

Politechnika Łódzka
Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Instytut Informatyki Stosowanej

PRACA DYPLOMOWA INŻYNIERSKA

Weryfikowalność i prywatność w systemach głosowania elektronicznego

Verifiability and privacy in e-voting systems

Mateusz Domałżek

Numer albumu: 215705

Opiekun pracy:
dr hab. Szymon Grabowski, prof. PŁ

Łódź, luty, 2021

Spis treści

1 Wstęp	4
1.1 Cel i zakres pracy	4
1.2 Dostępne rozwiązania	4
1.3 Historia głosowania	5
2 Kryptografia asymetryczna	6
2.1 Funkcje jednokierunkowe	6
2.2 Arytmetyka modularna	7
2.3 Problem logarytmu dyskretnego	7
2.4 Algorytm Rivesta-Shamira-Adlemana (RSA)	10
2.4.1 Generowanie kluczy	10
2.4.2 Szyfrowanie	11
2.4.3 Deszyfrowanie	11
2.5 Algorytm ElGamala (ELG)	12
2.6 Podpis cyfrowy	14
3 Bezpieczeństwo	16
3.1 Klasyfikacja systemów	17
3.2 Zagrożenia	18
3.3 Doświadczenia ze świata	20
3.3.1 Ukraina	20
3.3.2 Estonia	21
3.3.3 Holandia	24
4 Aplikacja	26
4.1 Ogólny opis	26

4.1.1	Perspektywa produktu	27
4.1.2	Funkcje produktu	27
4.1.3	Ograniczenia projektowe	28
4.1.4	Charakterystyka użytkowników	29
4.2	Wymagania funkcjonalne	29
4.3	Implementacja	49
4.3.1	Algorytm RSA	49
4.3.2	Moduł SMS	55
4.3.3	Geolokalizator	56
4.3.4	Głosowanie	58
4.4	Testy aplikacji	60
5	Podsumowanie	68

Streszczenie

Celem pracy było stworzenie internetowego systemu do przeprowadzania referendum, wykorzystującego kryptografię asymetryczną do utajniania decyzji wyborców. Platforma została zaprogramowana z użyciem ASP.NET CORE 5.0 MVC oraz środowiska Python 3.8. Do magazynowania danych wykorzystano SQL Server 2019, a także dysk zewnętrzny.

W pracy zaprezentowano niezbędne zagadnienia z zakresu kryptografii asymetrycznej oraz popularnych algorytmów szyfrujących, przybliżono również problematykę historii demokracji i światowych doświadczeń w dziedzinie e-votingu. Przedstawiono kryteria klasyfikacji systemów do głosowania elektronicznego oraz opisano zagrożenia wynikające z wykorzystania nowoczesnych technologii. W pracy zawarto także autorską implementację algorytmu RSA. Postawione cele zostały osiągnięte. Rezultatem pracy jest aplikacja Referendum, która została odpowiednio przetestowana. Praca zawiera opis przeprowadzonych testów oraz działania platformy.

Słowa kluczowe: głosowanie internetowe, kryptografia asymetryczna, RSA, Python, C#

Abstract

The aim of the study was to create an internet system for carrying out a referendum. The system makes use of asymmetric cryptography to classify elector's vote. Web based platform is programmed using ASP.NET CORE 5.0 MVC and Python 3.8 environment. SQL Server 2019 and external hard drive are capitalized to data storage.

In this paper, asymmetric cryptography, history of democracy, worldwide e-voting experiences and well-known encryption algorithms issues are presented. Furthermore, the classification of electronic voting systems and the risk of using modern technologies are also introduced. The paper contains original implementation of RSA algorithm. All goals have been achieved. The result of the study is Referendum application which has been properly tested. Description of conducted tests and the platform in action are included in the paper.

Key words: i-voting, asymmetric cryptography, RSA, Python, C#

1 Wstęp

1.1 Cel i zakres pracy

Celem pracy jest wykorzystanie kriptografii asymetrycznej do anonimizowania decyzji wyborców podczas przeprowadzania głosowania internetowego. Efektem prac będzie stworzenie systemu przeznaczonego do internetowego przeprowadzania ogólnokrajowego referendum. Platforma powinna umożliwiać weryfikowanie użytkowników poprzez wiadomości SMS, a także gromadzić dane na temat geolokalizacji. Wybrana metoda szyfrowania winna zostać zaimplementowana tak, aby spełniała obecne standardy bezpieczeństwa w zakresie długości kluczy. Dodatkowo opracowany system powinien opierać swą ideę na rozwiązańach, które są akceptowalne przez polskie prawo, a także posiadają ogólnoświatowe wzorce.

1.2 Dostępne rozwiązania

Wykorzystywanie nowoczesnych technologii w procesie wyborczym może przyczynić się do zwiększenia udziału obywateli w sprawowaniu władzy, a także zmniejszenia kosztów finansowych ponoszonych na organizację wyborów.

Dostępnym i wykorzystywany systemem w procesie głosowanie internetowego jest rozwiązanie estońskie [6]. Wybory parlamentarne w Estonii w 2019 roku ukazały skalę popularności internetowej platformy. Ponad 43% głosów zostało oddanych za pośrednictwem Internetu, co skutkowało przyrostem ponad 60 000 elektronicznych głosów w porównaniu do poprzednich wyborów. Estoński “IVXV” to system end-to-end, którego działanie można kompleksowo weryfikować. Wykorzystuje metody kryptograficzne umożliwiające wyborcy sprawdzenie, czy oddany głos został zaliczony jako ważny. Weryfikacja odbywa się bez ujawniania poufnych danych do wiadomości publicznej.

Głosowanie przez Internet było możliwe w Szwajcarii od początku XX wieku aż do 2019 roku, gdy odkryto lukę w zabezpieczeniach. Obecnie trwają prace nad opracowaniem nowego systemu [7]. Narzędzia cyfrowego wsparcia procesu wyborczego są używane w takich krajach jak Stany Zjednoczone Ameryki, a do niedawna także w Holandii, która po kilkuletniej przerwie planuje znowu wykorzystać system do automatycznego zliczania głosów w nadchodzących wyborach.

1.3 Historia głosowania

Demokracja to termin pochodzący od greckiego słowa dēmokratīā, które dosłownie tłumaczy się jako rządy ludu. Obecnie za wzorcowy przykład demokracji uważa się ustrój Aten w V w. p.n.e [8]. Jednakże prodemokratyczne społeczności istniały już w cywilizacji Sumerów około 2600 lat p.n.e, a także w starożytnych Indiach w gana-sangha w VI w. p.n.e [9]. Popularną formą sprawowania władzy w demokracji ateńskiej były Zgromadzenia (gr. Ekklēsiae) [10]. Obywatele podejmowali decyzje poprzez podnoszenie rąk. Co ciekawe, nawet w XXI w. starogreckie tradycje są praktykowane. Jedną z dostępnych metod głosowania w Parlamencie Europejskim jest skierowanie kciuka w górę lub w dół. Następnie Przewodniczący ocenia zwyczaj bez dokładnego przeliczania, jaka jest wola zgromadzonych. Wynik może budzić kontrowersje, dlatego Przewodniczący posiada kompetencję do powtórzenia głosowania z wykorzystaniem systemu elektronicznego, aby rozwijać wszelkie wątpliwości [11]. Starożytni Grecy również byli świadomi słabości takiego szacowania. Badania archeologiczne Palladianu rozpoczęte w latach 60. XX w. potwierdziły, że greccy sędziowie wykorzystywali przedmioty pomagające ustalić ostateczny werdykt. Zdaniem ekspertów żetony umieszczano w specjalnych urnach, co było jednoznaczne z opowiedzeniem się za winą lub uniewinnieniem oskarżonego [12]. Rzymianie także wykorzystywali elementy demokracji w sprawowaniu władzy. Choć zdaniem dr hab. Hanny Appel z Uniwersytetu Mikołaja Kopernika w Toruniu - historyczki i filolożki specjalizującej się w dziejach Wiecznego Miasta - nie próbowali skutecznie aktywizować większości obywateli w procesie wyborczym, co skutkowało frekwencją na poziomie co najwyżej 2% [13]. W przeciwieństwie do starożytnych Greków odrzucili pestki, liście laurowe czy muszle, jako przedmioty użyteczne w procesie wyborczym. Rzymianie wykorzystywali drewniane tabliczki w celu wyrażenia stanowiska poprzez umieszczenie nazwiska lub inicjałów kandydata. W procesie ustawodawczym i sądowniczym tabliczki były niewielkich rozmiarów, ponieważ zapisywano jedną literę. Pozytywny głos wyrażano zazwyczaj umieszczając symbol V, natomiast negatywny poprzez a [14]. Upadek Cesarstwa Rzymskiego rozpoczął epokę średniowiecza, w którym elementy dorobku antycznych ojców demokracji częściowo wdrażano we włoskich republikach miejskich. W okresie renesansu rozwijała się forma przedstawicielska jako wyraz woli narodu na obszarach dzisiejszej Anglii oraz Polski, co stanowiło wyraźny kontrast wobec absolutyzmu monarszego. Rewolucja angielska w XVII w., powstanie Stanów Zjednoczonych oraz wybuch rewolucji francuskiej przyczyniły się do dynamiczniejszego rozwoju ustrojów demokratycznych na świecie.

2 Kryptografia asymetryczna

Kryptografia jest dziedziną kryptologii, czyli nauki opartej na formalnym i zaawansowanym aparacie matematycznym. Kryptografia obejmuje zagadnienia poświęcone anonimizowaniu i utajnianiu danych przed nienależytym dostępem. Dane są trudne do rozszyfrowania przez podmiot nieznający klucza rozszyfrowywującego.

Kryptografia asymetryczna jest rodzajem kryptografii, w której występuje para kluczy:

1. klucz publiczny – ciąg danych o dostępie publicznym służący do zaszyfrowania danych.
2. klucz prywatny – tajny ciąg danych umożliwiający rozszyfrowanie danych.

2.1 Funkcje jednokierunkowe

Funkcje jednokierunkowe są jednym z ważniejszych narzędzi kryptografii asymetrycznej. Łatwość obliczeń w jedną stronę przy jednoczesnej trudności wyliczenia w drugą stronę stanowi zdecydowaną ich zaletę.

Definicja 2.1 (Funkcja jednokierunkowa) *Funkcja f jest funkcją jednokierunkową, jeśli:*

1. $y = f(x)$ jest łatwe do obliczenia oraz
2. $x = f^{-1}(y)$ jest bardzo trudne do obliczenia

Wyrażenie uznaje się za łatwe do obliczenia, gdy czasowa złożoność obliczeniowa jest wielomianowa. Natomiast za bardzo trudne do obliczenia uznaje się wyrażenia, które przekraczają obecne możliwości lub posiadają złożoność czasową uniemożliwiającą uzyskanie wyniku w rozsądny czasie. Jednocześnie nie można wykluczyć, że w przyszłości pojawią się algorytmy o korzystniejszej złożoności. Jak dotąd nie udowodniono, że jakakolwiek funkcja jest jednokierunkowa. Gdyby udało się stworzyć taki dowód, to rozwiązano бы jeden z problemów milenijnych [15]. Mianowicie wykazano by, iż $P \subset NP$ [16]. Obecnie można wyodrębnić trzy główne rodziny asymetrycznych algorytmów kryptograficznych. Pierwsza z nich to schemat faktoryzacji całkowitej bazujący na trudności rozkładu na czynniki pierwsze dużych liczb złożonych. Druga rodzina algorytmów opiera się na problemie logarytmu dyskretnego

w ciałach skończonych. Można także wyszczególnić kolejną grupę stanowiącą uogólnienie logarytmu dyskretnego poprzez schemat krzywej eliptycznej.

2.2 Arytmetyka modularna

Algorytmy kryptograficzne zarówno asymetryczne jak i symetryczne często opierają się na arytmetyce w ramach skończonej liczby elementów. Arytmetyka modularna jest systemem liczb całkowitych, w którym liczby przyjmują wartości od 0 do $n - 1$. Liczba n jest dowolną nieujemną liczbą całkowitą.

Definicja 2.2 (Operacja modulo) Niech $a, r, m \in \mathbb{Z}$ oraz $m > 0$, wtedy

$$a \equiv r \pmod{m}$$

jeśli m jest podzielne przez $a - r$.

r jest resztą z dzielenia a przez m .

Przykład 2.1 Rozważmy zbiór $\{0, 1, 2, 3, 4, 5\}$. Działanie $7 \pmod{6}$ możemy zapisać jako:

$$5 + 2 \equiv 1 \pmod{6}$$

Reszta z dzielenia 7 przez 6 wynosi 1.

2.3 Problem logarytmu dyskretnego

Logarytm $\log_a b$ jest liczbą c taką, że $a^c = b$ dla $a, b > 0 \wedge a \neq 1 \wedge a, b \in \mathbb{Z}$. Obliczenie wartości $\log_a b$ sprowadza się zatem do znalezienia odpowiedzi na pytanie, do jakiej potęgi należy podnieść a , aby otrzymać b . Teoria dotycząca grup, podgrup, a także innych terminów stanowi prymarną introdukcję w celu zrozumienia problemu logarytmu dyskretnego.

Definicja 2.3 (Grupa) Zbiór G z działaniem $\circ : G \times G \rightarrow G$ nazywamy grupą, jeśli spełnione są warunki:

1. działanie \circ jest łączne, czyli

$$\forall_{a,b,c \in G} a \circ (b \circ c) = (a \circ b) \circ c,$$

2. istnieje dokładnie jeden element $e \in G$ taki, że

$$\forall_{a \in G} a \circ e = e \circ a = a,$$

3.

$$\forall_{a \in G} \exists_{b \in G} a \circ b = b \circ a = e.$$

Przykładem grupy jest zbiór liczb rzeczywistych z dodawaniem oraz zerem będącym elementem neutralnym. Inne exemplum można zauważyć w zbiorze D_n wszystkich izometrii n-kąta foremnego ($n \geq 3$) wraz ze składaniem izometrii oraz przekształceniem identycznościowym - elementem neutralnym [17]. Jako antyprzykład grupy można wskazać zbiór liczb całkowitych bez zera z działaniem mnożenia, ponieważ brak a^{-1} dla $a \in \mathbb{Z}$ z wyjątkiem -1 oraz 1.

Twierdzenie 2.1 (Podgrupa) *Niepusty podzbiór $H \subset G$ jest podgrupą grupy G wtedy i tylko wtedy, gdy spełnione są warunki:*

1.

$$\forall_{a \in H} a^{-1} \in H.$$

2.

$$\forall_{a,b \in H} a \circ b \in H.$$

Przykład stanowią między innymi zbiory liczb całkowitych i wymiernych z działaniem dodawania będące podgrupami grupy zbioru liczb rzeczywistych, a także zbiór obrotów wokół środka symetrii n-kąta foremnego ($n \geq 3$).

Definicja 2.4 (Grupa cykliczna) *Jeśli każdy element grupy G jest postaci a^k oraz $a \in G$, to wówczas G jest grupą cykliczną generowaną przez element a .*

Jako exemplum możemy wskazać $\langle i \rangle = (\{1, 1, i, i\}, \cdot)$, które jest grupą cykliczną generowaną przez element i . Taka grupę można przedstawić jako $(\{1, i, i^2, i^3\}, \cdot)$.

Definicja 2.5 (Grupa skończona) Grupa (G, \circ) jest skończona, jeśli posiada skończoną liczbę elementów. Wówczas liczебność lub porządek grupy oznacza się jako $|G|$.

Przykładem grupy skończonej jest $(\mathbb{Z}_n, +)$. Liczebność grupy to $|\mathbb{Z}_n| = n$, gdyż $\mathbb{Z}_n = \{0, 1, 2, 3, \dots, n - 1\}$.

Twierdzenie 2.2 Zbiór \mathbb{Z}_n^* zawierający wszystkie liczby całkowite $i = \{0, 1, 2, \dots, n - 1\}$, dla których $NWD(i, n) = 1$ tworzy grupę abelową z mnożeniem modulo. Element neutralny wynosi $e = 1$.

Definicja 2.6 (Grupa abelowa) Grupę (G, \circ) nazywa się abelową lub przemienną, jeżeli działanie \circ w niej zawarte jest takie, że:

$$a \circ b = b \circ a$$

dla dowolnego $a, b \in G$

Problem logarytmu można opisać także dla logarytmu dyskretnego w ciałach skończonych o p elementach [18].

Definicja 2.7 (Problem logarytmu dyskretnego w ciałach skończonych o p elementach)

Mając skończoną cykliczną grupę \mathbb{Z}_p^* o porządku $p - 1$ i prymitywnym elemencie $\alpha, \beta \in \mathbb{Z}_p^*$ oraz $\alpha \neq \beta$, wówczas problemem logarytmu dyskretnego jest wyznaczenie liczby całkowitej $1 \leq x \leq p - 1$ takiej, że:

$$\alpha^x \equiv \beta \pmod{p}$$

Jako przykład można wskazać $2^x \equiv 36 \pmod{47}$. Jednym ze sposobów na ustalenie wyniku jest sprawdzenie wszystkich możliwych kombinacji. Rozwiązaniem dla tego przykładu będzie $x = 17$. Problem logarytmu dyskretnego można uogólnić, co sprawia, że jest tak szeroko wykorzystywany w kryptografii. Nie jest ograniczony tylko do grupy multiplikatywnej (tj. z działaniem mnożenia) \mathbb{Z}_p^* [19]. Może być zdefiniowany dla dowolnej grupy cyklicznej.

Definicja 2.8 (Uogólnienie problemu logarytmu dyskretnego) Mając skończoną cykliczną grupę G z działaniem \circ oraz mocy n , rozważmy dwa różne elementy $\alpha \in G$ oraz $\beta \in G$, dla których problemem logarytmu dyskretnego będzie znalezienie liczby całkowitej $x \in [1, n]$ takiej, że:

$$\beta = \underbrace{\alpha \circ \alpha \circ \dots \circ \alpha}_x = \alpha^x$$

2.4 Algorytm Rivesta-Shamira-Adlemana (RSA)

Algorytm RSA został zaprojektowany w 1977 roku przez Rona Rivesta, Adiego Shamira oraz Leonarda Adlemana. Bezpieczeństwo szyfrowania zapewnia trudność rozkładu na czynniki pierwsze wyniku mnożenia dwóch dużych liczb pierwszych. Każda z dużych liczb pierwszych powinna posiadać co najmniej 100 cyfr w systemie dziesiętnym. Operacje potęgowania dyskretnego są wykorzystywane przy szyfrowaniu i deszyfrowaniu.

2.4.1 Generowanie kluczy

Dobór kluczy jest najistotniejszym krokiem w działaniu algorytmu RSA. Stanowi to cechę charakterystyczną dla wszystkich algorytmów asymetrycznych. Generowanie kluczy zwykle nie jest problematyczne dla szyfrów strumieniowych oraz blokowych.

Wynik operacji generowania kluczy:

- klucz publiczny: $k_{pub} = (n, e)$
- klucz prywatny: $k_{pr} = (d)$

Lista kroków:

1. Wybierz losowo dwie duże liczby pierwsze p i q . Ze względów bezpieczeństwa powinny mieć podobną wielkość, ale różnić się długością o kilka cyfr.
2. Oblicz $n = p \cdot q$
3. Oblicz $\varphi(n) = (p - 1)(q - 1)$

4. Wybierz wykładnik $e \in \{1, 2, \dots, \varphi(n) - 1\}$ dla którego prawdziwy będzie warunek

$$NWD(e, \varphi(n)) = 1$$

5. Oblicz klucz prywatny $d = e^{-1} \pmod{\varphi(n)}$

Funkcja $NWD()$ zwraca największy wspólny dzielnik dla pary liczb e oraz $\varphi(n)$ [20].

2.4.2 Szyfrowanie

Dany jest klucz publiczny $(n, e) = k_{pub}$ oraz wiadomość do zaszyfrowania x , wtedy funkcja szyfrująca przyjmuje postać:

$$y = e_{k_{pub}}(x) \equiv x^e \pmod{n}$$

Wiadomość do zaszyfrowania x jest rozważana jako ciąg bitów oraz $x, y \in \mathbb{Z}_n$

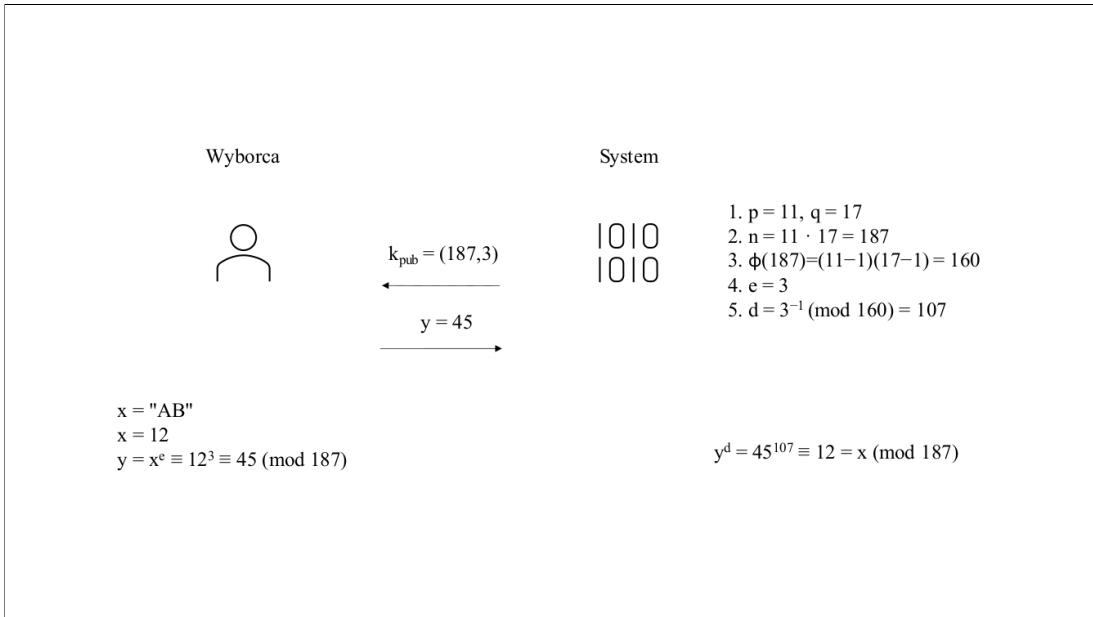
2.4.3 Deszyfrowanie

Dany jest klucz prywatny $d = k_{pr}$ oraz szyfrogram y , wtedy funkcja deszyfrująca przyjmuje postać:

$$x = d_{k_{pr}}(y) \equiv y^d \pmod{n}$$

Wiadomość do rozszyfrowania x jest rozważana jako ciąg bitów oraz $x, y \in \mathbb{Z}_n$

Poniżej zaprezentowano przykład działania algorytmu RSA. Wybrane liczby pierwsze nie są wystarczająco duże, aby zapewnić bezpieczeństwo. Implementacja algorytmu RSA użytego w projekcie znajduje się w rozdziale 4.3.1



Rysunek 1: Przykład działania RSA

2.5 Algorytm ElGamala (ELG)

Algorytm ELG został zaprojektowany w 1985 roku przez Tahera Elgamala. Opiera się na problemie logarytmu dyskretnego oraz koncepcji Diffiego–Hellmana. Szyfrowany tekst za każdym razem daje inny szyfrrogram z uwagi na fakt, iż występuje losowe dobieranie parametrów.

