

## PROJEKT ZESPOŁOWY

### Inteligentny zamek

---

*Autorzy:*

MACIEJ MARCINIĄK

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

# Spis treści

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Aktorzy systemu</b>                                | <b>4</b>  |
| <b>2</b> | <b>Opis składowych systemu</b>                        | <b>5</b>  |
| 2.1      | Urządzenie sterujące . . . . .                        | 5         |
| 2.2      | Aplikacja mobilna . . . . .                           | 7         |
| 2.3      | Aplikacja serwerowa obsługująca bazę danych . . . . . | 9         |
| <b>3</b> | <b>Diagram przypadków użycia</b>                      | <b>11</b> |
| <b>4</b> | <b>Diagramy sekwencji</b>                             | <b>13</b> |
| <b>5</b> | <b>Diagramy Klas</b>                                  | <b>22</b> |
| <b>6</b> | <b>Projekt bazy danych</b>                            | <b>25</b> |
| <b>7</b> | <b>Zabezpieczenia systemu</b>                         | <b>28</b> |
| <b>8</b> | <b>Widok graficzny systemu</b>                        | <b>29</b> |
| <b>9</b> | <b>Testowanie aplikacji</b>                           | <b>44</b> |

# 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).

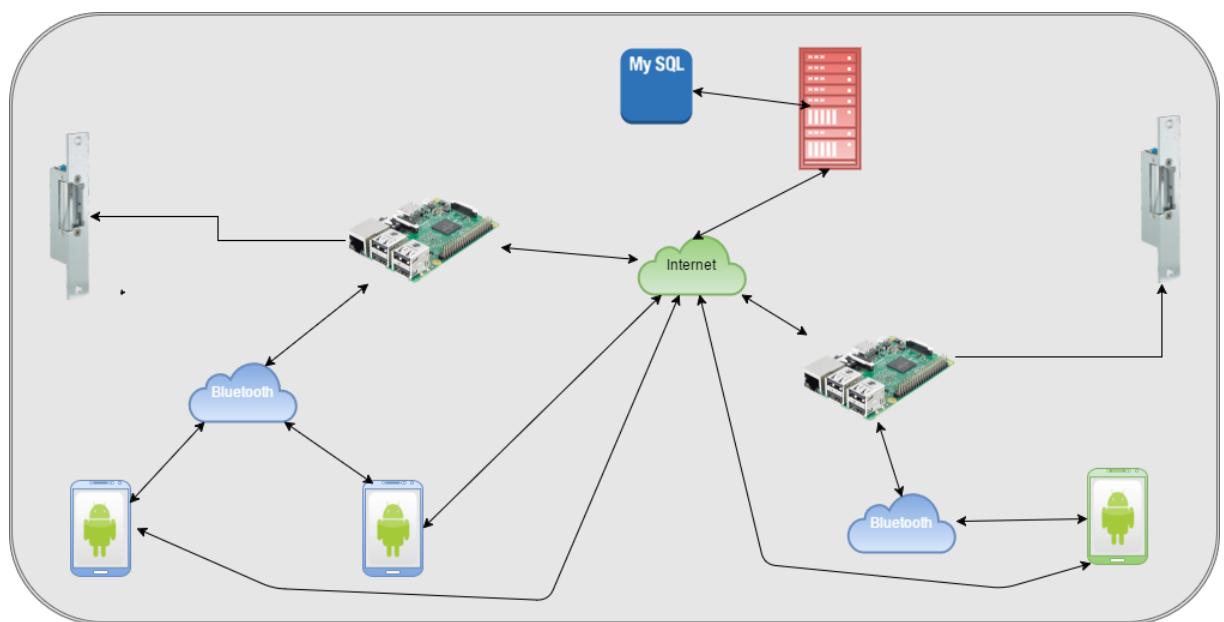
System składać się będzie z:

- 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 Rysunku 1.



Rysunek 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, chcącą 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,<br>Urządzenie mobilne              |
| Nasłuchiwanie połączenia bluetooth                | Oczekuje na przychodzące połączenia bluetooth   | RaspberryPi                                     |
| Nawiązanie połączenia bluetooth                   | Każda próba nawiązania połączenia bluetooth zostanie zaakceptowana                                  | RaspberryPi,<br>Urządzenie mobilne              |
| Pobranie pliku z kluczem cyfrowym przez bluetooth | Przesyłanie pliku z kluczem dostępowym z urządzenia mobilnego do Raspberry Pi poprzez bluetooth     | RaspberryPi,<br>Urządzenie mobilne,<br>Gość     |
| Weryfikacja poprawności klucza cyfrowego          | Porównanie danych z bazy danych z otrzymanymi w kluczu dostępowym                                   | RaspberryPi,<br>Aplikacja serwerowa             |
| Otwarcie zamka                                    | Otwarcie zamka poprzez wysłanie sygnału PWM do serwomechanizmu lub zezwolenie zamka elektronicznego | RaspberryPi,<br>Serwomechanizm/<br>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,
- zainstalowany interpreter języka Python 2.7 z pakietem bibliotek (RSA, Bluetooth, Lightblue, RPi.GPIO),
- 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.

Klucze przesyłane przez bluetooth powinny być szyfrowane algorytmem RSA (kluczem prywatnym). Klucz prywatny otrzymywany wraz z certyfikatem przypisany jest tylko do jednego zamka i przekazywany wraz z nim. W celu przeniesienia klucza dostępowego na inne urządzenie należy wyeksportować klucz, a następnie importować do nowego urządzenia.

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                  |
| Logowanie użytkownika                              | Zalogowanie do systemu poprzez podanie adresu IP serwera, loginu oraz hasła   | 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         |
| Importowanie z pliku klucza prywatnego             | Pobranie z pliku klucza prywatnego klucza dostępowego   | Urządzenie mobilne   |
| Eksportowanie do pliku klucza prywatnego           | Przesłanie do pliku klucza prywatnego potrzebnego do szyfrowania kluczy dostępowych   | Urządzenie mobilne   |

## 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 obsługująca 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 Apache2. 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 przedstawiona została w Tabeli 2.3.

Tabela 2.3: Tabela wymagań funkcjonalnych aplikacji serwerowej

| Funkcja                                 | Opis  | Aktorzy                                    |
|---|---|--|
| Utworzenie klucza cyfrowego użytkownika | Utworzenie pseudolosowego 128-bitowego klucza prywatnego i publicznego użytkownika potrzebnych do uwierzytelnienia kluczy dostępowych | 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         |

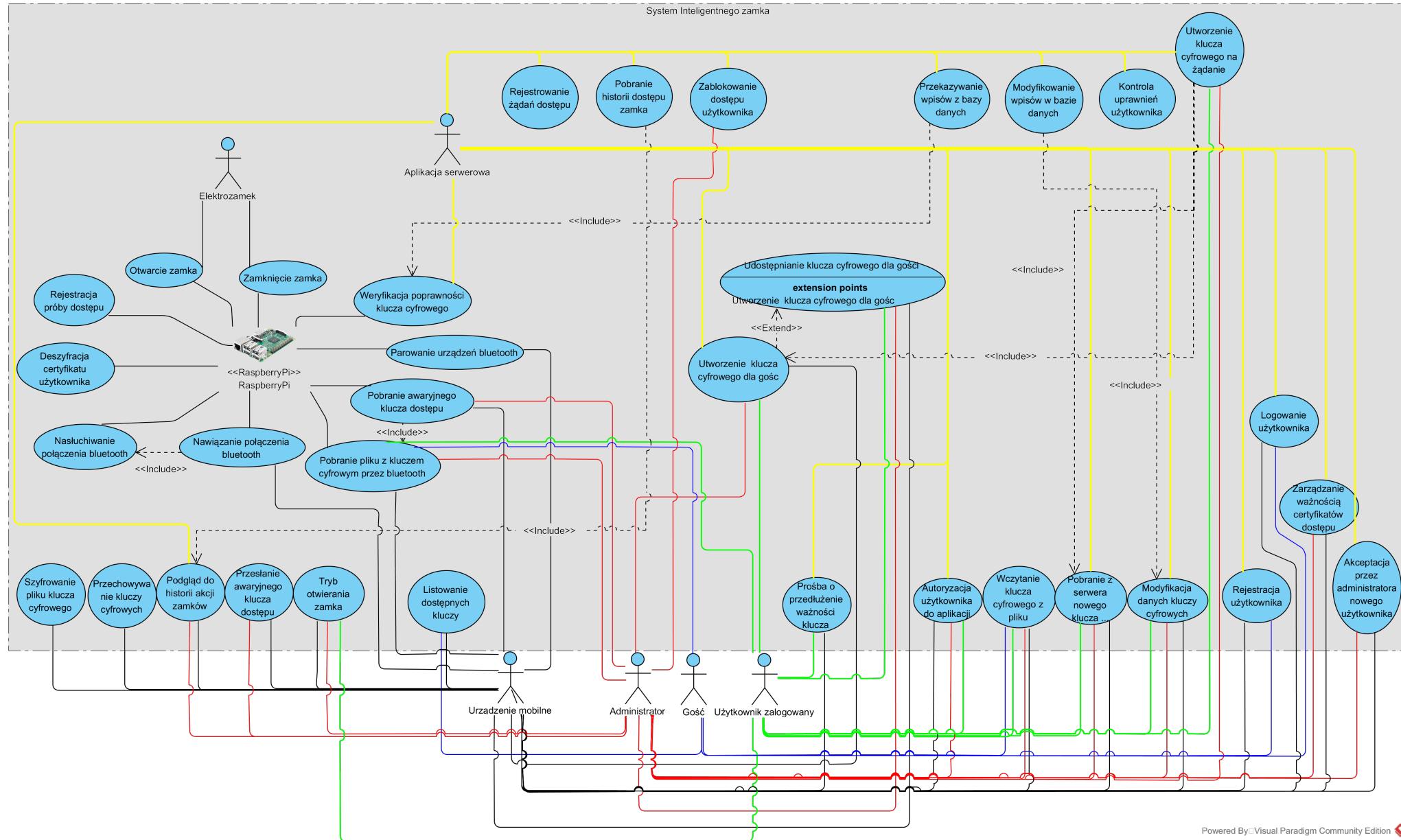
### **Wymagania niefunkcjonalne:**

- 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,
- zainstalowany interpreter języka Python 2.7 wraz z pakietem bibliotek (MySQL, RSA, Django),
- zainstalowany i skonfigurowany serwer Apache2,
- dostęp do Internetu do połączenia z zamkami i urządzeniami mobilnymi,
- zabezpieczenie bazy danych hasłem generowanym losowo.

# Rozdział 3

## Diagram przypadków użycia

Diagram przypadków użycia (funkcjonalności) systemu wraz z opowiadającymi aktorami przedstawiono na Rysunku 3.1.



Rysunek 3.1: Diagram przypadków użycia

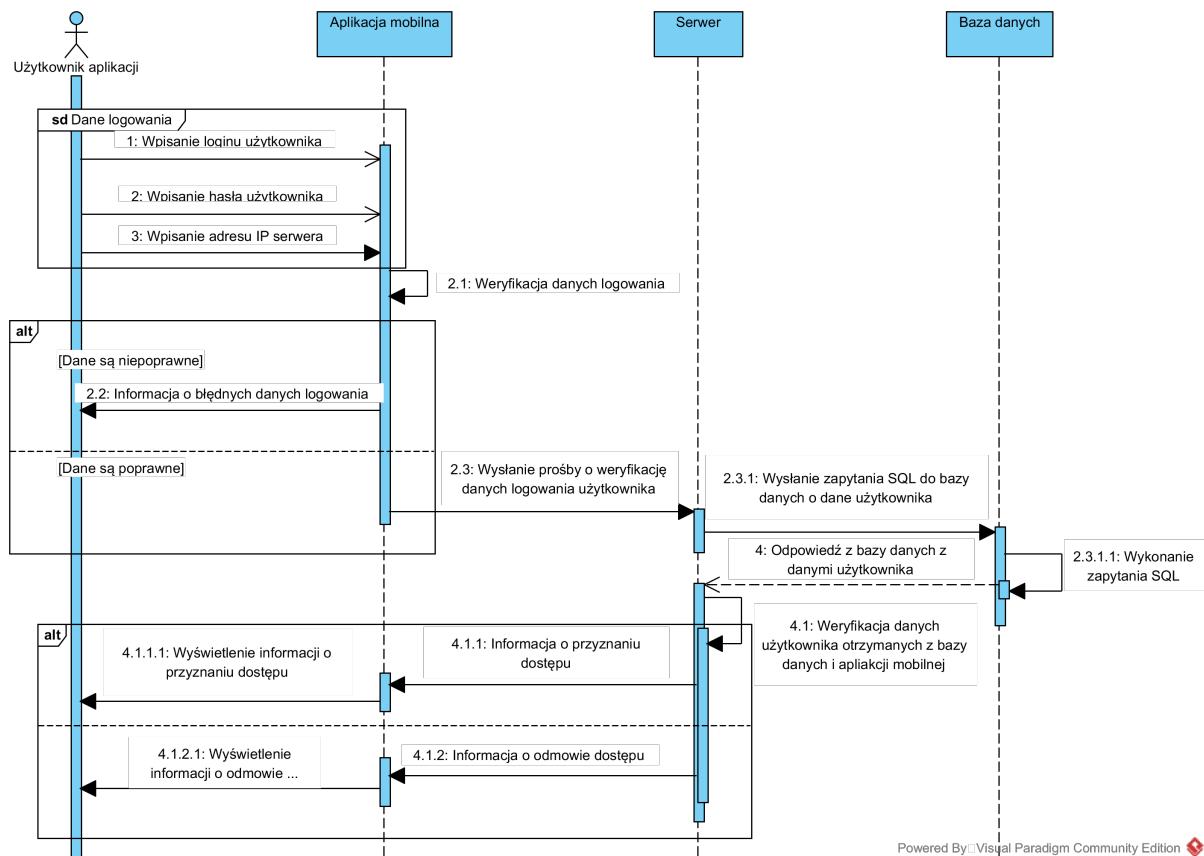
# Rozdział 4

## Diagramy sekwencji

Diagramy poniżej przedstawione mają mieć na celu przybliżyć ogólne działanie systemu. Schematy nie są odzwierciedleniem poszczególnych przypadków użycia, może zawierać nawet niepełne odniesienia do wielu.

