INFORMATYCZNY SYSTEM ZARZĄDZANIA BUDYNKIEM - INICJALIZACJA SYSTEMU ORAZ OBSŁUGA KONT UŻYTKOWNIKA

1. ZAKRES PROJEKTOWANEGO SYSTEMU

Projektowany na potrzeby projektu Dom 2020 informatyczny system zarządzania (SIZ) prezentowanym budynkiem stanowi swoisty łącznik, umożliwiając kompleksowe zarządzanie instalacjami oraz odbiorami z poziomu jednego, intuicyjnego panelu dotykowego zainstalowanego w budynku.

W projektowaniu systemów informatycznych wyróżnia się dwa typy wymagań. Pierwszy, wymagania funkcjonalne, to zbiór funkcji jakie powinien spełniać powstały system. Funkcjonalności te, w większości określane są na podstawie analizy potrzeb użytkownika końcowego, jednakże pewna ich część wynika z zasad użytkowania systemu oraz bezpieczeństwa przechowywanych w nim danych. Drugi typ wymagań stanowią wymagania niefunkcjonalne, inaczej zwane sprzętowymi. Definiują one parametry sprzętu, na którym będzie zainstalowany system.

Wśród głównych wymagań funkcjonalnych projektowanego SIZ odnoszących się do potrzeb użytkownika końcowego – mieszkańca budynku należą:

- zarządzanie wybranymi funkcjami zainstalowanych w budynku instalacji tj. rekuperator, zbiornik na deszczówkę, sterowanie chłodzeniem ogniw fotowoltaicznych za pomocą obiegu glikolu, zbiornik CWU, rolety czy kominek
- zarządzanie odbiorami, klasyfikującymi się jako odpowiednie do zdalnego zarządzania bez poczucia dyskomfortu u mieszkańców
- optymalizacja poziomu zużycia energii elektrycznej. Proces optymalizacji zużycia energii elektrycznej rozpoczyna się od zbierania danych dotyczących krzywej zapotrzebowania na energię mieszkańców budynku. Na podstawie zebranych danych, SIZ ustala średnie poziomy zużycia w dni powszednie, świąteczne oraz z uwzględnieniem pory roku. Porównanie otrzymanych krzywych z mocą otrzymywaną z zainstalowanych w budynku OZE pozwoli na wykrycie okresów szczytowego zapotrzebowania, które wymagają dodatkowego poboru energii elektrycznej z sieci. Sterowanie częścią odbiorów pozwoli na „złagodzenie" krzywej obciążeń tak, aby zminimalizować pobór energii z sieci. Natomiast w trakcie zbierania danych, przed wyznaczeniem krzywych średniego zapotrzebowania, możliwe jest ograniczanie zużycia energii elektrycznej za pomocą sterowania odbiorami oraz informowania mieszkańców o okresach, w których cena energii jest wysoka oraz niska
- komunikacja z „inteligentnym" licznikiem zainstalowanym w budynku
- monitorowanie podstawowych parametrów tj. temperatura i wilgotność powietrza, poziom tlenku węgla, dwutlenku węgla
- wykrywanie stanów awaryjnych
- obsługa procesu ładowania pojazdu elektrycznego
- zarządzanie monitoringiem otoczenia za pomocą kamer
- łączenie się z SIZ za pomocą Internetu (i połączenia szyfrowanego SSL) umożliwiające zarządzanie budynkiem z dowolnego miejsca
- „tryb wakacyjny" obejmujący sterowanie wybranymi instalacjami oraz oświetleniem podczas wyjazdu mieszkańców. Dodatkowo możliwość łączenia się z SIZ za pomocą Internetu gwarantuje monitoring (czujniki i kamery) budynku podczas nieobecności mieszkańców.

Wśród wymagań wynikających z zasad użytkowania systemu oraz bezpieczeństwa przechowywanych w nim danych wymienić można m.in.: inicjalizację (pierwsze

uruchomienie) systemu, procedury logowania obejmujące weryfikację haseł czy obsługę kont użytkowników.

W kolejnych podrozdziałach opisane zostaną funkcje realizowane przez system w ramach wymagań funkcjonalnych. Dla logicznego uporządkowania procesów, wymagania te zebrano w zbiory: inicjalizacja systemu, obsługa kont użytkownika, zarządzanie odbiorami, zarządzanie instalacjami, obsługa pojazdu elektrycznego oraz optymalizacja zużycia energii elektrycznej.

W końcowej części rozdziału zostaną zaprezentowane wymagania niefunkcjonalne zalecane dla projektowanego systemu.

2. WYMAGANIA FUNKCJONALNE – INICJALIZACJA SYSTEMU ORAZ OBSŁUGA KONT UŻYTKOWNIKA

Wymagania funkcjonalne, oprócz szczegółowego opisu, zilustrowane zostały za pomocą diagramów przypadków użycia oraz diagramów czynności wykonanych w notacji UML. Diagramy przypadków użycia [15] stosowane są do definiowania funkcjonalności analizowanego i projektowanego systemu jak również sposobów interakcji użytkownik – system. Innymi słowy, jest to graficzna prezentacja obejmująca przypadki użycia, aktorów oraz relacje występujące pomiędzy nimi. Pod pojęciem aktora należy rozumieć zestaw ról odgrywanych przez użytkowników końcowych projektowanego systemu w czasie wykonywania danego przypadku użycia. Jednakże aktorem może być nie tylko człowiek – użytkownik systemu, lecz również odbiór czy czas, np. określony dzień miesiąca. Diagram czynności [15] koncentruje się natomiast na dynamicznym aspekcie systemu, ilustrując sekwencję czynności i akcji oraz przepływy sterowania i danych realizowanych w procesach systemowych. Przepływy mogą mieć charakter sekwencyjny lub współbieżny. Diagramy czynności prezentują scenariusze przypadków użycia za pomocą sekwencji warunków – pętli, które mogą prowadzić do wykonania różnych zestawów akcji w zależności od otrzymanych danych wejściowych. Połączenie tych dwóch rodzajów diagramów umożliwi dokładne zrozumienie projektowanych procesów.

Należy równocześnie pamiętać, iż czytelność prezentowanych diagramów wymusza pominięcie części mniej istotnych funkcji, które zostały jednakże ujęte w opisie. Na przykład, większość wykonywanych czynności może być w każdej chwili anulowana przez użytkownika. Umieszczenie pytania „Czy anulowano?” po każdej akcji w diagramie czynności znacznie zmniejszyłoby jego czytelność.

Wybrane fragmenty procesów zostały dodatkowo opatrzone projektem interfejsu graficznego[i].

2.1. Inicjalizacja systemu

2.1.1. Pierwsze uruchomienie

Inicjalizacja systemu oznacza sparametryzowanie umożliwiające jak najlepsze dopasowanie jego funkcjonalności do charakterystyki budynku, w którym został zainstalowany. Proces ten odbywa się za pomocą wprowadzenia żądanych danych przez użytkownika. Dane te dotyczą podstawowych parametrów budynku, liczby mieszkańców, czujników i kamer oraz odbiorów, których praca będzie przez system koordynowana.

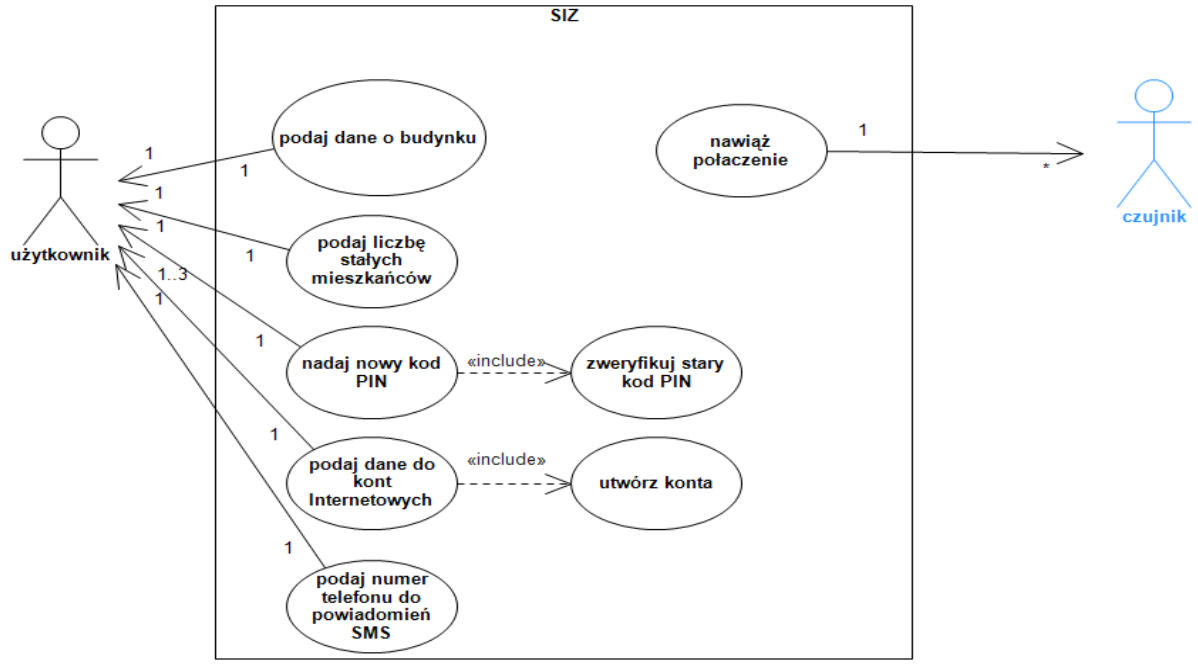
W celu ułatwienia procesu inicjalizacji systemu, zaprojektowana została możliwość wgrania parametrów budynku z zewnętrznego nośnika pamięci. Nośnik taki mógłby zostać nagrany przez biuro architektoniczne zajmujące się projektem budynku. W przypadku braku takiego nośnika, użytkownik wprowadza dane ręcznie.

Inicjalizacja systemu odbywa się jednorazowo na początku jego użytkowania. Jeżeli wystąpi konieczność modyfikacji wprowadzonych danych, np. w przypadku zwiększenia liczby mieszkańców, możliwe jest wprowadzenie zmian w ustawieniach systemowych.

Oprócz informacji o budynku oraz liczby mieszkańców na etapie inicjalizacji nadawane jest także nowe hasło zabezpieczające. Domyślnie do systemu wprowadzone jest hasło jednorazowe, które w trakcie pierwszego uruchomienia użytkownik zmienia na wybrane przez siebie. Hasło to wykorzystywane jest do autoryzacji wykonywania wybranych operacji, przy uruchamianiu systemu oraz przy logowaniu na konto użytkownika – administratora.

