Politechnika Poznańska
Wydział Elektryczny
Instytut Automatyki i Inżynierii Informatycznej

Projekt Zespołowy

Inteligentny zamek

Autorzy:
Maciej Marciniak
nr indeksu: 121996
e mail:
maciej.r.marcniak@student.put.poznan.pl

Damian Filipowicz
nr indeksu: 122002
e mail:
Damian.Filipowicz@student.put.poznan.pl

8 kwietnia 2017

Spis treści

Wstęp

Inteligentny zamek powinien być systemem, który ma na celu zastąpienie starego modelu zabezpieczeń różnego rodzaju drzwi i skrytek w którym używano tradycyjnych kluczy, czy szyfrów na klucze cyfrowe, którymi będzie można posługiwać się przy pomocy smartfonów z funkcją bluetooth. Celem tego rodzaju usprawnień będzie wyeliminowanie z życia codziennego sytuacji w których użytkownik musi posiadać pęki kluczy. Zamiast tego dzięki temu systemowi może wszystkie klucze przechowywać w jednym miejscu (smartfonie).

System składać się będzie z:

- urządzenia sterującego:
  - mikrokomputera Raspberry Pi 3,
  - serwomechanizmu / elektronicznego zamka,
- aplikacji mobilnej,
- aplikacji serwerowej obsługującej bazę danych.

System będzie spełniał wymagania dotyczące bezpieczeństwa poprzez zastosowanie szeregu funkcji kryptograficznych przy procesie uwierzytelniania jak i przy generowaniu kluczy takich jak np. funkcje skrótu, SSH, algorytmów szyfrowania asymetrycznego, systemu zarządzaniem kluczem publicznym (podpisu cyfrowego).

Używane klucze będą posiadały podpis cyfrowy, który jednoznacznie będzie definiował właściciela oraz stempel czasowy do określania ważności. Klucze będą mogły mieć w zależności od przeznaczenia różne okresy przedawnienia, np. właściciel mieszkający w danym domu posiadać będzie klucz o długim terminie ważności, a goście klucz jednorazowy, bądź kilku godzinny bez możliwości odnowienia. Wszelkie dane dostępowe będą generowane i dystrybuowane na serwerze systemu, z możliwością zdalnej prośby o utworzenie kluczy tylko przez uprawnione przez administratora osoby.

Ogólny schemat systemu znajduje się na Diagramie 1.

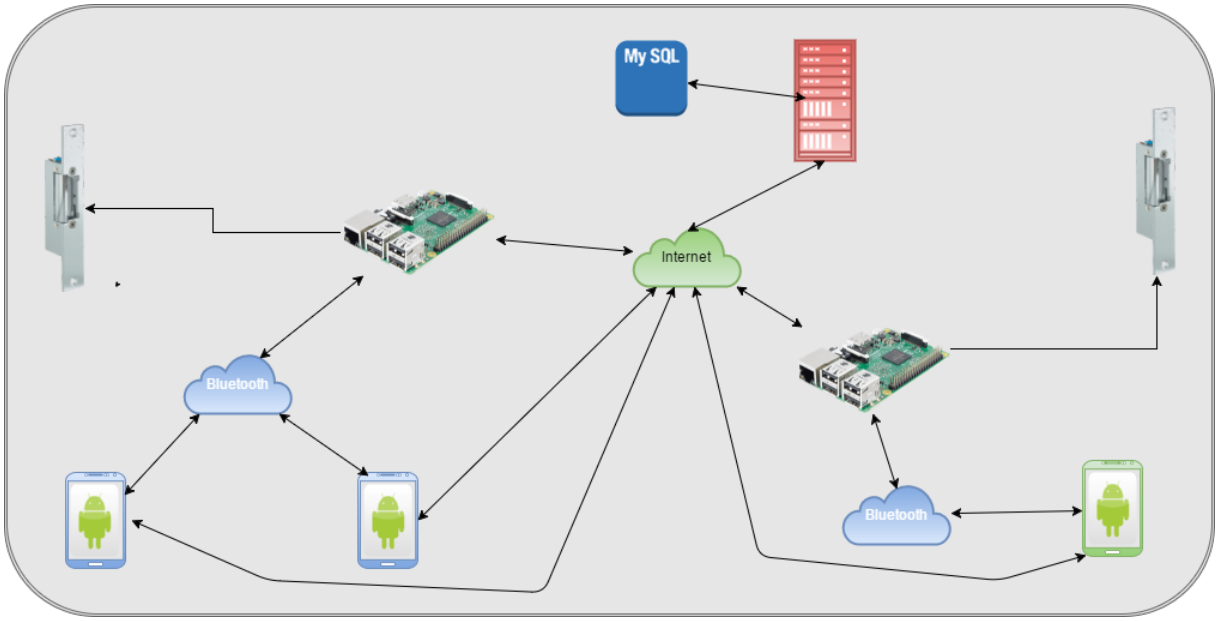

Diagram 1: Diagram wdrożeń

Rozdział 1

Aktorzy systemu

W systemie Inteligentnego zamka wyróżniamy następujących aktorów:

- **RaspberryPi** - jest to mikrokomputer Raspberry Pi 3 sterujący zamkiem,
- **serwomechanizm/elektrozamek** - jest to urządzenie służące do odblokowania/zablokowania zamka,
- **urządzenie mobilne** - jest to urządzenie posiadające system operacyjny, z funkcją bluetooth oraz posiadające możliwość instalacji aplikacji,
- **aplikacja serwerowa** - jest to program znajdujący się na serwerze z dostępem globalnym poprzez Internet,
- **użytkownik** - jest to osoba fizyczna operująca urządzeniem mobilnym, chcąca uzyskać dostęp do zamka.

Użytkowników dodatkowo dzieli się na grupy ze względu na uprawnienia w systemie:

- **gość** - posiada najniższe uprawniania, może jedynie posiadać klucze o krótkim okresie ważności, nie może generować nowych kluczy ani udostępniać ich,
- **użytkownik zalogowany** - posiada uprawnienia gościa, dodatkowo przechowywać może klucze o stałym dostępie do zamka (np. dostęp przez cały dzień),
- **administrator** - może wykonywać wszystkie czynności związane z uprawnieniami gościa i użytkownika zalogowanego, dodatkowo posiada dostęp do statystyk historii zamka, decyduje o rejestracji użytkowników zalogowanych.

Rozdział 2

Opis składowych systemu

2.1 Urządzenie sterujące

Zadaniem urządzenia sterującego, w którego skład wchodzić będą Raspberry Pi 3 oraz serwomechanizm/zamka elektronicznego jest weryfikacja klucza cyfrowego przesyłanego przez urządzenie mobilne oraz otwieranie zamka przy pozytywnym wyniku weryfikacji.

Oprogramowanie mikrokomputera obejmuje system Linux raspbian-jessie oraz szereg podprogramów napisanych w języku Python. Skrypty programów łączą się do serwera w celu pobrania informacji o poprawności i daty ważności certyfikatu dostępu.Jeśli dane będą poprawne to zostaje wysterowany serwomechanizm, który otwiera zamek, w przeciwnym przypadku użytkownik zostanie poinformowany o odmowie dostępu, a nieudana próba dostania się do systemu zarejestrowana zostanie w bazie danych wraz z danymi właściciela klucza. Funkcjonalność urządzenia sterującego przedstawiona została w Tabeli 2.1.

Tabela 2.1: Tabela wymagań funkcjonalnych urządzenia sterującego

Funkcja	Opis	Aktorzy
Parowanie urządzeń bluetooth	Parowanie bluetooth urządzenia mobilnego z Raspberry Pi	RaspberryPi, Urządzenie mobilne
Nasłuchiwanie połączenia bluetooth	Oczekuje na przychodzące połączenia bluetooth	RaspberryPi
Nawiązanie połączenia bluetooth	Każda próba nawiązania połączenia bluetooth zostanie zaakceptowana	RaspberryPi, Urządzenie mobilne
Pobranie pliku z kluczem cyfrowym przez bluetooth	Przesyłanie pliku z kluczem dostępowym z urządzenia mobilnego do Raspberry Pi poprzez bluetooth	RaspberryPi, Urządzenie mobilne, Gość
Weryfikacja poprawności klucza cyfrowego	Wysłanie zapytania w języku SQL do bazy danych	RaspberryPi, Aplikacja serwerowa
Otwarcie zamka	Otwarcie zamka poprzez wysłanie sygnału PWM do serwomechanizmu lub zezwolenie zamka elektronicznego	RaspberryPi, Serwomechanizm/ elektrozamek

Zamknięcie zamka	Zamknięcie zamka poprzez wysłanie sygnału PWM do serwomechanizmu lub zezwolenie zamka elektronicznego	RaspberryPi, Serwomechanizm/ elektrozamek
Rejestracja próby dostępu	Zapis każdej pozytywnej i negatywnej próby weryfikacji klucza cyfrowego w tabeli bazy danych	RaspberryPi
Deszyfracja certyfikatu użytkownika	Deszyfracja pliku z certyfikatem dostępowym używając klucza publicznego użytkownika	RaspberryPi
Pobranie awaryjnego klucza dostępu	Umożliwia wczytanie specjalnego hasła 512-bitowego do Raspberry Pi, który otwiera zamek bez konieczności dostępu do Internetu	RaspberryPi, Aplikacja mobilna, Administrator

