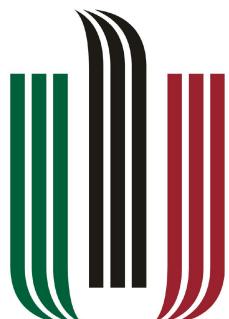


*Akademia Górnictwo - Hutnicza
im. Stanisława Staszica w Krakowie
al. A. Mickiewicza 30; 30-059 Kraków*



AGH



*Wydział Informatyki, Elektroniki
i Telekomunikacji
Bank Pekao S. A. Kraków
65 1240 2294
1111 0010 4756 7266*

Inżynieria Oprogramowania
Dokumentacja techniczna projektu

TheBileT

Autorzy:

Kawula Andrzej
Kowal Marcin
Marchewka Jacek
Mirosław Janusz
Nowicki Tomasz
Rodzoń Daniel

2014/2015

Spis treści

[1. Wprowadzenie](#)

[2. Diagram Use Case](#)

[2.1 Opis diagramu](#)

[2.1.1. User](#)

[2.1.2. City Guard](#)

[2.1.3. Administrator](#)

[3. Diagramy Sekwencji](#)

[3.1 User przed zalogowaniem](#)

[3.2 Użytkownik po zalogowaniu](#)

[3.3 Strażnik przed zalogowaniem](#)

[3.4 Strażnik po zalogowaniu](#)

[3.5 Administrator](#)

[4. Diagram bazy danych](#)

[4.1 Ogólny opis tabel i relacji](#)

[5. Polecenia służące do połączenia z web servicem](#)

[6. Testowanie połączenia z bazą danych](#)

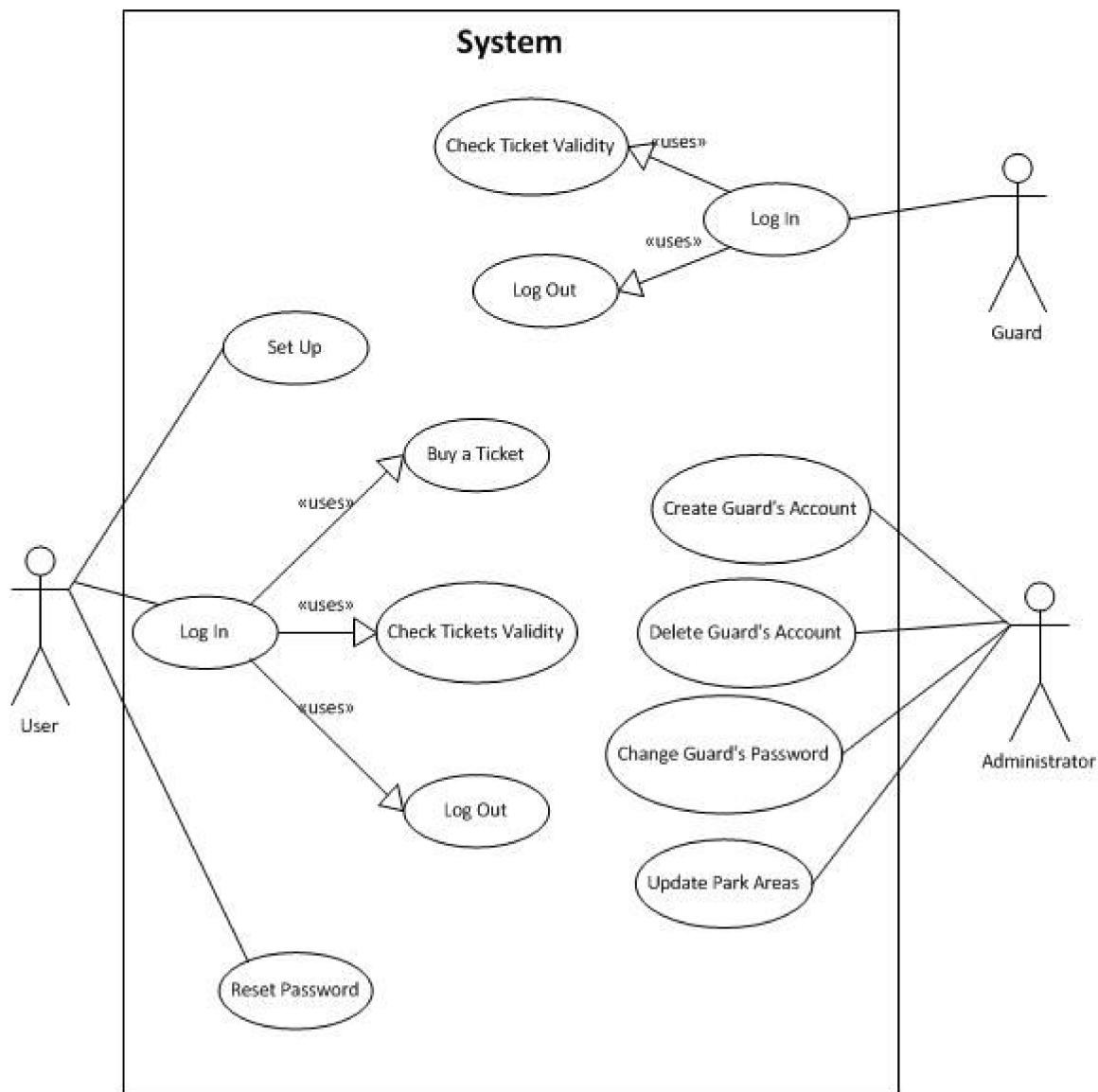
[7. Szyfrowanie połączenia](#)

