**BIURO NIERUCHOMOŚĆI**

**TESTY JEDNOSTKOWE**

**Tworzenie poziomu dostępu z określonego zbioru wartosci**

**Sprawdzanie warunków dodania poziomów dostępu**

/\*\*

\* Test of stworzPoziomDostepu method, of class PoziomDostepuManager.

\*/

@Test

public void testStworzPoziomDostepu() throws Exception {

System.out.println("stworzPoziomDostepu");

String poziom = "AGENT";

PoziomDostepuManager instance = new PoziomDostepuManager();

PoziomDostepu expResult = new PoziomDostepu();

expResult.setPoziom(poziom);

PoziomDostepu result = instance.stworzPoziomDostepu(poziom);

assertEquals(expResult, result);

}

/\*\*

\* Test of czyMoznaDodacPoziom method, of class PoziomDostepuManager.

\*/

@Test

public void testCzyMoznaDodacPoziom() {

KontoFacade kontoFacade=new KontoFacade();

System.out.println("czyMoznaDodacPoziom!");

Konto konto = kontoFacade.znajdzPoLoginie("agent");

String nowyPoziom = "MENADZER";

PoziomDostepuManager instance = new PoziomDostepuManager();

boolean expResult = TRUE;

boolean result = instance.czyMoznaDodacPoziom(konto, nowyPoziom);

assertEquals(expResult, result);

}

**Wyniki testu:**

czyMoznaDodacPoziom!

stworzPoziomDostepu

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

**GENEROWANIE skrótu MD5**

/\*\*

\* Test of generateMD5Hash method, of class MD5Generator.

\*/

@Test

public void testGenerateMD5Hash() throws Exception {

System.out.println("generateMD5Hash");

String haslo = "zaq1@WSX";

String expResult = "9e38e8d688743e0d07d669a1fcbcd35b";

String result = MD5Generator.generateMD5Hash(haslo);

assertEquals(expResult, result);

}

**Wynik testu:**

Running pl.lodz.p.it.ssbd2016.ssbd01.mok.utils.MD5GeneratorTest

generateMD5Hash

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

**Głęboka kopia obiektu wykorzystując serializację**

/\*\*

\* Test of deepCloneThroughSerialization method, of class CloneUtils.

\*/

@Test

public void testDeepCloneThroughSerialization() throws Exception {

System.out.println("deepCloneThroughSerialization");

Serializable source = new Konto(1L);

Object result = CloneUtils.deepCloneThroughSerialization(source);

assertNotSame(source, result);

}

**Wynik testu:**

Running pl.lodz.p.it.ssbd2016.ssbd01.Utils.CloneUtilsTest

deepCloneThroughSerialization

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

**Pobieranie wiadomosci ze słownika**

/\*\*

\* Test of getMessage method, of class MessageProvider.

\*/

@Test

public void testGetMessage() {

System.out.println("getMessage");

String key = "notyfikacja.rejestracja.tytul";

String expResult = "Konto zarejestrowane";

String result = MessageProvider.getMessage(key);

assertEquals(expResult, result);

