Kiedy telefon znajduje się w trybie odczytu i wykryje obecność karty uruchamiana jest procedura odczytu. Na podstawie bajtów historycznych rozpoznawany jest model karty. Próbie odczytu podlegają następujące aplikacje na karcie:

– Elektroniczna Legitymacja Studencka,

– Elektroniczna Legitymacja Doktoranta,

– Aplikacja biblioteczna jElib,

– Poznańska Elektroniczna Karta Aglomeracyjna.

Wybór aplikacji odbywa się poprzez polecenie SELECT. Nagłówek polecenia jest wybierany w zależności od modelu karty. Jeżeli dany model karty nie został rozpoznany lub wykonanie polecenia z danym nagłówkiem nie zakończyło się sukcesem próbowane jest powtórne wykonanie polecenia z nagłówkami domyślnymi, które zostały przedstawione w tablicy 5.1.

Tablica 5.1: Domyślne nagłówki polecenia SELECT do wyboru aplikacji

CLA	INS	P1	P2
00	A4	04	00
00	A4	04	04

Wybór pliku w aplikacji podobnie odbywa się poprzez polecenie SELECT z nagłówkiem zależnym od modelu karty lub nagłówkiem domyślnym. Nagłówki domyślne do wyboru pliku zostały przedstawione w tablicy 5.2.

Tablica 5.2: Domyślne nagłówki polecenia SELECT do wyboru pliku

CLA	INS	P1	P2
00	A4	02	00
00	A4	02	04

W celu odczytu pliku z karty wymagana jest znajomość wielkości pliku oraz jednostki danych. Wielkość pliku odczytywana jest z odpowiedzi karty na polecenie wyboru pliku. Domyślnie wielkość pliku znajduje się bezpośrednio w parametrze o tagu 80. Jednak niektóre modele kart zawierają wielkość pliku w parametrze o tagu 85 oznaczającym informacje zastrzeżone. W takim przypadku, aby pozyskać wielkość pliku wymagane jest przypisanie dla konkretnego modelu karty informacji, na których bajtach odpowiedzi na wybór pliku znajduje się wielkość pliku. Wielkość jednostki danych jest domyślnie zakładana jako 1 bajt. Jeżeli jest inna należy ją przypisać do modelu karty. Sam odczyt pliku wykonywany jest poprzez wykonywanie polecenia READ BINARY do odczytu kolejnych fragmentów pliku. Maksymalna wielkość odczytywanych danych w jednym poleceniu jest zależna od modelu karty. W przypadku braku tych informacji używana jest domyślna wartość 230 (dziesiętnie), która jest odpowiednia dla wielu różnych modeli.

Odczyt ELS lub ELD zaczyna się poprzez wybór aplikacji ELS. Jeżeli jej nie ma wykonywany jest wybór aplikacji ELD. Po pomyślnym wyborze którejś z tych aplikacji procedura odczytu jest podobna. Odczytywane są pliki EF.ELS/EF.ELD oraz EF.CERT. Z pliku EF.ELS/EF.ELD odczytywane są dane zgodnie z rozporządzeniami [9, 10]. Sprawdzana jest także poprawność certyfikatu w pliku EF.CERT oraz jego zgodność z plikiem EF.ELS/EF.ELD.

Z aplikacji jElib odczytywany jest identyfikator czytelnika Poznańskiej Fundacji Bibliotek Naukowych, który znajduje się pod identyfikatorem biblioteki "PFBN". Do tego odczytu używane jest polecenie aplikacji jElib GET ENTRY VALUE [11].

Sprawdzenie czy jest zainstalowana aplikacja PEKA na karcie odbywa się poprzez wybór tej aplikacji. Zakończenie się tego polecenia sukcesem oznacza obecność aplikacji PEKA.

5.2.5 Sprawdzanie certyfikatów

Plik EF.CERT znajdujący się w aplikacji ELS/ELD zawiera certyfikat X.509 dla osoby upoważnionej do wystawiania legitymacji studenckiej/doktoranta. Poprawność tego certyfikatu jest sprawdzana poprzez wykorzystanie klasy X509Certificate2 platformy .NET. Ze względu na to, że ta klasa nie jest dostępna na wszystkich platformach (nie ma jej w Windows Phone), to implementacja sprawdzania wszystkich rzeczy związanych z kryptologią odbywa się w projekcie Androidowym.

