



Koszalin, 21-02-2018 r.

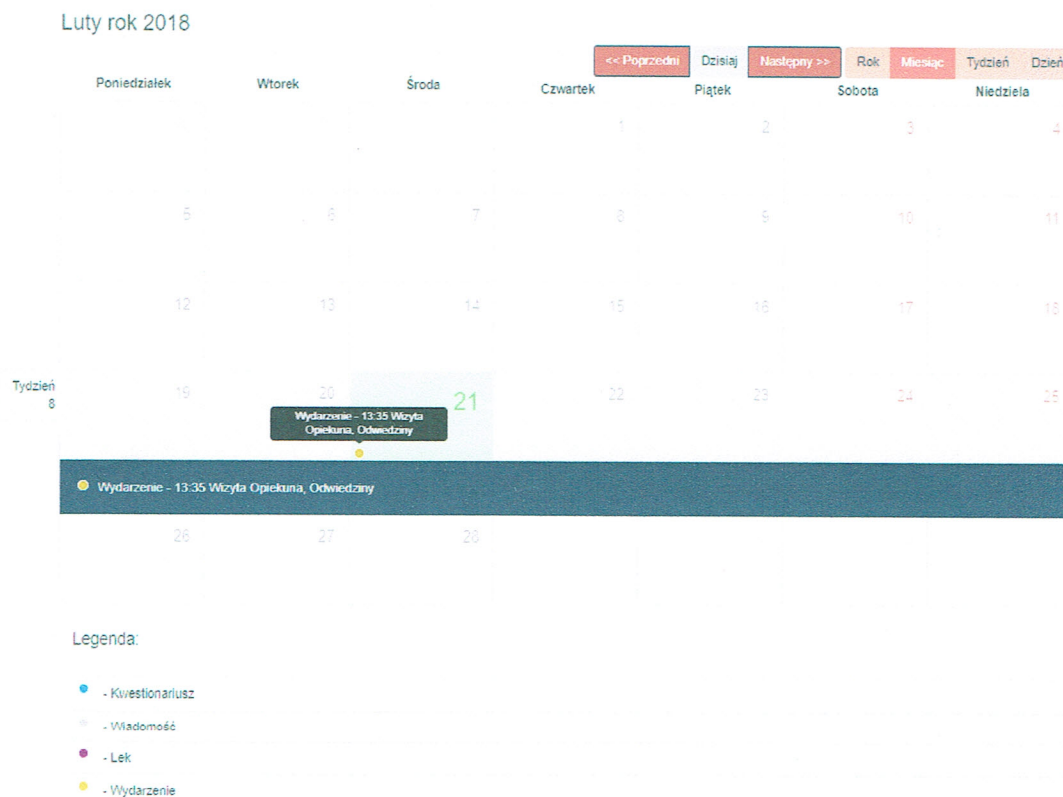
**Merlion sp. z o.o.**  
**ul. Różana 3**  
**75-212 Koszalin**

## Opis Innowacji - Terminarz

- **Terminarz(kalendarz)** – wyświetla wszystkie dni tygodnia z podziałem na miesiące (poprzez kliknięcie na dzień tygodnia mamy podgląd szczegółów zdarzenia)

Ustalono następujące pola:

1. Data
2. Kolor wyświetlanego zdarzenia
3. Nazwa zdarzenia
4. Krótki opis zdarzenia



**rys. 1. Widok kalendarza - Panel**

### Kod źródłowy nr 1 – Wyświetlanie wydarzenia:

```
package com.revoapp.notification;
```

```
import android.app.Activity;  
import android.content.Context;  
import android.content.Intent;  
import android.media.AudioManager;  
import android.media.MediaPlayer;  
import android.media.MediaPlayer.OnCompletionListener;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.View;  
import android.view.View.OnClickListener;  
import android.widget.Button;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import com.revoapp.api.database.calendar.pojo.Events;  
import com.revoapp.api.helper.BroadcastHelper;  
import com.revoapp.api.utils.DateTimeConvert;  
import com.revoapp.api.utils.Utils;  
import com.revoapp.api.utils.WakePhone;  
import com.revoapp.notification.database.CalendarContentResolver;  
import com.revoapp.notification.database.Notification;  
import com.revoapp.notification.database.NotificationDatabaseHandler;  
import com.revoapp.notification.helpers.PreferencesManager;  
import com.revoapp.notification.helpers.ThemeHelper;
```

```
import java.util.Calendar;
```

```
public class EventShowActivity extends Activity {  
    private Context context;  
    private int id;  
  
    private MediaPlayer mp;  
    private NotificationDatabaseHandler dbNotification;  
    private Notification notification;  
    private int count;  
    private int orientation;  
  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.event_show_activity);  
  
        context = this;  
        orientation = getResources().getConfiguration().orientation;  
  
        Utils.disableLockScreen(this);
```

```
if (getIntent().hasExtra("id"))
    id = getIntent().getExtras().getInt("id");

init();
}

private void init() {
    CalendarContentResolver db = new CalendarContentResolver();
    dbNotification = new NotificationDatabaseHandler(context);

    Events event = db.getEvent(this, id);
    notification = new Notification(MainActivity.eventAPP, id,
Calendar.getInstance().getTimeInMillis());

    LinearLayout ll = (LinearLayout) findViewById(R.id.alarm_layout);
    ll.setBackgroundResource(ThemeHelper.get_main_background());

    TextView title = (TextView) findViewById(R.id.alarm_title);
    if (event.getTitle() != null)
        title.setText(getResources().getString(R.string.event) + " " + event.getTitle());
    title.setSelected(true);

    TextView time = (TextView) findViewById(R.id.alarm_time);
    if (event.getDate() > 0)
        time.setText(DateTimeConvert.getTime(context, event.getDate()));
    else
        time.setText("");

    TextView message = (TextView) findViewById(R.id.alarm_message);
    message.setText(event.getMessage());

    WakePhone.acquire(context, "Alarm_one");

    Button butOk = (Button) findViewById(R.id.alarm_trun_off);

    butOk.setOnClickListener(takePillListener);

    mp = new MediaPlayer();
    mp.setOnCompletionListener(new OnCompletionListener() {

        @Override
        public void onCompletion(MediaPlayer arg0) {
            if (count < 3) {
                count++;
                arg0.seekTo(0);
                arg0.start();
            }
        }
    });

    mp.setAudioStreamType(AudioManager.STREAM_ALARM);
```

```
try {
    Uri myUri = Uri.parse("android.resource://" + context.getPackageName() + "/" +
R.raw.quest);
//    Uri myUri = null;
    mp.setDataSource(context, myUri);
    mp.prepare();
} catch (Exception e) {
}

mp.start();
}

private OnClickListener takePillListener = new OnClickListener() {

    @Override
    public void onClick(View v) {
        if (notification != null)
            dbNotification.deleteNotification(notification);
        PreferencesManager.setNotificationCount(context, dbNotification.getNotificationCount());

        onFinish();
    }
};

@Override
protected void onDestroy() {
    onFinish();
    super.onDestroy();
}

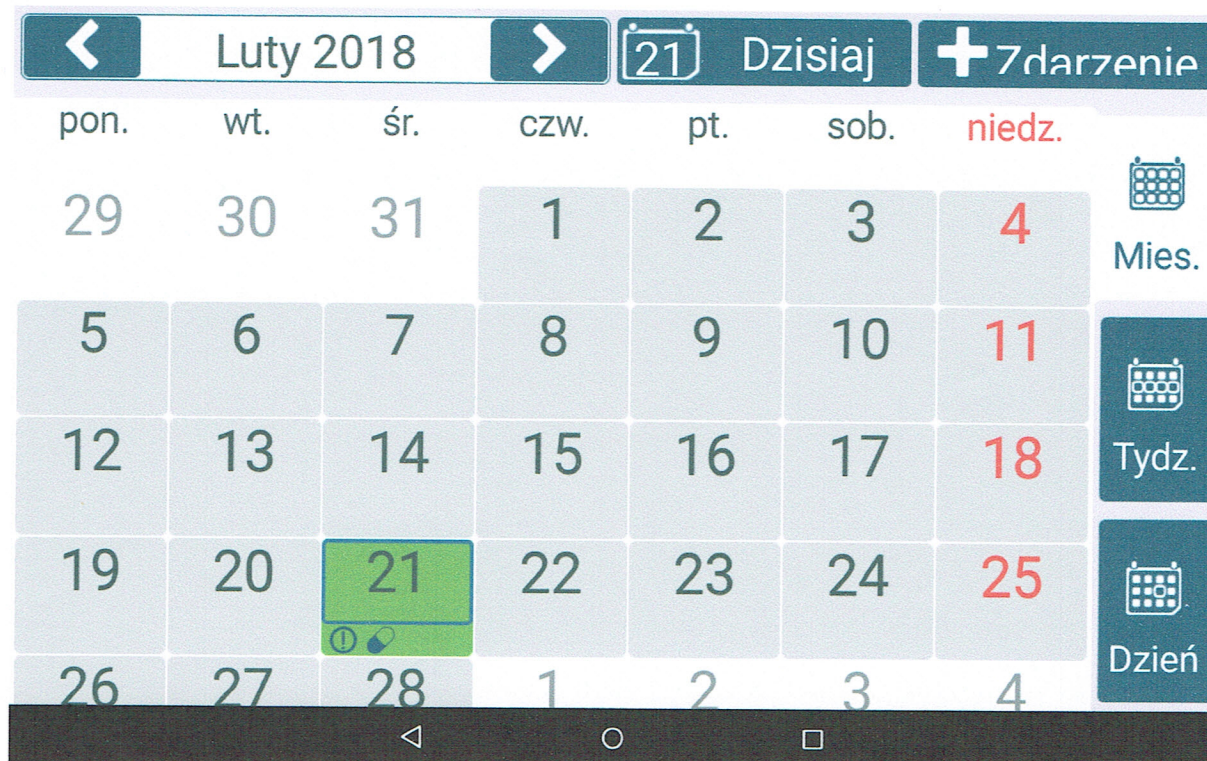
@Override
protected void onStop() {
    onFinish();
    super.onStop();
}

protected void onFinish() {
    if (mp != null)
        mp.release();

    Intent radiolIntent =
BroadcastHelper.getIntent(BroadcastHelper.ACTION_RADIO_RESUME_PLAYING);
    radiolIntent.putExtra("NOTIFICATION", true);
    context.sendBroadcast(radiolIntent);

    WakePhone.release();

    if(orientation == getResources().getConfiguration().orientation){
        finish();
    }
}
}
```