Istotnym elementem w zgłębieniu koncepcji Diffiego-Hellmana jest zrozumienie podstaw z zakresu algebry. Teoria dotycząca grup, podgrup, a także grup skończonych oraz cyklicznych stanowi prymarną introdukcję tego protokołu. Informacje na ten temat można znaleźć w rozdziale 2.3.

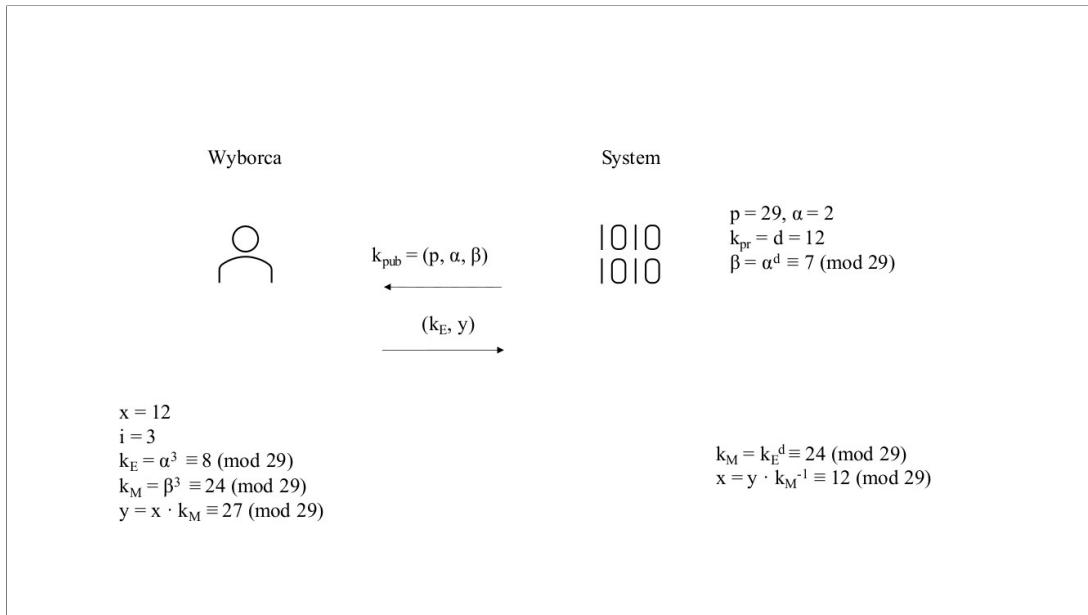
Introdukcja z zakresu podstaw algebry oraz problemu logarytmu dyskretnego stanowiła wystarczający materiał, aby przejść do analizowania problemu Diffiego-Hellmana. Poniżej zaprezentowane uogólnienie było bazą dla opracowania algorytmu ElGamala.

Definicja 2.9 (Problem Diffiego-Hellmana - uogólnienie) *Dana jest skończona, cykliczna grupa G rzędu n i $\alpha \in G$. Elementy $A = \alpha^a$ oraz $B = \alpha^b$ należą do G . Problemem Diffiego-Hellmana jest znalezienie elementu α^{ab} .*

W algorytmie ElGamala można wyodrębnić trzy etapy. Faza pierwsza polega na wyborze klu-

cka prywatnego oraz wyznaczeniu klucza publicznego, a następnie przekazaniu publicznego klucza. Kolejnymi etapami są szyfrowanie i deszyfrowanie wiadomości. Formalne ujęcie protokołu ElGamala zmienia kolejność operacji prezentowanych w koncepcji Diffiego-Hellmana.

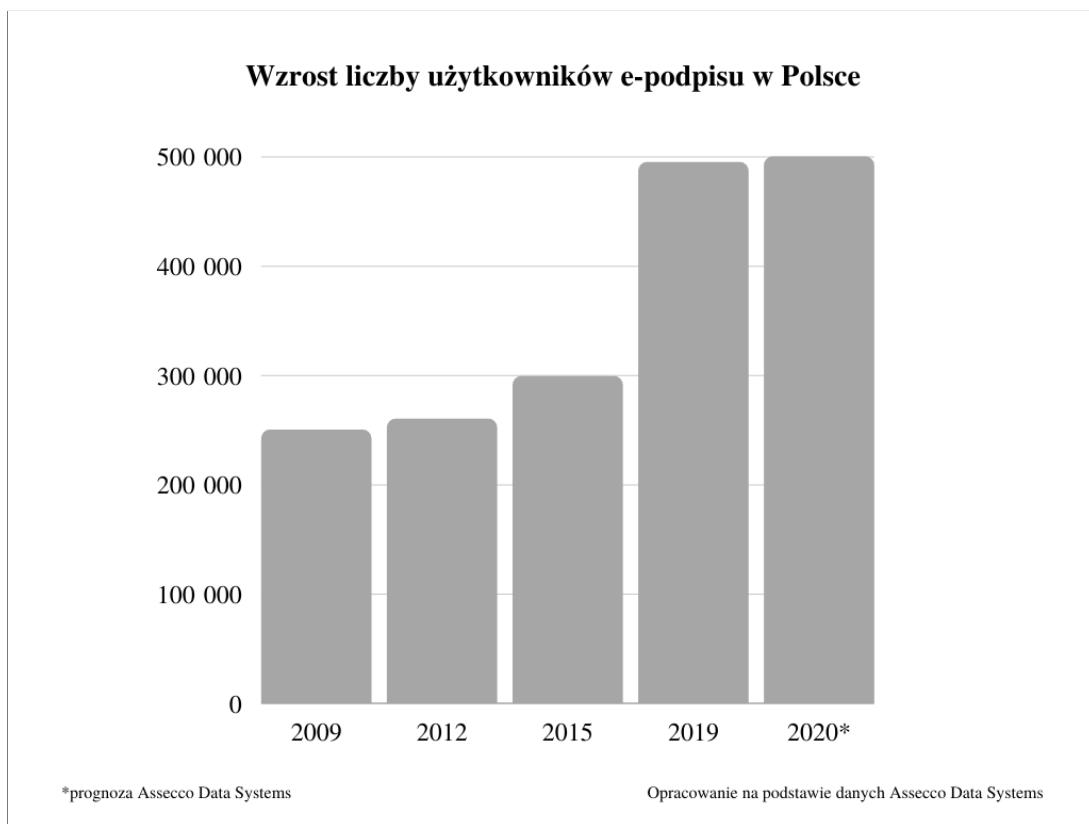
Działanie protokołu można zaprezentować na przykładzie wymiany wiadomości między wyborcą a systemem do głosowania. W pierwszej fazie system wybiera dużą liczbę pierwszą p oraz wyznacza prymitywny element $\alpha \in \mathbb{Z}_p^*$ lub α należący do podgrupy \mathbb{Z}_p^* . Kolejny krok stanowi wybór klucza prywatnego $k_{pr} = d \in \{2, \dots, p-2\}$, a następnie obliczenie klucza publicznego $k_{pub} = \beta = \alpha^d \pmod{p}$. Fazę kończy przekazanie klucza $k_{pub} = (p, \alpha, \beta)$ do konkretnego wyborcy. Kolejny etap polega na odebraniu klucza od systemu, a następnie wyznaczeniu parametru $i \in \{2, \dots, p-2\}$ po stronie wyborcy. W dalszej części obliczany jest klucz krótkotrwały $k_E \equiv \alpha^i \pmod{p}$ oraz klucz maskujący $k_M \equiv \beta^i \pmod{p}$. Przygotowane klucze wykorzystuje się do szyfrowania wiadomości $x \in \mathbb{Z}_p^*$, które można zapisać jako $y \equiv x \cdot k_M \pmod{p}$. Wiadomość przekazana jest do systemu jako (k_E, y) . W ostatniej fazie dochodzi do odebrania informacji od wyborcy oraz obliczenia klucza maskującego $k_M \equiv k_E^d \pmod{p}$. Element końcowy stanowi odszyfrowanie wiadomości $x \equiv y \cdot k_M^{-1} \pmod{p}$.



Rysunek 2: Przykład działania protokołu ElGamala

2.6 Podpis cyfrowy

Postępująca digitalizacja niesie ze sobą nie tylko korzyści i ułatwienia, ale także nowe wyzwania w zakresie bezpieczeństwa. Ochrona tożsamości jest niezwykle ważna zarówno podczas procesu wyborczego, jak i przy okazji dokonywania codziennych transakcji bankowych. W przeszło rozwijającej się rzeczywistości zaistniała konieczność wprowadzenia cyfrowego odpowiednika podpisu. Popularnym sposobem na sprawdzenie autentyczności dokumentów oraz wiadomości elektronicznych stał się podpis cyfrowy. Początek upowszechniania podpisu elektronicznego w Polsce datuje się na 1998 rok. Uruchomiono wówczas Centrum Certyfikacji Certum, które świadczyło usługi certyfikacyjne bazujące na technologii PKI. Pierwszy kwalifikowany podpis elektroniczny został wystawiony 31 stycznia 2003 roku ówczesnemu Prezesowi Rady Ministrów Leszkowi Milerowi. Obecnie popularność tego rozwiązania z każdym rokiem rośnie, a firma Assecco Data Systems szacuje, że w 2020 roku ilość nowych użytkowników kwalifikowanego e-podpisu w Polsce zwiększy się o pół miliona.



Rysunek 3: Wzrost liczby użytkowników e-podpisu w Polsce

Podobnie jak w przypadku tradycyjnych podpisów, tylko osoba, która ma do tego odpowiednie uprawnienia, powinna mieć możliwość kwitowania. Podstawową ideą podpisu

cyfrowego jest to, że podpisujący wiadomość używa do tego celu klucza prywatnego, natomiast ważność poświadczana się poprzez klucz publiczny. Ogólny schemat protokołu podpisu cyfrowego składa się z kilku etapów. W pierwszym kroku nadawca generuje klucz prywatny oraz publiczny. Następnie publikuje klucz publiczny. Kolejną czynnością, jaką musi wykonać nadawca, jest wygenerowanie sygnatury poprzez wykorzystanie klucza prywatnego oraz wiadomości. Ostatnia z czynności, za którą odpowiada nadawca, polega na wysłaniu wiadomości wraz z sygnaturą. Odbiorca na podstawie klucza publicznego i sygnatury może ocenić ważność dokumentu.

3 Bezpieczeństwo

Systemy do głosowania elektronicznego muszą umożliwiać bezpieczne przeprowadzenie wyborów. Proces nie może być w jakimkolwiek momencie zakłócony, gdyż istniałoby ryzyko fałszerstwa, a w konsekwencji utraty społecznego zaufania. W Polsce nad przebiegiem głosowania czuwa Państwo Komisja Wyborcza, a przepisy regulujące jej działania są ujęte w Kodeksie Wyborczym. Do kompetencji PKW należy kontrola prawidłowości prowadzenia i aktualizowania rejestru wyborców, a także sporządzania spisów wyborców. PKW zajmuje się również badaniem zgodności danych rejestru wyborców i spisów wyborców z danymi ewidencji ludności i aktów stanu cywilnego w każdej gminie. Istotnym obowiązkiem PKW jest także powoływanie okręgowych i rejonowych komisji wyborczych, a po zakończeniu głosowania opublikowanie wyników. Zgodnie z Konstytucją Rzeczypospolitej Polskiej obywatel ma prawo udziału w referendum oraz prawo wybierania Prezydenta Rzeczypospolitej, posłów, senatorów i przedstawicieli do organów samorządu terytorialnego, jeżeli najpóźniej w dniu głosowania kończy 18 lat. Nad zapewnieniem prawidłowego przebiegu procesu wyborczego nadzór sprawuje przewodniczący wraz z członkami okręgowych komisji wyborczych, urzędnicy wyborczy, a także mężowie zaufania. Ostatnia z wymienionych grup jest szczególnie ważna, gdyż realizuje obywatelską formę kontroli nad rzetelnością przebiegu wyborów.

Wdrożenie elektronicznych metod głosowania przyniosłoby nie tylko korzyści, ale także nowe zagrożenia. Rozważając wszelkie za i przeciw należy mieć na uwadze również fakt, że proces prodemokratycznych przemian będzie kształtał się w Polsce przez wiele lat. Wynika to nie tylko z faktu zaniedbań w edukacji i budowaniu społeczeństwa obywatelskiego, lecz również z powodu wciąż niewystarczającego poziomu cyfryzacji w życiu publicznym. Trudna sytuacja wywołana pandemią COVID-19 przyczyniła się do bardziej dynamicznego rozwoju i powszechności w korzystaniu z nowoczesnych technologii. Według badań Głównego Urzędu Statystycznego w 2020 roku dostęp do Internetu posiadało 90,4% gospodarstw domowych i było to aż 3,7 punktów procentowych więcej niż w roku poprzednim [22]. Konieczność prowadzenie edukacji na odległość wpłynęła także na strukturę dostępności, gdyż połączenie z siecią częściej posiadały gospodarstwa domowe z dziećmi niż bez nich. Jednakże wciąż widoczna jest rozbieżność ze względu na klasę miejsca zamieszkania. Wyjątkowa sytuacja wpłynęła również na zwiększenie liczby osób korzystających z usług administracji publicznej za pomocą Internetu. W 2020 roku 41,9% osób w wieku 16–74 lata komunikowało się z administracją publiczną

poprzez stronę internetową. Stanowiło to wzrost o 1,5 p. proc. w porównaniu do poprzedniego roku.

Bezpieczeństwo procesu wyborczego jest uzależnione także od powszechności. Konstytucja Rzeczypospolitej Polskiej gwarantuje bowiem zarówno czynny jak i bierny udział w wyborach, dlatego rozwój technologii głosowania elektronicznego powinien usprawniać nie tylko sam moment oddania głosu, ale również czynności poprzedzające, takie jak chociażby zbiórka podpisów poparcia dla kandydata czy kampania referendalna.

3.1 Klasyfikacja systemów

Cechy bezpieczeństwa pomagają w ocenie systemów do głosowania elektronicznego [23]. Spełnienie poniższych zasad skutkuje uznaniem danego rozwiązania za bezpieczne lub nieodpowiednie do wykorzystania w procesie wyborczym. Zgodnie z obowiązującym prawem w Polsce wybory do ciał przedstawicielskich są tajne, dlatego ważną cechą jest prywatność (ang. privacy). Obywatele powinni mieć możliwość wyrażenia swojej woli bez obawy o możliwość przypisania konkretnego głosu do wyborcy. Warto przy tym wspomnieć, że głosowanie korespondencyjne to powszechnie akceptowalne rozwiązanie. Estoński system wykorzystuje ideę tej metody i skutecznie implementuje mechanizm podwójnego kopertowania. Kolejną niezwykle istotną cechą systemu jest weryfikowalność (ang. verifiability). Należy jednak rozróżnić osobistą weryfikowalność (ang. personal verifiability) oraz powszechną weryfikowalność (ang. universal verifiability). Pierwsza z wymieniony gwarantuje każdemu wyborcy możliwość sprawdzenia, czy jego głos został zakwalifikowany jako poprawny i uwzględniony w ostatecznym wyniku. Druga metoda umożliwia zaś wgląd do szczegółowych wyników, aby ostateczny rezultat nie był podważany przez ogół społeczeństwa. W tradycyjny procesie wyborczym eksponowaną pozycję zajmują mężczyźni zaufani jako społeczni nadzorcy, dlatego za jedną z cech bezpieczeństwa uznaje się bezsporność (ang. dispute-freeness). Niezwykle istotne jest, aby w dowolnym momencie wyborów, istniała możliwość weryfikacji czynność podejmowanych przez administrację oraz zgodności tychże działań z protokołem głosowania przez niezależną osobę. Czynnikiem warunkującym bezpieczeństwo jest także dokładność (ang. accuracy). Niedopuszczalne są zatem błędy w działaniu, które wpływają na werdykt głosowania, a wynik musi odzwierciedlać faktyczne decyzje wyborców. Kryterium weryfikujących zdatność systemu określa również niemożność kwitowania (ang. receipt-freeness), która uniemożliwia

niepodważalne potwierdzenie wyboru. Dzięki tej cesze zapobiega się sprzedaży oraz kupowaniu głosów, ponieważ nie ma możliwości wystawienia oficjalnego zaświadczenia o dokonanym wyborze. Ważnym wyznacznikiem bezpieczeństwa jest kwalifikowalność (ang. *eligibility*). Istnieje bowiem konieczność sprawdzenia, czy dany głos został oddany przez osobę, która ma do tego prawo. Zgodnie z obowiązującym w Polsce prawem głos może oddać obywatel, który w dniu wyborów kończy 18 lat oraz nie jest pozbawiony praw publicznych prawomocnym orzeczeniem sądu, praw wyborczych prawomocnym orzeczeniem Trybunału Stanu, ani nie został ubezwłasnowolniony prawomocnym orzeczeniem sądu [24]. System należy ponadto skonstruować w taki sposób, aby uniemożliwiła reutylizację (ang. *un-reusability*), czyli wyborca powinien oddać tylko jeden poprawny głos, aby każdy miał jednakowy wpływ na wynik. Istotne jest również to, aby system cechował się uczciwością (ang. *fairness*). Jego konstrukcja powinna uniemożliwić publikowanie częściowych wyników głosowania przed oficjalnym zakończeniem wyborów, aby wszystkie rozwiązania lub kandydaci zachowali równe szanse. Nie zawsze istnieje także możliwość zapewnienia pełnej sprawności, jednak elektroniczny system powinien cechować się rzetelnością (ang. *robustness*). Oznacza to, iż w przypadku awarii lub okresowych zakłóceń dalej gwarantuje bezpieczeństwo. Kolejną ważną cechę stanowi całkowitość (ang. *completeness*). Koniecznie należy przy tym wspomnieć, że wszystkie oddane głosy muszą zostać poprawnie zliczone. Kryterium określającym bezpieczeństwo jest również solidność (ang. *soundness*). Żaden wyborca nie powinien móc zakłócić procesu wyborczego poprzez przerwanie własnego procesu. Cechą wciąż aktualna jest także nienaruszalność (ang. *inalterability*). System powinien uniemożliwić zmianę głosu po zakończeniu głosowania zarówno przez wyborcę, jak i urzeników czy osoby postronne. W przyszłości istnieje szansa na zmianę interpretacji tej zasady z uwagi na możliwość wprowadzenia instytucji odwołania posła w trakcie trwania kadencji, jednakże nie są obecnie realizowane w Polsce żadne rządowe prace w tym zakresie. Cechą, która warunkuje bezpieczeństwo jest także nieprzymuszalność (ang. *incoercibility*), czyli konstrukcja systemu uniemożliwia handel głosami oraz nie zmusza do udziału w wyborach [25].

3.2 Zagrożenia

Cyfryzacja w administracji publicznej tworzy nie tylko korzyści, ale również nowe zagrożenia. Przeniesienie kluczowych dla funkcjonowania państwa obszarów do sieci i wyko-

rzystanie przy tym nowych środków przekazu jest obarczone ryzykiem. Nie istnieją bowiem metody, które pozwoliły raz na zawsze, a co najważniejsze skutecznie chronić danych, gdyż wymagałoby to całkowitego zahamowania rozwoju ludzkości. Przywołane w tej pracy rozwiązania kriptografii asymetrycznej posiadają swego rodzaju termin zdatności do bezpiecznego użycia. Istotna rolę odgrywa rozmiar kluczy. Poniżej zaprezentowano tabelę zawierającą rekommendacje, które zapewniają bezpieczeństwo.

Metoda	Data	Algorytmy symetryczne	Faktoryzacja modułu	Logarytm dyskretny		Krzywa eliptyczna
				klucz	grupa	
Lenstra / Verheul [26]	2021	86	1937	153	1937	163
ECRYPT [27]	2018-2028	128	3072	256	3072	256
NIST [28]	2019-2030	112	2048	224	2048	224
ANSSI [29]	2021-2030	128	2048	200	2048	256
BSI [30]	2020-2022	128	2000	250	2000	250

Tabela 1: Minimalny rozmiar klucza, który gwarantuje bezpieczeństwo oraz przewidywany termin ważności konkretnego szacowania. Długość wszystkich kluczy jest określona w bitach.

Przeniesienie wyborów do Internetu mogłoby doprowadzić do likwidacji mężów zaufania, którzy pełnią funkcję obywatelskich nadzorców nad przebiegiem głosowania. Gdyby zdecydowano się na wdrożenie w pełni automatycznych systemów, to tylko określona grupa osób byłaby w stanie skutecznie i świadomie weryfikować cały proces. Wymagałoby to dużego zaufania ze strony obywateli wobec Państwowej Komisji Wyborczej oraz polityków, aby wyniki nie były podważane. Ponadto informatyzacja na szczeblu wyłaniania ciał przedstawicielskich tworzyłaby konieczność zainwestowania środków w rozwój cyberbezpieczeństwa. Istniałoby bowiem ryzyko ataków ze stron organizacji terrorystycznych lub służb wywiadowczych, których celem mogłoby być sfałszowanie wyborów lub sabotowanie.

Wspomniane wyżej zagrożenia, to nie jedyne realne skutki wdrożenia elektronicznych narzędzi demokracji. Systemy do głosowania podobnie jak banki czy rządowe serwisy, mogłyby paść ofiarą ataków DDoS. Na dany sygnał hakera zainfekowane komputery próbują w jednej chwili skorzystać z danej strony lub usługi, co prowadzi do wyczerpania dostępnych zasobów. Innym możliwym zagrożeniem jest atak na algorytmy szyfrujące, które bazują na faktoryzacji modułu. Wiele rozwiązań korzysta z optymalizacji algorytmu RSA polegającej na użyciu chińskiego twierdzenia o resztach. Jednym ze sposobów przeciwdziałania temu zagrożeniu

jest doprowadzenie do sytuacji, w której odszyfrowanie zajmuje zawsze tyle samo czasu.

3.3 Doświadczenia ze świata

Wiele krajów ma doświadczenie we wdrażaniu elektronicznych systemów do głosowania, a także w zakresie tworzenia oprogramowania wspomagającego klasyczne wybory. W tym rozdziale opisano doświadczenia Ukrainy, Estonii oraz Holandii w procesie wdrażania nowoczesnych narzędzi demokracji.

3.3.1 Ukraina

Prodemokratyczne zmiany na Ukrainie przyczyniły się do intensywniejszego rozwoju oraz informatyzacji administracji publicznej. Mimo trwającej wojny kraj ten kontynuuje wdrażanie instrumentów pozwalających na coraz powszechniejszy udział obywateli w sprawowaniu władzy. Zwycięstwo Wołodymyra Zełenskiego w wyborach prezydenckich w 2019 roku zapewniło miejsce w debacie publicznej nowoczesnym narzędziom demokracji [31]. Ukraińskie władze postanowiły zainwestować w rozwiązania oparte o technologię blockchain, czyli zdecentralizowany system służący do przechowywania i przesyłania informacji. Zastosowanie wspomnianego rozwiązania umożliwia przeprowadzanie przejrzystych i audytowalnych wyborów za pośrednictwem Internetu. Niweluje także konieczność przeliczania głosów w sposób tradycyjny, co przekłada się na szybsze otrzymanie wyników.