}

**Wynik testu:**

Running pl.lodz.p.it.ssbd2016.ssbd01.Utils.MessageProviderTest

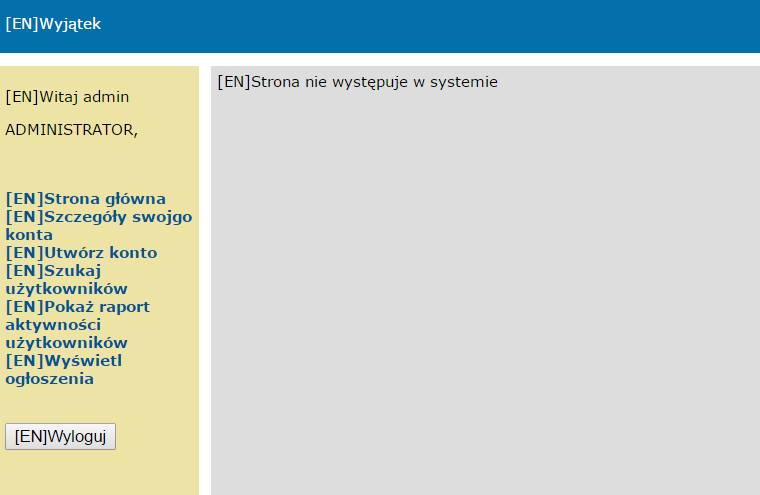
getMessage

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

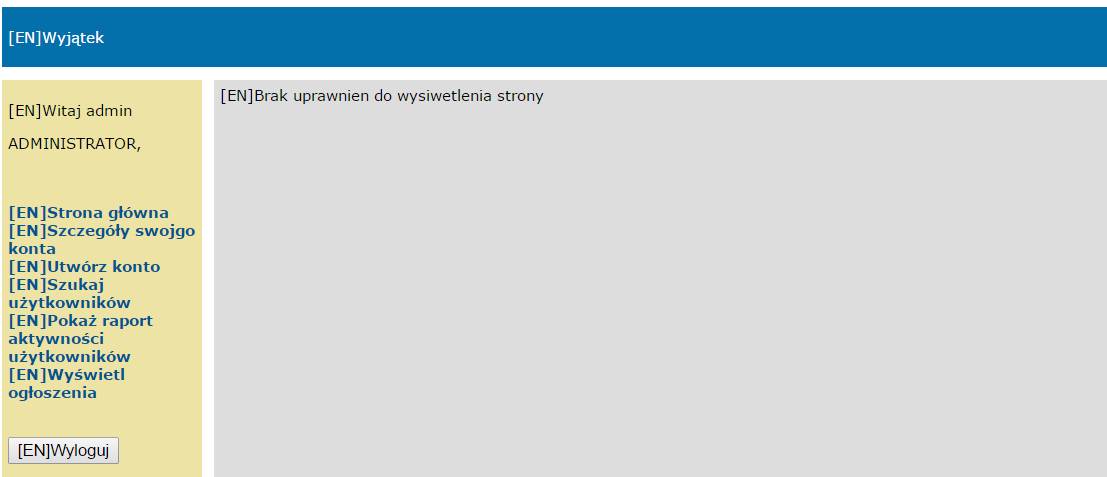
**TESTY BEZPIECZEŃSTWA**

**Przejście na stronę nieistniejącą w systemie**

****

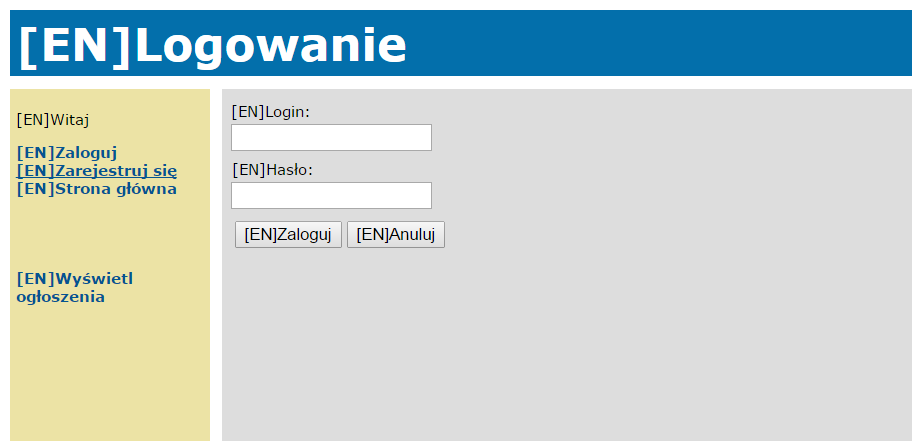
**Test pozytywny.**

**Przejście na stronę niedostępną dla administratora**

****

**Test pozytywny.**

**Przejście na stronę niedostępną dla gościa**

****

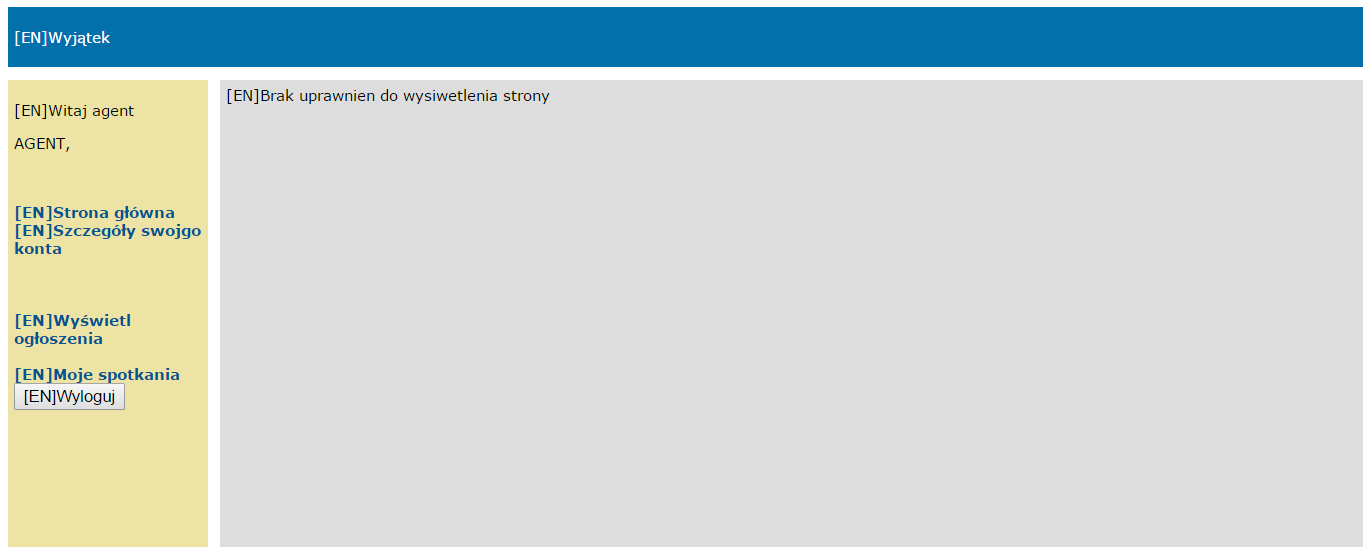
**Test pozytywny.**

**Przejście na stronę utworzenia konta przez użytkownika z poziomem dostępu agent**

****

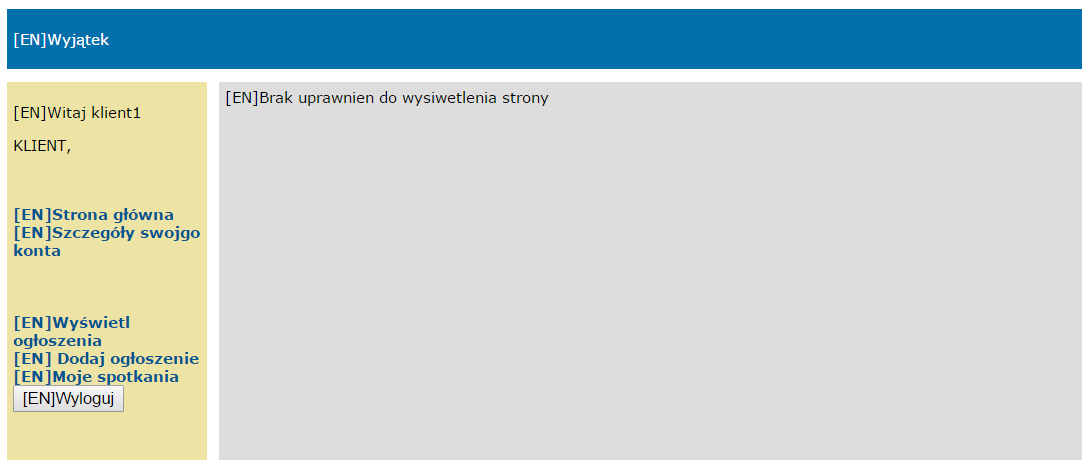
