

Politechnika Poznańska
Wydział Elektryczny
Instytut Automatyki i Inżynierii Informatycznej

Jakub Piotr
Wojciech Agaciński
Krzysztof Adamczak

Attendance Manager

Poznań, 2017

Spis treści

1	Wstęp	4
1.1	Cel i zakres pracy	4
1.2	Podział pracy	4
2	Opis dziedziny przedmiotowej pracy	5
2.1	Pojęcia i definicje	5
2.2	Stan wiedzy	5
2.3	Dyskusja	5
3	Wybór technologii informatycznych	6
3.1	ASP.NET Core Web API	6
3.2	MS SQL Server 2016	6
3.3	Microsoft Azure	6
3.4	Front End	6
3.5	Microsoft .NET C framework	7
3.6	Latex	7
3.7	Format JSON	7
3.8	Środowisko programistyczne Visual Studio	7
3.9	Środowisko programistyczne JetBrains WebStorm	7
4	Architektura systemu Attendance Manager	8
4.1	Schemat bazy danych	8
4.2	Moduł importu i eksportu danych	9
4.3	Wnioski	9
5	Implementacja	10
5.1	Wzorce projektowe i strukturalne	10
5.2	Model struktur danych	10
5.3	Model klas	10
6	Bezpieczeństwo systemu Attendance Manager	12
6.1	Anonimizacja danych	12
6.2	Techniki kryptograficzne	12
7	Instrukcja użytkowania aplikacji Attendance Manager	13
7.1	Strona główna	13
7.2	Lista wydarzeń - Active & Incoming	14
7.3	Lista wydarzeń - Expired	14
7.4	Strona ze szczegółami wydarzenia	15
7.5	Tworzenie nowego wydarzenia	15
8	Podsumowanie	16
	Spis rysunków	17

Spis tabel	17
9 Dodatki	18
9.1 Instalacja systemu Attendance Manager	18
9.2 Instrukcja użytkownika systemu Attendance Manager	18
9.3 Instrukcja administratora systemu Attendance Manager	18
9.4 Treść ankiety systemu Attendance Manager	18
10 Załączniki	19

1 Wstęp

1.1 Cel i zakres pracy

System ma za zadanie monitorować obecności studentów na zajęciach laboratoryjnych, wykorzystując istniejącą infrastrukturę informatyczną, tj. czytniki kart inteligentnych zainstalowane w komputerach oraz legitymacje studenckie.

<i>Lp.</i>	<i>Opis funkcjonalności</i>	<i>Dostępność</i>	<i>Priorytet</i>
1.	Odczytanie danych z legitymacji	-	Wysoki
2.	Ręczne wprowadzenie danych o obecności	a	Średni
3.	Dostęp do panelu administracyjnego przez web serwis	a	Wysoki
4.	Podgląd istniejących w systemie wydarzeń	a	Wysoki
5.	Planowanie wydarzeń jednorazowych	a	Wysoki
6.	Planowanie wydarzeń cyklicznych	a	Wysoki
7.	Ustalanie list dopuszczonych do wydarzenia uczestników	a	Średni

Tabela 1: Funkcjonalność systemu Attendance Manager(a – administrator)

1.2 Podział pracy

<i>Osoba odpowiedzialna</i>	<i>Zadanie</i>
Krzysztof Adamczak	Front end, Web serwis
Wojciech Agaciński	Web serwis, Baza danych
Jakub Piotr	Aplikacja kliencka

