

# System kontroli obecności na podstawie elektronicznych legitymacji studenckich

(Students attendance verifying system with the use of student card)

Dawid Szczyrk

Praca magisterska

**Promotor:** dr Jakub Michaliszyn

Uniwersytet Wrocławski  
Wydział Matematyki i Informatyki  
Instytut Informatyki

30 sierpnia 2020



Streszczenie

...



...



# Spis treści

<b>1. Wprowadzenie</b>	<b>7</b>
<b>2. Określenie wymagań</b>	<b>9</b>
2.1. Analiza obecnego sposobu sprawdzania obecności . . . . .	9
2.2. Zdefiniowanie wymagań . . . . .	10
2.2.1. Identyfikacja zajęć na których obecność była weryfikowana .	10
2.2.2. Zbieranie danych od studentów . . . . .	11
2.2.3. Wprowadzenie danych do systemu . . . . .	11
2.2.4. Weryfikacja poprawności danych . . . . .	11
2.3. Inne brane pod uwagę realizacje realizacje . . . . .	12
2.3.1. Responsywna aplikacja internetowa umożliwiająca wygodne rejestrwanie obecności studentów na zajęciach w samej apli- kacji przez internet albo telefon . . . . .	12
2.3.2. Urządzenie rejestrujące obecności za pośrednictwem sieci bez- przewodowej . . . . .	12
<b>3. Określenie struktury projektu</b>	<b>13</b>
3.1. Weryfikator - czytnik legitymacji studenckich . . . . .	13
3.1.1. Zastosowana platforma i peryferia . . . . .	14
3.1.2. Konstrukcja . . . . .	15
3.1.3. Funkcjonowanie . . . . .	15
3.1.4. Napotkane problemy . . . . .	16
3.1.5. Testowanie . . . . .	16
3.1.6. Inne koncepcje . . . . .	16

3.2. Menadżer obecności - umożliwiający katalogowanie daych . . . . .	16
3.3. Baza studentów - umożliwiająca identyfikację danych zebranych z ELS . . . . .	16
<b>4. Realizacja</b>	<b>17</b>
<b>5. Możliwości rozwoju</b>	<b>19</b>
<b>6. Problemy</b>	<b>21</b>
<b>7. Wnioski</b>	<b>23</b>

# Rozdział 1.

## Wprowadzenie

Celem niniejszej pracy jest zaprojektowanie użytecznego i niedrogiego w utrzymaniu systemu do kontroli obecności studentów na wykładzie.

Niemalże każde zajęcia na naszym instytucie zaczynają się od zebrania listy obecności. Odbywa się to w siermiężny sposób, ponieważ każdy ze studentów musi wpisać swoje dane na listę obecności. Taka forma jest uciążliwa, zajmuje kilka pierwszych minut zajęć, a studenci się rozpraszają podając sobie listę obecności. Dziś, kiedy do ochrony danych osobowych przykładą się coraz większą wagę, starodawne sposoby sprawdzania obecności są tym bardziej wątpliwe.

W dobie współczesnych zdobyczy technologicznych można uznać za zaskakujące, że żak sprzed kilku stuleci nie odczułby w procesie zbierania obecności niemal żadnej różnicy względem swojej współczesności (z tym zastrzeżeniem, że uczelniany system przechowujący dane byłby fizycznym dziennikiem obecności). Wydawać by się mogło, że obecne podejście jest tak zakorzenione w akademickiej kulturze, że nie ma możliwości tego zmienić. W niniejszej pracy chciałbym podjąć tę rękawicę i spróbować zaproponować rozwiązania które mogłyby cały proces znacznie usprawnić.

Poniższa praca przedstawia proces od postawienia wymagań do realizacji finalnej wersji wielokomponentowego projektu. Zależało mi, aby w ramach pracy magisterskiej w praktyce wykorzystać nowoczesne platformy umożliwiające realizację projektów informatycznych. Istotnym elementem w procesie wybierania przeze mnie rozwiązań, była chęć praktycznego wykorzystania wiedzy z zakresu systemów wbudowanych, dlatego - co w dalszej części pracy zostanie szerzej opowiedziane - zdecydowałem na zbudowanie własnego urządzenia na bazie platformy Arduino.





## Rozdział 2.

# Określenie wymagań

### 2.1. Analiza obecnego sposobu sprawdzania obecności

W celu doprecyzowania wymagań stawianych przed projektowanym przeze mnie system postanowiłem rozważyć znany mi z autopsji proces weryfikowania obecności uczestników zajęć na zajęciach prowadzonych przez Uniwersytet Wrocławski

Sprawdzanie listy obecności jest w dużej mierze uzależnione od preferencji prowadzącego i zwyczajów panujących na konkretnym wydziale - przykładowo obecność może być potwierdzona słownie - poprzez kolejne wywołanie osób z listy, albo pisemnie - poprzez udostępnienie listy na którą uczestnicy będą zobowiązani się wpisać.