Możliwym sposobem implementacji kryptologii, po połączeniu aplikacji z PPulse jest zamiana głównego projektu z PCL do .NET Standard. Istnieją i są tworzone biblioteki kryptologiczne w .NET Standard, które pozwoliłyby na przeniesienie tego kodu do części współdzielonej. Wówczas

nie byłyby potrzeby powielania kodu dla Androida, Windows Phone, itd. oraz możliwe byłoby skorzystanie z gotowego rozwiązania do analizy pliku w formacie SignedData. Jako że aplikacja PPulse wykonana jest w PCL i nie jest możliwe odwoływanie się do projektu .NET Standard z PCL, to zastosowanie tego rozwiązania wymagałoby zmiany projektu PPulse z PCL na .Net Standard. Z tego powodu w aplikacji nie zdecydowano się na skorzystanie z tego rozwiązania.

Sprawdzanie poprawności certyfikatu EF.CERT wymaga posiadania wszystkich certyfikatów na ścieżce certyfikacji. Dla poprawności certyfikatu wymagane jest aby główny certyfikat Urzędu Certyfikacji (Root CA) pochodził z Narodowego Centrum Certyfikacji (NCCert). Aby zapewnić autentyczność głównego certyfikatu umieszczany od jest na stałe w aplikacji. Gdy zostanie wydany nowy certyfikacji NCCert planowane jest wystawienie nowej wersji aplikacji. Jako że te certyfikaty mają ważność rzędu wielu lat, nie powinno to stanowić większego problemu.

Problemem natomiast jest to, skąd pobrać certyfikaty pośrednich Urzędów Certyfikacji (Intermidiate CA). Ten certyfikat może pochodzić z niezaufanego źródła, ponieważ jego autentyczność można sprawdzić dzięki posiadanemu certyfikatowi NCCert. Lista tych urządów zmienia się na bieżąco, więc nie jest możliwe pełne rozpowszechnianie ich razem z aplikacją. Zastosowano rozwiązanie polegające na tym, że certyfikat może znajdować się w jednym z dwóch miejsc. Po pierwsze certyfikat może być w aplikacji. Po drugie może być on dostarczony przez użytkownika do odpowiedniego folderu za telefonie. Gdy aplikacja nie posiada odpowiedniego certyfikatu, to informuje o tym użytkownika z prośbą o umieszczenie certyfikatu w folderze.

Ze względu na brak obsługi rozszerzenia certyfikatu zawierającego listę unieważnionych certyfikatów (CRL) oraz brak możliwości podania własnej listy CRL do klas platformy .NET, nie jest możliwe jej wykorzystanie do analizy CRL. Ze względu na to, że implementacja własnej obsługi CRL wykracza poza zakres pracy, nie jest ona sprawdzana w aplikacji.

Plik EF.ELS/EF.ELD zawiera dane osobowe posiadacza karty w strukturze SignedData (zgodnie z PKCS #7). Jako że wersja platformy .NET na system Android nie posiada implementacji klasy SignedData, to zastosowano własną implementację analizy struktury SignedData.

W certyfikacie sprawdzane jest w atrybucie „nazwa powszechna” obecność tekstu „osoba upoważniona do wystawiania legitymacji studenckiej” lub odpowiednio „osoba upoważniona do wystawiania legitymacji doktoranta”.

5.2.6 Zapis danych w pamięci telefonu

Zapis kart odbywa się w pliku `cards.txt` w prywatnym folderze aplikacji, dzięki czemu użytkownik nie ma bezpośredniego dostępu do tego pliku. Dla każdej karty zapisywane są pliki EF.CERT i EF.ELS / EF.ELD, ponieważ są one bezpośrednio używane podczas emulacji. Ponadto zapisywane są wszystkie dane wyświetlane dla użytkownika. Lista zapisanych kart jest serializowana w formacie JSON.

5.2.7 Emulacja kart

W trybie emulacji telefon z interfejsem NFC pełni funkcję karty elektronicznej. W systemie Android usługa emulacji NFC jest powiązana z identyfikatorami aplikacji (ang. *Application Identifier – AID*), które obsługuje. Polecenie wyboru aplikacji i następujące po nim polecenia są przekazywane do usługi ustawionej jako domyślna dla wybranego AID. Zadaniem usługi jest przygotowanie odpowiedzi APDU, która zostanie następnie wysłana do terminala.

