|  |
| --- |
|  |
| **"SOWLZWL" - Sprawdzanie obecności w laboratorium z wykorzystaniem legitymacji studenckiej** |
| **Dokumentacja techniczna** |
|  |

|  |
| --- |
| Eliza Olczak, Piotr Siwiński, Artur Jędryczkowski - Poznań 2017 |

Spis treści:

|  |  |  |
| --- | --- | --- |
| 1. | Wprowadzenie.......................................................................................................... | 3 |
| 1.1 | Opis programu.......................................................................................................... | 3 |
| 1.2 | Zakres........................................................................................................................ | 3 |
| 1.3 | Pojęcia....................................................................................................................... | 3 |
| 2 | Podział prac pomiędzy członków zespołu................................................................ | 4 |
| 3 | Funkcjonalności oferowane przez aplikację............................................................. | 4 |
| 4 | Wymaganie niefunkcjonalne................................................................................... | 5 |
| 4.1 | Wymagania produktowe.......................................................................................... | 5 |
| 4.2 | Wymagania organizacyjne....................................................................................... | 5 |
| 4.3 | Wymagania zabezpieczeń........................................................................................ | 5 |
| 4.4 | Wymagania zewnętrzne.......................................................................................... | 6 |
| 4.5 | Charakterystyka użytkowników.............................................................................. | 6 |
| 5 | Wybrane technologie................................................................................................ | 6 |
| 6 | Instrukcja użytkowania aplikacji.............................................................................. | 7 |
| 6.1 | Instrukcja użytkowania aplikacji webowej............................................................... | 7 |
| 6.2 | Instrukcja użytkowania interfejsu czytnika kart....................................................... | 14 |
| 7 | Model aplikacji......................................................................................................... | 17 |
| 7.1 | Model relacyjny bazy danych................................................................................... | 17 |
| 7.2 | Lista aktorów............................................................................................................ | 18 |
| 7.3 | Diagram przypadków użycia.................................................................................... | 19 |
| 7.4 | Diagramy aktywności............................................................................................... | 20 |
| 8 | Napotkane trudności……………………………………………………………………… | 26 |
|  |  |  |

1.Wprowadzenie

Niniejsza dokumentacja zawiera wymagania związane ze stworzeniem systemu SOWLZWL.

Wybierając temat kierowaliśmy się przede wszystkim jego użytecznością i praktycznością. Uważamy, że tego typu program jest potrzebny oraz, że implementacja tego typu problemu poszerzy nasze umiejętności techniczne. Stanowił on dla nas duże wyzwanie jednakże leżał on w zakresie naszych umiejętności.

1.1 Opis programu

Program ma na celu scentralizowanie zarządzania obecnością w laboratorium. W tym celu wykorzystaliśmy czytniki kart w komputerach laboratoryjnych. Zasada działania programu jest prosta, student przykłada swoją legitymacje studencką do czytnika, a program pobiera z niego dane przypisując je do odpowiednich tabel w bazie danych. Prowadzący zajęcia ma możliwość ich podglądu oraz edycji. Program gwarantuje możliwość importowania listy studentów, przydział danych do odpowiednich tabel ze względu na godziny, przedmiot, laboratoria, także eksport danych oraz ręczne modyfikowanie obecności.

1.2 Zakres

Dokument ten ma za zadanie sprecyzować, jak powinien wyglądać i jakie funkcjonalności powinien oferować cały system. Zawarte są w nim również informacje na temat komunikacji pomiędzy poszczególnymi modułami, a także podział prac poszczególnych członków zespołu.

1.3 Pojęcia

Słowniczek pojęć został stworzony w celu sprecyzowania używanych dalej terminów. Jest to zbiór definicji nazw pojawiających sie w następnych rozdziałach.

* gość - osoba korzystająca z systemu nie będąc zalogowana,
* użytkownik - osoba posiadająca konto w systemie, prowadzący zajęcia lub osoba zarządzająca obecnością na zajęciach,
* administrator - użytkownik panelu administratora, posiadający dodatkowe narzędzia do zarzadzania systemem, możliwość edytowania, osoba odpowiadająca za konserwację systemu
* oprogramowanie, program, aplikacja– całość informacji w postaci zestawu instrukcji, zaimplementowanych interfejsów i zintegrowanych danych przeznaczonych dla komputera do realizacji wyznaczonych celów. Celem naszego oprogramowania jest efektywne i scentralizowane sprawdzanie obecności w laboratorium za pomocą legitymacji studenckich.

2. Podział prac pomiędzy członków zespołu

Mając na uwadze kompetencje, doświadczenia, a także umiejętności każdego z członków naszego zespołu, na wstępie wyznaczyliśmy każdemu konkretne zadania do realizacji.

* Zapoznanie się z dokumentacją (Wszyscy),
* Utworzenie projektu konsolowego (Piotr),
* Projektu Webowego (Eliza),
* Dodanie widoków studenta, przedmiotu(Artur),
* Odczyt danych z legitymacji studenckiej (Piotr),
* Zapis w bazie danych (Artur, Eliza),
* Wyświetlenie danych w aplikacji webowej (Eliza),
* Połączenie modułu odczytu danych z aplikacją webową (Wszyscy),
* Poprawki, ukończenie aplikacji (Wszyscy)

3.Funkcjonalności oferowane przez aplikację

Funkcjonalność jest zbiorem atrybutów oprogramowania określających zdolność do dostarczenia funkcji zaspokajających wyznaczone i zakładane potrzeby, podczas używania w określonych warunkach.

* scentralizowane zarządzanie obecnością na laboratorium.

*Jest to jedna z najważniejszych funkcjonalności, mająca ułatwić zarządzanie prowadzącym przedmiot, usprawnić tę procedurę i ją uwiarygodnić. Prowadzący ma wgląd do danych zawierających informacje, kto jest obecny na jego zajęciach.*

* wykorzystanie czytników kart w komputerach laboratoryjnych

*Aby w sprawdzić obecność student przykłada legitymacje do czytnika znajdującego sie przy jego komputerze.*

* importowanie/wczytywanie listy studentów  
  *Prowadzący ma wgląd kto jest obecny na jego zajęciach(dane zawierają imię, nazwisko oraz numer indeksu) oraz kto powinien się na nich znajdować.*
* podział ze względu na godziny, przedmiot, laboratoria

*Dane są przyporządkowywane do odpowiednich tabel posortowanych według nazwy przedmiotu oraz godzin odbywania się zajęć.*

* eksport danych

*Dane można eksportować, gdyby prowadzący chciałby je sobie zapisać.*

* ręczne modyfikowanie obecności

*Prowadzący ma możliwość ręcznego modyfikowania tych danych.*

4. Wymagania niefunkcjonalne

Opisujemy tutaj czynniki wpływające na implementacje systemu i wymagania z nią związane. Znajdują się tu również pewne ograniczenia nałożone na projekt.

4.1 Wymagania produktowe

Ta sekcja przedstawia wymagania ograniczające działanie produktu. Są one niezbędne do poprawnego korzystania z funkcji systemu.

* użytkownik w celu korzystania z pełnej funkcjonalności aplikacji powinien posiadać własny czytnik SMART CART współpracujący z kartami mikroprocesorowymi standardów: ISO 7816 Classe A, B e C (5 V, 3 V, 1.8 V), PC/SC wersja 1.0/2.0 standard e EMV 4.1, wspomagający API: PC/SC, CT-API i współpracujący z kartami pamięci: syncrone 2-line, 3-line, SLE4418, 4428, 4432, 4442, AT88SC1608, 45D041 i interfejsem I2C
* aplikacja webowa i interfejs czytnika kart powinien być dostępny w języku polskim,
* czas uruchamiania aplikacji webowej powinien wynosić maksymalnie 10s, nie dotyczy to pierwszego uruchomienia aplikacji na danym komputerze
* przejścia pomiędzy widokami w aplikacji powinny być płynne,
* użytkownik w celu korzystania z pełnej funkcjonalnosci aplikacji powinien posiadac własny adres e-mail.

4.2 Wymagania organizacyjne

Wymagania organizacyjne są zbiorem ograniczeń dla programisty implementującego system. Pokazują one jak powinien przebiegać proces tworzenia produktu.

* do implementacji aplikacji webowej wykorzystany zostanie język C#,
* do implementacji czytnika kart wykorzystany zostanie język JAVA,
* do utworzenia bazy danych będzie wykorzystany zostanie język MSSQL,
* środowiskiem wykonawczym dla aplikacji webowej będzie przeglądarka internetowa
* programowanie odbywać się będzie w środowiskach MS Visual Studio 2015 i Microsoft SQL Server 2014 Management Studio,
* jako repozytorium zostanie użyty Git w serwisie github.com
* system należy wdrożyć do końca czerwca.

4.3 Wymagania zabezpieczeń

W celu zapewnienia bezpieczeństwa naszemu systemowi w fazie projektowej zebrano i spisano konieczne do zaimplementowania elementy.