Wymagania pozafunkcjonalne:

- jednocześnie może być weryfikowany tylko jeden użytkownik,
- zasięg połączenia bluetooth to maksymalnie 15m,
- niezbędny dostęp do Internetu do połączenia z aplikacją serwerową przy weryfikacji kluczy,
- narzut czasowy związany z weryfikacją poprawności klucza cyfrowego zależny od parametrów serwera i sieci,
- niezbędny ciągły dostęp do zasilania 5V o prądzie co najmniej 2.5A,
- ograniczenia prądowe dla serwomechanizmu lub zamka elektronicznego,
- narzut czasowy związany z uruchomieniem urządzenia - maksymalnie 20 sekund.

2.2 Aplikacja mobilna

Aplikacja mobilna w języku Java na platformę Android ma na celu przechowywanie w pamięci smartfona klucze cyfrowe użytkownika oraz możliwość komunikacji z człowiek-zamek-serwer. Program posiadać powinien interfejs graficzny, dzięki któremu będzie można wybrać, który zamek chce się otworzyć w danej chwili. Klucz cyfrowy przesyłany będzie bezprzewodowo do komputera sterującego zamkiem za pomocą sieci bluetooth. Aplikacja powinna posiadać również funkcję generowania kluczy tymczasowych, które można udostępniać osobom postronnym z ustalonym okresem ważności (jednorazowy, godzinny, od poniedziałku do piątku w godzinach od 8 do 16 itp.). W tym celu zostaje wysłana prośba do serwera poprzez Internet o wygenerowanie klucza o określonych parametrach. Funkcjonalność aplikacja przedstawiona została w Tabeli 2.2.

Tabela 2.2: Tabela wymagań funkcjonalnych aplikacji mobilnej

Funkcja	Opis	Aktorzy
Parowanie urządzeń bluetooth	Parowanie bluetooth urządzenia mobilnego z Raspberry Pi	Urządzenie mobilne, RaspberryPi
Nawiązywanie połączenia bluetooth	Nawiązanie połączenia bluetooth z konkretnym zamkiem identyfikując go jednoznacznie adresem MAC	Urządzenie mobilne, RaspberryPi
Przesłanie pliku klucza cyfrowego	Przesłanie pliku zawierającego klucz cyfrowy do urządzenia sterującego zamkiem. Komputer sterujący odsyła wynik weryfikacji (pozytywny lub negatywny)	Urządzenie mobilne, RaspberryPi
Utworzenie klucza cyfrowego dla gości	Utworzenie specjalnego klucza cyfrowego o ograniczonym dostępie oraz krótkim terminie ważności do użytku dla gości. Każde żądanie generowania klucza wymaga wpisania klucza bezpieczeństwa	Urządzenie mobilne, Aplikacja serwerowa, Użytkownik zalogowany
Udostępnianie klucza cyfrowego dla gości	Udostępnienie specjalnego klucza cyfrowego o ograniczonym dostępie oraz krótkim terminie ważności poprzez np. wiadomość MMS, bluetooth	Urządzenie mobilne, Użytkownik zalogowany
Wczytanie klucza cyfrowego z pliku	Umożliwia wczytanie do listy dostępnych zamków pliku klucza cyfrowego	Urządzenie mobilne, Gość
Pobranie z serwera nowego klucza cyfrowego	Umożliwia pobranie z serwera klucza cyfrowego i dodanie go do listy dostępnych zamków	Urządzenie mobilne, Aplikacja serwerowa, Użytkownik zalogowany
Prośba o przedłużenie ważności klucza	W celu przedłużenia ważności certyfikatu zostaje wysłana prośba do administratora systemu	Urządzenie mobilne, Aplikacja serwerowa, Użytkownik zalogowany
Listowanie dostępnych kluczy	Wyświetlenie na ekranie telefonu listy dostępnych kluczy do danych drzwi	Urządzenie mobilne, Gość

