

Міністерство освіти і науки України  
Харківський національний Університет радіоелектроніки

Кафедра «Програмна інженерія»

**ЗВІТ**  
з лабораторної роботи № 3  
з дисципліни «Архітектура програмного забезпечення»  
на тему: «Розробка Smart Device частини програмної системи»

Виконав:

ст. гр. ПЗП-20-3

Кулешов Микита

Перевірив:

доц. каф. ПІ

Побіженко І. О.

Харків 2023

## **ВСТУП**

Метою лабораторної роботи є розробка Smart Device частини проекту за темою: «Програмна система для автоматизації побудови розкладу та відстеження відвідування занять в навчальних закладах».

Хід лабораторної роботи:

- 1) опис компонентів системи;
- 2) опис прецедентів системи;
- 3) опис взаємодії девайсу з іншими частинами системи;
- 4) опис діяльності системи.

## 1 ДІАГРАМА КОМПОНЕНТІВ СИСТЕМИ

На рисунку 1.1 наведено діаграму компонентів, що описує архітектуру програми.

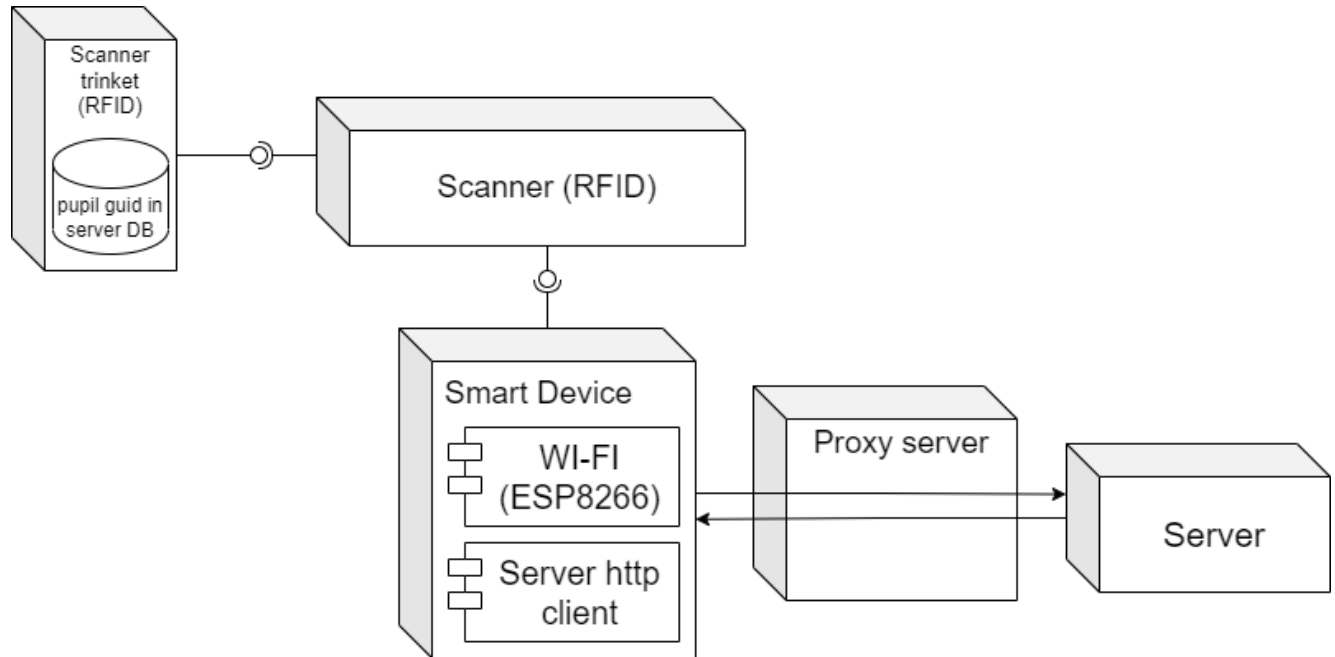


Рисунок 1.1 – Діаграма компонентів

Для створення програмного забезпечення було обрано наступні інструменти розробки:

- Arduino Rev 3 WIFI з вбудованим ESP8266 модулем для виконання, розгортання програмного коду, а також для відправки HTTP запитів.
- Для сканування даних використовуються RFID сканер та RFID чіп.
- Для вирішення проблем з надсиланням запитів у локальній мережі використовується проксі-сервер, написаний на node.js.

Як приклад, код сервісу, що відповідає за відправку запитів на сервер наведено у додатку А. Повний проект завантажено до хмарного сховища Google Disk [1].