### Logowanie użytkownika

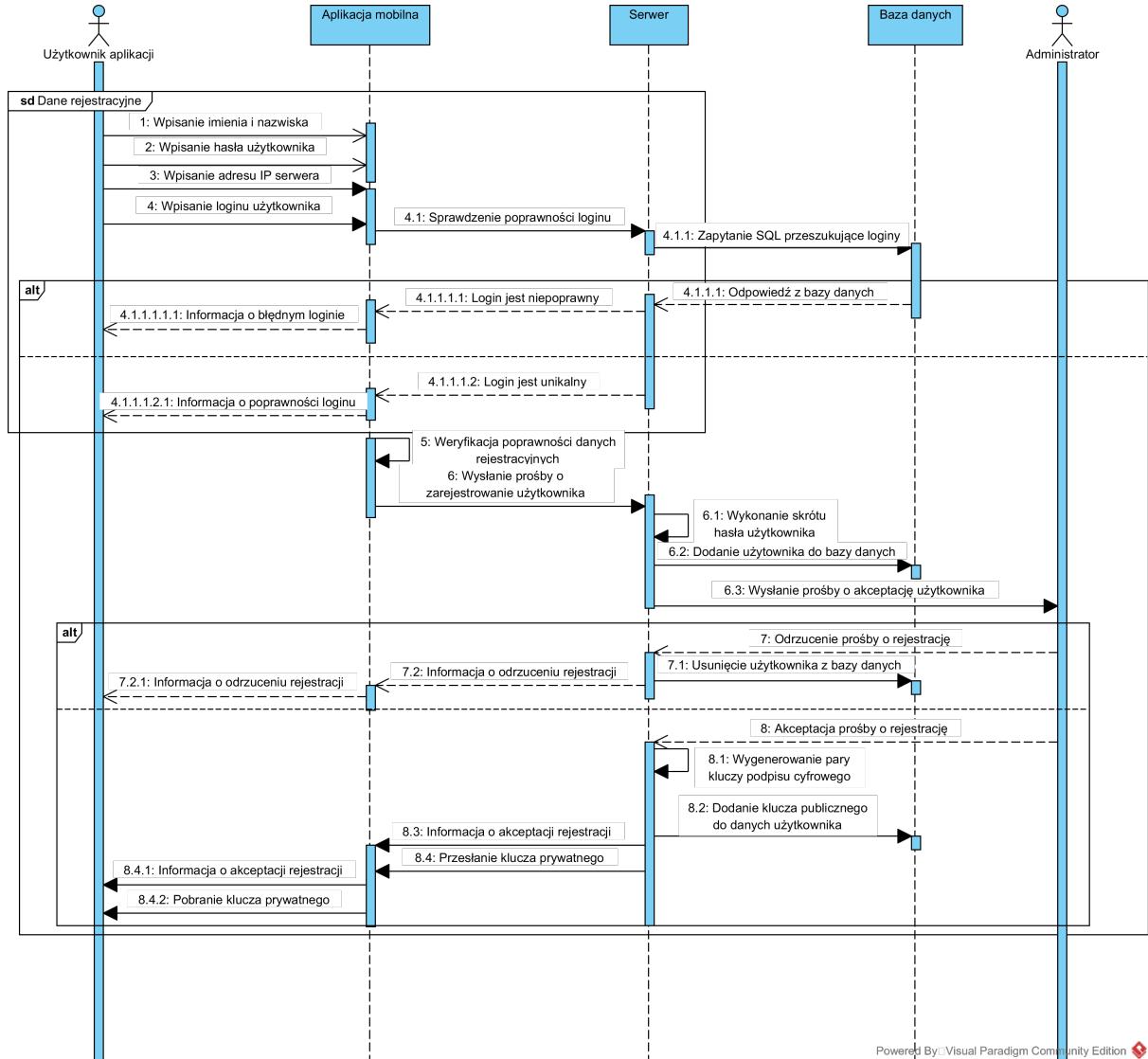
Diagram 4.1 opisuje proces logowania użytkownika do aplikacji mobilnej z weryfikacją danych na serwerze.



Rysunek 4.1: Diagram sekwencji logowanie użytkownika

# Rejestracja użytkownika

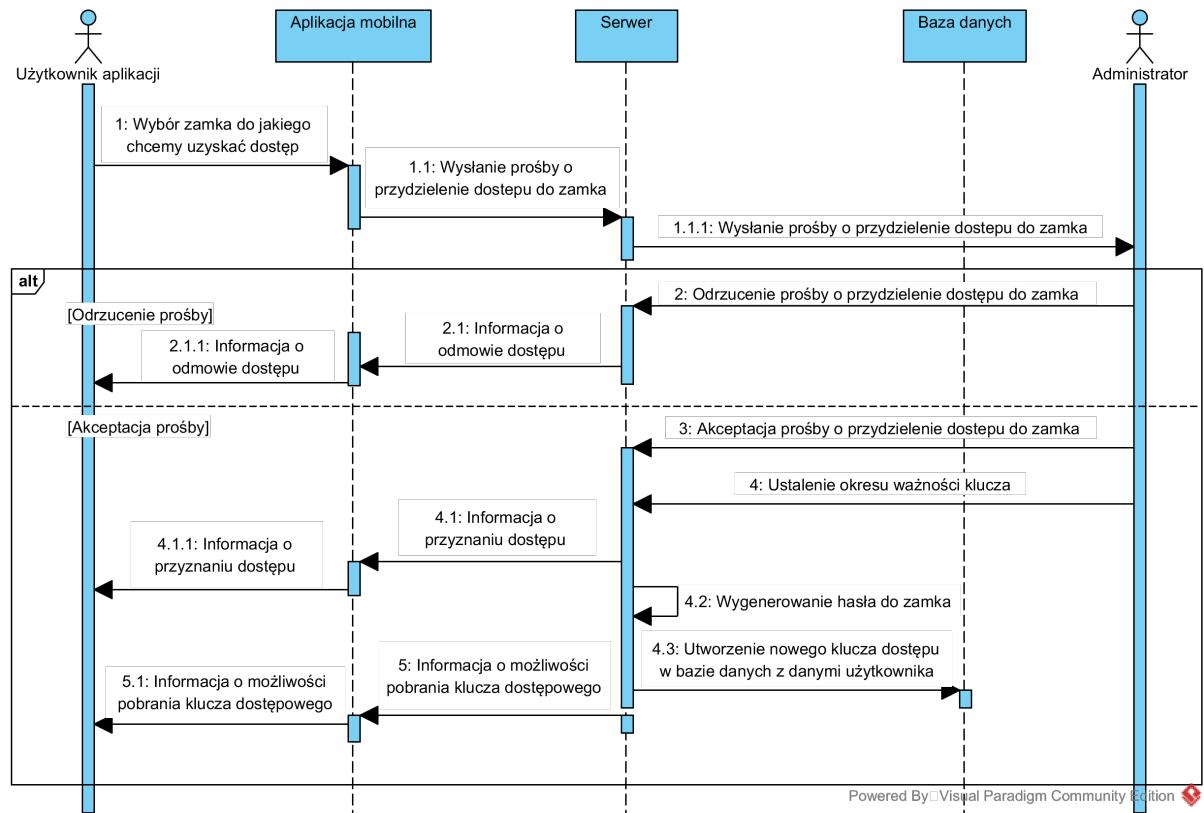
Diagram 4.2 opisuje proces rejestracji użytkownika do systemu z weryfikacją danych na serwerze oraz podjęciem decyzji przez administratora o przydzielaniu dostępu.



Rysunek 4.2: Diagram sekwencji rejestracja użytkownika

# Generowanie certyfikatu użytkownika

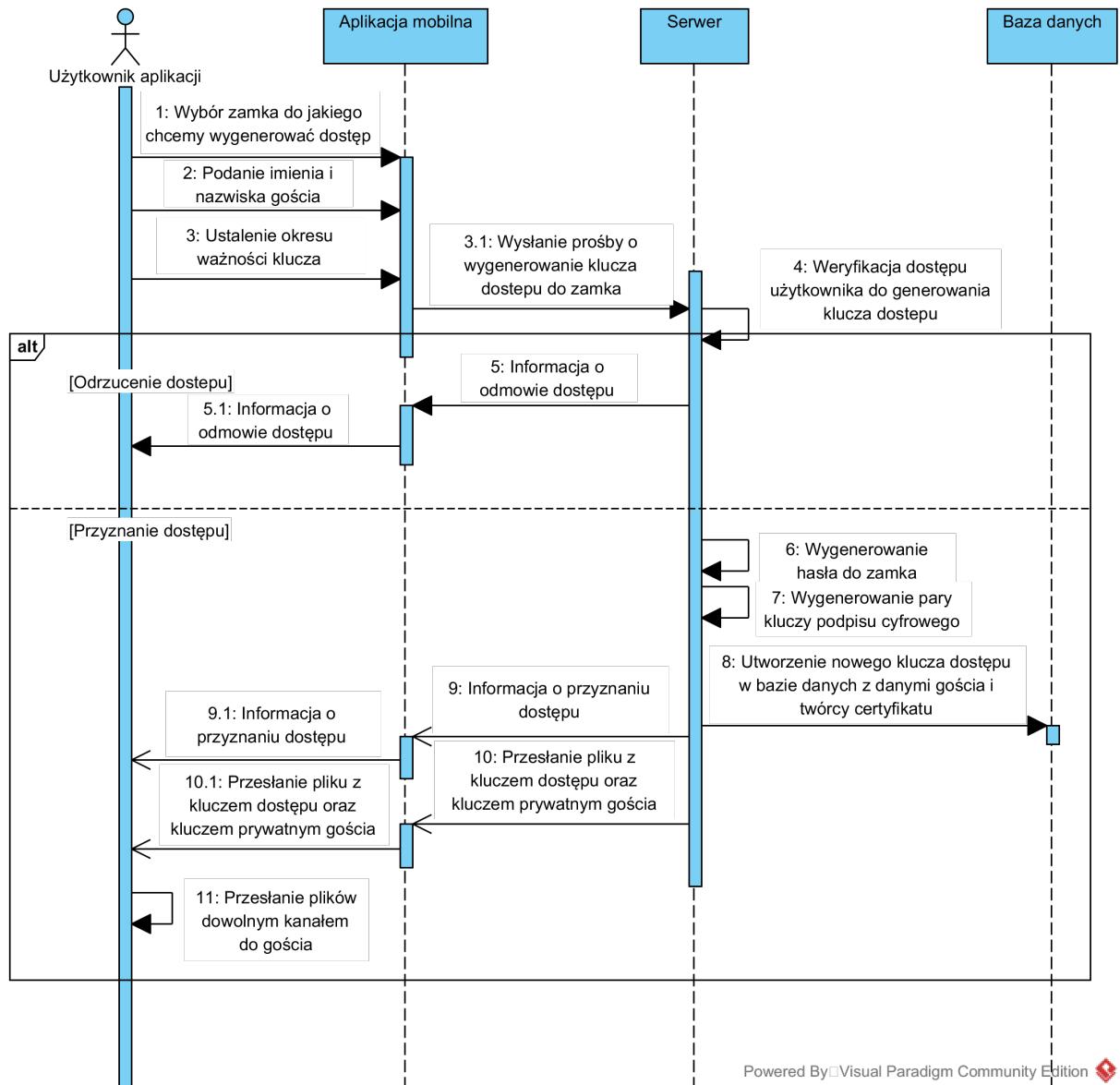
Diagram 4.3 przedstawia sekwencję operacji podczas generowania certyfikatu użytkownika oraz podjęcia decyzji przez administratora o przydzieleniu dostępu.



Rysunek 4.3: Diagram sekwencji generowanie certyfikatu użytkownika

# Generowanie certyfikatu dla gościa

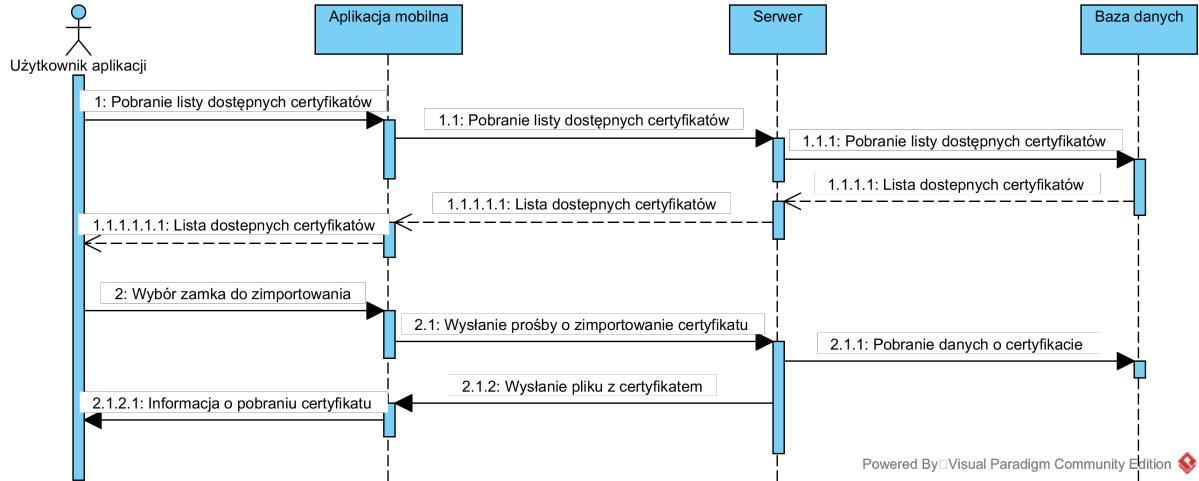
Diagram 4.4 przedstawia sekwencję operacji podczas generowania certyfikatu dla gościa przez użytkownika zalogowanego wraz z wyborem okresu ważności klucza oraz udostępnienie certyfikatu gościowi.



Rysunek 4.4: Diagram sekwencji generowanie certyfikatu gościa

# Importowanie certyfikatu z serwera

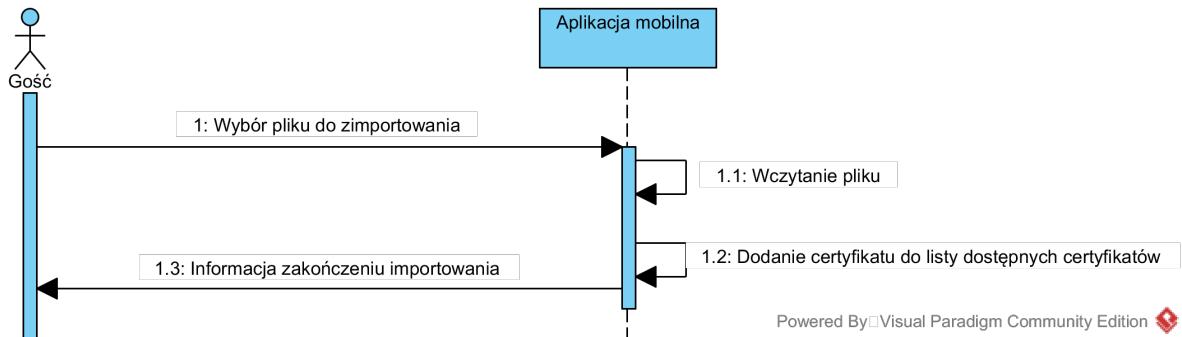
Diagram 4.5 opisuje operację importowania certyfikatu z serwera.