[8. Specyfikacja](#)

1. Wprowadzenie

Przedmiotem naszego rozważania jest stworzenie dokumentu technicznego dla aplikacji na urządzenia mobile umożliwiających zapłatę oraz kontrolę opłaty za miejsce parkingowe w Krakowie.

2. Diagram Use Case



2.1 Opis diagramu

2.1.1. User

- SetUp - użytkownik zakłada konto
 - 1) Użytkownik w panelu wstępny wybiera opcję stworzenia nowego konta.
 - 2) Użytkownik wpisuje dane potrzebne do założenia nowego konta (login, hasło, mail)
 - 3) Po zakończeniu powyższej procedury użytkownik kończy zakładanie konta przyciskiem wysyłając zapytanie do web service
 - 4) Użytkownik otrzymuje informację o powodzeniu lub jego braku przy zakładaniu konta
- LogIn - użytkownik loguje się na swoje konto
 - 1) Użytkownik w panelu wstępny wybiera opcję logowania
 - 2) Użytkownik wpisuje dane swojego konta (login, hasło) wysyłając zapytanie do web service
 - 3) Użytkownik otrzymuje informację o powodzeniu lub jego braku przy logowaniu
- LogOut - użytkownik wylogowywuje się ze swojego konta
 - 1) Użytkownik wylogowywuje się przyciskiem wylogowywania
- Buy a Ticket - użytkownik kupuje bilet w aplikacji podając numer rejestracyjny oraz strefę parkowania
 - 1) Użytkownik wybiera opcję Kup Bilet
 - 2) Użytkownik wpisuje dane potrzebne do zakupu biletu (nr. rejestracyjny, strefa parkowania)
 - 3) Użytkownik ustawia czas trwania biletu
 - 4) Użytkownik kupuje bilet klikając przycisk wysyłając zapytanie do web service
 - 5) Użytkownik otrzymuje informację o powodzeniu lub jego braku przy zakupie biletu
- Check the validity of yours tickets - użytkownik sprawdza swoje aktualne bilety
 - 1) Użytkownik wybiera opcję Sprawdź Bilety wysyłając zapytanie do web service
 - 2) Użytkownik otrzymuje informację o aktualnych biletach bądź informację o problemie z połączeniem
- Reset Password - użytkownik resetuje hasło
 - 1) Użytkownik wybiera opcję resetuj hasło
 - 2) Na mail użytkownika zostaje wysłany link resetujący hasło

2.1.2. City Guard

- LogIn - strażnik loguje się na swoje konto
 - 1) Strażnik wpisuje swoje dane logowania (hasło, login) i wciska przycisk zaloguj wysyłając zapytanie do web service
 - 2) Strażnik otrzymuje informację o sukcesie lub jego braku przy logowaniu
- LogOut - strażnik wylogowywuje się ze swojego konta

