AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Implementacja serwisu obsługującego opaskę lokalizacyjną opartą na platformie Arduino i module GPS/GSM

Wykonał: Tymoteusz Przyłucki
Promotor: dr inż. Piotr Kustra
Recenzent: prof. dr hab. inż. Mirosław Głowacki
Wydział Inżynierii Metali I Informatyki Przemysłowej

Miejsce i data prezentacji: 24.01.2019 Kraków

Cel pracy

Celem pracy jest stworzenie oprogramowania obsługującego urządzenia lokalizacyjne oparte na platformie Arduino z modułem GPS/GSM, które pozwoli na określenie przybliżonej lokalizacji osoby w określonym czasie.

Do zrealizowania głównego celu pracy postawiono następujące cele szczegółowe:

- Opracowanie aplikacji wspierającej opaskę lokalizacyjną opartej na platformie Arduino,
- Implementacja serwisu obsługującego opaskę lokalizacyjną,
- Opracowanie intuicyjnego interfejsu,
- Walidacja i testowanie oprogramowania,
- Zminimalizowanie kosztów obsługi serwisu poprzez użycie maksymalnie dużej ilości wolnego oprogramowania.

MQTT - Message Queue Telemetry Transport

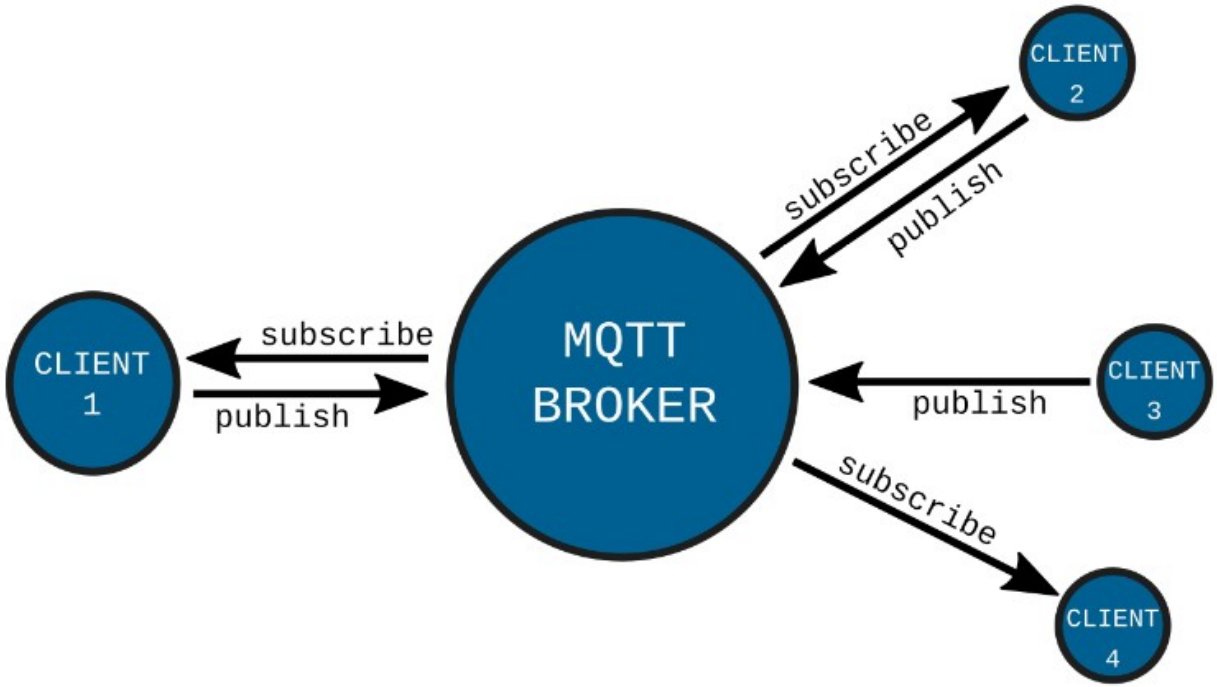
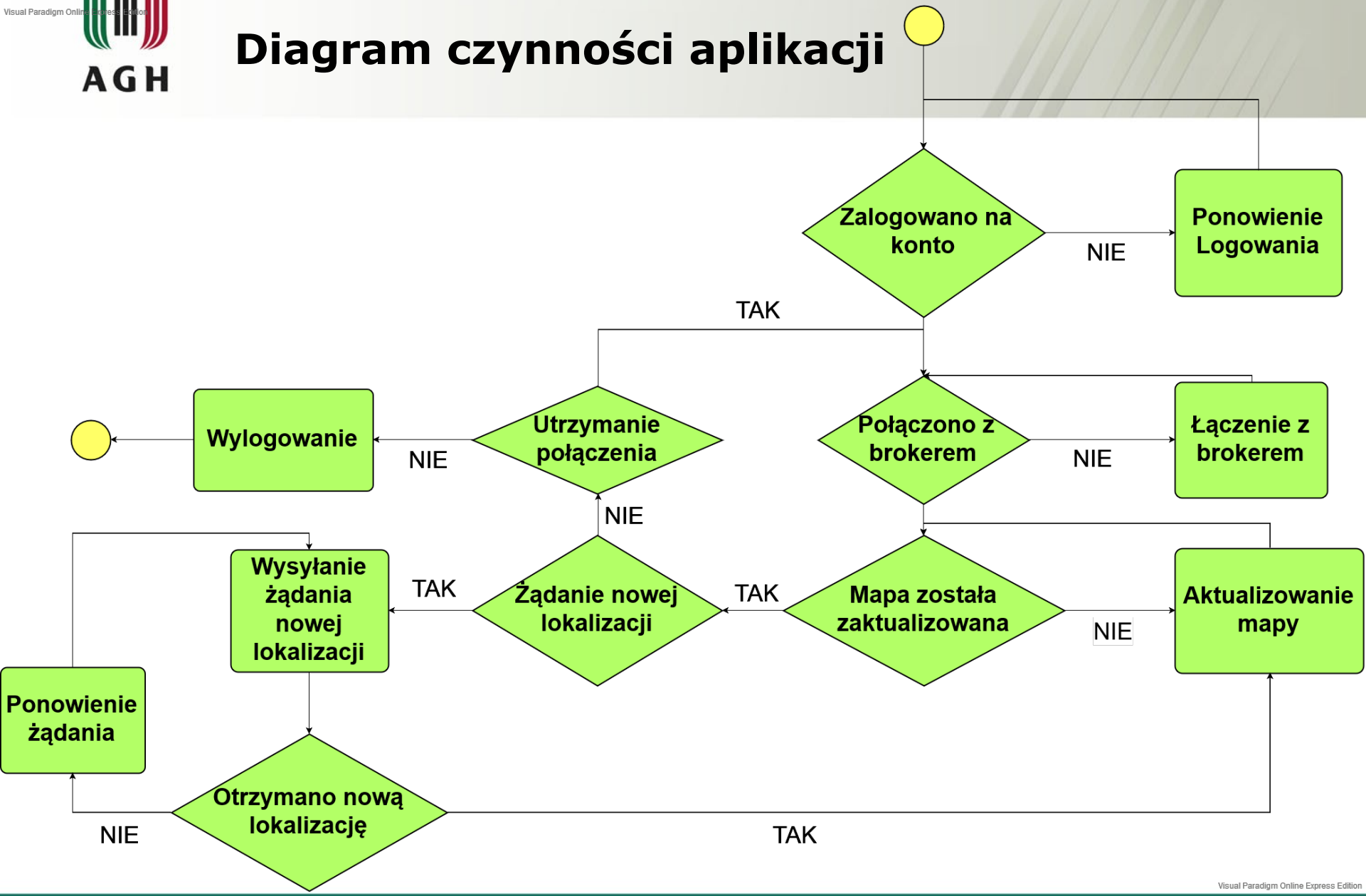