Dla uproszczenia dalszych rozważań postanowiłem przyjąć za ogólnie stosowaną metodę pisemną - która według moich obserwacji jest stosowana częściej - i na tej podstawie wyodrębnić czynności które składają się na wypełnienie listy obecności podczas zajęć.

Zbieranie informacji o obecności studentów na wykładzie jest czynnością na pierwszy rzut oka oczywistą, po bliższym przyjrzeniu się całej procedurze, okazuje się jednak, że daje się ona rozłożyć na jeszcze prostsze elementy.

- Przygotowanie kartki z listą obecności (przypisanie listy obecności do zajęć których będzie ona dotyczyła).
- Udostępnienie kartki uczestnikom oraz wpisywanie się na listę.
- Weryfikacja poprawności zebranej listy poprzez przeliczenie osób obecnych na zajęciach.
- Przeniesienie zebranych danych do uczelnianego systemu.

Powyższy schemat zbierania listy obecności podatny jest na dodatkowe błędy. Nie trudno wyobrazić sobie sytuację w której student intencjonalnie podaje nie swoje

dane w celu zaliczenia obecności innej osobie. Kolejnym krokiem który może doprowadzić do błędów na ostatecznej liście obecności jest błąd ludzki podczas przenoszenia danych z kartki do uczelnianego systemu komputerowego. Podobny błąd trudno jest od razu wychwycić ze względu na to, że studenci zazwyczaj nie przeglądają na bieżąco list obecności dostępnych np. na usosie. Kiedy na koniec semestru obecność jest rozliczana bardzo trudno dojść do faktów.

Pierwszym elementem nadającym się w moim mniemaniu do usprawnienia jest samo tworzenie listy obecności z informacją na temat zajęć których taka lista będzie dotyczyła. Fizyczna lista, poza czasem poświęconym na jej stworzenie musi być po odbytych zajęciach przechowywana - co stwarza ryzyko jej zgubienia. Dodatkowo istnieje prawdopodobieństwo, że trafi w niepowołane ręce i ktoś nieżyczliwy wpisze na nią dodatkowe nazwiska. Do tego wszystkiego dochodzi jeszcze aspekt ekologiczny, ponieważ każda kolejna kartka to dodatkowy element wyrwany matce naturze.

Również udział studentów w rejestrowaniu własnej obecności może ulec optymalizacji. Wpisanie własnego imienia i nazwiska, a czasem numeru albumu zabiera czas, naraża na upublicznienie dane osobowe i utrudnia studentowi śledzenie wykładu.

Weryfikacja poprawności listy jest nie tylko pracochłonna, ale również podatna na błędy. Czas poświęcony na policzenie uczestników rośnie wprost proporcjonalnie do ich liczby. Dodatkowo prowadzący zajęcia może się zwyczajnie pomylić albo ktoś dopisać do listy już po tym sprawdzeniu.

Z całą pewnością usprawnienia wymaga proces przenoszenia danych z fizycznego nośnika do uczelnianego systemu komputerowego. Problemem jest nie tylko to, że podczas wpisywania obecności do komputerowego systemu można popełnić błąd, ale również to, że jest to proces czasochłonny.

## **2.2. Zdefiniowanie wymagań**

Na podstawie poszczególnych kroków w procesie zbierania obecności, wyodrębniłem cztery właściwości przez pryzmat których mogłem ocenić potencjalne rozwiązania. Dodatkowo w dalszych rozważaniach do grona kryteriów oceny poszczególnych możliwości dołączyłem jeszcze dwa istotne elementy - przewidywany koszt, oraz stopień skomplikowania rozwiązania.

### **2.2.1. Identyfikacja zajęć na których obecność była weryfikowana**

Jestem w stanie zidentyfikować dwa możliwe podejścia do tego problemu. Rozwiązanie tradycyjne realizuje to zadanie w naturalnej kolejności tzn. w momencie zbierania obecności wiemy dla jakich zajęć jest ona zbierana. W moim przekonaniu warto rozważyć opcję odwrotną, tzn. najpierw zbierać dane (pamiętając jedynie kiedy zostały zebrane), a w dalszej kolejności decydować jakich zajęć one dotyczą.

