

POLITECHNIKA KRAKOWSKA  
WYDZIAŁ FIZYKI, MATEMATYKI I  
INFORMATYKI STOSOWANEJ

---

Kierunek: Informatyka

**PROJEKT  
SYSTEMY WBUDOWANE**

**Internetowa kamera klient-serwer z wykorzystaniem telefonu z systemem Android (serwer) oraz komputera lub innego urządzenia posiadającego przeglądarkę z obsługą protokołu HTTP (klient).**

**Autorzy:**

Konrad Zapała  
Tomasz Ziębiec

Kraków 2012

# **1. Wstęp.**

## **1.1. Założenia projektu**

Aplikacja powinna być napisana pod telefony z systemem Android. Powinna działać jako serwer, natomiast komputer który się łączy z telefonem jako klient. Telefon na podstawie swojej kamery internetowej generuje obraz, który następnie jest wysyłany przez sieć Wifi lub LAN do komputera, a tam w przeglądarce, obraz ten jest odbierany i odczytywany przez klientów.

## **1.2. Środowisko pracy.**

Pracę pisaliśmy w środowisku Eclipse, które świetnie współpracuje z Software Development Kit (SDK) Androida (środowisko to jest polecane przez większość developerów aplikacji na system Android). Telefon jaki posłużył nam do testowania to HTC, na którym zainstalowany był system Android w wersji 2.1 tzw. „Eclair”. Dodatkowo jeszcze w celu lepszej współpracy aplikację umieściliśmy na serwerze github.com jako zdalne repozytorium. Jest tutaj pewne ograniczenie, ponieważ w wersji darmowej można tworzyć repozytoria jednak są one publiczne (każdy może mieć wgląd w naszą pracę oraz może prześledzić jak aplikacja powstawała na każdym kolejnym etapie). Dzięki repozytorium mogliśmy skorzystać z bardzo cennej sobie w branży IT: Systemu Kontroli Wersji (svn). Tutaj z pomocą przyszło nam narzędzie GIT (uważane w ostatnim czasie za lepszą wersję svn-a). Dokumentację natomiast sporządziliśmy w programie OpenOffice, wersja finalna natomiast zostanie udostępniona jako plik z rozszerzeniem pdf.

## **1.3. Technologie wykorzystane w aplikacji.**

Jak wiemy natywnym językiem programowania jaki występuje na platformach korzystających z Android-a jest język Java. Oprócz tego Google udostępnił także dla swojego systemu kilka innych narzędzi dla programistów bezpośrednio związanych z możliwościami jakie dają nam telefony. Dlatego też jądro (serwer aplikacji oraz obsługa aparatu) stworzyliśmy w technologii Java razem z Software Development Kit (SDK) Androida.

Nad wyglądem aplikacji po stronie serwera (telefonu), korzystaliśmy dodatkowo z XML-a, standard jaki Google wymaga w aplikacjach.

Po stronie klienta posłużyliśmy się technologiami HTML, CSS oraz JavaScript. HTML i CSS do wyglądu zewnętrznego (po stronie przeglądarki), JavaScript natomiast przy odbieraniu zdjęcia z serwera.

## **1.4. Ramowy plan wykonania pracy w poszczególnych etapach.**

- 1.** Założenie repozytorium dla aplikacji na githubie (śledzenie zmian w aplikacji);
- 2.** Napisanie serwera obsługującego komunikację między telefonem, a klientami korzystającymi z przeglądarek internetowych:
  - a) uruchomienie serwera przyciskiem „start server”;
  - b) próba wysłania tekstu „Hello World” do klienta;
  - c) wysłanie obrazka i innych plików z folderu assets (folder ten przechowuje pliki, które wykorzystuje aplikacja) przez serwer i odczyt po stronie klienta;