Rysunek 4.5: Diagram sekwencji generowanie importowanie certyfikatu z serwera

# Importowanie certyfikatu z pliku

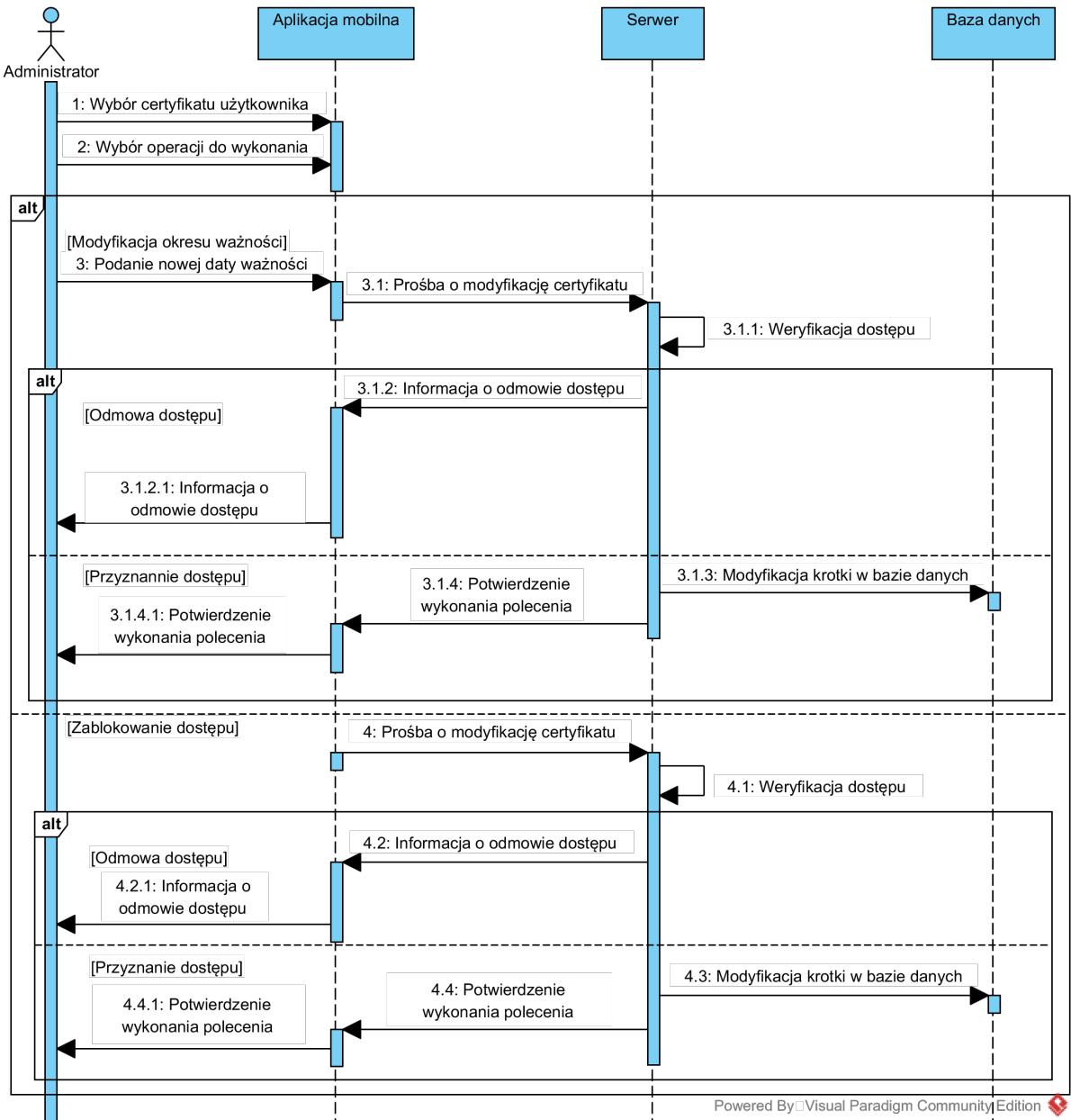
Diagram 4.6 opisuje operację importowania certyfikatu z pliku.



Rysunek 4.6: Diagram sekwencji generowanie importowanie certyfikatu z pliku

# Modyfikacja certyfikatu

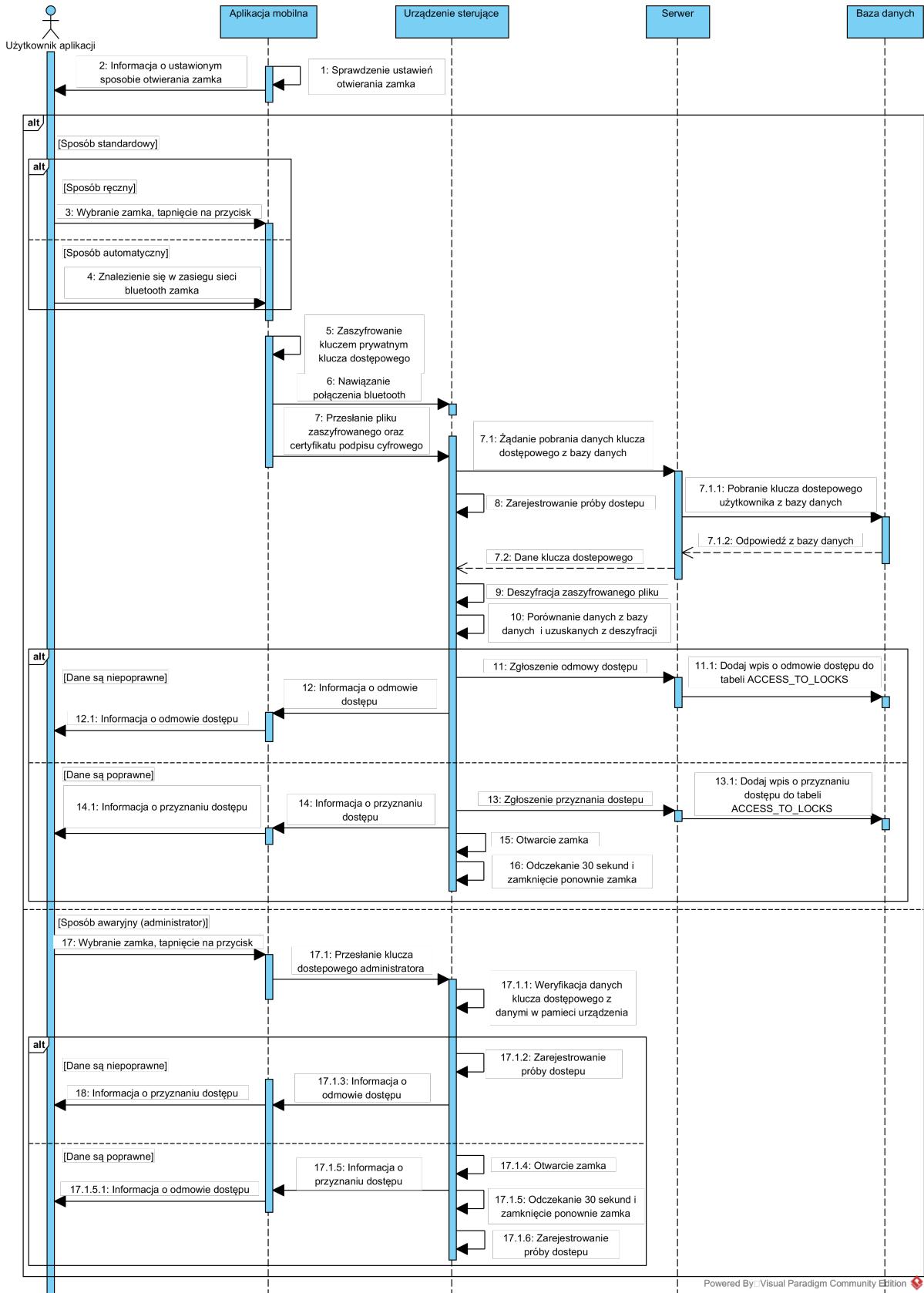
Diagram 4.7 opisuje proces modyfikacji, przez administratora, certyfikatów znajdujących się w bazie danych.



Rysunek 4.7: Diagram sekwencji modyfikacji certyfikatu

# Otwieranie zamka

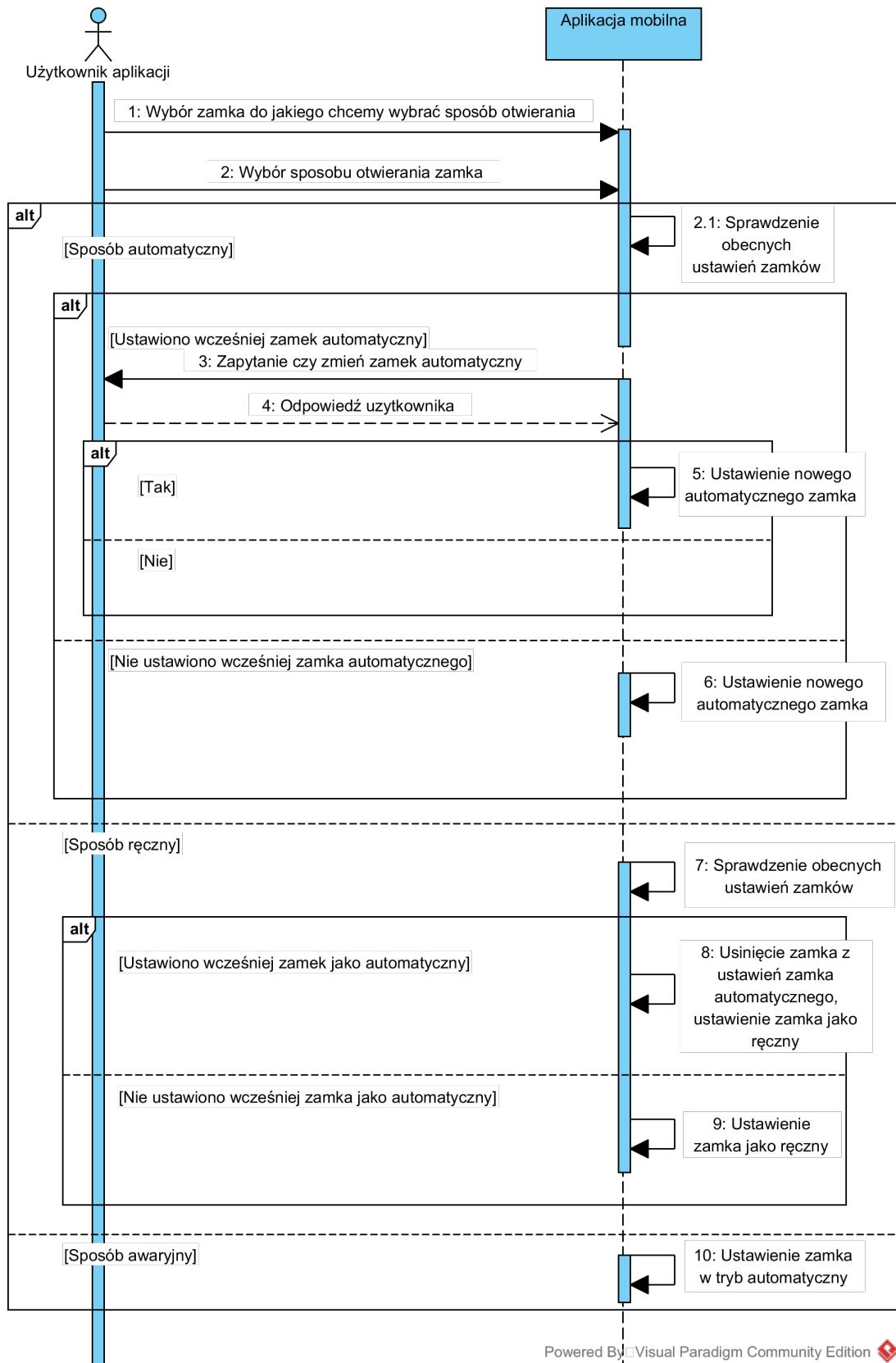
Diagram 4.8 przedstawia operacje wykonywane podczas otwierania zamka wraz z jego automatycznym zamknięciem.



Rysunek 4.8: Diagram sekwencji otwieranie zamka

# Wybór sposobu otwierania zamka

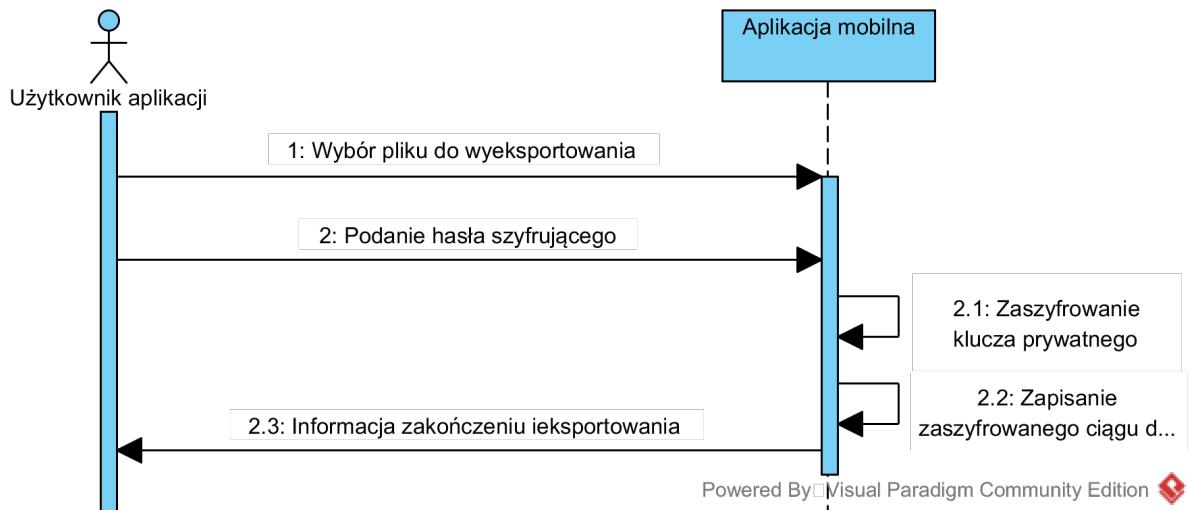
Diagram 4.9 przedstawia operacje wykonywane podczas wyboru sposobu otwierania zamka.