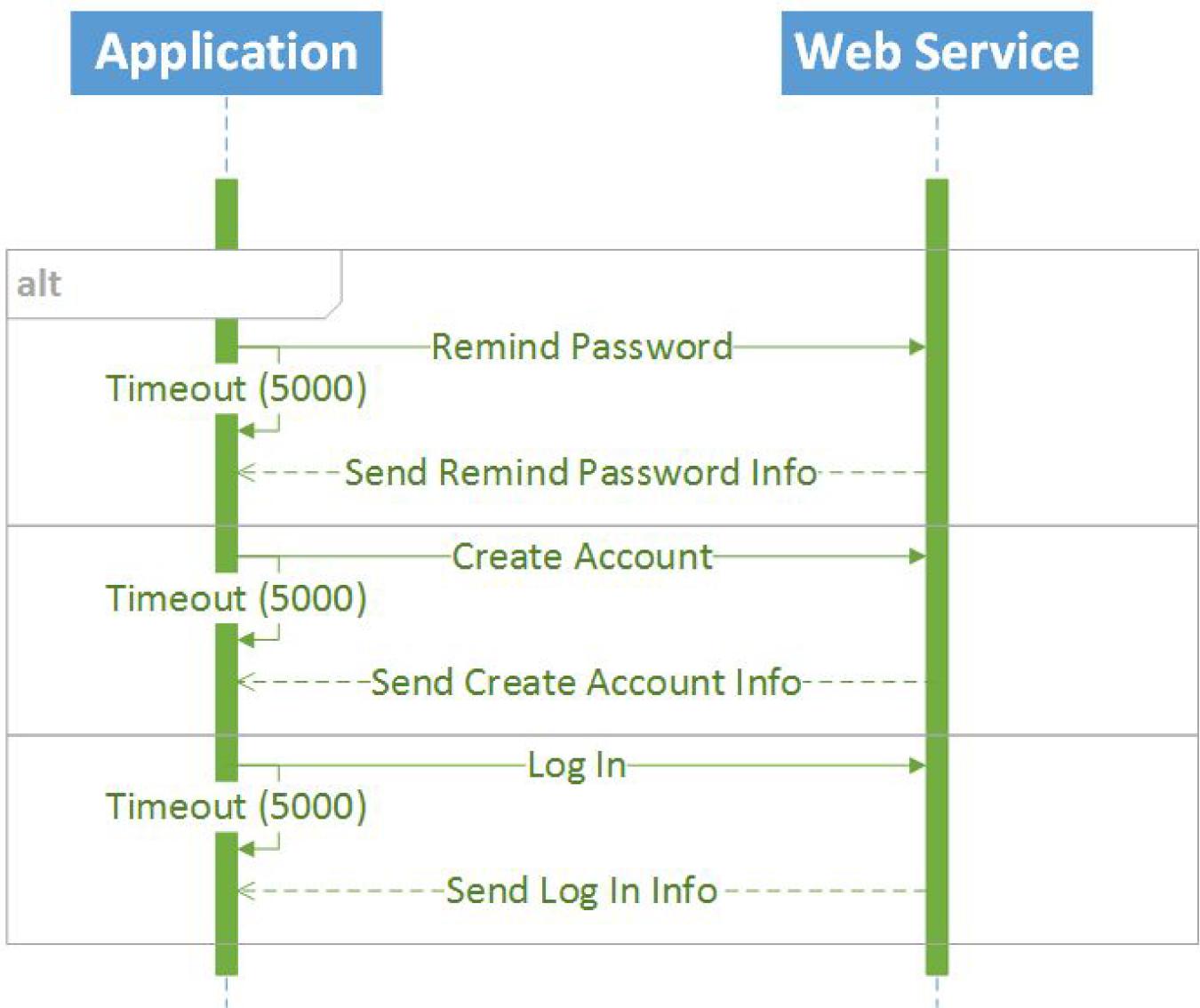
- 1) Strażnik wylogowywuje się używając przycisku wylogowywania
- Check the validity of ticket- strażnik sprawdza ważność biletu podając numer rejestracyjny samochodu
 - 1) Strażnik wpisuje numer rejestracyjny samochodu wciska przycisk sprawdź wysyłając zapytanie do web service
 - 2) Strażnik otrzymuje informację o bilecie dla danego samochodu bądź informację o problemie z połączeniem z web servicem

2.1.3. Administrator

- Set Up an account for City Guard - admin zakłada konto dla strażnika
 - 1) Admin wybiera opcję załóż konto
 - 2) Admin wpisuje dane potrzebne do założenia konta
 - 3) Admin wciska przycisk zasłóż konto wysyłając zapytanie do web service
 - 4) Admin otrzymuje informację o założonym koncie lub informację o nieudanej próbie założenia konta
- Removal the Guard's account - admin usuwa konto strażnika
 - 1) Admin wybiera opcję usuń konto
 - 2) Admin wpisuje dane usuwanego konta (//to do)
 - 3) Admin wciska przycisk usuń konto wysyłając zapytanie do web service
 - 4) Admin otrzymuje informację o usuniętym koncie lub informację o nieudanej próbie usunięcia konta
- Reset Guard's password - admin resetuje hasło strażnika
 - 1) Admin wybiera opcję reset hasła
 - 2) Admin wpisuje dane konta (login)
 - 3) Admin wciska przycisk reset hasła wysyłając zapytanie do web service
 - 4) Admin otrzymuje informację o modyfikowanym koncie lub informację o nieudanej próbie zresetowania hasła
- Update Park Area's - admin aktualizuje strefy parkowania
 - 1) Admin wybiera opcję Aktualizacji stref parkowania
 - 2) Admin wpisuje dane potrzebne do aktualizacji
 - 3) Admin wciska przycisk aktualizuj wysyłając zapytanie do web service
 - 4) Admin otrzymuje informację o zaktualizowanych strefach parkowania lub informację o nieudanej próbie aktualizacji

