

APLIKACJA DO SPRAWDZANIA OBECNOŚCI STUDENTÓW ZA POMOCĄ LEGITYMACJI

Marcin Boiński, Piotr Kotlarz

Uniwersytet Kazimierza Wielkiego
Wydział Informatyki
ul. Mikołaja Kopernika 1, 85-074 Bydgoszcz
e-mail: marcin.boinski@student.ukw.edu.pl

Streszczenie: W pracy przedstawiono system składający się ze skanera legitymacji studenckich oraz aplikacji do zarządzania obecnościami studentów. Ma on rozwiązać problem niejednorodności i czasochłonności procesów weryfikacji obecności w trakcie zajęć. Porównany został z dwoma podobnymi rozwiązaniami funkcjonującymi na rynku. Wykorzystano wiele technologii, języków programowania i bibliotek. W wyniku powstał nowoczesny, przyjazny dla użytkownika system zarządzania obecnościami.

Słowa kluczowe: MySQL, PHP, Arduino, MQTT, C#, .NET, Windows Forms, skaner, RFID

APPLICATION FOR CHECKING STUDENT ATTENDANCE USING STUDENT ID CARDS

Abstract: The paper presents a system consisting of a student ID card scanner and an application for managing student attendance. It is designed to solve the problem of inconsistency and time-consuming processes of attendance verification during classes. It was compared with two similar solutions available on the market. Many technologies, programming languages, and libraries were used. The result is a modern, user-friendly attendance management system.

Keywords: MySQL, PHP, Arduino, MQTT, C#, .NET, Windows Forms, scanner, RFID

1. WSTĘP

Ważnym elementem współczesnych procesów dydaktycznych, zarazem w szkolnictwie wyższym, jak i na niższych szczeblach edukacji, jest monitorowanie obecności uczniów. Zwiększanie frekwencji pozytywnie wpływa na ich wyniki w nauce. Metody oparte na ręcznym sporządzaniu list obecności są niejednolite i czasochłonne. W praktyce zaobserwować można dwa różne typy list: papierowe oraz elektroniczne. Mimo różnych wad oraz zalet obu podejść, łączy je wymóg nakładu pracy w tworzeniu i ich wypełnianiu.

Odpowiedzią na te problemy jest rozwiązanie, które usprawni proces kontroli frekwencji poprzez jego ujednoczenie. Proponowanym podejściem jest zautomatyzowanie procedury poprzez opracowanie aplikacji wspomagającej działania prowadzących zajęcia wykorzystującej skaner legitymacji studenckich. Umożliwi to zarówno ujednoczenie procedur, jak i zaoszczędzenie cennego czasu. Sprawi to również, że dane o frekwencji będą bardziej dokładne i wiarygodne. Nauczyciele

zostaną odciążeni z monotonnego procesu sprawdzania obecności, który zostanie przyspieszony i stanie się bardziej przyjazny zarówno dla studentów, jak i prowadzących zajęcia.

2. NARZĘDZIA

Rozwiązanie powstało dzięki połączeniu wielu różnych języków programowania, bibliotek czy urządzeń.

Skaner legitymacji składa się z modułów Arduino ESP8266 oraz RFID-RC522. Umożliwiają one skanowanie chipów zawartych w legitymacjach studenckich, odczytanie ich UID (Unique Identifier) oraz komunikację z bazą danych i aplikacją komputerową poprzez sieć WiFi. Użyto również kilku zewnętrznych bibliotek [3-6].

Dane w projekcie przechowywane są w bazie danych utworzonej przy pomocy MySQL. Do wykonywania poleceń w bazie pośredniczą skrypty napisane w języku PHP [13-15].

Aplikacja komputerowa wykorzystano obiektowy język programowania C# wraz z .NET Framework.