Tabela 2: Podział pracy

2 Opis dziedziny przedmiotowej pracy

2.1 Pojęcia i definicje

- Karta elektroniczna - uniwersalny nośnik danych w postaci karty wykonanej z plastiku z umieszczonym na niej (lub wewnątrz niej) jednym lub kilkoma układami scalonymi (chip), które pozwalają na ochronę procesu logowania użytkownika, kontrolę dostępu i zawartych na niej danych. Może być odczytywana za pomocą urządzeń automatycznych, np. przy zawieraniu i rozliczaniu transakcji finansowych oraz w kasach cyfrowych. Karty elektroniczne mają rozmiar zgodny z formatem ID-1 (85,60 53,98 mm) określony normą ISO/IEC 7810 tak jak tradycyjne karty kredytowe z paskiem magnetycznym. Często posiadają również taki pasek i mogą być odczytywane w urządzeniach nie obsługujących kart elektronicznych.
- Elektroniczna Legitymacja Studencka (ELS) - rodzaj legitymacji, nowa postać tradycyjnej papierowej legitymacji studenckiej. Została wprowadzona przepisami Rozporządzenia Ministra Nauki i Szkolnictwa Wyższego z dnia 2 listopada 2006 w sprawie dokumentacji przebiegu studiów. Elektroniczna Legitymacja Studencka ma formę karty z tworzywa sztucznego z wbudowanym procesorem, a jej wzór graficzny jest jednolity dla wszystkich uczelni i został określony w ww. rozporządzeniu. Legitymacja wyposażona jest w chip stykowy oraz może być wyposażona w bezstykowy interfejs Mifare. Według rozporządzenia jest ona dokumentem poświadczającym status studenta. Dodatkowo może być ona wykorzystywana jako karta biblioteczna, karta dostępu do laboratoriów lub bilet komunikacji miejskiej.

2.2 Stan wiedzy

Powszechnie dostępne czytniki kart elektronicznych, wśród których można wymienić te wbudowane w komputery laboratoryjne na terenie Politechniki Poznańskiej oraz moduły NFC w nowoczesnych smartfonach - pozwalają uzyskać dostęp do danych zapisanych na Elektronicznych Legitymacjach Studenckich. Te dane pozwalają nie tylko zidentyfikować posiadacza karty, ale także określić jego uprawnienia jak i przynależność wewnątrz jednostek uczelnianych.

2.3 Dyskusja

Wykorzystanie Elektronicznych Legitymacji Studenckich do rejestracji obecności może budzić wiele obaw odnosnie bezpieczeństwa informacji. Legitymacje zawierają dane osobowe uważane za poufne, m. in PESEL, dlatego też wrażliwe dane odczytane przez system powinny być odpowiednio zabezpieczone przed nieuprawnionym dostępem, a najlepiej niedostępne dla administratorów oprócz przypadków w których te dane są niezbędne.

3 Wybór technologii informatycznych

3.1 ASP.NET Core Web API

Do stworzenia interfejsu webowego odpowiadającego za pośrednictwo w komunikacji pomiędzy aplikacją kliencką, bazą danych oraz panelem administracyjnym w przeglądarce Internetowej zostało wykorzystane środowisko ASP.NET Core Web API. Jest to nowoczesne narzędzie o otwartym kodzie źródłowym stworzone przez firmę Microsoft. Umożliwia ono tworzenie wydajnej części serwerowej systemów na każdej platformie.

3.2 MS SQL Server 2016

Do przechowywania danych został wykorzystany serwer bazodanowy Microsoft SQL Server 2016 wraz z relacyjną bazą danych. Jest to stabilne i sprawdzone rozwiązanie, które doskonale poradzi sobie z zadaniem przechowywania dużej ilości danych o stałej strukturze, co wpisuje się w specyfikację systemu.

3.3 Microsoft Azure

Całość rozwiązania po stronie serwera została umieszczona w chmurze hybrydowej Microsoft Azure. Umożliwia to wygodne i efektywne zarządzanie i monitorowanie pracą systemu bez potrzeby tworzenia infrastruktury, np. maszyn wirtualnych na fizycznych komputerach.

3.4 Front End

Aplikacja kliencka napisana została w języku TypeScript z wykorzystaniem framework'a Angular. Jest to niezwykle rozbudowany framework stworzony przede wszystkim z myślą o obsłudze dużej ilości danych. Dzięki wielu wbudowanym funkcjom umożliwia on tworzenie aplikacji bez konieczności instalowania wielu dodatkowych bibliotek.