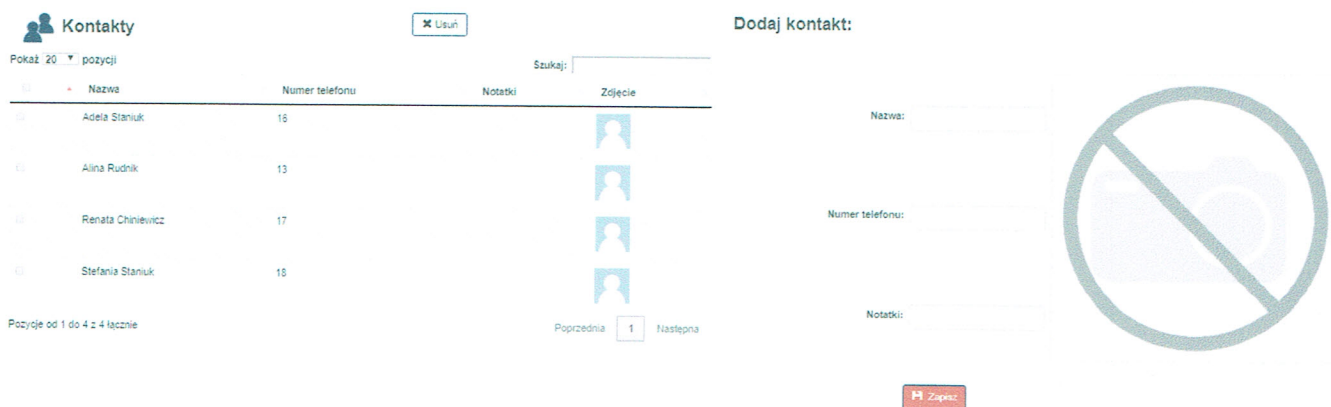


rys. 2. Widok kalendarza – Terminal

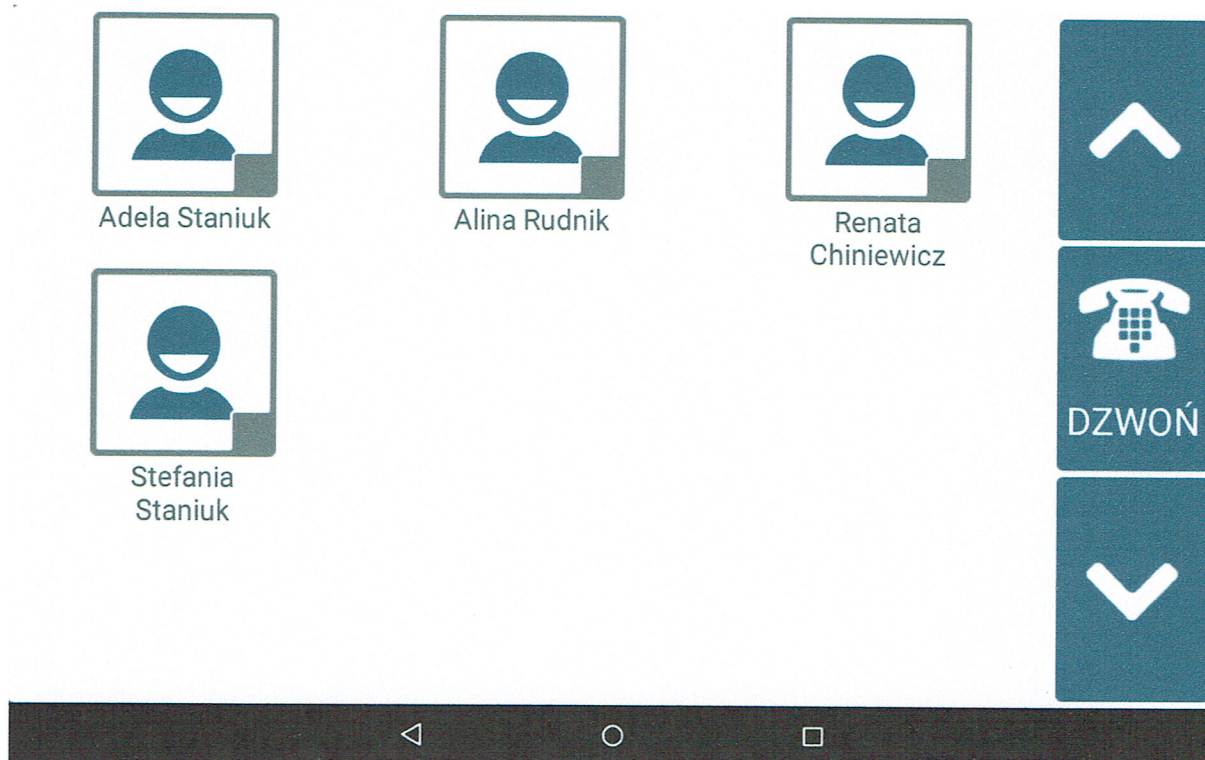
- **Kontakty** – tworzenie kontaktu/wpisanie w formularz oraz analogicznie możliwość usunięcia

Ustalono następujące pola:

1. Nazwa
2. Notatka
3. Nr telefonu(również nr skrócone)
4. Zdjęcie(format .png lub .jpg)




**rys. 3. Zarządzanie listą kontaktów – Panel**



**rys. 4. Lista kontaktów – Terminal**

- **Radio** – wyświetla listę stacji radiowych (po przydzieleniu przez resellera zasobów). Każde radio generuje strumień audio i nadaje w czasie rzeczywistym.

 **Lista stacji radiowych**

Pokaż 20 pozycji Szukaj:

Nazwa	Typ radio	<input type="checkbox"/>
3FM		<input type="checkbox"/>
BBC World	BBC	<input type="checkbox"/>
BBC World Service	BBC	<input type="checkbox"/>
Capital		<input type="checkbox"/>
Channel 103FM		<input type="checkbox"/>
Classic FM		<input type="checkbox"/>
Heart (London)		<input type="checkbox"/>
Jazz FM (London)		<input type="checkbox"/>
KISS		<input type="checkbox"/>
PR Radio Gdańsk		<input type="checkbox"/>
Radio Eska		<input type="checkbox"/>
Radio Kaszebe		<input type="checkbox"/>
Radio Koszalin		<input type="checkbox"/>
Radio Plus (Gdańsk)		<input type="checkbox"/>
Radio RMF FM		<input type="checkbox"/>
Radio ZET		<input type="checkbox"/>
Smooth Radio (London)		<input type="checkbox"/>
Yorkshire Coast Radio		<input type="checkbox"/>
Złote przeboje		<input type="checkbox"/>