**Test pozytywny.**

**Przejście na stronę edycji konta użytkownika przez użytkownika z poziomem dostępu agent**

****

**Test pozytywny.**

**Przejście na stronę dostępną dla admina**

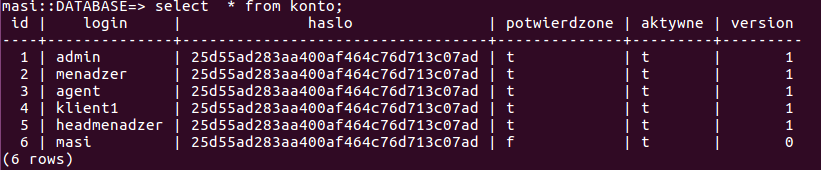
****

**Test pozytywny.**

**TESTY AKCEPTACYJNE**

**Rejestracja użytkownika w systemie**

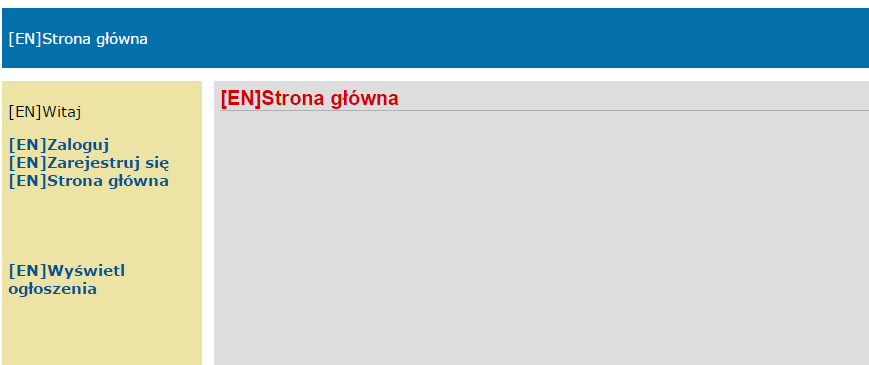
****

****

**Test pozytywny.**

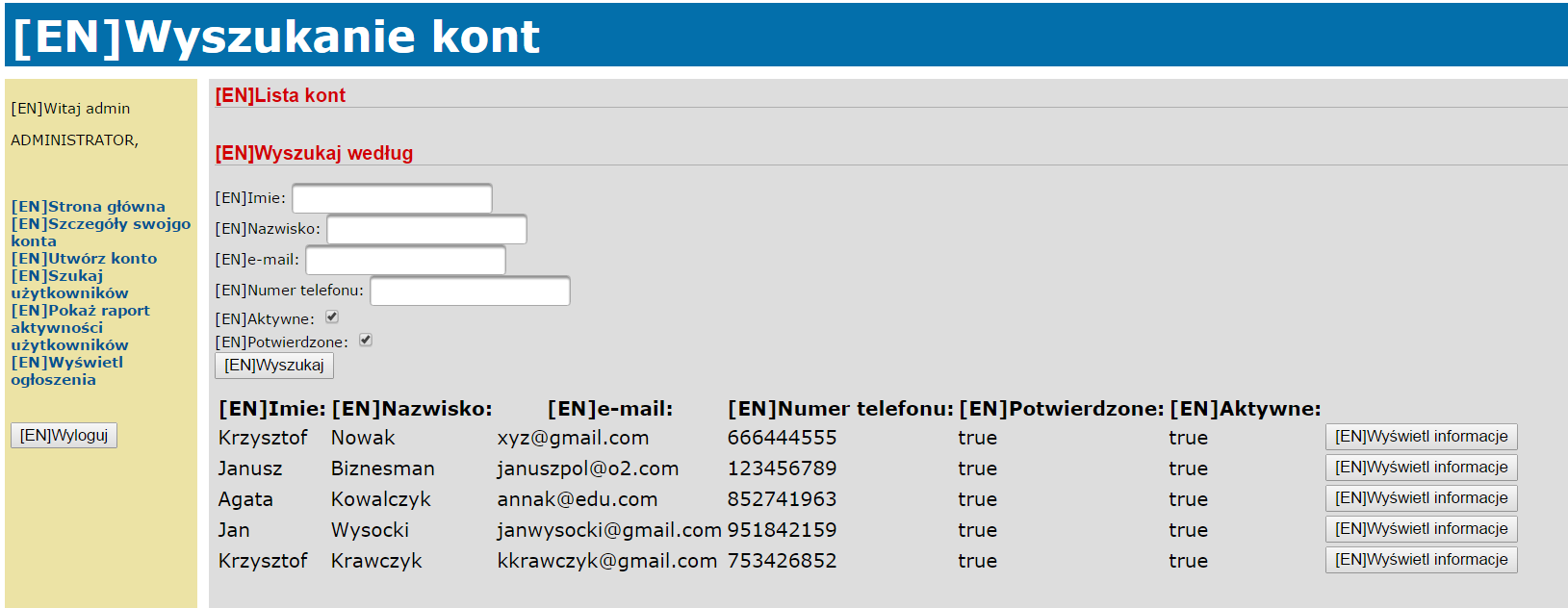
**Wylogowanie użytkownika z systemu**

****

****

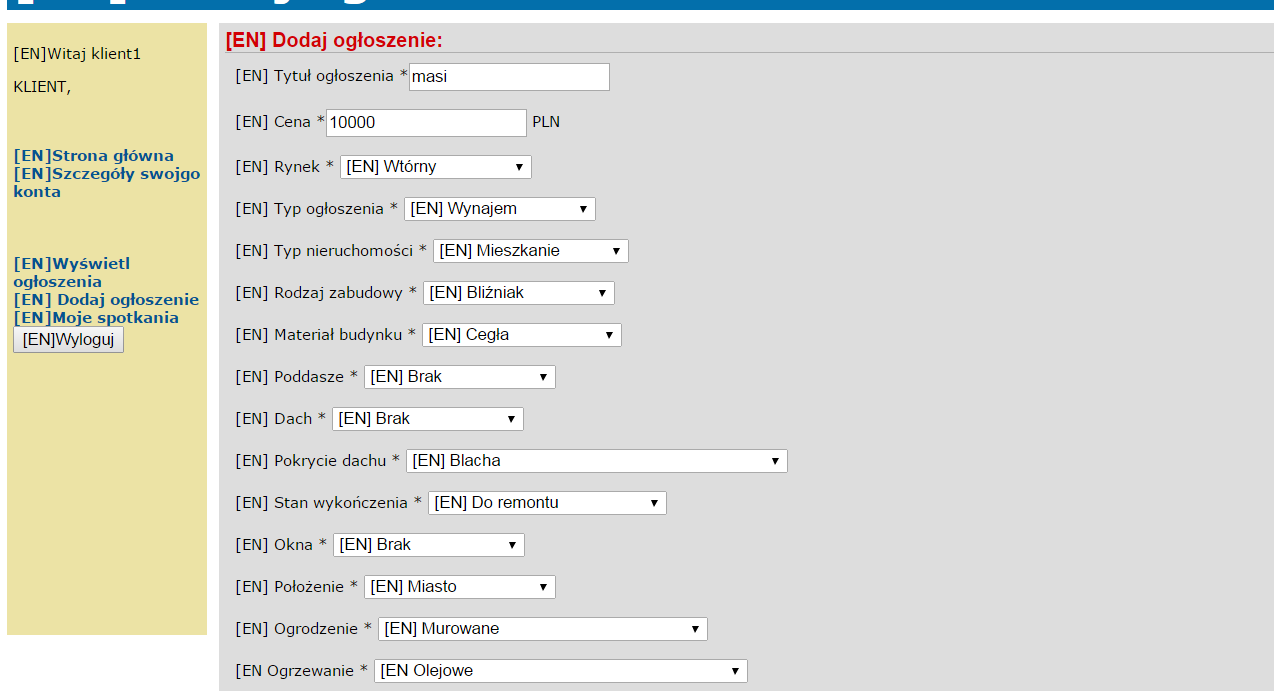
**Test pozytywny.**

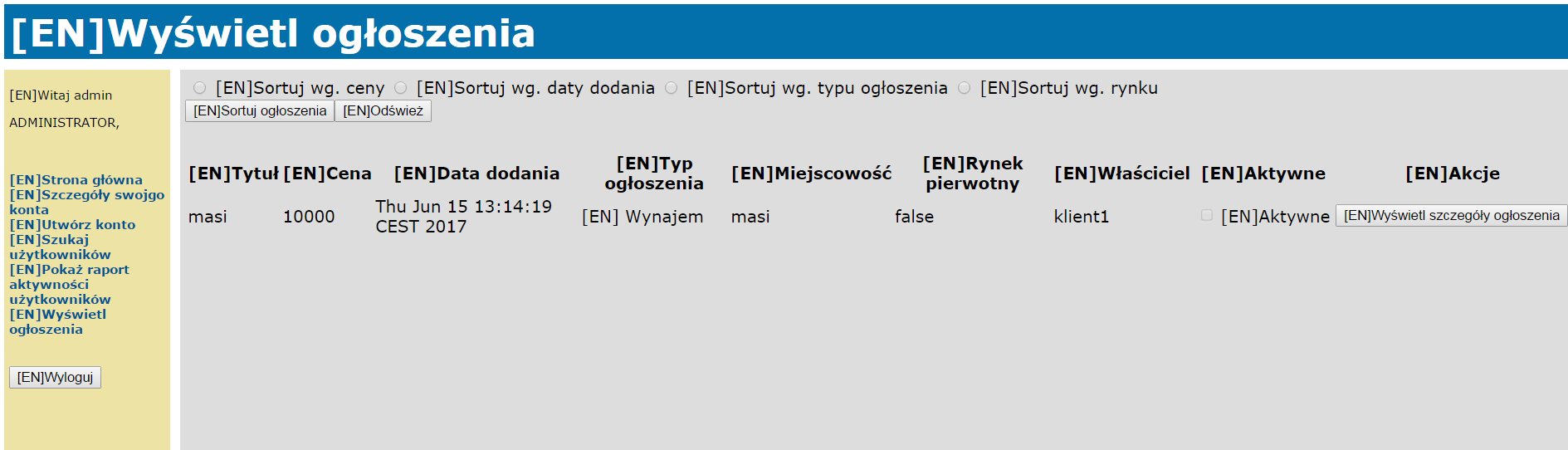
**Wyszukaj użytkowników po kryterium**

****

**Test pozytywny.**

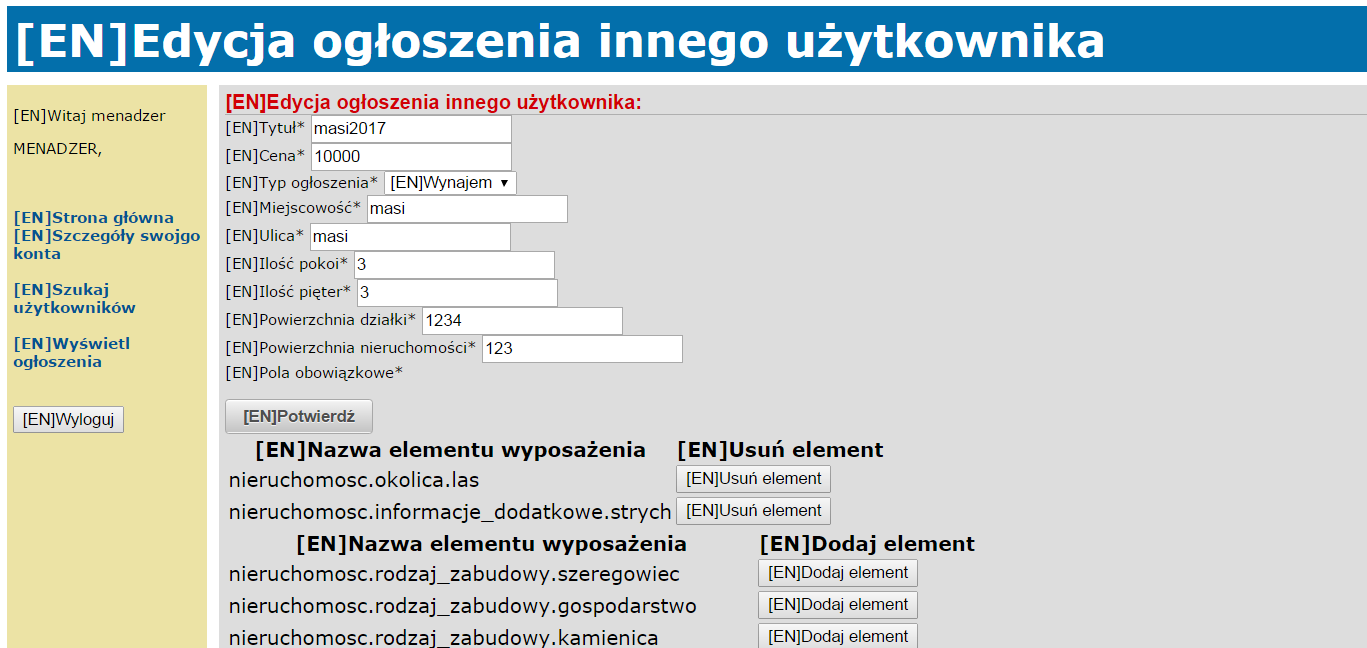