Pierwszy ukraiński system do głosowania elektronicznego z wykorzystaniem blockchain został stworzony w 2016 roku. E-VOX:NaRada to dedykowane dla samorządów narzędzie do przetwarzania danych i głosowania, którego rezultat publikowany jest w Internecie. Rozwiązanie poprawia standardy sprawowania władzy i umożliwia proces weryfikacji działalności polityków w czasie rzeczywistym. Oprogramowanie udostępniono na zasadach darmowej licencji open-source, a za obszar testowy uznano urzędy w miastach rejonu bałckiego. Rozwiążane określa się mianem aplikacji zdecentralizowanej działającej jako program łańcucha bloków Ethereum [33]. Część przeznaczona dla klienta jest dedykowana dla systemu Android. Kod źródłowy platformy do głosowania opublikowano oraz umieszczono w sieci w 2016 roku, natomiast aktualizacji repozytorium dokonano w 2017 roku [32].

Blockchain jest coraz częściej wykorzystywany w aranżacjach internetowych głosowań.

Stworzono warunki do organizacji elekcji w 2017 roku do Rady Nadzorczej Ukraińskiej Fundacji Kultury, która jest koordynowana przez Ministerstwo Kultury Ukrainy. Rada Nadzorcza stanowi organ nadzorczy fundacji i składa się z 9 osób. Dwie osoby zostały wybrane w głosowaniu przez instytucje kulturalne, a kolejne dwie wyłoniono również przez sieć, lecz wyboru dokonali przedstawiciele społecznych organizacji zajmujących się kwestiami kultury. Nie wszyscy członkowie rady zostali wybrani przez Internet, ale jest to dobry początek drogi ku wdrażaniu nowoczesnych technologii w demokracji. Warto także zaznaczyć, że głosowanie nie było jednorazowym przedsięwzięciem i powrócono do tego ponownie w listopadzie oraz grudniu 2020 roku [34]. Ukraina testuje nie tylko rozwiązania bazujące na blockchain, ale daje szanse także innym koncepcjom. Dobrym przykładem są wybory z 2015 roku do rad publicznych przy Krajowym Biurze Antykorupcyjnym. Prace nad doskonaleniem platformy według zapewnień ukraińskich władz wciąż są kontynuowane, a ponowne wybory w maju 2020 roku uwiarygadniają prezentowane stanowisko [35].

Wykorzystywane przez Ukrainę systemy są różne, choć posiadają także podobieństwa. Niestety nie wszystkie zostały w pełni upublicznione, dlatego trudno przeprowadzić w tej kwestii dokładne i rzetelne badanie. Udostępnione dokumentacje techniczne także posiadają luki, które zaburzają proces poznawczy. Jednakże wdrożone rozwiązania pozwalają wyciągać pewne wnioski. Analizowane systemy do głosowania wykorzystują Internet, dlatego można określić je jako skalowane i przyszłościowe narzędzia do przeprowadzania powszechnych wyborów. Ambicje przedstawicieli władzy oraz zaawansowane prace nad systemami do głosowania elektronicznego kreują pozytywną wizję dla technologicznego i społecznego rozwoju Ukrainy.

3.3.2 Estonia

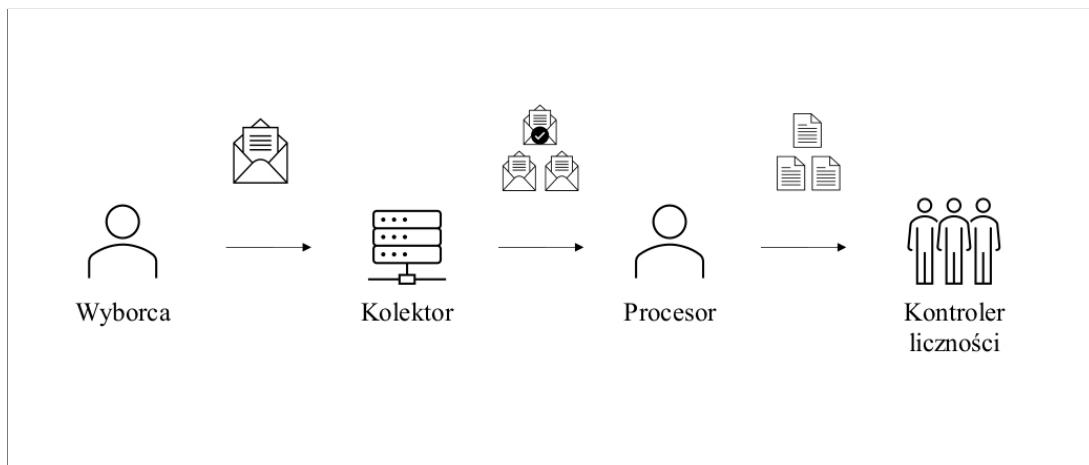
Głosowanie przez Internet w Estonii jest możliwe od 2005 roku. Wieloletnie prace nad rozwojem nowoczesnych narzędzi demokracji spowodowały, że ponad 43% wyborców skorzystało podczas ostatnich wyborów parlamentarnych z systemu do i-votingu [36]. Pierwsze kompleksowe badanie bezpieczeństwa przeprowadzono w 2003 roku, a następne w 2010 roku [37]. Kolejne lata były czasem, w którym wykrywano kolejne zagrożenia oraz niedoskonałości, co doprowadziło do powstania nowego systemu IVXV w 2017 roku.

Kod źródłowy platformy do głosowania został udostępniony w Internecie wraz z tech-

niczny opisem [6]. W dokumentacji wskazano także na funkcję organizatora, który sprawuje nadzór i rozdziela role, a ponadto decyduje o przechowywaniu kluczy prywatnych oraz zasadach zliczania głosów.

Ogólny schemat systemu IVXV zawiera następujące elementy oraz działania:

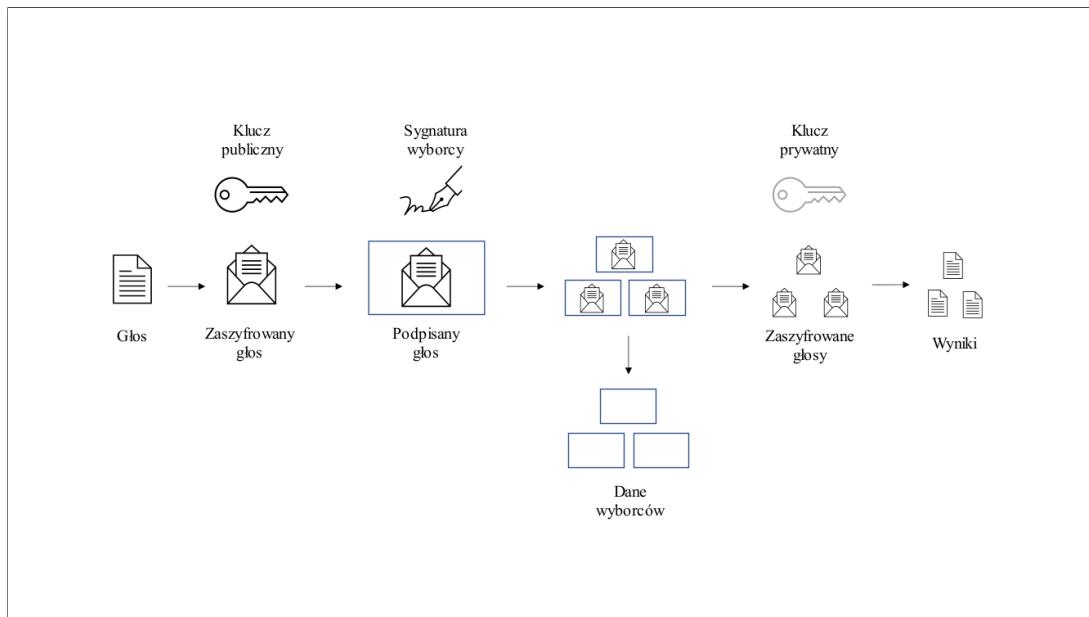
- Wyborca — głosuje za pomocą aplikacji, szyfruje, podpisuje cyfrowo i wysyła swój wybór do kolektora.
- Kolektor — pomaga w utworzeniu elektronicznego głosu, dostarcza i gromadzi niezbędne dane oraz przekazuje zebrane informacje pod koniec głosowania do procesora.
- Procesor — przetwarza zebrane głosy. Weryfikuje cyfrowe podpisy, unieważnia powtórzone głosy, również w przypadku głosowania jednocześnie w lokalu oraz przez system internetowy. Ponadto procesor sortuje głosy według okręgów, utajnia i przesyła do kontrolera liczności.
- Kontroler liczności — komponent organizatora wyborów pełniący funkcję nadzorcy. Przechowuje klucze prywatne. Otwiera anonimowe głosy oraz dodaje do wyniku.



Rysunek 4: Ogólny schemat systemu IVXV

Estoński system do głosowania bazuje na schemacie kopertowym, czyli jest implementacją tradycyjnych wyborów korespondencyjnych. Zamknięta koperta z głosem jest umieszczana w zewnętrznej kopercie zawierającej dane wyborcy oraz podpis cyfrowy. Aplikacja szyfruje głos i generuje losową liczbę, która jest wyborczym odpowiednikiem klucza publicznego.

Głos zaszyfrowany kluczem publicznym deszyfruje się za pomocą klucza prywatnego. Zliczanie głosów jest ostatnim etapem procesu wyborczego. W celu zapewnienia prywatności głosy nie są zliczane w czasie trwania głosowania.



Rysunek 5: Schemat koperty w systemie IVXV

W 2019 roku rozgorzała publiczna debata na temat IVXV, dlatego Ministerstwo Handlu Zagranicznego i Informatyki podjęło decyzję o powołaniu komitetu do przeprowadzenia analizy i nakreślenia strategii rozwoju dla elektronicznych narzędzi demokracji. Wyniki prac zostały udostępnione w kwietniu 2020 roku [38]. Scharakteryzowano także potencjalne podmioty chcące zaburzyć proces wyborczy.

Wyodrębniono następujące grupy:

- Niezależni hakerzy (ang. civil hacktivist) — osoby poszukujące rozgłosu, siejące dezinformację lub przeciwnicy rządu albo systemu politycznego;
 - Kandydaci — pojedyncze osoby chcąc uzyskać lepszy wynik wyborczy;
 - Zorganizowane grupy o podobnych poglądach lub przynależności partyjnej — osoby posiadające średnie zasoby techniczne o bardzo dobrych możliwościach skalowania ataków;
 - Zagraniczne podmioty — organizacje działające na zlecenie obcych wywiadów posiadające znaczne zasoby.

Najpoważniejsze ataki na estoński system polegały na nieautoryzowanym użyciu urządzeń e-ID, czyli elektronicznych dowodów osobistych. Skutkiem takiej luki może być ponowne głosowanie i nadpisane pierwotnego wyboru. Sugerowanym przez specjalistów rozwiązańem jest weryfikacja od początku do końca (E2E), która umożliwia sprawdzenie przez wyborcę, czy jego głos został zaliczony i uznany za ważny. Jednakże przedstawiona koncepcja nie gwarantuje uczciwości, ponieważ potencjalny wyborca mógł nie wziąć udziału w głosowaniu, a mimo to niepożądana osoba oddała za niego głos z użyciem fałszywego e-ID.

W styczniu 2021 roku opublikowano komunikat [39] o zmianie ustawy o referendum i ordynacji wyborczej, zniesieniu ciszy wyborczej oraz nowy harmonogram głosowania. Wprowadzono elektroniczną listę wyborców, dzięki której obywatele będą mogli głosować z dowolnego lokalu wyborczego, co wcześniej było utrudnione. Skrócono także czas przeznaczony na głosowanie. Od tego roku nie będzie już możliwe głosowanie online w ostatnim dniu wyborów, ale obywatel będzie mógł anulować podjętą przez Internet decyzję w niedzielę w lokalu wyborczym za pomocą karty do głosowania. Wszystko wskazuje na to, że wybory na urząd prezydenta oraz do rad gmin odbędą się w 2021 roku bez przeszkód.

3.3.3 Holandia

Holenderskie doświadczenia w zakresie głosowania elektronicznego uwidaczniają, że proces ten nie zawsze jest łatwy, przewidywalny oraz pozbawiony wad. Reorganizacja holenderskiego systemu koncentruje się na stworzeniu cyfrowych narzędzi wsparcia i jak dotąd nie osiągnięto w tej kwestii zadowalających efektów. Prace nad automatyzacją wyborów rozpoczęły się na przełomie XX i XXI wieku. Skupiono się na opracowaniu maszyn do głosowania, które z powodzeniem funkcjonują w Stanach Zjednoczonych Ameryki oraz Kanadzie [40]. Jednakże w 2007 roku maszyny zostały zakazane, a jako najważniejszy powód wskazano brak dowodów rzeczowych w postaci kart do głosowania. Nie zakazano jednak elektronicznego wsparcia procesu wyborczego, dlatego w 2008 roku Holenderska Rada Wyborcza rozpoczęła prace nad systemem ułatwiającym przeliczanie głosów. Powstałe narzędzie było szeroko wykorzystywane aż do roku 2017. Wówczas uznano, iż jest to system niebezpieczny i zawodny, dlatego Ministerstwo Spraw Wewnętrznych Holandii podjęło zdecydowane kroki mające na celu natychmiastowe wyłączenie elektronicznego wsparcia. W opublikowanym przez National Democratic Institute raporcie[41] wskazano wieloletnie problemy oraz brak możliwości nadzoru nad

stosowaną technologią, a do tego brak umiejętności określenia przez władzę dokładnych wymagań dotyczących funkcjonalności, bezpieczeństwa oraz integralności. Nieumiejętność wydawania wytycznych oraz sprawowania nadzoru doprowadziła do sytuacji, w której dostawcy nie aktualizowali technologii zgodnie z nowoczesnymi wymogami bezpieczeństwa. Ministerstwo zignorowało także problemy z maszynami w Irlandii, a także obiekcie zgłasiane przez Holenderską Radę Wyborczą.

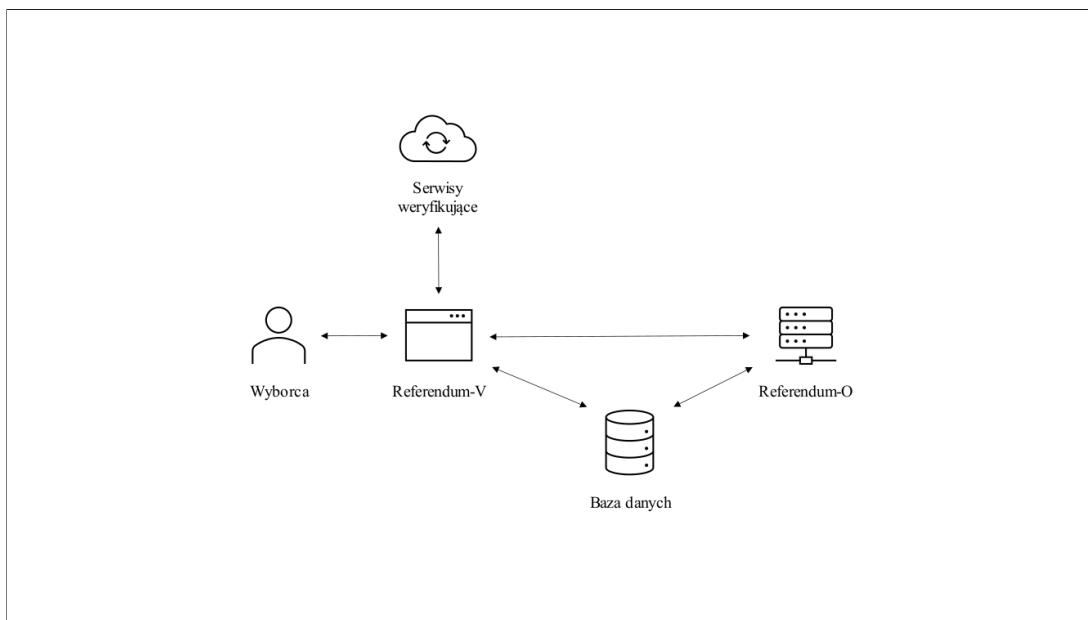
Po serii kontrowersyjnych referendum Senat zdecydował 10 lipca 2018 roku uchylić ustawę o referendum doradczym [42], co także negatywnie wpłynęło na dostępność insytucji demokracji bezpośredniej w Holandii. Trudna sytuacja doprowadziła do konfliktu między Holenderską Radą Wyborczą a Ministerstwem Spraw Wewnętrznych, co ostatecznie doprowadziło do powołania niezależnego mediatora, którego zadaniem było zażeganie sporu oraz doprowadzenie do nakreślenia planu. W lutym 2018 roku opublikowano rezultat tych prac. Zgłoszono między innymi konieczność stworzenia projektu i ogłoszenia przetargu na nowe narzędzie cyfrowe do wspierania procesu wyborczego. W wyniku przeprowadzonych działań uczyniono Holenderską Radę Wyborczą instytucją odpowiedzialną za utrzymanie, zarządzanie oraz rozwój narzędzi cyfrowych, a ponadto przyznano uprawnienia do monitorowania jakości tworzonych rozwiązań.

Holendrzy poczynili zdecydowane kroki w celu wykorzystania nowoczesnych technologii podczas planowanych wyborów w marcu 2021 roku. Rozpoczęto prace nad udoskonaleniem systemu Ondersteunende Software Verkiezingen (OSV), a także podjęto decyzję o nakreśleniu nowych wymagań bezpieczeństwa oraz koncepcji dla przyszłego narzędzia, które zastąpi dotychczas znane systemy. OSV2020 jest przeznaczony do elektronicznego wsparcia procesu wyborczego. W każdym lokalu wyborczym następuje ręczne przeliczenie głosów, natomiast nadzędne jednostki wyborcze wykorzystują OSV2020 do sumowania oraz rozkładu mandatów. Ponadto partie polityczne za pomocą odrębnych modułów dostarczają listy kandydatów. Całość składa się z trzech niezależnych komponentów: OSV2020-PP, OSV2020-KS oraz OSV2020-U. Pierwszy jest przeznaczony dla partii politycznych, natomiast drugi dedykowany jest dla kandydatów, a ostatni pozwala ustalić wyniki głosowania. Raport firmy Expleo [43] zlecony przez Holenderską Radę Wyborczą opublikowany w październiku 2020 roku prezentuje system OSV2020, jako gotowy i spełniający wymogi bezpieczeństwa. Można zatem domniemywać, że narzędzie zostanie ostatecznie wykorzystane podczas planowanych na marzec 2021 roku wyborów do holenderskiego parlamentu.

4 Aplikacja

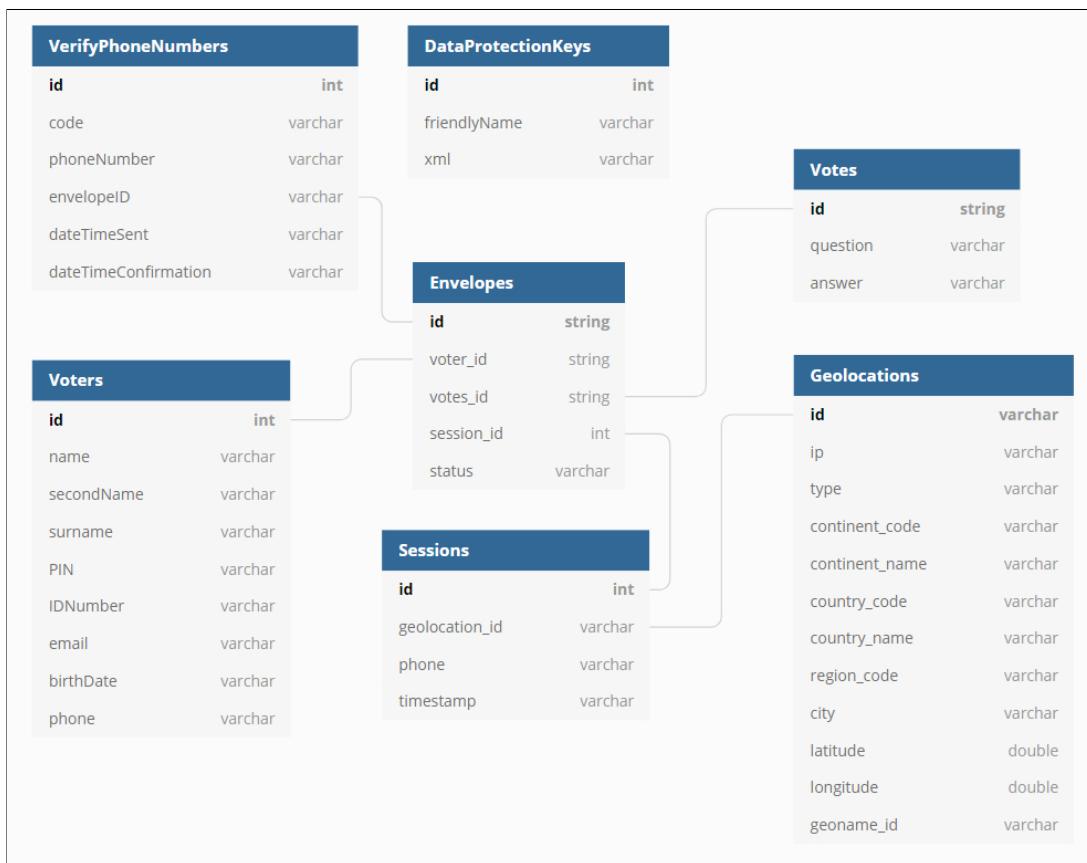
4.1 Ogólny opis

Opracowana platforma do głosowania internetowego składa się z dwóch głównych części. Aplikacja Referendum-O przygotowuje dane niezbędne do procesu szyfrowania, a także implementuje algorytm RSA, który umożliwia utajnienie informacji wyborczych oraz odpowiada za ustalenie wyniku poprzez końcowe zliczenie wszystkich poprawnych głosów. Aplikacja Referendum-V jest dedykowana dla wyborców, aby obsłużyć ich żądania, a ponadto zawiera mechanizmy do weryfikacji tożsamości. Aplikacja Referendum stworzona jest w ASP.NET CORE MVC 5.0 z wykorzystaniem języka C# oraz zawiera moduły zaprogramowane w środowisku Python 3.8. Platforma używa SQL Server 2019 do zarządzania bazą danych, a także przechowuje wrażliwe informacje na zewnętrznych dyskach.



Rysunek 6: Schemat ogólny platformy referendalnej

System do głosowania internetowego korzysta z dwóch magazynów danych. Baza *Keys* przechowuje klucze publiczne i kody SMS zawarte w tabelach *VerifyPhoneNumbers* oraz *DataProtectionKeys*. Pozostałe tabele są magazynowane w bazie *WebApplication*. Poniżej zaprezentowano schemat utworzonych tabel i łączących je relacji.