Pozycje od 1 do 19 z 19 łącznie

Poprzednia  Następna

rys. 5. Lista stacji radiowych – Panel

## Kod źródłowy nr 2 – Inicjacja listy stacji radiowych:

```
private void init() {

    DatabaseHandler databaseHandler = new DatabaseHandler(context);

    stationList = new ArrayList<Station>();
    // stationList.addAll(Templates.getSampleStations());
    stationList.addAll(databaseHandler.getAllStations());

    LinearLayout mainLayout = (LinearLayout) findViewById(R.id.mainLayout);
    mainLayout.setBackgroundResource(ThemeHelper.get_main_background());

    ListView stationsLV = (ListView) findViewById(R.id.stationsLV);
    adapter = new StationListAdapter(context, R.layout.stations_list_item, stationList);
    stationsLV.setAdapter(adapter);
    stationsLV.setDivider(getResources().getDrawable(ThemeHelper.get_separator_list()));
    stationsLV.setOnItemClickListener(stationOnClickListener);
    //stationsLV.setOnScrollListener(scrollListener);

    if(orientation == ImageHelper.LANDSCAPE){
        sizeOfButton = setSizelImage(0.21, 1);
    } else if (orientation == ImageHelper.PORTRAIT){
        sizeOfButton = setSizelImage(0.15, 0);
    }

    Button volumeDownB = (Button) findViewById(R.id.volumeDownB);
    volumeDownB.setBackgroundResource(ThemeHelper.get_component_button_selector());

    if(orientation == ImageHelper.LANDSCAPE){

    volumeDownB.setCompoundDrawablesWithIntrinsicBounds(ImageHelper.getScaledDrawableFromResources(this, ThemeHelper.get_icon_100_vol_down(), sizeOfButton, sizeOfButton), null, null, null);
    } else if (orientation == ImageHelper.PORTRAIT){
        volumeDownB.setCompoundDrawablesWithIntrinsicBounds(null,
        ImageHelper.getScaledDrawableFromResources(this, ThemeHelper.get_icon_100_vol_down(),
        sizeOfButton, sizeOfButton), null, null);
    }

    //
    volumeDownB.setCompoundDrawablesWithIntrinsicBounds(ThemeHelper.get_icon_100_vol_down(),
    0, 0, 0);
    volumeDownB.setOnClickListener(volumeButtonListener);
    volumeDownB.setHeight(sizeOfButton);

    Button volumeUpB = (Button) findViewById(R.id.volumeUpB);
    volumeUpB.setBackgroundResource(ThemeHelper.get_component_button_selector());

    if(orientation == ImageHelper.LANDSCAPE){
```

```
volumeUpB.setCompoundDrawablesWithIntrinsicBounds(ImageHelper.getScaledDrawableFromResources(this, ThemeHelper.get_icon_100_vol_up(), sizeOfButton, sizeOfButton), null, null, null);
    } else if (orientation == ImageHelper.PORTRAIT){
        volumeUpB.setCompoundDrawablesWithIntrinsicBounds(null,
ImageHelper.getScaledDrawableFromResources(this, ThemeHelper.get_icon_100_vol_up(),
sizeOfButton, sizeOfButton), null, null);
    }

//    volumeUpB.setCompoundDrawablesWithIntrinsicBounds(ThemeHelper.get_icon_100_vol_up(),
0, 0, 0);
volumeUpB.setOnClickListener(volumeButtonListener);
volumeUpB.setHeight(sizeOfButton);

MediaPlayerStream.createMediaPlayer();

playB = (RevoButton) findViewById(R.id.playB);
if(!MediaPlayerStream.mediaPlayer.isPlaying()){
    playB.setText(getResources().getText(R.string.play));
} else {
    playB.setText(getResources().getText(R.string.pause));
}
playB.setOnClickListener(playButtonListener);
playB.setBackgroundResource(ThemeHelper.get_component_button_selector());
if(orientation == ImageHelper.LANDSCAPE){

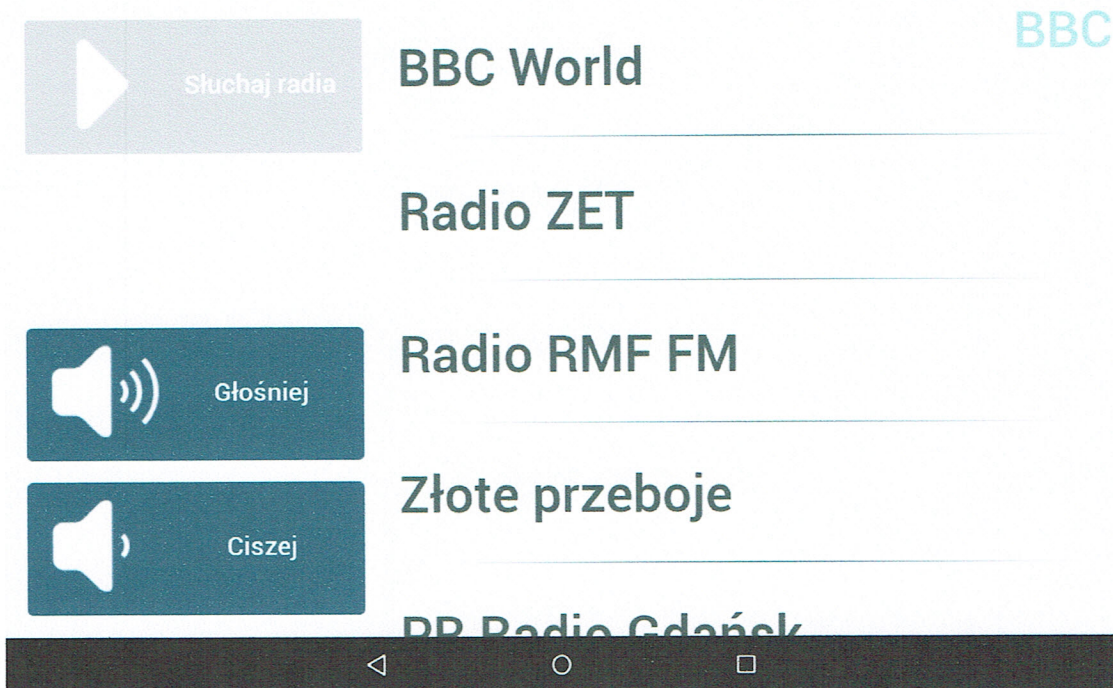
playB.setCompoundDrawablesWithIntrinsicBounds(ImageHelper.getScaledDrawableFromResources(
this, ThemeHelper.get_icon_100_play(), sizeOfButton, sizeOfButton), null, null, null);
//    playB.setCompoundDrawablesWithIntrinsicBounds(ThemeHelper.get_icon_100_play(), 0, 0,
0);
    if (MediaPlayerStream.mediaPlayer.isPlaying()) {

playB.setCompoundDrawablesWithIntrinsicBounds(ImageHelper.getScaledDrawableFromResources(
this, ThemeHelper.get_icon_100_pause(), sizeOfButton, sizeOfButton), null, null, null);
//    playB.setCompoundDrawablesWithIntrinsicBounds(ThemeHelper.get_icon_100_pause(), 0,
0, 0);
    }
    } else if (orientation == ImageHelper.PORTRAIT){
        playB.setCompoundDrawablesWithIntrinsicBounds(null,
ImageHelper.getScaledDrawableFromResources(this, ThemeHelper.get_icon_100_play(),
sizeOfButton, sizeOfButton), null, null);
        if (MediaPlayerStream.mediaPlayer.isPlaying()) {
            playB.setCompoundDrawablesWithIntrinsicBounds(null,
ImageHelper.getScaledDrawableFromResources(this, ThemeHelper.get_icon_100_pause(),
sizeOfButton, sizeOfButton), null, null);
        }
    }
}

playB.setHeight(sizeOfButton);

audioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
currentVolume = audioManager.getStreamVolume(AudioManager.STREAM_MUSIC);

checkIfRadiolsPlaying(); }
```