Jest to aplikacja typu Single Page Application czyli aplikacja w której cały proces ładowania strony został przeniesiony na stronę użytkownika. Umożliwiło to odciążenie serwera wystawiającego stronę internetową oraz przyspieszenie działania aplikacji. Dla użytkownika sprawia także wrażenie bardziej płynnego działania, bez konieczności długiego czasu oczekiwania na przejście do innej części strony.

Interfejs użytkownika stworzony został w style Google Material Design dzięki wykorzystaniu biblioteki Angular Material. Do utworzenia układu strony wykorzystana została biblioteka angular/flex-layout. Dodatkowo zostały wykorzystane niektóre elementy z biblioteki Bootstrap 4.0.

Środowiskiem uruchomieniowym aplikacji klienckiej jest Node.js czyli środowisko uruchomieniowe zaprojektowane do tworzenia wysoce skalowalnych aplikacji internetowych. Do zarządzania paczkami oraz bibliotekami wykorzystywanymi w aplikacji używany jest Node Package Manager (NPM).

3.5 Microsoft .NET C framework

Aplikacja kliencka obsługująca rejestrowanie użytkownika poprzez czytnik ELS, została napisana przy użyciu standardowych bibliotek C.

3.6 Latex

Dokumentacja została sporządzona przy pomocy oprogramowania LaTeX.

3.7 Format JSON

(ang. *JavaScript Object Notation*) został wykorzystany do przesłania odczytanych danych z ELS do Web API.

3.8 Środowisko programistyczne Visual Studio

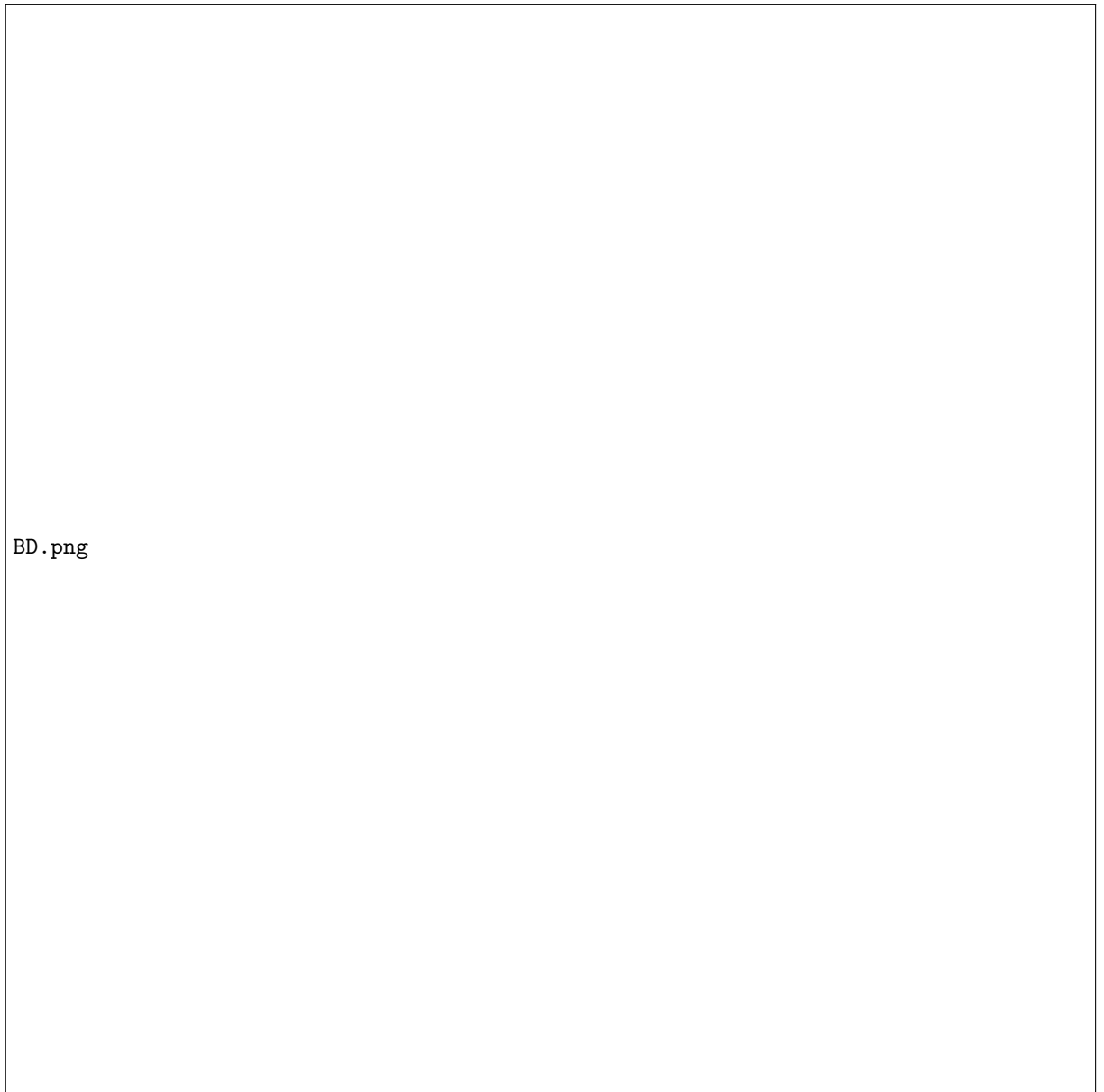
Do prac nad elementami systemu stworzonymi w technologiach Microsoft .NET wykorzystane zostało środowisko programistyczne Microsoft Visual Studio. Jest to kompleksowe rozwiązanie umożliwiające programistom tworzenie i zarządzanie dużymi rozwiązaniami.