## 2 ДІАГРАМА ПРЕЦЕДЕНТІВ

В використанні девайсу є лише один актор - учень. Щоб наглядно показати наявний функціонал та його доступність в залежності від ролі, було створено діаграму прецедентів (див. рис. 2.1).

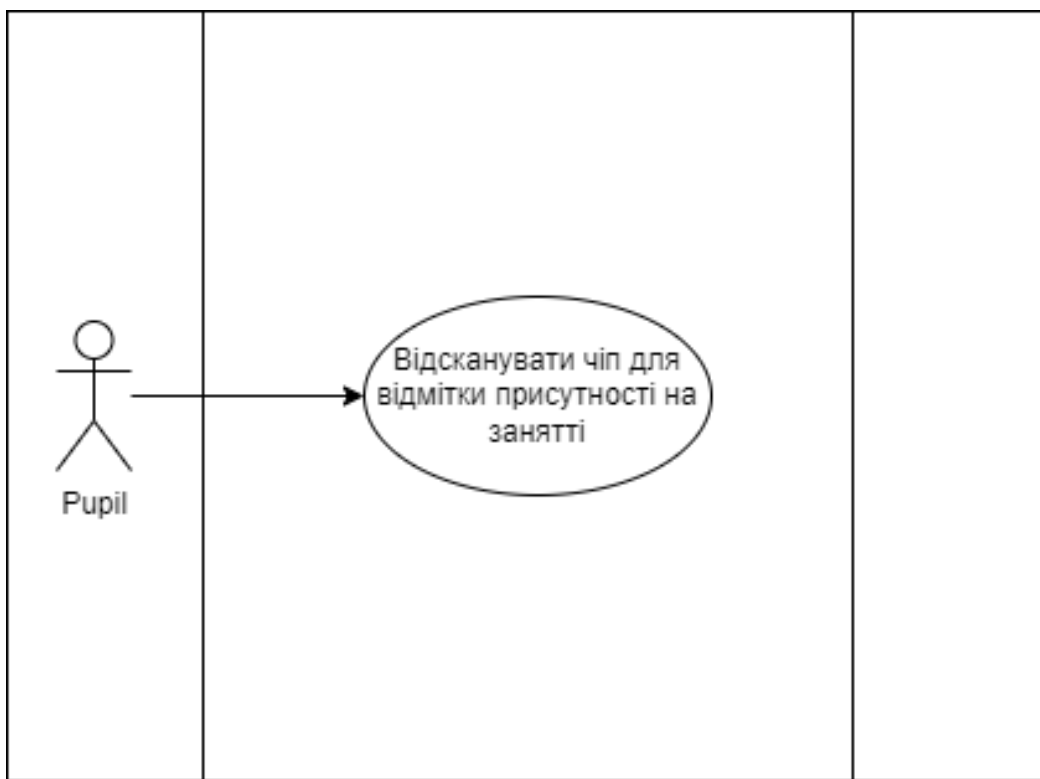


Рисунок 2.1 – Діаграма прецедентів

Головна мета девайсу – зчитати інформацію з піднесеного до сканера чіпу та відправити її на сервер.

### 3 ДІАГРАМА ВЗАЄМОДІЇ

На рисунку 3.1 представлена діаграма взаємодії, що ілюструє яким чином частина смарт девайсу спілкується із серверною та мобільною частинами.