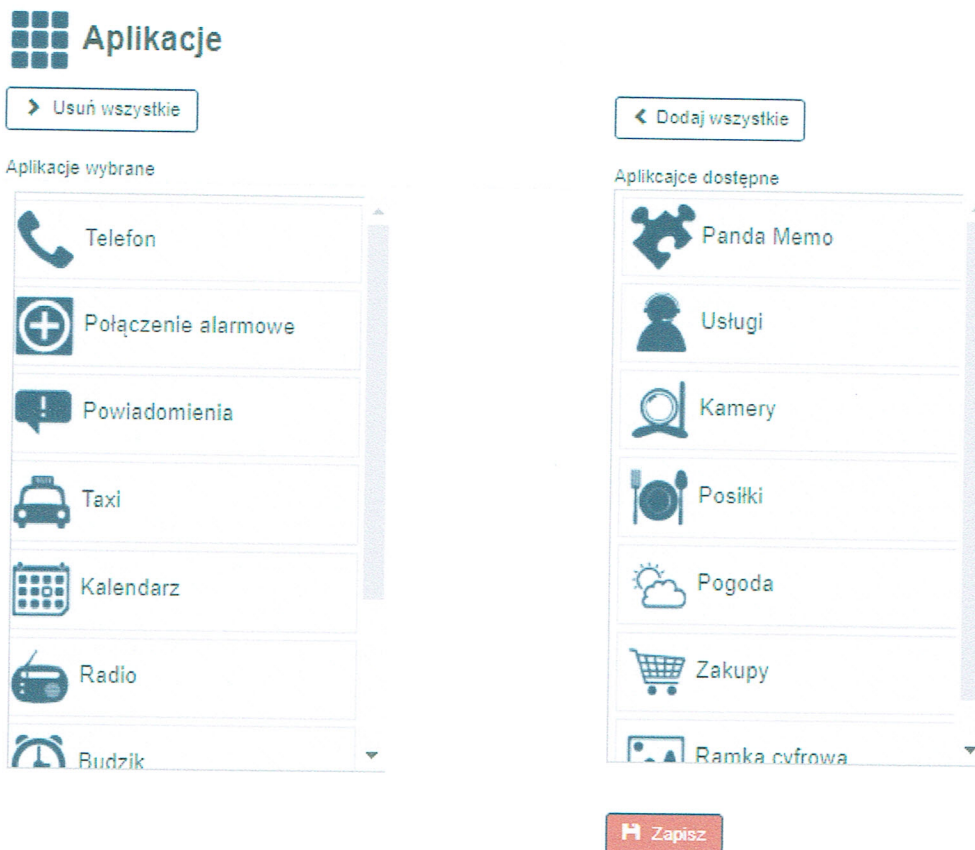


rys. 6. Lista stacji radiowych – Terminal

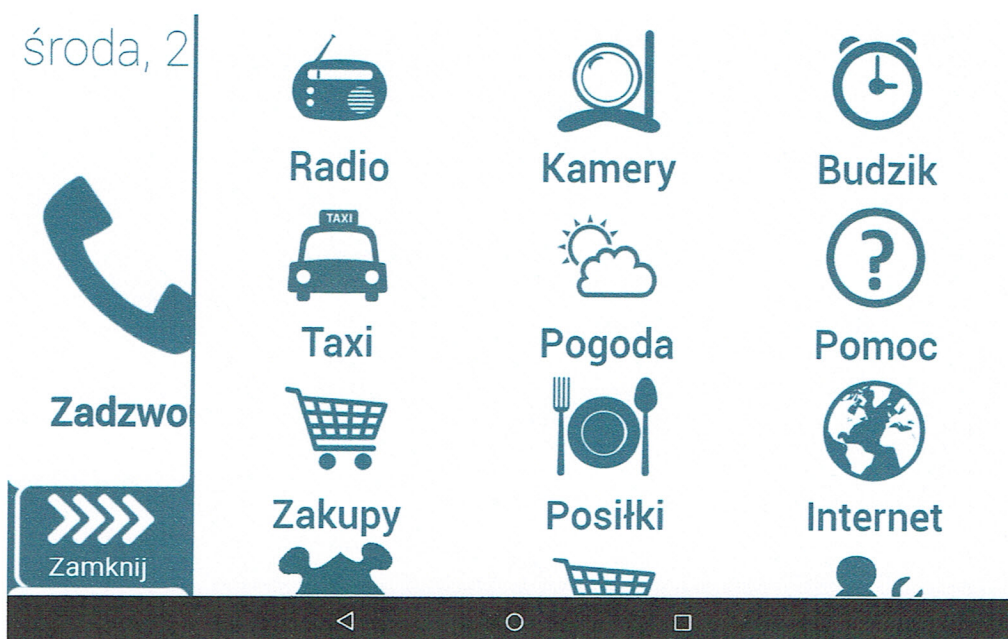
- **Aplikacje** – przydziela reseller dla terminali w domu.

Dostępne aplikacje:

1. Telefon – wyświetla dostępne kontakty i umożliwia wykonywanie połączeń
2. Połączenie alarmowe SOS - „czerwony guzik”
3. Powiadomienie – wyświetla wszystkie powiadomienia
4. Kalendarz – wyświetla kalendarz i dodane do niego zdarzenia
  - terminarz zażywania leków
  - terminarz wizyt lekarskich
  - terminarz urodzin/imienin
  - terminarz audycji/filmów
  - inne np. Budzik
5. Wyświetlenie dodanego zdjęcia
6. Radio – pozwala na uruchomienie odtwarzania wybranej stacji
7. Pomoc(Samouczek) – Wybór zmian aplikacji po przeciągnięciu z pola „Aplikacje dostępne” do pola „Aplikacje wybrane” lub poprzez kliknięcie „dodaj wszystkie” i „usuń wszystkie”



rys. 7. Zarządzanie aplikacjami – Panel

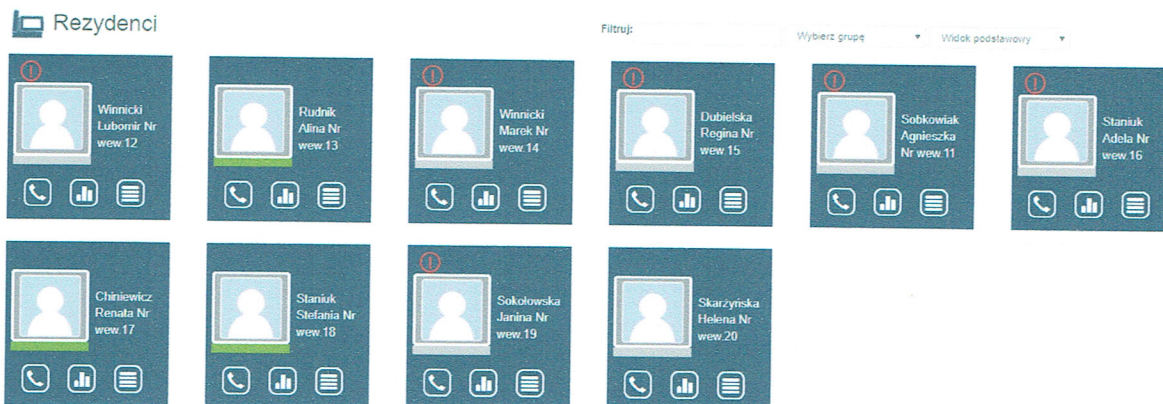


rys. 8. Widok aplikacji – Terminal

- **Home – rezydenci.** Wyświetla dane rezydentów oraz statusy urządzeń dedykowanych w domach rezydentów.

Ustalono następujące pola:

1. Zdjęcie(format .png lub .jpg)
2. Nazwisko
3. Imię
4. nr skrócony
5. Alert



rys. 9. Widok rezydentów – Panel

- **Widok** - winien być podzielony na grupy gdzie po wybraniu danej grupy wyświetlone zostaną tylko osoby przypisane do wybranej grupy.

Ustalono następujące pola:

1. Leki
2. Kwestionariusze
3. Wiadomości

rys. 10. Widok grupy rezydentów – Panel

- **Status** – pod zdjęciem każdego profilu widoczne są 3 przyciski:
  1. Połącz – po kliknięciu uruchamiane jest połączenie głosowe między osobą monitorującą a rezydentem po numerze skróconym
  2. Intercom – po kliknięciu uruchamiane jest połączenie głosowe między osobą monitorującą a rezydentem po numerze skróconym, dodatkowo połączenie jest automatycznie odbierane przez urządzenie znajdujące się u rezydenta oraz zostaje uruchomiony moduł głośnomówiący
  3. Szczegóły – po kliknięciu następuje przejście do profilu rezydenta

rys. 11. Status rezydenta – Panel



- **Leki** – wyświetla zaplanowane powiadomienia dotyczące zażycia leków dla wybranych rezydentów.