Rysunek 7: Schemat bazy danych

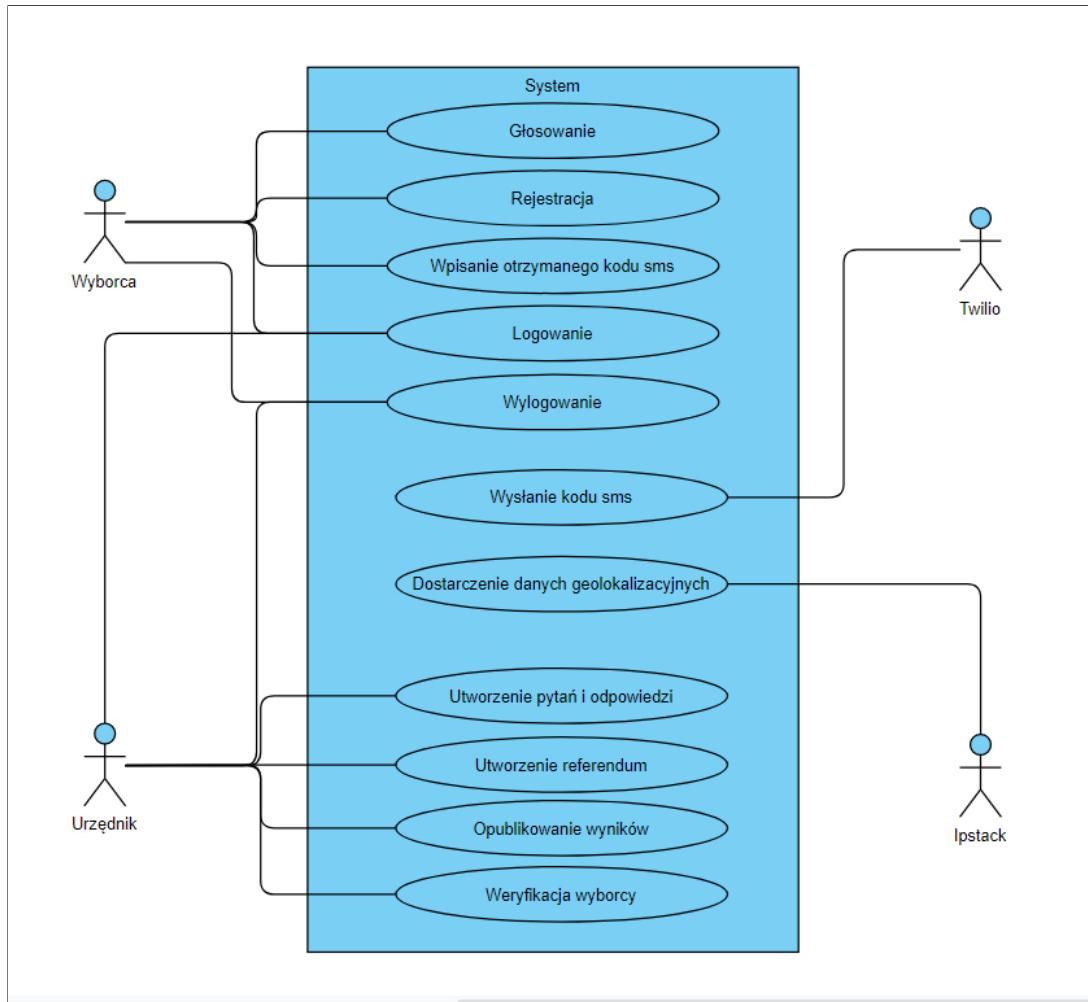
4.1.1 Perspektywa produktu

Platforma do głosowania ma szansę na dalszy rozwój zarówno na potrzeby sektora publicznego, jak i prywatnego. Algorytm RSA używa do utajniania decyzji wyborców 4096-bitowych kluczów, zatem spełnia aktualne wymogi bezpieczeństwa opisane w rozdziale 3.2. Obecnie nie są w Polsce wykorzystywane aplikacje do głosowania internetowego, które umożliwiałby organizację powszechnych wyborów do ciał przedstawicielskich lub ogólnokrajowych referendum. Projekt zawiera autorską implementację algorytmu RSA oraz stanowi elektroniczny odpowiednik koncepcji wyborów korenspondencyjnych.

4.1.2 Funkcje produktu

Platforma posiada niezbędne funkcje do przeprowadzenia internetowego referendum. Urzędniccy pełnią funkcję organizatorów, dlatego tworzą pytania, ustalają tematykę i czas trwania głosowania. Ponadto sprawują pieczę nad przebiegiem całego procesu oraz mogą analizo-

wać statusy głosów. Wyborcy posiadają tylko niezbędne możliwości interakcji z systemem polegające na wprowadzeniu danych osobowych, oddaniu głosu oraz wpisaniu otrzymanego kodu z wiadomości SMS. System komunikuje się także z zewnętrznymi serwisami weryfującymi.



Rysunek 8: Diagram przypadków użycia

4.1.3 Ograniczenia projektowe

Aplikacja Referendum jest dedykowana dla urządzeń z systemem Windows 10 lub nowszym. Wymagane jest posiadanie interpretera środowiska Python w wersji co najmniej 3.8. W celu zapewnienia poprawnego działania należy użyć przeglądarki Google Chrome w wersji 88.0.4324.104 lub nowszej.

4.1.4 Charakterystyka użytkowników

System do przeprowadzania internetowego referendum posiada mechanizmy do interakcji z niezależnymi podmiotami. Aktorzy określają spójny zbiór ról odgrywanych przez użytkowników podczas komunikacji z określonym przypadkiem użycia. Platforma posiada następujący pakiet aktorów:

- Urzędnik — nadzorca procesu wyborczego. Twórca pytań oraz odpowiedzi, a także czasowych ram referendum. Publikator wyników otrzymanych poprzez interakcję z narzędziami weryfikacyjnymi.
- Wyborca — użytkownik serwisu. Twórca konta posiadający dostęp do karty wyborczej po zalogowaniu.
- Twilio — zewnętrzny serwis dostarczający usługę wysyłania wiadomości SMS na potrzeby weryfikacji wyborcy.
- Ipstack — zewnętrzny serwis przekazujący informację na temat lokalizacji wyborcy na podstawie adresu IP.

4.2 Wymagania funkcjonalne

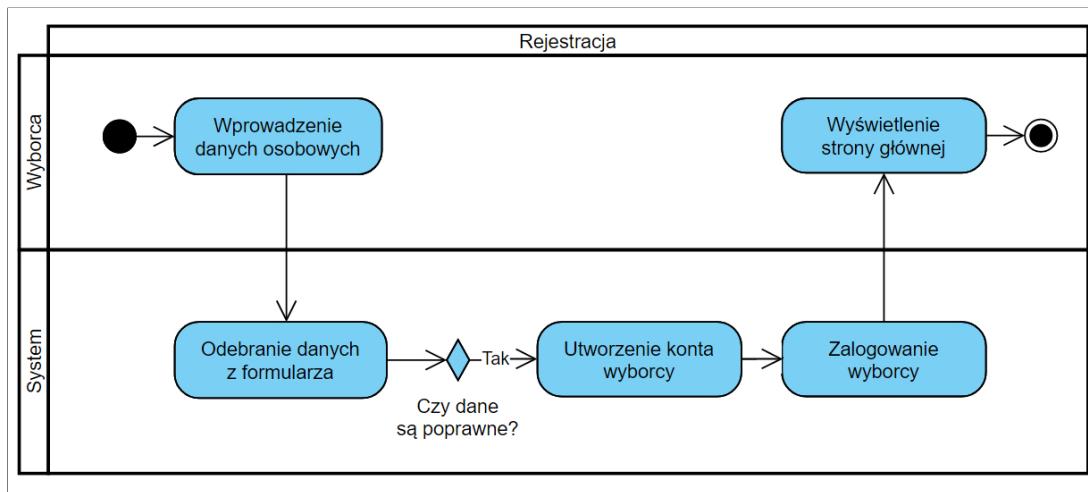
Serwis referendalny posiada szereg funkcjonalności, umożliwiających organizację internetowych wyborów. Niezwyczajne jest dokładne określenie zasad, atrybutów oraz sekwencji zdarzeń, gdyż nie można dopuścić do wystąpienia nieprzewidzianych zachowań. Poniżej zaprezentowano przypadki użycia określające rejestrację, logowanie, przygotowywanie referendum, a także inne czynności podejmowane podczas głosowania oraz finalizujące proces wyborczy. Przedstawiono zarówno sekwencje poprawne opisane poprzez ciągi podstawowe, jak i negatywne, wykorzystując przy tym ciągi alternatywne.

Rejestracja

Przypadek definiujący sposób rejestracji wyborcy w systemie do przeprowadzania referendum.

Ciąg podstawowy:

1. Wyborca wprowadza do odpowiednich pól formularza imię, drugie imię (opcjonalnie), nazwisko, datę urodzenia, numer dowodu osobistego, PESEL, adres e-mail oraz hasło.
2. Wyborca zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Zarejestruj".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę wyborcy.
5. Stwierdzenie zgodności z modelem wzywa utworzenie konta wyborcy.
6. Wyborca zostaje zalogowany.
7. Wyświetlenie strony głównej wraz z numerem PESEL zalogowanego wyborcy.

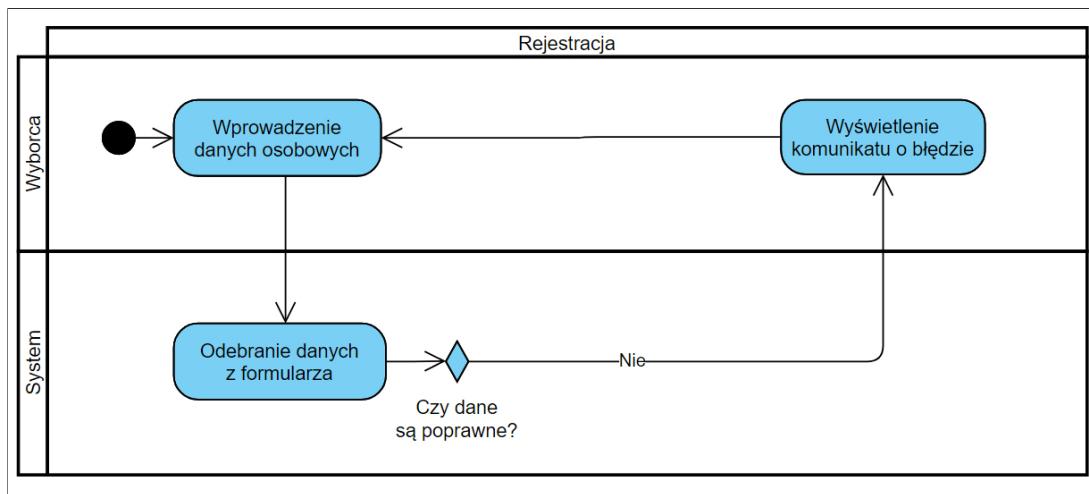


Rysunek 9: Ciąg podstawowy - rejestracja

Ciąg alternatywny:

1. Wyborca wprowadza do odpowiednich pól formularza imię, drugie imię, nazwisko, datę urodzenia, numer dowodu osobistego, PESEL, adres e-mail oraz hasło.
2. Wyborca zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Zarejestruj".

3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę wyborcy.
5. Stwierdzenie niezgodności z modelem wyborcy wzywa komunikat o błędzie.
6. Powrót do punktu nr 1.



Rysunek 10: Ciąg alternatywny - rejestracja

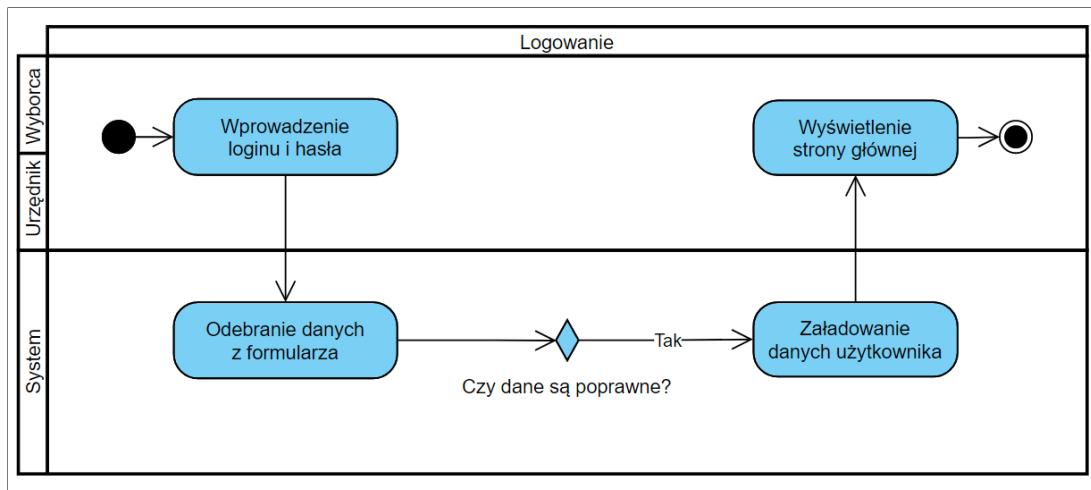
Logowanie

Przypadek definiujący sposób logowania wyborcy oraz urzędnika do systemu referendalnego.

Ciąg podstawowy:

1. Wyborca lub urzędnik wprowadza login oraz hasło do formularza.
2. Wyborca lub urzędnik zatwierdza dane wprowadzone do formularza poprzez kliknięcie przycisku "Zaloguj".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z danymi w bazie.
5. Stwierdzenie zgodności z modelem wzywa załadowanie danych.

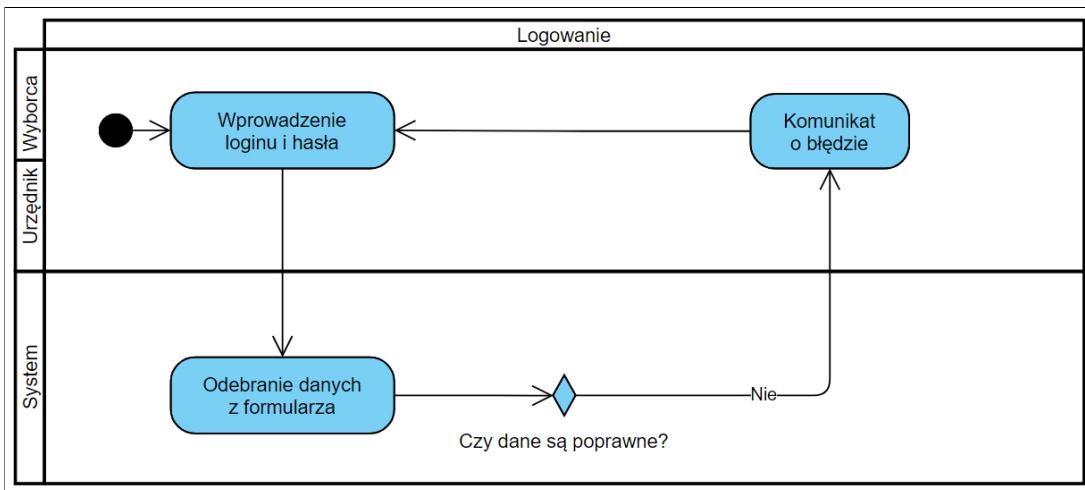
6. Wyświetlenie strony głównej wraz z numerem PESEL zalogowanego wyborcy lub urzędnika.



Rysunek 11: Ciąg podstawowy - logowanie

Ciąg alternatywny:

1. Wyborca lub urzędnik wprowadza login oraz hasło do formularza.
2. Wyborca lub urzędnik zatwierdza dane wprowadzone do formularza poprzez kliknięcie przycisku "Zaloguj".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z danymi w bazie.
5. Stwierdzenie niezgodności z modelem wyzwala komunikat o błędzie.
6. Powrót do punktu nr 1.



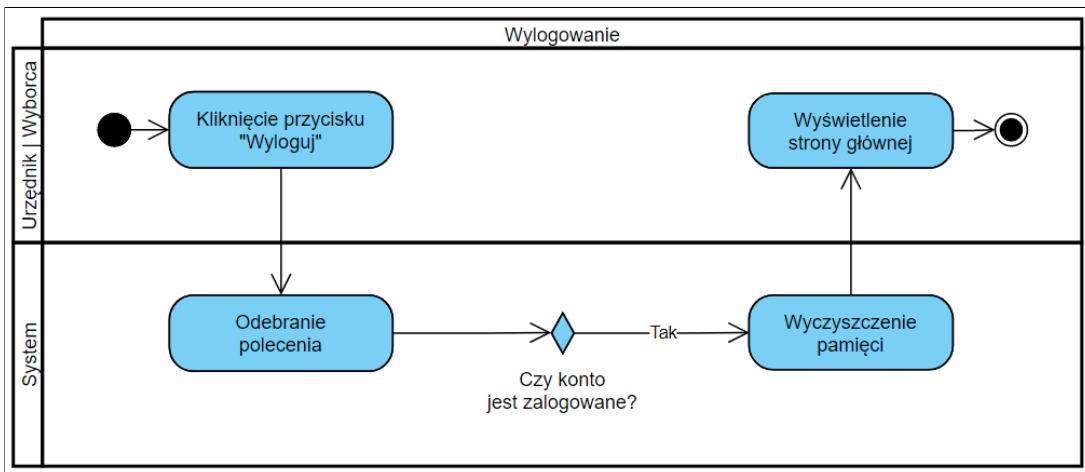
Rysunek 12: Ciąg alternatywny - logowanie

Wylogowanie

Przypadek definiujący sposób wylogowania wyborcy oraz urzędnika z systemu referendalnego.

Ciąg podstawowy:

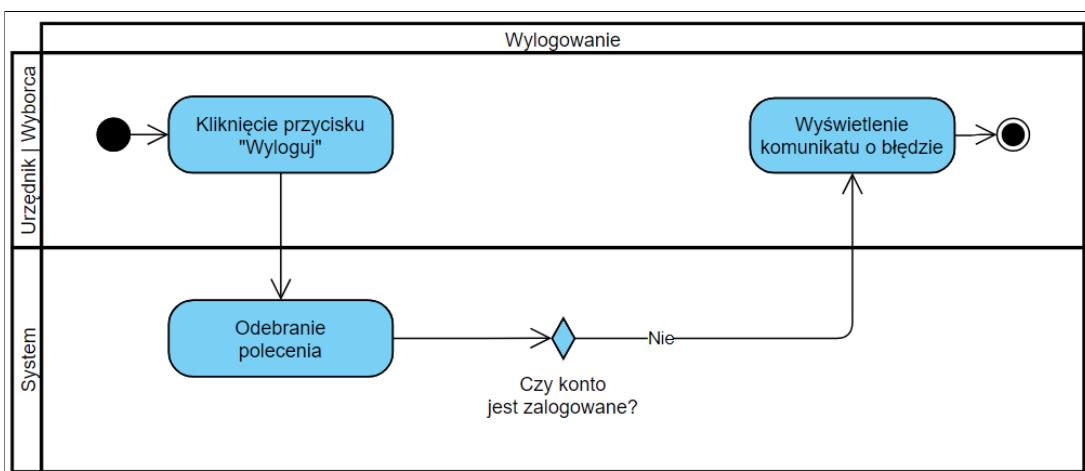
1. Wyborca lub urzędnik kliką przycisk "wyloguj" w menu strony.
2. System odbiera polecenie o wylogowaniu.
3. System przetwarza otrzymane informacje i sprawdza czy właściciel konta jest zalogowany.
4. Stwierdzenie zalogowania użytkownika lub wyborcy.
5. Wyczyszczenie pamięci podręcznej.
6. Wyświetlenie strony głównej.



Rysunek 13: Ciąg podstawowy - wylogowanie

Ciąg alternatywny:

1. Wyborca lub urzędnik kliknie przycisk "wyloguj" w menu strony.
2. System odbiera polecenie o wylogowaniu.
3. System przetwarza otrzymane informacje i sprawdza czy właściciel konta jest zalogowany.
4. Stwierdzenie niezalogowania wyborcy lub urzędnika wzywa komunikat o błędzie, co kończy przypadek użycia.



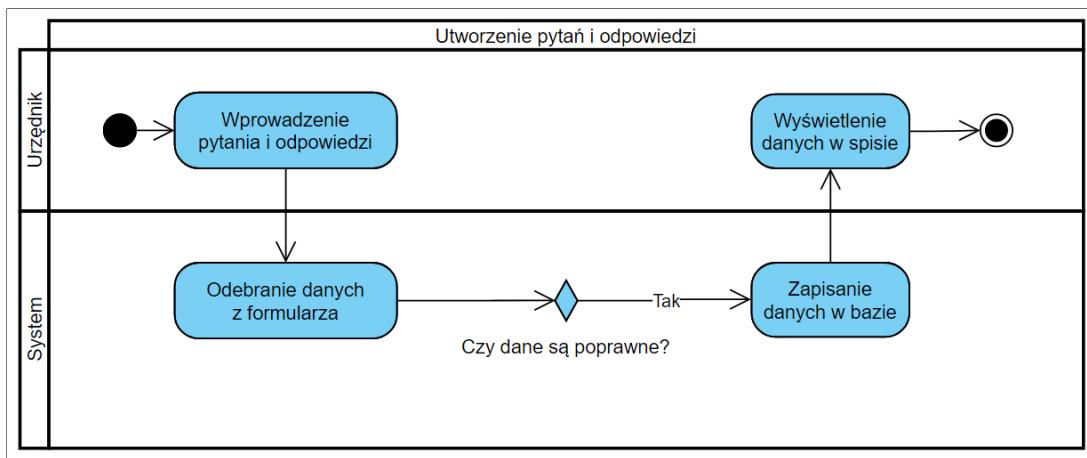
Rysunek 14: Ciąg alternatywny - wylogowanie

Utworzenie pytań i odpowiedzi

Przypadek definiujący sposób utworzenia zagadnień poprzez wprowadzenie pytań i odpowiedzi, które znajdą się na kartach wyborczych.

Ciąg podstawowy:

1. Urzędnik wprowadza do odpowiednich pól formularza treść pytania, odpowiedzi pozytywnej oraz odpowiedzi negatywnej.
2. Urzędnik zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Utwórz".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę pytania.
5. Stwierdzenie zgodności z modelem wzywa zapisanie danych w bazie przez system.
6. Użytkownik zostaje przekierowany do aktualnego spisu pytań.