W ten sposób unikamy konieczności wprowadzenia do nośnika informacji na temat tego jakich zajęć dotyczą zebrane dane.

### 2.2.2. Zbieranie danych od studentów

W tej kwestii brałem pod uwagę trzy możliwości. Tradycyjne rozwiązanie realizowane poprzez wpisywanie się na listę może zostać zastąpione poprzez użycie bardziej nowoczesnego sposobu identyfikacji - np. wbudowaną funkcję legitymacji studenckiej tj. identyfikację zbliżeniową. Innym sposobem byłoby zastąpienie fizycznej kartki wypełnianej przez osoby uczesniczące w zajęciach cyfrowym formularzem udostępnianym przez internetową aplikację. Ostatnią rozważaną przeze mnie możliwością jest skorzystanie z istniejącego albo stworzenie własnego systemu do rozpoznawania twarzy.

### 2.2.3. Wprowadzenie danych do systemu

To zadanie może być realizowane na dwa sposoby. Pierwszym z nich jest rejestrowanie obecności w systemie od razu przy jej odnotowaniu. W tym wypadku wymagane jest bezpośrednie połączenie z systemem. Drugim - zebranie wszystkich danych na wstępie i późniejsze, zbiorcze wprowadzanie ich do aplikacji. W tym wypadku niezbędny będzie nośnik na którym dane zostaną przetransportowane, wskazane jest również zaszyfrowanie tych danych na czas transportu.

### 2.2.4. Weryfikacja poprawności danych

W przypadku tradycyjnego zbierania obecności weryfikacja jest niedoskonała, ale konieczna ze względu na to że nośnik danych - kartka z listą obecności - udostępniana jest szerokiej publiczności. Podobny problem będzie występował jeśli kartkę papieru zastąpimy cyfrowym tabletem. Możemy również odseparować medium i każdą zebraną obecność weryfikować osobno.

W kontekście powyższych rozważań zdecydowałem się na następujące założenia

- Implementowany przeze mnie system powinien być w stanie zbierać obecności bez wcześniejszego określenia w kontekście jakich zajęć jest to robione.
- Dane od studentów powinny być zbierane bez ingerencji wykładowcy i nie zaburzając porządku zajęć.
- Wprowadzanie danych powinno odbywać się w taki sposób, żeby uniknąć pomyłki.
- Student nie powinien mieć możliwości ingerencji w listę obecności.

## **2.3. Inne brane pod uwagę realizacje realizacji**

### **2.3.1. Responsywna aplikacja internetowa umożliwiająca wygodne rejestrowanie obecności studentów na zajęciach w samej aplikacji przez internet albo telefon**

W mojej ocenie to najprostsze rozwiązanie składające się z jednolitego systemu, w którym jednocześnie można zbierać obecności uczestników wykładu, oraz je przechowywać i przeglądać. Jej niewątpliwym plusem jest całkowite wyeliminowanie problemu związanego z wprowadzaniem danych do systemu katalogującego obecności, ponieważ dane wprowadzane są wprost do systemu. Identyfikacja konkretnych zajęć w ramach których rejestrowana jest obecność, odbywałaby się na początku zajęć poprzez wyklikanie w aplikacji. Obecność uczestników na zajęciach byłaby sprawdzana przez prowadzącego i od razu wprowadzana do systemu. Ze względu na swoją czasochłonność wydaje się to największym minusem tego rozwiązanie.

### **2.3.2. Urządzenie rejestrujące obecności za pośrednictwem sieci bezprzewodowej**

To rozwiązanie początkowo wydało mi się bardzo atrakcyjne i przez długi czas miałem zamiar je zrealizować. Planowałem rozszerzyć mikrokontroler o moduł do bezprzewodowej łączności z internetem i w stworzonej aplikacji udostępnić interfejs do odnotowywania obecności. Po dłuższych rozważaniach okazało się jednak, że takie rozwiązanie przerasta moje możliwości. Po pierwsze moduł do łączności bezprzewodowej okazał się kapryśny. Dodatkowo ten sposób na zbieranie danych wymagałby umożliwienia powiązania obecności z zajęciami, co pociągałoby dołączenie do urządzenia dodatkowego interfejsu który by na to pozwalał.

## Rozdział 3.

# Określenie struktury projektu

Finalnie zdecydowałem, żeby oddzielić sposób zbierania danych od ich przechowywania i przetwarzania.

Komponent umożliwiający studentowi odnotowanie swojej obecności na wykładzie - weryfikator - postanowiłem oprzeć o platformę arduino - znaną mi z zajęć systemów wbudowanych.