3. Diagramy Sekwencji

3.1 User przed zalogowaniem

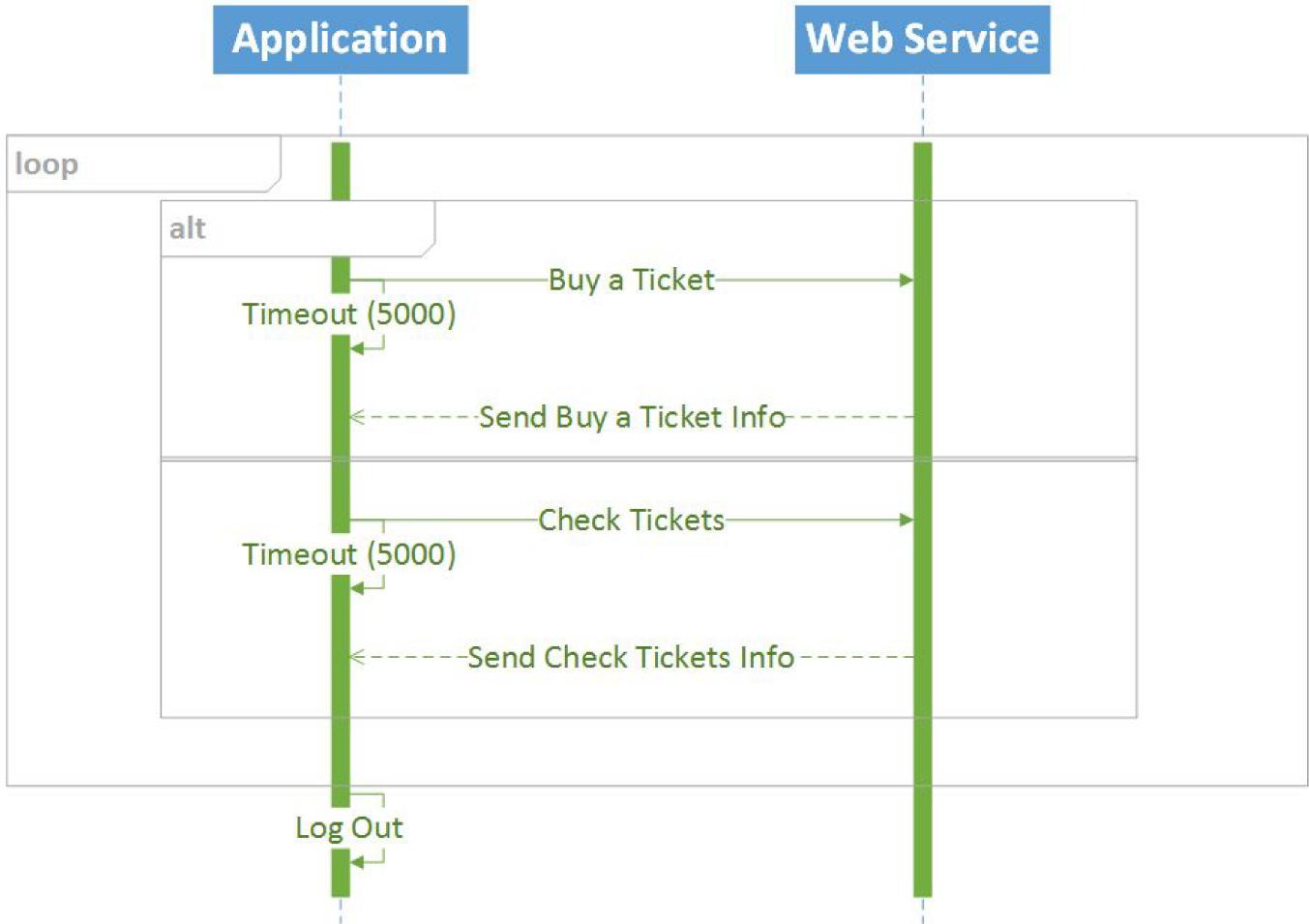


Użytkownik wchodząc do aplikacji ma do wyboru trzy opcje:

- założenie konta
- wysłanie prośby o przypomnienie hasła
- zalogowanie się do systemu

Jeżeli dana operacja się nie powiedzie to wysyłany jest komunikat zwrotny. Ponadto każda operacja jest ograniczona pięciosekundowym timeoutem m. in. na wypadek utraty połączenia z siecią.