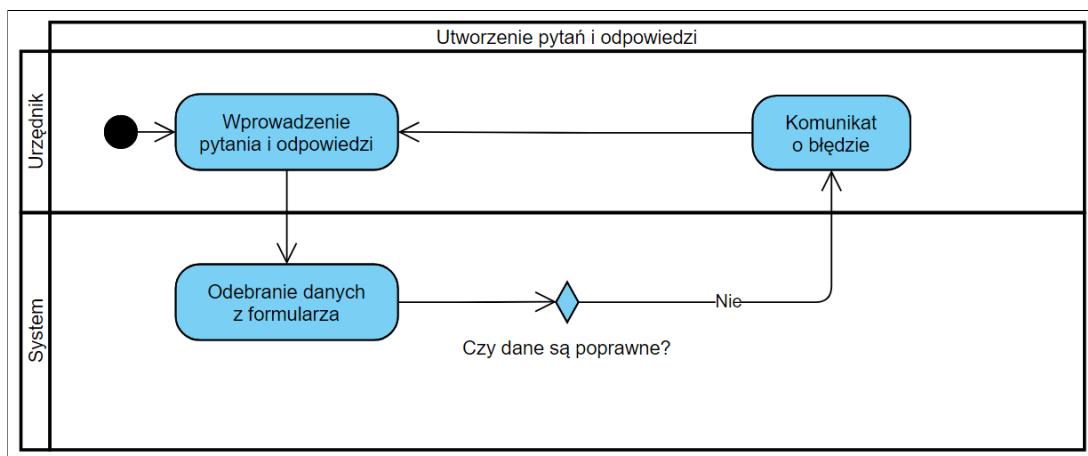


Rysunek 15: Ciąg podstawowy - utworzenie pytań i odpowiedzi

Ciąg alternatywny:

1. Urzędnik wprowadza do odpowiednich pól formularza treść pytania, odpowiedzi pozytywnej oraz odpowiedzi negatywnej.
2. Urzędnik zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Utwórz".

3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę pytania.
5. Stwierdzenie niezgodności z modelem powoduje wyświetlenie komunikatu o błędzie dla każdego pola formularza.
6. Powrót do punktu nr 1



Rysunek 16: Ciąg alternatywny - utworzenie pytań i odpowiedzi

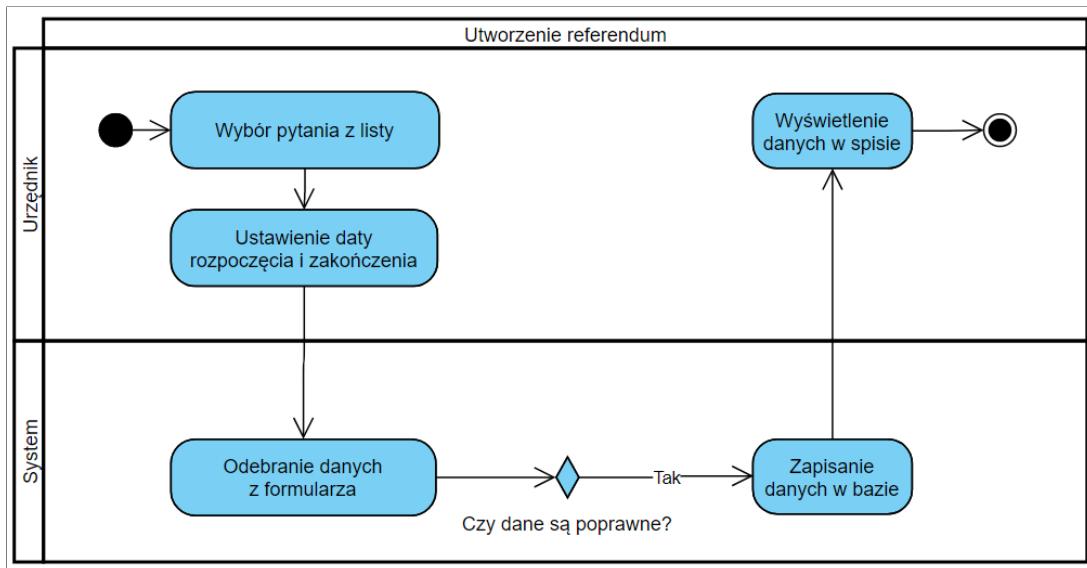
Utworzenie referendum

Przypadek definiujący sposób utworzenia referendum poprzez wprowadzenie terminu oraz wyboru pytania.

Ciąg podstawowy:

1. Urzędnik wybiera z listy pytanie, a następnie wpisuje datę rozpoczęcia oraz datę zakończenia do pól formularza.
2. Urzędnik zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Utwórz".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę referendum.

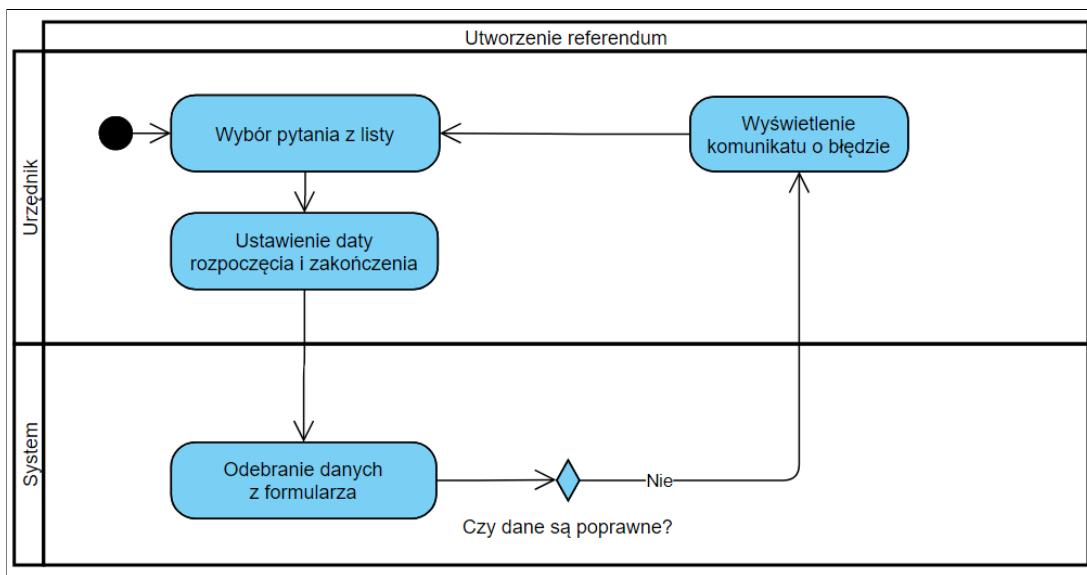
5. Stwierdzenie zgodności z modelem wyzwala zapisanie danych w bazie przez system.
6. Użytkownik zostaje przekierowany do aktualnego spisu referendów.



Rysunek 17: Ciąg podstawowy - utworzenie referendum

Ciąg alternatywny:

1. Urzędnik wybiera z listy pytanie, a następnie wpisuje datę rozpoczęcia oraz datę zakończenia do pól formularza.
2. Urzędnik zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Utwórz".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę pytania.
5. Stwierdzenie niezgodności z modelem powoduje wyświetlenie komunikatu o błędzie dla każdego pola formularza.
6. Powrót do punktu nr 1



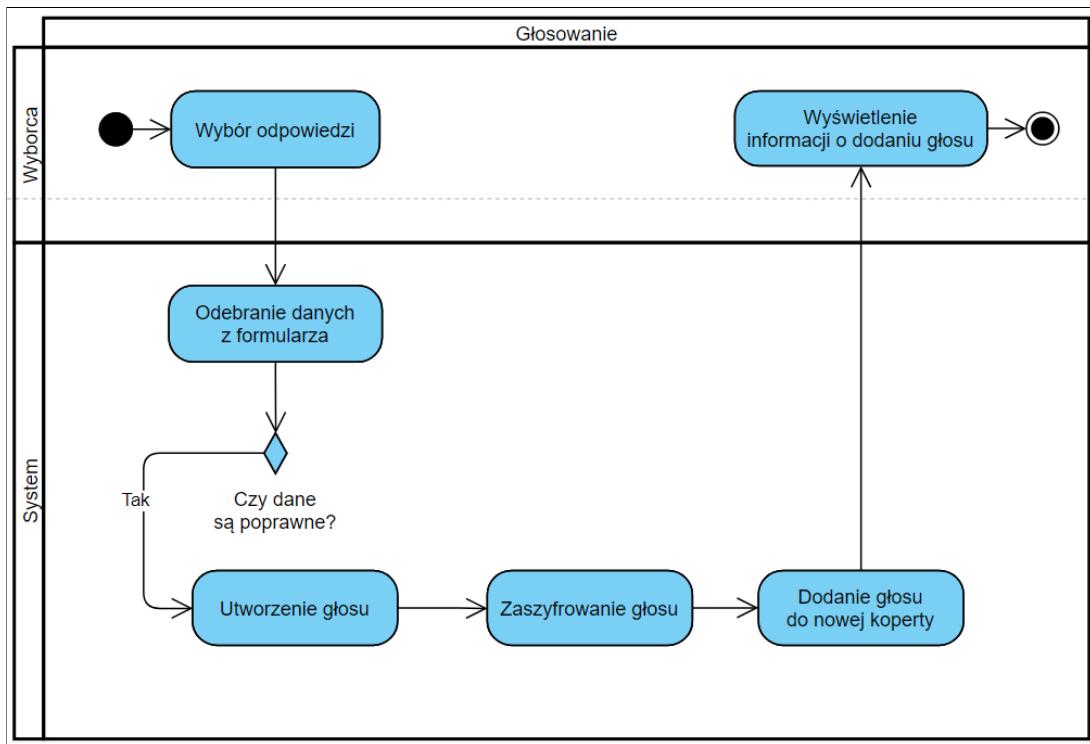
Rysunek 18: Ciąg alternatywny - utworzenie referendum

Głosowanie

Przypadek definiujący sposób głosowania przez wyborcę podczas trwającego referendum.

Ciąg podstawowy:

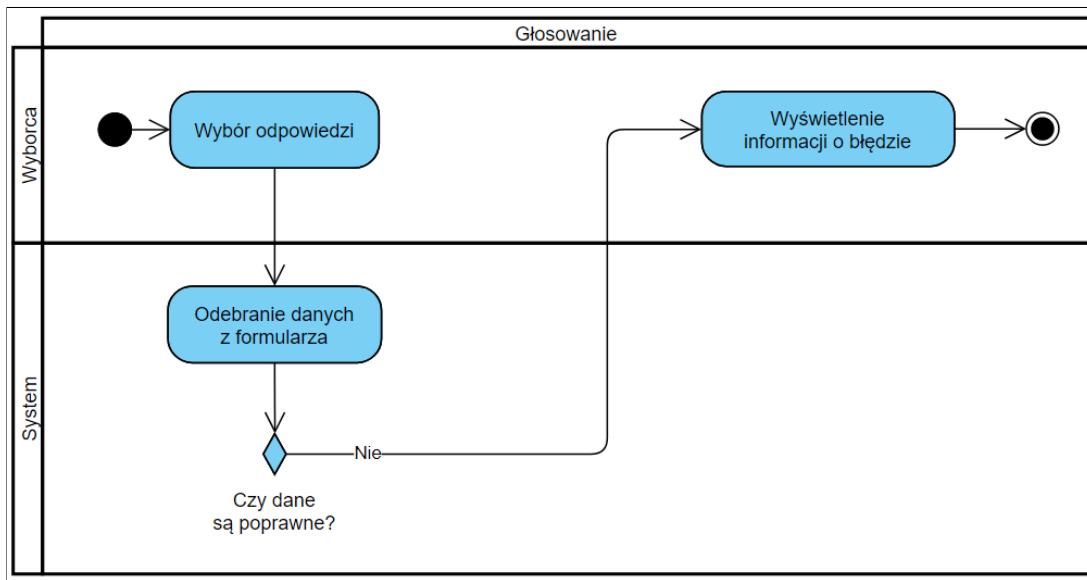
1. Wyborca odpowiada na pytanie referendalne poprzez zaznaczenie przycisku radiowego.
2. Wyborca zatwierdza podjętą decyzję poprzez kliknięcie przycisku "Zatwierdź".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę głosu.
5. Stwierdzenie zgodności z modelem wyzwala procedurę szyfrowania głosu przez uruchomienie skryptu w środowisku Python.
6. Zaszyfrowana odpowiedź oraz pytania są dodawane do nowej koperty, która otrzymuje status oznaczający powodzenie dodania głosu.
7. Wyborca otrzymuje komunikat o dodaniu głosu do koperty.



Rysunek 19: Ciąg podstawowy - głosowanie

Ciąg alternatywny:

1. Wyborca odpowiada na pytanie referendalne poprzez zaznaczenie przycisku radiowego.
2. Wyborca zatwierdza podjętą decyzję poprzez kliknięcie przycisku "Zatwierdź".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę wyniku.
5. Stwierdzenie niezgodności z modelem powoduje wyświetlenie komunikatu o błędzie.



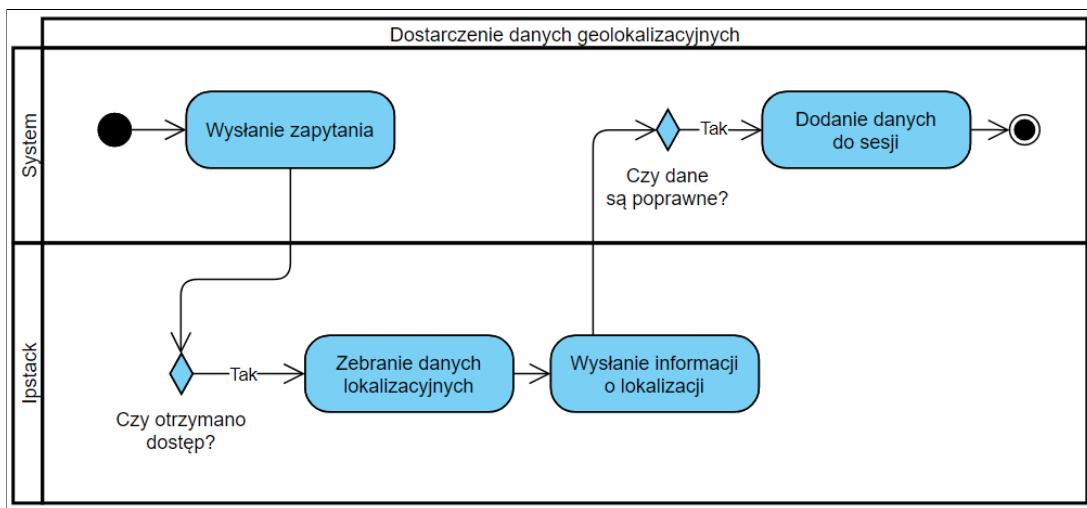
Rysunek 20: Ciąg alternatywny - głosowanie

Dostarczenie danych geolokalizacyjnych

Przypadek definiujący sposób dostarczenia danych na temat geolokalizacji zalogowanego wyborcy z zewnętrznego systemu Ipstack.

Ciąg podstawowy:

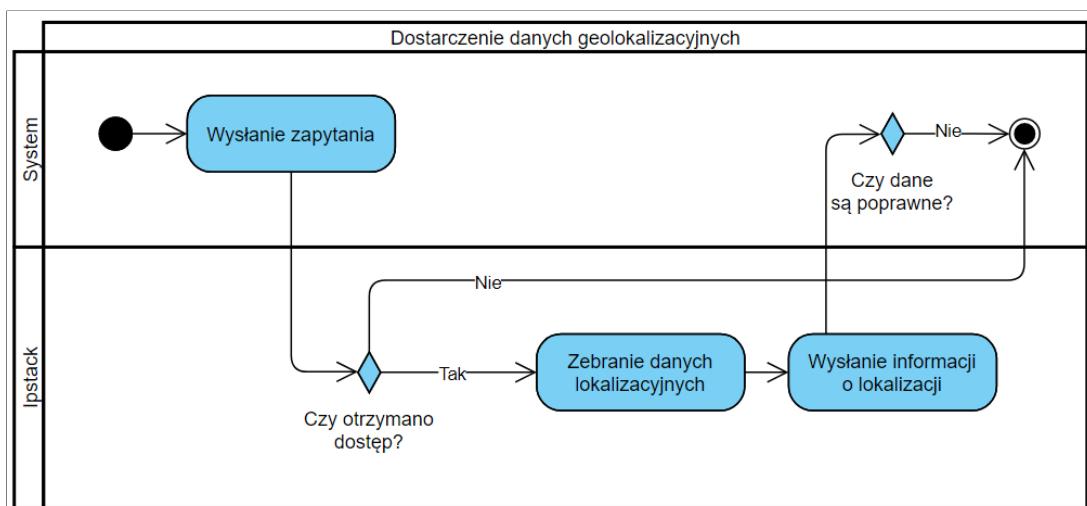
1. System wysyła zapytanie z adresem IP wyborcy oraz kluczem dostępu do Ipstack.
2. Ipstack akceptuje zapytanie.
3. Ipstack zbiera dane geolokalizacyjne na podstawie przesłanego adresu.
4. Ipstack wysyła dane na temat geolokalizacji do systemu.
5. Stwierdzenie kompletności otrzymanych danych przez system.
6. System dodaje dane do sesji wyborcy.



Rysunek 21: Ciąg podstawowy - dostarczenie danych geolokalizacyjnych

Ciąg alternatywny:

1. System wysyła zapytanie z adresem IP wyborcy oraz kluczem dostępu do Ipstack.
2. Ipstack akceptuje zapytanie, a w przeciwnym razie kończy przypadek użycia.
3. Ipstack zbiera dane geolokalizacyjne na podstawie przesłanego adresu.
4. IPstack wysyła dane na temat geolokalizacji do systemu.
5. Stwierdzenie niekompletności otrzymanych danych przez system kończy przypadek użycia.



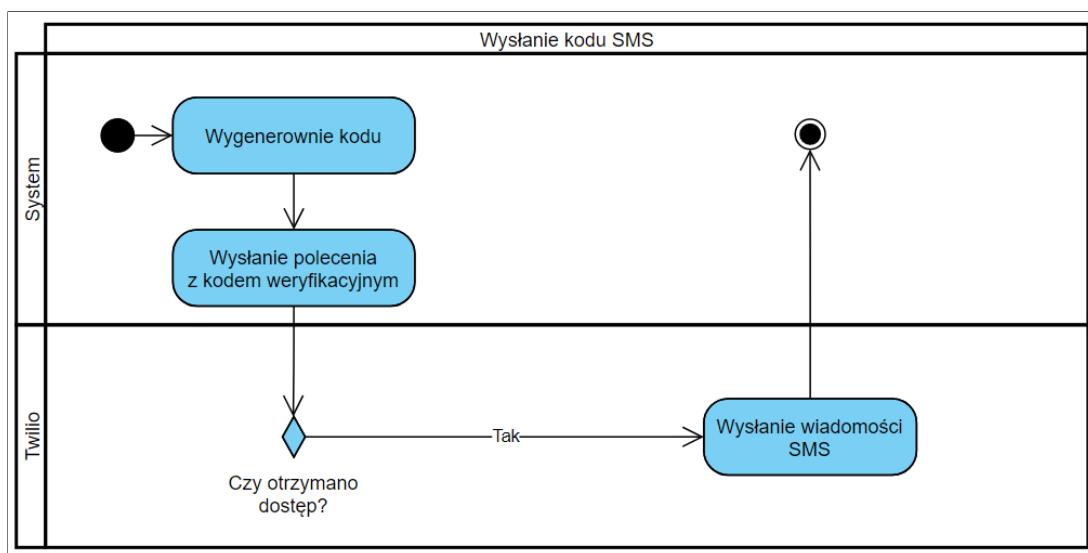
Rysunek 22: Ciąg alternatywny - dostarczenie danych geolokalizacyjnych

Wysłanie kodu SMS

Przypadek definiujący sposób wysłania kodu w wiadomości SMS na numer telefonu wyborcy przez zewnętrzny system Twilio.

Ciąg podstawowy:

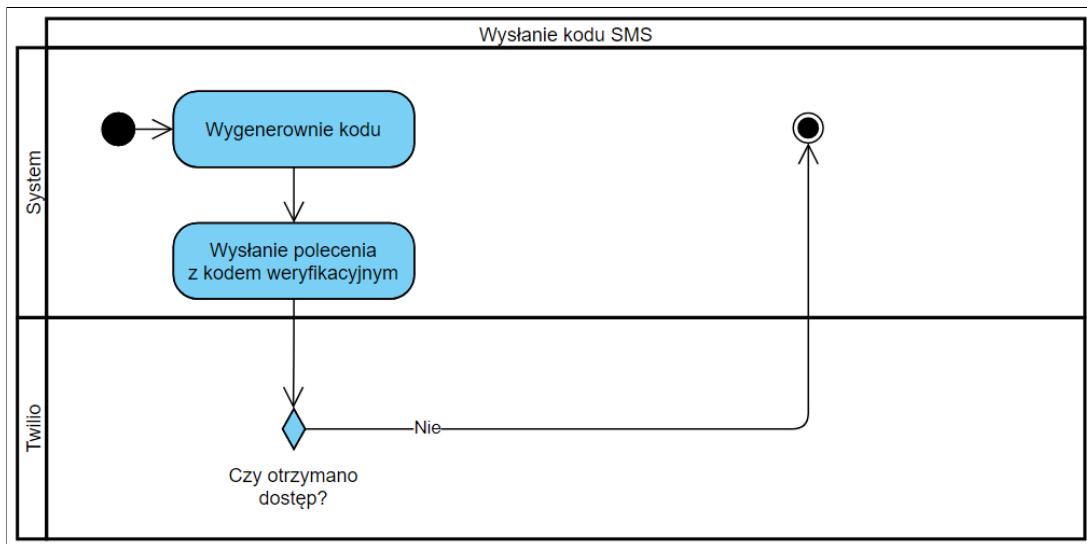
1. System generuje losowy kod liczbowy.
2. System wysyła polecenie nadania wiadomości SMS z kodem weryfikacyjnym, a ponadto załącza token, identyfikator Twilio oraz numer telefonu wyborcy.
3. Stwierdzenie poprawności tokena oraz identyfikatora Twilio.
4. Twilio wysyła wiadomość SMS na numer wyborcy wraz z kodem weryfikacyjnym.



Rysunek 23: Ciąg podstawowy - wysłanie kodu SMS

Ciąg alternatywny:

1. System generuje losowy kod liczbowy.
2. System wysyła polecenie nadania wiadomości SMS z kodem weryfikacyjnym, a ponadto załącza token, identyfikator Twilio oraz numer telefonu wyborcy.
3. Stwierdzenie odmowy dostępu przez Twilio i zakończenie przypadku użycia.