Emulowana jest wyłącznie aplikacja ELS lub ELD w zależności od rodzaju legitymacji. Rozszerzenie emulacji o inne aplikacje – na przykład aplikację biblioteczną jElib – jest omówione w podrozdziale 7.2.

Obsługiwany jest minimalny zbiór poleceń potrzebny do realizacji funkcji aplikacji ELS.

Bajt klasy

Jedyna obsługiwana wartość bajtu CLA to 00 (wszystkie podane dalej wartości liczbowe przedstawione są w zapisie szesnastkowym). Wartość ta oznacza następujące własności:

- polecenie jest ostatnim lub jedynym poleceniem łańcucha,
- brak zabezpieczenia danych (ang. *secure messaging*),
- domyślny kanał logiczny.

Polecenie o innej wartości CLA powoduje wysłanie odpowiedzi ze statusem 6800.

Polecenie SELECT

Poprawne wykonanie polecenia SELECT powoduje wybranie pliku wskazanego w polu danych. Kolejne polecenia odczytu będą dotyczyły wybranego pliku. Aby wybrać plik dedykowany, pole danych powinno zawierać identyfikator aplikacji. Wybór pliku elementarnego wymaga podania dwubajtowego identyfikatora pliku. Krótkie identyfikatory plików nie są rozpoznawane przez aplikację.

Opis struktury APDU polecenia SELECT oraz obsługiwane wartości parametrów przestawia tablica 5.3. Bajty statusu SW1 i SW2 informują o rezultacie wykonania polecenia. W tablicy 5.4 zostały wyszczególnione zwracane bajty statusu w zależności od przebiegu wykonania polecenia.

Generowana odpowiedź zawiera strukturę FCI wybranego pliku, gdy parametr P2 wskazuje na żądanie wysłania tej struktury. W innym przypadku odpowiedź zawiera tylko bajty statusu.

Tablica 5.3: Obsługiwane parametry polecenia SELECT

Pole	Wartość	Znaczenie
CLA	00	Patrz 5.5.1
INS	A4	Polecenie SELECT
P1	04	Wybór DF według nazwy
	00	Wybór DF lub EF
	02	Wybór EF pod bieżącym DF
P2	00	Pole danych odpowiedzi jest strukturą FCI
	0C	Brak pola danych w odpowiedzi
L_c	01–FF	Długość pola danych polecenia
pole danych		Nazwa DF lub EF
L_e	00–FF	Długość pola danych odpowiedzi, pole pominięte dla P2 = 0C

Polecenie READ BINARY

Polecenie READ BINARY umożliwia odczyt danych pliku elementarnego. Plik musi zostać wcześniej wybrany poleceniem SELECT. Struktura APDU dla polecenia READ BINARY jest przedstawiona w tablicy 5.5. Obsługiwana jest tylko wersja polecenia, w której bajt INS ma wartość B0.

Tablica 5.4: Odpowiedzi na polecenie SELECT

SW1–SW2	Znaczenie według normy ISO-/IEC 7816-4	Przyczyna	Pole danych odpowiedzi
6400	Błąd przetwarzania	Brak danych karty	-
6700	Nieprawidłowa długość APDU	Brak wymaganych pól CLA, INS, P1, P2	-
		Długość pola danych niezgodna z L_c	-
6800	Nieobsługiwana funkcja w bajcie CLA	Bajt CLA ma wartość różną od 00	-
6A87	Długość pola danych niezgodna z P1–P2	Brak pola danych	-
6A81	Funkcja P1–P2 nieobsługiwana	P1 nie należy do {00, 02, 04}	-
		P2 nie należy do {00, 0C}	-
6A82	Nie znaleziono pliku lub aplikacji	Wskazany plik nie istnieje	-
		Emulacja karty jest wyłączona	-
6D00	Nieobsługiwana instrukcja	Bajt INS nie należy do {A4, B0}	-
9000	Sukces	Plik został wybrany	FCI/-

Polecenie może wymagać odczytu jedynie fragmentu pliku. Offset oznacza adres początkowy fragmentu i jest kodowany przez bajty P1–P2. Długość fragmentu danych jest wyznaczana na podstawie pola L_e.