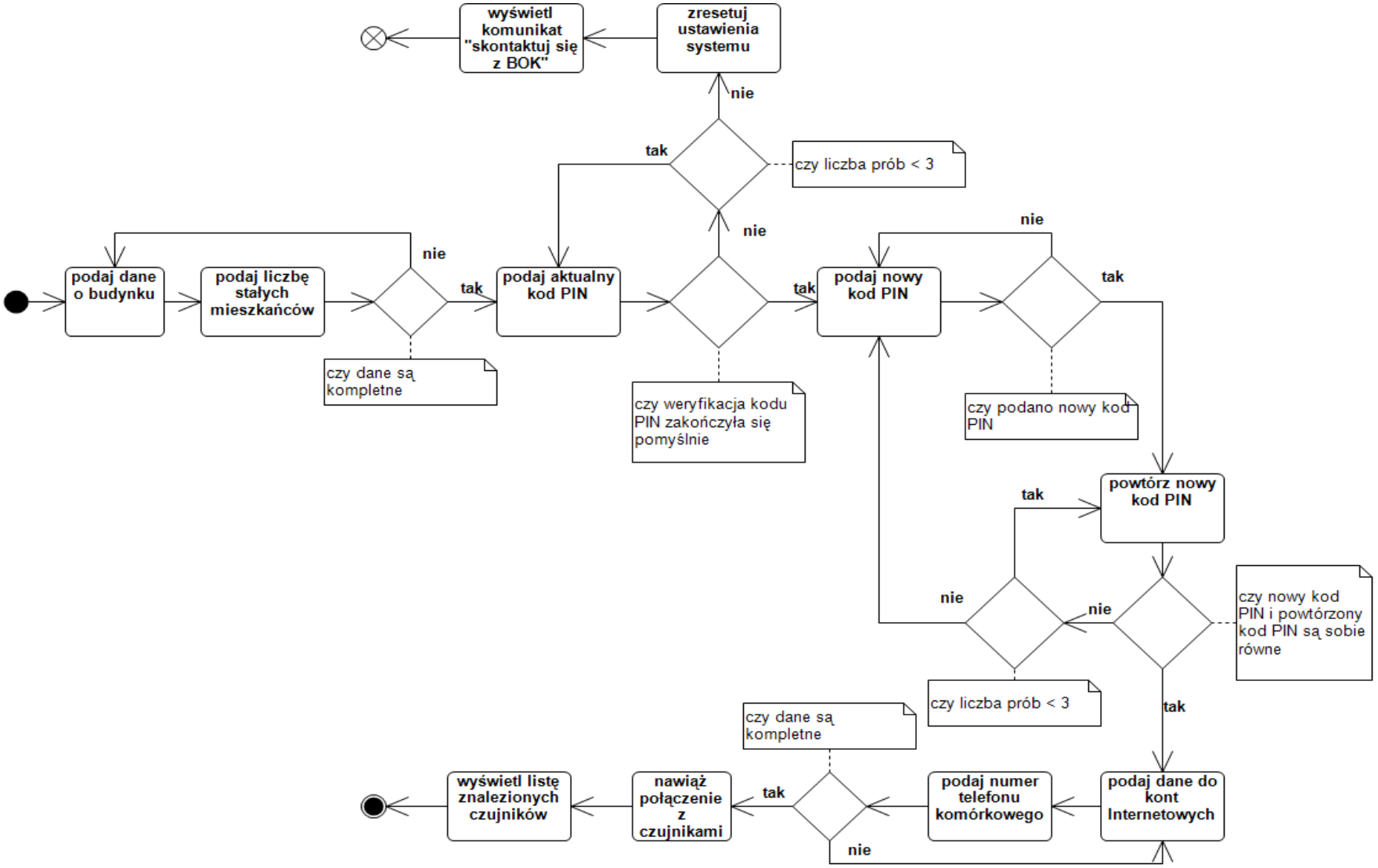
Ostatnim elementem procesu inicjalizacji jest zdefiniowanie kont użytkowników posiadających dostęp do systemu przez Internet. Dostęp ten umożliwia zarządzanie wybranymi aspektami systemowymi po zalogowaniu się na stronę internetową. Dozwolone jest założenie trzech takich kont użytkowników. Możliwe jest także podanie jednego numeru telefonu komórkowego, na który będą wysyłane powiadomienia SMS. W przypadku, gdy dane te nie zostały wprowadzone w trakcie inicjalizacji lub konieczności ich modyfikacji, możliwe jest ich uzupełnienie oraz korekta poprzez menu ustawienia systemowe.

Diagram przypadków użycia ilustrujący proces inicjalizacji systemu został przedstawiony poniżej.



Rysunek numer 1. Diagram przypadków użycia – inicjalizacja systemu.
Źródło: opracowanie własne.

Szczegółowy przebieg procesu został zilustrowany za pomocą diagramu czynności przedstawionym na następnym rysunku.

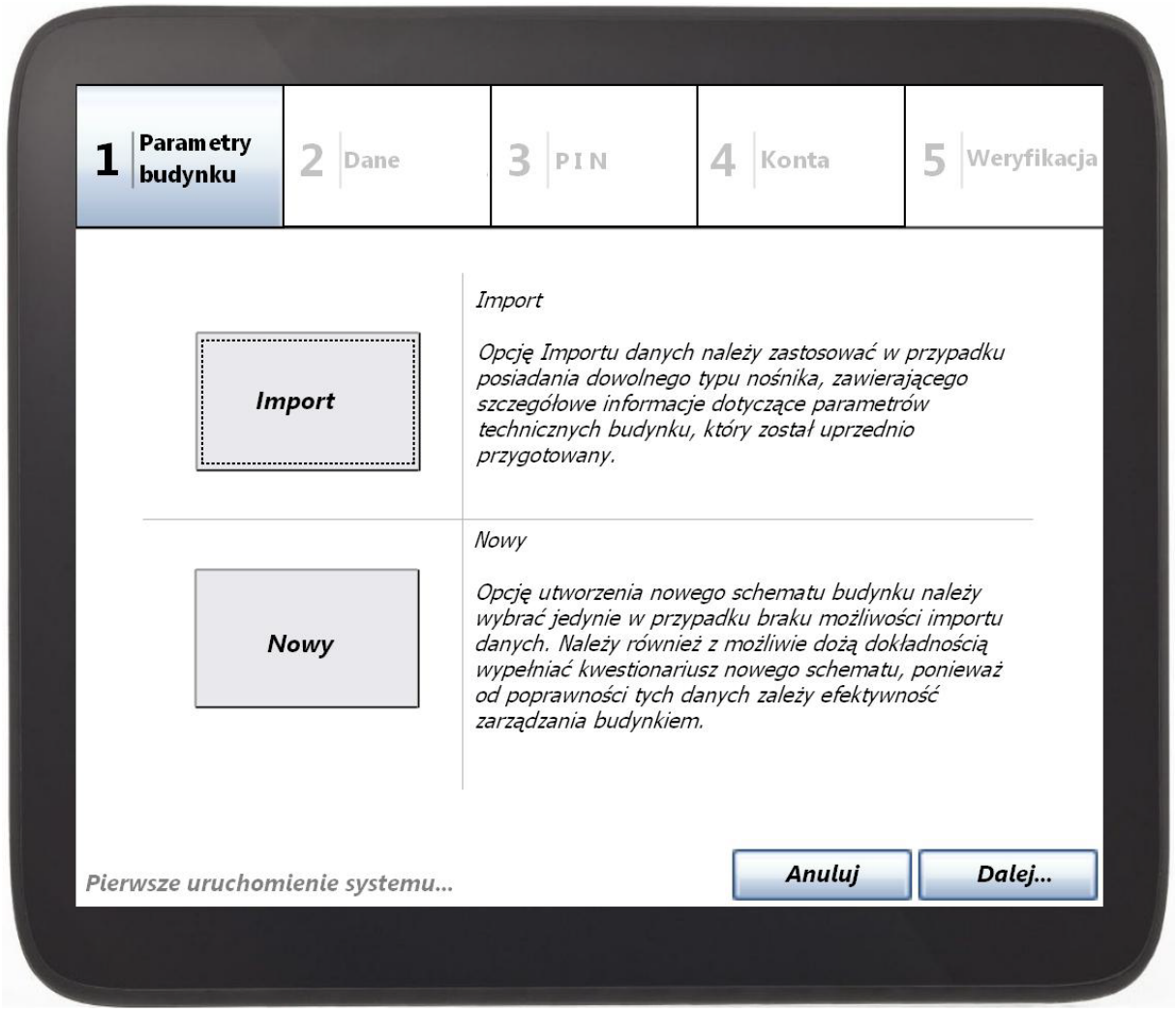


Rysunek numer 2. Diagram czynności – inicjalizacja systemu.
Źródło: opracowanie własne.

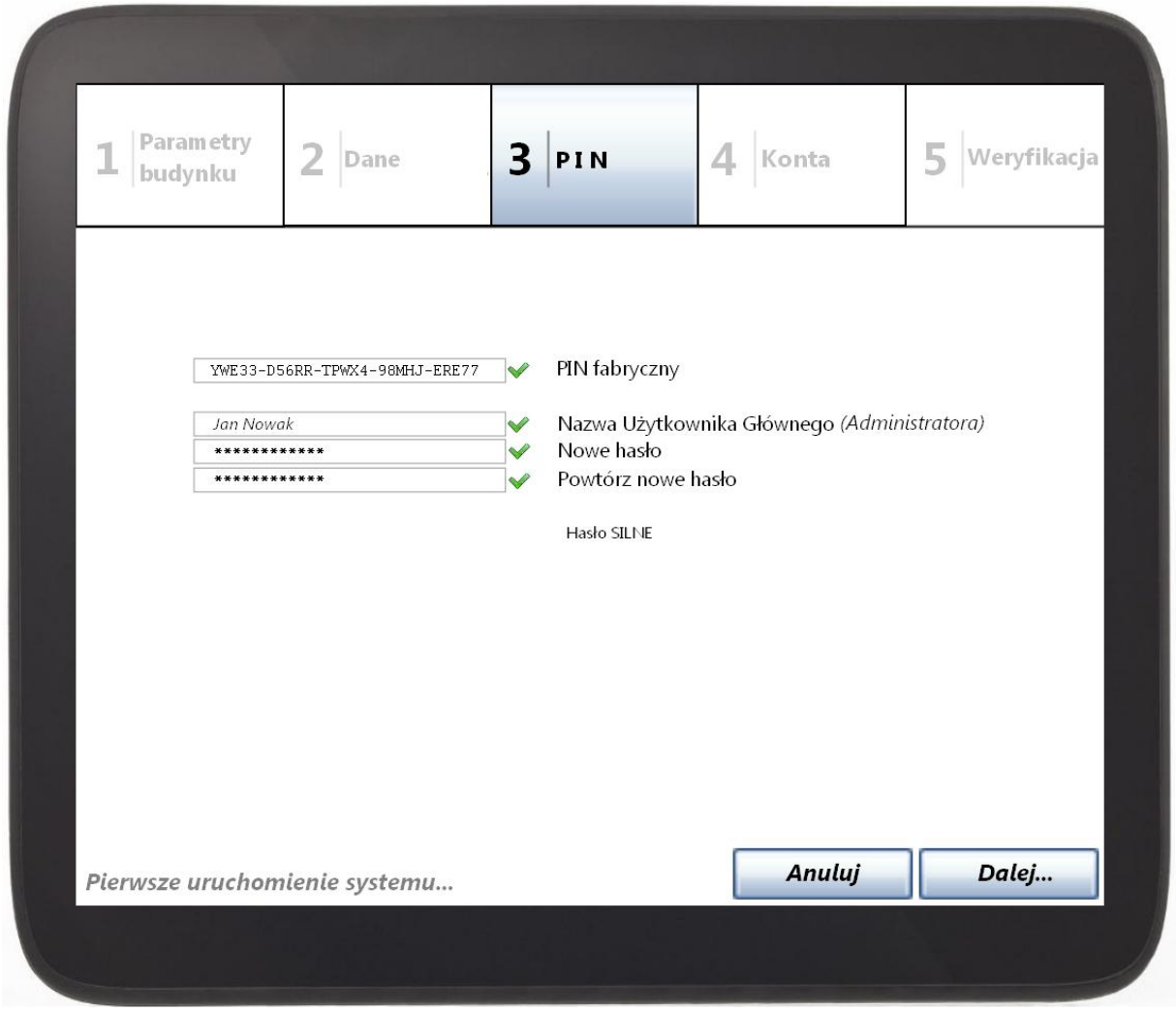
Proces rozpoczyna się od wprowadzenia danych o budynku oraz liczby stałych mieszkańców. System weryfikuje ich kompletność i dopiero po jej uzyskaniu przechodzi do następnego etapu. Użytkownik wprowadza otrzymany, fabryczny kod PIN (hasło).

Po trzech nieudanych próbach wprowadzone ustawienia zostają zresetowane oraz wyświetlany jest komunikat o konieczności kontaktu z Biurem Obsługi Klienta. Po pozytywnej weryfikacji wprowadzonego hasła użytkownik wprowadza własne, nowe hasło a następnie wprowadza je ponownie w celu jego potwierdzenia. Jest to standardowa procedura wykorzystywana przy zmianie haseł. Zabezpiecza ona przed błędami przy wprowadzaniu hasła jak na przykład omyłkowe wybranie dodatkowego klawisza. Po trzech nieudanych próbach powtórnego wprowadzenia hasła, system wraca do etapu pierwszego wprowadzenia nowego hasła. W kolejnych etapach użytkownik wprowadza dane dotyczące kont do dostępu przez Internet oraz numer telefonu komórkowego a system weryfikuje ich kompletność. Jeżeli dane są kompletne, system próbuje nawiązać połączenie z czujnikami zainstalowanymi w budynku oraz kamerami monitoringu (jeżeli są zainstalowane). Po zakończeniu tego etapu wyświetlana jest lista znalezionych urządzeń. W przypadku gdy wystąpiły problemy z połączeniem, przykładowo gdy któryś z czujników był w momencie inicjalizacji systemu wyłączony, można przeprowadzić proces komunikacji wybierając odpowiednią opcję z menu systemowego.

