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

Krzysztof Łuczak
nr indeksu: 122008
e mail:
krzysztof.t.luczak@student.put.poznan.pl

Dawid Wiktorski
nr indeksu: 122056
e mail:
dawid.wiktorski@student.put.poznan.pl

12 czerwca 2017

Spis treści

Rozdział 1

Wstęp

System ten będzie służyć do uzyskiwania podpisów cyfrowych ze stemplem czasowym, który będzie wiarygodny poprzez zastosowanie urzędu certyfikacyjnego w postaci serwera połączonego z bazą danych. Użytkownik przy pomocy aplikacji lub strony internetowej będzie przesyłał dokument wraz z funkcją skrótu, następnie serwer będzie swoim własnym podpisem dawał stempel czasowy, który będzie umieszczał w bazie danych i odsyłał użytkownikowi tak zwany plik magnetyczny [1] wraz znacznikiem czasowy. Posiadając plik magnetyczny Dzięki temu dokumenty będą miały wiarygodny znacznik daty podpisu pozwalający określić dokładną godzinę zatwierdzenia dokumentu w systemie autentykacji.

W skład systemu będzie wchodzić:

- aplikacja serwerowa z bazą danych,
- aplikacja webowa,
- aplikacja mobilna,
- aplikacja desktopowa.

[1] Plik służacy do pobrania dokumentu z repozytorium serwera, utworzony na podstawie skrótu dokumentu oryginalnego oraz znacznika czasowego

Rozdział 2

Organizacja pracy

2.1 Podział zadań

Realizacja projektu odbywać się będzie w modułach tworzonych współbieżnie:

- dokumentowania projektu,
- tworzenia aplikacji serwerowej,
- tworzenia aplikacji webowej,
- tworzenia aplikacji mobilnej,
- tworzenia aplikacji desktopowej,
- mechanizm uwierzytelniania.

Wyszczególniony podział modułów pomiędzy członków zespołu przedstawiony został w Tabeli 2.1.

Tabela 2.1: Podział zadań projektowych

Osoba	Rola
Maciej Marciniak	Kierownik, programista: dokumentacja, organizacja zespołu
Dawid Wiktorski	Programista: aplikacja mobilna na system Android, aplikacja serwerowa
Krzysztof Łuczak	Programista: aplikacja webowa, aplikacja serwerowa, mechanizm uwierzytelniania
Damian Filipowicz	Programista, aplikacja desktopowa, aplikacja serwerowa

2.2 Harmonogram pracy

Harmonogram pracy zespołu przedstawiony został na wykresie Gantta, który znajduje się na Rys.

Zadania:	Podzadania:	Osoba:	Termin rozpoczęcia	Termin zakończenia	T1 - 14.03.20017	T2 - 21.03.2017	T3 - 28.03.2017	T4 - 04.04.2017	T5 - 11.04.2017	T6 - 25.04.2017	T7 - 09.05.2017	T8 - 16.05.2017	T9 - 23.05.2017	T10 - 30.05.2017	T11 - 06.06.2017	T12 - 13.06.2017
Organizacja pracy	Tworzenie dokumentacji	Maciej Marciniak	T1	T12												
	Utworzenie repozytorium	Wszyscy	T2	T2												
	Utworzenie diagramów UML	Maciej Marciniak	T3	T7												
	Przydział prac	Maciej Marciniak	T1	T1												
	Zapoznanie się z tematem projektu	Wszyscy	T1	T1												
Zapoznanie z technologiami	Java Android	Dawid Wiktorski	T2	T2												
	Django / Flask	Krzysztof Łuczak	T2	T2												
	Bootstrap / HTML / CSS	Krzysztof Łuczak	T2	T2												
	WPF	Damian Filipowicz	T2	T2												
Apliakcja desktopowa	Wybór środowiska pracy	Damian Filipowicz	T3	T4												
	Opracowanie koncepcji (wygląd, funkcjonowanie)	Wszyscy	T3	T5												
	Wykonanie wynkcji skrótu	Damian Filipowicz	T6	T7												
	Funkcja do przesłania dokumenu	Damian Filipowicz	T6	T10												
	Połączenie z serwerem	Damian Filipowicz	T10	T11												
	Bezpieczeństwo	Damian Filipowicz	T11	T11												
Aplikacja serwerowa z bazą danych	Wybór środowiska pracy	Wszyscy	T3	T4												
	Opracowanie koncepcji (funkcjonowanie)	Wszyscy	T3	T5												
	Utworzenie bazy danych	Wszyscy	T5	T6												
	Implementacja algorytmów	Wszyscy	T5	T6												
	Utworzenie metod połączenia dla pozostałych modułów	Wszyscy	T6	T9												
	Bezpieczeństwo	Wszyscy	T9	T10												
Aplikacja mobilna	Wybór środowiska pracy	Dawid Wiktorski	T3	T4												
	Opracowanie koncepcji (wygląd, funkcjonowanie)	Wszyscy	T3	T5												
	Utworzenie layouťu	Dawid Wiktorski	T6	T6												
	Wykonanie funkcji skrótu	Dawid Wiktorski	T6	T7												
	Połączeine z serwerem	Dawid Wiktorski	T7	T10												
	Odebranie funkcji skrótu z stemplem czasowym z serwera	Dawid Wiktorski	T10	T10												
	Funkcje zarządzania kontem użytkownika oraz plikami	Dawid Wiktorski	T10	T11												
	Bezpieczeństwo	Dawid Wiktorski	T10	T11												
Apikacja webowa	Wybór środowiska pracy	Krzysztof Łuczak	T3	T4												
	Opracowanie koncepcji (wygląd, funkcjonowanie)	Wszyscy	T3	T5												
	Fronend	Krzysztof Łuczak	T5	T6												
	Backend	Krzysztof Łuczak	T6	T9												
	Osadzenie modułów w sieci globalnej	Krzysztof Łuczak														
	Bezpieczeństwo	Krzysztof Łuczak	T10	T11												
Dodatkowe zadania	Osadzenie modułów w sieci globalnej	Wszyscy	T11	T11												
	Testowanie systemu	Wszyscy	T11	T11												
	Mechanizm uwierzytelniania	Krzysztof Łuczak	T11	T11												