* hasło będzie musiało mieć minimum 8 znaków, posiadać minimum jedna małą literę, minimum jedną wielką literę oraz minimum jeden znak specjalny lub cyfrę,
* baza danych musi zostać zabezpieczona przed SQL injection.

4.4 Wymagania zewnętrzne

Wymagania zewnętrzne określają ogólne ramy projektu, których nie można pominąć podczas tworzenia systemu.

* korzystanie z aplikacji powinno być łatwe i intuicyjne nie wymagające dodatkowego przeszkolenia,
* system powinien posiadać zabezpieczenia przed niepowołanym dostępem do funkcji administrowania,
* przechowywanie i udostępnianie danych osobowych powinno odbywać się zgodnie z zapisami ustawy z dnia 29 sierpnia 1997 r. o ochronie danych osobowych.

4.5 Charakterystyka użytkowników

System przeznaczony jest dla osób prowadzących zajęcia na uczelniach wyższych i chcących sprawdzać obecność studentów za pomocą legitymacji, lub pojedynczych osób chcących prowadzących aktywności ze studentami i pragnących weryfikować ich obecność za pomocą czytnika legitymacji studenckich.

5. Wybrane technologie

* C#

*Język obiektowy z hierarchią o jednym elemencie nadrzędnym, zarządzający pamięcią*

*w środowisko uruchomieniowym. Umożliwia w czasie pracy analizę struktury kodu z poziomu tego kodu. Zawiera bogate biblioteki klas BCL, umożliwiające rozwijanie dynamicznych aplikacji internetowych (ASP.NET) .*

* ASP .NET MVC 5

*Platforma aplikacyjna do budowy aplikacji internetowych opartych na wzorcu Model-View-Controller (MVC) oparty na technologii ASP.NET. Oferuje wykorzystanie silnika ASP.NET pozwalającego na wykorzystanie wielu komponentów takich jak mechanizmy zarządzania stanem aplikacji, mechanizmy uwierzytelniania i autoryzacji, cache itd.*

*Posiada model programistyczny (API) oparty na interfejsach, pozwalający na łatwą rozbudowę, dorabianie i testowanie poszczególnych komponentów. Posiada także elastyczny mechanizm mapowania adresów.*

* MS SQL

*System zarządzania bazą danych, wspierany i rozpowszechniany przez korporację Microsoft. MS SQL Server jest platformą bazodanową typu klient-serwer. Odznacza się wydajnością, niezawodnością i skalowalnością. Przede wszystkim są tu zaimplementowane wszelkie mechanizmy wpływające na bezpieczeństwo operacji (m.in. procedury wyzwalane).*

* JAVA FX

*Java FX jest domyślną biblioteką definiowania graficznego interfejsu użytkownika w języku Java począwszy od wersji 8. Zastąpiła ona starsze biblioteki takie jak Swing i AWT. Charakteryzuje się przede wszystkim dobrą separacją warstw, wsparciem dla architektury MVC (Model View Controller) i udostępnia do tego użyteczne narzędzie o nazwie Scene Builder.*

* Java™ Smart Card I/O API

*API Javy obsługujące czytniki kart.*

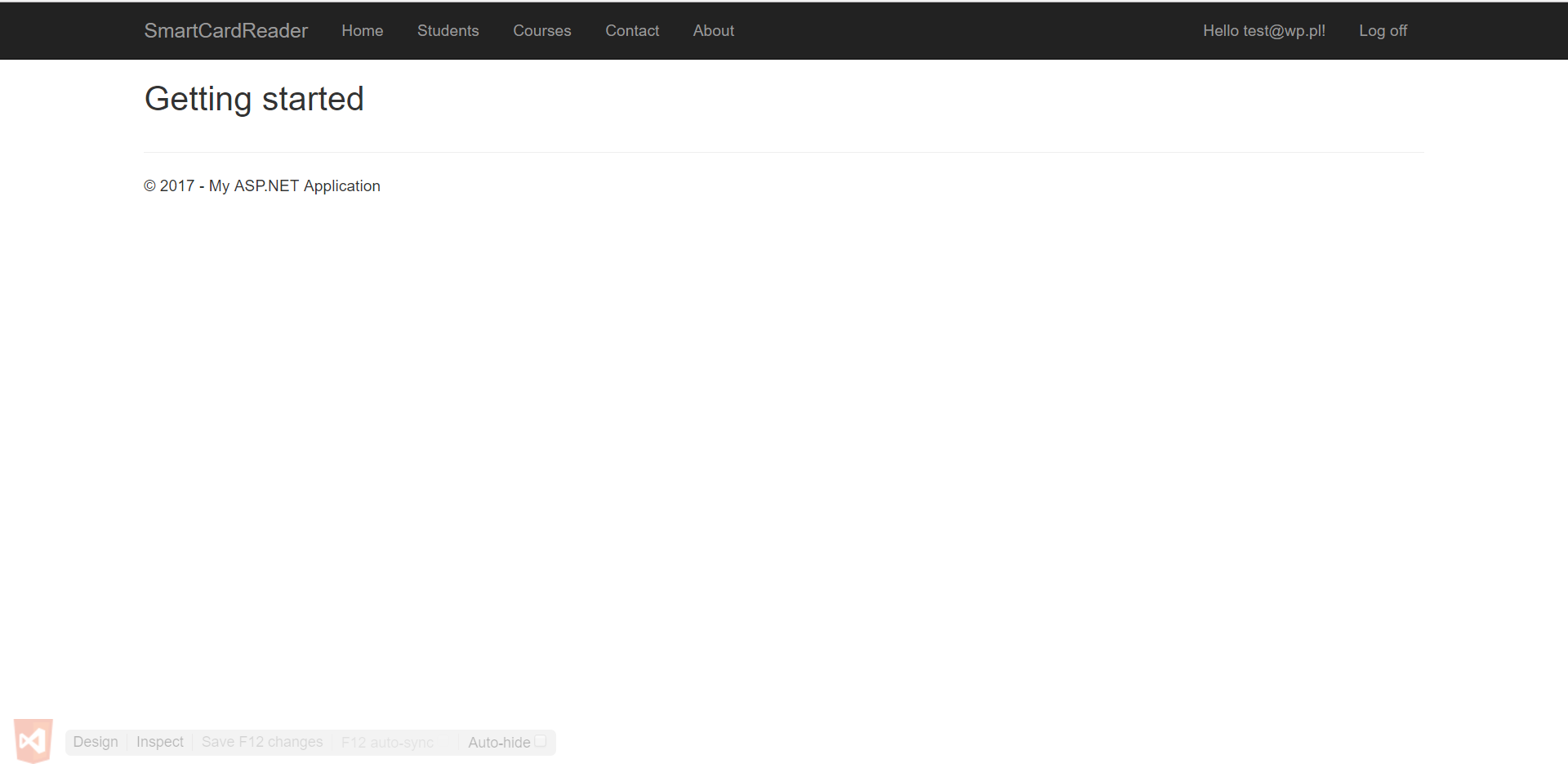
6. Instrukcja użytkowania aplikacji

Niniejsza instrukcja opisuje poszczególne funkcje naszego oprogramowania - poświęć chwilę na zapoznanie się z nią tak, abyś nie miał żadnych wątpliwości dotyczących obsługi naszego produktu.