Modyfikacja danych kluczy cyfrowych	Modyfikacja nazw użytkownika, zamków. Pozwala spersonalizować opis zamków	Urządzenie mobilne, Aplikacja serwerowa, Użytkownik zalogowany
Szyfrowanie pliku klucza cyfrowego	Szyfrowanie algorytmem RSA klucza cyfrowego z wykorzystaniem klucza prywatnego	Urządzenie mobilne
Przechowywanie kluczy cyfrowych	Przechowywanie kluczy cyfrowych (szyfrowanych) w pamięci telefonu	Urządzenie mobilne
Podgląd do historii akcji zamków	Umożliwia przeglądanie historii akcji zamka, tzn. daty otwarcia przez kogo, daty zamknięcia	Urządzenie mobilne, Aplikacja serwerowa, Administrator
Autoryzacja użytkownika do aplikacji	Logowanie użytkownika poprzez podanie hasła i loginu do odblokowania aplikacji	Urządzenie mobilne, Aplikacja serwerowa, Użytkownik zalogowany
Rejestracja użytkownika	Założenie nowego konta użytkownika w systemie	Urządzenie mobilne, Aplikacja serwerowa, Gość
Akceptacja przez administratora nowego użytkownika	Administrator systemu może zaakceptować i nadać uprawniania użytkownika	Urządzenie mobilne, Aplikacja serwerowa, Administrator
Zarządzanie ważnością certyfikatów dostępu	Dodawanie, usuwanie ważności certyfikatów dostępowych. Usunięcie praw użytkownika nie skutkuje unieważnieniem wygenerowanych przez niego certyfikatów	Urządzenie mobilne, Aplikacja serwerowa, Administrator
Przesłanie awaryjnego klucza dostępu	Umożliwia wczytanie specjalnego hasła 512-bitowego do Raspberry Pi, który otwiera zamek bez konieczności dostępu do Internetu	Urządzenie mobilne, RaspberryPi, Administrator
Tryb otwierania zamka	Komunikacja z Raspberry może odbywać się automatycznie lub na żądanie wyzwalane przyciskiem otwierania zamka z poziomu aplikacji	Urządzenie mobilne, RaspberryPi, Użytkownik zalogowany

Wymagania pozafunkcjonalne:

- narzut czasowy związany z procesem szyfrowania kluczy cyfrowych (zależny od parametrów urządzenia mobilnego),
- zabezpieczenie transmisji danych poprzez szyfrowanie przy pomocy asymetrycznych kluczy cyfrowych,
- wymagany dostęp do Internetu do zarządzania kluczami, czy logowania,
- przyznanie uprawnień aplikacji do modułu bluetooth, wysyłania wiadomości MMS, Internetu,
- język aplikacji Polski,
- wersja androida minimalna 4.4, docelowa 5.0.

2.3 Aplikacja serwerowa obsługującego bazę danych

Rolą serwera w tym systemie będzie przechowywanie danych dostępowych w bazie danych MySQL oraz generowanie nowych kluczy cyfrowych poprzez program w języku Python. Aplikacja serwerowa oparta powinna być o technologię Python oraz serwera http Nginx. Serwer postawiony powinien być na odrębnym urządzeniu od instalacji zamka, lecz dopuszcza się ze względów ekonomicznych również postawienie serwera na wybranym (jeśli w systemie znajduje się wiele zamków) urządzeniu Raspberry Pi. Funkcjonalność aplikacji serwerowej i bazy danych przedstawiona została w Tabeli 2.3.

Tabela 2.3: Tabela wymagań funkcjonalnych aplikacji serwerowej obsługującej bazę danych

Funkcja	Opis	Aktorzy
Utworzenie klucza cyfrowego na żądanie	Utworzenie pseudolosowego 128-bitowego klucza cyfrowego	Aplikacja serwerowa, Użytkownik zalogowany
Kontrola uprawnień użytkownika	Weryfikacja uprawnień użytkownika do wykonania danej czynności	Aplikacja serwerowa
Modyfikowanie wpisów w bazie danych	Pośredniczenie w modyfikacji danych zawartych w bazie danych	Aplikacja serwerowa
Przekazywanie wpisów z bazy danych	Pośredniczenie w przekazywaniu danych pobieranych z bazy danych	Aplikacja serwerowa
Rejestrowanie żądań dostępu	Zapisywanie danych użytkownika ubiegającego się o dostęp do serwera	Aplikacja serwerowa
Pobranie historii dostępu zamka	Pobranie statystyk związanych z historią dostępu do zamka	Aplikacja serwerowa, Administrator
Zablokowanie dostępu użytkownika	Zablokowanie certyfikatu dostępowego, np. w przypadku kradzieży telefonu	Aplikacja serwerowa, Administrator

Wymagania niefunkcjonalne:

- ograniczenie pamięci dostępnej dla bazy danych (32Gb - pamięć niezbędna dla systemu operacyjnego i oprogramowania),
- ograniczenie liczby obsługiwanych zamków zależna od wielkości dostępnej pamięci i liczby użytkowników,
- narzut czasowy związany z generowaniem nowych kluczy,
- ograniczenie liczby użytkowników wykonujących jednocześnie żądania do serwera - 9 urządzeń,
- wymagany system operacyjny Linux dedykowany pod Raspberry,
- dostęp do Internetu do połączenia z zamkami i urządzeniami mobilnymi,
- zabezpieczenie bazy danych hasłem generowanym losowo.