Rys. 2.1: Diagram weryfikacja stempli czasowych

Rozdział 3

Idea stempli czasowych

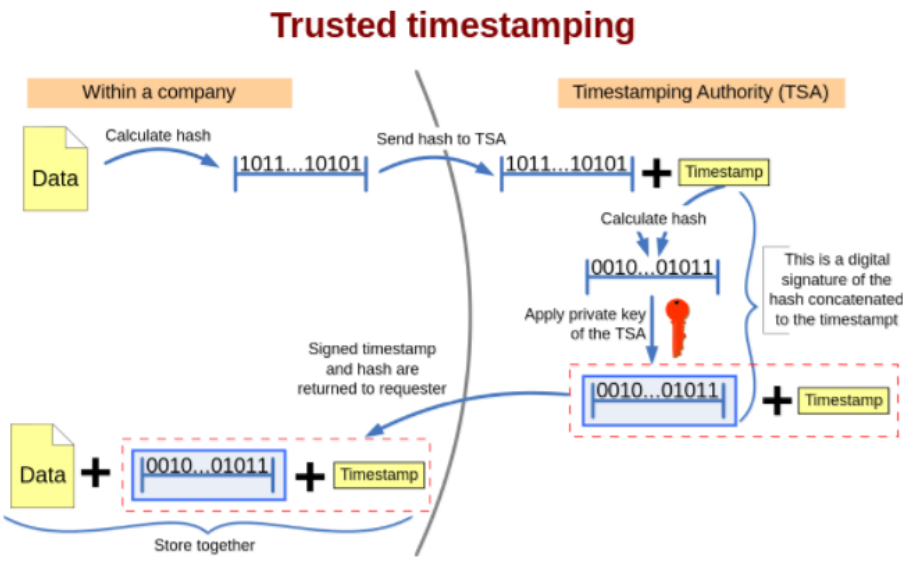
Na dokument podpisany stemplem czasowym składają się trzy pliki:

- podpisywany dokument,
- stempel czasowy,
- wygenerowany certyfikat uwierzytelniający stempel czasowy.

Tworzenie dokumentu podpisanego stemplem czasowym

Tworzenie certyfikatu odbywa się po stronie zaufanego urzędu poprzez dodanie do funkcji skrótu podpisywanego dokumentu, stempla czasowego, a następnie z utworzonej paczki ponownie tworzy się funkcję skrótu. Ostatecznie skrót obu plików szyfruje się kluczem prywatnym i przesyła się z powrotem do użytkownika.

Schemat przebiegu tworzenia dokumentu przedstawiony jest na Rys. 3.1.