Rysunek 4.9: Diagram sekwencji ustawienia sposobu otwierania zamka

## Eksportowanie klucza prywatnego użytkownika

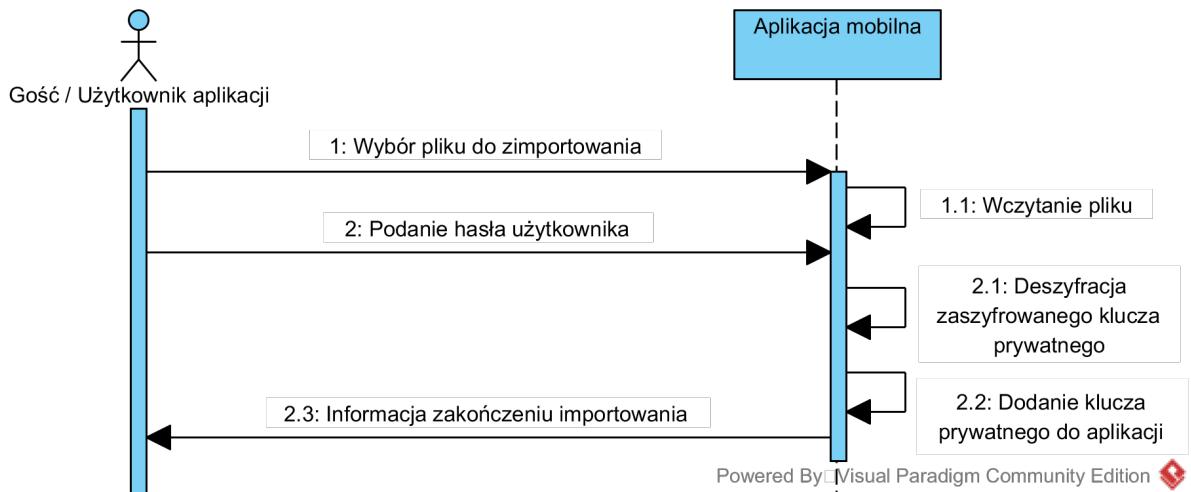
Diagram 4.10 opisuje operację eksportowania klucza prywatnego użytkownika.



Rysunek 4.10: Diagram sekwencji eksportowania klucza prywatnego użytkownika

## Importowanie klucza prywatnego użytkownika

Diagram 4.11 opisuje operację importowania klucza prywatnego użytkownika.



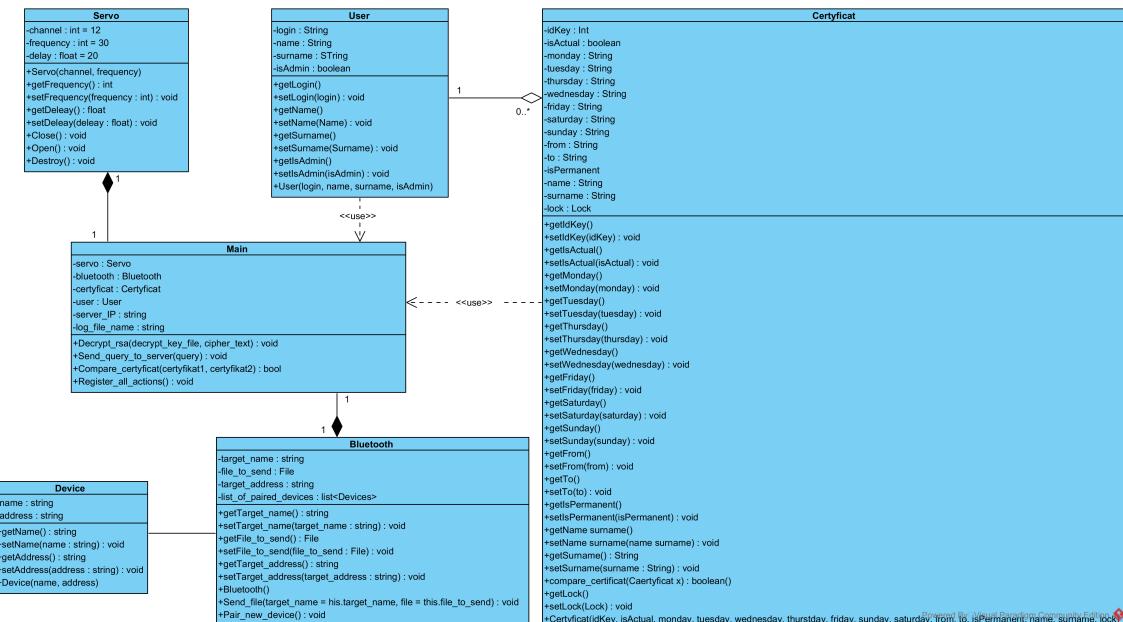
Rysunek 4.11: Diagram sekwencji importowania klucza prywatnego użytkownika

# Rozdział 5

## Diagramy Klas

### Urządzenie sterujące

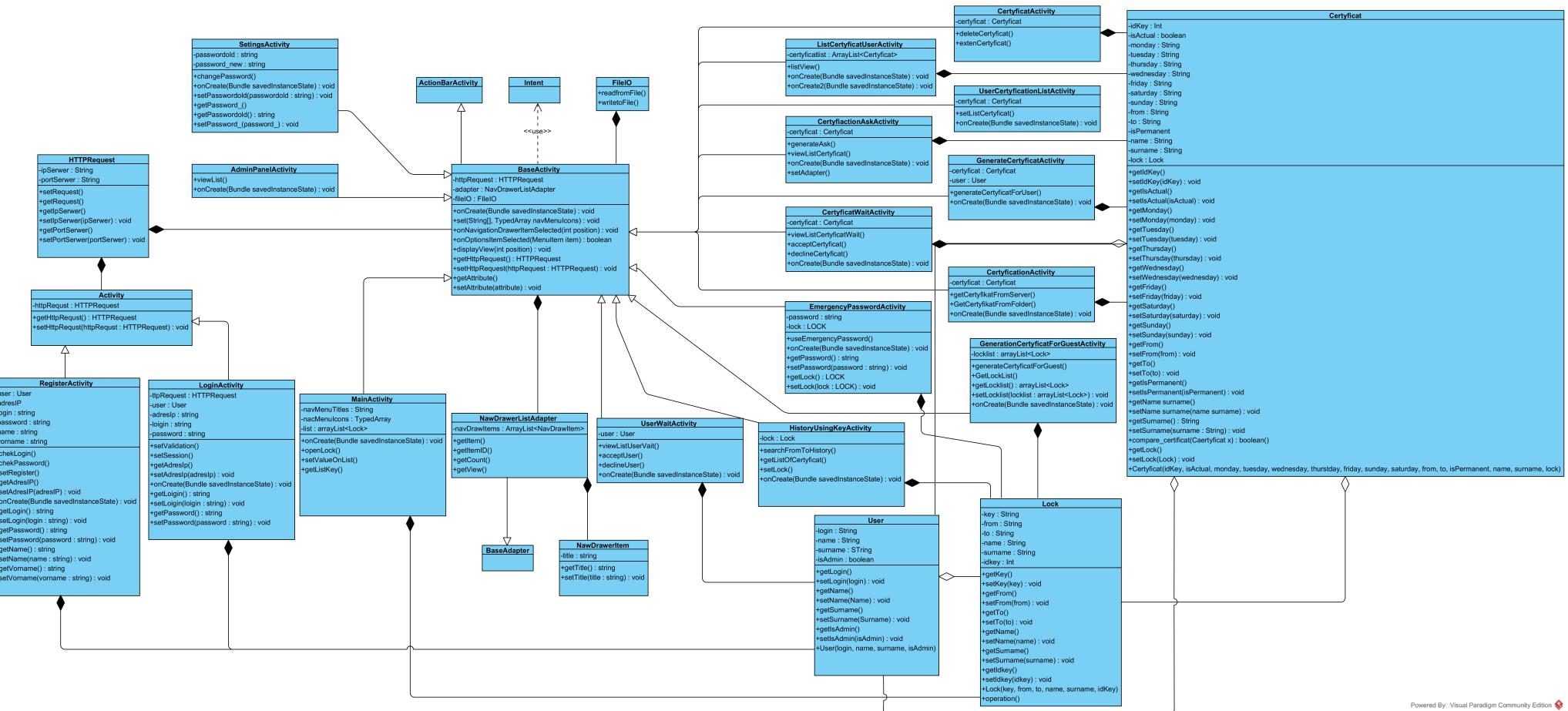
Diagram 5.1 przedstawia opis klas znajdujących się w urządzeniu sterującym.



Rysunek 5.1: Diagram klas urządzenia sterującego

### Aplikacja mobilna

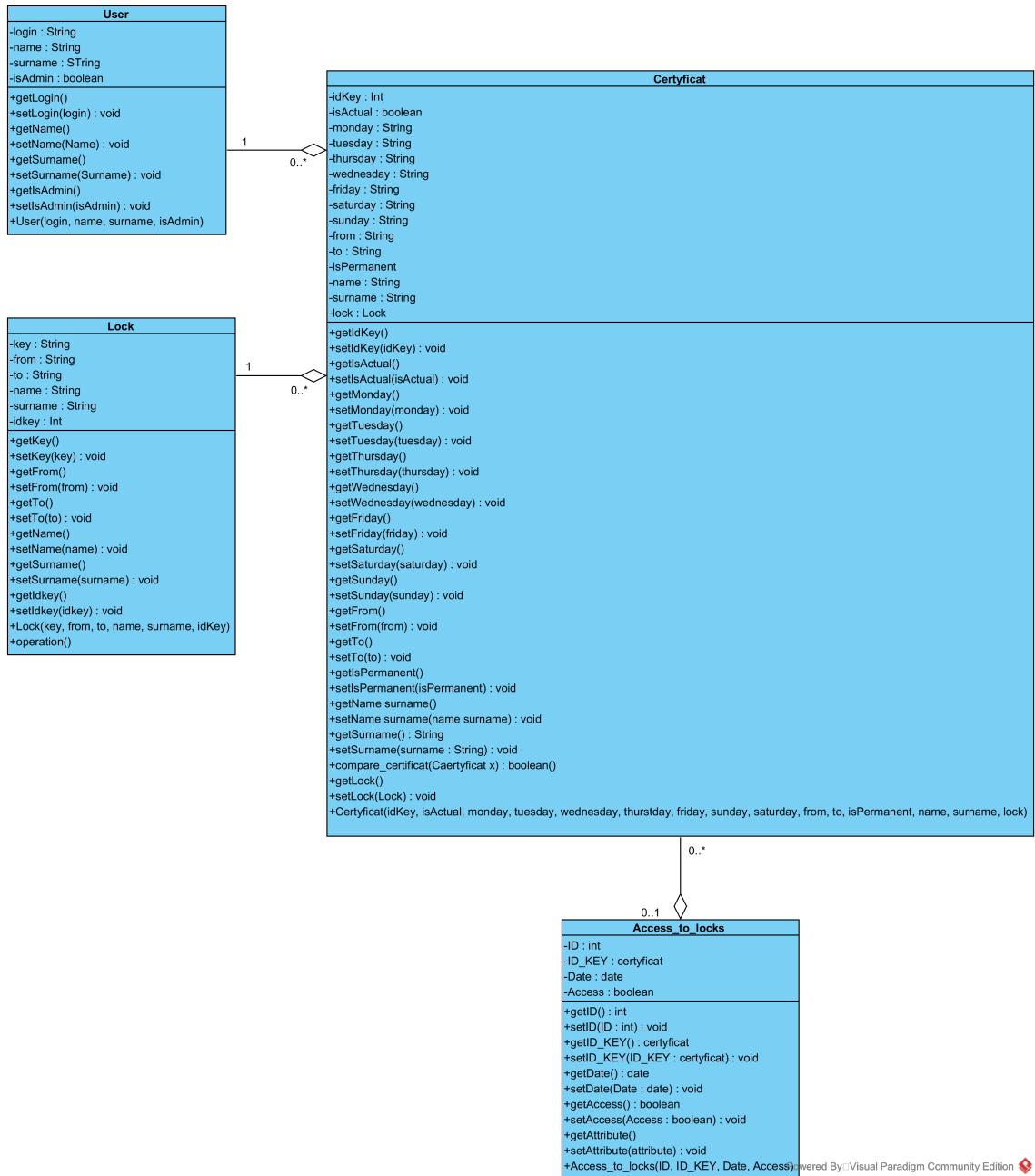
Diagram 5.2 przedstawia opis klas znajdujących się w aplikacji mobilnej.



Rysunek 5.2: Diagram klas aplikacji mobilnej

# Aplikacja serwerowa

Diagram 5.3 przedstawia opis klas znajdujących się w aplikacji serwerowej.



Rysunek 5.3: Diagram klas aplikacji serwerowej

# Rozdział 6

## 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,
- **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,
- **PUBLIC\_KEY** - klucz publiczny użytkownika potrzebny do podpisu cyfrowego,
- **NAME** - imię użytkownika,
- **SURNAME** - nazwisko użytkownika,
- **IS\_ADMIN** - pole booleanskie 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,
- **MAC\_ADDRESS** - adres fizyczny urządzenia sterującego zamkiem,
- **LOCALIZATION** - nieobowiązkowe pole opisujące fizyczne położenie zamka.

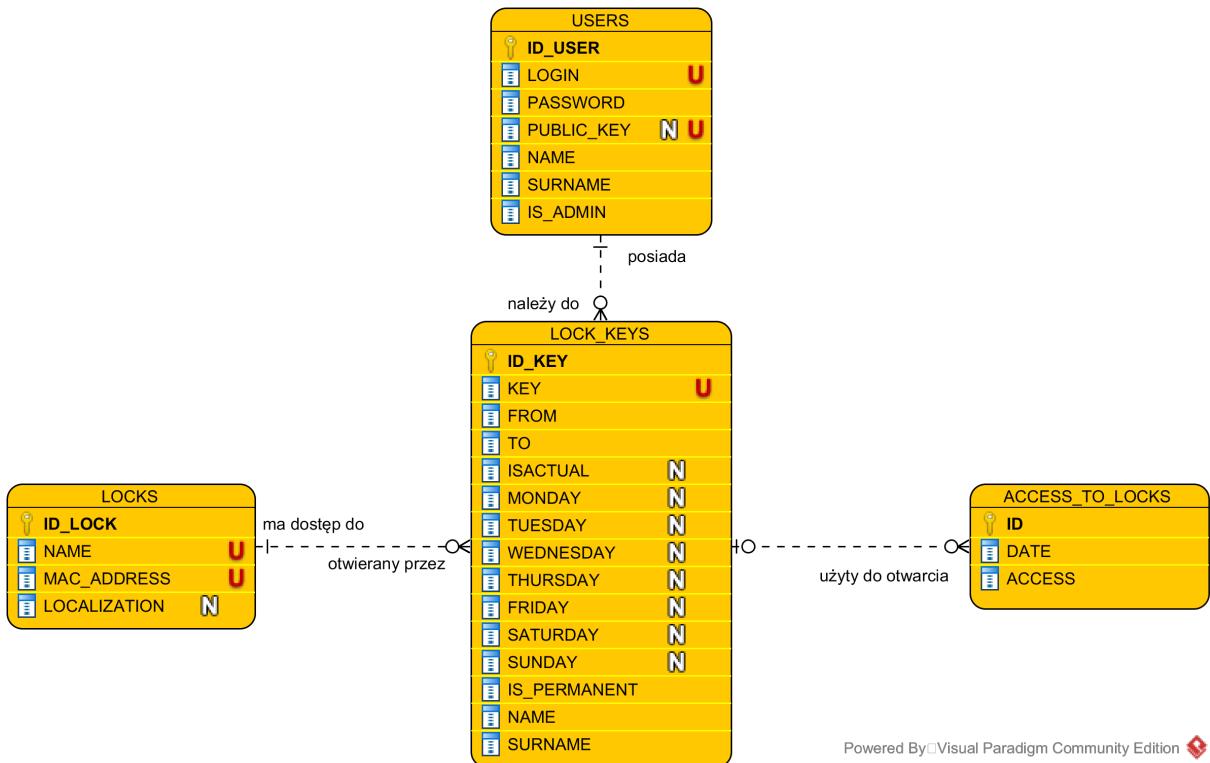
Klucz dostępowy składa się z:

- **ID\_KEY** - unikalny identyfikator (klucz główny) klucza dostępowego składający się z 10 cyfr,
- **ID\_LOCK** - klucz obcy do tabeli przechowującej dostępne zamki,
- **ID\_USER** - klucz obcy do tabeli przechowującej dane użytkownika, jest to pole służące do określenia kto utworzył klucz dostępu,
- **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,
- **MONDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp w poniedziałki,
- **TUESDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp we wtorki,
- **WEDNESDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp w środy,
- **THURSDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp w czwartki,
- **FRIDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp w piątki,
- **SATURDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp w soboty,
- **SUNDAY** - słowne określenie, w których godzinach zostanie przyznany dostęp w niedziele,
- **IS\_PERNAMENT** - zmienna boolowska oznaczająca czy dostęp jest zawsze,
- **NAME** - imię osoby, której dotyczy certyfikat,
- **SURNAME** - nazwisko osoby, której dotyczy certyfikat.