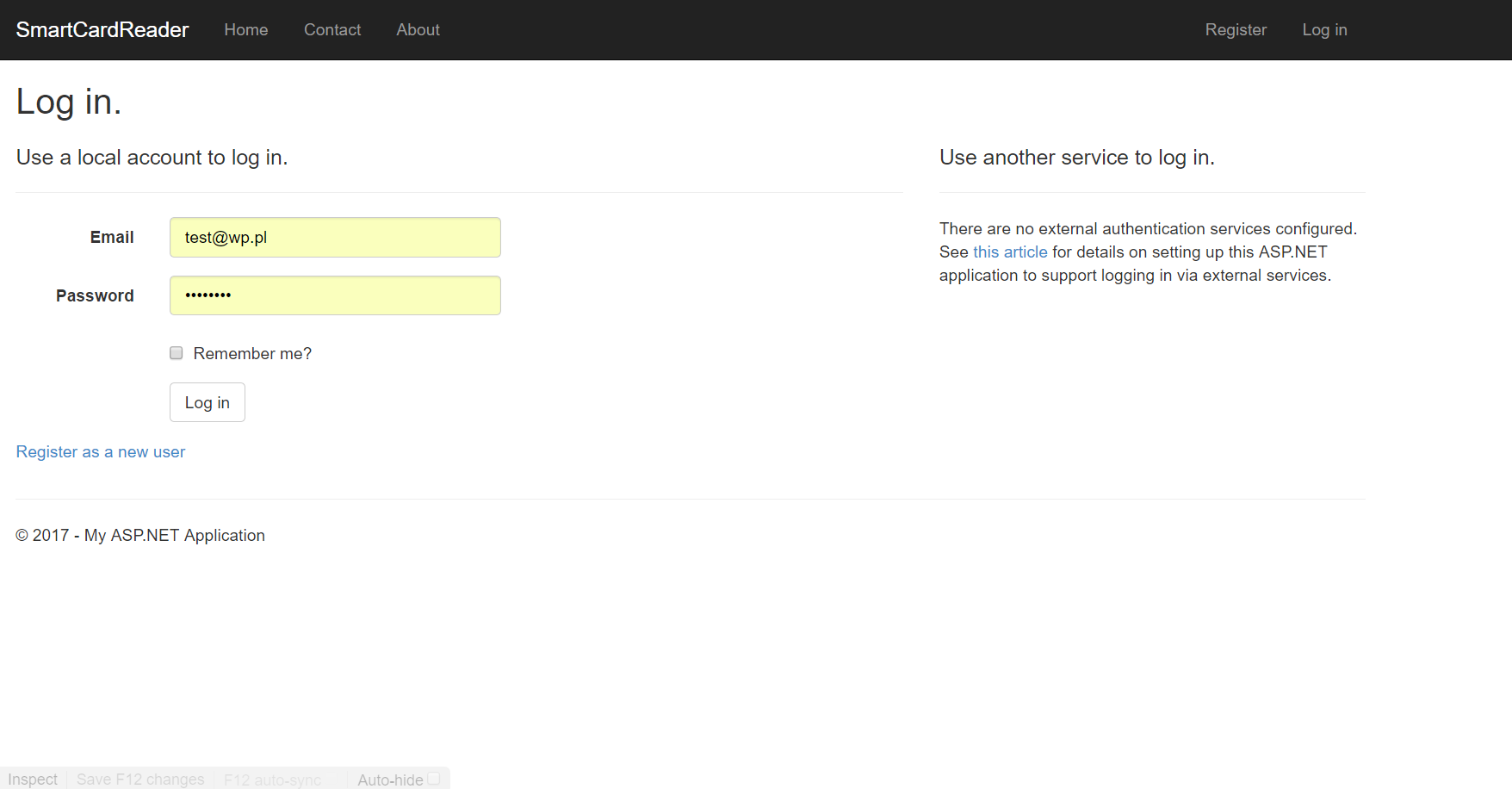
6.1 Instrukcja użytkowania aplikacji webowej

Zanim zaczniemy być aktywnym użytkownikiem naszej aplikacji należy się zarejestrować.

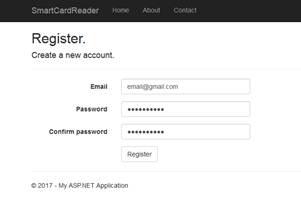
Na zdjęciu nr 1 widzimy interfejs startowy modułu webowego. Na nim wybieramy opcję "Register" i rozpoczynamy rejestrację. Na zdjęciu nr 2 widzimy szczegóły tej operacji.



*zdjęcie nr 1 Ekran startowy.*

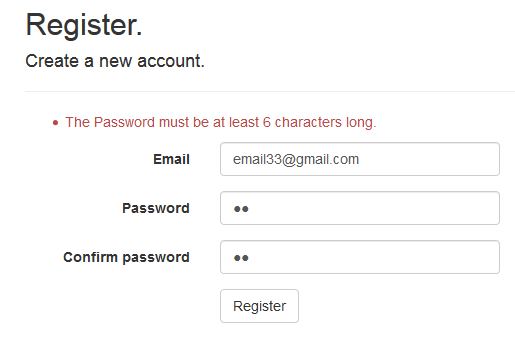


Zdjęcie nr 2 Ekran logowania

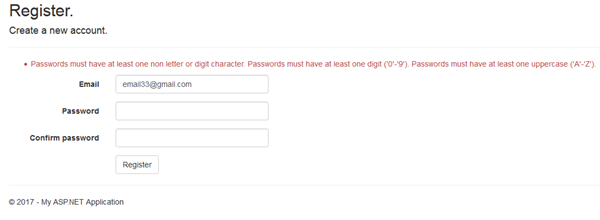


*zdjęcie nr 3 Ekran rejestracji.*

Podajemy nasz email oraz hasło, które musi spełniać określone wymagania: mieć więcej niż 6 znaków i z czego hasło musi zawierać przynajmniej jedną cyfrę, jedną wielką literę, i jeden znak specjalny. Jeżeli nie spełnimy tych wymagań, program zgłasza wyjątek który widać na zdjęciu nr 3 i 4.

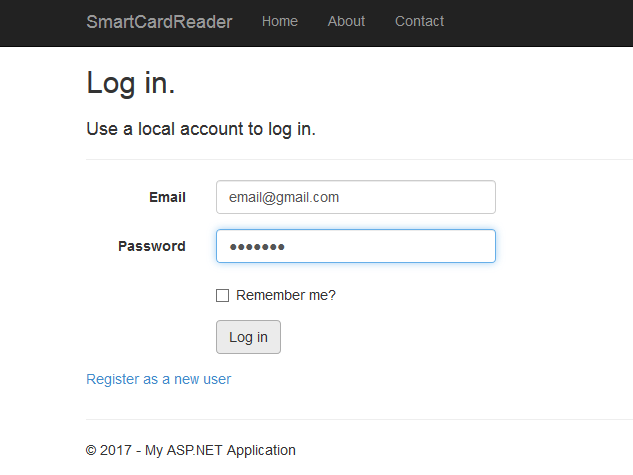


*zdjęcie nr 4 Ekran rejestracji ze zgłaszający wyjątek o za krótkim haśle.*



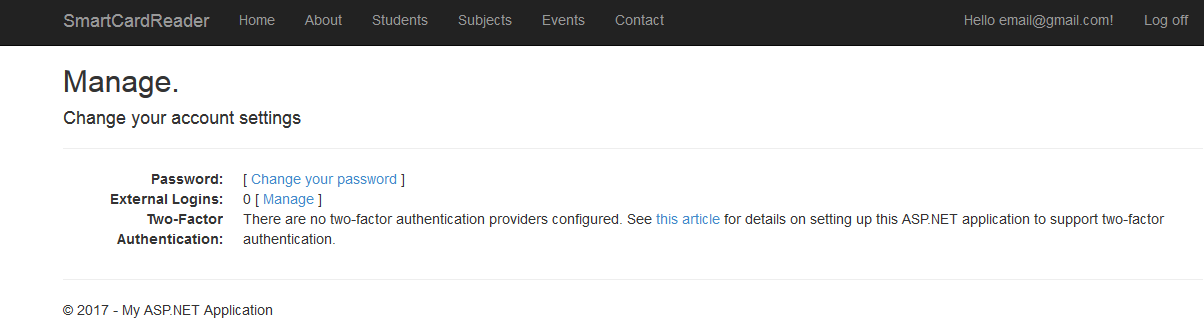
*zdjęcie nr 5 Ekran rejestracji ze zgłaszający wyjątek o haśle nie spełniającym określonych wymagań.*

Jeżeli posiadamy konto możemy się zalogować tak jak widać to na zdjęciu nr 5, podając ten sam email i hasło, które podawaliśmy przy rejestracji.

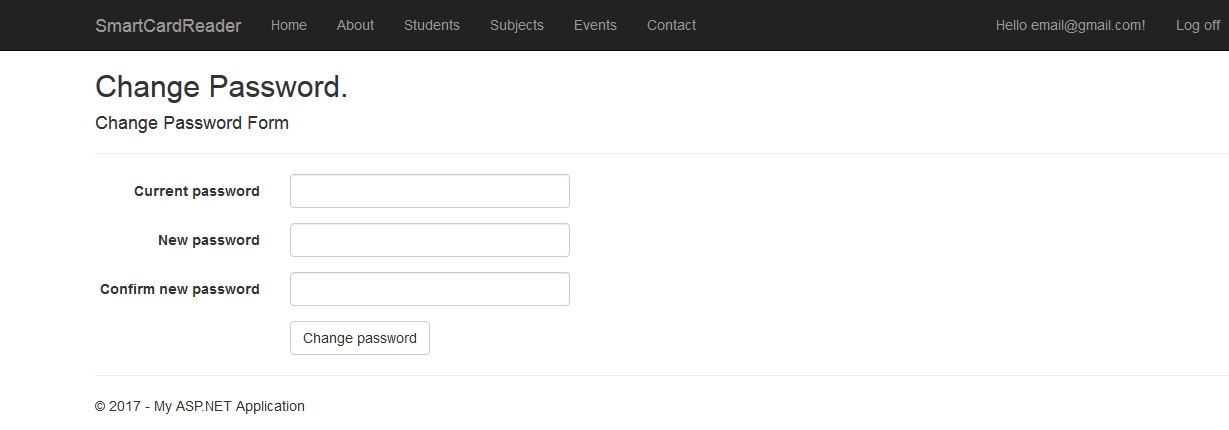


*zdjęcie nr 6 Ekran logowania.*

Po zalogowaniu na interfejsie pojawiają się nam nowe opcje: "Students","Subjects" i"Class". Możemy także zarządzać naszym kontem czyli zmienić hasło. Aby to uczynić klikamy na nasz email znajdujący się na pasku. Wchodzimy do ekranu zarządzania naszym kontem, który widać na zdjęciu 6. Następnie wybieramy opcję "Change your password", która przenosi nas do ekranu zmiany hasła, który widzimy na zdjęciu 7.