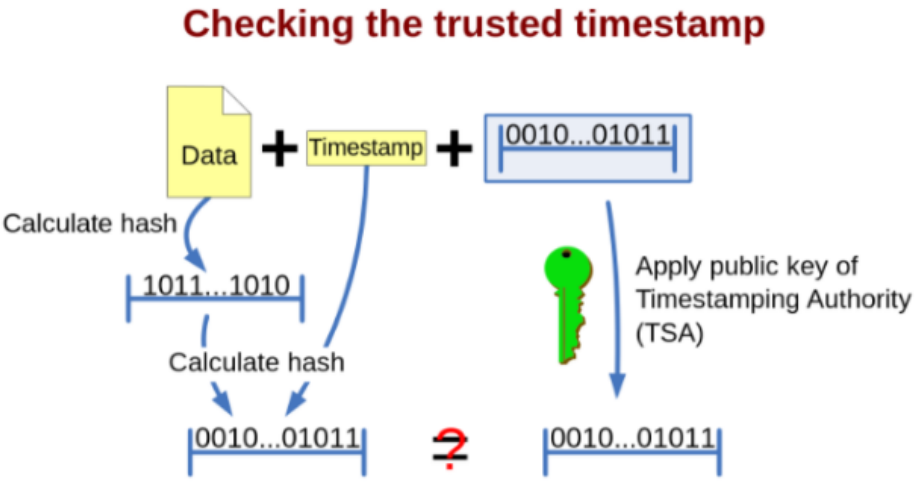


Rys. 3.1: Diagram tworzenia stempli czasowych

Weryfikacja dokumentu podpisanego stemplem czasowym

Weryfikacja certyfikatu odbywa się po stronie użytkownika poprzez wykonanie funkcji skrótu dokumentu, następnie dodanie do niej dołączonego stempla czasowego i ponowne wykonanie funkcji skrótu. Certyfikat należy odszyfrować kluczem publicznym użytkownika, po czym porównać utworzone ciągi znaków. Jeżeli otrzymane pliki są identyczne, to dokument został podpisany o podanej godzinie, która umieszczona jest w dołączonym pliku stempla czasowego.

Schemat przebiegu weryfikacji dokumentu przedstawiony jest na Rys. 3.2.



Rys. 3.2: Diagram weryfikacja stempli czasowych

Rozdział 4

Aktorzy systemu

W systemie wyróżniamy następujących aktorów:

- **Serwer** — główna inteligencja systemu operująca na bazie danych, pełni funkcję urzędu certyfikującego,
- **Aplikacja webowa** — aplikacja internetowa działająca wraz z serwerem udostępniająca interfejs graficzny dla użytkownika,
- **Aplikacja mobilna** — aplikacja kliencka zainstalowana na urządzeniu mobilnym z systemem android,
- **Aplikacja desktopowa** — aplikacja kliencka znajdująca się na komputerze PC z systemem operacyjnym Windows,
- **Użytkownik** — osoba fizyczna operująca aplikacją kliencką, chcąca podpisać lub odczytać dokument,
- **Gość** — osoba fizyczna niezalogowana do systemu.

Rozdział 5

Opis składowych systemu

5.1 Aplikacja serwerowa

Aplikacja serwerowa spełnia główną rolę w systemie. Jest urzędem podpisującym dokumenty stemplem czasowym oraz repozytorium dla plików. Wymagania funkcjonalne stawiane aplikacji serwerowej przedstawiono w Tabeli 5.1.

Tabela 5.1: Tabela wymagań funkcjonalnych aplikacji serwerowej

Funkcja	Opis	Aktorzy
Odbieranie plików	Umożliwienie odbierania plików z dokumentami wysyłanych z poziomu aplikacji klienta	Aplikacja serwerowa, Aplikacja kliencka
Podpisanie pliku stemplem czasowym	Utworzenie sygnatury potwierdzającej podpisanie pliku o danej godzinie	Aplikacja serwerowa
Dodanie pliku do repozytorium	Umieszczenie plików wraz z stemplem czasowym na dysk serwera	Aplikacja serwerowa
Udostępnianie pliku z repozytorium	Pobranie konkretnego pliku z repozytorium oraz udostępnienie go użytkownikowi	Aplikacja serwerowa, Aplikacja kliencka
Rejestracja użytkowników	Weryfikacja poprawności danych oraz dodawanie nowych użytkowników do systemu	Aplikacja serwerowa, Aplikacja kliencka
Logowanie użytkowników	Uwierzytelnianie użytkownika, tworzenie tokenów sesji	Aplikacja serwerowa, Aplikacja kliencka

5.2 Aplikacja kliencka — webowa, mobilna i desktopowa

Aplikacja kliencka znajduje się po stronie klienta w postaci interfejsu graficznego. Wymagania funkcjonalne stawiane aplikacji webowej przedstawiono w Tabeli 5.2.