Na rysunkach numer 3 oraz 4 przedstawiono dwa interfejsy graficzne prezentujące fragmenty opisywanego procesu.



Rysunek numer 3. Interfejs graficzny – inicjalizacja systemu, parametry budynku.
Źródło: opracowanie własne



Rysunek numer 4. Interfejs graficzny – inicjalizacja systemu, definiowanie nowego hasła.
Źródło: opracowanie własne

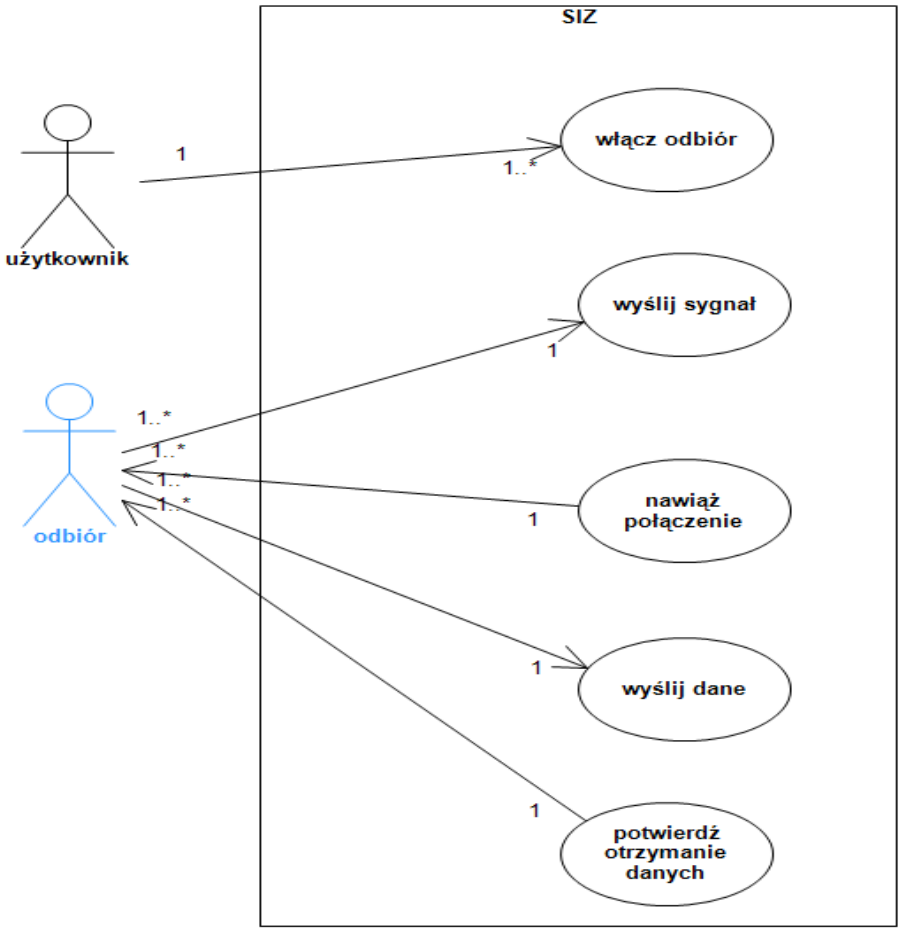
2.1.2. Inicjalizacja systemu – odbiory „inteligentne”

Pod pojęciem „inteligentnych” odbiorów należy rozumieć urządzenia gospodarstwa domowego wyposażone w modemy do komunikacji bezprzewodowej. Urządzenia te posiadają zdolność do współpracy z SIZ: system może sterować pracą tych urządzeń. O ile dziś odbiory „inteligentne” nie są mocno rozpowszechnione, traktowane są jako technologie niezbyt odległej przyszłości. Z tego powodu należy przewidzieć ich udział w projektowaniu informatycznego systemu zarządzania budynkiem inteligentnym. W niniejszym opracowaniu za odbiory „inteligentne” uznano: pralkę, zmywarkę, ładowarkę pojazdu elektrycznego, suszarkę do ubrań oraz cykl rozmrażania lodówki.

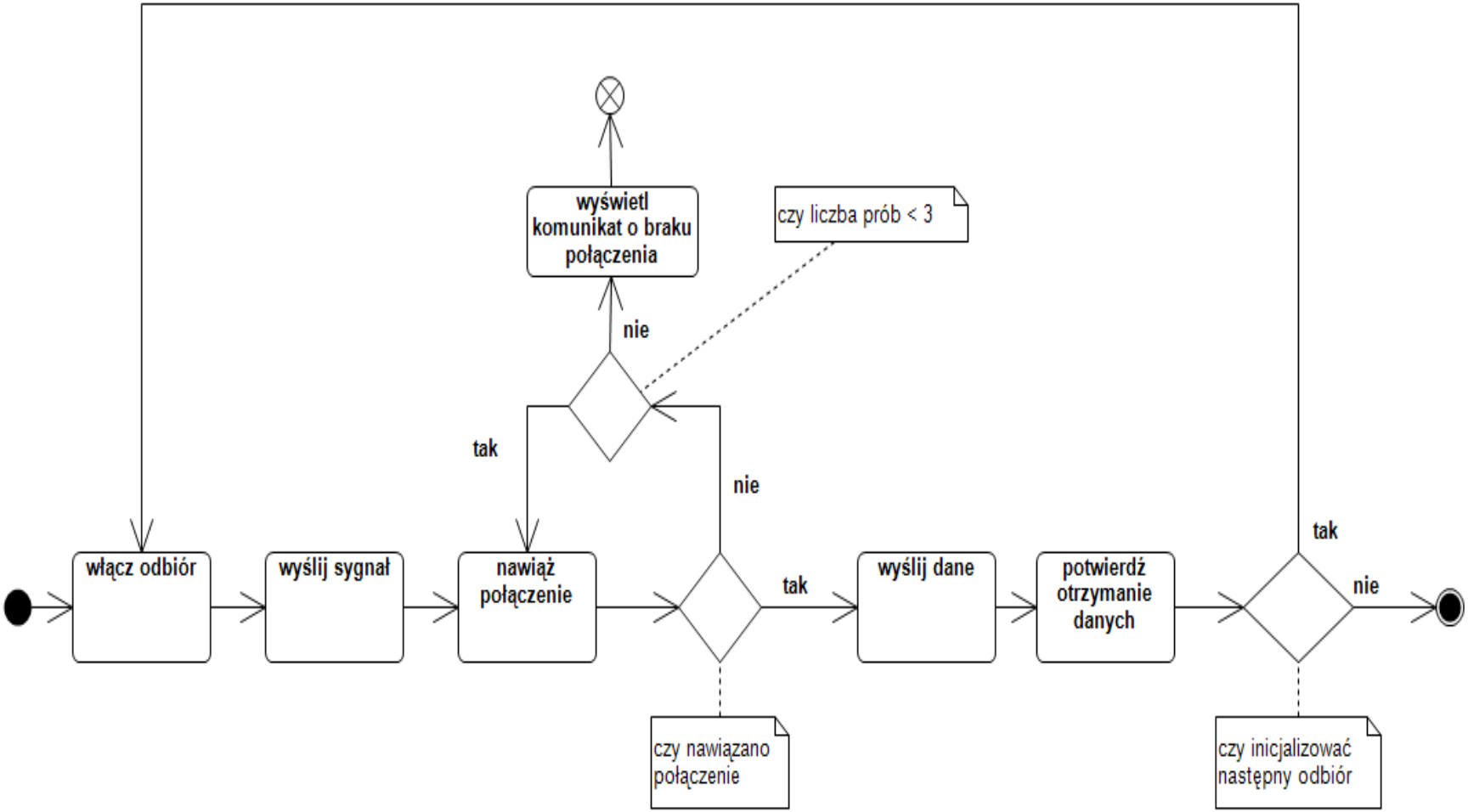
Inicjalizacja takich urządzeń wymaga od użytkownika ich podłączenia tak aby system mógł nawiązać z nimi łączność. W razie problemów z nawiązaniem połączenia, system podejmuje kolejno trzy próby. Jeżeli wszystkie zakończą się niepowodzeniem, wyświetlany jest komunikat informujący o zaistniałym błędzie. Po nawiązaniu komunikacji następuje dodanie odbioru do bazy danych w systemie oraz pobranie wybranych parametrów odbioru. Przykładowo, jeżeli dodawanym odbiorem byłaby zmywarka, pobrane dane dotyczyłyby parametrów zaimplementowanych w urządzeniu programów zmywania: ilości zużywanej

energii elektrycznej, czasie trwania, nazwie programu, itp. Informacje te znajdą zastosowanie w projektowaniu harmonogramów oraz w procesie optymalizacji zużycia energii elektrycznej.

Po dodaniu jednego odbioru system rozpoczyna wprowadzanie kolejnych. Proces ten trwa dopóki SIZ nie wprowadzi danych o wszystkich odbiorach, z którymi nawiązała połączenie. Na końcu procesu zostanie wyświetlona lista odbiorów, które zostały dodane pomyślnie. Jeżeli z jakiegokolwiek powodu inicjalizacja odbioru rozpoczęła się ale nie zakończyła sukcesem, informacja ta również zostanie wyświetlona w końcowym komunikacie. Funkcjonalność ta została ujęta na diagramie numer 5 oraz 6.



Rysunek numer 5. Diagram przypadków użycia – inicjalizacja systemu, odbiory „inteligentne”.
Źródło: opracowanie własne.



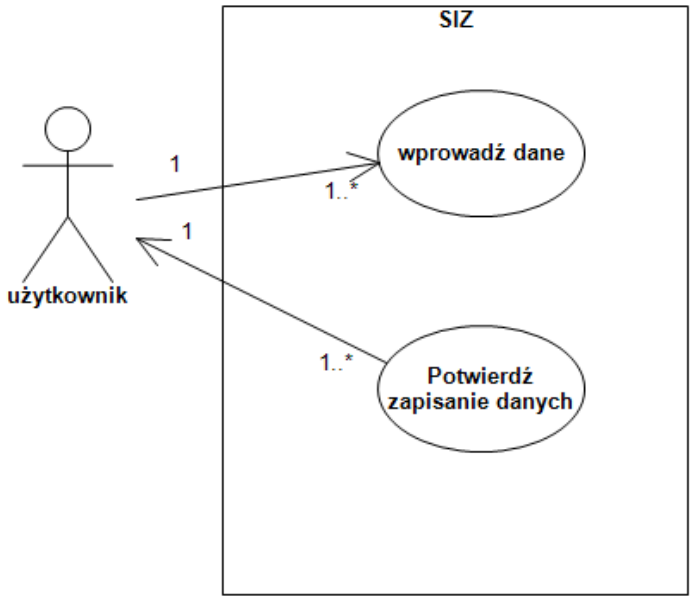
Rysunek numer 6. Diagram czynności – inicjalizacja systemu, odbiory „inteligentne”.
Źródło: opracowanie własne

Na rysunku numer 7 przedstawiono interfejs graficzny ilustrujący inicjalizację odbiorów „inteligentnych”.