Rolę oprogramowania odpowiedzialnego za katalogowanie zebranych przez weryfikator danych będzie w moim systemie ogólnodostępna aplikacja internetowa, umożliwiająca założenie konta i zarządzanie zebranymi obecnościami w kontekście całego przedmiotu i w kontekście pojedynczych zajęć. Dodatkowo ma ona umożliwiać generowanie raportów ze zagregowanymi danymi, tak aby dało się łatwo prześledzić obecności w kontekście całego semestru.

Kolejnym elementem mojego systemu, niejako wymuszonym przez brak dostępu do rzeczywistych danych uniwersyteckich, jest aplikacja pozwalająca na powiązanie danych pobranych z legitymacji studenckiej z danymi pozwalającymi na jego identyfikację. Jest to aplikacja rozłączna względem systemu zarządzającego obecnościami osób uczesniczących w zajęciach, wystawiająca RESTowe api.

### 3.1. Weryfikator - czytnik legitymacji studenckich

Zgodnie z wymaganiami projektowane przeze mnie urządzenie powinno udostępniać co najmniej następujące możliwości.

- Możliwość podłączenie zewnętrznego nośnika danych.
- Stworzenie pliku na którym zapamiętani zostaną uczestnicy zajęć
- Zebranie informacji z elektronicznej legitymacji studenckiej, potwierdzające obecność uczesników
- Wyświetlanie informacji o stanie urządzenia, w szczególności sukcesy odczytania danych z legitymacji

- Zapamiętanie czasu kiedy zebrane zostały obecności

Dodatkowo, chciałbym żeby cała elektronika schowana była wewnątrz plastikowego opakowania, a zasilanie dostarczane zapomocą kabla podłączonego do sieci elektrycznej. Opakowanie powinno zostać wyposażone w otwory na kabel zasilający, zewnętrzny nośnik danych, wyświetlacz. Czytnik kart RFID powinien być tak zlokalizowany aby umożliwić łatwe zebranie danych z ELS.

### 3.1.1. Zastosowana platforma i peryferia

Arduino to cała platforma programistyczna umożliwiająca łatwe projektowanie i testowanie układów składających się z wielu elektronicznych modułów. Podstawą każdego projektu Arduino jest układ elektroniczny oparty na mikroprocesorze udostępniający, w zależności od wersji, szereg peryferiów. Do wyboru użytkownika pozostaje cała gama wersji płytek drukowanych spod znaku Arduino. Dodatkowo platforma udostępniona jest na licencji open hardware, co sprawiło, że z projektem związana jest olbrzymia międzynarodowa społeczność entuzjastów elektroniki. Pod egidą platformy udostępnione jest również środowisko programistyczne umożliwiające wgrywanie na podłączoną do komputera płytkę własnych programów.

Ciekawą możliwością dostępną dla użytkowników platformy jest możliwość skonstruowania własnej płytki z procesorem na bazie powszechnie dostępnych schematów gotowych układów. Może to być rozwiązanie bardzo atrakcyjne w przypadku kiedy projektantowi zależy na ograniczeniu kosztów układu. W moim rozwiązaniu postanowiłem nie iść tak daleko i skorzystać z jednego z gotowych układów. Podstawą mojego weryfikatora jest postanowiłem uczynić płytkę Arduino Uno opartą na mikroprocesorze AVR ATmega328. Kontroler w tym układzie jest bardzo prosty, posiada bardzo mocno ograniczone zasoby, co wymusza dodatkowy wysiłek włożony w zarządzanie pamięcią i czasem procesora. Do mojej dyspozycji miałem następujące zasoby

- 32 kB Flash - przestrzeń na kod programu
- 2 kb SRAM zajmowane przez pamięć operacyjną.
- 2 porty SPI, port I2C
- Jednostka obliczeniowa o częstotliwości zegara 20MHz

Dodatkowo, w celu zapewnienia wszystkich wymaganych funkcjonalności przez projektowany przeze mnie weryfikator, skorzystałem z następujących dodatkowych modułów:

- Czytnik RFID - do odczytywania danych z legitymacji studenckiej wyposażonej w interfejs zbliżeniowy, korzystający z drugiego interfejsu SPI