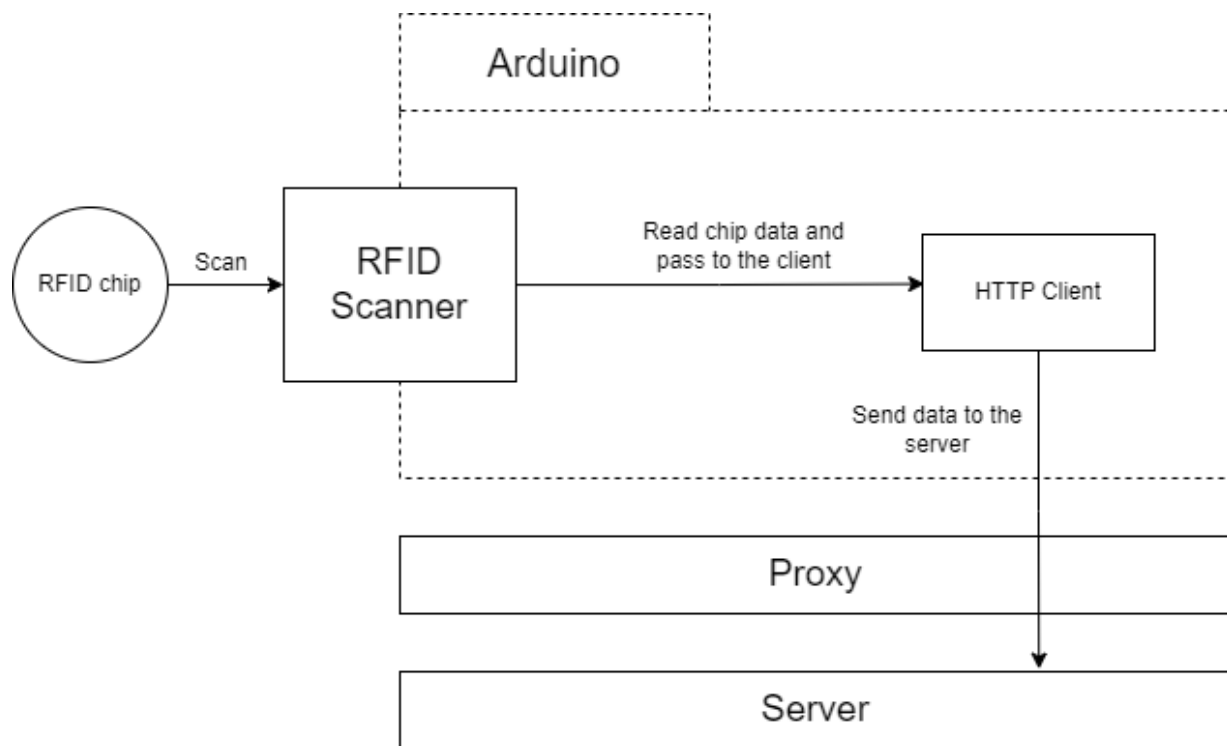


Рисунок 3.1 – діаграма взаємодії клієнтської частини із серверною

До смарт девайсу приєднано сканер, який постійно очікує на вхідні дані, які зберігаються у чіпі, який індивідуальний для кожного учня. При піднесенні чіпа до сканера, другий зчитує необхідну інформацію та передає її до клієнтського коду, який відправляє цю інформацію на сервер.

## 4 ДІАГРАМА ДІЯЛЬНОСТІ

На рисунку 4.1 представлена діаграма діяльності смарт девайс частини для випадку коли до сканера подноситься чіп.