Ustalono następujące pola:

1. Nazwa
2. Dawka
3. Notatki
4. Od-do
5. Godzina

**Leki**

Pokaż 20 pozycji

+ Dodaj nowy lek    ✕ Usuń

Szukaj:

Nazwa	Od	Do	Dawka	Notatki	Edytuj
Apap	2018-02-21	2018-02-21	1 tabletki		Edit

Pozycje od 1 do 1 z 1 łącznie

Poprzednia 1 Następna

**Szczegóły leku**

Nazwa	Data	Potwierdzono	Dawka	Notatki
Apap	21-02-2018 15:00		1 tabletki	

Dodaj nowy lek

Nazwa: \_\_\_\_\_ Od: \_\_\_\_\_

Dawka: \_\_\_\_\_ Do: \_\_\_\_\_

Notatki: \_\_\_\_\_ Godzina: \_\_\_\_\_

+ Dodaj godzinę

+ Dodaj nowy lek

rys. 12. Zarządzanie lekami – Panel



**Apap**  
21-02-2018 15:00  
**1 tabletki**

**Potwierdź**

rys. 13. Powiadomienie o zażyciu leku – Terminal

### Kod źródłowy nr 3 – Wyświetlanie leku:

```
package com.revoapp.notification;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.text.format.DateFormat;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.revoapp.api.Config;
import com.revoapp.api.database.calendar.CalendarRDH;
import com.revoapp.api.database.calendar.pojo.Pills;
import com.revoapp.api.helper.BroadcastHelper;
import com.revoapp.api.logger.RevoLogger;
import com.revoapp.api.utils.ImageHelper;
import com.revoapp.api.utils.Utils;
import com.revoapp.api.utils.WakePhone;
import com.revoapp.notification.database.CalendarContentResolver;
import com.revoapp.notification.database.Notification;
import com.revoapp.notification.database.NotificationDatabaseHandler;
import com.revoapp.notification.helpers.PreferencesManager;
import com.revoapp.notification.helpers.ThemeHelper;

import java.text.SimpleDateFormat;
import java.util.ArrayList;

public class PillsShowActivity extends Activity {
    private Context context;
    private ArrayList<Pills> pills;
    private TextView pillName, pillTime, pillInfo, pillQuantity;

    private MediaPlayer mp;
    private NotificationDatabaseHandler dbNotification;
    private Notification notification;
    private int selectedPillIndex = 0;

    private SimpleDateFormat df;

    private Handler handler;
```

```
private int orientation;

private CalendarContentResolver calendarContentResolver;

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.show_pill_alarm_activity);

    context = this;
    orientation = getResources().getConfiguration().orientation;

    Utils.disableLockScreen(this);

    if (DateFormat.is24HourFormat(context))
        df = new SimpleDateFormat("dd-MM-yyyy HH:mm");
    else
        df = new SimpleDateFormat("dd-MM-yyyy hh:mm a");

    initView();
    dbNotification = new NotificationDatabaseHandler(context);
    initPills();

    WakePhone.acquire(this, "Alarm_one");
}

private void initView() {
    RelativeLayout ll = (RelativeLayout) findViewById(R.id.alarm_layout);
    ll.setBackgroundResource(ThemeHelper.get_main_background());

    pillName = (TextView) findViewById(R.id.pill_title);
    pillTime = (TextView) findViewById(R.id.pill_time);
    pillQuantity = (TextView) findViewById(R.id.pill_quantity);
    pillInfo = (TextView) findViewById(R.id.pill_describe);

    int sizeOfImg = 0;

    if (orientation == ImageHelper.LANDSCAPE) {
        sizeOfImg = ImageHelper.getScaledImageFromScreenWidth(this, 0.13);
    } else if (orientation == ImageHelper.PORTRAIT) {
        sizeOfImg = ImageHelper.getScaledImageFromScreenHeigh(this, 0.13);
    }

    ImageView pillIV = (ImageView) findViewById(R.id.pillIV);
    pillIV.setBackgroundDrawable(ImageHelper.getScaledDrawableFromResources(this,
    ThemeHelper.get_icon100_pill(), sizeOfImg, sizeOfImg));
    // pillIV.setBackgroundResource(ThemeHelper.get_icon100_pill());

    Button takePill = (Button) findViewById(R.id.take_pill);
    takePill.setOnClickListener(takePillListener);
}
```

```
private void initPills() {
    calendarContentResolver = new CalendarContentResolver();
    pills = new ArrayList<Pills>();

    Bundle b = getIntent().getBundleExtra("idBundle");
    if (b != null) {
        ArrayList<Integer> idList = b.getIntegerArrayList("id");
        for (int i : idList)
            pills.add(calendarContentResolver.getPill(this, i));

        prepareMediaPlayer();
    } else if (getIntent().hasExtra("id")) {
        int pillId = getIntent().getExtras().getInt("id");
        if (pillId > 0) {
            pills.add(calendarContentResolver.getPill(this, pillId));
        }
    }

    if (pills.size() > 0) {
        selectedPillIndex = 0;

        showNextPill();
    }
}

private void takePill() {
    if (notification != null) {
        removeNotification();
        new InsertConfirmAsync().execute(notification.getMissedId());
    }
}

private void showNextPill() {
    RevoLogger.log(RevoLogger.DEBUG, this.getClass().getName(), "showNextPill",
        "selectedPillId = " + selectedPillIndex);
    notification = new Notification(MainActivity.medicationAPP,
        pills.get(selectedPillIndex).getmId(), pills.get(selectedPillIndex).getmDateTime());

    pillName.setText(pills.get(selectedPillIndex).getmPillName());
    pillQuantity.setText(pills.get(selectedPillIndex).getmQuantity());
    pillInfo.setText(pills.get(selectedPillIndex).getmDescribe());
    pillTime.setText(df.format(notification.getMissedDate() * 1000));

    setDismissTask(notification.getMissedDate() * 1000);
}

private void setDismissTask(long pillTime) {
    if (handler != null)
        handler.removeCallbacksAndMessages(null);
}
```

```
long alarmTime = Config.PILL_EXPIRATION_TIME - (System.currentTimeMillis() - pillTime);
RevoLogger.log(RevoLogger.DEBUG, this.getClass().getName(), "setDismissTask",
    "alarmTime = " + alarmTime);

handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        RevoLogger.log(RevoLogger.DEBUG, this.getClass().getName(), "setDismissTask",
            "PILL_EXPIRED");
        removeNotification();
        selectedPillIndex++;
        if (selectedPillIndex < pills.size())
            showNextPill();
        else
            onFinish();
    }
}, alarmTime);
}

private void removeNotification() {
    dbNotification.deleteNotification(notification);
    PreferencesManager.setNotificationCount(context, dbNotification.getNotificationCount());
}

private OnClickListener takePillListener = new OnClickListener() {
    @Override
    public void onClick(View v) {
        selectedPillIndex++;
        if (selectedPillIndex < pills.size()) {
            takePill();
            showNextPill();
        } else {
            takePill();
            onFinish();
        }
    }
};

@Override
protected void onStop() {
    onFinish();
    super.onStop();
}

private void prepareMediaPlayer() {
    mp = new MediaPlayer();
    mp.setAudioStreamType(AudioManager.STREAM_ALARM);

try {
    Uri myUri = Uri.parse("android.resource://" + context.getPackageName() + "/" + R.raw.pill);
```

```
        mp.setDataSource(context, myUri);
        mp.prepare();
    } catch (Exception e) {
    }

    mp.setLooping(true);
    mp.start();
}

private void onFinish() {
    if (handler != null)
        handler.removeCallbacksAndMessages(null);

    if (mp != null) {
        try {
            if (mp.isPlaying())
                mp.stop();
        } catch (IllegalStateException e) {
            RevoLogger.log(RevoLogger.ERROR, this.getClass().getName(), "onFinish",
                "IllegalStateException e = " + e.getMessage());
        }

        mp.release();
    }

    WakePhone.release();

    Intent radioIntent =
BroadcastHelper.getIntent(BroadcastHelper.ACTION_RADIO_RESUME_PLAYING);
    radioIntent.putExtra("NOTIFICATION", true);
    context.sendBroadcast(radioIntent);

    if (orientation == getResources().getConfiguration().orientation) {
        finish();
    }
}

private class InsertConfirmAsync extends AsyncTask<Integer, Void, Void> {
    @Override
    protected Void doInBackground(Integer... params) {
        if (params[0] != -1) {
            CalendarRDH rdh = new CalendarRDH(context);
            int result = rdh.insertMedicineConfirm(params[0]);
            int localResult = calendarContentResolver.takePill(context, params[0]);
            RevoLogger.log(RevoLogger.DEBUG, this.getClass().getName(), "InsertConfirmAsync",
                "InsertConfirmAsync apiResult = " + result + ", localResult = " + localResult);
        }
        return null;
    }
}
}
```