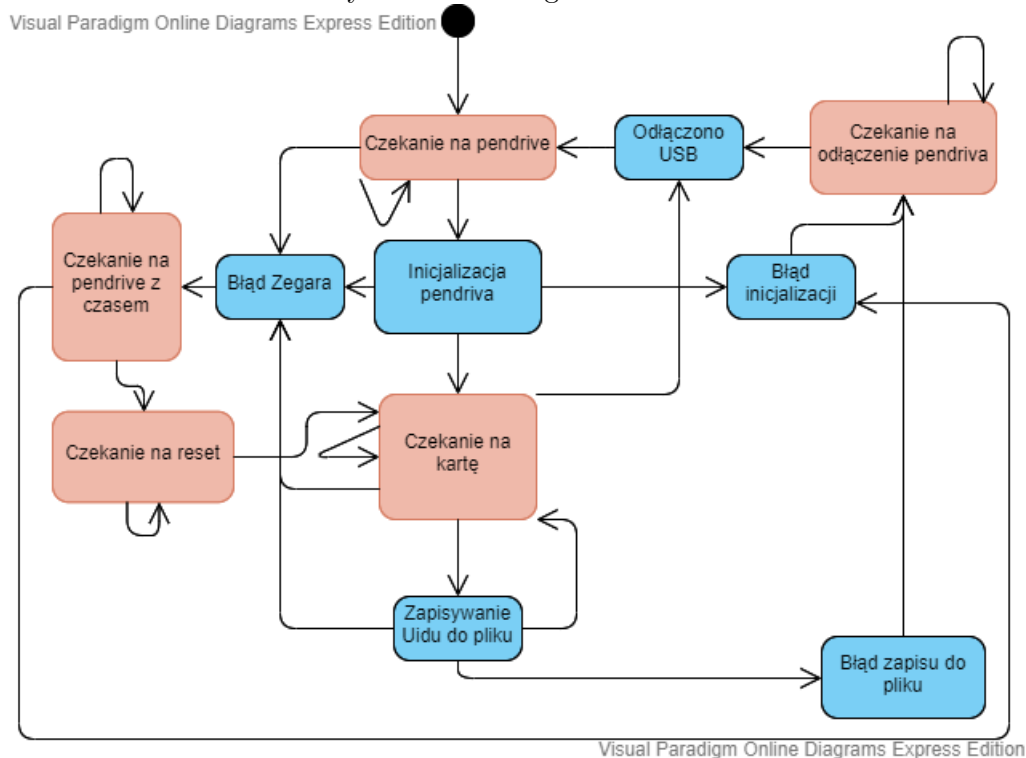
- USB Host Shield - dodające port USB, korzystający z interfejsu SPI, umożliwiający podłączenie pendriva
- Wyświetlacz LCD wraz z potencjometrem do ustawiania kontrastu, przekazujący urztkownikowi informacje na temat stanu urządzenia
- RTC - zegar czasu rzeczywistego z baterią do zapamiętywania czasu zbierania danych, korzystający z interfejsu I2C
- Buzzer piezo wydający dźwiękowe sygnały w razie sukcesu

### 3.1.2. Konstrukcja

### 3.1.3. Funkcjonowanie

Podczas analizy działania weryfikatora wyodrębniłem 12 stanów w których może znaleźć się urządzenie

Rysunek 3.1: Diagram stanów



**3.1.4. Napotkane problemy****3.1.5. Testowanie****3.1.6. Inne koncepcje****3.2. Menadżer obecności - umożliwiający katalogowanie danych****3.3. Baza studentów - umożliwiająca identyfikację danych zebranych z ELS**

Zaproponowane przeze mnie rozwiązanie składa się z dwóch części:

1. Urządzenia opartego o platformę Arduino rozszerzoną o dodatkowe moduły niezbędne do odczytania potrzebnych informacji
  - Moduł RFID - odczytujący unikalny identyfikator z legitymacji studenckiej
  - USB Host Shield - pozwalający na podłączenie zewnętrznej pamięci do Arduino i zapisanie zebranych informacji
  - Wyświetlacz LCD + Buzzer - pozwalający na komunikację z użytkownikiem
  - Zegar czasu rzeczywistego
2. Aplikacji internetowej pozwalającej na tworzenie i zarządzanie wykładami. Portal umożliwia utworzenia własnego użytkownika, do którego można przypisywać zajęcia, w ramach których można rejestrować obecności studentów zebrane przy pomocy opisanego wyżej urządzenia.



## Rozdział 4.

# Realizacja



## Rozdział 5.

# Możliwości rozwoju

- dostęp do usos i propagowanie zebranych danych na stronach uczelnianych
- Udostępnienie systemu dla studentów i umożliwienie monitorowania swoich obecności (logowanie tylko przez uczelnianego maila, albo konta utworzone wyłącznie przez administratora)
- powiadomienia mailowe do studenta o zbliżającym się wyczerpaniu limitu nieobecności



## Rozdział 6.

## Problemy



## Rozdział 7.

## Wnioski

...