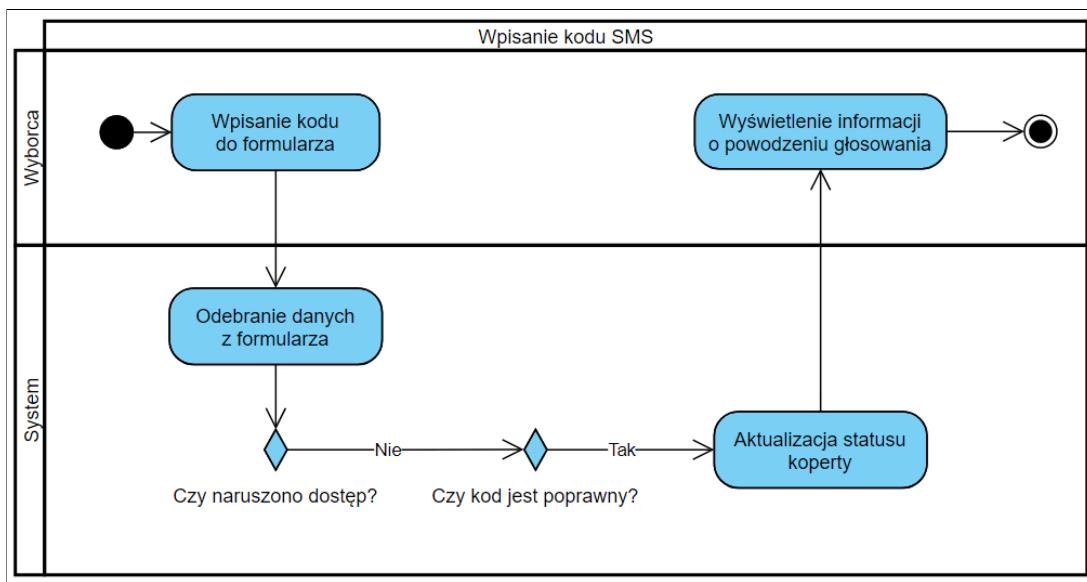
Rysunek 24: Ciąg alternatywny - wysłanie kodu SMS

Wpisanie kodu SMS

Przypadek definiujący sposób potwierdzenia tożsamości wyborcy poprzez wpisanie kodu otrzymanego na numer telefonu w wiadomości SMS.

Ciąg podstawowy:

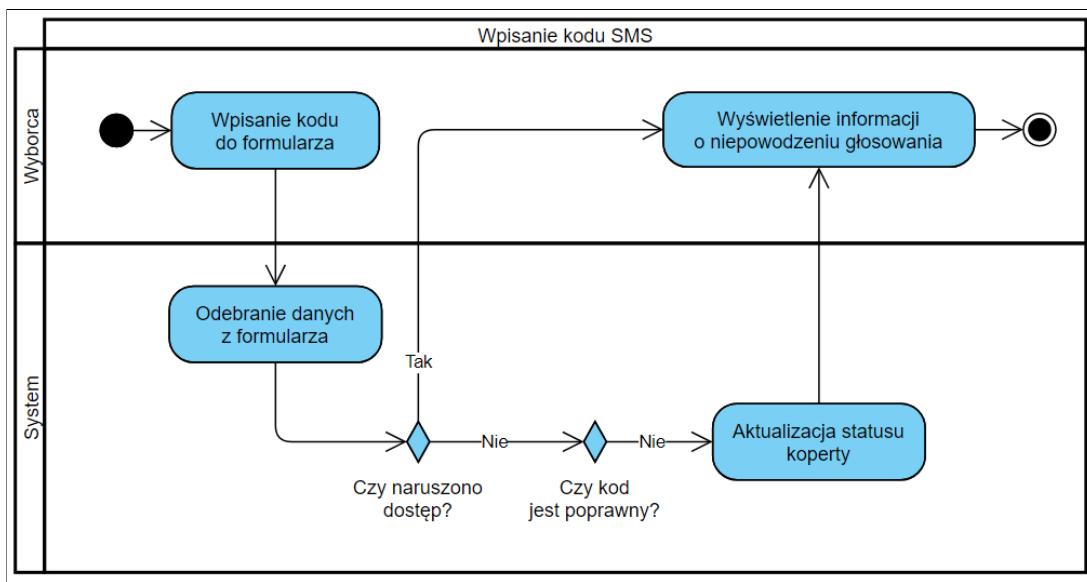
1. Wyborca wpisuje otrzymany kod w wiadomości SMS do formularza.
2. System odbiera dane otrzymane z formularza.
3. Stwierdzenie braku naruszenia dostępu do danych.
4. Stwierdzenie poprawności wpisanego do formularza kodu z wiadomości SMS.
5. System aktualizuje status koperty wyborcy i oznacza jako prawidłowo przeprowadzony proces wyborczy.
6. Wyświetlenie informacji o powodzeniu głosowania wraz z indywidualnym kodem potwierdzającym udział w wyborach.



Rysunek 25: Ciąg podstawowy - wpisanie kodu SMS

Ciąg alternatywny:

1. Wyborca wpisuje otrzymany kod w wiadomości SMS do formularza.
2. System odbiera dane otrzymane z formularza.
3. System stwierdza brak naruszenia dostępu do danych, a w przeciwnym razie wyświetla informację o niepowodzeniu głosowania i kończy przypadek użycia.
4. Stwierdzenie niepoprawności wpisanego do formularza kodu z wiadomości SMS.
5. System aktualizuje status koperty wyborcy i oznacza jako zawierająca błędny kod weryfikacyjny.
6. Wyświetlenie informacji o niepowodzeniu głosowania.



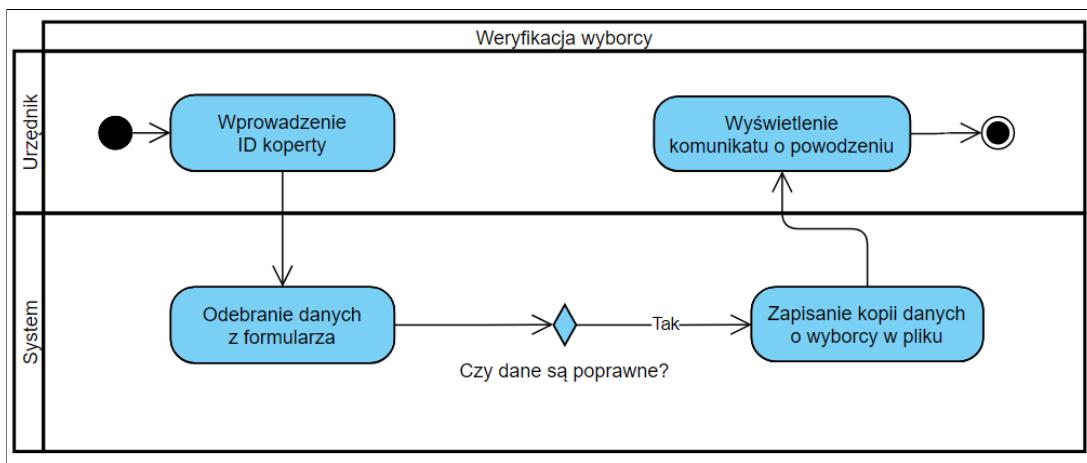
Rysunek 26: Ciąg alternatywny - wpisanie kodu SMS

Weryfikacja wyborcy

Przypadek definiujący sposób weryfikacji wyborcy, do którego należy koperta po- przez zapisanie wszystkich jego danych personalnych do pliku.

Ciąg podstawowy:

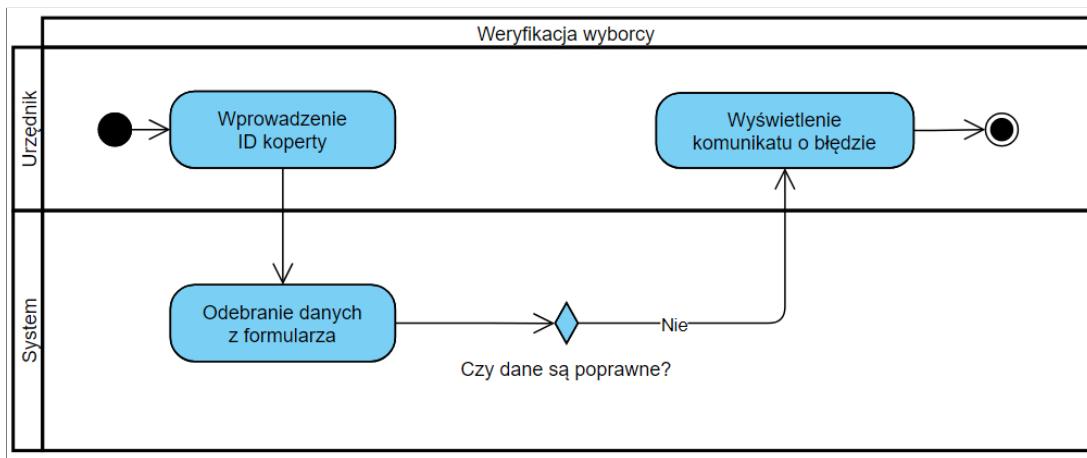
1. Urzędnik wprowadza numer identyfikacyjny koperty do formularza.
2. System odbiera dane otrzymane z formularza.
3. System stwierdza obecność koperty o podanym numerze identyfikacyjnym w bazie.
4. System zapisuje dane personalne wyborcy, do którego należy koperta w pliku XML.
5. Wyświetlenie informacji o powodzeniu weryfikacji wyborcy.



Rysunek 27: Ciąg podstawowy - weryfikacja wyborcy

Ciąg alternatywny:

1. Urzędnik wprowadza numer identyfikacyjny koperty do formularza.
2. System odbiera dane otrzymane z formularza.
3. System stwierdza brak koperty o podanym numerze identyfikacyjnym w bazie.
4. Wyświetlenie informacji o nieodnalezieniu koperty o podanym numerze identyfikacyjnym.



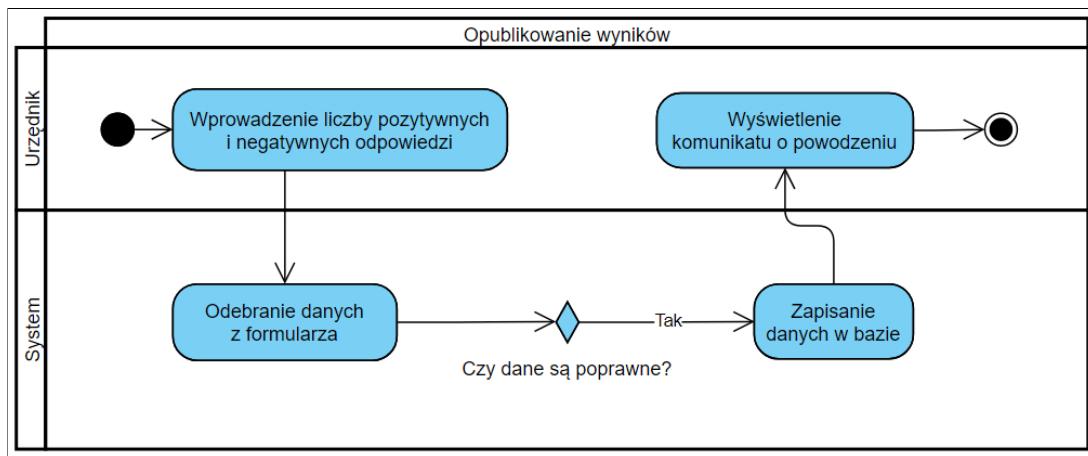
Rysunek 28: Ciąg alternatywny - weryfikacja wyborcy

Opublikowanie wyników

Przypadek definiujący sposób opublikowania wyników referendum, którego proces głosowania został już zakończony.

Ciąg podstawowy:

1. Urzędnik wprowadza liczbę pozytywnych oraz negatywnych odpowiedzi do pól formularza.
2. Urzędnik zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Utwórz".
3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę wyniku.
5. Stwierdzenie zgodności z modelem wzywa zapisanie danych w bazie przez system.
6. Wyświetlenie komunikatu o skutecznym opublikowaniu wyników.

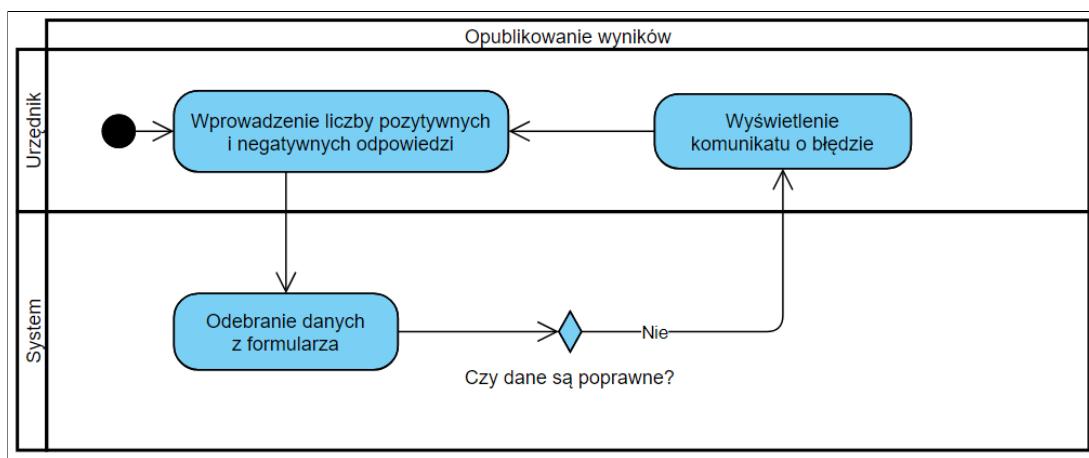


Rysunek 29: Ciąg podstawowy - opublikowanie wyników

Ciąg alternatywny:

1. Urzędnik wprowadza liczbę pozytywnych oraz negatywnych odpowiedzi do pól formularza.
2. Urzędnik zatwierdza dane dodane do formularza poprzez kliknięcie przycisku "Utwórz".

3. System odbiera dane z formularza.
4. System przetwarza otrzymane informacje i sprawdza zgodność z modelem danych określającym strukturę wyniku.
5. Stwierdzenie niezgodności z modelem powoduje wyświetlenie komunikatu o błędzie dla każdego wadliwego pola formularza.
6. Zakończenie przypadku publikowania wyników.



Rysunek 30: Ciąg alternatywny - opublikowanie wyników

4.3 Implementacja

System do przeprowadzania internetowego referendum został zaimplementowany w dwóch różnych środowiskach. Referendum-V jest aplikacją stworzoną w ASP.NET CORE MVC 5.0 z wykorzystaniem języka C#, natomiast Referendum-O została napisana w języku Python 3.8. Kontrolery logowania oraz rejestracji zostały wygenerowane z modeli szkieletowych dostępnych w Visual Studio.

4.3.1 Algorytm RSA

Kryptografia asymetryczna umożliwia utajnianie informacji z wykorzystaniem pary kluczy. Algorytm Rivesta-Shamira-Adlemana (RSA), mimo że został zaprojektowany w 1977 roku, wciąż zapewnia bezpieczeństwo. Zaimplementowane rozwiążanie wykorzystuje klucze o długości 4096 bitów, które trzeba wygenerować na podstawie bardzo dużych liczb pierwszych, różniące się długością o kilka cyfr. W tym celu należy przygotować zbiór liczbowy, który posłuży do generowania kluczy. Poniższy kod prezentuje proces wyznaczania liczby pierwszej o długości od 1984 do 2112 bitów. Utworzona pary zawsze tworzą klucz 4096-bitowy.

```
def pair_generator_of_prime_numbers(bits=2048):
    diff = bits // 32
    allBits = bits * 2
    pBits = bits + diff
    qBits = bits - diff
    p = prime_number(pBits, 3)
    q = prime_number(qBits, 3)

    switch = False
    while (p * q).bit_length() != allBits:
        if switch:
            p = prime_number(pBits, 3)
        else:
            q = prime_number(qBits, 3)

        switch = not switch

    return p, q
```

Kod 1: Generowanie par losowych liczb pierwszych o łącznej długości 4096 bitów

Funkcja *prime_number()* rozpoczyna proces poszukiwania liczb pierwszych z wykorzystaniem puli wątków. Uruchamiana jest funkcja *get_prime()*, która w przypadku odn-

leżenia liczby nieparzystej sprawdza jej pierwszość, a następnie zwraca wartość. Całość jest realizowana dzięki pakietowi multiprocessing [44]. Przepływ informacji między procesami jest jednokierunkowy.

```
def prime_number(bits, threads_number):
    (receiving_pipe, outgoing_pipe) = multiprocessing.Pipe(duplex=False)
    try:
        process = [multiprocessing.Process(target=get_prime, args=(bits, outgoing_pipe))
                   for _ in range(threads_number)]
        for p in process:
            p.start()

        prime_num = receiving_pipe.recv()
    finally:
        receiving_pipe.close()
        outgoing_pipe.close()

    for p in process:
        p.terminate()

    return prime_num
```

Kod 2: Tworzenie puli wątków do generowania liczb pierwszych

Ponadto funkcja *get_prime()* wykorzystuje do losowania liczb *getrandbits()*, która znajduje się w bibliotece random. Sprawdzenie, czy liczba jest nieparzysta, pozwala zaoszczędzić czas. Zakłada się, że parametr *bits* ≥ 2 .

```
def get_prime(bits, pipe: Connection):
    while True:
        n = random.getrandbits(bits - 1) * 2 + 1
        if test_miller_rabin(n, 2):
            pipe.send(n)
            return
```

Kod 3: Generowanie losowej liczby pierwszej

Wylosowana liczba jest poddawana testowi Millera-Rabina [45], który określa jej pierwszość. Jeżeli liczba jest pierwsza, to funkcja zwraca True, a w przeciwnym razie False. Prawdopodobieństwo błędu wynosi $(\frac{1}{4})^n$ przy n przebiegach i różnych podstawach a . Dla 56 iteracji prawdopodobieństwo pomyłki wynosi 1 do 4^{56} , natomiast dla 2 przebiegów jest to 6,25%.

```

def test_miller_rabin(p, n):
    if p == 2 or p == 3:
        return True
    if p % 2 == 0:
        return False
    d = p - 1
    s = 0
    while d % 2 == 0:
        s = s + 1
        d //= 2
    for i in range(n):
        a = random.randrange(2, p - 1) # [2;p-2]
        x = pow(a, d, p)
        if x == 1 or x == p - 1:
            continue
        j = 1
        while j < s and x != p - 1:
            x = pow(x, 2, p)
            if x == 1:
                return False
            j = j + 1
        if x != p - 1:
            return False
    return True

```

Kod 4: Test pierwszości Millera-Rabina

Kolejnym krokiem w implementacji algorytmu RSA jest wygenerowanie par kluczy. Poniższy kod prezentuje przepis na otrzymanie klucza publicznego (n, e) oraz prywatnego (d, n) przechowywanego w krotce. Rozwiążanie *generate_keys()* używa funkcji wbudowanej *pow()* do wyznaczenia parametru d .

```

def generate_keys():
    p, q = pair_generator_of_prime_numbers(2048)
    n = p * q
    phi = (p - 1) * (q - 1)
    e = chooseE(phi)
    d = pow(e, -1, phi)
    return (n, e), (d, n)

```

Kod 5: Generowanie kluczy

Losowy wybór wykładnika e jest realizowany poprzez funkcję *chooseE(phi)*. Poszukuje się takiego parametru e , dla którego największa wspólna wielokrotność e oraz ϕ odpowiada wartościowej 1.

```

def chooseE(phi):
    e = random.randrange(1, phi)
    g = math.gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = math.gcd(e, phi)
    return e

```

Kod 6: Wyznaczenie parametru e

Wygenerowane pary kluczy należy przechowywać w bezpiecznym miejscu, ponieważ naruszenie dostępu spowoduje utratę tajności głosowania. Autorski format pliku *.govkey* jest przystosowany do magazynowania danych służących zarówno do szyfrowania, jak i od-szyfrowania. Plik przygotowany za pomocą funkcji *save_govkey()* należy przechowywać na zewnętrznym dysku bez dostępu do Internetu aż do zakończenia etapu głosowania.

```

def save_govkey(public_key , private_key):
    f = open('storage.govkey', 'a')
    f.write("-----BEGIN KEY PAIR-----" + "\n")
    f.write(public_key)
    f.write("\n")
    f.write(private_key)
    f.write("\n")
    f.write("-----END KEY PAIR-----" + "\n")

```

Kod 7: Zapis kluczy do pliku govkey

Klucze publiczne umieszczane są w bazie danych, gdyż używa się ich do szyfrowania decyzji wyborców podczas etapu głosowania. Klucze są przechowywane jako obiekty klasy *DataProtectionKey* [46]. Indywidualny numer identyfikacyjny umożliwia między innymi odnalezienie poszukiwanego klucza, natomiast pola odpowiadające dacie określają czas stworzenia, aktywacji oraz przedawnienia. Przekazany parametr *key* do funkcji *add_key_to_db()* jest przetworzony przy użyciu kodowania transportowego Base64 [47], natomiast *d* odpowiada aktualnej dacie UTC [48] zapisanej na przykład jako 2021-01-10T23:58:52.517487Z.

```

def add_key_to_db(keyid, d, key):
    temp = ""
    temp += '<?xml version="1.0" encoding="utf-8"?>'
    temp += '<key id="' + str(keyid) + '" version="1">'
    temp += '<creationDate>' + str(d) + '</creationDate>'
    temp += '<activationDate></activationDate>'
    temp += '<expirationDate></expirationDate>'
    temp += '<encryptedKey>'

```

```

temp += str(key)
temp += '</encryptedKey></key>'
key_name = "key-" + str(keyid)
database.insertKey(key_name, temp)

```

Kod 8: Zapis klucza publicznego do bazy danych

Gdy klucze są już wygenerowane i odpowiednio zmagazynowane, można przystąpić do etapu szyfrowania. Program wykorzystuje do tego celu funkcję *encrypt()*, która przyjmuje ciąg bajtów jako parametr *msg* oraz klucz publiczny *public_key* będący krotką. Klucz publiczny przed przekazaniem do funkcji *encrypt()* został odkodowany z pomocą metody *base64_decode()*. Wartość zwróconą stanowi zaszyfrowany ciąg bajtów.

```

def encrypt(msg, public_key):
    n, e = public_key
    key_length = byte_size(n)
    b = basket(msg, key_length)
    x = bytes2int(b)
    encrypted = pow(x, e, n)
    return int2bytes(encrypted, key_length)

```

Kod 9: Szyfrowanie wiadomości

Obliczenie długości klucza jest realizowane poprzez funkcję *byte_size()*. Wartość zwracana odpowiada ilości bajtów potrzebnych do przechowania liczby przekazanej w parameterze.

```

def byte_size(num):
    if num == 0:
        return 1
    return ceil_div(num.bit_length(), 8)

```

Kod 10: Obliczenie ilości bajtów

Funkcja *ceil_div()* oblicza sufit [49] z ilorazu długości bitowej *num* oraz liczby reprezentującej ilość bitów w jednym bajcie. Do przeprowadzenia tej operacji wykorzystywana jest także wbudowana funkcja *divmod()* [50].

```

def ceil_div(num, div):
    q, mod = divmod(num, div)
    if mod:
        q += 1
    return q