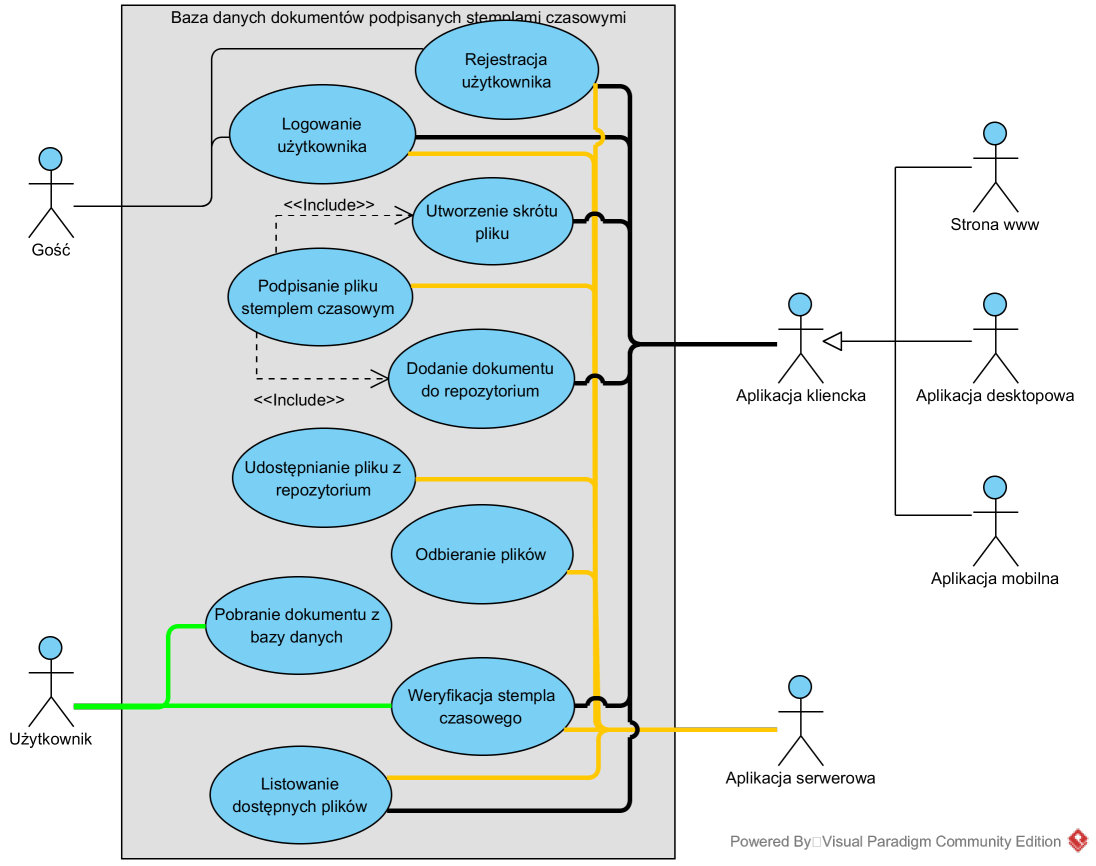
Tabela 5.2: Tabela wymagań funkcjonalnych aplikacji klienckiej

Funkcja	Opis	Aktorzy
Odbieranie plików z aplikacji serwerowej	Pobierania plików z dokumentami znajdujących się w repozytorium serwera	Aplikacja serwerowa, Aplikacja kliencka
Wysyłanie plików	Dodanie plików do repozytorium dokumentów	Aplikacja serwerowa, aplikacja kliencka
Weryfikacja stempla czasowego	Sprawdzenie wiarygodności stempla czasowego	Aplikacja kliencka, Aplikacja serwerowa
Wykonanie skrótu dokumentu	Użycie funkcji skrótu na dokumencie	Aplikacja kliencka
Logowanie użytkownika	Możliwość uzyskania uprawnień użytkownika zalogowanego	Aplikacja kliencka, Aplikacja serwerowa
Rejestracja użytkownika	Możliwość utworzenia konta użytkownika w systemie	Aplikacja kliencka, Aplikacja serwerowa
Listowanie dostępnych plików	Wyświetlenie dostępnych w repozytorium plików	Aplikacja kliencka, Aplikacja serwerowa

Rozdział 6

Diagram przypadków użycia

Diagram przypadków użycia (funkcjonalności) systemu wraz z opowiadającymi aktorami przedstawiono na Rys. 6.1.