3.9 Środowisko programistyczne JetBrains WebStorm

Do utworzenia aplikacji klienckiej wykorzystano program WebStorm od JetBrains. Jest to wysoce zaawansowane środowisko programistyczne stworzone przede wszystkim do tworzenia aplikacji w języku JavaScript. Posiada on wiele wbudowanych narzędzi takich jak NPM czy Gulp dzięki czemu znacznie ułatwia ono proces tworzenia aplikacji.

4 Architektura systemu Attendance Manager

4.1 Schemat bazy danych



BD.png

Rys. 1: Schemat bazy danych

4.2 Moduł importu i eksportu danych

Inferfejs administracyjny systemu podczas działania korzysta z interfejsu webowego stworzonego zgodnie z zasadami REST. Interfejs ten wykorzystuje notację JSON do serializacji zwracanych danych oraz protokół HTTP do komunikacji.

4.3 Wnioski

- Wykorzystanie wzorca REST API oraz formatu danych JSON pozwala integrację systemu z innymi modułami stworzonymi na dowolnej platformie z wykorzystaniem dowolnego języka programowania wspierającego w.w. technologie.

5 Implementacja

5.1 Wzorce projektowe i strukturalne

- Model View Controller - wzorec strukturalny wspierający zasadę Separation of Concerns - podział części serwerowej systemu na części odpowiedzialne za widoki, logikę biznesową oraz obsługę zapytań HTTP
- Inversion of Control - wzorec strukturalny odpowiedzialny za zapewnienie luźnego powiązania pomiędzy klasami i likwidowanie zależności poprzez wykorzystanie Wstrzykiwania Zależności.
- Unit of Work - wzorec projektowy zapewniający operowanie na jednej jednostce transakcji bazodanowych w obrębie zapytania do serwisu
- Repository Pattern - wzorec projektowy zapewniający spójny format operacji CRUD na różnych strukturach danych

5.2 Model struktur danych

Struktury danych odzwierciedlone w strukturach bazodanowych składają się z klas według modelu Plain Old CLR Object

5.3 Model klas

Podstawowe klasy występujące w systemie

- Attendee - uczestnik wydarzenia
- Course - przedmiot dydaktyczny
- CourseType - rodzaj zajęć
- CourseUnit - jednostka dydaktyczna obejmująca przedmiot oraz rodzaj zajęć
- Event - wydarzenie
- Lecturer - wykładowca, prowadzący zajęcia
- Room - pomieszczenie
- TimeSlot - przedział czasowy

Podstawowe klasy występujące w aplikacji klienckiej.