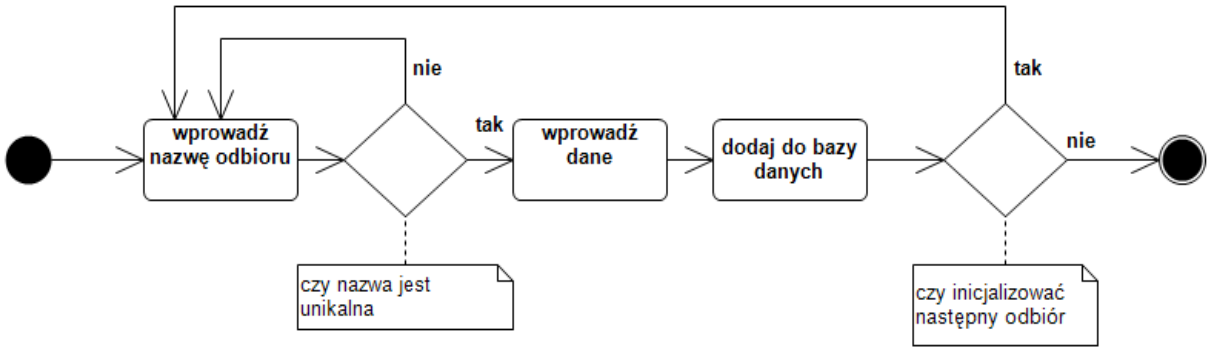
Rysunek numer 7. Interfejs graficzny – inicjalizacja systemu, odbiory „inteligentne”.
Źródło: opracowanie własne

2.1.3. Inicjalizacja systemu – odbiory tradycyjne

Odbiory tradycyjne nie są wyposażone w moduł komunikacji z systemem, zatem nie może on sterować ich pracą. Pełnią one istotną rolę w planowaniu zapotrzebowania na energię elektryczną dla gospodarstwa domowego oraz w procesie jego optymalizacji, dlatego też dane o parametrach tych odbiorów powinny zostać ujęte w systemie. Użytkownik samodzielnie wprowadza informacje w procesie ich inicjalizacji, jak również w okresie eksploatacji systemu informacje o ich używaniu. Zakres wprowadzanych informacji jest taki sam jak w przypadku odbiorów „inteligentnych”. Proces inicjalizacji urządzeń tradycyjnych został przedstawiony kolejno na rysunkach 8 oraz 9 i, pomijając etap komunikacji systemu z odbiorem, przedstawia on się analogicznie do procesów z poprzedniego podrozdziału.



Rysunek numer 8. Diagram przypadków użycia – inicjalizacja systemu, odbiory tradycyjne.
Źródło: opracowanie własne.



Rysunek numer 9. Diagram czynności – inicjalizacja systemu, odbiory tradycyjne.
Źródło: opracowanie własne.

2.2. OBSŁUGA KONT UŻYTKOWNIKA

System domyślnie posiada wbudowane konto administratora. Do konta tego przypisane jest hasło nadawane w procesie inicjalizacji. Jak wspominano w założeniach projektowanego systemu, przewidywane są dodatkowo konta umożliwiające zarządzanie i monitoring wybranych funkcji poprzez Internet za pomocą połączenia szyfrowanego. Zakładanie dodatkowych kont zostało opisane w procesie inicjalizacji systemu. Przewidziana jest możliwość edycji oraz założenia nowych kont w menu systemowym.

2.2.1. Logowanie do systemu przez Internet

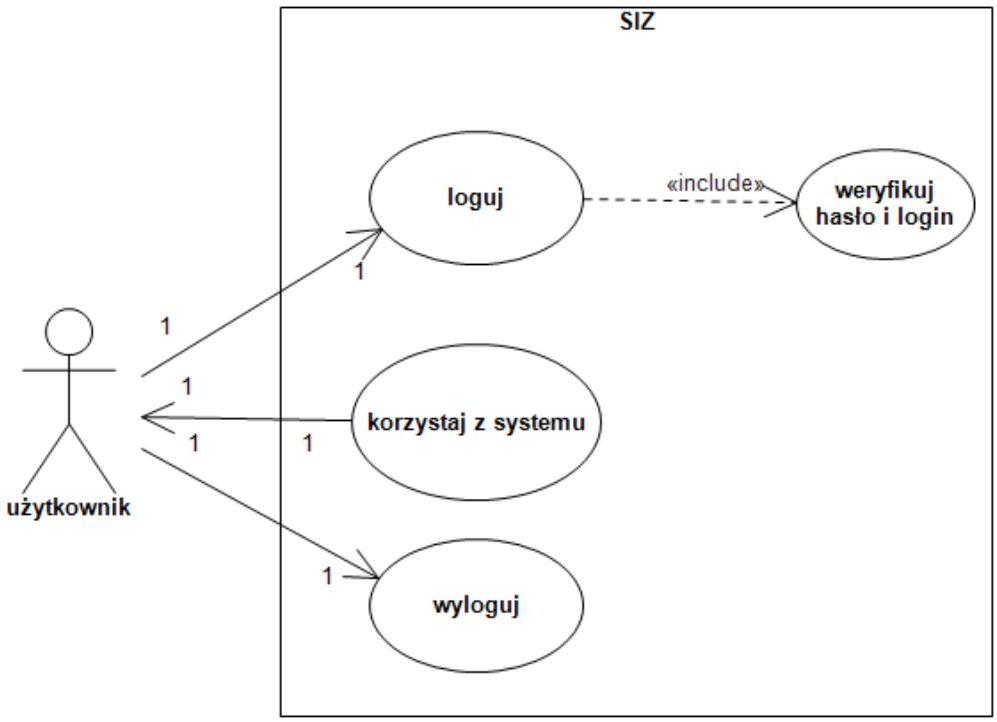
Procedura logowania do systemu polega na weryfikacji hasła oraz loginu. Po nieudanych trzech próbach zalogowania, konto zostaje zablokowane. Jeżeli weryfikacja przebiegnie pomyślnie, użytkownik otrzymuje dostęp do funkcji systemu w zakresie:

– monitorowania komunikatów wysyłanych przez czujniki,
– podglądu z zainstalowanych kamer,
– przeglądania raportów,
– planowania oraz śledzenia wykonywania harmonogramu,
– zarządzania odbiorami,
– monitorowania stanu naładowania baterii pojazdu elektrycznego (jeżeli jest podłączony do stacji ładującej),
– zarządzania wybranymi instalacjami,
– dokonywania zmian w ustawieniach systemu.

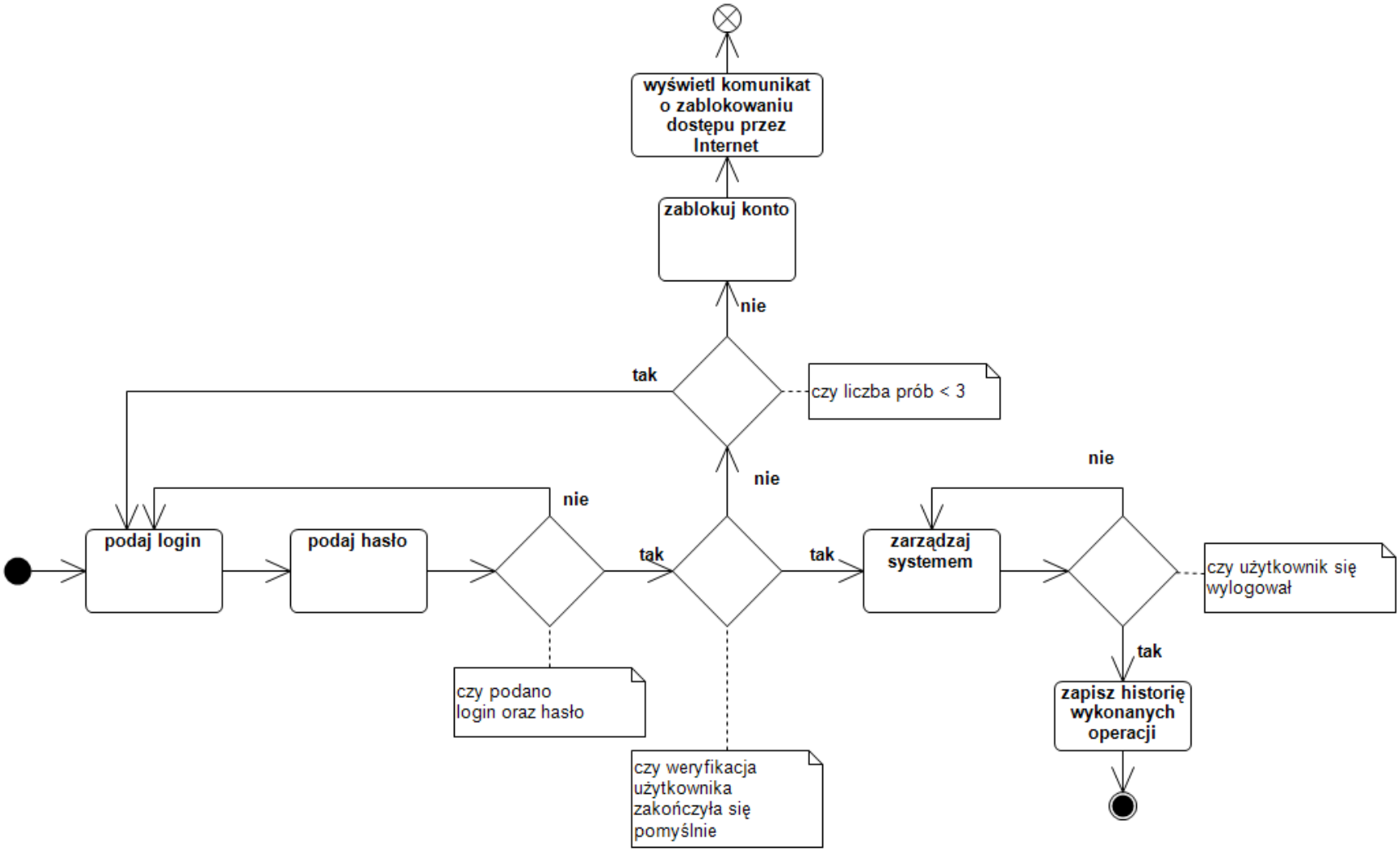
Wszystkie polecenia zlecone do systemu przez użytkownika korzystającego z dodatkowego konta posiadają mniejszy priorytet niż te pochodzące z domowego konta administratora. W ten sposób praca systemu będzie mniej podatna na błędy wynikające z równolegle podanych komend od różnych użytkowników. W przypadku gdy użytkownik konta dodatkowego próbuje skorzystać z zasobu zajętego w tym samym czasie przez użytkownika pracującego na koncie administratora, otrzymuje on dostęp w formie „tylko do odczytu" bez możliwości edycji.

Jeżeli dwóch użytkowników zaloguje się do systemu przez Internet wówczas posiadają oni te same uprawnienia i jako ostateczne zmiany zostaną zapisane te dokonane jako ostatnie. W przypadku gdy obydwoje próbują uzyskać dostęp do tego samego zasobu, użytkownik który zgłosił zapotrzebowanie później otrzymuje dostęp „tylko do odczytu".

Proces logowania został przedstawiony na dwóch poniższych diagramach.