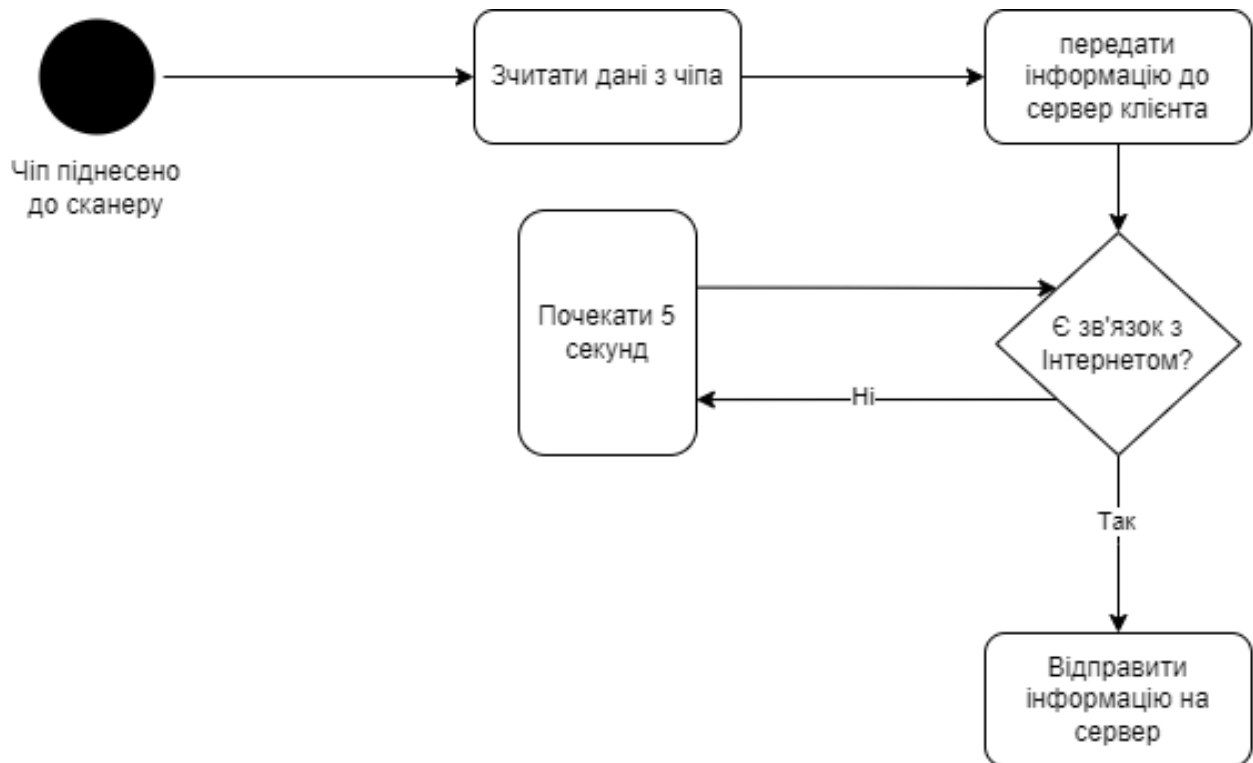


Рисунок 4.1 – діаграма пакетів клієнтської частини

При кожному піднесенні чіпу до сканера, з нього зчитується інформація та передається в смарт девайс на обробку. Перед відправкою на сервер іде перевірка чи девайс підключено до Інтернету, якщо ні, то девайс чекає секунд та робить повторну перевірку. Цей цикл продовжується до тих пір поки підключення не буде встановлено.



## **ВИСНОВКИ**

В ході виконання лабораторної роботи була розроблена smart device частина до проекту на тему «Програмна система для автоматизації побудови розкладу та відстеження відвідування занять в навчальних закладах».

Представлені діаграм компонентів, прецедентів, взаємодії та діяльності. Для кожної з них надано текстове пояснення окремих рішень.

Запис з демонстрацією роботи серверної частини можна подивитись за посиланням до хмарного сховища Google Disk [2].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Код проекту. Google Disk. URL:  
[https://drive.google.com/drive/folders/1rsjPAMyISp\\_3pC6S-logbB4lSajXxhW6?usp=sharing](https://drive.google.com/drive/folders/1rsjPAMyISp_3pC6S-logbB4lSajXxhW6?usp=sharing) (дата звернення: 26.05.2023).
2. Відео демонстрація роботи. Google Disk. URL:  
[https://drive.google.com/drive/folders/1rsjPAMyISp\\_3pC6S-logbB4lSajXxhW6?usp=sharing](https://drive.google.com/drive/folders/1rsjPAMyISp_3pC6S-logbB4lSajXxhW6?usp=sharing) (дата звернення: 26.05.2023).

## ДОДАТОК А

Код сервер клієнта, який відповідає за відправку даних на сервер.

```
1. /**
   a. BasicHTTPClient.ino

   b. Created on: 24.05.2015

2. */

3. #include <Arduino.h>

4. #include <ESP8266WiFi.h>
5. #include <ESP8266WiFiMulti.h>

6. #include <ESP8266HTTPClient.h>

7. #include <WiFiClient.h>

8. ESP8266WiFiMulti WiFiMulti;

9. void setup() {

10.     Serial.begin(115200);
11.     // Serial.setDebugOutput(true);

12.     Serial.println();
13.     Serial.println();
14.     Serial.println();

15.     for (uint8_t t = 4; t > 0; t--) {
16.         Serial.printf("[SETUP] WAIT %d...\n", t);
17.         Serial.flush();
18.         delay(1000);
19.     }

20.     WiFi.mode(WIFI_STA);
21.     WiFiMulti.addAP("Xiaomi_3F1D", "12345678");
22.     }

23.     String proxyApiUrl = "http://192.168.31.238:3001";

24.     void loop() {
25.         // wait for WiFi connection
26.         if ((WiFiMulti.run() == WL_CONNECTED)) {

27.             WiFiClient client;

28.             HTTPClient http;
```

```

29.     Serial.print("[HTTP] begin...\n");
30.     if (http.begin(client, proxyApiUrl + "/attendance")) { //
    HTTP

31.     Serial.print("[HTTP] POST...\n");
32.     // start connection and send HTTP header
33.     int httpCode = http.POST();

34.     // httpCode will be negative on error
35.     if (httpCode > 0) {
36.     // HTTP header has been send and Server response header has
    been handled
37.     Serial.printf("[HTTP] POST... code: %d\n", httpCode);

38.     // file found at server
39.     if (httpCode == HTTP_CODE_OK || httpCode ==
    HTTP_CODE_MOVED_PERMANENTLY) {
40.     String payload = http.getString();
41.     Serial.println(payload);
42.     }
43.     } else {
44.     Serial.printf("[HTTP] POST... failed, error: %s\n",
    http.errorToString(httpCode).c_str());
45.     }

46.     http.end();
47.     } else {
    i. Serial.println("[HTTP] Unable to connect");
48.     }
49.     }

50.     delay(10000);
51.     }

```