```

Kod 11: Obliczenie sufitu z wyniku ilorazu dwóch liczb

Zanim wiadomość zostanie zaszyfrowana, musi zostać odpowiednio zmodyfikowana i wypełniona. Funkcja *basket()* przetwarza ciąg bajtów i dodaje losowe dane. Wiadomość przekazana jako parametr jest dołączana na koniec sekwencji. Uzupełnienie zwiększa również trudność w odszyfrowaniu z uwagi na losowość zapewnianą przez funkcję wbudowaną *urandom()*, która zwraca bajty ze specyficznego dla systemu operacyjnego źródła. Opracowane dane powinny być wystarczająco nieprzewidywalne dla aplikacji kryptograficznej.

```
def basket(msg, target_length):
    data = b''
    data_length = target_length - len(msg) - 3
    while len(data) < data_length:
        n = data_length - len(data)
        temp_data = os.urandom(n + 5)
        temp_data = temp_data.replace(b'\x00', b'')
        data = data + temp_data[:n]
    return b''.join([b'\x00\x02', data, b'\x00', msg])
```

Kod 12: Wypełnienie wiadomości

Zanim wiadomość zostanie przekazana do zaszyfrowania, konieczne jest przekształcenie jej do postaci liczby całkowitej. W tym celu opracowano *bytes2int()*. Funkcja *int.from_bytes()* przyjmuje jako parametr ciąg bajtów, informację o tym, że najbardziej znaczący bajt znajduje się na początku oraz wskazanie, że liczba jest bez znaku.

```
def bytes2int(b):
    return int.from_bytes(b, 'big', signed=False)
```

Kod 13: Konwersja bajtów na liczbę całkowitą

Funkcja wbudowana *pow(x, e, n)* [51] zwraca wynik działania $x^e \pmod{n}$ będący liczbową reprezentacją zaszyfrowanego głosu. Następnym etapem szyfrowania jest przekonwertowanie ciągu liczbowego do jego bajtowej reprezentacji. Funkcja *int2bytes()* przekształca liczbę zgodnie z Big-Endian [52], czyli najbardziej znaczący bajt umieszczany jest jako pierwszy. Do konwersji wykorzystywana jest także wbudowana funkcja Pythona *to_bytes()* oraz inne funkcje matematyczne.

```
def int2bytes(num, size=0):
    n_bytes = max(1, math.ceil(num.bit_length() / 8))
    if size > 0:
        return num.to_bytes(size, 'big')
    return num.to_bytes(n_bytes, 'big')
```

Kod 14: Konwersja liczby całkowitej na bajty

Zakodowana decyzja wyborcy jest umieszczana w bazie danych, a po zakończeniu głosowania jest odszyfrowywana za pomocą klucza prywatnego. Przed zmagazynowaniem informacja jest poddawana kodowaniu transportowemu Base64 [47]. Utajnione głosy oczekują na moment wyjęcia z kopert, a docelowo ustalenia wyników. Odszyfrowanie jest realizowane za pomocą funkcji *decrypt()*. Przyjmuje ona jako parametr zaszyfrowaną wiadomość *msg* w postaci ciągu bajtów oraz klucz prywatny, który przed przekazaniem do funkcji *decrypt()* został odkodowany z pomocą metody *base64_decode()*. Wbudowana funkcja *find()* umożliwia znalezienie separatora między wiadomością a bajtowym wypełnieniem.

```
def decrypt(msg, private_key):
    d,n = private_key
    length = byte_size(n)
    encrypted = bytes2int(msg)
    decrypted = pow(encrypted, d, n)
    vote = int2bytes(decrypted, length)
    sep_id = vote.find(b'\x00', 2)
    return vote[sep_id + 1:]
```

Kod 15: Odszyfrowanie wiadomości

4.3.2 Moduł SMS

Usługę wysyłania wiadomości SMS z kodem weryfikacyjnym umożliwia zewnętrzny serwis Twilio [53]. Komunikacja jest możliwa dzięki API, które obsługuje zapytania wysypane z aplikacji referendalnej. Usługodawca umożliwia utworzenie specjalnego numeru telefonu oraz udostępnia opcję blokowania wiadomości zwrotnych. Kontroler SMS został umieszczony w aplikacji Referendum-V i jest odpowiedzialny za dostarczenie kodu do wyborcy w celu potwierdzenia jego tożsamości. Dostawca daje możliwość zaimplementowania usługi w dowolnym środowisku, dlatego jest to rozwiązanie intratne także dla innych technologii. Do prawidłowego działania wymagane jest podanie danych dostępowych w postaci *accountSid* oraz *authToken*, które można uzyskać po zalogowaniu na konto Twilio. Usługa jest komercyjna, dlatego przed wdrożeniem należy wykupić pakiet, a także numer telefonu. Serwis prowadzi swoją działalność na całym świecie. Metoda *SendSms()* przyjmuje jako parametr numer telefonu poprzedzony telefonicznym kodem kraju oraz kod weryfikacyjny dla wyborcy.

```

using Twilio;
using Twilio.AspNet.Core;
using Twilio.Rest.Api.V2010.Account;
using Twilio.Types;

namespace Sms.Controllers
{
    public class SmsController : TwilioController
    {
        public string SendSms(string phoneNumber, string code)
        {
            var accountSid = "ACxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
            var authToken = "59xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx";
            TwilioClient.Init(accountSid, authToken);
            var to = new PhoneNumber(phoneNumber);
            var from = new PhoneNumber("+12xxxxxxxx");
            var message = MessageResource.Create(
                to: to,
                from: from,
                body: "Twój kod weryfikacyjny to: " + code);
            return message.Sid;
        }
    }
}

```

Kod 16: Kontroler SMS

4.3.3 Geolokalizator

Niezwykle istotne z punktu widzenia organizatora jest ustalenie lokalizacji wyborcy. Pewne grupy użytkowników mogą próbować zakłócić proces referendalny, dlatego należy dołożyć wszelkich starać, aby nie dopuścić do naruszenia dostępu. Zewnętrzny serwis Ipstack [54] dostarcza informacji o wyborcy na podstawie adresu IP.

```

using Microsoft.AspNetCore.Mvc;
using WebApplication.Models;
using Newtonsoft.Json.Linq;
using System.IO;
using System.Net;

namespace WebApplication.Controllers
{
    public class IPController : Controller
    {
        public string GetCurrentIP()

```

```

    {
        var ip = Request.HttpContext.Connection.RemoteIpAddress;
        return ip.ToString();
    }
    public Geolocation GetGeolocation()
    {
        string IP = GetCurrentIP();
        if (IP != null)
        {
            Geolocation geolocation = new Geolocation();
            string url = "http://api.ipstack.com/" + IP + "?access_key=0xxxxxxxxd57";
            var request = System.Net.WebRequest.Create(url);
            using (WebResponse wrs = request.GetResponse())
            using (Stream stream = wrs.GetResponseStream())
            using (StreamReader reader = new StreamReader(stream))
            {
                string json = reader.ReadToEnd();
                var obj = JObject.Parse(json);
                geolocation.ID = (string)obj["ip"];
                geolocation.type = (string)obj["type"];
                geolocation.continent_code = (string)obj["continent_code"];
                geolocation.continent_name = (string)obj["continent_name"];
                geolocation.country_code = (string)obj["country_code"];
                geolocation.country_name = (string)obj["country_name"];
                geolocation.region_code = (string)obj["region_code"];
                geolocation.city = (string)obj["city"];
                geolocation.latitude = (double)obj["latitude"];
                geolocation.longitude = (double)obj["longitude"];
                geolocation.geoname_id = (string)obj["location"]["geoname_id"];
                return geolocation;
            }
        }
        return null;
    }
}

```

Kod 17: Kontroler lokalizacji

Metoda *GetCurrentIP()* zwraca adres IP wyborcy, natomiast *GetGeolocation()* wysyła za-pytnie oraz odbiera odpowiedź od dostawcy usługi. Serwis w wiadomości zwrotnej nadaje informację w formacie JSON. Dostęp do platformy jest darmowy, jednakże występują limity zapytań oraz złożoność odpowiedzi. Wybierając płatny pakiet, otrzymujemy także dane, czy wyborca korzysta z proxy [55].

4.3.4 Głosowanie

Wyborcy podczas trwającego referendum muszą poprawnie oddać głos. Jednym z ważniejszych kontrolerów czuwających nad przebiegiem akcji jest *VotingController*. Funkcja odpowiedzialna za obsługę wyboru odpowiedzi to *Select()*. Stworzony za pomocą formularza głos jest odbierany, a następnie szyfrowany. Na początku tworzona jest nowa koperta za pomocą *CreateEnvelope()*. Kolejną ważną czynnością jest znalezienie wolnego klucza publicznego. Przeszukiwanie jest realizowane za pośrednictwem *SearchFreeKey()*. Znaleziony klucz jest następnie aktywowany oraz określa się jego ważność. Wyselekcjonowaną wartość klucza z formatu XML przekazuje się do funkcji uruchamiającej skrypt Pythona, który zajmuje się zaszyfrowaniem wiadomości. Następne czynności polegają na umieszczeniu danych w bazie, a także wygenerowaniu losowego kodu SMS. Przygotowana wiadomość jest wysyłana za pośrednictwem funkcji *SendSms()* z kontrolera *SmsController()*.

```
[Authorize]
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Select([Bind("Id,Question,Answer")] Vote vote)
{
    Envelope envelope = CreateEnvelope();
    Vote obj = new Vote();
    string key = null;

    var row = await SearchFreeKey();
    var dt = DateTime.UtcNow;
    row.Xml = SetXMLValue(row.Xml, "key/activationDate", dt.ToString("yyyy-MM-ddTHH:mm:ss.
    ffffffZ"));
    dt.AddDays(30);
    row.Xml = SetXMLValue(row.Xml, "key/expirationDate", dt.ToString("yyyy-MM-ddTHH:mm:ss.
    ffffffZ"));
    _contextKeys.DataProtectionKeys.Update(row);
    await _contextKeys.SaveChangesAsync();

    key = GetXMLValue(row.Xml, "key/encryptedKey");

    obj.Question = vote.Question;
    obj.Answer = PythonScript(vote.Answer, key);
    obj.Id = row.FriendlyName.Substring(KeyIdStartIndex);
    _context.Add(obj);

    envelope.VoteId = obj.Id;
    envelope.Timestamp = DateTime.Now;
```

```

VerifyPhoneNumber verifyPhoneNumber = new VerifyPhoneNumber();
verifyPhoneNumber.PhoneNumber = getUserPhoneNumber();
verifyPhoneNumber.EnvelopeID = envelope.Id.ToString();
verifyPhoneNumber.Code = new Random().Next(1000, 100000).ToString();
verifyPhoneNumber.DateTimeSent = DateTime.UtcNow.ToString();

_contextKeys.Add(verifyPhoneNumber);
await _contextKeys.SaveChangesAsync();

new SmsController().SendSms(verifyPhoneNumber.PhoneNumber, verifyPhoneNumber.Code);
_context.Add(envelope);
await _context.SaveChangesAsync();

verifyPhoneNumber = new VerifyPhoneNumber();
verifyPhoneNumber.PhoneNumber = getUserPhoneNumber();

return View("Verify", verifyPhoneNumber);
}

```

Kod 18: Funkcja do obsługi karty wyborczej

Kolejnym ważnym etapem po udanym wyborze odpowiedzi oraz wprowadzeniu kodu weryfikacyjnego jest sprawdzenie poprawności danych dostarczonych przez głosującego. W tym celu przeszukuje się bazę, aby odnaleźć kod dla podanego numeru. Jeżeli kod z bazy różni się od tego wprowadzonego przez użytkownika, wówczas koperta otrzymuje status B. Oznacza to, że proces weryfikacji przeszedł nieprawidłowo, dlatego głos nie zostanie wliczony do wyniku. Dokonuje się także sprawdzenia, czy nie naruszono dostępu do danych. Należy pamiętać, iż próba zweryfikowania numeru innej osoby również wpłynie na nieważność głosu, ponieważ zostanie przypisany status C. Jeżeli proces potwierdzenia tożsamości zakończy się sukcesem, wówczas koperta uzyska status a i będzie wliczona do puli ważnych głosów.

```

[Authorize]
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Verify([Bind("Code,PhoneNumber")] VerifyPhoneNumber
    verifyPhoneNumber)
{
    var envelope = new Envelope();
    VerifyPhoneNumber verifyPhoneNumberDB = new VerifyPhoneNumber();
    verifyPhoneNumberDB = await _contextKeys.verifyPhoneNumbers.FirstOrDefaultAsync(n => n.
        PhoneNumber == verifyPhoneNumber.PhoneNumber);
    if (verifyPhoneNumberDB != null)
    {
        var Id = verifyPhoneNumberDB.EnvelopeID;
        envelope = await _context.Envelopes.FindAsync(Id);
    }
}

```

```

        if (envelope != null)
        {
            var currentUserId = this.User.FindFirstValue(ClaimTypes.NameIdentifier);
            if (envelope.WebApplicationUserId != currentUserId)
            {
                envelope.Status = Status.C;
                _context.Update(envelope);
                await _context.SaveChangesAsync();
                return View("Fail");
            }
            if (verifyPhoneNumber.Code != verifyPhoneNumberDB.Code)
            {
                envelope.Status = Status.B;
                _context.Update(envelope);
                await _context.SaveChangesAsync();
                return View("Fail");
            }
            envelope.Status = Status.A;
            Result result = new Result();
            result.VerificationKey = envelope.VoteId;
            result.Message = message;
            _context.Update(envelope);
            await _context.SaveChangesAsync();
            return View("Success", result);
        }
        else
        {
            return View("Fail");
        }
    }
    return View("Fail");
}

```

Kod 19: Funkcja weryfikująca

4.4 Testy aplikacji

Zaprojektowana platforma do przeprowadzania internetowych referendum została odpowiednio przetestowana. Każdy dostępny moduł poddano sprawdzeniu i uzyskał pozytywną ocenę. W celu weryfikacji poprawność działania platformy rozpisano referendum, w którym obywatele mogli wyrazić stosunek wobec obecności Rzeczypospolitej Polskiej w Unii Europejskiej. Zadano pytanie "Czy wyraża Pani / Pan zgodę na przystąpienie Rzeczypospolitej Polskiej do Unii Europejskiej?" oraz umożliwiono udzielenie odpowiedzi poprzez zaznaczenie "TAK"

lub "NIE". W głosowaniu wzięło udział 12 osób, z czego dwa głosy określono jako nieważne. Protokół głosowania został wygenerowany jako plik XML, który należy opublikować, aby wyborcy mogli zweryfikować, czy ich głos został wliczony do puli poprawnych.

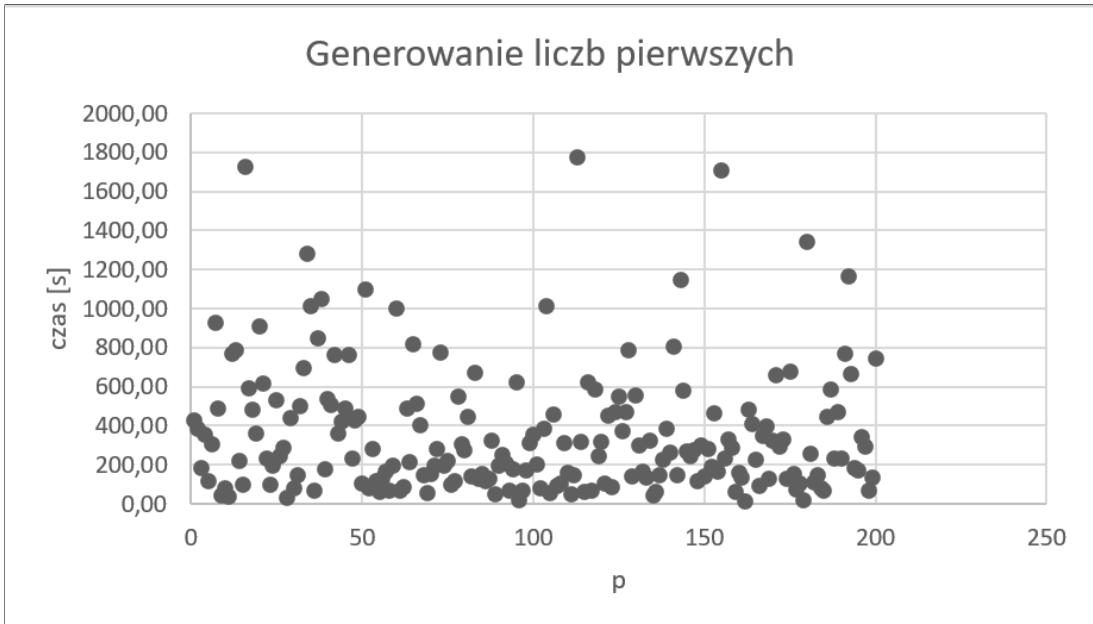
```
<results>
  <result id="0" version="1">
    <date>2021-01-31T21:29:08.422604Z</date>
    <question>Czy wyraża Pani / Pan zgodę na przystąpienie Rzeczypospolitej Polskiej do Unii Europejskiej?</question>
    <positiveAnswers>8</positiveAnswers>
    <negativeAnswers>2</negativeAnswers>
    <correctVotes>
      <vote id="0">
        <id>d512a4d4-1b61-4d52-896c-12b1ceaa3ffe</id>
      </vote>
      <vote id="1">
        <id>a5079a06-ec59-4806-8e18-fef294028280</id>
      </vote>
      <vote id="2">
        <id>cabff00c6-0bb7-492b-b439-eb619600fe71</id>
      </vote>
      <vote id="3">
        <id>eda64cc8-8c45-4b6b-8dab-6ec430a7790f</id>
      </vote>
      <vote id="4">
        <id>32565f55-ebba-457f-8aac-d5213e581c99</id>
      </vote>
      <vote id="5">
        <id>585f826f-8c7d-47fa-bf2f-fb6c58ca8b5d</id>
      </vote>
      <vote id="6">
        <id>c47d1250-dc94-4ce2-b873-9d2aee7a65c2</id>
      </vote>
      <vote id="7">
        <id>dd885e1d-03fc-42c0-b4dd-7961dd57146b</id>
      </vote>
      <vote id="8">
        <id>42c140b6-2e0b-42d6-8a4e-870e31b9e1c2</id>
      </vote>
      <vote id="9">
        <id>297a8f1b-e364-4fe5-afec-716ddf5d4bc1</id>
      </vote>
    </correctVotes>
    <incorrectVotes>
      <vote id="0">
        <id>66d89e87-3208-4187-8506-68af20b071ab</id>
      </vote>
      <vote id="1">
        <id>5f5eb30d-a202-4ea5-8dea-97b4f738207c</id>
      </vote>
    </incorrectVotes>
  </result>
</results>
```

Rysunek 31: Wyniki głosowania

Zapewnienie anonimowości wyborcom wymagało zaimplementowania algorytmu RSA, który wykorzystywał liczby pierwsze o długości od 1984 do 2112 bitów. Proces poszukiwania takich liczb mógł trwać nawet kilka minut, dlatego niezbędne było przygotowanie zbioru przed rozpoczęciem głosowania. Wygenerowano 200 par liczb pierwszych, a następnie zamieszczono w pliku tekstowym, rozdzielaając je znakiem nowej linii. Liczby w parze zostały odseparowane znakiem ";".

Rysunek 32: Wygenerowane liczby pierwsze

Proces generowania odpowiednich par liczb pierwszych trwał 71051 sekund. Generowanie jednej pary zajmowało średnio 355,25 sekund. Wykorzystany sprzęt posiadał procesor Intel® Core™ i5-7200U 2,5GHz oraz 8GB pamięci RAM DDR4. System zainstalowany na urządzeniu to 64-bitowy Windows 10 Home w wersji 10.0.18363. Komputer mógł równolegle wykonywać maksymalnie 4 procesy. Poniżej zaprezentowano wykres prezentujący czas wyszukiwania kolejnych par liczb pierwszych.



Rysunek 33: Generowanie liczb pierwszych - czas wyszukiwania kolejnych par

Przygotowany zbiór posłużył do stworzenia kluczy. Utworzono specjalny format do przechowywania, który otrzymał nazwę "govkey". W pliku zawarto klucze publiczne oraz prywatne zgrupowane w pary. w celu zapewnienia bezpieczeństwa podczas trwania etapu głosowania był przechowywany na urządzeniu zewnętrznym. Strukturę pliku zaprezentowano poniżej.

```
-----BEGIN KEY PAIR-----
PUBLIC_KEY
PRIVATE_KEY
-----END KEY PAIR-----
-----BEGIN KEY PAIR-----
PUBLIC_KEYn
PRIVATE_KEYn
-----END KEY PAIR-----
```

Kod 20: Para kluczy w pliku govkey

Przeprowadzone testy platformy do głosowania objęły funkcję rejestracji. Utworzone konto zostało użyte w referendum, a także w procesie wyborczym. Do formularza wprowadzono dane osobowe oraz hasło.

The screenshot shows a registration form titled "Rejestracja" (Registration). It includes fields for personal information: Name (Imię), Middle name (Drugie imię), Surname (Nazwisko), Date of birth (Data urodzenia), National ID card number (Numer dowodu osobistego), PESEL number (Numer PESEL), Email, and Password (Hasło). A note states that registration via other services is not possible due to technical reasons, mentioning the "Profil Zaufany". A "Zarejestruj" (Register) button is at the bottom. The footer indicates the platform is used for the referendum.

Referendum Głosowanie Wyniki Rejestracja Logowanie

Rejestracja

Utwórz nowe konto

Imię
Mateusz

Drugie imię
Grzegorz

Nazwisko
Domałązek

Data urodzenia
07.03.1998

Numer dowodu osobistego
[REDACTED]

Numer PESEL
[REDACTED]

Email
mateuszdomalazek@gmail.com

Hasło

Powtórz hasło

Zarejestruj

Referendum © 2021 - Mateusz Domałązek

Rysunek 34: Rejestracja konta wyborcy

Kolejnym etapem było odnalezienie aktywnego referendum, aby rozpocząć proces decyzyjny. Wyborcy mogą bowiem głosować tylko w obecnie trwających wydarzeniach. Nie istnieje możliwość oddania głosu w sprawie, która została już rozstrzygnięta, a jej wyniki opublikowane.

The screenshot shows a voting page for an active referendum. It features a "Głosowanie już trwa!" (Voting is already underway!) message and instructions for users. A "Oddaj głos" (Cast your vote) button is present. The top right corner shows the user's PESEL number and a log-out link.

Referendum Głosowanie Wyniki PESEL 980307... Wyloguj

Referendum

Głosowanie już trwa!

Zapraszamy do wzięcia udziału w internetowym referendum. Każdy obywatel ma prawo zagłosować tylko raz po uprzednim zarejestrowaniu i zalogowaniu. Aktywny telefon umożliwia weryfikację, dlatego jest niezbędny, gdyż wystąpi konieczność wprowadzenia kodu, który zostanie wysłany za pomocą SMS. Podczas korzystania z platformy należy znajdować się na terytorium Rzeczypospolitej Polskiej.

Oddaj głos

Rysunek 35: Aktywne referendum

Wygenerowana karta wyborcza posiada pytanie, a także możliwe do zaznaczenia odpowiedzi. Użytkownik wybiera jedną opcję, a następnie zatwierdza swoją decyzję za pomocą przycisku "Zatwierdź".