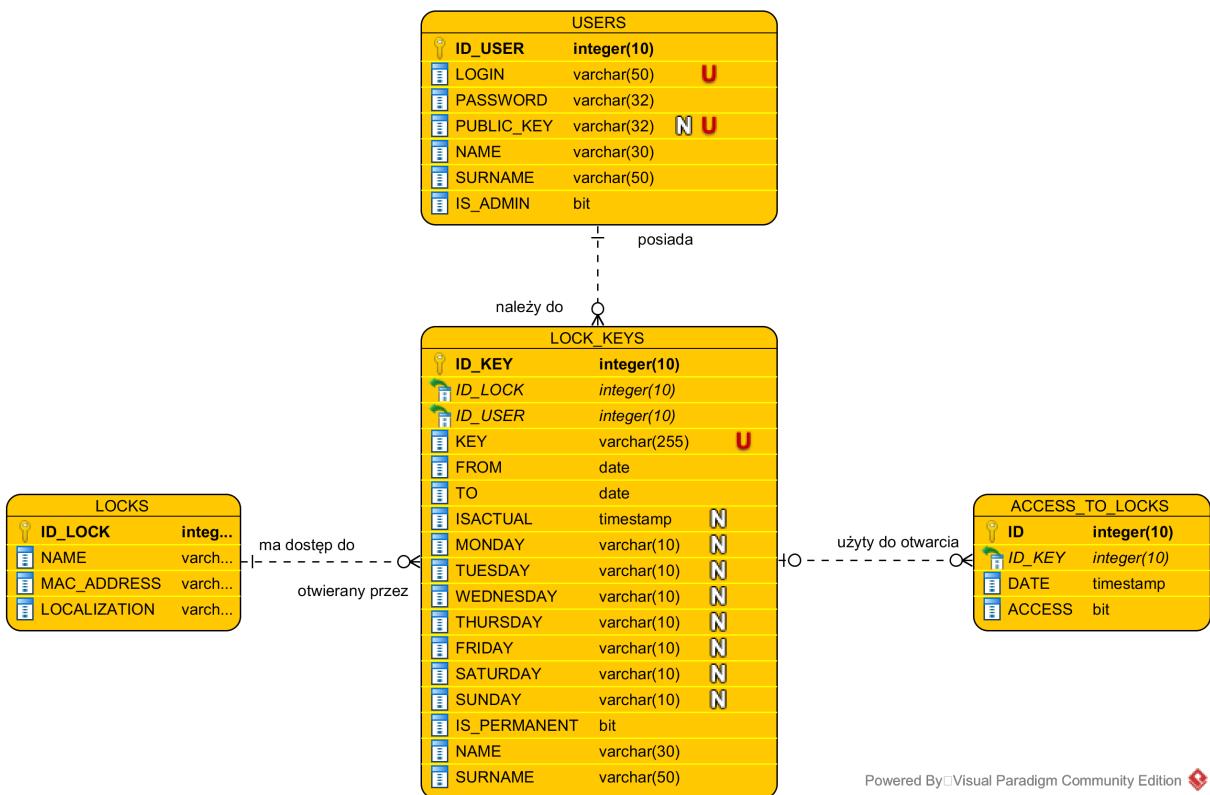
W tabeli archiwizującej akcje na zamku znajdują się takie dane jak:

- **ID** - unikalny identyfikator (klucz główny) akcji wykonanej na certyfikacie składający się z 10 cyfr,
- **ID\_KEY** - klucz obcy do tabeli 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 Rysunkach 6.1 i 6.2.



Rysunek 6.1: Diagram encji bazy danych



Rysunek 6.2: Diagram relacji bazy danych

# Rozdział 7

## Zabezpieczenia systemu

Inteligentny zamek ma na celu zabezpieczanie wejścia do pomieszczeń, dlatego sam system powinien być również bezpieczny. Podstawowymi elementami jakie należy chronić są:

- hasła,
- dane użytkowników,
- ważniejsze funkcje systemu (tj. generowanie nowych kluczy),
- transmisję danych,
- zasilanie urządzenia,
- dostępność do instalacji.

Hasła przechowywane powinny być w bazie danych w postaci skrótu funkcji SHA-3. Każde hasło składać powinno się z przynajmniej 6 znaków i zawierać co najmniej jedną cyfrę.

Ogólne dane użytkowników znajdować się będą w bazie danych, do której jedynie bezpośredni dostęp będzie z aplikacji serwera lub z konta administratora. Hasło dostępu generowane powinny być za pomocą generatora liczb pseudolosowych i być o długości co najmniej 400bitów oraz przechodzić podstawowe testy losowości.

Zabezpieczeniem ważniejszych (wrażliwych) funkcji systemu odbywać się będzie po stronie aplikacji serwerowej. Serwer będzie zezwalał tylko na operacje dozwolone dla danego typu użytkownika. Do wrażliwych funkcji należy przede wszystkim pobieranie danych z bazy danych oraz ich modyfikacje. Inną formą zabezpieczenia operacji będzie podawanie hasła, mimo iż użytkownik będzie zalogowany. Weryfikacja hasła ma na celu ochronę przed chwilowym przejęciem telefonu i wygenerowaniem np. sobie dostępu do zamka.

Każde połączenie internetowe pomiędzy mikrokomputerem Raspberry Pi i serwerem będzie zabezpieczone protokołem IPSec. Komunikacja bluetooth nie będzie szyfrowana, lecz certyfikaty przesyłane będą podpisane cyfrowo kluczem asymetrycznym RSA. Pary kluczy (prywatny i publiczny) będą przypisane do danego certyfikatu i tracące ważność wraz z nim.

Zabezpieczeniem fizycznym systemu przed utratą zasilania z sieci energetycznej, będzie zamontowanie równolegle zasilania baterijnego załączanego w momencie zaniku głównego. Ochroną systemu przed ingerencją w sprzęt będzie umieszczenie zabudowanego mikrokomputera po wewnętrznej stronie pomieszczenia, tak żeby dostęp miały tylko osoby które miały pozwolenie przejść przez drzwi.

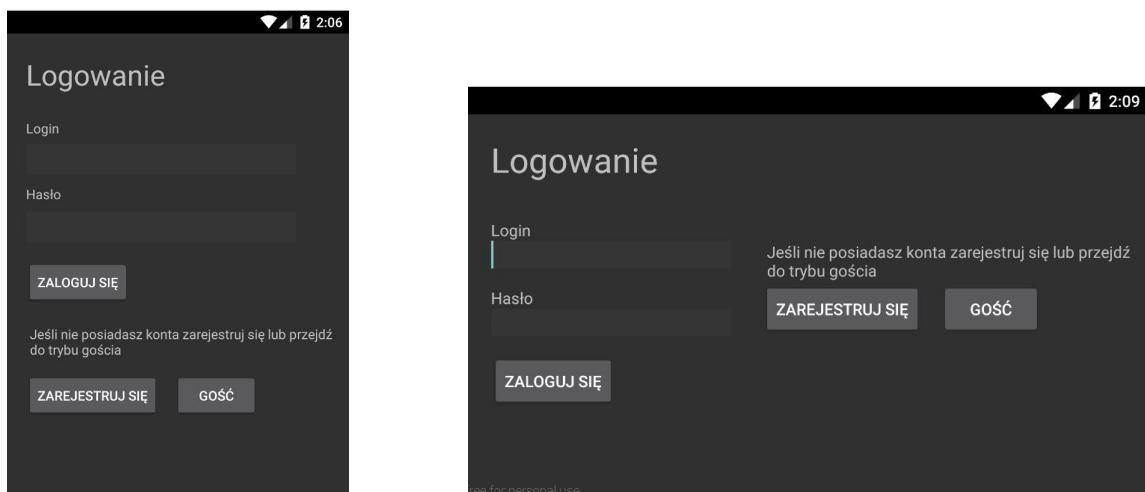
# Rozdział 8

## 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.

### Panel logowania użytkownika

Widok umożliwia zalogowanie się użytkownika do systemu poprzez podanie loginu i hasła w odpowiednie pola, a następnie kliknięcie w przycisk “ZALOGUJ SIĘ”. Jeśli nie posiada się konta, można je utworzyć poprzez przycisk “ZAREJESTRUJ SIĘ” lub przejść do panelu Gościa przyciskiem “GOŚĆ”. (Rysunek 8.1 i 8.2)

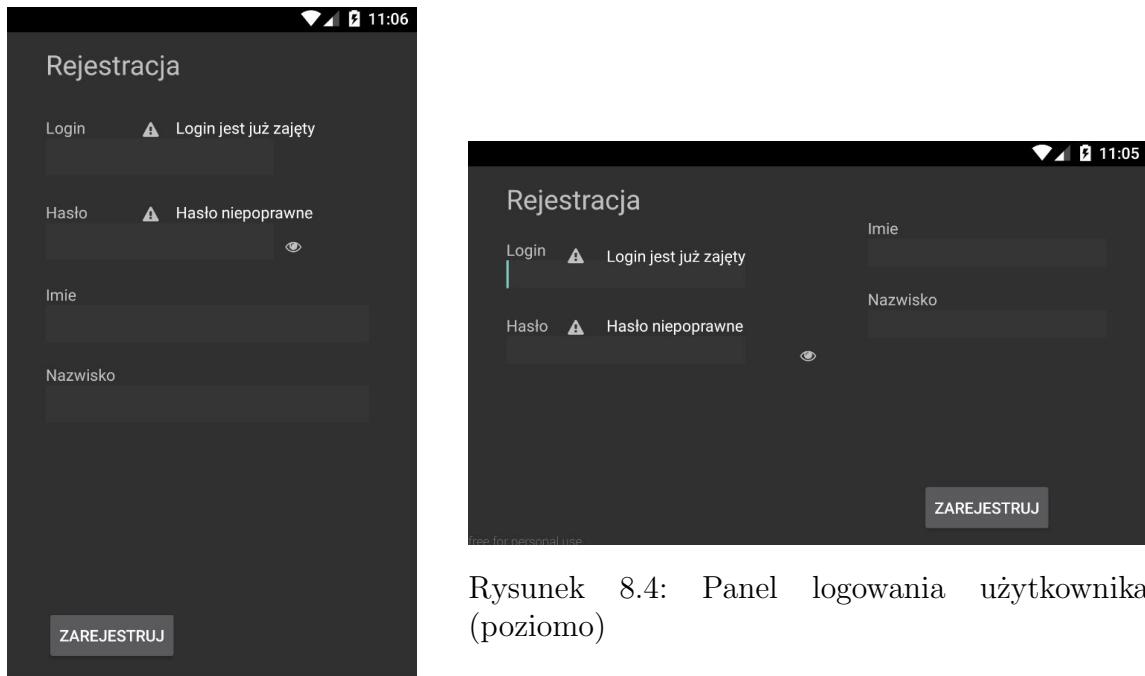


Rysunek 8.2: Panel logowania użytkownika (poziomo)

Rysunek 8.1: Panel logowania użytkownika (pionowo)

## Panel rejestracji użytkownika

Panel rejestracji służy do utworzenia nowego użytkownika poprzez podanie loginu, hasła, imienia i nazwiska. Po upewnieniu się, że wszystkie dane są poprawne, aby zakończyć proces rejestracji, klikamy przycisk “ZAREJESTRUJ”. (Rysunek 8.3 i 8.4)

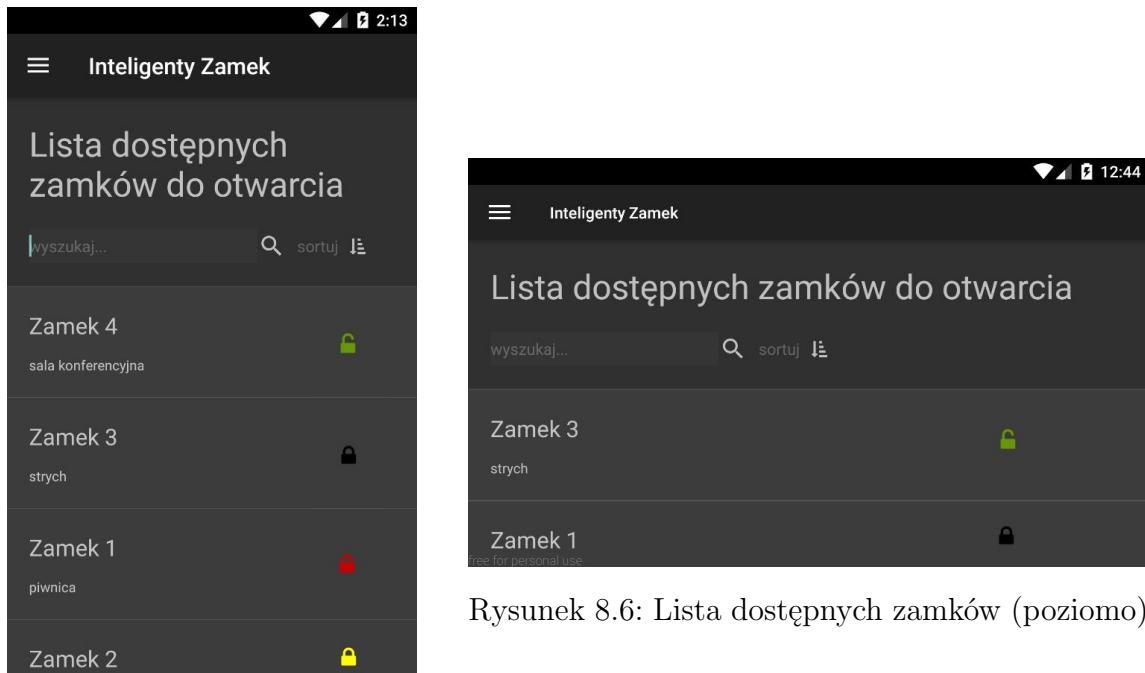


Rysunek 8.4: Panel logowania użytkownika (poziomo)