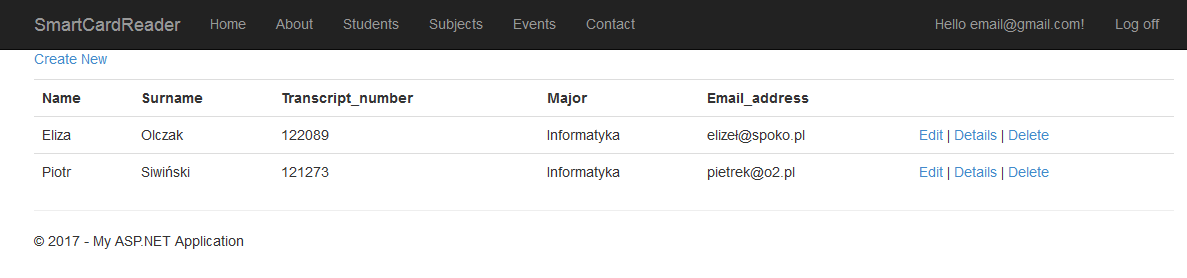


*zdjęcie nr 7 Ekran zarządzania kontem.*



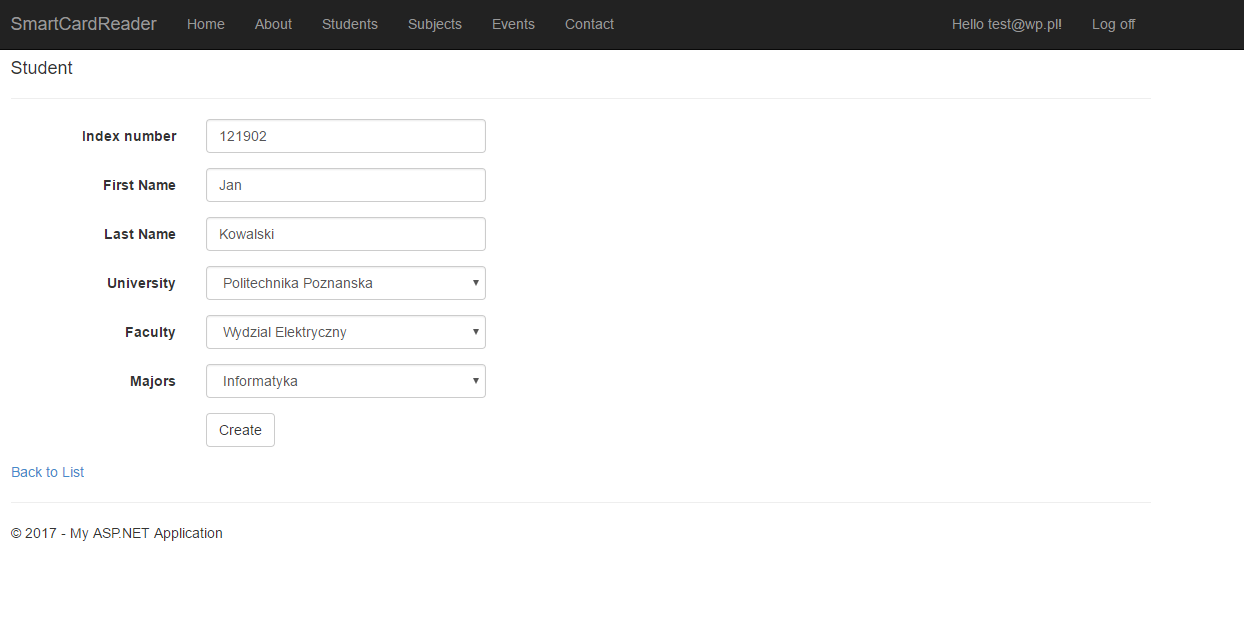
*zdjęcie nr 8 Ekran zmiany hasła.*

Jeżeli chcielibyśmy dodać studenta do naszych zajęć wybieramy na pasku opcję "Students". Otwiera to listę dodanych już studentów, którą widać na zdjęciu 8. Wybieramy opcję "Create New", która otwiera nam odpowiedni interfejs. Tak jak widzimy na zdjęciu 9, należy podać imię, nazwisko, numer indeksu, kierunek i email studenta. Aby dodać kierunek najpierw należy wybrać Uczelnię.

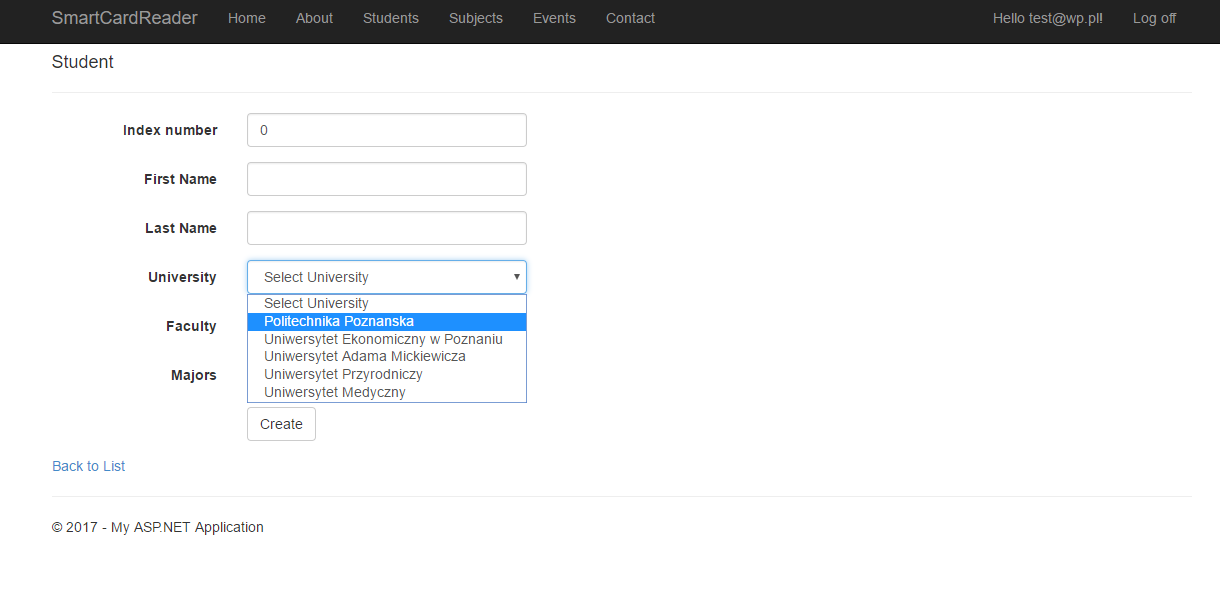


*zdjęcie nr 9 Lista studentów*

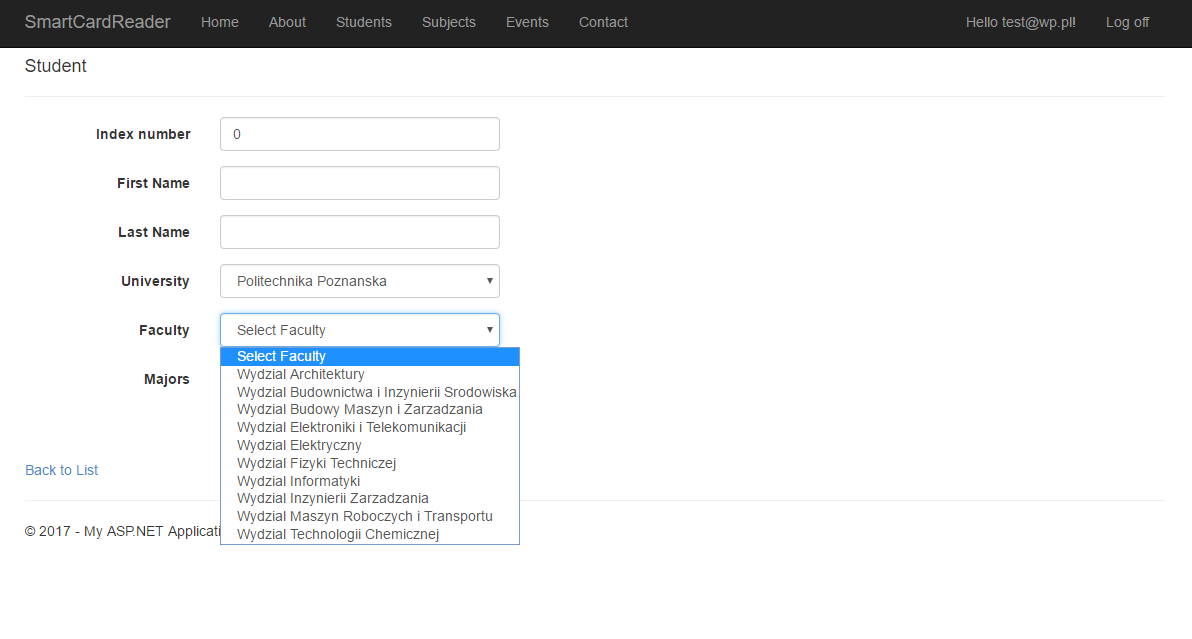
Z ekranu listy studentów, możemy edytować dane wybierając opcję "Edit", podglądać szczegóły wybierając opcję "Details" i usuwać studentów wybierając opcję "Delete".