- ElectronicStudentCardData - model do przechowywania danych z ELS
- User - klasa która na podstawie klasy ElectronicStudentCardData tworzy użytkownika.

- CElectronicStudentCardContactDataReader - klasa zajmuje się odczytaniem danych z ELS
- ACR122UReader - klasa obsługująca czytnik ELS
- Event - wydarzenie

6 Bezpieczeństwo systemu Attendance Manager

6.1 Anonimizacja danych

W przypadku tworzenia raportów oraz danych statystycznych na podstawie informacji zebranych przez system dane powinny być anonimizowane, aby osoby postronne nie miały dostępu do wrażliwych danych

6.2 Techniki kryptograficzne

Techniki kryptograficzne mogą zostać wykorzystane do zabezpieczenia wrażliwych danych w systemie, m. in. haseł użytkownika czy numerów PESEL uczestników wydarzeń.

7 Instrukcja użytkowania aplikacji Attendance Manager

7.1 Strona główna

Rys. 2: Ekran główny aplikacji klienckiej

Strona główna aplikacji zawiera podstawowe informacje dotyczące projektu takie jak: tytuł, opis, technologie oraz skład zespołu. W menu aplikacji po lewej stronie znajduje się przycisk otwierający menu główne aplikacji. W menu aplikacji mamy do wyboru następujące ekrany:

- Home - prowadzi do strony startowej aplikacji (strony obecnej na zdjęciu)
- Events - strona z listą wydarzeń
- Add event - strona służąca do utworzenia nowego wydarzenia

7.2 Lista wydarzeń - Active & Incoming

Rys. 3: Ekran listy wydarzeń

Jest to strona zawierająca listę wydarzeń. Została ona podzielona na dwie zakładki:

- Active Incoming - zakładka zawiera tabelę z wydarzeniami nadchodzącymi oraz aktywnymi
- Expired - zakładka zawiera tabelę z wydarzeniami które już się wydarzyły i zakończyły

Na powyższym zdjęciu przedstawiona została zakładka 'Active Incoming'. Zawiera ona tabelę wydarzeń nadchodzących oraz aktywnych. W tabeli znajdują się pola:

- Name - kolumna zawierająca nazwę wydarzenia
- Course name - jest to nazwa kursu do którego przypisane zostało dane wydarzenie
- Status - jest to obecny status wydarzenia ('Active' lub 'Incoming')
- Cyclical - pola informujące czy wydarzenie jest cykliczne
- Restricted - pole informujące czy wydarzenie ma ograniczoną listę uczestników mogących wziąć udział w wydarzeniu
- Next date - kolumna z datą wydarzenia
- Time - kolumna z przedziałem czasowym w którym wystąpi wydarzenie

W ostatniej kolumnie znajdują się przyciski akcji.

- Show details - po wciśnięciu tego przycisku nastąpi przeniesienie na stronę zawierającą wszystkie informacje na temat wybranego wydarzenia
- Remove - przycisk służący usunięciu wybranego wydarzenia

7.3 Lista wydarzeń - Expired

Rys. 4: Ekran listy nieaktywnych wydarzeń

Na powyższym zdjęciu przedstawiona została zakładka z wydarzeniami już zakończonymi. Zawiera ona tabelę z następującymi kolumnami:

- Name - kolumna zawierająca nazwę wydarzenia

- Course name - jest to nazwa kursu do którego przypisane zostało dane wydarzenie
- Expired date and time - kolumna ta zawiera datę oraz przedział czasowy w którym odbyło się wydarzenie
- Restricted - pole informujące czy wydarzenie ma ograniczoną listę uczestników mogących wziąć udział w wydarzeniu

W tabeli znajdują się także przycisk 'Show details' który umożliwia podgląd szczegółów danego wydarzenia.