- **Profil rezydenta** – aktualizacja danych wpisywanych do formularza. Dostępne rubryki danych: imię; nazwisko; adres; płeć; data urodzenia; grupa krwi; telefon stacjonarny; telefon komórkowy; stan zdrowia; diagnozy; notatki dzienne; kieruj(dwie opcje – zawsze do opiekuna call center, potem kolejne połączenie zgodnie z załączoną listą).
1. Zdjęcie(format .png lub .jpg)
  2. Określa status dostępności osoby monitorowanej(kolor czerwony – linia zajęta, kolor zielony – linia wolna, kolor szary – urządzenie odłączone)
  3. Wyświetlanie aktualnych danych osoby monitorowanej(belka), określa nr wew.(skrótowy) oraz identyfikator terminala
  4. Ikona baterii pokazująca poziom naładowania baterii
  5. Informacja o sygnale wifi oraz o połączeniu kablowym Ethernet
  6. Połącz
  7. Intercom

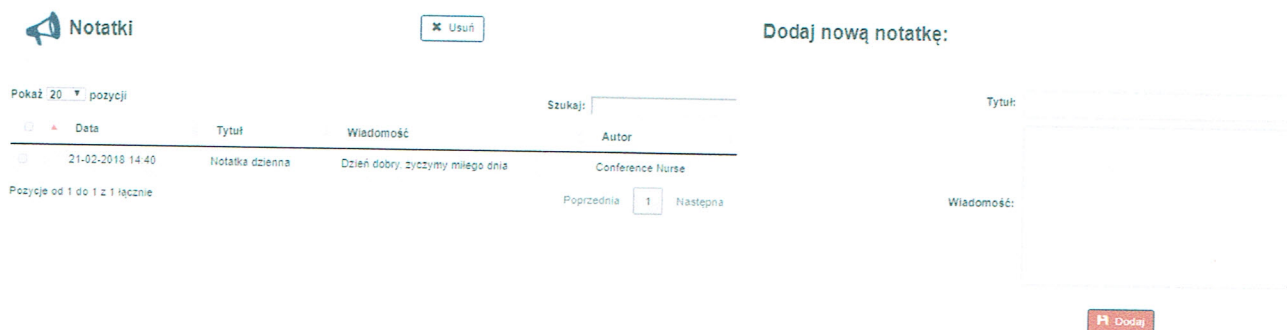
rys. 14. Profil rezydenta – Panel



- **Dodawanie notatek** - poprzez wpisywanie w formularzu.

Ustalono następujące pola:

1. Data
2. Tytuł
3. Treść
4. Autor



**Notatki** ✕ Usuń

Pokaż 20 pozycji

Szukaj:

Data	Tytuł	Wiadomość	Autor
21-02-2018 14:40	Notatka dzienna	Dzień dobry, życzymy miłego dnia	Conference Nurse

Pozycje od 1 do 1 z 1 łącznie

Poprzednia 1 Następna

**Dodaj nową notatkę:**

Tytuł:

Wiadomość:

Dodaj

**rys. 15. Zarządzanie notatkami – Panel**

- **Raporty** – generator danych za wyznaczony okres.

Ustalono następujące dane:

1. Leki
2. Kwestionariusze
3. Komunikaty
4. Historia połączeń alarmowych

Ustalono następujące pola do zapisu danych:

1. Data od-do
2. Lekarstwa
3. Kwestionariusze
4. Komunikaty
5. Historia Alarmów


## Raporty

Data od:

Data do:

Include:

- Lekarstwa
- Kwestionariusze
- Komunikaty
- Historia alarmów
- Sensory
- Sensory medyczne