Rys. 6.1: Diagram przypadków użycia

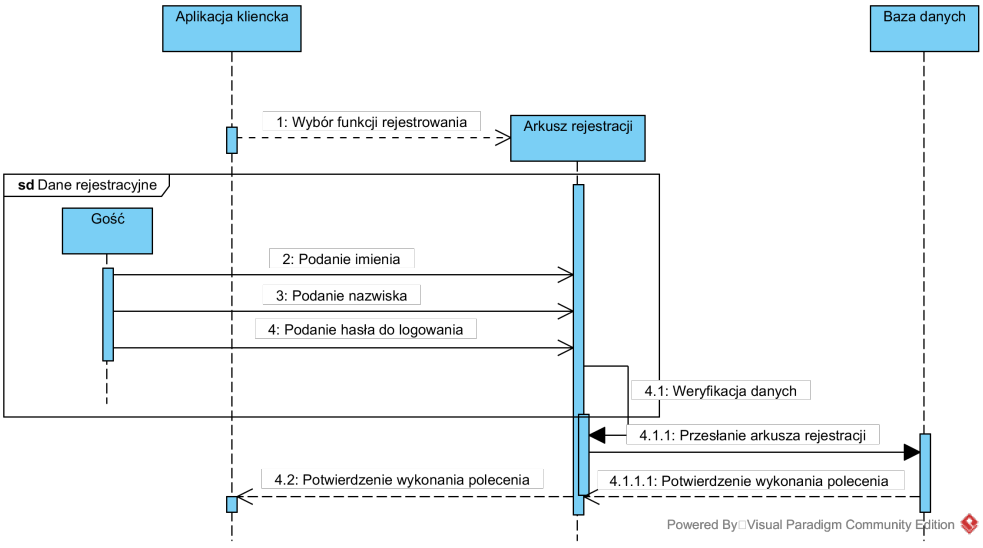
Rozdział 7

Diagramy sekwencji

Diagramy przedstawione w tym rozdziale mają na celu przybliżenie ogólnego działania systemu. Schematy nie są odzwierciedleniem poszczególnych przypadków użycia, mogą zawierać szczątkowe odniesienia do wielu z nich.

Rejestracja użytkownika

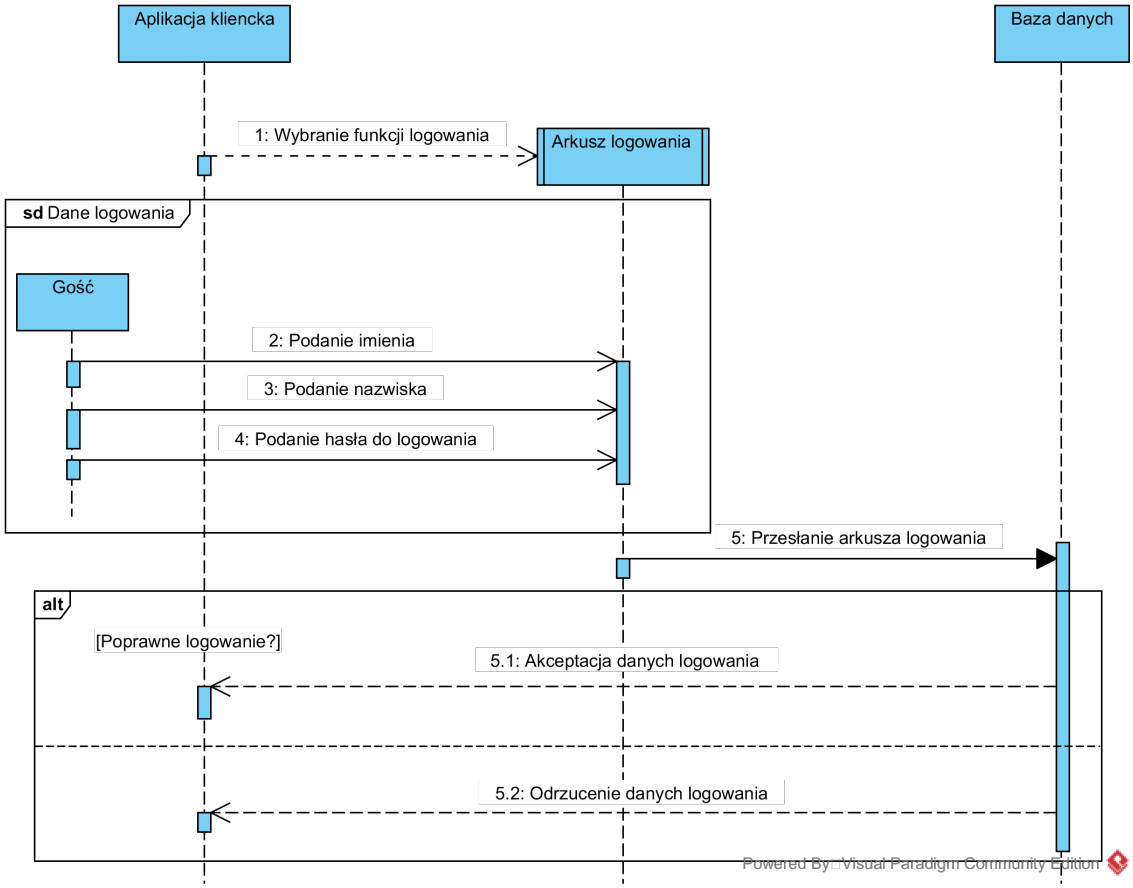
Diagram znajdujący się na Rys. 7.1 przedstawia sekwencje akcji wykonywanych podczas rejestracji użytkownika.