Odpowiedzi na polecenie READ BINARY zwracane przez aplikację są zawarte w tablicy 5.6. Status 6282 jest wysyłany w przypadku, gdy część żądanego fragmentu wykracza poza koniec pliku. Odpowiedź zawiera wówczas maksymalną liczbę bajtów możliwą do odczytania.

Tablica 5.5: Obsługiwane parametry polecenia READ BINARY

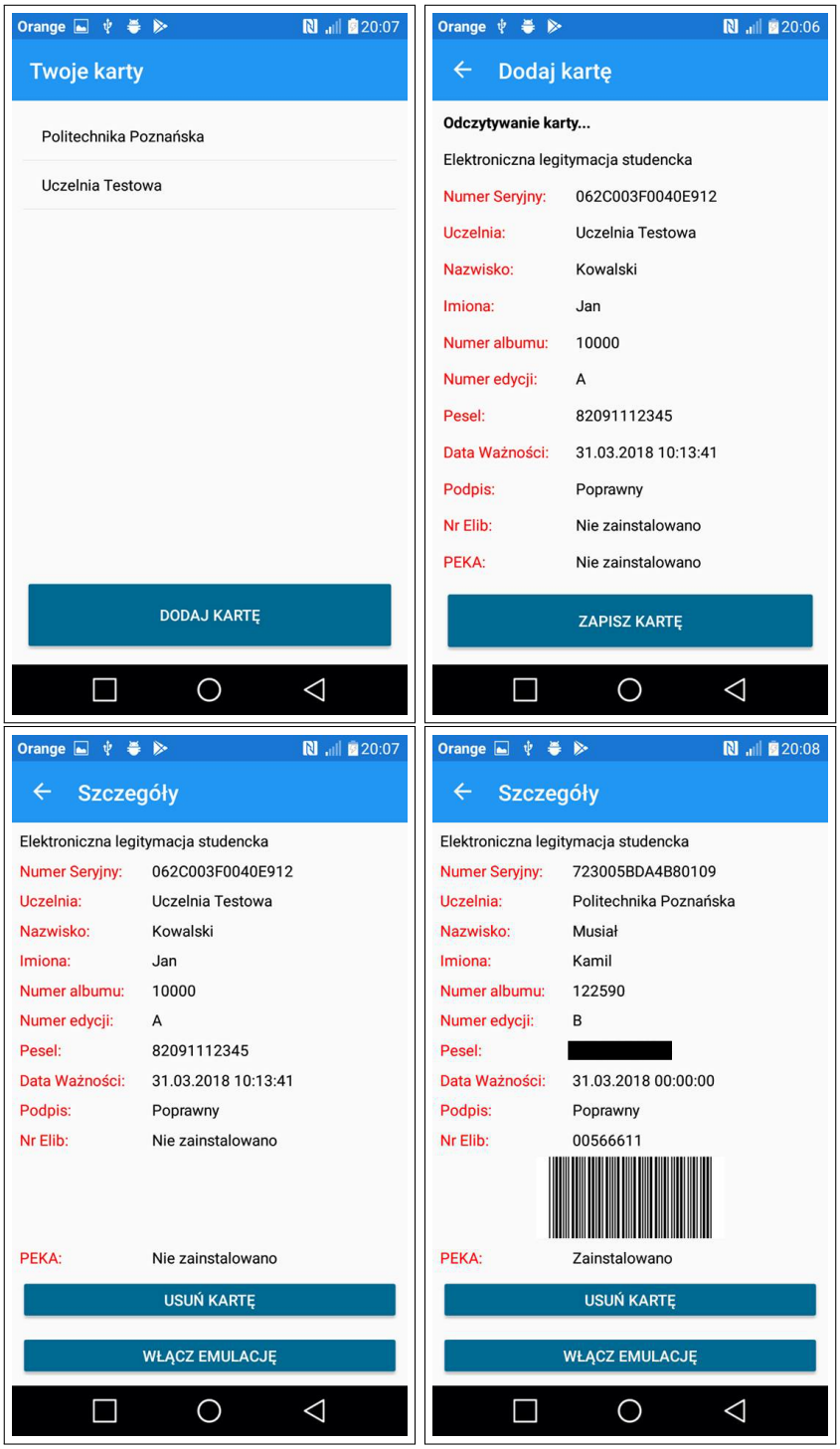
Pole	Wartość	Znaczenie
CLA	00	Patrz 5.5.1
INS	B0	Polecenie READ BINARY
P1–P2	0000–7FFF	Offset
L_e	00–FF	Długość pola danych odpowiedzi