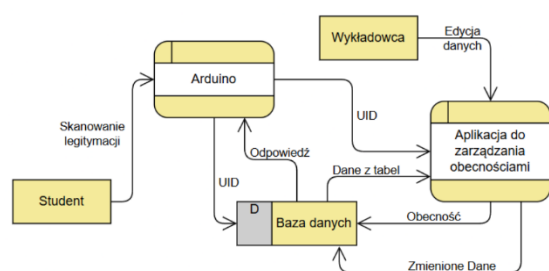
Odpowiada on za całą logikę aplikacji. Do stworzenia interfejsu wykorzystano Windows Forms ze względu na spory zbiór gotowych kontrolki i wygodę korzystania [7-8]. Dodatkowo wykorzystano pakiety NuGet m. in. do poprawienia wyglądu aplikacji czy obsługi formatu JSON [9-12].

Komunikacja między skanerem legitymacji i aplikacją komputerową odbywa się za pośrednictwem protokołu MQTT. Jest to bardzo oszczędne rozwiązanie działające w modelu publikacji i subskrybentów. Urządzenia nie komunikują się bezpośrednio, tylko poprzez centralny broker, który odbiera publikowane dane i rozsyła do subskrybentów danego tematu.

Wszystkie te elementy ściśle ze sobą współpracują, zapewniając sprawne działanie rozwiązania.

3. OPIS ROZWIĄZANIA

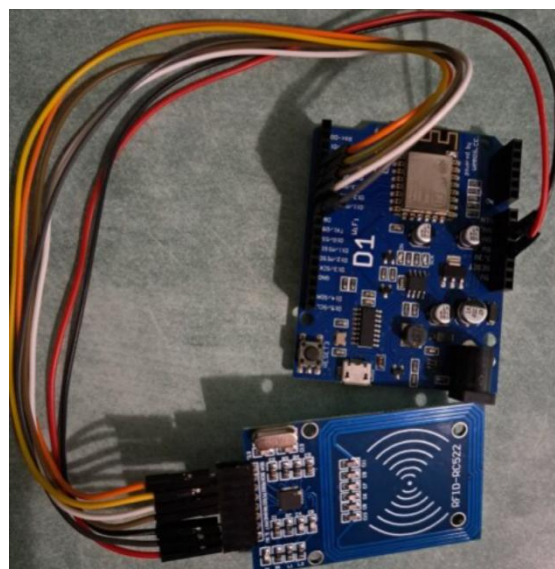
Rozwiązanie składa się ze skanera legitymacji, aplikacji do zarządzania frekwencją, bazy danych i brokera do komunikacji między skanerem i aplikacją. Student przykłada swoją legitymację do skanera, który odczytuje jej UID, a następnie przesyła je do bazy danych. Jeśli identyfikator nie istnieje w bazie danych, jest do niej dodawany. Jeśli nie, UID jest przesyłane dalej przez brokera MQTT do aplikacji, aby ta zaznaczyła obecność. Wykładowca ma możliwość edytowania danych z bazy, np. tworzenia nowych list obecności czy ręcznej zmiany obecności studenta.



Rysunek 1 Diagram przepływu danych

Powyższy diagram przedstawia sposób komunikacji poszczególnych elementów rozwiązania. Pierwszy element systemu, czyli skaner legitymacji, składa się z modułu skanującego chipy RFID legitymacji oraz z modułu Arduino, który odbiera i przesyła dalej unikalny identyfikator skanowanej legitymacji. Dzieje się to bezprzewodowo poprzez sieć WiFi, z którą ESP8266 musi się połączyć, aby poprawnie działać. Obudowa skanera powstała w technice druku 3D. Program mikrokontrolera decyduje, czy przesłać dane do aplikacji na podstawie odpowiedzi od bazy danych, która zależy

od tego, czy UID jest już w bazie, czy nie. Oba moduły skanera pokazano na rysunku 2.