**Dodanie ogłoszenia do systemu**

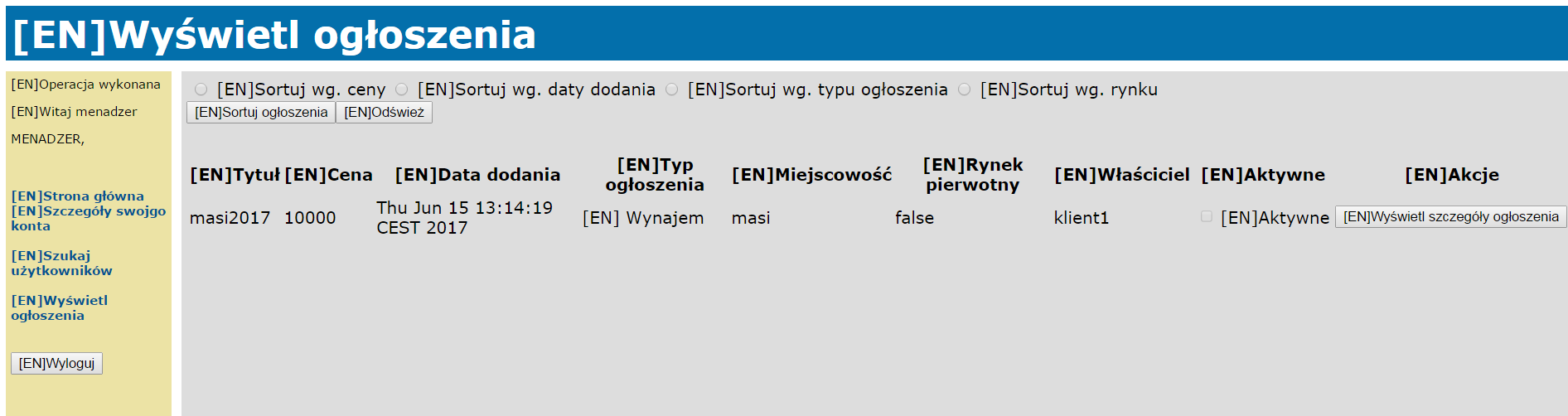
****

****

**Test pozytywny.**

**Edycja ogłoszenia innego użytkownika**

****

****

**Test pozytywny.**

**Zmiana agenta w wybranym ogłoszeniu**

****

****

**Test pozytywny.**

**TESTY WYDAJNOŚCIOWE**

**Głęboka kopia obiektu wykorzystując serializację**

/\*\*

\* Test of deepCloneThroughSerializationW method, of class CloneUtils.

\*/

@Test

public void testDeepCloneThroughSerializationW() throws Exception {

System.out.println("deepCloneThroughSerializationW");

long start = System.currentTimeMillis();;

Serializable source = new Konto(1L);

Object result = CloneUtils.deepCloneThroughSerialization(source);