Rysunek 8.3: Panel logowania użytkownika (pionowo)

## Panel listy zamków

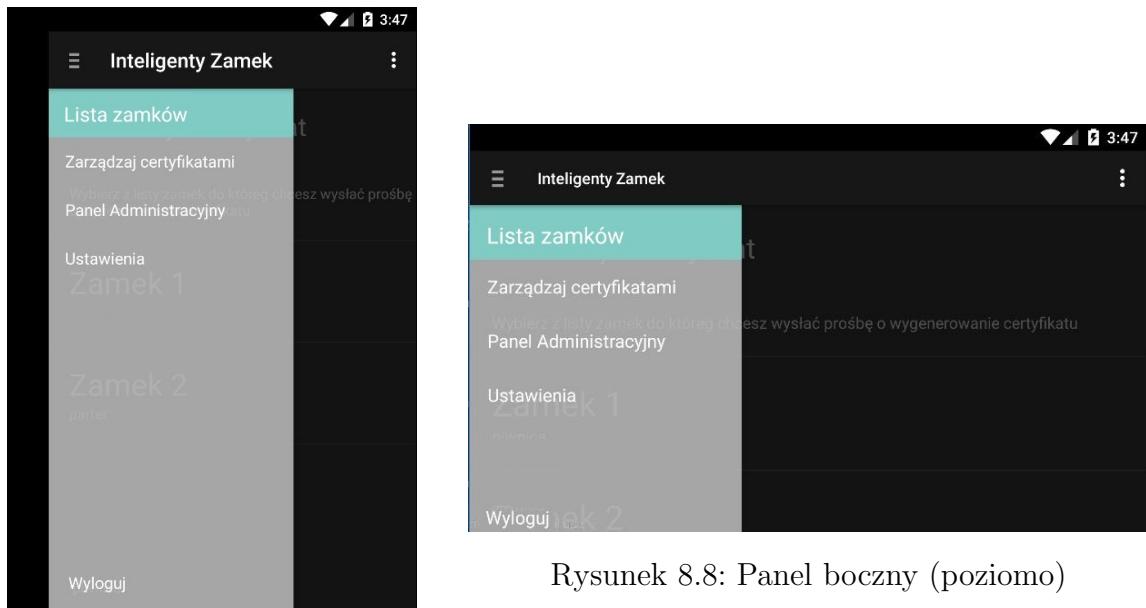
Widok listy dostępnych zamków przedstawia listę nazw zamków do jakich dany użytkownik ma dostęp. Ułatwieniem jest możliwość sortowania wyników i wyszukiwanie po nazwach. Kliknięcie w nazwę zamka powoduje otwarcie zamka. Ustawić można również zamek, który ma być otwierany automatycznie gdy jest się w pobliżu zamka. (Rysunek 8.5 i 8.6)



Rysunek 8.5: Lista dostępnych zamków (pionowo)

## Panel boczny

Panel boczny pozwala na szybkie przełączanie pomiędzy widokami. Chowany jest po lewej stronie ekranu. Umożliwia przechodzenie odpowiednio do listy zamków, zarządzania certyfikatami, panelu administracyjnego oraz ustawień. (Rysunek 8.7 i 8.8)

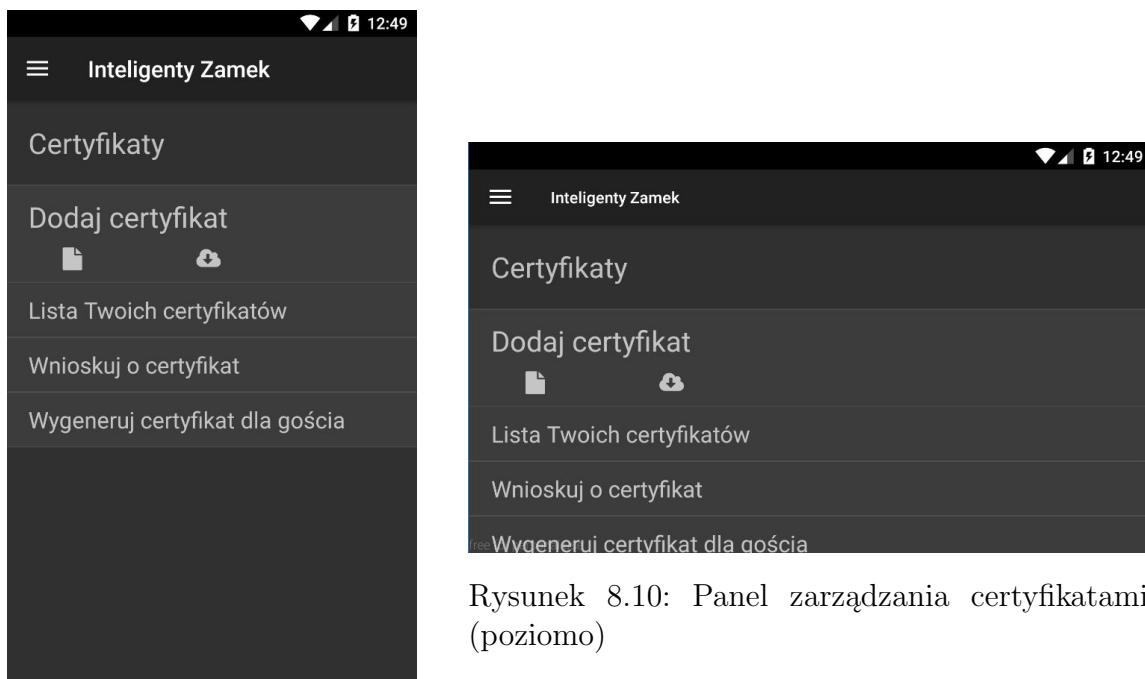


Rysunek 8.8: Panel boczny (poziomo)

Rysunek 8.7: Panel boczny (pionowo)

## Panel zarządzania certyfikatami

Panel zarządzania certyfikatami umożliwia wybór funkcji dodania certyfikatu w dwóch wariantach (rozwijana lista) - dodania z pliku lub ściągnięcia z serwera. Kolejne pozycje to lista posiadanych certyfikatów, wysłanie wniosku o utworzenie nowego certyfikatu oraz ostatnia opcja to wygenerowanie certyfikatu dla gościa. (Rysunek 8.9 i 8.10)

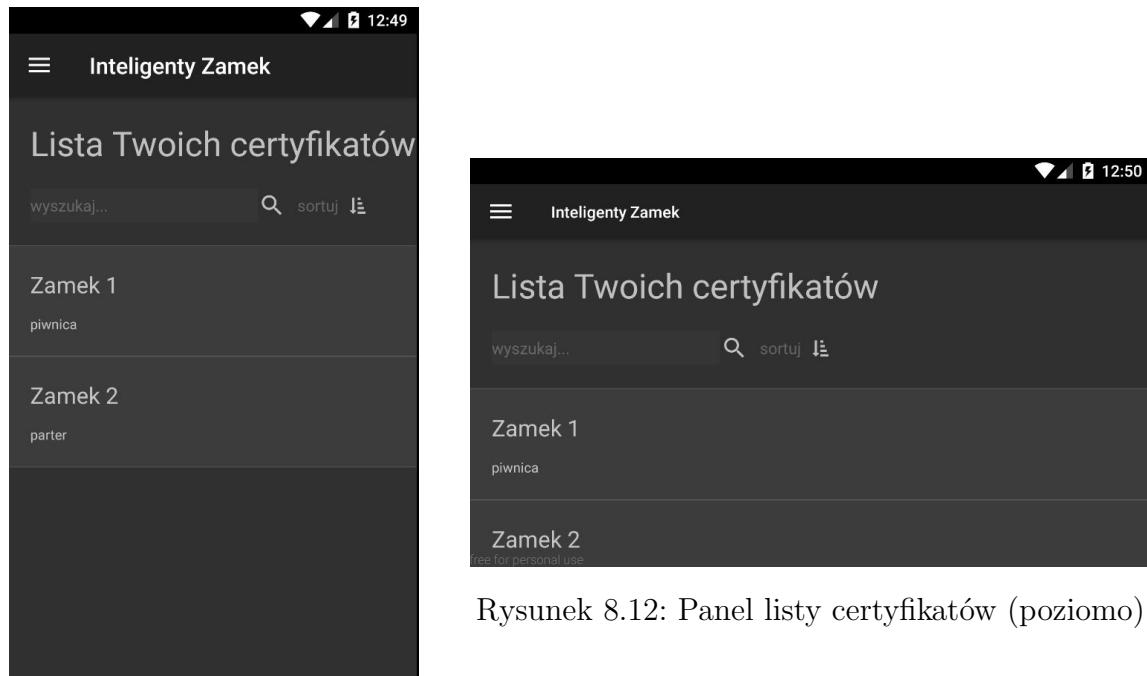


Rysunek 8.9: Panel zarządzania certyfikatami (pionowo)

Rysunek 8.10: Panel zarządzania certyfikatami (poziomo)

## Panel listy certyfikatów

Panel listy certyfikatów, jest listą aktualnych certyfikatów należących do użytkownika. Kliknięcie w dany certyfikat przenosi do widoku szczegółowego związanego z operacjami na certyfikacie. (Rysunek 8.11 i 8.12)

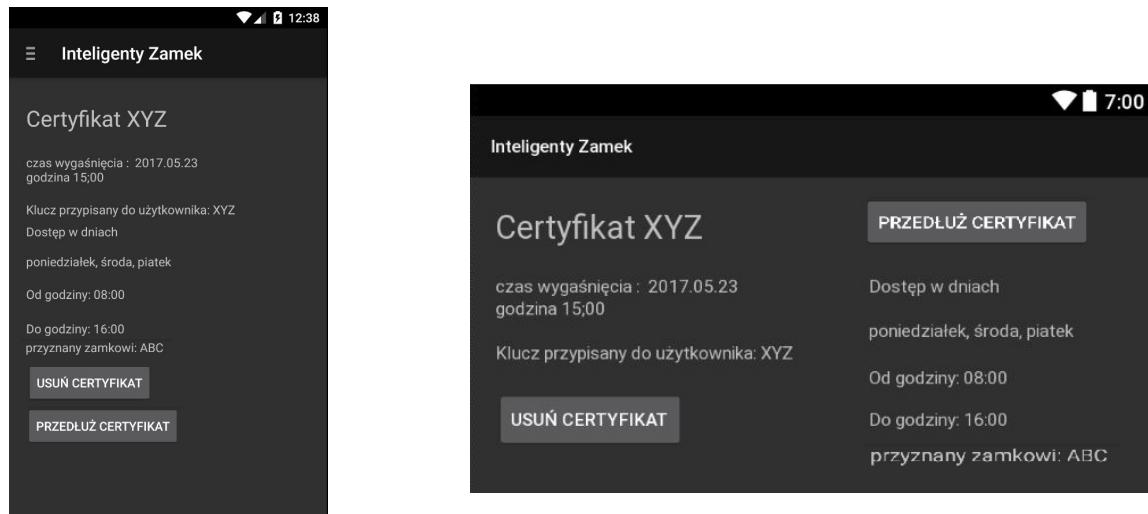


Rysunek 8.11: Panel listy certyfikatów (pionowo)

Rysunek 8.12: Panel listy certyfikatów (poziomo)

## Panel certyfikatu

Panel certyfikatu u góry zawiera nazwę certyfikatu, poniżej informacje o dacie wygaśnięcia, którego zamku dotyczy oraz w jakim czasie przyznaje dostęp. Na dole dostępne są dwa przyciski pozwalające usunąć certyfikat lub wysłać prośbę o przedłużenie ważności. (Rysunek 8.13 i 8.14)

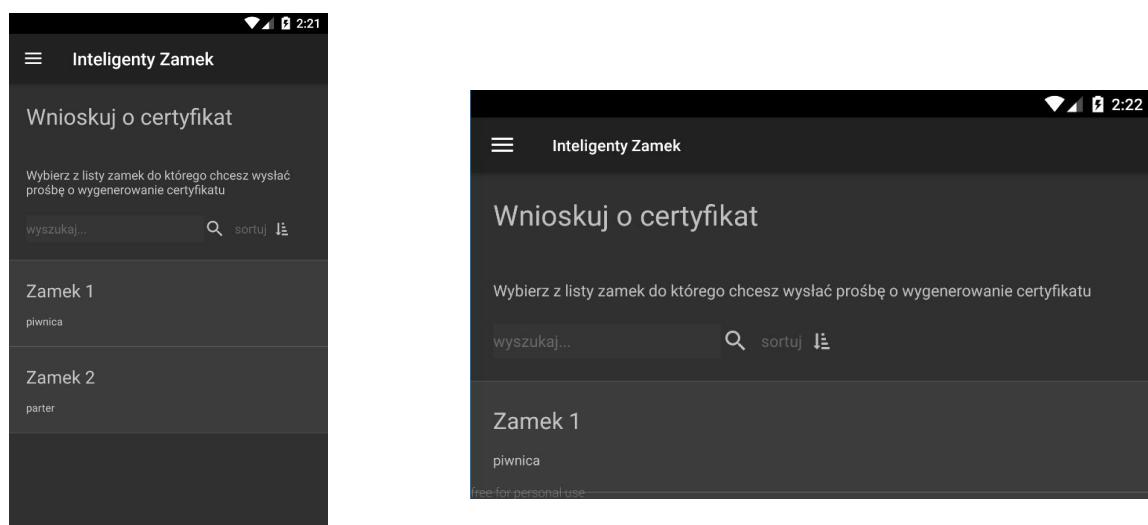


Rysunek 8.14: Panel certyfikatu (poziomo)

Rysunek 8.13: Panel certyfikatu (pionowo)

## Panel wnioskowania o certyfikat

Panel wnioskowania o certyfikat polega na wybraniu z listy wszystkich zamków, konkretnego do którego chcemy uzyskać dostęp i wysłać wniosek o przydzielenie dostępu. (Rysunek 8.15 i 8.16)