Rozdział 3

Diagram przypadków użycia

Funkcjonalność systemu przedstawiono na Diagramie 3.1 przypadków użycia.

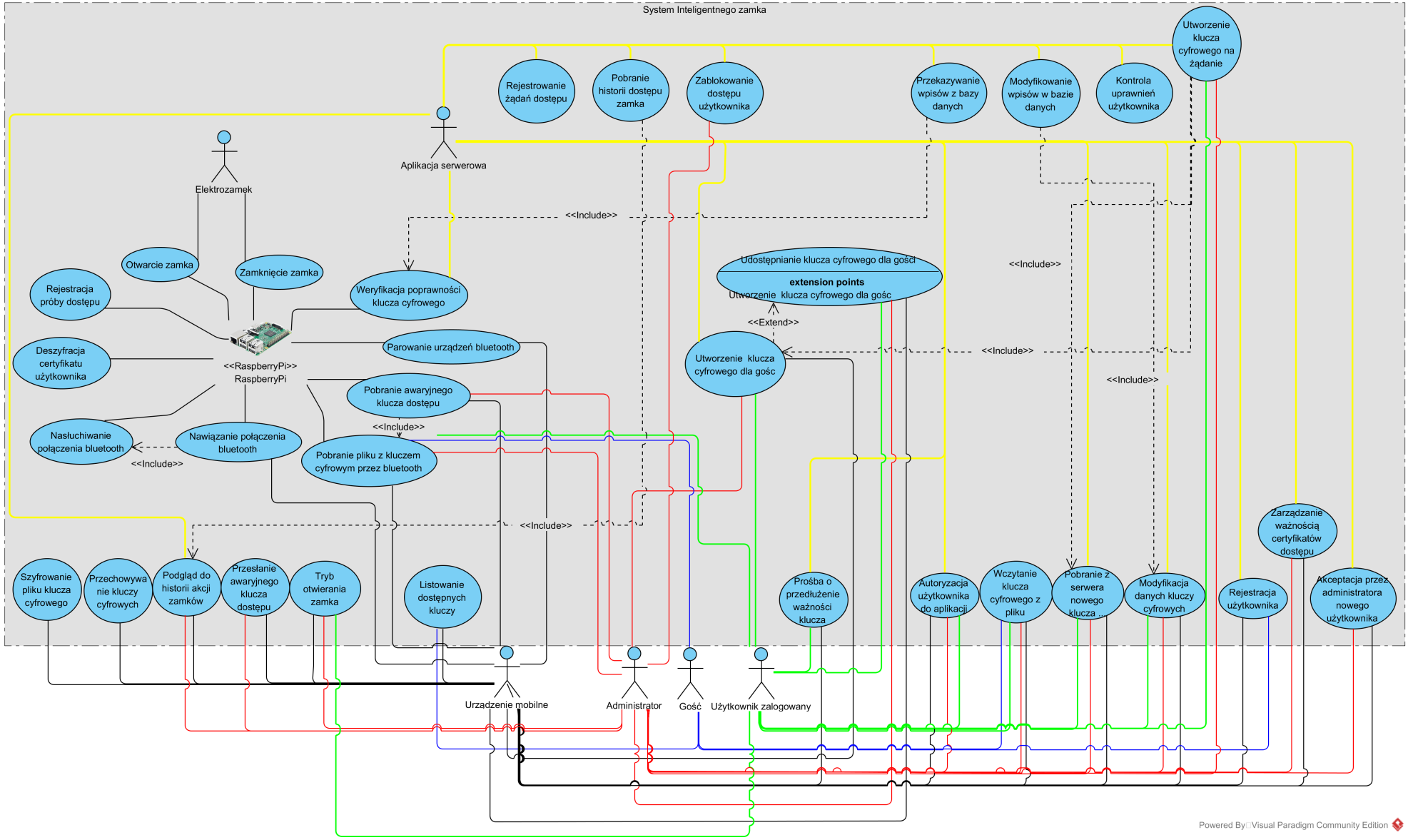

Diagram 3.1: Diagram przypadków użycia

Rozdział 4

Projekt bazy danych

Baza danych przechowywać będzie składać się z pięciu tabel:

- **USERS** - przechowuje dane użytkowników oraz dane niezbędne przy weryfikacji logowania,
- **LOCKS** - zawiera informacje na temat dostępnych w systemie zamków,
- **ACCESS_TO_LOCKS** - archiwizuje próby użycia certyfikatów,
- **TYPES_ACCESS** - przechowuje zdefiniowane typy uprawnień dostępowych do zamków,
- **LOCKS_KEYS** - zawiera wszystkie klucze dostępowe użytkowników.