Tablica 5.6: Odpowiedzi na polecenie READ BINARY

SW1–SW2	Znaczenie	Przyczyna	Pole danych odpowiedzi
6400	Błąd przetwarzania	Brak danych karty	-
6700	Nieprawidłowa długość APDU	Brak wymaganych pól CLA, INS, P1, P2	-
		Długość pola danych niezgodna z L_c	-
6800	Nieobsługiwana funkcja w bajcie CLA	Bajt CLA ma wartość różną od 00	-
6986	Brak wybranego pliku	Brak wybranego pliku	-
		Wybrany plik nie jest EF	-
6282	Koniec pliku	Nie odczytano zadanej liczby bajtów	Fragment danych pliku
6B00	Offset za końcem pliku	Offset jest większy niż rozmiar pliku	-
6D00	Nieobsługiwana instrukcja	Bajt INS nie należy do {A4, B0}	-
9000	Sukces	Odczytano zadaną liczbę bajtów	Fragment danych pliku

5.2.8 Interfejs użytkownika

Poniżej przedstawiono wygląd kolejnych ekranów aplikacji mobilnej.



Rysunek 5.2: Aplikacja ELS – zrzuty ekranu

5.3 Testy

Podczas tworzenia aplikacji konieczne jest przeprowadzenie odpowiedniej liczby testów, żeby zapewnić jakość i niezawodność końcowego produktu. Testowanie zostało oparte na testach funkcjonalnych sprawdzających poprawność wszystkich elementów oprogramowania. Aplikacja została przetestowana na kilku różnych urządzeniach mobilnych z systemem Android, aby sprawdzić czy wszystkie funkcje działają poprawnie niezależnie od telefonu oraz wersji sytemu operacyjnego.

Podczas testów na światło dzienne wyszły problemy, które nie zostały zidentyfikowane na etapie projektowania aplikacji. Jeden z takich problemów dotyczył trybów działania interfejsu NFC. Zaobserwowano, że po włączeniu emulacji legitymacji na telefonie, czytnik okazjonalnie nie reagował na zbliżanie telefonu z emulowaną legitymacją. Tylko w kilku przypadkach następowało nawiązanie połączenia. Okazało się, że przyczyną tego zachowania był włączony tryb odczytu NFC na telefonie. Błąd został rozwiązany przez automatyczne wyłączenie odczytu NFC po opuszczeniu ekranu dodawania karty. Był to jeden z trudniejszych do przewidzenia i zdiagnozowania problem, który spowodował znaczące opóźnienie dalszych testów aplikacji.

Oprócz testowania, przez programistów wykonywane są wzajemne inspekcje, które pomagają sprawdzić poprawność fragmentów kodu i przyspieszają rozwiązanie trudnych problemów.

5.3.1 Testy akceptacyjne

Testy akceptacyjne przeprowadzane są po to, by określić, czy stworzone elementy systemu spełniają początkowe oczekiwania. Z tego względu ich przeprowadzeniem również powinny zajmować się osoby niezwiązane z implementacją danej funkcjonalności, a najlepiej osoby niezwiązane z projektem. W tym celu ułatwienia testów akceptacyjnych przygotowane zostały przypadki i scenariusze testowe.

Przypadki testowe

Tablica 5.7: P1: Wyświetlenie danych odczytanej karty

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa przypadku testowego	Odczyt	**Sygnatura przypadku testowego**	P1
Testowana funkcjonalność	Wyświetlenie danych odczytanej karty		
Sposób dostępu	W menu głównym wybranie opcji dodania nowej karty i zbliżenie karty do telefonu		
Dane wejściowe	ELS/ELD zbliżona do telefonu		
Dane wyjściowe	Wyświetlone dane z pamięci ELS/ELD oraz informacja o podpisie cyfrowym		

Tablica 5.8: P2: Zapis odczytanych danych z karty do telefonu

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa przypadku testowego	Zapis	**Sygnatura przypadku testowego**	P2
Testowana funkcjonalność	Zapisz odczytanych danych z karty do telefonu		
Sposób dostępu	Użytkownik zalogowany do ELS/ELD		
Dane wejściowe	Dane ELS/ELD odczytane z karty		
Dane wyjściowe	Dane ELS/ELD zapisane w pamięci telefonu		