Diagram czynności aplikacji



Połączenie aplikacji z brokerem

```
client_username = request.user.username
client = mqtt.Client(client_username)
client.on_connect = on_connect
client.on_log = on_log
client.on_disconnect = on_disconnected
client.username_pw_set(broker_user_name,password_to_server)
client.connect(broker_host,int(broker_port))
client.on_message = on_message
client.loop_start()
client.subscribe("FollowBand/00:11:22:33:44:55",0)
time.sleep(1)
client.loop_forever()
client.disconnect()
```

Połączenie aplikacji z brokerem

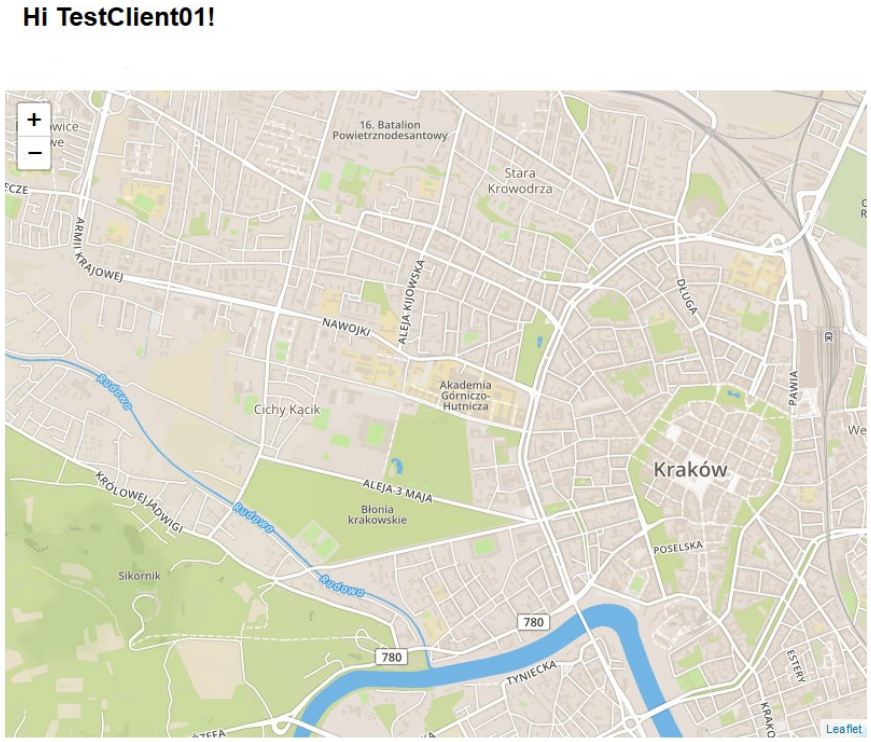
```
def on_message(client, userdata, msg):
    topic=msg.topic
    m_decode = str(msg.payload.decode("utf-8","ignore"))
    print("Message received: ",m_decode)
    m_decode = re.findall(r'\d+',m_decode)
    user = User.objects.get(username=username)
    temporary_localization = Localization(client = user ,posX =
        m_decode[0], posY = m_decode[1])
    temporary_localization.save()
    client.loop_stop()
```