Rys. 7.1: Diagram sekwencji rejestracji użytkownika

Logowanie użytkownika

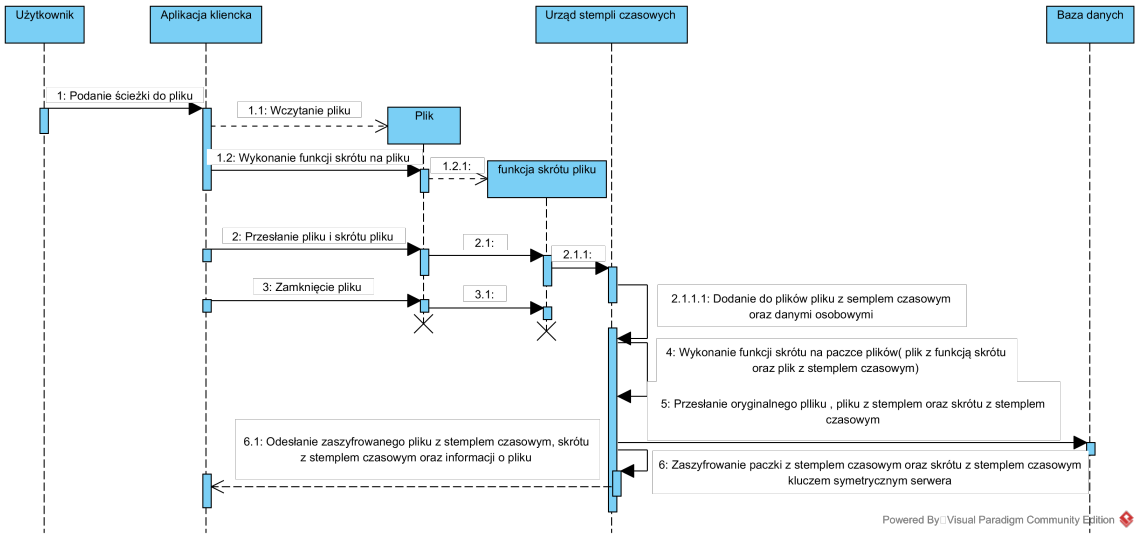
Diagram znajdujący się na Rys. 7.2 przedstawia sekwencje akcji wykonywanych podczas logowania użytkownika.



Rys. 7.2: Diagram sekwencji logowanie użytkownika

Podpis pliku stemplem czasowym

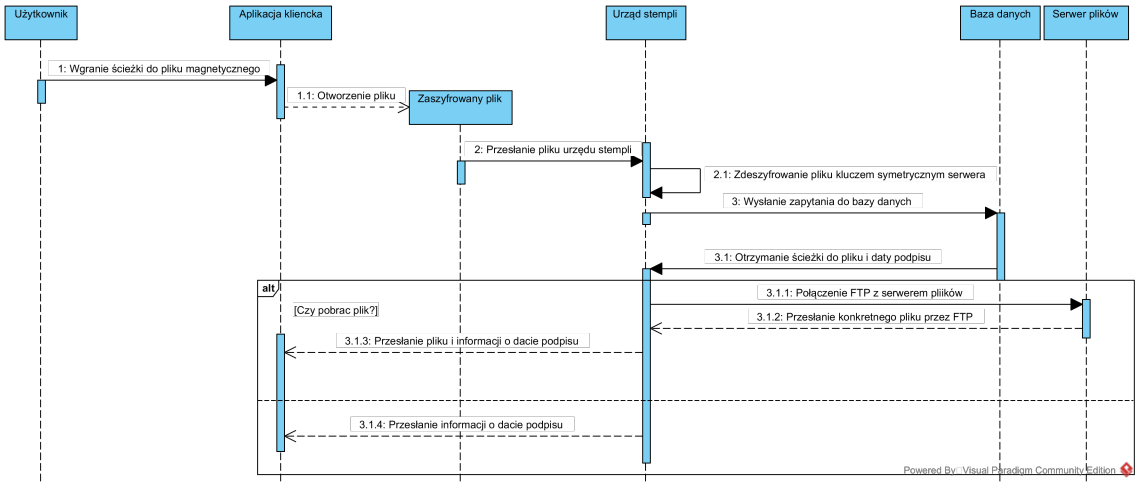
Diagram znajdujący się na Rys. 7.3 przedstawia sekwencje akcji wykonywanych przy podpisywaniu pliku stemplem czasowym.