Tablica 5.9: P3: Emulacja wybranej karty

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa przypadku testowego	Emulacja	**Sygnatura przypadku testowego**	P3
Testowana funkcjonalność	Emulacja wybranej karty		
Sposób dostępu	Użytkownik zalogowany do ELS/ELD		
Dane wejściowe	Dane o ELS/ELD zapisane w pamięci telefonu		
Dane wyjściowe	Funkcjonalność ELS/ELD dostępna przez użycie aplikacji emulującej		

Tablica 5.10: Usunięcie wybranej karty

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa przypadku testowego	Usuwanie	**Sygnatura przypadku testowego**	P4
Testowana funkcjonalność	Usunięcie wybranej karty		
Sposób dostępu	Użytkownik zalogowany do ELS/ELD		
Dane wejściowe	Dane o ELS/ELD zapisane w pamięci telefonu		
Dane wyjściowe	Brak danych o ELS/ELD zapisanych w pamięci telefonu		

Tablica 5.11: Odczyt danych karty za pomocą czytnika

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa przypadku testowego	Odczyt za pomocą czytnika	**Sygnatura przypadku testowego**	P5
Testowana funkcjonalność	Odczyt danych karty za pomocą czytnika		
Sposób dostępu	Użytkownik zalogowany do ELS/ELD		
Dane wejściowe	Karta ELS / ELD lub emulacja ELS / ELD za pomocą aplikacji		
Dane wyjściowe	Dane o ELS / ELD odczytane za pomocą czytnika kart		

Scenariusze testowe

Tablica 5.12: Odczyt emulowanej karty czytnikiem kart

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa scenariusza testowego	Odczyt czytnikiem kart	**Sygnatura scenariusza testowego**	S1
Testowana funkcjonalność	Odczyt emulowanej karty czytnikiem kart		
Sposób dostępu	Zgodnie z przypadkami testowymi		
Sygnatury kolejnych przypadków testowych	P3 P5		

Tablica 5.13: Odczyt emulowanej karty za pomocą aplikacji

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa scenariusza testowego	Odczyt emulowanej karty za pomocą aplikacji	**Sygnatura scenariusza testowego**	S2
Testowana funkcjonalność	Odczyt emulowanej karty za pomocą aplikacji		
Sposób dostępu	Zgodnie z przypadkami testowymi		
Sygnatury kolejnych przypadków testowych	P3 P1		

Tablica 5.14: Cykl życia zapisanych danych

Nazwa systemu/modułu	ELS/ELD	**Wersja systemu/modułu**	TODO
Nazwa scenariusza testowego	Cykl życia	**Sygnatura scenariusza testowego**	S2
Testowana funkcjonalność	Cykl życia zapisanych danych		
Sposób dostępu	Zgodnie z przypadkami testowymi		
Sygnatury kolejnych przypadków testowych	P1 P2 P4		

5.4 Dokumentacja

Przy pomocy programu Visual Studio, na podstawie komentarzy utworzonych do kodu źródłowego w projekcie, udało się wygenerować plik XML zawierający opis klas, metod oraz zmiennych użytych podczas pisania aplikacji. W celu ułatwienia jego odczytu wykorzystano program Sandcastle Help File Builder, który z pliku XML wygenerował przejrzystą oraz interaktywną dokumentację w formacie HTML, która została umieszczona jako dodatek na płycie CD dołączonej do pracy inżynierskiej.

Rozdział 6

Zalecenia wdrożeniowe

6.1 Plan wdrożenia oprogramowania

Celem pracy było przygotowanie funkcjonalności związanej z Elektroniczną Legitymacją Studencką i Elektroniczną Legitymacją Doktoranta dla aplikacji mobilnej uczelni (PPulse) z wykorzystaniem interfejsu NFC. Funkcjonalność ta była przygotowywana w formie odrębnej aplikacji, dzięki czemu można było w łatwy sposób koordynować pracę grupy i na bieżąco testować aplikację bez konieczności angażowania w to osób trzecich. Język programowania oraz narzędzia główne zostały z góry narzucone, aby po ukończeniu projektu wdrożenie oprogramowania nie sprawiło żadnych problemów. Kod aplikacji został przekazany do weryfikacji do Działu Rozwoju Oprogramowania - jednostki organizacyjnej Politechniki Poznańskiej odpowiedzialnej za aplikację PPulse. Powstały kod prototypu ELS zostanie tam zintegrowany z istniejącym już kodem znajdującym się w PPulse. Przygotowana funkcjonalność zostanie dodana w postaci nowego modułu o nazwie ELS i będzie dostępna z menu głównego aplikacji.