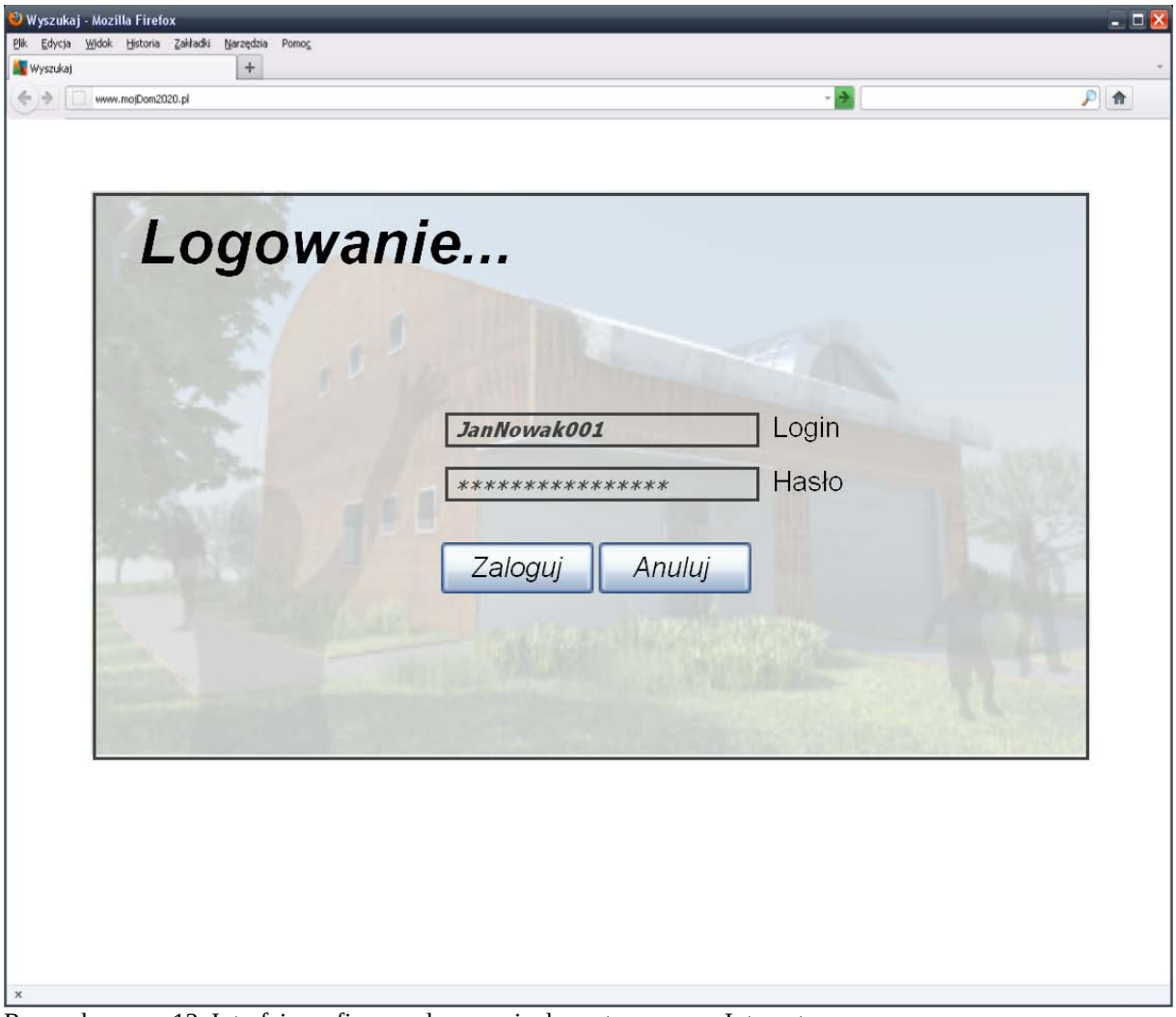


Rysunek numer 10. Diagram przypadków użycia – logowanie do systemu przez Internet.
Źródło: opracowanie własne.



Rysunek numer 11. Diagram czynności – logowanie do systemu przez Internet.
Źródło: opracowanie własne.

Proces logowania, wylogowania oraz dostępu do wybranych funkcji systemowych za pomocą Internetu prezentują interfejsy graficzne z rysunków numer 12, 13, 14, 15 oraz 16.

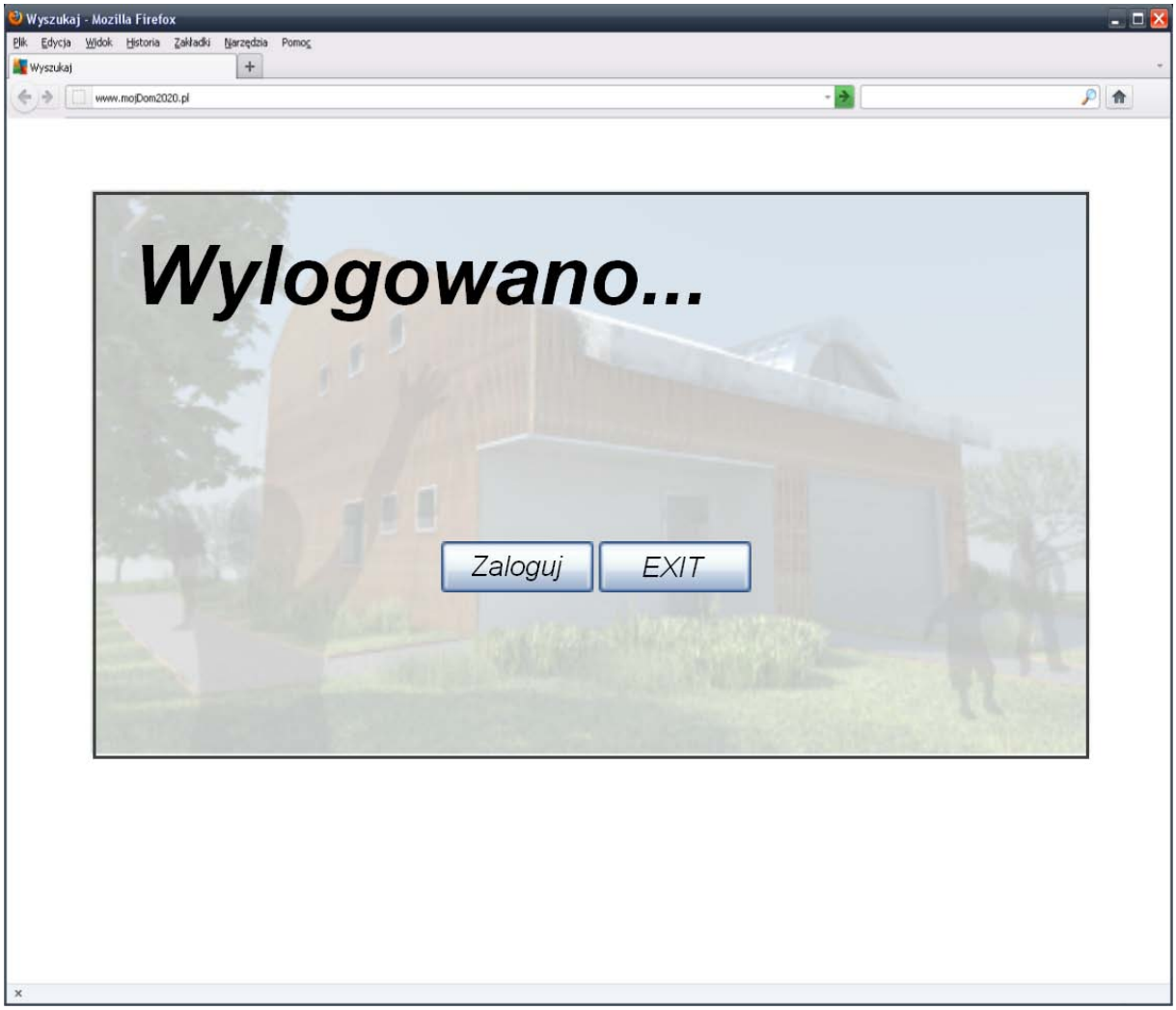


Rysunek numer 12. Interfejs graficzny – logowanie do systemu przez Internet.
Źródło: opracowanie własne.

Rysunek numer 13. Interfejs graficzny – dostęp do wybranych funkcji systemowych – raporty dobowe.
Źródło: opracowanie własne

Rysunek numer 14. Interfejs graficzny – dostęp do wybranych funkcji systemowych – optymalizacja.
Źródło: opracowanie własne

Rysunek numer 15. Interfejs graficzny – dostęp do wybranych funkcji systemowych – raporty miesięczne.
Źródło: opracowanie własne



Rysunek numer 16. Interfejs graficzny – wylogowanie.
Źródło: opracowanie własne

Dla procesu logowania została także sporządzona przykładowa implementacja za pomocą PHP[ii]. Wybrane jej fragmenty zaprezentowano poniżej.

Rejestracja nowego użytkownika:

```
<?php
/*
       Plik odpowiedzialny za formularz rejestracji nowego użytkownika
*/
include_once ("mojadb.inc");

//jesli nie kliknięto przycisku wyślij pokaż formularz rejestracji
if (empty ($_POST['wyslij']))
       {
?>
<head>
       <META HTTP-EQUIV="Content-type"CONTENT="text/html;charset=utf-8">
       <META HTTP-EQUIV="Content-Language"CONTENT="pl">
    <title>Zarejestruj się</title>
</head>

<form action="rejestracja.php" method="post">

<table>
<tr><td> Login: </td><td><input type="text" name="login" /> </td></tr>
<tr><td> Haslo: </td><td><input type="password" name="pass1" /> </td></tr>
<tr><td> Potwierdź hasło: </td><td><input type="password" name="pass2" /> </td></tr>
<tr><td> Imię</td><td><input type="text" name="imie" /> </td></tr>
<tr><td> Nazwisko: </td><td><input type="text" name="nazwisko" /> </td></tr>
<tr><td> E-mail: </td><td><input type="text" name="email" /> </td></tr>
```

```
<tr><td> <input type="submit" value="Wyślij" name="wyslij"/> </td>
<td> <input type="reset" value= "Wyczyść> </td></tr>
</table>

</form>


<?php
	}
//wysyłanie danych z formularza do bazy
else
	{
	$imie=$_POST['imie'];
	$nazwisko=$_POST['nazwisko'];
	$login=$_POST['login'];
	$pass1=$_POST['pass1'];
	$pass2=$_POST['pass2'];
	$haslo=md5($pass1);
	$email=$_POST['email'];

	//sprawdzenie czy wszystkie pola formularza zostaly wypełnione
	if   (empty($login)   ||   empty($pass1)   ||   empty($pass2)   ||   empty($imie)   ||
empty($nazwisko) || empty($email))
		{
		echo "Wypelnij wszystkie pola!";
		}
	else
		{
		//sprawdzenie poprawności hasła
		if ($pass1 != $pass2)
			{
			echo "Podane hasła nie są równoważne!";
			}
		else
			{
			$if_user_exist = "select login from users where login like '$login'";
			$result_if_user_exist = $if_user_exist, $con) or die (mysql_error());
			$result_if_user_exist = $db->num_rows($result_if_user_exist);
			$t=getdate();
			//wygenerowanie time_tokena
			$time_token = date('Y-m-j H:i:s',$t[0]+172800); //dzisiejsza data plus
24 godziny
			//$today=date('Y-m-j H:i:s',$t[0]);

			if ($result_if_user_exist == 0)
				{
				//dodanie konta użytkownika do bazy
				$numer_aktywacji= rand(1000000000,9999999999);
				$add_user = "insert into users (login, pass, imie, nazwisko,
email, aktywacja, time_token_activation) values ('$login', '$haslo', '$imie', '$nazwisko',
'$email', '$numer_aktywacji', '$time_token')";
				//wysłanie wiadomości email z linkiem do aktywacji konta
				mysql_query($add_user, $con) or die (mysql_error());
				$do  = "$imie $nazwisko <$email>";
				$temat = "Dane rejestracyjne serwisu Beens Inteligent Home!";
				$wiadomosc = "
					<html><head>
					<meta                          http-equiv=\"Content-Type\"
content=\"text/html;charset=utf-8\">
					<style type=\"text/css\">
					<!--
						body { font-family: arial, helvetica, sans-serif;
font-size: 10pt; color: black; background: white;}
						.header1  {  border-bottom:  2px  solid  #ffcc00;
width:  100%;  font-weight:  bold;  font-size:  11pt;  font-family:  verdana,  arial,  sans-serif;
white-space: nowrap; color: #515151;}
						.content1  {  margin-top:  0;  margin-bottom:  0;
width: 100%; background: white;}
						.content2  {  margin-top:  0;  margin-bottom:  0;
width: 100%; background: #f6f5f3; }
						td   {   font-size:   10pt;   vertical-align:   top;
padding: 3px; padding-top: 10px; }
						.footer1 { margin: 0; padding: 0; border-top: 1px
solid #ffcc00; width: 100%;}
						.footer2 { margin: 0; padding: 0; border-top: 1px
solid #d0d0d0; width: 100%;}
						.form  {  background:  white;  border-bottom:  1px
solid #c3c3c3;}
						.form1 {border-top:     1px     solid     #c3c3c3;
background: #eeeeee;}
						.form2  {border-top:  1px  solid  #c3c3c3;  border-
left: 1px solid #c3c3c3;}
					-->
					</style></head>
					<body>
					<div  class='header1'>Witaj  $imie!<br  /><br  />Gratulacje!
Dziękujemy za wybranie serwisu Beens Inteligent Home!</div><br />
```

```
                                        <table width='550' cellspacing='0' cellpadding='5'
class='form'>
                                        <tr>
                                                <td width='30%' class='form1'> </td>
                                                <td  class='form1'><b>Dane  podane  podczas
rejestracji</b></td>
                                        </tr>
                                        <tr>
                                                <td                              width='30%'
class='form1'><b>Imię</b> </td>
                                                <td class='form2'>$imie<br /></td>
                                        </tr>
                                        <tr>
                                                <td                              width='30%'
class='form1'><b>Nazwisko</b> </td>
                                                <td class='form2'>$nazwisko</br /></td>
                                        </tr>
                                        <tr>
                                                <td                              width='30%'
class='form1'><b>Login</b> </td>
                                                <td class='form2'>$login<br /></td>
                                        </tr>
                                        <tr>
                                                <td                              width='30%'
class='form1'><b>Hasło</b> </td>
                                                <td class='form2'>$pass1<br /></td>
                                        </tr>
                                        </table><br />
                                <div class='footer1'> </div>
                                <table  class='content1'  cellpadding='0'  cellspacing='5'
width='550'>
                                        <tr>
                                                <td width='550' class='content2'>
                                             <b>Aby  dokonać  aktywacji  konta  w  ciągu  48
godzin musisz kliknąć link poniżej:<br />
                                                <a
href='http://$web_adress/aktywacja.php?user=$login&numer_aktywacji=$numer_aktywacji'>Aktywacja
konta $login</a></b><br />
                                                        <br />
                                                        <b>Masz      czas      do      <font
color=red>$time_token</font></b>
                                                </td>
                                        </tr>
                                        </table>
                                <div class='header1'> </div>
                                <br />
                                <div>Pozdrawiamy,<br />Zespół Beens Inteligent Home<br />
                                <a href=mailto:admin@localhost>admin@localhost</a></div>
                                </body>
                                </html>
                        ";
                        $naglowki  = "MIME-Version: 1.0\r\n";
                        $naglowki   .=    "Content-type:    text/html;    charset=windows-
1250\r\n";
                        $naglowki    .=    "From:    Serwis    Beens    Inteligent    Home
<powiadomienie@localhost>\r\n";
                        //komunikat o powodzeniu wysłania email
                        if (mail($email, $temat, $wiadomosc, $naglowki))
                                {
                                echo "Gratulacje!  Na  adres  ".$email."  został  przesłany
email z danymi użytkownika oraz linkiem do aktywacji konta!";
                                }
                        else
                                {
                                $rollback_add_user    =    "DELETE    FROM    USERS    WHERE
login='$login';";
                                mysql_query($rollback_add_user,      $con)      or      die
(mysql_error());
                                exit("Błąd    podczas    wysyłania    maila    dotyczącego
rejestracji! Skontaktuj sie z <a href=mailto:admin@localhost>administratorem</a>");
                                }
                        }
                else
                        {
                        echo "Użytkownik o podanym loginie już istnieje! <br />
                        <a href=\"javascript:history.back();\">Powrot</a> ";
                        }
                }
        }

?>
```