Wiersz tabeli USERS zawierać musi:

- **ID_USER** - unikalny identyfikator (klucz główny) użytkownika składający się z 10 cyfr,
- **LOGIN** - unikalna nazwa użytkownika niezbędna podczas logowania, zawierająca nie więcej niż 255 znaków,
- **PASSWORD** - hasło zapisane w postaci skrótu, potrzebne do autoryzacji dostępu użytkownikowi,
- **NAME** - imię użytkownika,
- **SURNAME** - nazwisko użytkownika,
- **IS_ADMIN** - pole boolowskie wskazujące czy dany użytkownik jest administratorem czy nie.

Zamek opisywany jest poprzez kolumny:

- **ID_LOCK** - unikalny identyfikator (klucz główny) zamka składający się z 10 cyfr,
- **NAME** - unikalna nazwa zamka,
- **LOCALIZATION** - nieobowiązkowe pole opisujące fizyczne położenie zamka.

Typy uprawnień posiadać będę takie parametry jak:

- **ID_TYPE** - unikalny identyfikator (klucz główny) typu dostępu składający się z 10 cyfr,
- **NAME** - unikalna nazwa typu dostępu,
- **DAYS** - binarny ciąg oznaczającym których dni dotyczy dostęp, najstarszy bit oznacza poniedziałek, zaś najmłodszy niedzielę, np. 1011000 oznacza dostęp w poniedziałek, środę i czwartek, w pozostałe dni nie,
- **FROM_HOURS** - godzina od której obowiązuje dostęp,
- **TO_HOURS** - godzina do której obowiązuje dostęp,
- **DESCRIPTION** - nieobowiązkowy pole charakteryzujące typ dostępu.

Klucz dostępowy składa się z:

- **ID_KEY** - unikalny identyfikator (klucz główny) klucza dostępowego składający się z 10 cyfr,
- **ID_LOCK** - klucz obcy do tabeli tabeli przechowującej dostępne zamki,
- **ID_USER** - klucz obcy do tabeli tabeli przechowującej dane użytkownika, jest to pole służące do określenia kto utworzył klucz dostępu,
- **ACCESS_TYPE** - klucz obcy do tabeli tabeli przechowującej typy dostępu, jest to pole służące do określenia jakiego rodzaju dostęp ma klucz dostępowy,
- **KEY** - unikalna wartość certyfikatu dostępu,
- **FROM** - data od której obowiązuje klucz,
- **TO** - data do której obowiązuje klucz,
- **ISACTUAL** - data wygaśnięcia klucza, jeśli równa TO, oznacza to że klucz utracił ważność z powodu czasu, jeśli różna oznacza, to że zablokowano z innego powodu ważność,
- **NAME** - imię osoby, której dotyczy certyfikat,
- **SURNAME** - nazwisko osoby, której dotyczy certyfikat.

W tabeli archiwizującej akcje na zamku znajdują się takie dane jak:

- **ID** - unikalny identyfikator (klucz główny) akcji wykonanej na certyfikacie składający się z 10 cyfr,
- **ID_KEY** - klucz obcy do tabeli tabeli przechowującej klucze dostępowe, dzięki tej informacji możemy uzyskać dane o zamku który został otwierany jak również do kogo należał klucz,
- **DATE** - dokładna data z godziną użycia klucza dostępowego,
- **ACCESS** - binarna flaga informująca czy dostęp został przyznany czy odmówiony.

Diagramy bazy danych odpowiednio encji i relacji przedstawione zostały na Diagramie 4.1 i 4.2.