Rys. 7.3: Diagram sekwencji podpisu pliku stemplem czasowym

Weryfikacja stempla czasowego

Diagram znajdujący się na Rys. 7.4 przedstawia sekwencje akcji wykonywanych podczas weryfikacji wiarygodności stempla czasowego.



Rys. 7.4: Diagram sekwencji weryfikacji stempla czasowego

Rozdział 8

Projekt bazy danych

Baza danych składa się z 3 użytkowych tabel. Projekt zawiera więcej pozycji, lecz są one definiowane domyślnie przez aplikację serwerową zaimplementowaną w Django i nie są wykorzystywane. Tabele używane w projekcie:

- **Auth_user** — przechowuje dane użytkowników systemu,
- **Main_app_documents** — zawiera dokumenty znajdujące się w repozytorium serwera,
- **Main_app_tokens** — przetrzymuje tokeny sesji użytkowników.

Tabela Auth_user składa się z:

- **id** — unikalny identyfikator użytkownika,
- **password** — hasło użytkownika przechowywane w postaci skrótu,
- **last_login** — data ostatniego zalogowania użytkownika,
- **is_superuser** — określa czy użytkownik posiada uprawnienia administratora,
- **first_name** — imię użytkownika,
- **last_name** — nazwisko użytkownika,
- **email** — adres email potrzebny do rejestracji,
- **is_staff** — czy ma dostęp do strony administratora,
- **is_active** — czy konto użytkownika jest aktywne,
- **data_joined** — data utworzenia konta,
- **username** — login użytkownika.

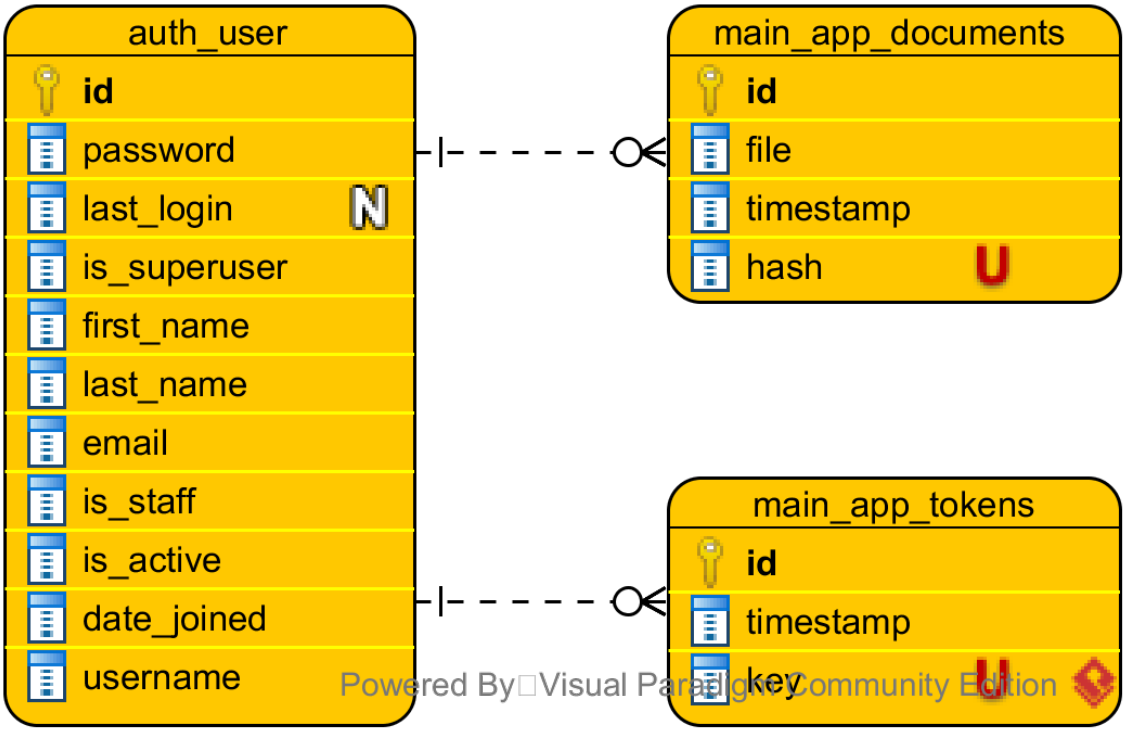
Tabela Main_app_documents zawiera:

- **id** — unikalny identyfikator dokumentu,
- **owner_id** — klucz obcy, identyfikator użytkownika, który jest właścicielem dokumentu,
- **file** — ścieżka dostępu do pliku znajdującego się w pamięci dysku serwera,
- **timestamp** — znacznik czasowy podpisu dokumentu,
- **hash** — skrót zawartości dokumentu.

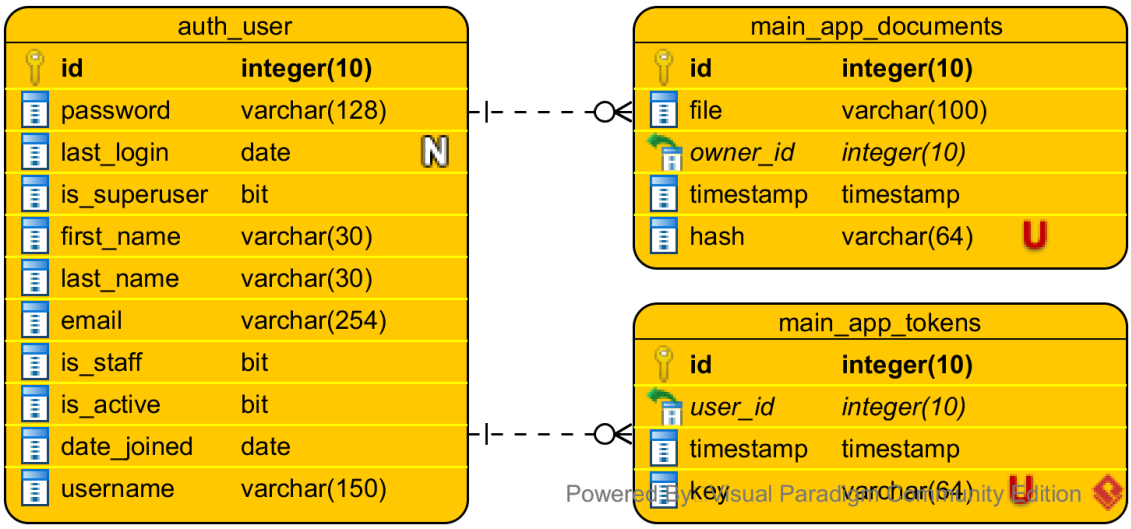
Tabela Main_app_tokens przetrzymuje takie dane jak:

- **id** — unikalny identyfikator tokena sesji,
- **user_id** — klucz obcy, identyfikator użytkownika,
- **timestamp** — czas ważności tokena,
- **key** — unikalna wartość tokena.

Diagramy bazy danych odpowiednio encji i relacji przedstawione zostały na Rys. 8.1 i Rys. 8.2.



Rys. 8.1: Diagram encji bazy danych



Rys. 8.2: Diagram relacji bazy danych

Rozdział 9

Diagramy klas

9.1 Aplikacja serwerowa

9.2 Aplikacja webowa

9.3 Aplikacja desktopowa

9.4 Aplikacja mobilna

Rozdział 10

Implementacja

Kod źródłowy systemu wraz z postępami pracy znajduje się w Repozytorium GitHub.

Rozdział 11

Widok graficzny systemu, obsługa interfejsu

11.1 Aplikacja webowa

11.2 Aplikacja desktopowa

11.3 Aplikacja mobilna

Rozdział 12

Perspektywy rozwoju

System na stan obecny ogranicza użytkownika do korzystania z systemu android, jeśli chodzi o aplikację mobilną oraz wymusza korzystanie z systemu Windows w celu użycia aplikacji desktopowej. W przyszłości można rozszerzyć moduły o kompatybilność z systemami IOS i Linux.

Dodatkową funkcjonalność systemu jaką należy rozważyć w perspektywach rozwoju jest dodawanie wielu plików jednocześnie, jak również wprowadzenie możliwość dodawania dokumentów wymaganych wielu podpisów przez różnych użytkowników.

System z scentralizowaną bazą danych może być narażony na przeciążenia oraz przepełnienie pamięci. Rozwiązaniem jest zastosowanie rozproszonych bazy danych.