Aktywacja konta:

```
<?php
/*
	Plik odpowiedzialny za aktywacje konta po otrzymaniu tokena z listu email
*/
?>

<head>
	<META HTTP-EQUIV="Content-type"CONTENT="text/html;charset=utf-8">
	<META HTTP-EQUIV="Content-Language"CONTENT="pl">
    <title>Aktywacja konta</title>
</head>
<?php
//dołączenie plików
include_once ("mojadb.inc");

if (empty ($_REQUEST['user']) || empty ($_REQUEST['numer_aktywacji'])) //sprawdzenie czy pola
user i numer_aktywacji zostały przekazane przez adres URL
	{
	echo	"Błąd	aktywacji	-	skontaktuj	się	z	<a
href=mailto:admin@localhost>administratorem</a> !!";
	}
else
	{
	//przepisanie wartośc i przekazanych przez link URL do zmiennych
	$login = $_REQUEST['user'];
	$numer_aktywacji = $_REQUEST['numer_aktywacji'];
	//sprawdzenie w bazie nazwy użytkownika
	$if_user_activated = "select login, aktywacja from users where lower(login) like
lower('$login') and aktywacja like 'YES'";
	$result_if_user_activated	=	mysql_query($if_user_activated,	$con)	or	die
(mysql_error());
	$result_if_user_activated = mysql_num_rows($result_if_user_activated);
	if ($result_if_user_activated == 0) //sprawdzenie czy uzytkownik czasami juz nie
dokonal aktywacji
		{
		$if_user_exist = "select login, aktywacja from users where lower(login) like
lower('$login') and aktywacja like '$numer_aktywacji'";
		$result_if_user_exist	=	mysql_query($if_user_exist,	$con)	or	die
(mysql_error());
		$result_if_user_exist = mysql_num_rows($result_if_user_exist);
		if ($result_if_user_exist == 0) //sprawdzenie czy uzytkownik w ogóle istnieje w
bazie lub jego numer aktywacji jest poprawny
			{
			echo "Błąd aktywacji!!</br>
				Brak podanego użytkownika w bazie lub zły  numer aktywacji </br>
				Skontaktuj	sie	z	<a
href=mailto:admin@localhost>administratorem</a> !!";
			}
		else
			{
			//pobranie time tokena użytkownika z bazy
			$time_token	=	"SELECT	time_token_activation	FROM	users	WHERE
lower(login) LIKE lower('$login')";
			$time_token = mysql_query($time_token, $con) or die (mysql_error());
			$time_token = mysql_fetch_row($time_token);
			$time_token = date('Y-m-j H:i:s',$time_token[0]);
			$t=getdate();
			$today=date('Y-m-j H:i:s',$t[0]);

			if ($time_token > $today) //sprawdzenie czy uzytkownik nie przekroczyl
czasu do aktywacji
				{
				echo "Minął czas na aktywację konta '$login', odczekaj 4 godziny
i	wypełnij	ponownie	formularz	rejestracji	pamiętając	później,	że	na	aktywację	masz	48
godzin.";
				}
			else
				{
				//aktywacja konta użytkownika
				$user_activation	=	"UPDATE	users	SET	aktywacja	=	'YES',
date_activation = '$today' WHERE login like '$login';";
				$result_user_activation = mysql_query($user_activation, $con) or
die (mysql_error());

				if ($result_user_activation == 1) //sprawdzenie czy uzytkownik
zostal poprawnie zaaktywowany
					{
					echo	"Użytkownik	<b>$login</b>	został	poprawnie
zaaktywowany! </br>
						<a
href='http://$web_adress/logowanie.php'><b>$login</b>,	przejdź	do	strony	logowania!</a>
</br>";
					}
				else
```

```
                                        {
                                        echo "Błąd połączenia z bazą podczas aktywacji!!</br>
                                                Skontaktuj          się          z          <a
href=mailto:admin@localhost>administratorem</a> !!";
                                        }
                                }
                        }
                }
        else
                {
                echo "$login, aktywacja została już dokonana!!!";
                }
        }
?>
```

Logowanie użytkownika i tworzenie dla niego sesji:

```
<?php
/*
        Plik odpowiedzialny za zalogowanie użytkownika i utworzenie dla niego sesji
*/
header('Content-Type: text/html; charset=UTF-8');
session_start();
include_once ("mojadb.inc");

//jeśli użytkownik nie jest zalogowany pokaż formularz logowania oraz link do utworzenia
nowego konta
if (empty ($_POST['zaloguj']))
        {
        ?>
<html><head><meta http-equiv="content-type" content="text/html; charset=UTF-8"></meta></head>

        <form action="logowanie.php" method="post">

        <table>
        <tr><td>Login:</td><td><input type="text" name="login" /></td></tr>
        <tr><td>Haslo:</td><td><input type="password" name="pass" /></td></tr>
        <tr><td><input type="submit" value="Zaloguj" name="zaloguj"/></td>
        <td><input type="reset" value= "Wyczyść"/></td></tr>
        </table>
        <?php    print    "<a    href=http://$web_adress/rejestracja.php>Rejestracja    nowego
użytkownika</a>" ?>
        </form>
        </html>
        <?php
        }
else
        {
        //przepisanie wartości z linku URL do zmiennych
        $_SESSION['user'] = $_POST['login'];
        $login=$_POST['login'];
        $pass=$_POST['pass'];
        if (empty($login) || empty($pass))
                {
                header('location: logowanie.php');
                }
        else
                {
                //sprawdzenie czy użytkownik znajduje się w bazie
                $if_user_exist = "select login from users where login like '$login'";
                $result_if_user_exist    =    mysql_query($if_user_exist,    $con)    or    die
(mysql_error());
                $result_if_user_exist = mysql_num_rows($result_if_user_exist);

                if ($result_if_user_exist == 0) //użytkownika nie ma w bazie
                        {
                        echo "Nie ma takiego użytkownika w bazie serwisu lub wystąpił błąd
hasła!<br />
                        <a href=http://$web_adress/logowanie.php>Spróbuj ponownie</a>";
                        }
                else
                        {
                        //sprawdzenie czy użytkownik dokonał aktywacji konta
                        $if_user_activated = "select login, aktywacja from users where login
like '$login' and aktywacja like 'YES'";
                        $result_if_user_activated = mysql_query($if_user_activated, $con) or die
(mysql_error());
                        $result_if_user_activated = mysql_num_rows($result_if_user_activated);
                        //brak aktywacji
                        if ($result_if_user_activated == 0)
                                {
                                echo "$login, aktywacja nie została przeprowadzona!!!<br />
```

```
                                        Sprawdź mail jaki został wysłany do Ciebie z serwisu
Beens Inteligent Home po Twojej rejestracji, tam znajdują się informacje o aktywacji konta.";
                                }
                        //aktywacja ok - przekierowanie na stronę główną
                        else
                                {
                                header('location:  homepage.php');
                                }
                        }
                }
        }
?>
```

Skrypt bazy dla konta użytkownika:

```
-- Skrypt bazy dla konta użytkowników
--
-- phpMyAdmin SQL Dump
-- version 2.8.2.4
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Wersja serwera: 5.0.24
-- Wersja PHP: 5.1.6
--
-- Baza danych: `mojadb`
--

-- --------------------------------------------------------

--
-- Struktura tabeli dla  `users`
--

CREATE TABLE `users` (
  `ID` int(4) NOT NULL auto_increment,
  `login` varchar(20) character set latin1 NOT NULL,
  `pass` varchar(30) character set latin1 NOT NULL,
  `imie` varchar(20) character set latin1 NOT NULL,
  `nazwisko` varchar(20) character set latin1 NOT NULL,
  `email` varchar(30) character set latin1 NOT NULL,
  `aktywacja` varchar(20) character set latin1 NOT NULL,
  `date_activation` datetime default NULL,
  `time_token_activation` datetime NOT NULL,
  PRIMARY KEY  (`ID`),
  KEY `ID` (`ID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=42 ;

--
-- Zrzut danych tabeli `users`
--
-- przykładowy użytkownik
INSERT   INTO   `users`   (`login`,   `pass`,   `imie`,   `nazwisko`,   `email`,   `aktywacja`,
`date_activation`, `time_token_activation`) VALUES ('user', '0cc175b9c0f1b6a831c399e2697726',
'Jan', 'Kowalski', 'admin@localhost', 'YES', '2011-11-25 21:51:59', '2011-11-29 20:51:49');
```