Rysunek 2 Elementy skanera legitymacji

Baza danych stworzona została przy pomocy MySQL. Przechowuje ona kluczowe dane potrzebne do działania projektu, jak UID legitymacji, dane studentów czy listy obecności. Zawiera ona także dane pomocnicze, które pomagają symulować działanie rozwiązania w realnych warunkach. Są to dane m.in. nauczanych przedmiotów, wykładowców czy grup zajęciowych. Komunikacja między bazą a aplikacją i skanerem odbywa się za pośrednictwem skryptów PHP, które łączą się z bazą i wykonują polecenia takie jak *INSERT*, *UPDATE* czy *DELETE*.

Komunikacja między aplikacją a skanerem odbywa się za pośrednictwem brokera MQTT. Na potrzeby symulacji działania systemu, uruchamiany jest przez aplikację komputerową, jednak w realnym środowisku powinien on działać na serwerze. Odbiera on dane od publisherów (skaner) i przesyła do subskrybentów tematu (aplikacja).

Aplikacja komputerowa powstała przy użyciu języka C# z .NET Framework oraz Windows Forms. Jej głównym zadaniem jest umożliwienie zarządzania listami obecności przez prowadzących zajęcia. Umożliwia ona także modyfikowanie pozostałych danych, jak dane studentów, wykładowców czy grup zajęciowych. Aplikacja podzielona jest na podstrony dostępne poprzez przyciski w menu bocznym. Interfejs aplikacji pokazano na rysunku 3.



Rysunek 3 Ekran główny aplikacji komputerowej

Modułowy sposób wykonania pozwala na łatwą i tanią skalowalność systemu. Koszt obu modułów skanera jest niewielki, broker i baza danych mogą z łatwością obsługiwać więcej urządzeń. Elementy rozwiązania mogą być rozwijane i modyfikowane niezależnie od siebie nawzajem.

4. PODSUMOWANIE I WNIOSKI

Opracowany system umożliwia szybką i wygodną weryfikację tożsamości oraz automatyczne zaznaczanie obecności w bazie danych. Możliwe jest to dzięki skanerowi legitymacji, który odczytuje UID i przesyła je poprzez broker MQTT do aplikacji, na której obecność danego studenta jest zaznaczana. Opracowane rozwiązanie znacząco przyspiesza proces sprawdzania frekwencji i ogranicza możliwość fałszowania obecności. Każdy z elementów rozwiązania można usprawnić i rozwinąć aby poprawić doświadczenie korzystania z niego przez użytkowników. Mogłyby to być m.in. dodatkowe zabezpieczenia przed próbami ataków hakerskich, aplikacja dla studentów, aby ci mogli sprawdzać własną frekwencję czy usprawnienie działania systemu, aby uniknąć awarii.

Literatura

1. <https://opteam.pl/rozwiwania-it/szkolnictwo-wyzsze/elektroniczna-lista-obecnosci>
2. <https://esku.cloud>
3. Wprowadzenie do Arduino / Massimo Banzi, Michael Shiloh ; przekład: Maria Chaniewska, Marek Włodarz
4. <https://github.com/esp8266/Arduino>
5. <https://github.com/miguelbalboa/rfid>
6. <https://github.com/knolleary/pubsubclient>
7. Język C# 6.0 i platforma .NET 4.6 / Andrew Troelsen i Philip Japikse ; [przekład WITKOM Witold Sikorski; Maciej Baranowski]

8. Aplikacje Windows Forms .NET w C# / Paweł Chłosta
9. <https://github.com/JamesNK/Newtonsoft.Json>
10. <https://github.com/awesome-inc/FontAwesome.Sharp>
11. <https://github.com/dotnet/MQTTnet>
12. <https://fontawesome.com>
13. Apache: podręcznik administratora / Rich Bowen, Daniel López Ridruejo, Allan Liska; przekł. z jęz. ang. Maciej Baranowski
14. PHP, MySQL i Apache: intensywny trening / Julie C. Meloni ; [tł. Jarosław Dobrzański]
15. PHP 7: algorytmy i struktury danych / Mizanur Rahman ; [tłumaczenie Łukasz Suma]