*Zdjęcie nr 10 Ekran dodawania studenta.*

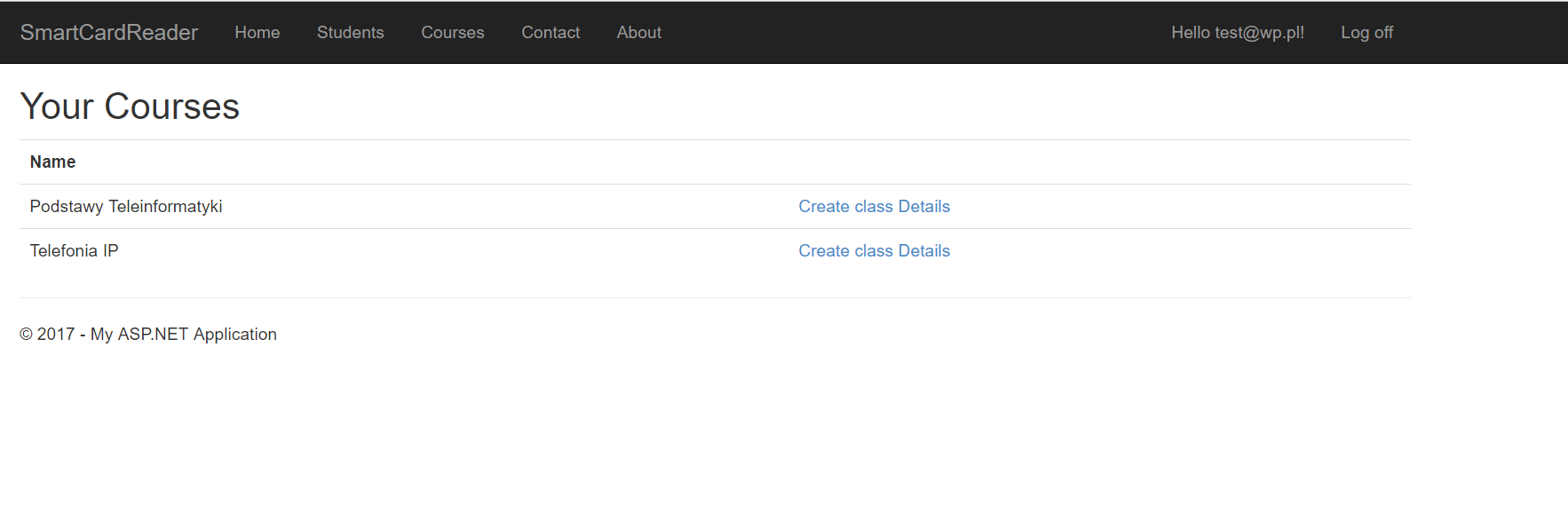


*Zdjęcie nr 11 Ekran dodawania studenta.*



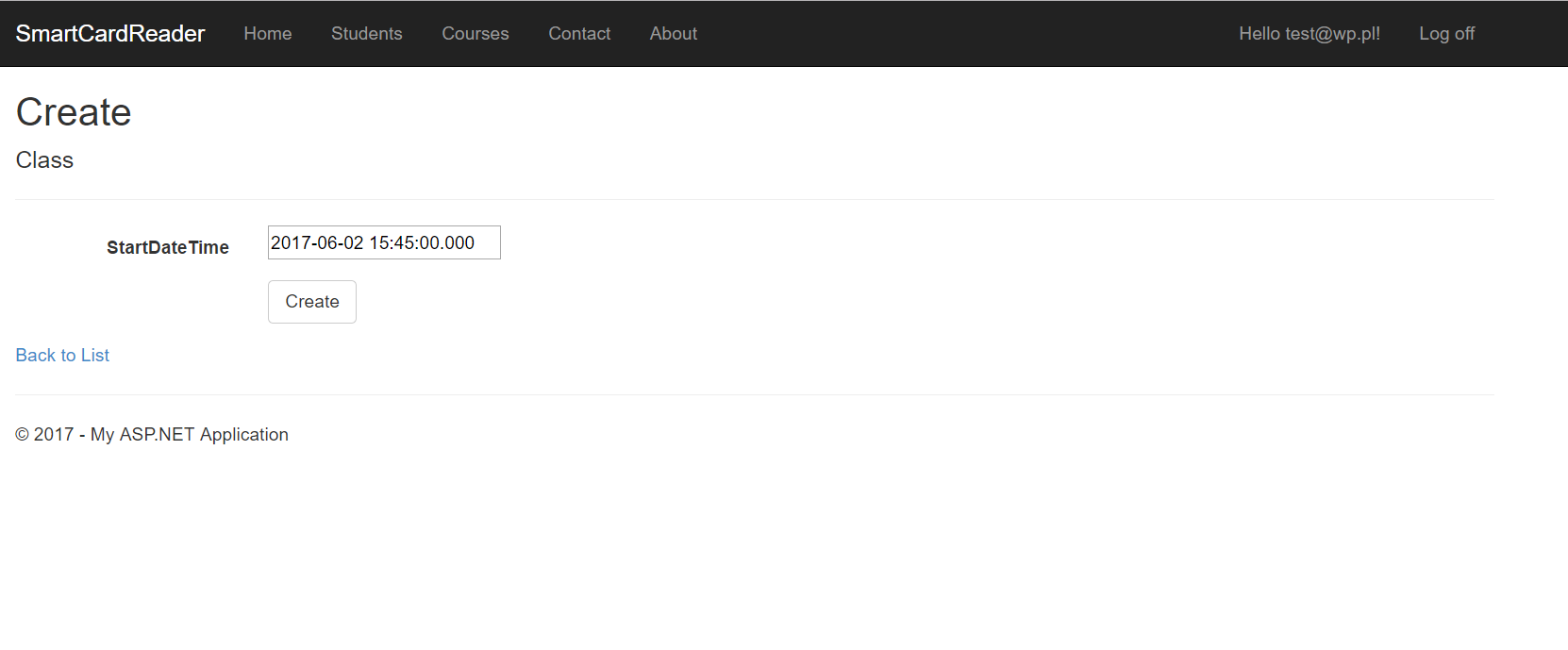
*Zdjęcie nr 12 Ekran dodawania studenta.*

Wybierając opcję "Courses" możemy zarządzać przedmiotami, które prowadzimy i dodawać zajęcia. Na zdjęciu nr 10 widzimy listę dodanych przedmiotów. Celem dodania zajęć wybieramy opcję "Create Class", która otwiera nam odpowiedni interfejs widoczny na zdjęciu 11. Należy podać datę wraz z godziną rozpoczęcia zajęć.