7.4 Strona ze szczegółami wydarzenia

Rys. 5: Ekran z informacjami o nadchodzącym wydarzeniu

Strona ta zawiera szczegółowe informacje na temat wydarzenia. Informacje zostały podzielone na sekcje:

- Basic informations - sekcja zawierająca podstawowe informacje o wydarzeniu
- Course informations - sekcja zawierająca informację o kursie do którego podpięte jest wydarzenie. Sekcja ta jest opcjonalna. W przypadku braku podpiętego kursu sekcja ta nie zostanie wyświetlona
- Lecturer informations - sekcja zawierająca informację o przypisanym do wydarzenia wykładowcy
- List of eligible participants - w przypadku gdy wydarzenie jest zamknięte (posiada listę uczestników uprawnionych do wzięcia udziału w wydarzeniu) pojawia się sekcja z listą uprawnionych uczestników
- Attendance list - sekcja zawierająca listę obecności. Pojawia się w przypadku gdy event jest aktywny lub zakończony

Na tej stronie możliwa jest edycja wydarzenia. W zależności od statusu wydarzenia niektóre pola/sekcje mogą być niedostępne do edytowania.

- Wydarzenie nadchodzące - możliwa jest edycja prawie wszystkich pól oprócz listy obecności
- Wydarzenie aktywne - możliwa jest edycja jedynie listy obecności
- Wydarzenie zakończone - możliwa jest edycja jedynie listy obecności

7.5 Tworzenie nowego wydarzenia

Jest to strona służąca dodawaniu nowych wydarzeń. Zawiera ona formularz z odpowiednimi polami. Pola wymagane zostały odpowiednio oznaczone. Możliwość dodawania listy uczestników odblokowuje się po wybraniu opcji 'Restricted'.

Rys. 6: Ekran dodawania nowego wydarzenia

8 Podsumowanie

Podczas realizacji projektu udało się zrealizować podstawowe funkcjonalności systemu, które zostały podane w specyfikacji. Dzięki stworzeniu szkieletu architektury oraz aplikacji klienckich oraz serwerowej możliwa jest dalsza rozbudowa systemu poprzez stopniowe dodawanie funkcjonalności, które system mógłby posiadać.

W trakcie trwania rozwoju systemu, zespół spotkał się z licznymi problemami natury technicznej oraz organizacyjnej, jednak nie napotkał żadne przeszkody w realizacji projektu. Wdrożenie systemu na środowisko produkcyjne wiązało by się jednak z dodatkowym nakładem pracy, związanym między innymi z dokładną analizą przepisów prawnych regulujących kwestie danych osobowych, które zbiera i przetwarza system.

Spis rysunków

1	Schemat bazy danych	8
2	Ekran główny aplikacji klienckiej	13
3	Ekran listy wydarzeń	14
4	Ekran listy nieaktywnych wydarzeń	14
5	Ekran z informacjami o nadchodzącym wydarzeniu	15
6	Ekran dodawania nowego wydarzenia	16

Spis tablic

1	Funkcjonalność systemu Attendance Manager(a – administrator)	4
2	Podział pracy	4

9 Dodatki

9.1 Instalacja systemu Attendance Manager

9.2 Instrukcja użytkownika systemu Attendance Manager

9.3 Instrukcja administratora systemu Attendance Manager

9.4 Treść ankiety systemu Attendance Manager

10 Załączniki

- treść pracy w formacie LATEX,
- implementację systemu Attendance Manager,
- kody uruchomieniowe systemu Attendance Manager.