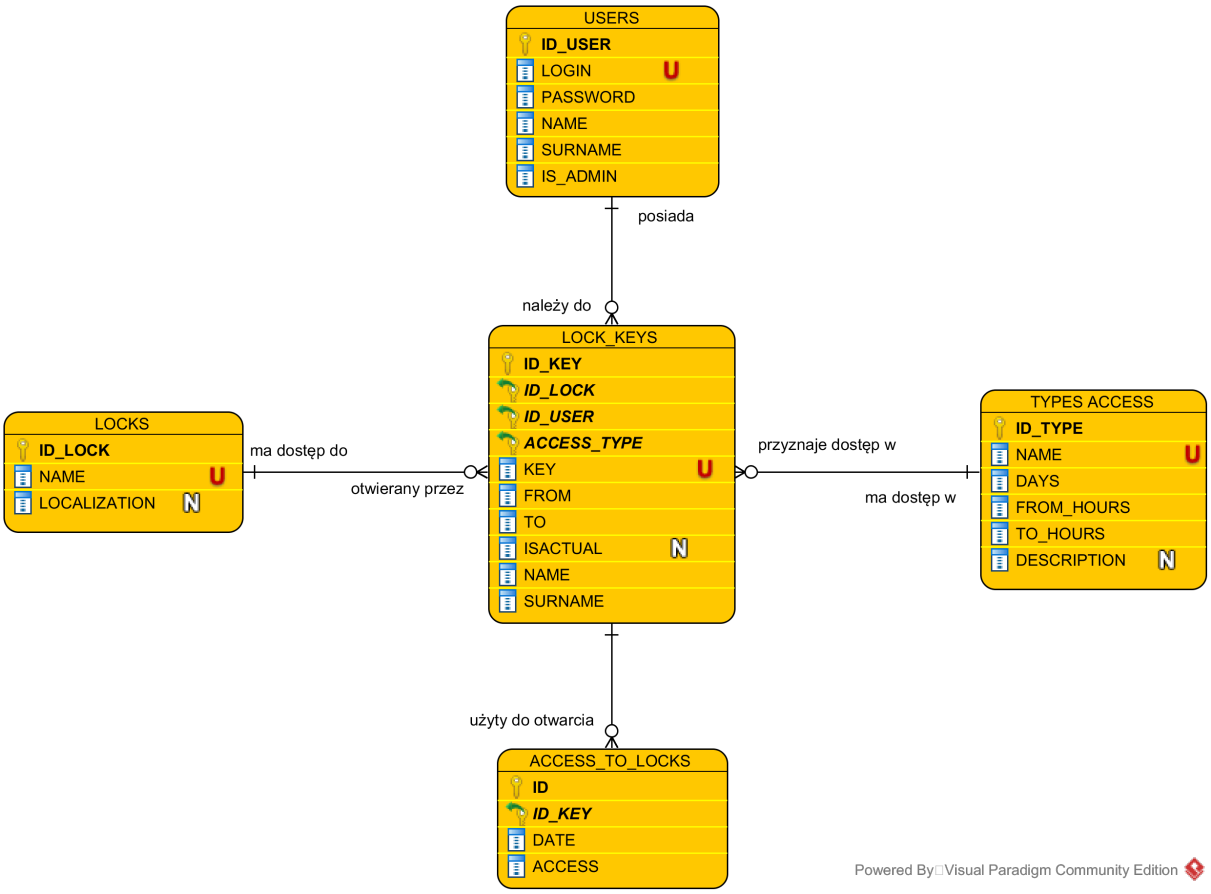

Diagram 4.1: Diagram encji bazy danych

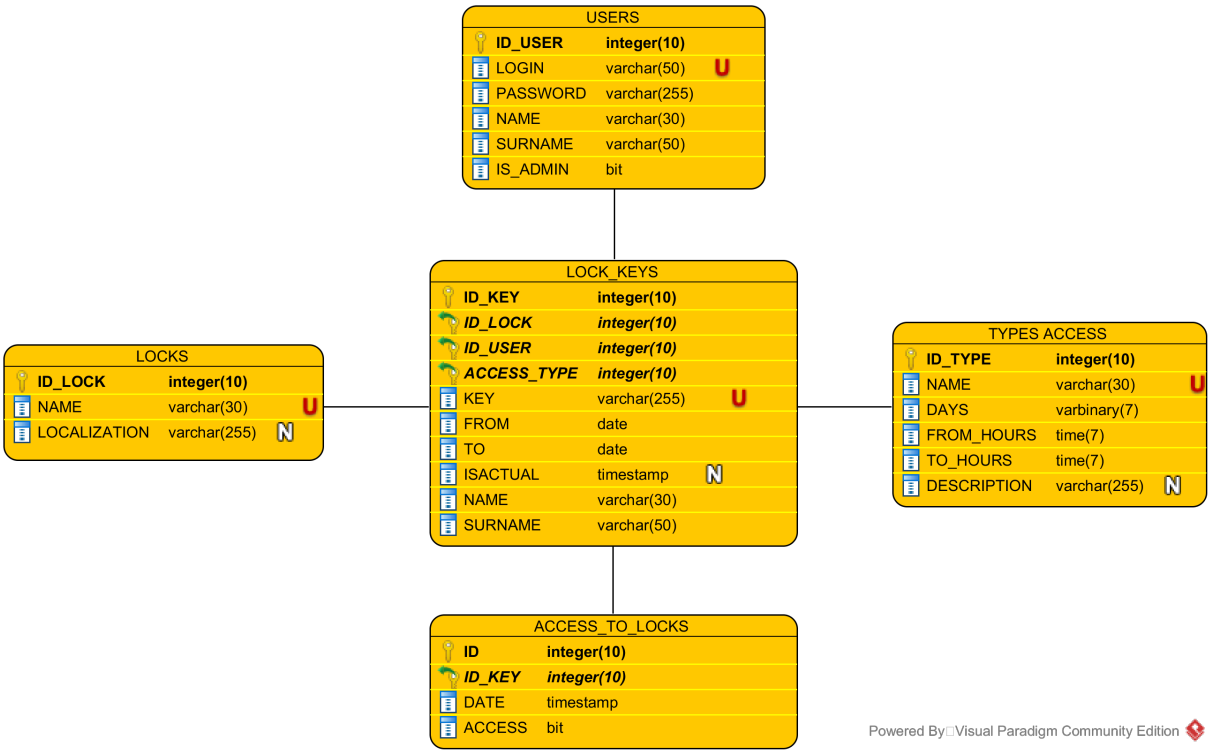

Diagram 4.2: Diagram relacji bazy danych

Rozdział 5

Widok graficzny systemu

Jedynym elementem graficznym systemu Inteligentnego Zamka jest aplikacja znajdująca się na urządzeniach mobilnych. Poniżej opisano,krótko poszczególne widoki wykonane w środowisku Android Studio.

5.1 Panel logowania użytkownika

5.2 Panel rejestracji

5.3 Panel listy zamków

Widok listy dostępnych zamków przedstawia listę nazw zamków do jakich dany użytkownik ma dostęp. Ułatwieniem jest możliwość sortowania wyników i wyszukiwanie po nazwach. Kliknięcie w nazwę zamka powoduje otwarcie zamka. Ustawić można również zamek, który ma być otwierany automatycznie gdy jest się w pobliżu zamka. (Rysunek 5.1 i 5.2)