3.2 Użytkownik po zalogowaniu



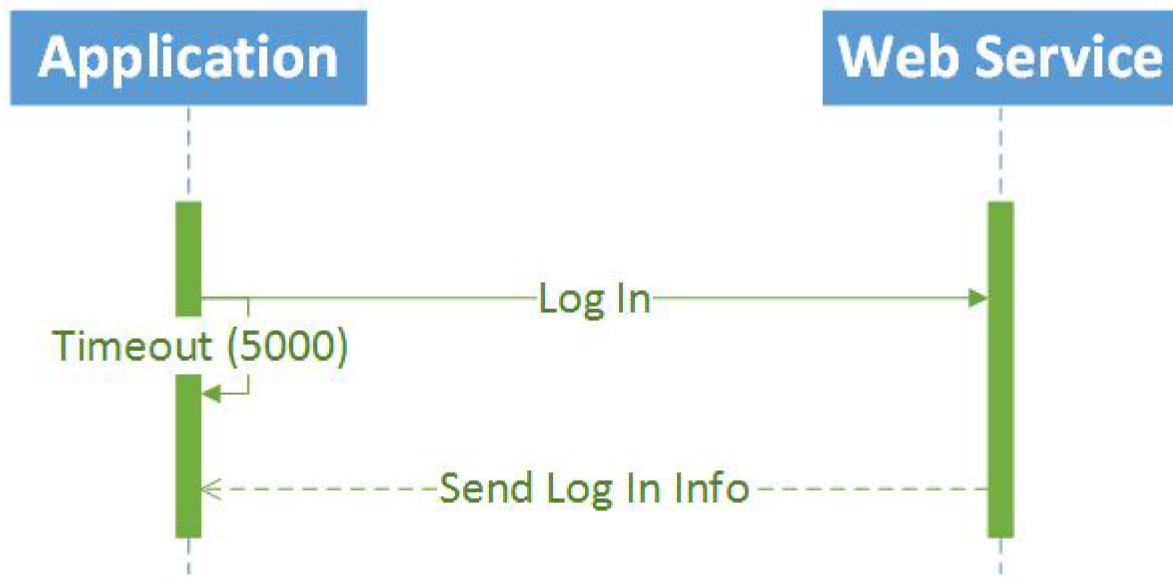
Po poprawnym zalogowaniu (oraz ewentualnym uprzednim założeniu konta) użytkownik ma w pętli do wyboru dwie opcje:

- zakup biletu
- sprawdzenie ważności biletów

Jeżeli dana operacja się nie powiedzie to wysyłany jest komunikat zwrotny. Ponadto każda operacja jest ograniczona pięciosekundowym timeoutem m. in. na wypadek utraty połączenia z siecią.

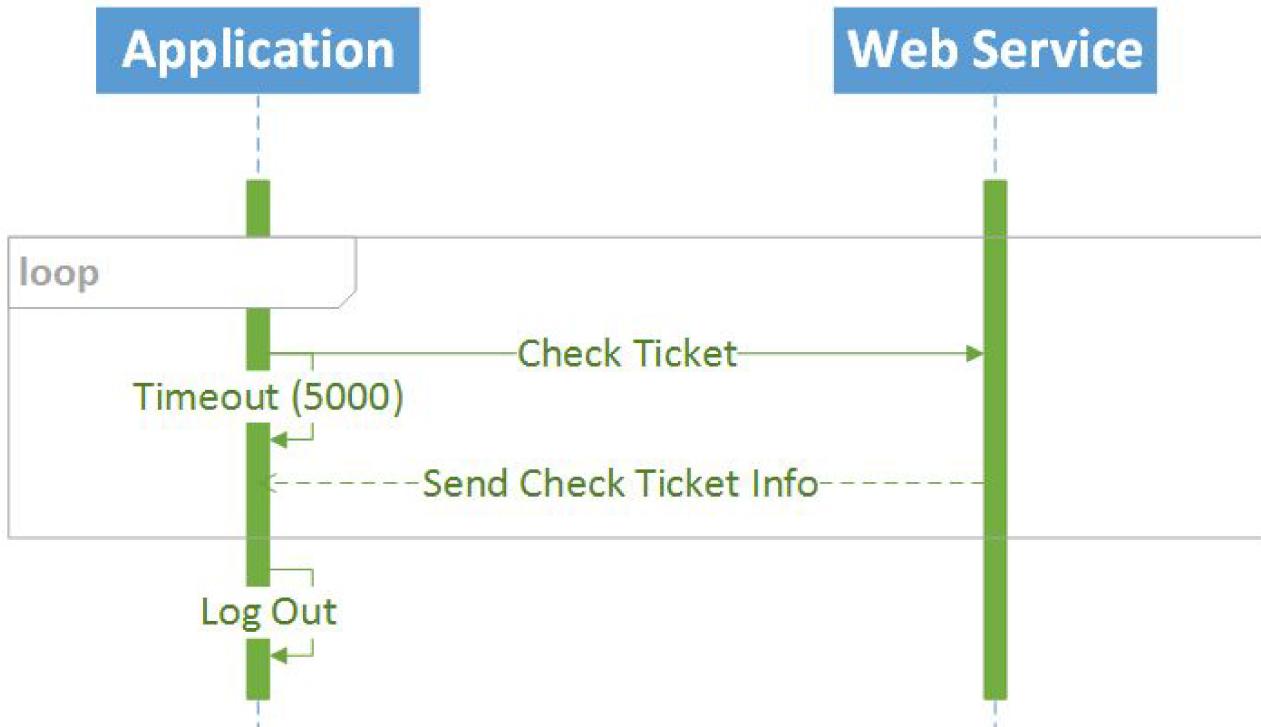
Pętla zostaje zakończona w momencie wylogowania się użytkownika.