The screenshot shows a user interface for a referendum. At the top, there are navigation links: 'Referendum', 'Głosowanie', and 'Wyniki'. On the right, it displays 'PESEL 980307' and a 'Wyloguj' (Logout) link. Below this, a section titled 'Karta do głosowania' contains a question: 'Czy wyraża Pan / Pan zgodę na przystąpienie Rzeczypospolitej Polskiej do Unii Europejskiej?'. Two radio buttons are shown: one selected for 'TAK' (Yes) and one unselected for 'NIE' (No). At the bottom of this section is a blue 'Zatwierdź' (Confirm) button.

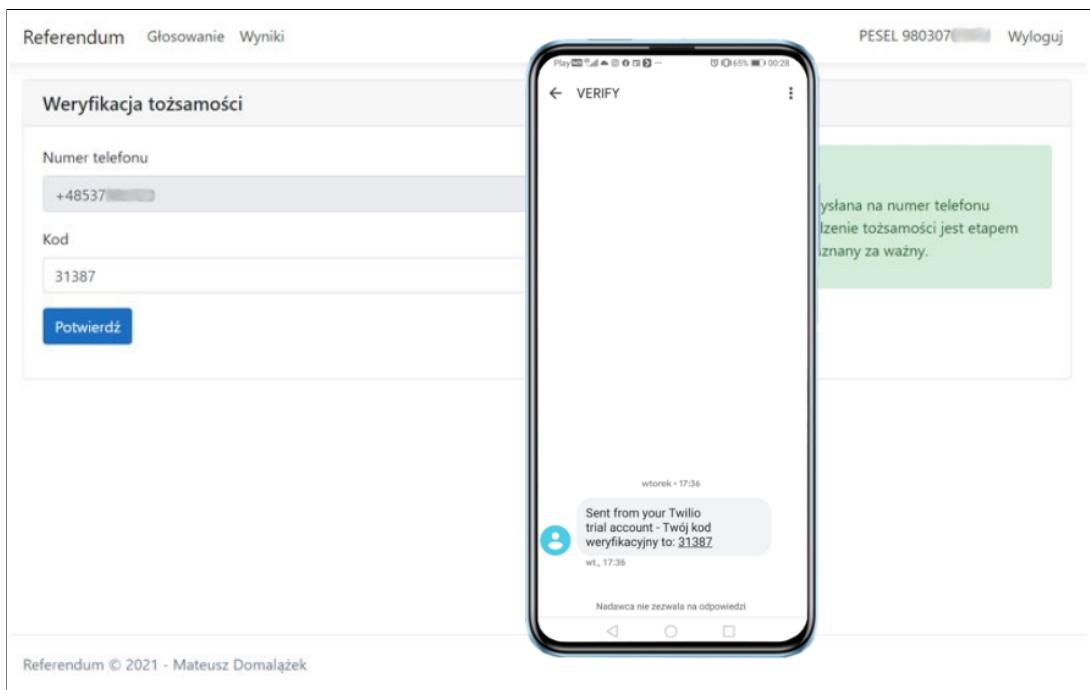
Rysunek 36: Karta do głosowania

Weryfikacja tożsamości polega na podaniu kodu z wiadomości SMS. W Polsce istnieje obowiązek przechowywania danych personalnych klientów również osób korzystających z telefonów na kartę [56], dlatego taka metoda została uznana za bezpieczną.

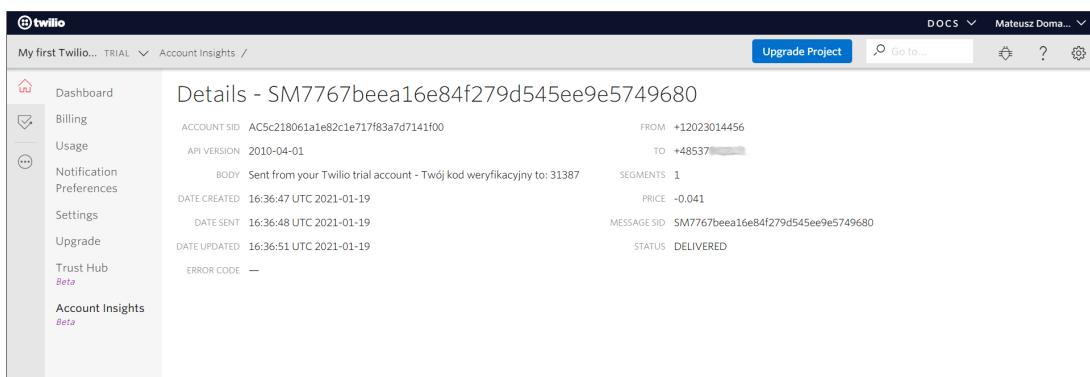
The screenshot shows a voter verification interface. At the top, there are navigation links: 'Referendum', 'Głosowanie', and 'Wyniki'. On the right, it displays 'PESEL 980307' and a 'Wyloguj' (Logout) link. Below this, a section titled 'Weryfikacja tożsamości' contains fields for 'Numer telefonu' (phone number +48537...) and 'Kod' (code). To the right, a green box contains the text: 'Kod weryfikacyjny' and 'Wiadomość SMS z kodem została wysłana na numer telefonu podany podczas rejestracji. Potwierdzenie tożsamości jest etapem koniecznym, aby głos mógł zostać uznany za ważny.' Below these fields is a blue 'Potwierdź' (Confirm) button.

Rysunek 37: Aktywne referendum

Po zatwierdzeniu karty wyborczej użytkownik otrzymuje wiadomość SMS z kodem liczbowym. Ustawienia Twillo zostały dobrane w taki sposób, aby uniemożliwić odpowiedź na otrzymaną informację. Zapis wykonanych połączeń jest także przechowywany przez Twilio wraz ze statusem określającym dostarczenie wiadomości. Bilingi umożliwiają również ustalenie lokalizacji wyborcy z uwagi na poniesione koszty związane z realizacją usługi telekomunikacyjnej.

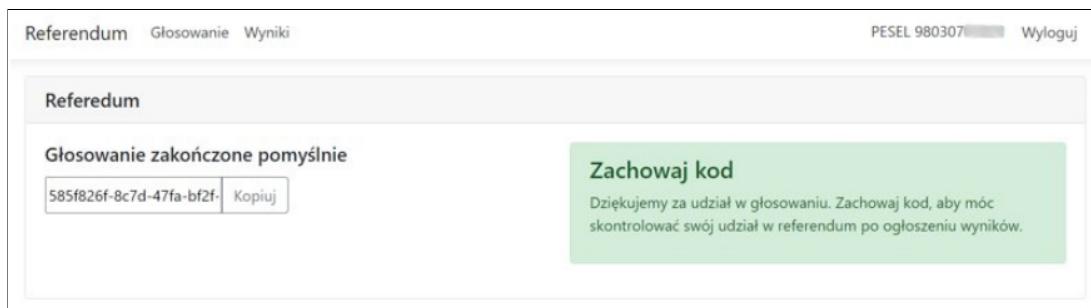


Rysunek 38: Dostarczona wiadomość SMS z kodem



Rysunek 39: Potwierdzenie dostarczenia wiadomości

Zwieńczeniem etapu decyzyjnego jest otrzymanie kodu identyfikacyjnego. Powinien zostać zachowany przez wyborcę, aby zweryfikować czy głos został zaliczony do ostatecznego werdyktu. Wygenerowana sekwencja jest numerem identyfikacyjnym klucza publicznego. Przykład zawiera kod "585f826f-8c7d-47fa-bf2f-fb6c58ca8b5d", który można odnaleźć w zbiorze głosów zaliczonych w protokole głosowania z rys. 31.



Rysunek 40: Zakończenie głosowania

5 Podsumowanie

Wynikiem działań przeprowadzonych w ramach niniejszej pracy inżynierskiej jest system do głosowania internetowego o nazwie Referendum. Opracowana aplikacja spełnia założenia oraz cele postawione na wstępie. Zrealizowano autorską implementację algorytmu Rivest-Shamira-Adlemana (RSA) dla kluczy 4096-bitowych, którą dokładnie opisano w rozdziale 4.3.1. Aplikacja referendalna stanowi internetowy odpowiednik wyborów korepondencyjnych i bazuje na koncepcji estońskiego systemu IVXV opisanego szerzej w rozdziale 3.3.2. Praca zawiera również aktualne informacje na temat rozwoju elektronicznych form wsparcia procesu wyborczego w Holandii, a także doświadczenia Ukrainy we wdrażaniu technologii Blockchain. Przeanalizowano możliwe zagrożenia wynikające z przeniesienia narzędzi demokracji bezpośredniej do sieci oraz dokonano klasyfikacji elektronicznych systemów, a całość tej analizy ujęto w rozdziale 3.

Aplikacja korzysta z usług zewnętrznych partnerów. Wysyłanie wiadomości SMS jest realizowane za pośrednictwem Twilio, natomiast informacje o lokalizacji wyborców są otrzymywane poprzez API portalu Ipstack, o czym więcej można przeczytać w rozdziale 4.3.2 oraz 4.3.4. Ponadto przedstawiono algorytm Tahera Elgamala oraz ideę podpisu cyfrowego. Prezentowana w pracy teoria mimo upływu czasu jest wciąż wykorzystywana w stworzeniu nowych rozwiązań kryptograficznych. W celu odpowiedniego przyswojenia treści zawarto także podstawę wiedzę z zakresu arytmetyki modularnej oraz algebry.

W przyszłości platforma może zostać rozszerzona o możliwość weryfikacji wyborcy poprzez wykorzystanie aplikacji mobilnej oraz graficzny interfejs dla organizatorów głosowania, aby ułatwić proces nadzoru nad działaniem witryny. Autor zamierza kontynuować prace nad udoskonaleniem aplikacji, aby przyczynić się do stworzenia pierwszego w Polsce systemu do przeprowadzania ogólnokrajowego referendum.

Literatura

- [1] Christof Paar, Jan Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Bochum, 2010
- [2] Robert Krimmer i inni, *5th International Joint Conference, E-Vote-ID 2020*, Bregenz, 2020
- [3] Peter Castenmiller, Arjan Dikmans, *Pushing water uphill; Renewal of the Dutch electoral process – 5th International Joint Conference, E-Vote-ID 2020*, Haga, 2020
- [4] Dmytro Khutkyy, *Blockchain-Enabled Electronic Voting: Experiments in Ukraine – 5th International Joint Conference, E-Vote-ID 2020*, Florencja, 2020
- [5] Jan Willemson, Sven Heiberg i Kristjan Krips, *Planning the next steps for Estonian Internet voting – 5th International Joint Conference, E-Vote-ID 2020*, Tartu, 2020
- [6] Valimised, IVXV online voting system [Online] <https://github.com/vvk-ehk/ivxv> (dostęp 02.2021)
- [7] Sarah Jamie Lewis, Olivier Pereira, Vanessa Teague, Trapdoor commitments in the SwissPost e-voting shuffle proof [Online] <https://people.eng.unimelb.edu.au/vjteague/SwissVote> (dostęp 02.2021)
- [8] PWN, definicja demokracji [Online] <https://encyklopedia.pwn.pl/haslo/demokracja;3891717.html> (dostęp 02.2021)
- [9] Paul F. Kisak, *The History of Democracy: "Historic Origins to The Present"* str. 2-4, London, 2016
- [10] Britannica, Ecclesia ancient Greek assembly [Online] <https://www.britannica.com/topic/Ecclesia-ancient-Greek-assembly> (dostęp 02.2021)
- [11] Parlament Europejski, Jak głosują Posłowie w Parlamencie Europejskim? [Online] <https://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+IM-PRESS+20060628STO09319+0+DOC+XML+V0//PL> (dostęp 02.2021)
- [12] Polskie Radio, Ateny: greccy sędziowe głosowali żetonami! [Online] <https://www.polskiерadio.pl/23/266/Artykul/591707,Ateny-greccy-sedziowe-glosowali-zetonami> (dostęp 02.2021)

- [13] Twoja Historia, Dlaczego starożytni Rzymianie nie chcieli brać udziału w wyborach? Wyjaśnienia polskiej badaczki [Online] <https://twojahistoria.pl/2019/03/07/dlaczego-starozytni-rzymianie-nie-chcieli-brac-udzialu-w-wyborach-wyjasnienia-polskiej-badaczki/> (dostęp 02.2021)
- [14] Taylor L. Ross, *Roman voting assemblies from the Hannibalic War to the dictatorship of Caesar*, str. 34-35, Ann Arbor, 1990
- [15] Wikipedia, Problemy milenijne [Online] https://pl.wikipedia.org/wiki/Problemy_milenijne (dostęp 02.2021)
- [16] Wikipedia, Problem NP [Online] https://pl.wikipedia.org/wiki/Problem_NP (dostęp 02.2021)
- [17] Elżbieta Kotlicka-Dwurznik, Bożenna Szkopińska, Witold Walas, *Elementy algebra i geometrii analitycznej*, str. 7-9, Łódź, 2011
- [18] Encyclopedia of Mathematics, Ciało skończone o p elementach [Online] https://encyclopediaofmath.org/wiki/Prime_field (dostęp 02.2021)
- [19] Wikipedia, Grupa multiplikatywna [Online] https://pl.wikipedia.org/wiki/Grupa_multiplikatywna (dostęp 02.2021)
- [20] MathEdu, Największy wspólny dzielnik [Online] <http://www.math.edu.pl/nwd> (dostęp 02.2021)
- [21] Agata Pilitowska, *Algebra i jej zastosowania - konspekt wykładu*, Warszawa, 2018
- [22] GUS, raport - społeczeństwo informacyjne w 2020 roku [Online] <https://stat.gov.pl/obszary-tematyczne/nauka-i-technika-spoleczenstwo-informacyjne/spoleczenstwo-informacyjne/spoleczenstwo-informacyjne-w-polsce-w-2020-roku,2,10.html> (dostęp 02.2021)
- [23] Michał Rajkowski, Systemy głosowania elektronicznego [Online] http://cygnus.tele.pw.edu.pl/~zkotulsk/seminarium/System_glosowania.pdf (dostęp 02.2021)
- [24] Dziennik Ustaw, Kodeks wyborczy rozdział 2 [Online] <https://sip.lex.pl/akty-prawne/dziennik-ustaw/kodeks-wyborczy-17679859/dz-1-roz-2> (dostęp 02.2021)

- [25] Cristina Pérez Espés, *Effectiveness of ecognocracy: a socialemconomic approach* str. 92 Zaragoza, 2015,
- [26] Arjen K. Lenstra i Eric R. Verheul, *Selecting Cryptographic Key Sizes*, Journal Of Cryptology, tom 14, str. 255-293, 2001.
- [27] Nigel P. Smart, Michel Abdalla, Tor Erling Bjørstad and others Algorithms Key Size and Protocols Report 2018 [Online] <https://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf> (dostęp 02.2021)
- [28] Elaine Barker, Recommendation for Key Management NIST [Online] <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf> (dostęp 02.2021)
- [29] Agence nationale de la sécurité des systèmes d'information, Referentiel General de Se-curite ANSSI [Online] https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B1.pdf (dostęp 02.2021)
- [30] Federal Office for Information Security, Cryptographic Mechanisms: Recommendations and Key Lengths BSI, 2020 https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile (dostęp 02.2021)
- [31] Ukraine Crisis, Five questions from the president. What does Zelensky want from Ukrainians? [Online] <https://uacrisis.org/en/five-questions-from-the-president> (dostęp 02.2021)
- [32] Narada, evox voting system [Online] <https://bitbucket.org/evoxvoting/> (dostęp 02.2021)
- [33] Ethereum, smart-contracts [Online] <https://ethereum.org/en/developers/docs/smart-contracts/> (dostęp 02.2021)
- [34] Ukrainian Cultural Foundation, the election for part of the Supervisory Board of the UCF, 2020 [Online] <https://ucf.in.ua/en/news/26-11-2020> (dostęp 02.2021)
- [35] RGC, voting results, 2020 [Online] <https://nabu.gov.ua/novyny/rezultaty-reytyngovogo-internet-golosuvannya-za-radu-gromadskogo-kontrolyu-pry-nabu-vi> (dostęp 02.2021)
- [36] Valimised, voting result Estonia, 2019 [Online] <https://rk2019.valimised.ee/en/voting-result/voting-result-main.html> (dostęp 02.2021)

- [37] Arne Ansper, Ahto Buldas, Aivo Jürgenson, E-voting concept security: analysis and measures, 2010 [Online] https://www.valimised.ee/sites/default/files/uploads/eng/E-voting_concept_security_analysis_and_measures_2010.pdf (dostęp 02.2021)
- [38] Cybernetica, Mobile voting feasibility study and risk analysis, 2020 [Online] https://www.valimised.ee/sites/default/files/uploads/eng/2020_m-voting-report.pdf (dostęp 02.2021)
- [39] Valimised, government announcement - Voting in polling places becomes more flexible this year, 2020 [Online] <https://www.valimised.ee/en/voting-polling-places-becomes-more-flexible-year> (dostęp 02.2021)
- [40] Dominionvoting, firma sprzedająca sprzęt i oprogramowanie do głosowania elektronicznego [Online] <https://www.dominionvoting.com/about/> (dostęp 02.2021)
- [41] NDI, Electronic Voting Project Management in the Netherlands [Online] <https://www.ndi.org/e-voting-guide/examples/e-voting-project-management-netherlands> (dostęp 02.2021)
- [42] Kiesraad, Senate approves repeal of Advisory Referendum Act [Online] <https://english.kiesraad.nl/latest-news/news/2018/07/12/senate-approves-repeal-of-advisory-referendum-act> (dostęp 02.2021)
- [43] OSV, Toetsingsrapport Expleo OSV2020 programma Uitslagvaststelling [Online] <https://www.kiesraad.nl/verkiezingen/adviezen-en-publicaties/formulieren/2020/10/5/toetsingsrapport-expleo-osv2020-programma-uitslagvaststelling> dostęp (02.2021)
- [44] Python, dokumentacja biblioteki multiprocessing [Online] <https://docs.python.org/3.8/library/multiprocessing.html> dostęp (02.2021)
- [45] Wikipedia, test Millera-Rabina [Online] https://en.wikipedia.org/wiki/Test_Millera-Rabina dostęp (02.2021)
- [46] Microsoft, dokumentacja klasy DataProtectionKey [Online] <https://docs.microsoft.com/en-us/dotnet/api/microsoft.aspnetcore.dataprotection.entityframeworkcore.dataprotectionkey> (dostęp 02.2021)
- [47] Wikipedia, kodowanie transportowe base64 [Online] <https://pl.wikipedia.org/wiki/Base64> (dostęp 02.2021)

- [48] Wikipedia, uniwersalny czas koordynowany [Online] <https://pl.wikipedia.org/wiki/UTC> (dostęp 02.2021)
- [49] Wikipedia, opis funkcji floor oraz ceiling [Online] https://en.wikipedia.org/wiki/Floor_and_ceiling_functions (dostęp 02.2021)
- [50] Python, opis funkcji divmod [Online] <https://docs.python.org/3.8/library/functions.html#divmod> (dostęp 02.2021)
- [51] Python, opis funkcji pow [Online] <https://docs.python.org/3.8/library/functions.html#pow> (dostęp 02.2021)
- [52] Brian Hook, *Write portable code. A guide to developing software for multiple platforms, San Francisco, 2005*
- [53] Twilio, dokumentacja techniczna [Online] <https://www.twilio.com/docs/sms> (dostęp 02.2021)
- [54] Ipstack, dokumentacja techniczna [Online] <https://ipstack.com/documentation> (dostęp 02.2021)
- [55] Wikipedia, opis serwera proxy [Online] https://en.wikipedia.org/wiki/Proxy_server (dostęp 02.2021)
- [56] Bezprawnik, rejestracja telefonów na kartę [Online] <https://bezprawnik.pl/rejestracja-telefonu-na-karte/> (dostęp 02.2021)

Spis kodów

1	Generowanie par losowych liczb pierwszych o łącznej długości 4096 bitów	49
2	Tworzenie puli wątków do generowania liczb pierwszych	50
3	Generowanie losowej liczby pierwszej	50
4	Test pierwszości Millera-Rabina	51
5	Generowanie kluczy	51
6	Wyznaczenie parametru e	52
7	Zapis kluczy do pliku govkey	52
8	Zapis klucza publicznego do bazy danych	52
9	Szyfrowanie wiadomości	53
10	Obliczenie ilości bajtów	53
11	Obliczenie sufitu z wyniku ilorazu dwóch liczb	53
12	Wypełnienie wiadomości	54
13	Konwersja bajtów na liczbę całkowitą	54
14	Konwersja liczby całkowitej na bajty	54
15	Odszyfrowanie wiadomości	55
16	Kontroler SMS	56
17	Kontroler lokalizacji	56
18	Funkcja do obsługi karty wyborczej	58
19	Funkcja weryfikująca	59
20	Para kluczy w pliku govkey	63

Spis rysunków

1	Przykład działania RSA	12
2	Przykład działania protokołu ElGamala	13
3	Wzrost liczby użytkowników e-podpisu w Polsce	14
4	Ogólny schemat systemu IVXV	22
5	Schemat koperty w systemie IVXV	23
6	Schemat ogólny platformy referendalnej	26
7	Schemat bazy danych	27
8	Diagram przypadków użycia	28
9	Ciąg podstawowy - rejestracja	30
10	Ciąg alternatywny - rejestracja	31
11	Ciąg podstawowy - logowanie	32
12	Ciąg alternatywny - logowanie	33
13	Ciąg podstawowy - wylogowanie	34
14	Ciąg alternatywny - wylogowanie	34
15	Ciąg podstawowy - utworzenie pytań i odpowiedzi	35
16	Ciąg alternatywny - utworzenie pytań i odpowiedzi	36
17	Ciąg podstawowy - utworzenie referendum	37
18	Ciąg alternatywny - utworzenie referendum	38
19	Ciąg podstawowy - głosowanie	39
20	Ciąg alternatywny - głosowanie	40
21	Ciąg podstawowy - dostarczenie danych geolokalizacyjnych	41
22	Ciąg alternatywny - dostarczenie danych geolokalizacyjnych	41
23	Ciąg podstawowy - wysłanie kodu SMS	42
24	Ciąg alternatywny - wysłanie kodu SMS	43

25	Ciąg podstawowy - wpisanie kodu SMS	44
26	Ciąg alternatywny - wpisanie kodu SMS	45
27	Ciąg podstawowy - weryfikacja wyborcy	46
28	Ciąg alternatywny - weryfikacja wyborcy	46
29	Ciąg podstawowy - opublikowanie wyników	47
30	Ciąg alternatywny - opublikowanie wyników	48
31	Wyniki głosowania	61
32	Wygenerowane liczby pierwsze	62
33	Generowanie liczb pierwszych - czas wyszukiwania kolejnych par	62
34	Rejestracja konta wyborcy	64
35	Aktywne referendum	64
36	Karta do głosowania	65
37	Aktywne referendum	65
38	Dostarczona wiadomość SMS z kodem	66
39	Potwierdzenie dostarczenia wiadomości	66
40	Zakończenie głosowania	67