long time = System.currentTimeMillis() - start;

System.out.println(time);

assertTrue(time<50);

}

**Wyniki testu:**

Running pl.lodz.p.it.ssbd2016.ssbd01.Utils.CloneUtilsTest

deepCloneThroughSerializationW

31

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

**Tworzenie poziomu dostępu z określonego zbioru wartosci**

**Sprawdzanie warunków dodania poziomów dostępu**

/\*\*

\* Test of stworzPoziomDostepu method, of class PoziomDostepuManager.

\*/

@Test

public void testStworzPoziomDostepu() throws Exception {

System.out.println("stworzPoziomDostepu");

long start = System.currentTimeMillis();

String poziom = "AGENT";

PoziomDostepuManager instance = new PoziomDostepuManager();

PoziomDostepu result = instance.stworzPoziomDostepu(poziom);

long time = System.currentTimeMillis() - start;

System.out.println(time);

assertTrue(time<50); }

/\*\*

\* Test of czyMoznaDodacPoziom method, of class PoziomDostepuManager.

\*/

@Test

public void testCzyMoznaDodacPoziom() {

KontoFacade kontoFacade=new KontoFacade();

System.out.println("czyMoznaDodacPoziom!");

long start = System.currentTimeMillis();

Konto konto = kontoFacade.znajdzPoLoginie("agent");

String nowyPoziom = "MENADZER";

PoziomDostepuManager instance = new PoziomDostepuManager();

boolean result = instance.czyMoznaDodacPoziom(konto, nowyPoziom);

long time = System.currentTimeMillis() - start;