Zaznaczanie pozycji na mapie

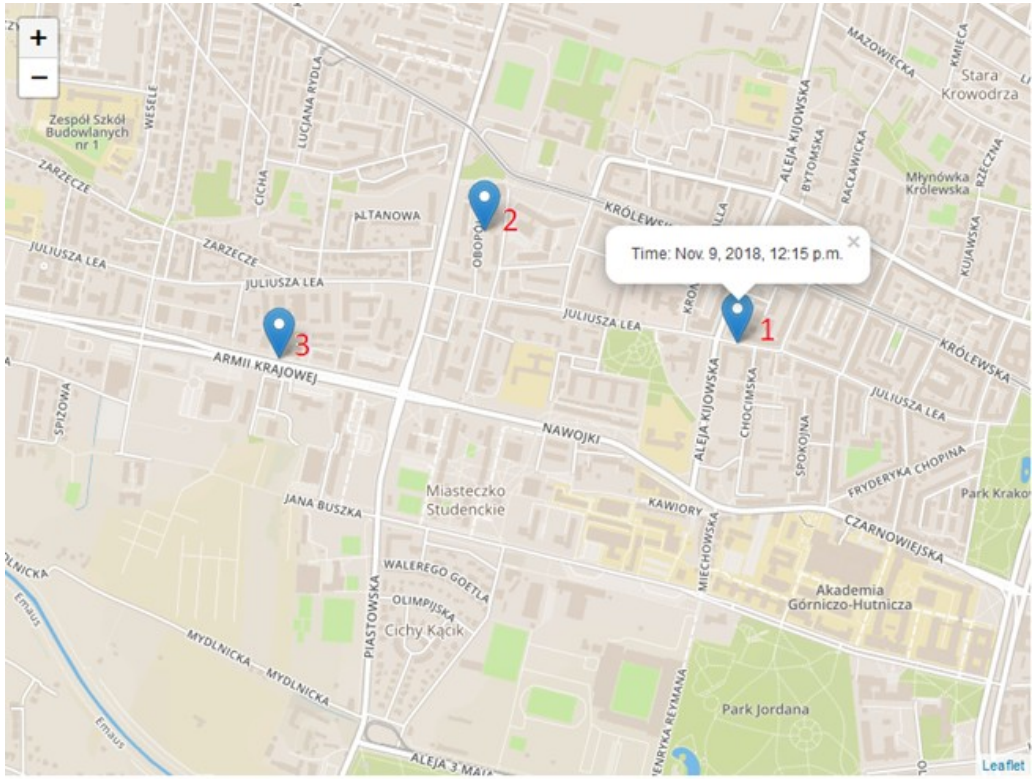
```
def localizationView(request):
    localizations = Localization.objects.all()
    context = {'localizations': localizations,}
return render(request, "detail.html", context)
-----------------------------------------------------------------------
{% for localization in localizations reversed %}
{% if user == localization.client %}
<hr>

<h1>{{ localization.posX }}</h1>
<h1>{{ localization.posY }}</h1>
<h1><button
onclick='add_sign_to_map("{{ localization.posX }}","{{ localization.posY }}
","{{ localization.created_date }}")'>{{
localization.created_date}}</button></h1>

{% endif %}
{% endfor %}
<hr>
{% endif %}
{% endblock %}
```

Zaznaczanie pozycji na mapie

```
<div id="mapid" style="width: 800px; height:
600px;"></div>
<script>
  var posX;
  var posY;
  var mymap = L.map('mapid')
         .setView([50.042382,19.930133], 13);

    function add_sign_to_map(x,y,time){
        mymap.setView([x,y],13);
        L.marker([x, y]).addTo(mymap)

         .bindPopup("Time: " +
                    time).openPopup();

        L.tileLayer(layer, {
            maxZoom: 18,
            id:
'mapbox.streets'}).addTo(mymap);
</script>
```



Zaznaczanie pozycji na mapie



3	2	1
50.070766	50.073511	50.071101
19.899859	19.906718	19.915182
Dec. 9, 2018, 12:27 p.m.	Dec. 9, 2018, 12:34 p.m.	Nov. 9, 2018, 12:15 p.m.

Wnioski

- Opracowano aplikację wspierającą opaskę lokalizacyjną opartą na platformie Arduino,
- Zaimplementowano serwis obsługujący opaskę lokalizacyjną,
- Opracowano intuicyjny interfejs użytkownika,
- Zwalidowano i przetestowano oprogramowanie,
- Zminimalizowano koszty obsługi serwisu poprzez użycie maksymalnie dużej ilości wolnego oprogramowania.

Dziękuję za uwagę