3.3 Strażnik przed zalogowaniem



Strażnik wchodzący do aplikacji ma do wyboru tylko jedną opcję, czyli zalogowanie się do systemu. W przypadku niepowodzenia otrzyma on stosowny komunikat zwrotny. Ponadto operacja logowania jest ograniczona pięciosekundowym timeoutem m. in. na wypadek utraty połączenia z siecią.

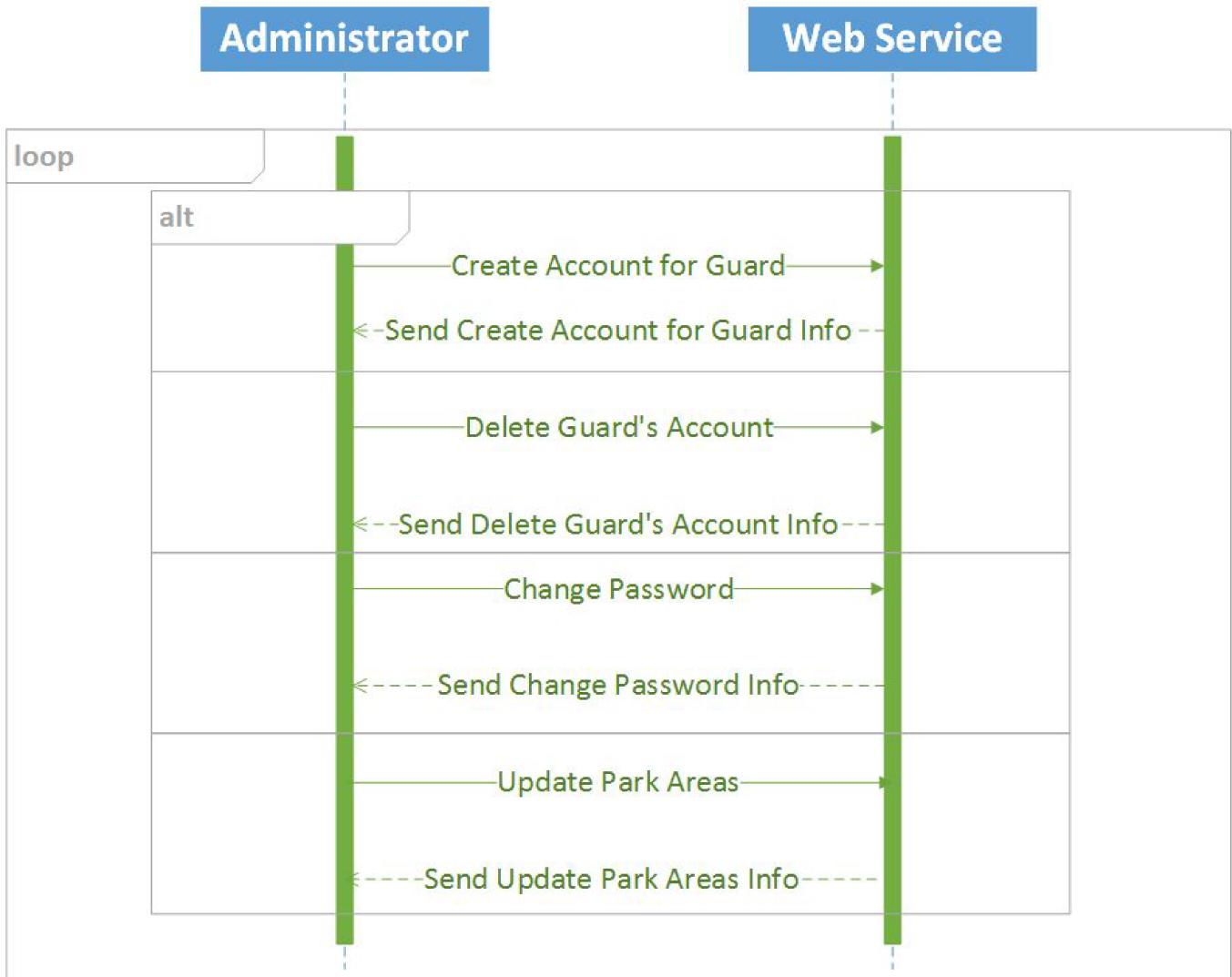