System.out.println(time);

assertTrue(time<100);

}

**Wyniki testu:**

czyMoznaDodacPoziom!

15

stworzPoziomDostepu

67

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

**GENEROWANIE skrótu MD5**

/\*\*

\* Test of generateMD5Hash method, of class MD5Generator.

\*/

@Test

public void testGenerateMD5Hash() throws Exception {

System.out.println("generateMD5Hash");

long start = System.currentTimeMillis();

String haslo = "zaq1@WSX";

String expResult = "9e38e8d688743e0d07d669a1fcbcd35b";

String result = MD5Generator.generateMD5Hash(haslo);

long time = System.currentTimeMillis() - start;

System.out.println(time);

assertTrue(time<50);

}

**Wynik testu:**

Running pl.lodz.p.it.ssbd2016.ssbd01.mok.utils.MD5GeneratorTest

generateMD5Hash

38

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**

**Pobieranie wiadomosci ze słownika**

/\*\*

\* Test of getMessage method, of class MessageProvider.

\*/

@Test

public void testGetMessage() {

System.out.println("getMessage");

long start = System.currentTimeMillis();

String key = "notyfikacja.rejestracja.tytul";

String result = MessageProvider.getMessage(key);

long time = System.currentTimeMillis() - start;

System.out.println(time);

assertTrue(time<50);

}

**Wynik testu:**

Running pl.lodz.p.it.ssbd2016.ssbd01.Utils.MessageProviderTest

getMessage

9

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

**Test pozytywny.**