Biblioteka do obsługi bazy:

```
<?php
/*
       Biblioteka do obsługi bazy
*/

class DB_Sql {

  /* public: connection parameters */
  var $Host     = "";
  var $Database = "";
  var $User     = "";
  var $Password = "";

  /* public: configuration parameters */
  var $Auto_Free     = 0;     ## Set to 1 for automatic mysql_free_result()
  var $Debug         = 0;     ## Set to 1 for debugging messages.
  var $Halt_On_Error = "yes"; ## "yes" (halt with message), "no" (ignore errors quietly),
"report" (ignore errror, but spit a warning)
  var $PConnect      = 0;     ## Set to 1 to use persistent database connections
  var $Seq_Table     = "db_sequence";

  /* public: result array and current row number */
  var $Record   = array();
```

```
var $Row;

/* public: current error number and error text */
var $Errno    = 0;
var $Error    = "";

/* public: this is an api revision, not a CVS revision. */
var $type     = "mysql";
var $revision = "1.2";

/* private: link and query handles */
var $Link_ID  = 0;
var $Query_ID = 0;

var $locked   = false;      ## set to true while we have a lock

/* public: constructor */
function DB_Sql($query = "") {
    $this->query($query);
}

/* public: some trivial reporting */
function link_id() {
  return $this->Link_ID;
}

function query_id() {
  return $this->Query_ID;
}

/* public: connection management */
function connect($Database = "", $Host = "", $User = "", $Password = "") {
  /* Handle defaults */
  if ("" == $Database)
    $Database = $this->Database;
  if ("" == $Host)
    $Host     = $this->Host;
  if ("" == $User)
    $User     = $this->User;
  if ("" == $Password)
    $Password = $this->Password;

  /* establish connection, select database */
  if ( 0 == $this->Link_ID ) {

    if(!$this->PConnect) {
      $this->Link_ID = mysql_connect($Host, $User, $Password);
    } else {
      $this->Link_ID = mysql_pconnect($Host, $User, $Password);
    }
    if (!$this->Link_ID) {
      $this->halt("connect($Host, $User, \$Password) failed.");
      return 0;
    }

    if (!@mysql_select_db($Database,$this->Link_ID)) {
      $this->halt("cannot use database ".$Database);
      return 0;
    }
  }

  return $this->Link_ID;
}

/* public: discard the query result */
function free() {
    @mysql_free_result($this->Query_ID);
    $this->Query_ID = 0;
}

/* public: perform a query */
function query($Query_String) {
  /* No empty queries, please, since PHP4 chokes on them. */
  if ($Query_String == "")
    /* The empty query string is passed on from the constructor,
     * when calling the class without a query, e.g. in situations
     * like these: '$db = new DB_Sql_Subclass;'
     */
    return 0;

  if (!$this->connect()) {
    return 0; /* we already complained in connect() about that. */
  };

  # New query, discard previous result.
  if ($this->Query_ID) {
    $this->free();
```

```
  }

  if ($this->Debug)
    printf("Debug: query = %s<br>\n", $Query_String);

  $this->Query_ID = @mysql_query($Query_String,$this->Link_ID);
  $this->Row   = 0;
  $this->Errno = mysql_errno();
  $this->Error = mysql_error();
  if (!$this->Query_ID) {
    $this->halt("Invalid SQL: ".$Query_String);
  }

  # Will return nada if it fails. That's fine.
  return $this->Query_ID;
}

/* public: walk result set */
function next_record() {
  if (!$this->Query_ID) {
    $this->halt("next_record called with no query pending.");
    return 0;
  }

  $this->Record = @mysql_fetch_array($this->Query_ID);
  $this->Row   += 1;
  $this->Errno  = mysql_errno();
  $this->Error  = mysql_error();

  $stat = is_array($this->Record);
  if (!$stat && $this->Auto_Free) {
    $this->free();
  }
  return $stat;
}

/* public: position in result set */
function seek($pos = 0) {
  $status = @mysql_data_seek($this->Query_ID, $pos);
  if ($status)
    $this->Row = $pos;
  else {
    $this->halt("seek($pos) failed: result has ".$this->num_rows()." rows.");

    /* half assed attempt to save the day,
     * but do not consider this documented or even
     * desireable behaviour.
     */
    @mysql_data_seek($this->Query_ID, $this->num_rows());
    $this->Row = $this->num_rows();
    return 0;
  }

  return 1;
}

/* public: table locking */
function lock($table, $mode = "write") {
  $query = "lock tables ";
  if(is_array($table)) {
    while(list($key,$value) = each($table)) {
      // text keys are "read", "read local", "write", "low priority write"
      if(is_int($key)) $key = $mode;
      if(strpos($value, ",")) {
        $query .= str_replace(",", " $key, ", $value) . " $key, ";
      } else {
        $query .= "$value $key, ";
      }
    }
    $query = substr($query, 0, -2);
  } elseif(strpos($table, ",")) {
    $query .= str_replace(",", " $mode, ", $table) . " $mode";
  } else {
    $query .= "$table $mode";
  }
  if(!$this->query($query)) {
    $this->halt("lock() failed.");
    return false;
  }
  $this->locked = true;
  return true;
}

function unlock() {

  // set before unlock to avoid potential loop
  $this->locked = false;
```

```
  if(!$this->query("unlock tables")) {
    $this->halt("unlock() failed.");
    return false;
  }
  return true;
}

/* public: evaluate the result (size, width) */
function affected_rows() {
  return @mysql_affected_rows($this->Link_ID);
}

function num_rows() {
  return @mysql_num_rows($this->Query_ID);
}

function num_fields() {
  return @mysql_num_fields($this->Query_ID);
}

/* public: shorthand notation */
function nf() {
  return $this->num_rows();
}

function np() {
  print $this->num_rows();
}

function f($Name) {
  if (isset($this->Record[$Name])) {
    return $this->Record[$Name];
  }
}

function p($Name) {
  if (isset($this->Record[$Name])) {
    print $this->Record[$Name];
  }
}

/* public: sequence numbers */
function nextid($seq_name) {
  /* if no current lock, lock sequence table */
  if(!$this->locked) {
    if($this->lock($this->Seq_Table)) {
      $locked = true;
    } else {
      $this->halt("cannot lock ".$this->Seq_Table." - has it been created?");
      return 0;
    }
  }

  /* get sequence number and increment */
  $q = sprintf("select nextid from %s where seq_name = '%s'",
            $this->Seq_Table,
            $seq_name);
  if(!$this->query($q)) {
    $this->halt('query failed in nextid: '.$q);
    return 0;
  }

  /* No current value, make one */
  if(!$this->next_record()) {
    $currentid = 0;
    $q = sprintf("insert into %s values('%s', %s)",
              $this->Seq_Table,
              $seq_name,
              $currentid);
    if(!$this->query($q)) {
      $this->halt('query failed in nextid: '.$q);
      return 0;
    }
  } else {
    $currentid = $this->f("nextid");
  }
  $nextid = $currentid + 1;
  $q = sprintf("update %s set nextid = '%s' where seq_name = '%s'",
            $this->Seq_Table,
            $nextid,
            $seq_name);
  if(!$this->query($q)) {
    $this->halt('query failed in nextid: '.$q);
    return 0;
  }
```

```
  /* if nextid() locked the sequence table, unlock it */
  if($locked) {
    $this->unlock();
  }

  return $nextid;
}

/* public: return table metadata */
function metadata($table = "", $full = false) {
  $count = 0;
  $id    = 0;
  $res   = array();

  /*
   * Due to compatibility problems with Table we changed the behavior
   * of metadata();
   * depending on $full, metadata returns the following values:
   *
   * - full is false (default):
   * $result[]:
   *   [0]["table"]  table name
   *   [0]["name"]   field name
   *   [0]["type"]   field type
   *   [0]["len"]    field length
   *   [0]["flags"]  field flags
   *
   * - full is true
   * $result[]:
   *   ["num_fields"] number of metadata records
   *   [0]["table"]  table name
   *   [0]["name"]   field name
   *   [0]["type"]   field type
   *   [0]["len"]    field length
   *   [0]["flags"]  field flags
   *   ["meta"][field name]  index of field named "field name"
   *   This last one could be used if you have a field name, but no index.
   *   Test:  if (isset($result['meta']['myfield'])) { ...
   */

  // if no $table specified, assume that we are working with a query
  // result
  if ($table) {
    $this->connect();
    $id = @mysql_list_fields($this->Database, $table);
    if (!$id) {
      $this->halt("Metadata query failed.");
      return false;
    }
  } else {
    $id = $this->Query_ID;
    if (!$id) {
      $this->halt("No query specified.");
      return false;
    }
  }

  $count = @mysql_num_fields($id);

  // made this IF due to performance (one if is faster than $count if's)
  if (!$full) {
    for ($i=0; $i<$count; $i++) {
      $res[$i]["table"] = @mysql_field_table ($id, $i);
      $res[$i]["name"]  = @mysql_field_name  ($id, $i);
      $res[$i]["type"]  = @mysql_field_type  ($id, $i);
      $res[$i]["len"]   = @mysql_field_len   ($id, $i);
      $res[$i]["flags"] = @mysql_field_flags ($id, $i);
    }
  } else { // full
    $res["num_fields"]= $count;

    for ($i=0; $i<$count; $i++) {
      $res[$i]["table"] = @mysql_field_table ($id, $i);
      $res[$i]["name"]  = @mysql_field_name  ($id, $i);
      $res[$i]["type"]  = @mysql_field_type  ($id, $i);
      $res[$i]["len"]   = @mysql_field_len   ($id, $i);
      $res[$i]["flags"] = @mysql_field_flags ($id, $i);
      $res["meta"][$res[$i]["name"]] = $i;
    }
  }

  // free the result only if we were called on a table
  if ($table) {
    @mysql_free_result($id);
  }
  return $res;
}
```

```
  /* public: find available table names */
  function table_names() {
    $this->connect();
    $h = @mysql_query("show tables", $this->Link_ID);
    $i = 0;
    while ($info = @mysql_fetch_row($h)) {
      $return[$i]["table_name"]      = $info[0];
      $return[$i]["tablespace_name"] = $this->Database;
      $return[$i]["database"]        = $this->Database;
      $i++;
    }

    @mysql_free_result($h);
    return $return;
  }

  /* private: error handling */
  function halt($msg) {
    $this->Error = @mysql_error($this->Link_ID);
    $this->Errno = @mysql_errno($this->Link_ID);

    if ($this->locked) {
      $this->unlock();
    }

    if ($this->Halt_On_Error == "no")
      return;

    $this->haltmsg($msg);

    if ($this->Halt_On_Error != "report")
      die("Session halted.");
  }

  function haltmsg($msg) {
    printf("</td></tr></table><b>Database error:</b> %s<br>\n", $msg);
    printf("<b>MySQL Error</b>: %s (%s)<br>\n",
      $this->Errno,
      $this->Error);
  }

}
?>
```

Plik przechowujący parametry do połączenia z bazą:

```
<?php
/*
       Plik przechowujący parametry do połaczenia z bazą
*/
//include_once("db_mysql.inc");
$con = mysql_connect('localhost','root','root');
mysql_select_db('mojadb',$con);
$web_adress = "localhost/beens_inteligent_home";
$path_to_temp = "c:/wamp/tmp\\";
$path_images = "images/";
/*
class mydb extends DB_Sql{

var $Host = "localhost";
var $Database = "mojadb";
var $User = "root";
var $Password = "root";
}
*/
?>
```

Wyświetlanie strony głównej po zalogowaniu:

```
<?php
//strona główna
header('Content-Type: text/html; charset=UTF-8');
session_start();
include_once ("mojadb.inc");

//sprawdzneie czy użytkownik jest zalogowany i ma uprawnienia
if (empty($_SESSION['user']))
       {
       echo "<a href=http://$web_adress/logowanie.php>Strona logowania</a>";
       }
```

```
else
	{
//wyświetlenei strony głównej z menu
print "
	<head>
		<META HTTP-EQUIV='Content-type'CONTENT='text/html;charset=utf-8'>
		<META HTTP-EQUIV='Content-Language'CONTENT='pl'>
	    <title>Serwis Beens Inteligent Home!</title>
	</head>
	";
print "
	<table width=100% height=100%>
	";
$user = $_SESSION['user'];
print "
	Witaj $user!!</br>
	<a href=http://$web_adress/control_hot.php>Sterowanie ogrzewaniem</a></br>
	<a href=http://$web_adress/control_water.php>Sterowanie gospodarką wodną</a></br>
	<a href=http://$web_adress/statistics.php>Statystyki</a></br>
	<a href=http://$web_adress/rejestracja.php>Dodaj użytkownika</a></br>
	<a href=http://$web_adress/logout.php>Wyloguj</a></br>
	";
	}
?>
```