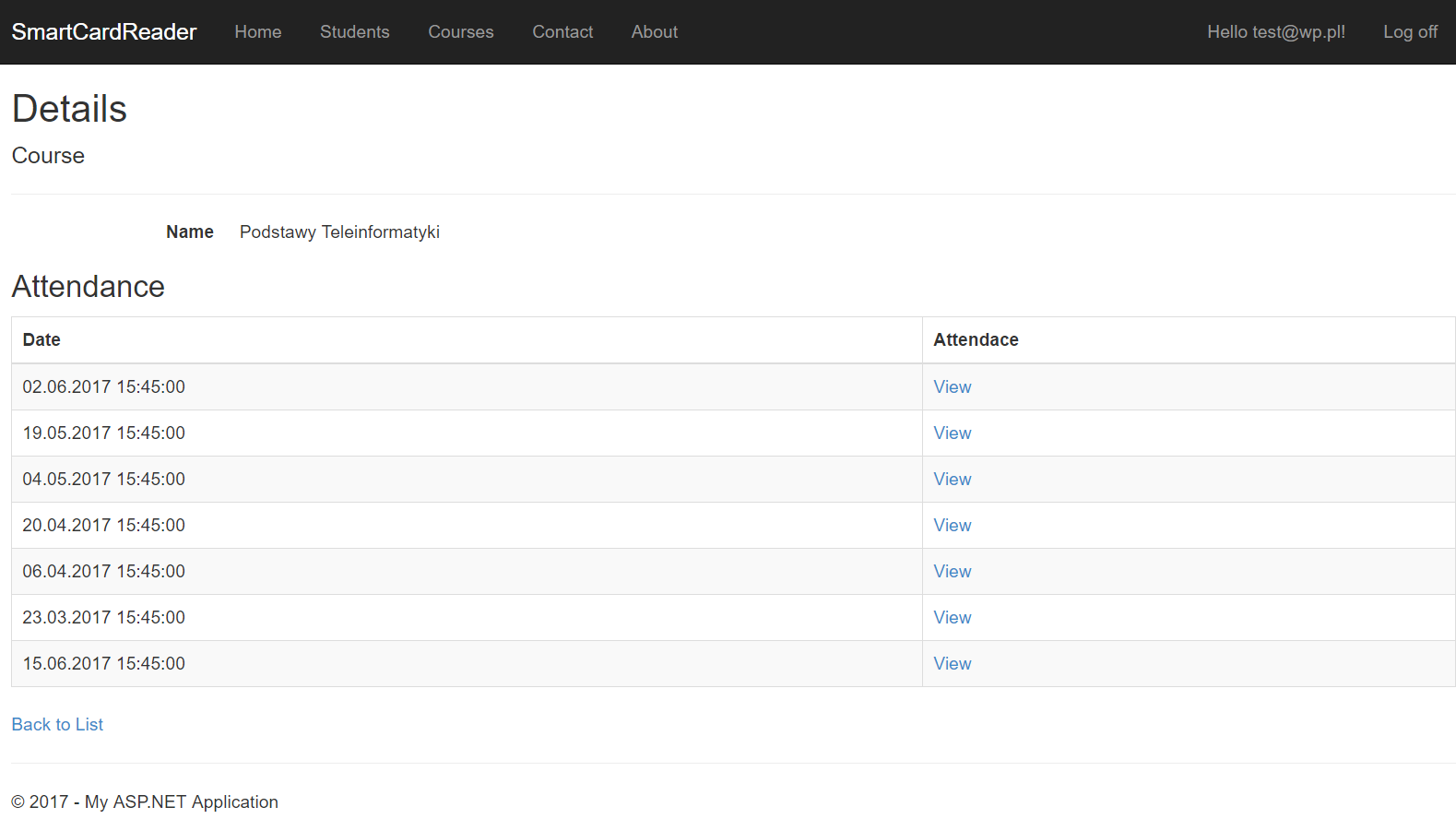
*zdjęcie nr 13 Lista przedmiotów.*

Analogicznie z ekranu listy przedmiotów, możemy edytować dane wybierając opcję "Edit", podglądać szczegóły wybierając opcję "Details" i usuwać przedmioty wybierając opcję "Delete".



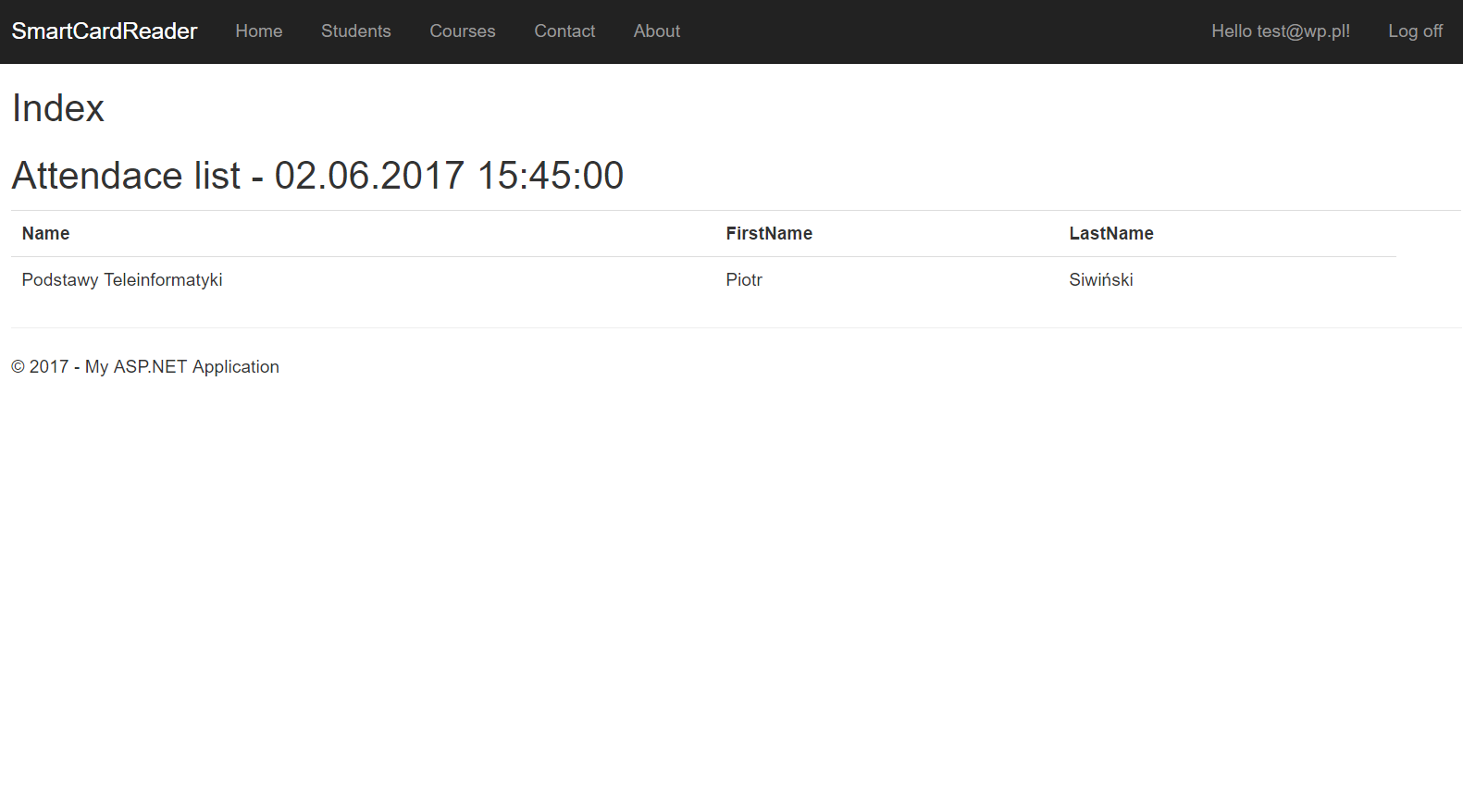
*Zdjęcie nr 14 Ekran dodawania zajęć.*

Możemy również oglądać listę zajęć oraz listę obecności. W tym celu należy wybrać opcję „Details”. Zostanie nam przedstawiony ekran widoczny na zdjęciu

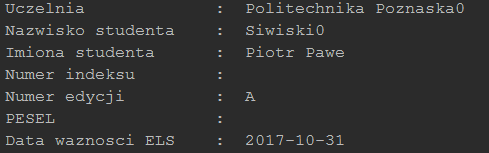


*Zdjęcie nr 15 Ekran dodawania przedmiotu.*

Z tego widoku możemy oglądać studentów obecnych na danych zajęciach. W tym celu wybieramy opcję „View”.  
Ujrzymy listę obecności jak pokazano na zdjęciu nr 13



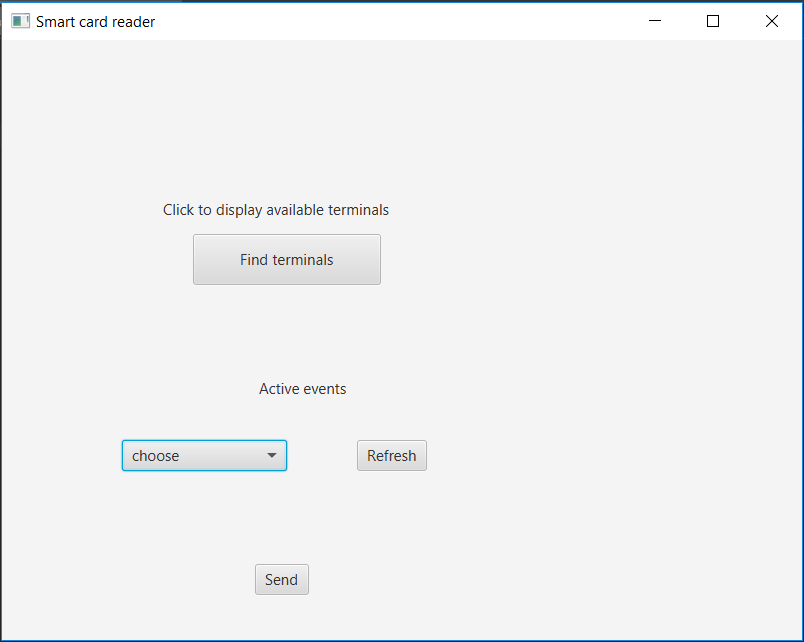
*Zdjęcie nr 16 Ekran dodawania przedmiotu.*