 **Generuj raport**

**rys. 16. Generowanie raportów – Panel**

# OŚWIADCZENIA DO POLISY UBEZPIECZENIA SPRZĘTU ELEKTRONICZNEGO

## Nr 1028522345



<b>1</b>	<b>Okres ubezpieczenia:</b> od <b>22.02.2018 r.</b> do <b>23.04.2018 r.</b>
<b>2</b>	<b>Ubezpieczający: "MERLION" SP. Z O.O.</b> Adres siedziby: ROŻANA 3, 75-220 KOSZALIN E-mail: Klient nie posiada Telefon: Klient nie posiada REGON: 320654722
<b>3</b>	<b>Składka łączna: 95,39 PLN</b> Sposób płatności: Karta płatnicza Składka została opłacona w całości
<b>4</b>	<b>Informacje do oceny ryzyka</b> <b>Miejsce ubezpieczenia: ROŻANA 3, 75-212 KOSZALIN</b> 1) <b>Działalność prowadzona w lokalizacji wraz z PKD:</b> a) Działalność w zakresie telekomunikacji przewodowej (61.10.Z)
<b>5</b>	<b>Oświadczenia</b> 1. Oświadczam, że przed zawarciem niniejszej umowy otrzymałem ogólne warunki ubezpieczenia sprzętu elektronicznego, ustalone uchwałą Zarządu PZU SA nr UZ/9/2016 z dnia 15.01.2016 r. 2. Oświadczenie o przebiegu ubezpieczenia: 1) w ciągu ostatniego roku liczba szkód wypłaconych wyniosła: 0, łączna suma wypłat: 0,00 PLN 2) w ciągu trzech ostatnich lat liczba szkód wypłaconych wyniosła: 0, łączna suma wypłat: 0,00 PLN 3. Oświadczam, że przed zawarciem umowy ubezpieczenia uzyskałem informację, że do umowy ubezpieczenia mają zastosowanie przepisy prawa polskiego.
<b>6</b>	<b>Postanowienia dodatkowe lub odmienne</b> <b>Postanowienia dodatkowe do umowy/ów ubezpieczenia stwierdzonych niniejszą Polisa</b> 1. Do OWU mających zastosowanie do niniejszej umowy ubezpieczenia / umowy ubezpieczenia obowiązkowego wprowadza się postanowienie dodatkowe w brzmieniu: Podmiotem uprawnionym w rozumieniu ustawy o pozasądowym rozpatrywaniu sporów konsumenckich, właściwym dla PZU SA do pozasądowego rozpatrywania sporów, jest Rzecznik Finansowy, którego adres strony internetowej jest następujący: <a href="http://www.rf.gov.pl">www.rf.gov.pl</a> . 2. W umowie ubezpieczenia zawartej na cudzy rachunek Ubezpieczający zobowiązany jest doręczyć Ubezpieczonemu OWU, które mają zastosowanie do tej umowy ubezpieczenia, na piśmie lub za zgodą Ubezpieczonego na innym trwałym nośniku. W przypadku umowy ubezpieczenia, w której okres ochrony ubezpieczeniowej rozpoczyna się później niż w dniu zawarcia umowy ubezpieczenia, OWU powinny być doręczone Ubezpieczonemu przed objęciem go ochroną ubezpieczeniową. Przekazanie OWU na trwałym nośniku, wymaga uprzedniej zgody Ubezpieczonego. Na żądanie PZU SA Ubezpieczający zobowiązany jest przedstawić dowód wykonania tej powinności.
	<b>Potwierdzam dane kontaktowe</b> "MERLION" SP. Z O.O. E-mail: Klient nie posiada Telefon: Klient nie posiada
	<b>Dodatkowych informacji udzieli:</b> Oddział PZU ul. GNIEŹNIĘSKA 7, 75-950 KOSZALIN E-mail: <a href="mailto:pbedkowski@pzu.pl">pbedkowski@pzu.pl</a> tel.: +48 502252403
	Data zawarcia umowy: 21.02.2018 r.
	<b>"MERLION" SP. Z O.O.</b> Ubezpieczający
	Podpis/pieczeń ubezpieczającego
	<b>DORADCA KLIENTA</b> <i>Marta Nowak-Piotrowiak</i> Pieczeń i podpis przedstawiciela PZU SA



**801 102 102 lub (or) +48 22 566 55 55 pzu.pl**  
w przypadku awarii lub wypadku (in case of accident)



1	<b>Okres ubezpieczenia:</b> od 22.02.2018 r. do 23.04.2018 r.	
2	<b>Ubezpieczający: "MERLION" SP. Z O.O.</b> Adres siedziby: RÓŻANA 3, 75-220 KOSZALIN E-mail: Klient nie posiada	REGON: 320654722 Telefon: Klient nie posiada
3	<b>Ubezpieczony: "MERLION" SP. Z O.O.</b> Adres siedziby: RÓŻANA 3, 75-220 KOSZALIN E-mail: Klient nie posiada	REGON: 320654722 Telefon: Klient nie posiada

**Miejsce ubezpieczenia: RÓŻANA 3, 75-212 KOSZALIN**

4	<b>I. UBEZPIECZENIE SPRZĘTU ELEKTRONICZNEGO</b>	
	<b>Zakres ubezpieczenia</b>	<b>Suma ubezpieczenia</b>
	• Sprzęt przenośny	9 200 PLN
	<b>Rozszerzenie zakresu ubezpieczenia</b>	<b>Franszyza redukcyjna</b>
	• Klauzula sprzętu przenośnego	25%, nie mniej niż 500 PLN

5	<b>Franszyzy</b>	<b>Franszyza redukcyjna</b>
	Przenośny sprzęt elektroniczny	10%, nie mniej niż 500 PLN

6 **Składka łączna: 95,39 PLN**  
**Składka została opłacona w całości**

7 **Warunki ubezpieczenia**  
1. Do umowy ubezpieczenia mają zastosowanie ogólne warunki ubezpieczenia sprzętu elektronicznego, ustalone uchwałą Zarządu PZU SA nr UZ/9/2016 z dnia 15.01.2016 r.

8 **Postanowienia dodatkowe**  
8 **Ubezpieczenie mienia**  
1. Integralną część niniejszej umowy stanowi wykaz ubezpieczonych składników majątku.

9 **Oświadczenia**  
1. Oświadczam, że przed zawarciem niniejszej umowy otrzymałem ogólne warunki ubezpieczenia sprzętu elektronicznego, ustalone uchwałą Zarządu PZU SA nr UZ/9/2016 z dnia 15.01.2016 r.  
2. Oświadczam, że przed zawarciem umowy ubezpieczenia uzyskałem informację, że do umowy ubezpieczenia mają zastosowanie przepisy prawa polskiego.

10 **Postanowienia dodatkowe lub odmienne**  
**Postanowienia dodatkowe do umowy/ów ubezpieczenia stwierdzonych niniejszą Polisa**  
1. Do OWU mających zastosowanie do niniejszej umowy ubezpieczenia / umowy ubezpieczenia obowiązkowego wprowadza się postanowienie dodatkowe w brzmieniu:  
Podmiotem uprawnionym w rozumieniu ustawy o pozasądowym rozpatrywaniu sporów konsumenckich, właściwym dla PZU SA do pozasądowego rozpatrywania sporów, jest Rzecznik Finansowy, którego adres strony internetowej jest następujący: www.rf.gov.pl.  
2. W umowie ubezpieczenia zawartej na cudzy rachunek Ubezpieczający zobowiązany jest doręczyć Ubezpieczonemu OWU, które mają zastosowanie do tej umowy ubezpieczenia, na piśmie lub za zgodą Ubezpieczonego na innym trwałym nośniku. W przypadku umowy ubezpieczenia, w której okres ochrony ubezpieczeniowej rozpoczyna się później niż w dniu zawarcia umowy ubezpieczenia, OWU powinny być doręczone Ubezpieczonemu przed objęciem go ochroną ubezpieczeniową. Przekazanie OWU na trwałym nośniku, wymaga uprzedniej zgody Ubezpieczonego. Na żądanie PZU SA Ubezpieczający zobowiązany jest przedstawić dowód wykonania tej powinności.

**Dodatkowych informacji udzieli:**  
Oddział PZU  
ul. GNIEŹNIŃSKA 7, 75-950 KOSZALIN  
E-mail: pbedkowski@pzu.pl tel.: +48 502252403

Data zawarcia umowy: 21.02.2018 r.

"MERLION" SP. Z O.O. Ubezpieczający	 Podpis/pieczęć ubezpieczającego	 Pieczęć i podpis przedstawiciela PZU SA
--	-------------------------------------	---



801 102 102 lub (or) +48 22 566 55 55 pzu.pl  
w przypadku awarii lub wypadku (in case of accident)

DSP/P/1028522345/1875/pc:100000140947252/BE20