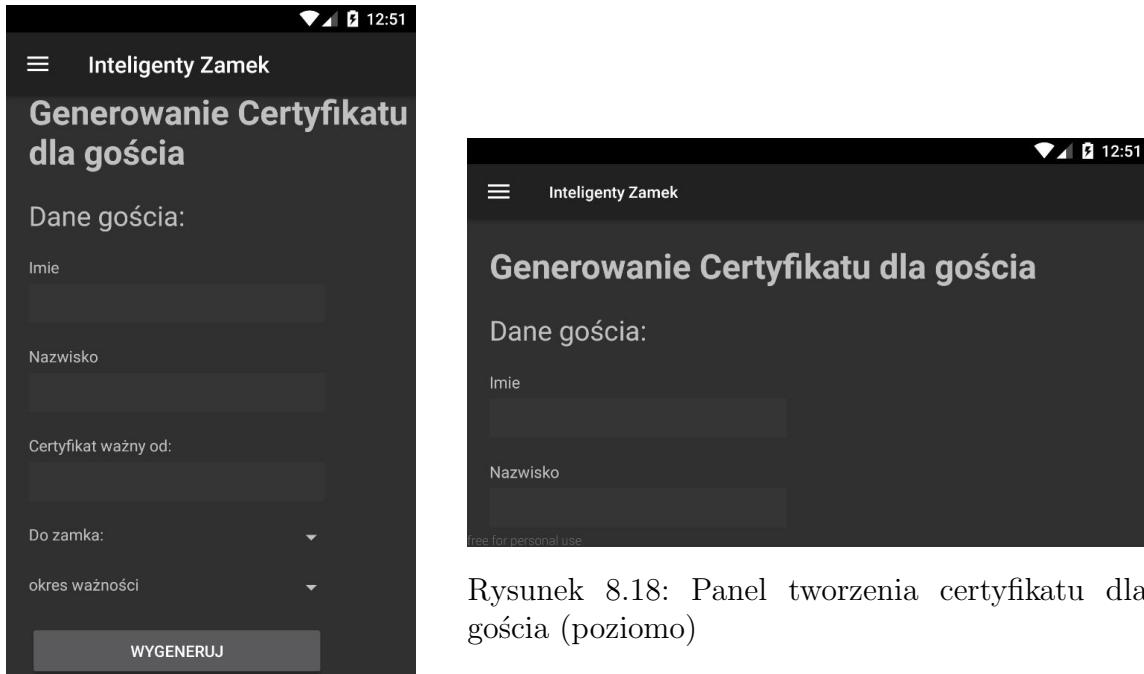


Rysunek 8.16: Panel wnioskowania o certyfikat

Rysunek 8.15: Panel wnioskowania o certyfikat (poziomo)

## Panel tworzenia certyfikatu dla gościa

Panel tworzenia certyfikatu gościa składa się z danych gościa (imię i nazwisko), daty, od której obowiązuje certyfikat, zamka którego dotyczy oraz jak długo pozostanie ważny (wybór z rozwijanej listy). Aby zakończyć proces generowania należy kliknąć przycisk “WYGENERUJ”. (Rysunek 8.17 i 8.18)



Rysunek 8.17: Panel tworzenia certyfikatu dla gościa (pionowo)

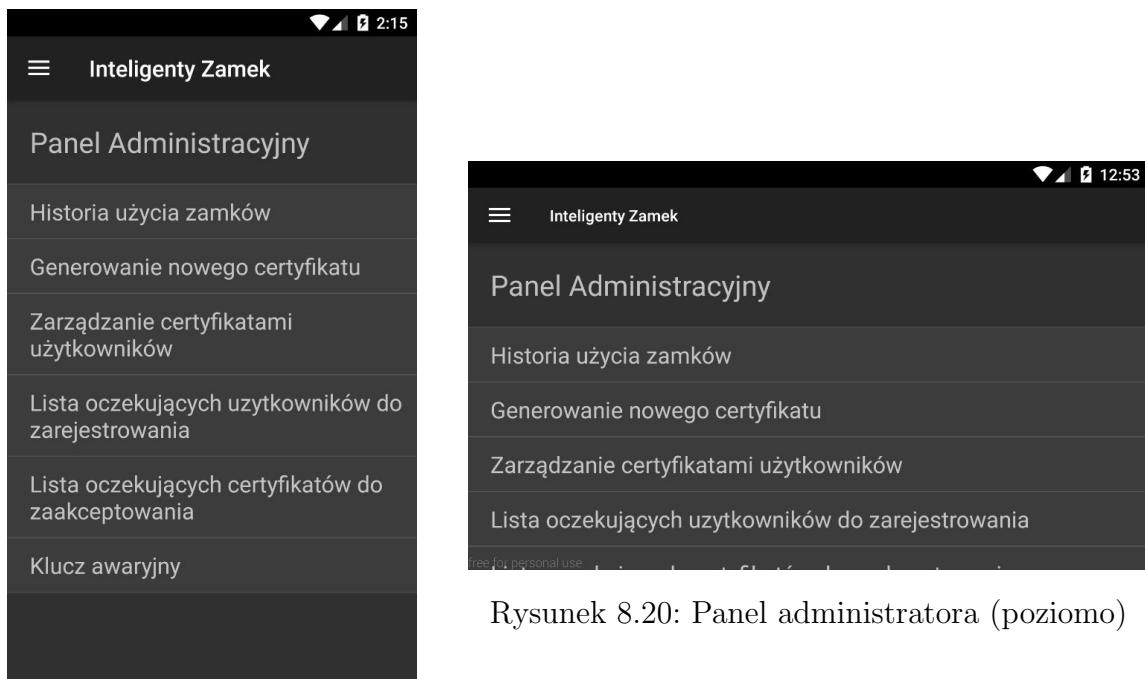
Rysunek 8.18: Panel tworzenia certyfikatu dla gościa (poziomo)

# Panel administratora

W panelu administratora znajdują się 6 przycisków do administrowania systemem zamków:

- „Historia użycia zamków”
- „Generowanie nowego certyfikatu” ,
- „Zarządzanie certyfikatami użytkowników” ,
- „Lista oczekujących użytkowników do zarejestrowania” ,
- „Lista oczekujących certyfikatów do zaakceptowania” ,
- „Klucz awaryjny” .

Po kliknięciu każdego przycisku przechodzi się do nowego odpowiadającego widoku. (Rysunek 8.19 i 8.20)

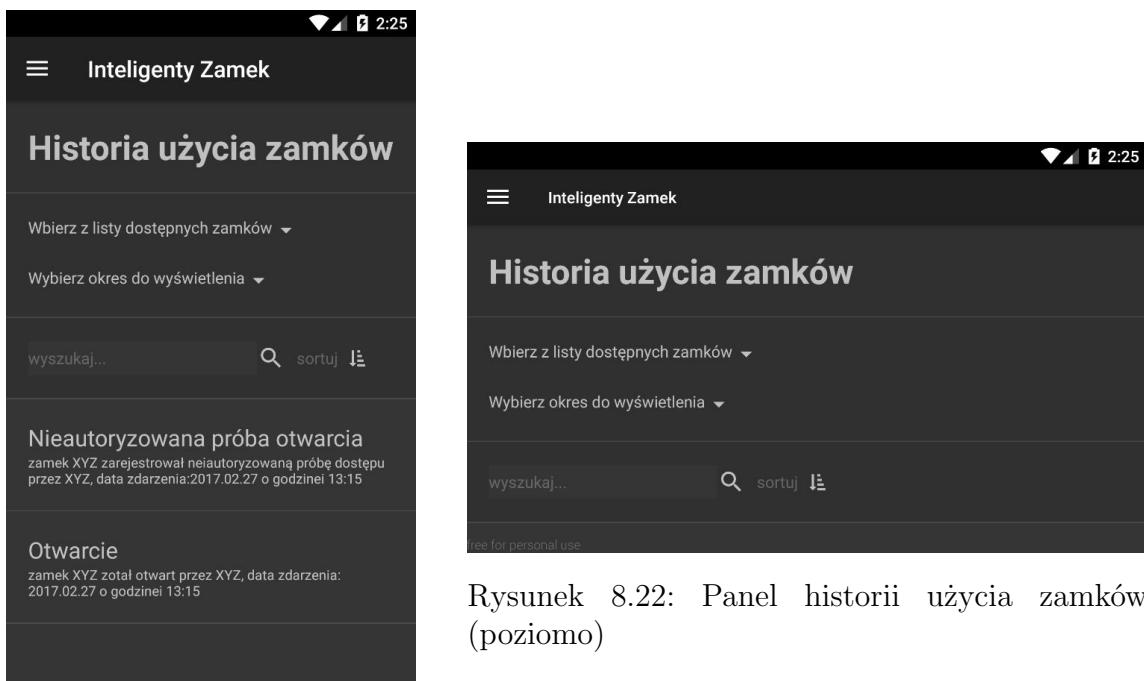


Rysunek 8.19: Panel administratora (pionowo)

Rysunek 8.20: Panel administratora (poziomo)

## Panel historii użycia zamków

Panel historii użycia zamków składa się z rozwijanej listy wszystkich zamków znajdujących się w systemie, służącej do filtrowania wyników. Dodatkowo wybrać można okres z jakiego wyniki są prezentowane. Pozycje opisane kolorem czerwonym dotyczą operacji odrzucanych, białe przyznających dostęp do zamka. (Rysunek 8.21 i 8.22)

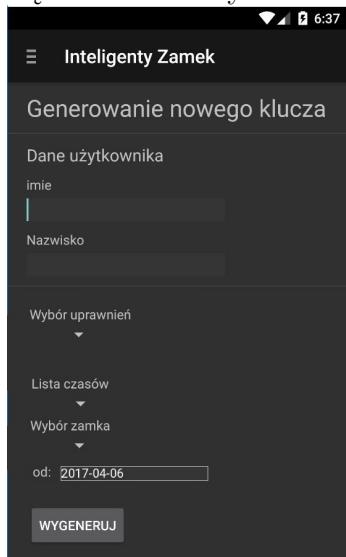


Rysunek 8.21: Panel historii użycia zamków (pionowo)

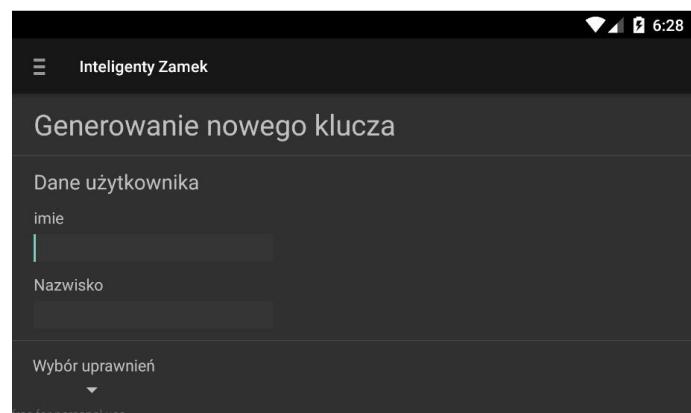
Rysunek 8.22: Panel historii użycia zamków (poziomo)

# Panel generowania nowego certyfikatu (administrator)

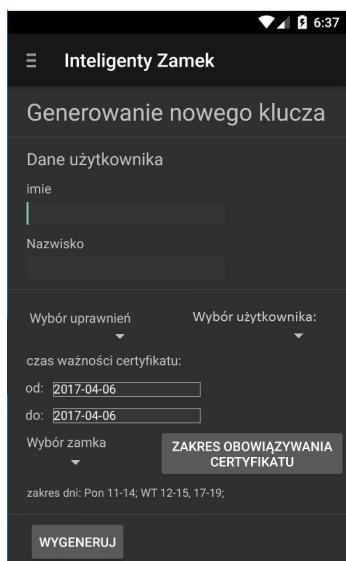
Panel generowanie nowego certyfikatu (administrator) służy do tworzenia nowych certyfikatów przez administratora. W pierwszych polach podaje się imię i nazwisko kogo dotyczy certyfikat. W następnych administrator wybiera typ certyfikatu (gość / użytkownik zalogowany). W zależności od wyboru dostępne są określone pola. Wspólne to od kiedy jest dostęp i jakiego zamka dotyczy. Dla gościa należy wybrać na jak długo jest dostęp, a dla zalogowanego użytkownika do kiedy jest ważność i w jakich godzinach danych dni tygodni ma dostęp (po kliknięciu przycisku "ZAKRES OBOWIĄZYWANIA CERTYFIKATU" przechodzi się do widoku wyboru zakresu godziny). (Rysunek 8.23, 8.24, 8.25 i 8.26)



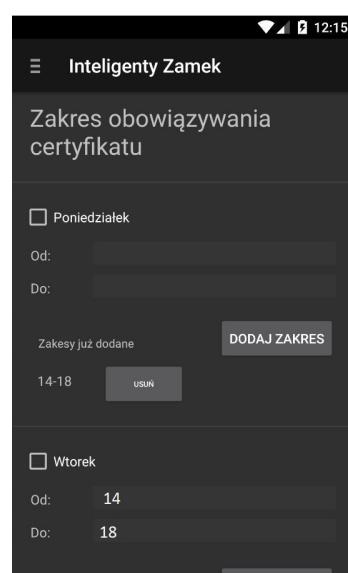
Rysunek 8.23: Panel generowania nowego klucza dla gościa (administrator) (pionowo)



Rysunek 8.24: Panel generowania nowego klucza (administrator)(poziomo)



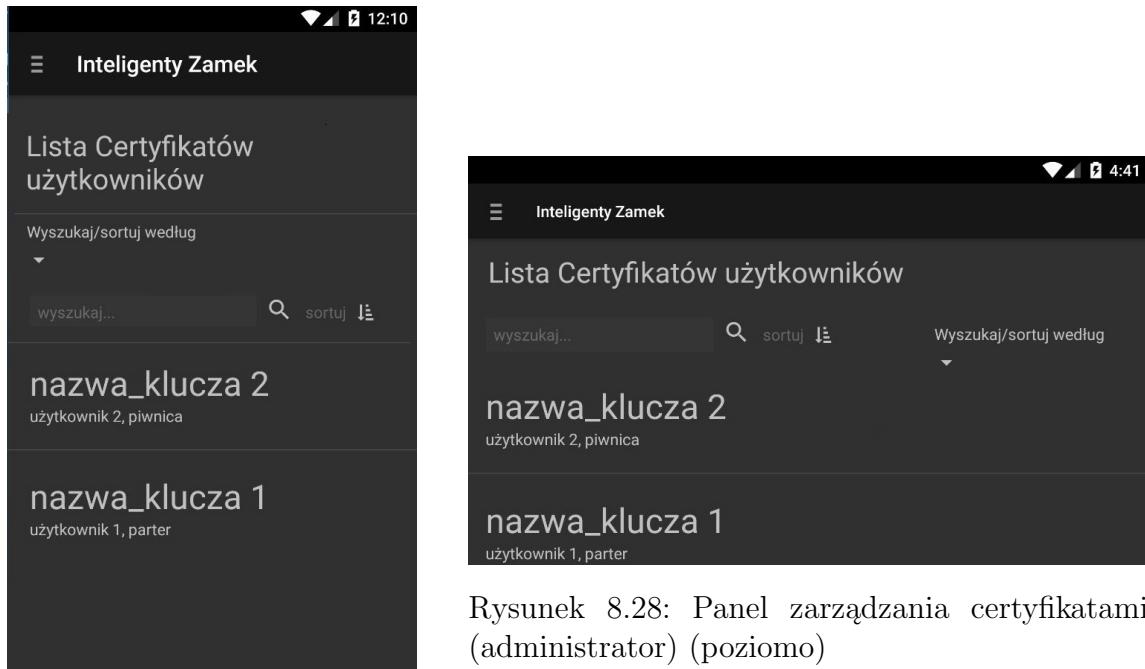
Rysunek 8.25: Panel generowania nowego klucza dla użytkownika zalogowanego (administrator)(poziomo)