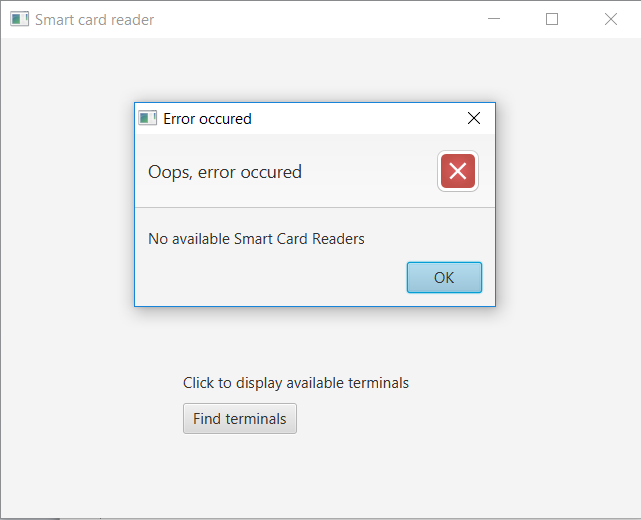
*Zdjęcie nr 17 Dane odczytane z legitymacji.*

6.2 Instrukcja użytkowania interfejsu czytnika kart

Uruchamiamy aplikacje czytnika kart. W celu wczytania legitymacji do czytnika wybieram opcję „Find terminals” widoczną na zdjęciu nr 13. W razie braku odpowiedniego sprzętu, złego jego podłączenia lub niespełniania przez oprzyrządowanie wymagań określonych przez oprogramowanie pojawi się odpowiednia informacja widoczna na zdjęciu nr 14.

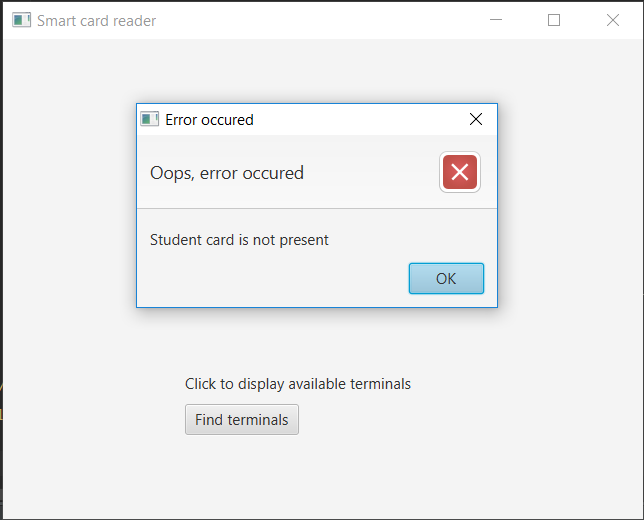


*Zdjęcie nr 18 Interfejs czytnika kart.*



*Zdjęcie nr 19 Interfejs czytnika kart pokazujący błąd braku sprzętu do wczytywania SMART CARD.*

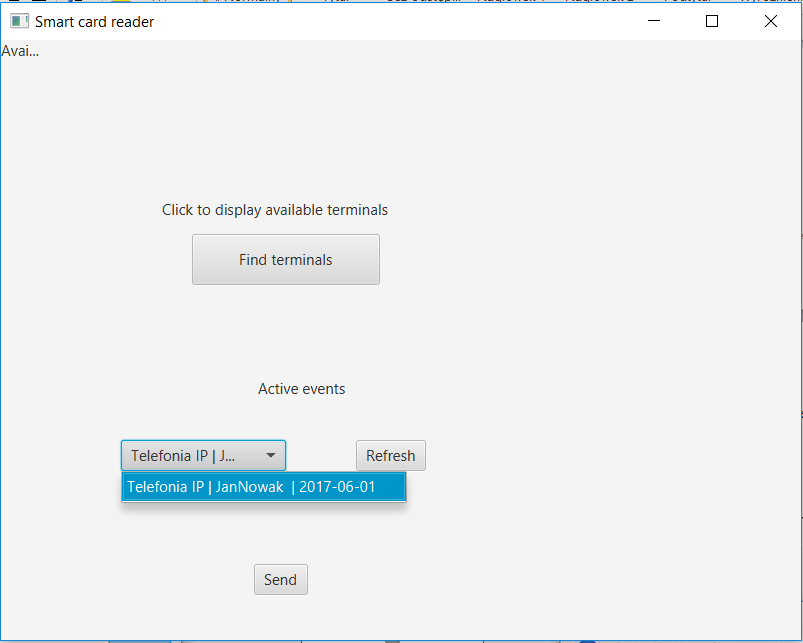
Jeżeli nie przyłożymy legitymacji studenckiej lub zrobimy tego w odpowiedni sposób pokaże nam się okno ze stosowną informacją widoczną na zdjęciu nr 15.



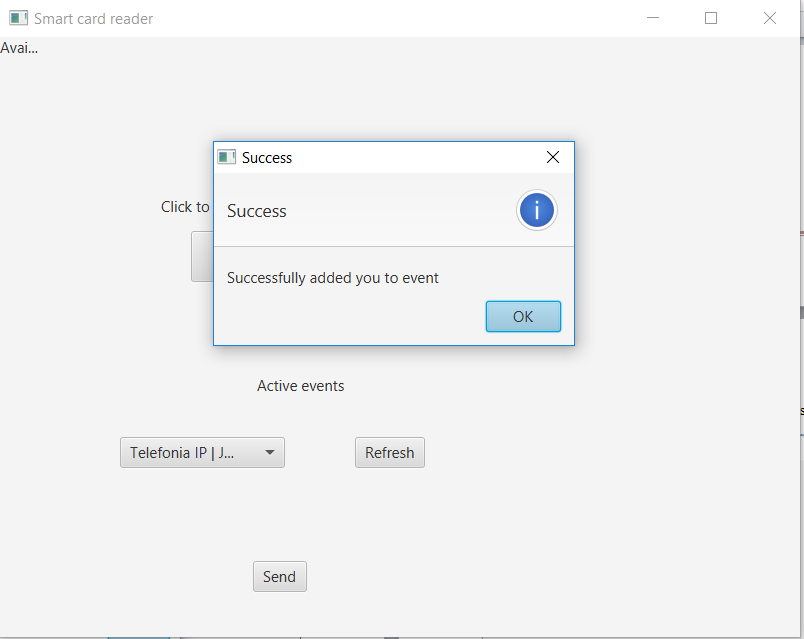
*Zdjęcie nr 20 Interfejs czytnika kart pokazujący brak lub błędne przyłożenie legitymacji studenckiej.*

Jeżeli operacja zakończy się sukcesem pojawi się okno z danymi z naszej legitymacji zawierające pierwsze i drugie imię, nazwisko, numer indeksu oraz nazwę uczelni widoczne na zdjęciu 16. Informacje te zostaną zapisane w bazie danych i będzie można je obejrzeć oraz edytować w aplikacji webowej.

Celem zapisania się na zajęcia należy wybrać zajęcia z listy. W tym celu klikamy na przycisk „Refresh”



*Zdjęcie nr 21 Interfejs czytnika kart pokazujący brak lub błędne przyłożenie legitymacji studenckiej.*



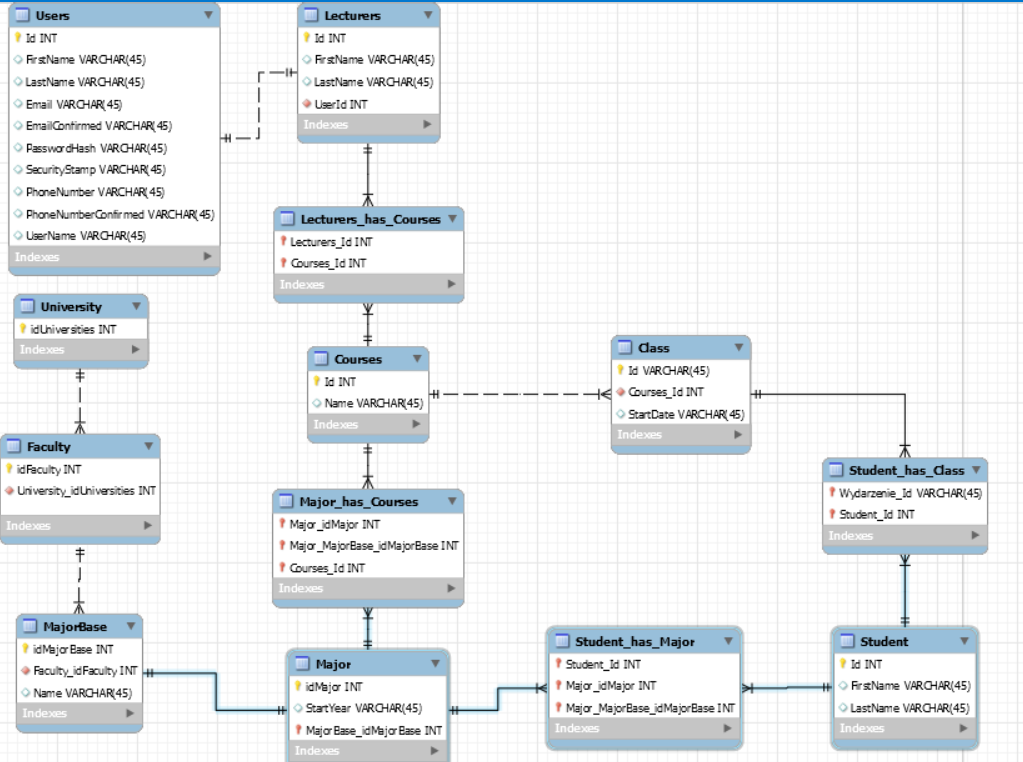
*Zdjęcie nr 22 Interfejs czytnika kart pokazujący dane wczytane z legitymacji studenckiej.*