6.2 Dalszy rozwój aplikacji

Dzięki utworzeniu pliku konfiguracyjnego aplikacja nie wymaga żadnych modyfikacji kodu oraz jego rekompilacji w przypadku zmian naniesionych w strukturze danych na poziomie kart elektronicznych. Również w przypadku, gdy zostanie wydana nowa karta wystarczy edytować ten plik i dodać do niego strukturę danych tej karty aby aplikacja mogła ją odczytywać i emulować. Dzięki temu koszt utrzymywania aplikacji jest niewielki i procedura konserwacji jest stosunkowo prosta do przeprowadzenia. W przyszłości może być ona rozszerzona o funkcję lokalnego potwierdzania tożsamości. Gdyby aplikacja mogła pełnić rolę uwierzytelniającą właściciela oraz gdyby telefon dzięki emulacji mógł służyć np. w tramwajach czy autobusach jako karta PEKA, ELS w Telefonie posiadałaby już wszystkie dotychczasowe funkcje karty elektronicznej i ta nie byłaby już potrzebna. Używanie telefonu w takich sytuacjach byłoby znacznie wygodniejsze niż używanie kart plastikowych. Dzięki temu aplikacja mogłaby zyskać dużą popularność, a w przyszłości może nawet wyprzeć karty legitymacyjne z użycia. Obecne rozwiązanie obsługuje jedynie karty, które zostały spersonalizowane w Międzyuczelnianym Centrum Personalizacji Legitymacji Studenckiej i zostanie wdrożone w aplikację dedykowaną dla Politechniki Poznańskiej. Udostępnienie prototypu na licencji otwartej umożliwiłoby innym uczelniom lub firmom również na tym skorzystać. W przyszłości ELS w Telefonie może obejmować legitymacje wydane w całej Polsce, a ponadto, gdyby legitymacje elektroniczne miały być wprowadzone również do szkół podstawowych i średnich, a taki projekt rozporządzenia jest obecnie procedowany w parlamencie, można by je także objąć w aplikacji. Aktualnie sprawdzanie ważności podpisu cyfrowego jest dokonywane na podsta-

wie certyfikatu wgranego do telefonu wraz z aplikacją lub poprzez umieszczenie go w odpowiednim folderze przez użytkownika. Poprawnym rozwiązaniem byłoby stworzenie systemu, który dostarcza certyfikaty, które podpisują certyfikat na karcie. Mogłyby być np. pobierane ze specjalnego serwera do tego przeznaczonego. Istnieje również pomysł, aby aplikacja przechowywała zdjęcie studenta/doktoranta, które jest na legitymacji. Jako, że na ELS nie ma tej informacji w postaci elektronicznej, zdjęcie musiałoby pochodzić z systemu personalizacji lub systemu dziekanatowego. Innym pomysłem jest przechowywanie zdjęć kart jako całości. Takie rozwiązanie jest stosowane m.in. w niektórych aplikacjach do kart lojalnościowych i również można je wziąć pod uwagę. Ponieważ aplikacja przechowuje dane osobowe, w celu ich ochrony powinien zostać zaimplementowany mechanizm, który je szyfruje podczas zapisu. Odbywałoby się to poprzez podawanie hasła, które służyłoby jako klucz szyfrujący.

Rozdział 7

Podsumowanie

Cel pracy inżynierskiej został osiągnięty. W ramach projektu powstała w pełni działająca aplikacja, wykorzystująca interfejs NFC do odczytu oraz emulacji danych znajdujących się na kartach legitymacji studenckich oraz doktoranckich. Niestety dość poważną wadą jest jej niska przenośność. W początkowych założeniach aplikacja miała być przeznaczona na wszystkie mobilne systemy operacyjne. To wymaganie jednak szybko zostało zweryfikowane i zredukowano docelowe platformy tylko do systemu Android. Wszelkie inne założenia projektu zostały w pełni lub w znacznym stopniu zrealizowane i wielokrotnie przetestowane z pozytywnymi wynikami.