Rysunek 8.26: Panel panel wyboru zakresu certyfikatu

# Panel zarządzania certyfikatami (administrator)

Panel zarządzania certyfikatami użytkowników (administrator) jest widokiem tylko wszystkich aktywnych certyfikatów w systemie. Administrator klikając na pozycję przechodzi do panelu certyfikatu opisanego wyżej. Tam może usunąć dostęp lub go przedłużyć. Ułatwieniem jest możliwość wyboru typu sortowania. (Rysunek 8.27 i 8.28)

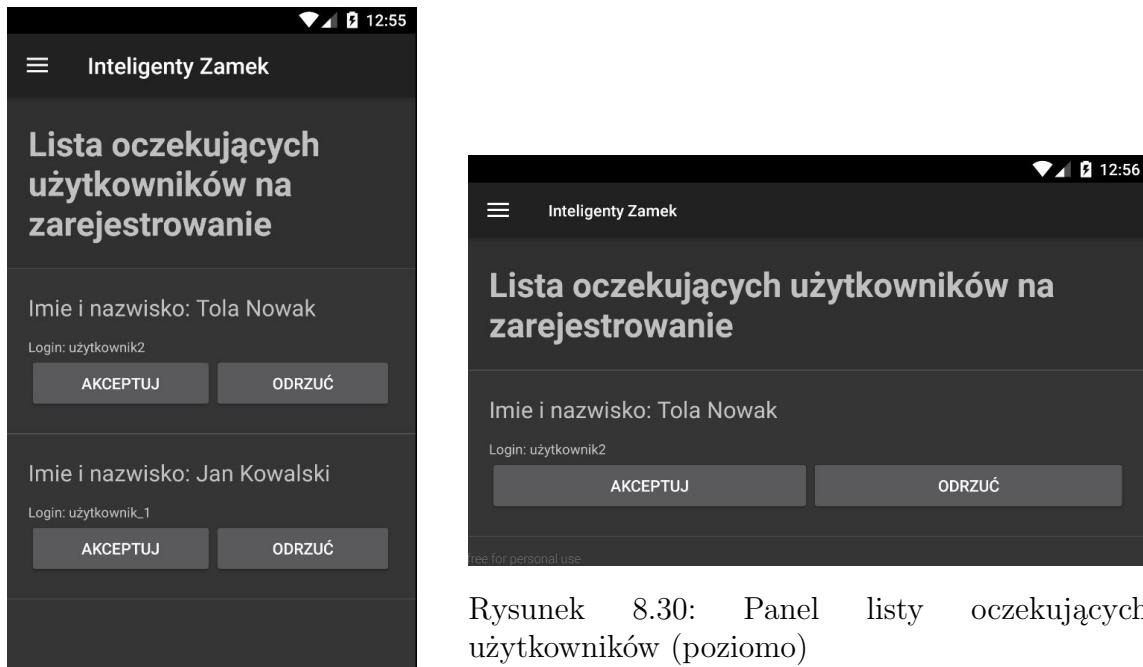


Rysunek 8.27: Panel zarządzania certyfikatami (administrator) (pionowo)

Rysunek 8.28: Panel zarządzania certyfikatami (administrator) (poziomo)

## Panel listy oczekujących użytkowników do rejestracji

Panel listy oczekujących użytkowników jest listą wszystkich gości, którzy ubiegają się o zarejestrowanie. PO kliknięciu w odpowiednią pozycję pojawiają się dwie opcje: “AKCEPTUJ” lub “ODRZUĆ”. (Rysunek 8.29 i 8.30)

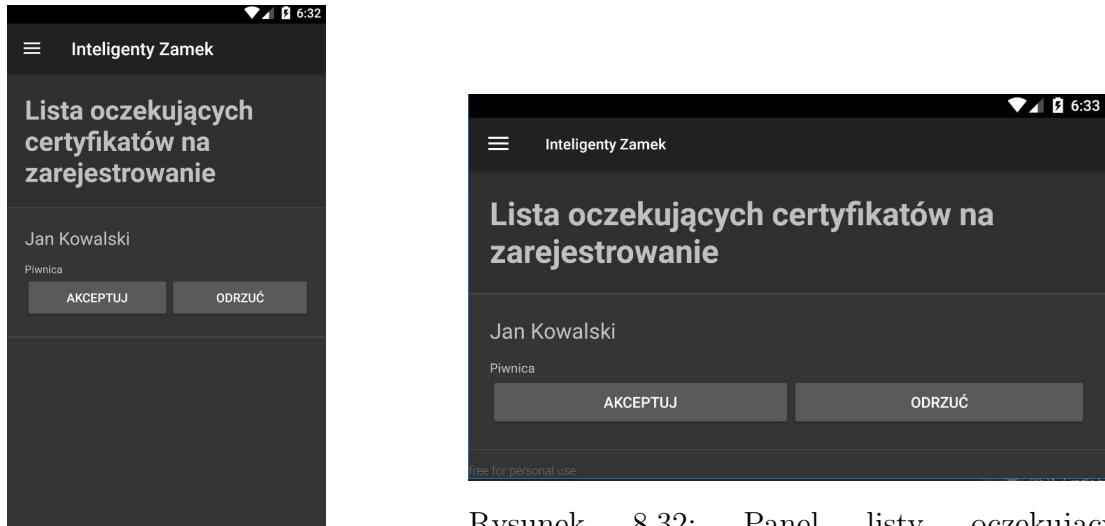


Rysunek 8.29: Panel listy oczekujących użytkowników (pionowo)

Rysunek 8.30: Panel listy oczekujących użytkowników (poziomo)

# Panel listy oczekujących certyfikatów do wygenerowania

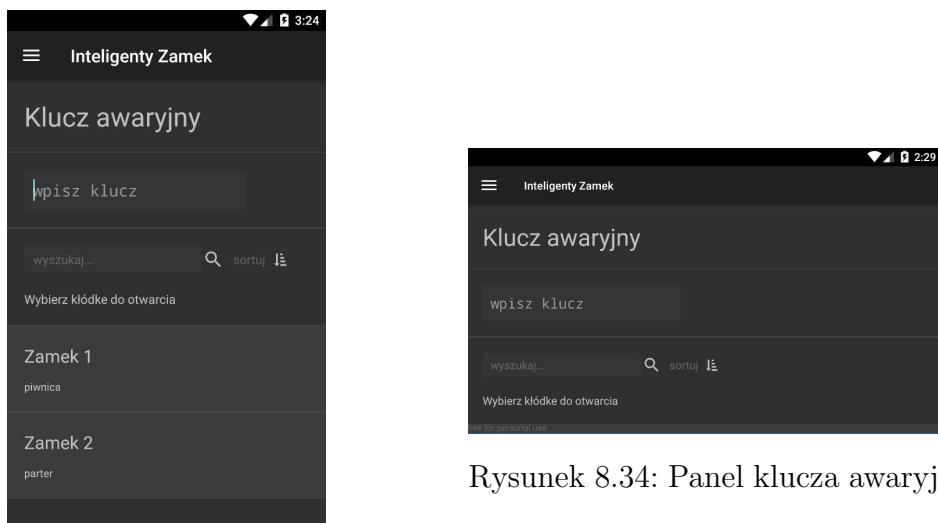
Panel listy oczekujących certyfikatów jest listą wszystkich certyfikatów, które ubiegają się o akceptację administratora. Po kliknięciu w odpowiednią pozycję pojawiają się dwie opcje: "AKCEPTUJ" lub "ODRZUĆ". (Rysunek 8.31 i 8.32)



Rysunek 8.31: Panel listy oczekujących certyfikatów (pionowo)  
Rysunek 8.32: Panel listy oczekujących certyfikatów (poziomo)

# Panel klucza awaryjnego

Panel klucza awaryjnego służy do otwierania zamków w sytuacji awaryjnej, np. gdy zamek nie ma dostępu do Internetu lub innej awarii. (Rysunek 8.33 i 8.34)

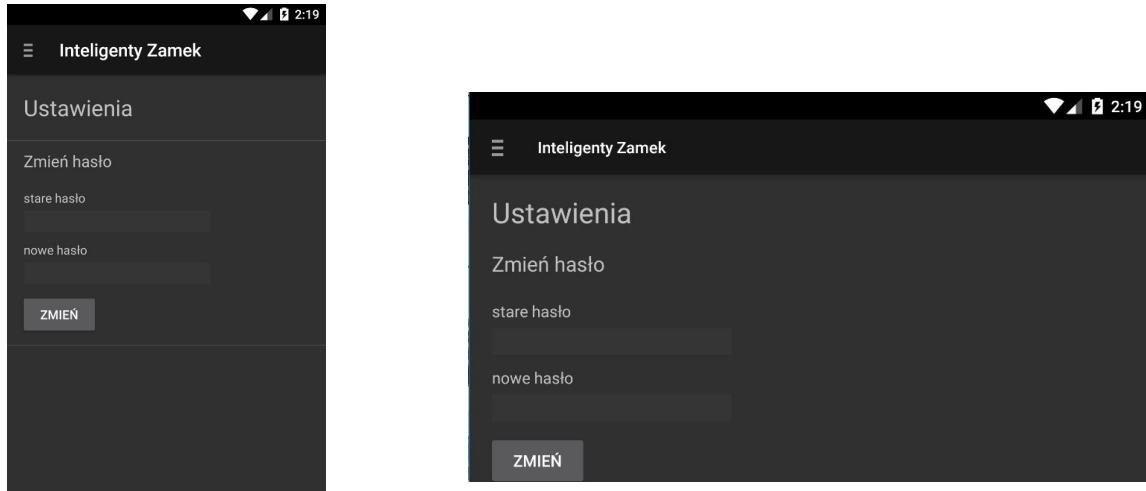


Rysunek 8.33: Panel klucza awaryjnego (pionowo)

Rysunek 8.34: Panel klucza awaryjnego (poziomo)

## Panel ustawień konta

W panelu ustawień użytkownik może zmienić hasło do swojego konta. Wymagane jest podanie starego hasła, a następnie nowego. Aby zakończyć czynność klikamy na “ZMIEŃ”. (Rysunek 8.35 i 8.36)



Rysunek 8.36: Panel ustawień konta (poziomo)

Rysunek 8.35: Panel ustawień konta (pionowo)

# Rozdział 9

## Testowanie aplikacji

Główne zagadnienia które należy przetestować:

- generator liczb pseudolosowych,
- zasilanie urządzeń,
- wydajność systemu,
- komunikatywność aplikacji mobilnej (człowiek-system oraz system-człowiek),
- główne funkcje systemu podczas operacji otwierania/zamykania zamków oraz dystrybucji kluczy,

Generator liczb pseudolosowych powinien być przetestowany pod kontem testów losowości, takie jak test pojedynczych bitów, serii, długiej serii, pokerowy oraz dwójkowy. Próby wykonywane powinny być wielokrotnie, dla wielu utworzonych próbek.

System oparty jest o poprawne działanie sprzętu, należy więc przetestować reakcję urządzeń na spadki napięcia podczas spoczynku, pracy zwykłej oraz przy dużym obciążeniu. Raspberry Pi w razie awarii zasilania sieciowego, przełączy się na zasilanie baterijne. Poprzez częste przełączanie sposobu zasiania, np. przy użyciu tranzystora należy sprawdzić, czy w trakcie przejścia urządzenia nie wyłączy się z powodu krótkiego całkowitego zaniku napięcia. Test zasilania zweryfikować powinien również czas działania urządzeń pod zasilaniem baterijnym.

Testem wydajności systemu jest sprawdzenie odporności na przeciążenia systemu, choćby przy wielu żądaniach do serwera. System zakłada obsługę wielu zamków przy jednym serwerze, dlatego należy zweryfikować działanie bazy danych oraz aplikacji serwerowej pod kątem wielu zapytań jednocześnie. Wykonać taki test można przy pomocy aplikacji Apache JMeter. Program zbada serwer pod kątem szybkości oraz odpowiedzi httprequest.

Jedynym elementem kontaktującym się z użytkownikiem jest aplikacja mobilna. Program powinien w pierwszej fazie testów zostać sprawdzony automatycznie przy pomocy narzędzia Robotium, pozwalającym na utworzeniu sekwencji operacji jakie mają po sobie nastąpić, aby osiągnąć zamierzony efekt. Scenariusz testu obejmować powinien reakcję na:

- podanie błędnych kluczy dostępowych podczas procesu otwierania zamka,
- zlego lub przedawnionego klucza prywatnego,
- braku połączenia z internetem,
- braku połączenia bluetooth,

- błędnego logowania użytkownika,
- niewłaściwych danych przy rejestracji użytkownika.

Po przejściu aplikacji przez testy automatyczne, należy przekazać aplikację użytkownikom, w pierwszej kolejności posiadającym wiedzę informatyczną oraz znającym założenia systemu. Użytkownicy wyrażają opinie na temat działania, czytelności aplikacji podczas ankiety. Następnym krokiem jest testowanie aplikacji przez osoby nie posiadające wiedzy informatycznej, lecz zaznajomione z tematyką systemu. Ostatnim krokiem testów aplikacji są testy przez użytkowników bez wiedzy informatycznej oraz bez szczegółowej wiedzy o tematyce projektu. Każdy etap musi uzyskać pewien próg pozytywnych ocen, przy braku zdania testu należy wrócić do początku. Ankieta zawierać powinna pytania na temat estetyki aplikacji, komunikatywności z użytkownikiem.

Testowanie głównej funkcji systemu, tzn. otwierania/zamykania zamka oraz dystrybucji kluczy dostępowych zrealizowane zostanie podczas wcześniej wymienionych testów. Dodatkowo, również przy użyciu narzędzia Robotium wykonane zostaną scenariusze przykładowych użyć systemu. Przykładowy scenariusz:

*Użytkownik zalogowany (UZ) wnioskuje o klucz dostępowy do zamka A, B i C na okres tygodnia. Administrator (AD) akceptuje wniosek dla zamka B i C, dla zamka A odrzuca. UZ tworzy klucz gościa do zamka B jednorazowy (zał. powodzenie). UZ próbuje otworzyć zamek B (zał. powodzenie). AD blokuje dostęp UZ do zamka B. Gość próbuje otworzyć zamek B (zał. powodzenie). UZ próbuje otworzyć ponownie zamek B (zał. niepowodzenie). Gość próbuje otworzyć zamek B (zał. niepowodzenie). UZ próbuje utworzyć klucz dostępu dla gościa do zamka B na 10 minut (zał. niepowodzenie). UZ próbuje utworzyć klucz dostępu dla gościa do zamka C na 10 minut (zał. powodzenie). Gość próbuje otworzyć zamek C (zał. powodzenie). Gość próbuje ponownie otworzyć zamek C po upływie 10 minut (zał. niepowodzenie). UZ próbuje otworzyć zamek C po upływie tygodnia od utworzenia klucza dostępu (zał. niepowodzenie).*

Każda operacja wykonana w scenariuszy będzie mierzona ile czasu zajmuje, pozwoli to na określenie czasu oczekiwania aplikacji na odpowiedzi, w celu optymalizacji działania.