3.4 Strażnik po zalogowaniu



Po poprawnym zalogowaniu strażnik może w pętli sprawdzać ważność biletu. Jeżeli ta operacja się nie powiedzie to wysyłany jest komunikat zwrotny. Ponadto jest ona ograniczona pięciosekundowym timeoutem m. in. na wypadek utraty połączenia z siecią.

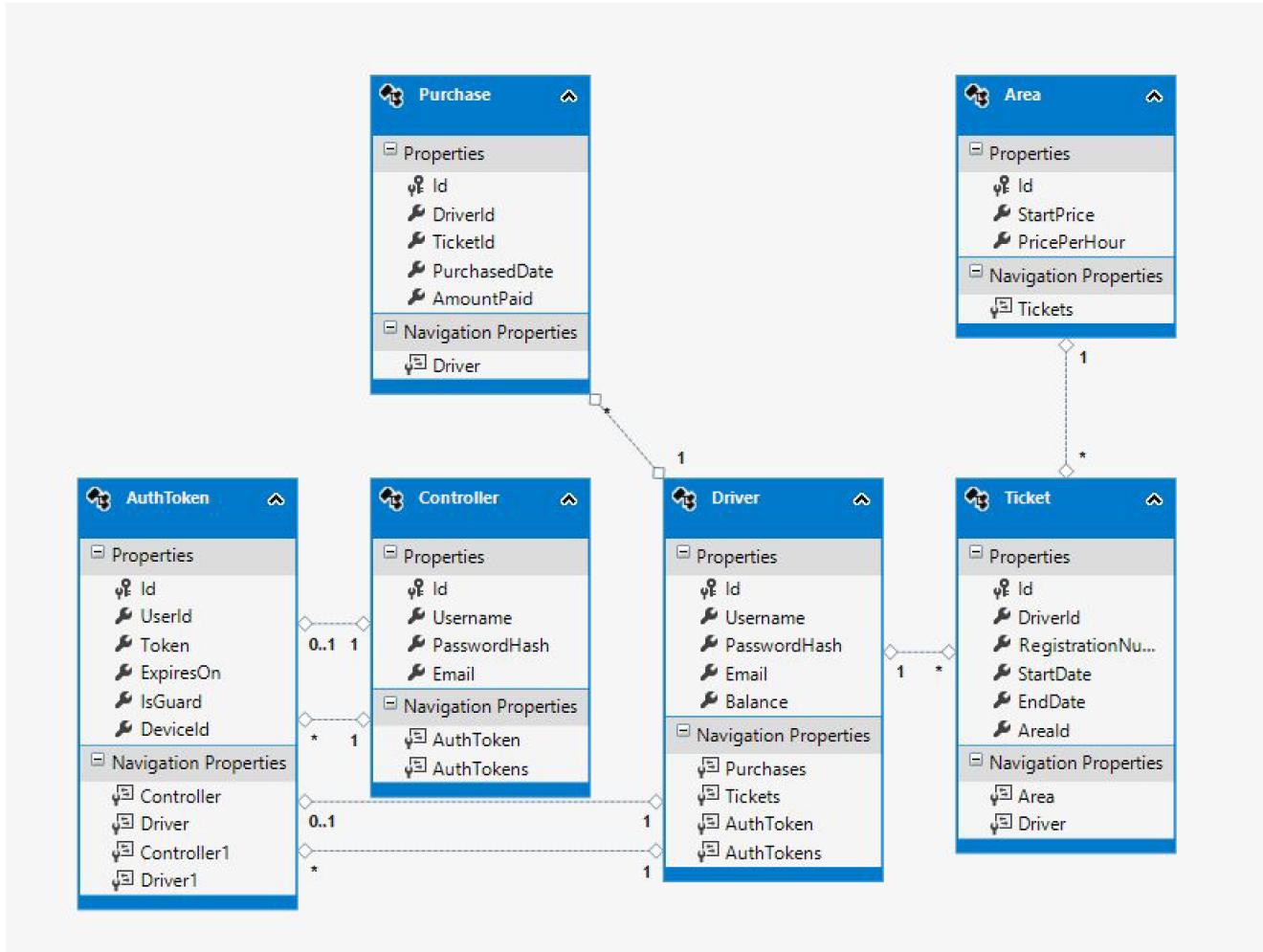
Pętla zostaje zakończona w momencie wylogowania się strażnika.