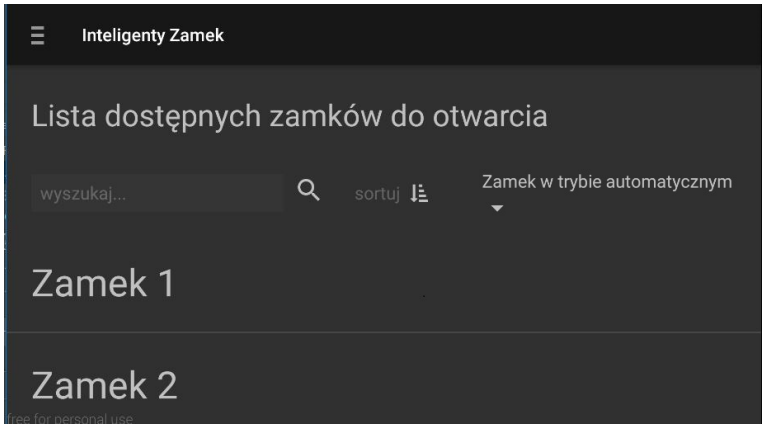

Diagram 5.2: Lista dostępnych zamków (poziomo)

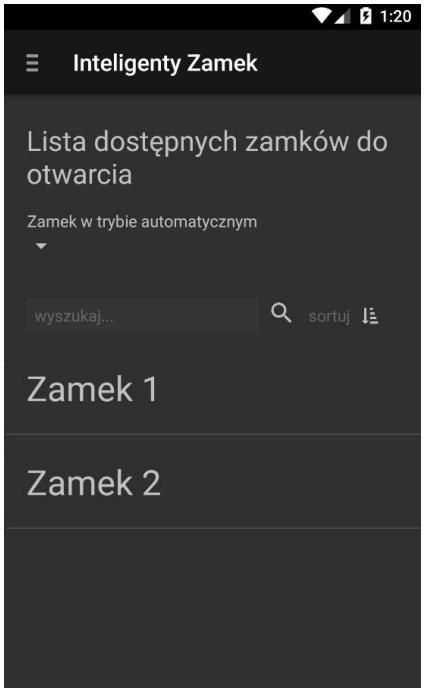

Diagram 5.1: Lista dostępnych zamków (pionowo)

5.4 Panel boczny

5.5 Panel zarządzania certyfikatami

5.6 Panel listy aktualnych certyfikatów

5.7 Panel certyfikatu

5.8 Panel wnioskowania o certyfikat

5.9 Panel tworzenia certyfikatu dla gościa

5.10 Panel administratora

5.11 Panel historii użycia zamków

5.12 Panel generowania nowego certyfikatu (administrator)

5.13 Panel dodawania typu dostępu

5.14 Panel zarządzania certyfikatami (administrator)

5.15 Panel listy oczekujących użytkowników

5.16 Panel listy oczekujących certyfikatów

5.17 Panel ustawień konta