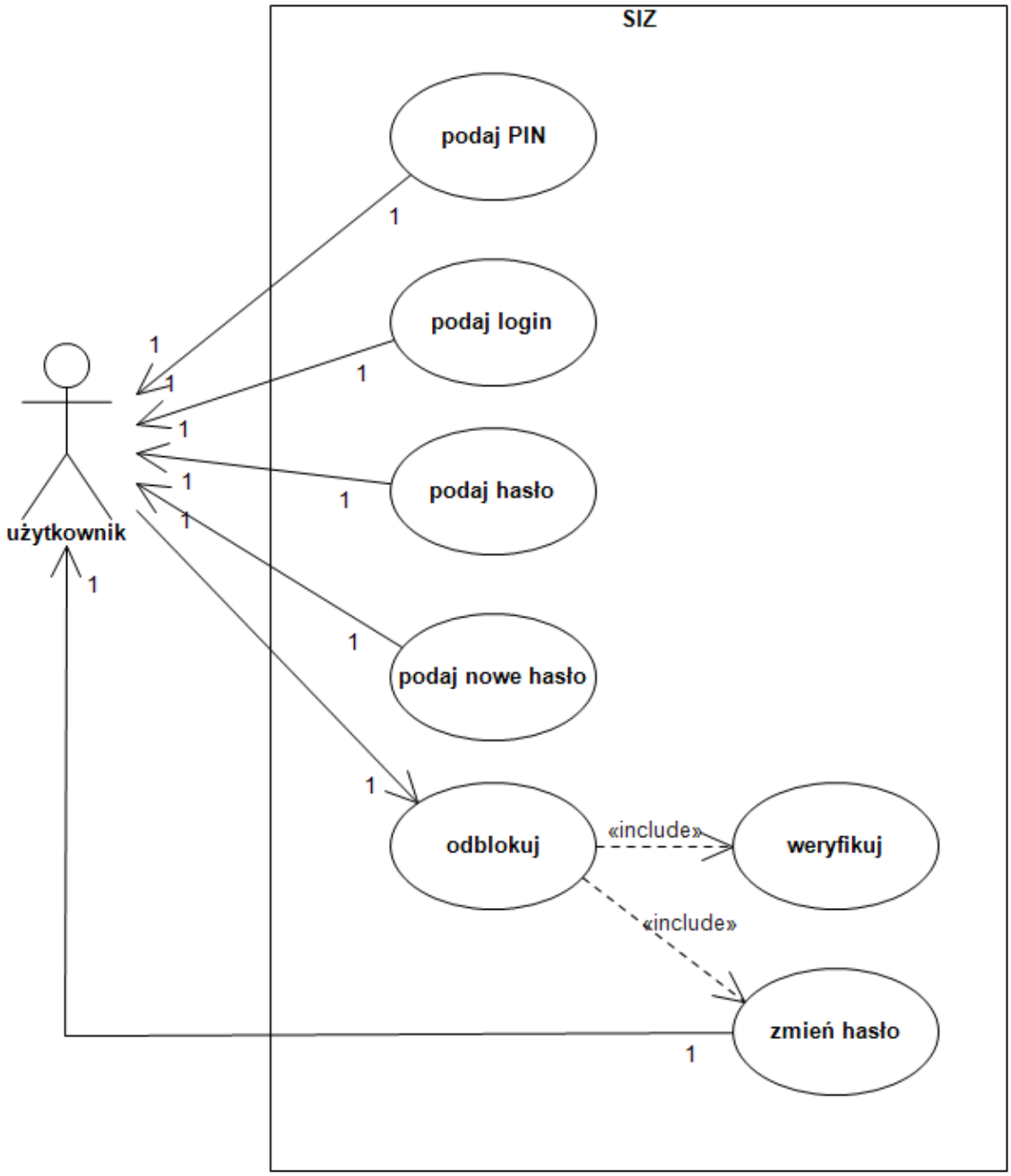
Wylogowanie użytkownika:

```
<?php
/*
	Plik odpowiedzialny za wylogowanie użytkownika i usunięciu jego sesji
*/
include_once ("mojadb.inc");
header('Content-Type: text/html; charset=UTF-8');
session_start();
if (empty($_SESSION['user']))
	{
	echo "<a href=http://$web_adress/logowanie.php>Strona logowania</a>";
	}
else
	{
	$user = $_SESSION['user'];
	unset($_SESSION['user']);
	session_destroy();
	echo "<b>$user</b>, został wylogowany!</br>";
	echo "<a href=http://$web_adress/logowanie.php>Strona logowania</a>";
	}
?>
```

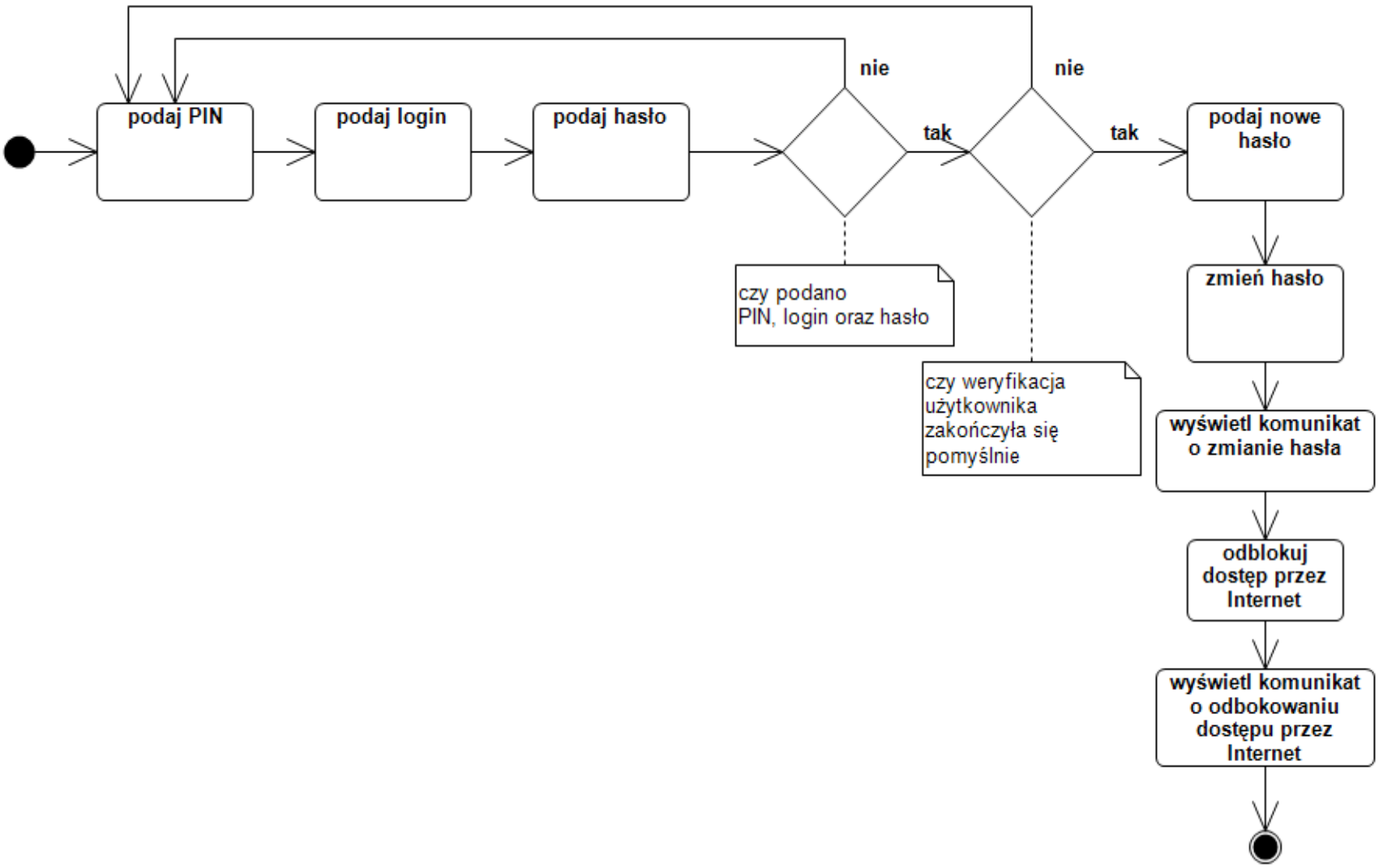
2.2.2. Odblokowanie konta użytkownika

Odblokowanie konta może odbyć się jedynie z konta administratora. Użytkownik podaje hasło administratora a następnie login i hasło do zablokowanego konta. System wymusza kompletne podanie tych danych. Jeżeli weryfikacja hasła administratora oraz loginu użytkownika została zakończona pomyślnie, użytkownik podaje nowe hasło do zablokowanego konta a następnie potwierdza je w sposób analogiczny jak przy jego zakładaniu. Po zmianie hasła system odblokowuje konto, informując o tym za pomocą wyświetlonego komunikatu.

Rysunki numer 17 oraz 18 ilustrują opisaną procedurę, natomiast rysunek numer 19 interfejs graficzny.



Rysunek numer 17. Diagram przypadków użycia – odblokowywanie konta użytkownika.
Źródło: opracowanie własne



Rysunek numer 18. Diagram czynności – odblokowywanie konta użytkownika.
Źródło: opracowanie własne.

Rysunek numer 19. Interfejs graficzny – odblokowywanie konta użytkownika, zmiana hasła.
Źródło: opracowanie własne.

LITERATURA

1. Lugaric L., Krajcar S., Simic Z., " Smart City - Platform for Emergent Phenomena Power System Testbed Simulator", IEEE PES Conference on Innovative Smart Grid Technologies Europe, October 11-13, 2010, 2048017
2. Raport Komisji Europejskiej, „ICT for a Low Carbon Economy. Smart Buildings.", Bruksela, lipiec 2009
3. Rekomendacje Komisji Wspólnot Europejskich „Commission Recommendation of 9.10.2009 on mobilising Information and Communications Technologies to facilitate the transition to an energy-efficient, low-carbon economy", Bruksela, październik, 2009
4. Raport Komisji Europejskiej „Impacts of Information and Communication Technologies of Energy Efficiency", Bruksela, wrzesień 2008
5. "Demand Response as a resource for the adequacy and operational reliability of the power systems. Explanatory Note". ETSO, 2007.
6. "Demand Side Response in the National Electricity Market. Case Studies." Energy Users Association of Australia.
7. "Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them." US Departament of Energy
8. Nieuwenhout F., "Flexible electricity grids", Report of Work Package 1, EOS-LT project FLEXIBEL.
9. "Enhancement of Demand Response. FINAL STATUS REPORT", Nordel Demand Response Group, 2006
10. Goldberg M., "Measure Twice, Cut Once", IEEE Power & energy magazine, May/june 2010
11. Brooks A., Lu E., Reicher D., Spirakis Ch., Weihl B., "Demand Dispatch", IEEE Power & energy magazine, May/june 2010
12. Opracowanie modelu stosowania mechanizmów DSR na rynku energii w Polsce, wykonane na zlecenie PSE Operator S.A., Konstancin-Jeziorna 2009
13. Lui T.J., Stirling W., Marcy H.O., "Get smart", IEEE Power & energy magazine, May/june 2010
14. "SMART 2020: Enabling the low carbon economy in the information age.", The Climate Group, 2008
15. Wrycza St., Marcinkowski B., Wyrzykowski K., „Język UML 2.0 w modelowaniu systemów informatycznych", wydawnictwo Helion, 2005
16. Jabłońska M.R., "*Rola informatyki w budownictwie energooszczędnym*", publikacja powstała w ramach projektu "Bioenergia dla Regionu - Zintegrowany Program Rozwoju Doktorantów", www.bioenergiadlaregionu.eu, Łódź 2011
17. Jabłońska M.R., Zieliński J.S. "*Electric vehicles' influence on smart grids*", "Aktualne problemy w elektroenergetyce 2011", tom II, Jurata 08-10.06.2011
18. Jabłońska M.R., "*Aktualne trendy w badaniach nad reakcją strony popytowej oraz możliwości ich implementacji w warunkach krajowych*", Rynek Energii 3(94)/2011, wydawnictwo KAPRINT, Lublin 2011
19. Jabłońska M.R., Adrian Ł., Janicki M., Klimek A., Pawlak J., Tkacz E., Znajdek K., "*Dom 2020 - projekt niezależnego energetycznie, inteligentnego domu energooszczędnego*", współautorzy:, "Dolnośląski Dom Energooszczędny", Wrocław 2011
20. Turker H., Bacha S., Chatroux D.: Impact of Plug-in Hybrid Electric Vehicles (PHEVS) on the French Electric Grid. SG 2048068
21. Aggeler D., Canales F., Zelaya H., Coccia A., Butcher N., Apeldoorn O.: Ultra-Fast DC-Charge Infrastructures for EV-Mobility and Future Smart Grids SG 2006809

22. ICT for Breakthrough Industry Transformation. ICT for a Low Carbon Economy Smart Electricity Distribution Networks. European Commission Information Society and Media, July 2009, 36-39.

[i] Opracowanie interfejsów: Marta R. Jabłońska, Jakub Kusztelak
[ii] Opracowanie fragmentów kodu: Jakub Kusztelak