### 3. Obsługa kamerki/aparatu w telefonie:

- a) obsługa aparatu w telefonie;
- b) zapis/odczyt do wewnętrznego bufora pamięci, ewentualnie w celu sprawdzenia wyświetlanie obrazka w telefonie;
- c) wysłanie obrazka odczytanego do klienta;

### 4. Odczytywanie z serwera (telefonu) odpowiedniego obrazu i wysłanie do klienta:

- a) próba odczytu poprzez metodę GET w przeglądarce oraz za pomocą skryptu JavaScript;
- b) „klatkowanie” zdjęć z odpowiednim czasem i prezentacja po stronie klienta;

### 5. Dopracowanie ostatecznej wersji aplikacji, przygotowanie dokumentacji oraz przedstawienie projektu;

## 2. Budowa aplikacji.

### 2.1. *Plik AndroidManifest.xml*

Naszym zdaniem jest to jeden z głównych plików naszej aplikacji. Bez niego niestety ale nasza aplikacja nie działała by właściwie. Jest tak ponieważ Google wprowadziło politykę, że każda aplikacja, jeżeli w jakiś sposób ingeruje w zasoby użytkownika systemu, to tylko poprzez dodanie odpowiedniego wpisu, może cokolwiek zmienić za jego zgodą. Dzięki temu jest zachowane pewne bezpieczeństwo dla użytkownika. Przedstawiamy zatem najważniejsze zezwolenia:

```
<uses-permission android:name="android.permission.INTERNET" />
```

To zezwolenie odpowiada za możliwość korzystania z internetu przez naszą aplikację.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

To zezwolenie pozwala nam na dostęp do informacji na temat połączeń nawiązanych z telefonem.

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

Tutaj korzystamy z zezwolenia na sprawdzenie czy łączymy się telefonem przez WiFi.

```
<uses-permission android:name="android.permission.CAMERA" />
```

Oraz ostatnie zezwolenie na korzystanie z aparatu znajdującego się w telefonie.

### 2.2. *Wygląd aplikacji, plik main.xml*

W pliku main.xml znajdują się komponenty z których korzystamy w naszej aplikacji. My skorzystaliśmy z następujących:

- 2 pola tekstowe (TextView) do wyświetlenia podstawowych informacji na temat aplikacji;
- 2 przyciski (Button-y), jeden do uruchamiania serwera i zatrzymywania go, a drugi do uruchamiania aparatu w telefonie;
- jeden komponent SurfaceView który odpowiada za obsługę zdarzenia „pstrykania” zdjęcia (jest on ukryty u nas w aplikacji);

- ostatni komponent do wyświetlenia aktualnie przechwyconego zdjęcia czyli `ImageView`.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/wifiInfo"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/button1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/button2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="fill_parent"
        android:layout_height="134dp"
        android:layout_weight="0.59" />

    <SurfaceView
        android:id="@+id/surfaceView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</LinearLayout>
```

////////////////////////////////////

### 3. Problemy jakie wystąpiły podczas pisania projektu:

- Przesłanie statycznych plików z folderu assets (już działa).
- Wysłanie obrazka z kamerki (już działa).
- Odbiór po stronie klienta, testy ze skryptem JavaScript (już działa).
- „Klatkowanie” obrazków na serwerze (działa na wirtualu);
- Przesył i odczyt kilku obrazków po stronie klienta;

#### **4. Postępy:**

- a) repozytorium: <https://github.com/konri1990/ProjektSystemyWbudowane>;
- b) konfiguracja środowiska eclipse z SDK Androida;
- c) założenie przykładowego serwera (z buttonem start/stop) na androidzie
- d) próba wysłania przykładowego tekstu przez protokół HTTP (Hello Android),
- e) wysłanie pliku css i dwóch obrazków (wcześniej zapisanych statycznie w folderze assets) oraz klatkowanie ich po stronie klienta;
- f) podstawowa obsługa aparatu w telefonie;
- g) odebranie zdjęcia (jednego) z telefonu po stronie klienta;