Aplikacja umożliwia odczyt informacji i sprawdzenie ważności legitymacji studenckich oraz doktoranckich. Aplikacja również sprawdza karty pod kątem posiadania dodatkowych aplikacji na nich zainstalowanych (PEKA, karta biblioteczna). Odczytane karty można zapisać i przechowywać w pamięci telefonu. Funkcja emulowania wcześniej zapisanych kart także została zaimplementowana i działa bez zarzutu. Sprawdzanie rzeczy kryptologicznych zostało zaimplementowane częściowo.

Przygotowana funkcjonalność będzie dostępna do ogólnego użytku po wdrożeniu jej do uczelnianej aplikacji PPulse przez Dział Rozwoju Oprogramowania - jednostkę organizacyjną Politechniki Poznańskiej. W obecnym stanie aplikacja nie jest w stanie w pełni zastąpić legitymacji studenckiej czy doktoranckiej. Przede wszystkim podstawową funkcją legitymacji jest możliwość potwierdzenia tożsamości jej właściciela. Aktualnie nie ma na rynku rozwiązania oferującego dokonanie tego za pomocą urządzenia mobilnego. Istnieje natomiast równolegle rządowy projekt mDokumenty, którego zamysłem jest wprowadzenie możliwości potwierdzania swoich danych osobowych przy użyciu telefonu komórkowego. Wykorzystywana do tego będzie zupełnie inna aplikacja, aczkolwiek jest to krok do zupełnego przeniesienia funkcjonalności legitymacji do telefonu. Niestety w obecnej chwili nie ma możliwości wykorzystania interfejsu NFC w celu zastąpienia karty PEKA podczas korzystania z komunikacji miejskiej. PEKA to aplikacja zainstalowana na karcie legitymacyjnej tak samo jak ELS/ELD czy aplikacja biblioteczna. Emulować można jedynie dane odczytane z tych aplikacji jeśli je dostarczają, natomiast nie da się sprawić, aby aplikacja PEKA działała w taki sam sposób jak na karcie legitymacyjnej. Być może w przyszłości emulacja będzie umożliwiała dokonywanie tego typu rzeczy, co sprawi, że telefon będzie już w pełni mógł zastąpić legitymację, a karty elektroniczne wyjdą z użycia.

Literatura

[1] M. Gosławski, *Electronic Student Identity Card Management System at the Poznan University of Technology*, Politechnika Poznańska, 2011

[2] ISO/IEC 7816-4:2005 *Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*

[3] ISO/IEC 14443-4:2016 *Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol*

[4] B. Kaliski, *PKCS #7: Cryptographic Message Syntax Version 1.5*, RFC 2315, IETF, 1998

[5] P. Nazimek, *Inżynieria programowania kart inteligentnych*, Politechnika Warszawska, 2005

[6] W. Rankl, Effing W., *Smart Card Handbook, Fourth Edition*, Wiley, 2010

[7] S. A. Ahson, M. Ilyas, *Near Field Communications Handbook*, CRC Press, 2012

[8] https://nfc-forum.org/, dostęp 27.04.2018 r.

[9] Załącznik nr 3 do Rozporządzenia Ministra Nauki i Szkolnictwa Wyższego z dnia 16 września 2016 r. w sprawie dokumentacji przebiegu studiów (Dz.U. 2016 poz. 1554)

[10] Rozporządzenie Ministra Nauki i Szkolnictwa Wyższego z dnia 9 sierpnia 2017 r. w sprawie studiów doktoranckich i stypendiów doktoranckich (Dz.U. 2017 poz. 1696)

[11] J. Tomczak, *Aplikacja Biblioteczna jELIB*, Praca dyplomowa magisterska, Politechnika Poznańska, 2009

[12] https://developer.xamarin.com/, dostęp 27.04.2018 r.

[13] http://www.mcp.poznan.pl/, dostęp 27.04.2018 r.

[14] Politechnika Poznańska, System Elektronicznej Legitymacji Studenckiej, Struktura elektroniczna ELIB, 2011

[15] ETSI TS 101 733, Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES), 2013

[16] ITU-T, ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 2002

[17] https://developer.apple.com/documentation/corenfc

[18] https://support.microsoft.com/en-us/help/4001737/products-reaching-end-of-support-for-2017