3.5 Administrator



Administrator może wykonać jedną z następujących czynności: stworzyć konto dla strażnika, usunąć konto strażnika, zmienić hasło konta strażnika lub uaktualnić strefy parkowania.

4. Diagram bazy danych



4.1 Ogólny opis tabel i relacji

- ❖ Tabela Controller zawiera informacje o kontach strażników miejskich
- ❖ Tabela Area zawiera informacje o strefach parkowania- cenę początkową oraz cenę za godzinę parkowania.
- ❖ Tabela Drivers zawiera informację na temat kont kierowców.
- ❖ Tabela Ticket zawiera informację o bilecie- informację o kierowcy, początku i końcu biletu oraz o strefie parkowania.
- ❖ Tabela Purchase zawiera informację o zakupionych biletach - informację o kierowcy, cenie zakupionego biletu i jego wartość.
- ❖ Tabela AuthToken zawiera informację o Tokenach- obiektach reprezentujących zamówiony bilet wykorzystywanych przez strażników.
- ❖ Tabele Area oraz Ticket są połączone relacją 1:N
- ❖ Tabele Drivers oraz Ticket są połączone relacją 1:N
- ❖ Tabele Drivers oraz Purchase są połączone relacją 1:N
- ❖ Tabela Driver oraz AutoToken są połączone relacją 1:N
- ❖ Tabela Guards oraz AutoToken są połączone relacją 1:N

5. Polecenia służące do połączenia z web servicem

D - dla kierowcy

C - dla kontrolera

- **D** ~/web.ashx/register (*username*, *password*, *email* : string)
returns "OK", error message otherwise
- **D** ~/web.ashx/login (*username*, *password*: string)
returns **authToken** : string, error message otherwise
- **D** ~/web.ashx/buyTicket (*registrationNumber*, *startDate*, *endDate*, *authToken* : string, *areald*: int)
returns "OK", error message otherwise
- **D** ~/web.ashx/listTickets (*authToken* : string)
returns {**JSON:Tickets**}, error message otherwise
- **D** ~/web.ashx/getBalance (*authToken* : string)
returns **balance** : decimal, error message otherwise
- **C** ~/web.ashx/loginController (*username*, *password* : string)
returns **authToken** : string, error message otherwise
- **C** ~/web.ashx/checkTicket (*registrationNumber*:string, *areald*: int, *authToken* : string)
returns "OK", error message otherwise

{**JSON:Tickets**} response:

```
[  
  {  
    registrationNumber : "KR45604",  
    startDate : "2015-11-12T23:23:11",  
    endDate : "2015-11-12T23:23:11",  
    areald : 3  
  }, ...  
]
```

Format daty: YYYY-MM-DDTHH:mm:ss

Data odpowiadająca strefie środkowo-europejskiej (CEST)

6. Testowanie połączenia z bazą danych

Domena testowa: <https://thebilit.usetitan.com> (zamiast ~)

Testowy user (ma dużo kasy w Balance wiec można kupować bilety). Wiadomo, można rejestrować nowych:

Username: **tester1**

Hasło: **tester1**

Email: icarotka@gmail.com

Testowy kontroler:

Username: **kontroler1**

Hasło: **kontroler1**

Dostępne Areald: **1 i 2**

7. Szyfrowanie połączenia

Połączenie jest szyfrowane przy pomocy protokołu sieciowego SSL, który realizuje szyfrowanie, uwierzytelnianie serwera i zapewnia integralność oraz poufność przesyłanych danych. W momencie nawiązania połączenia z webservicem następuje ustalenie algorytmów oraz kluczy szyfrujących stosowanych następnie przy przekazywaniu danych między przeglądarką a serwerem. W pierwszej fazie nawiązywania połączenia SSL server i przeglądarka wymieniają tzw. certyfikaty, które są odpowiednikiem dokumentu tożsamości dla serwera WWW oraz dla użytkownika. Użytym certyfikatem w systemie jest ComodoSSL.

8. Uwagi dotyczące aplikacji

- wszystkie operacje zachodzą po stronie webservice, aplikacja zarówno kierowcy jak i strażnika jedynie wysyłają odpowiednie żądania przy pomocy GET i otrzymują odpowiedź od webservice, która powiadamia o sukcesie żądania lub o porażce wraz z typem błędu,
- aplikacja nie posiada po swojej stronie żadnego cache, w którym przechowywałaby informacje o aktualnych biletach,
- wszystkie informacje o biletach oraz użytkownikach są trzymane w bazie danych po stronie serwera.