7. Model aplikacji

W tym miejscu zawierają się informację dotyczące aplikacji, szczególnie ważne dla administratora systemu. Każdy podpunkt tego rozdziału skupia się na innym aspekcie implementacji.

7.1 Model relacyjny bazy danych

Relacyjna baza danych to opisany i zorganizowany zbiór tabel połączonych relacjami – związkami między sobą. Ten sposób przechowywania informacji pozwala na uniknięcie redundancji (powtarzania się danych) oraz przeprowadzanie analiz na podstawie wielu tabel. Każda tabela składa się z rekordów. Poszczególne rekordy składają się z pól przechowujących jedną daną. Na rysunku 1 widzimy model bazy danych dotyczący naszego oprogramowania.



Rys. 1 Model bazy danych.

7.2 Lista aktorów

Aktor (ang. *actor*) jest rolą, którą pełni użytkownik w stosunku do systemu oraz przypadków użycia. Reprezentuje spójny zbiór ról, które są odgrywane przez użytkowników przypadku użycia w czasie interakcji z tym przypadkiem. Aktorem może być człowiek, urządzenie lub inny system. Nie musi  być fizycznym obiektem. Istotne jest, by pełnił określoną funkcję wobec systemu i przypadku użycia, którego używa. Aktor reprezentuję rolę, w którą człowiek, urządzenie bądź inny system może się wcielić w trakcie współpracy z modelowanym systemem. Szczególną uwagę należy zwrócić na fakt, iż aktor zawsze reprezentuje otoczenie systemu (nie jest częścią systemu) i musi być w interakcji, chociaż  
z jednym przypadkiem użycia.

W tabeli 1 widzimy zbiór aktorów występujących w systemie oraz ich krótkie opisy.

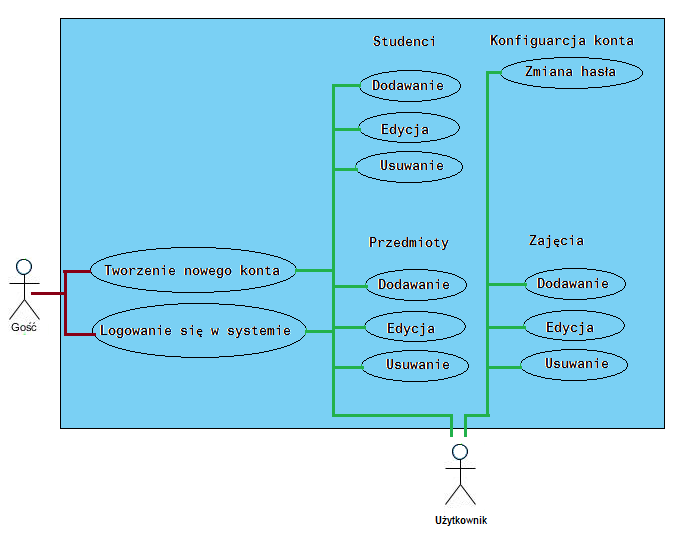
|  |  |
| --- | --- |
| **Aktor** | **Opis** |
| Gość | Użytkownik niezalogowany. |
| Użytkownik | Użytkownik zalogowany. |

*Tabela 1 Aktorzy.*

7.3 Diagram przypadków użycia

Diagram przypadków użycia (ang. use case diagram) jest diagramem, który przedstawia funkcjonalność systemu wraz z jego otoczeniem. Pozwalają na graficzne zaprezentowanie własności systemu tak, jak są one widziane po stronie użytkownika. Służą do zobrazowania usług, które są widoczne z zewnątrz systemu. Diagram przypadków użycia, mimo iż jest zbudowany z kilku elementów, odgrywa najważniejszą rolę w procesie projektowania systemu; opisuje bowiem wymagania funkcjonalne, jakim system musi sprostać, i otoczenie, w którym się znajduje. Diagram ten jest agregatem funkcji usług, które wykonuje system. Poza specyfikacją, diagram ten pozwala na identyfikację funkcjonalności, weryfikację postępów w modelowaniu i implementacji, a także wspomaga komunikację pomiędzy uczestnikami projektu.

Na rysunku 2 widzimy, że większość możliwości dostępna jest jedynie z poziomu zalogowanego użytkownika. Widać, że jedynie samo logowanie oraz rejestracja nie wymagają uprzedniego zalogowania się w systemie.



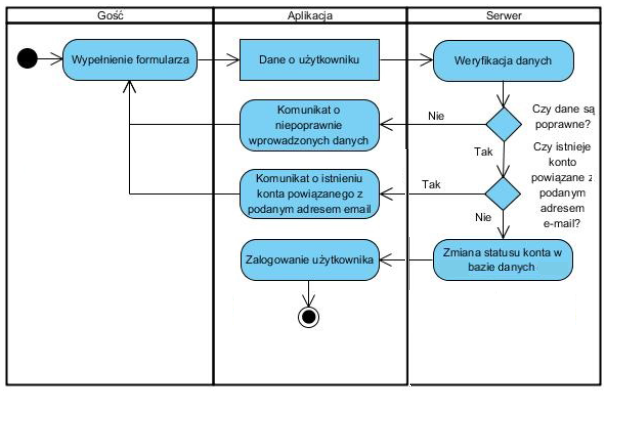
*Rys. 2 Diagram przypadków użycia*

7.4 Diagramy aktywności

Diagram czynności (ang. activity diagram) jest diagramem interakcji, który służy do modelowania dynamicznych aspektów systemu. Jego zasadniczą funkcją jest przedstawienie sekwencji kroków, które są wykonywane przez modelowany fragment systemu. Diagram sekwencji pozwala także na prezentację przepływów współbieżnych oraz na zaprezentowanie zmian stanów obiektów podczas przechodzenia pomiędzy czynnościami. Definiując ogólnie, diagram czynności jest stosowany do modelowania behawioralnych aspektów systemu i obrazuje strumień kolejno wykonywanych czynności (ang. fast track).

Rejestracja

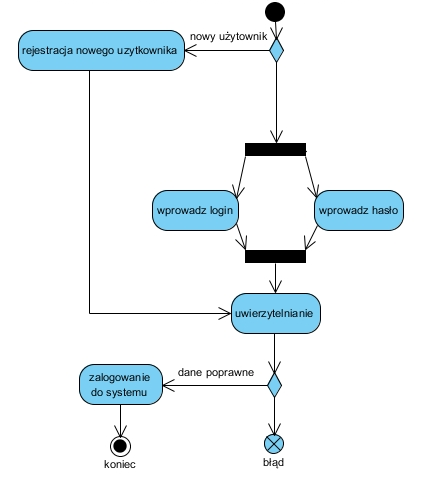
Aby się zarejestrować w naszym systemie użytkownik musi podać odpowiednie dane: email i hasło. Jeżeli hasło nie spełnia wymogów lub konto powiązane z tym mailem już istnieje to użytkownik nie utworzy konta i pozostanie w interfejsie rejestracji. Jeżeli operacja przebiegnie pomyślnie zmiana zostaje odnotowana w bazie, a użytkownik zostanie zalogowany.



*Rys. 3 Rejestracja*

Logowanie

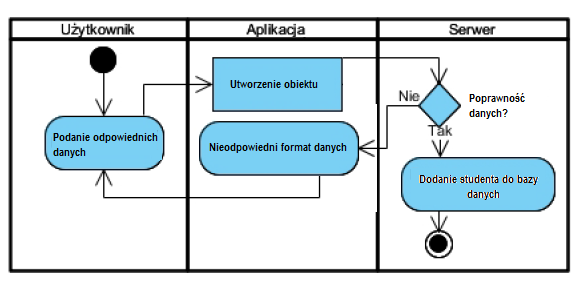
Aby się zalogować w naszym systemie użytkownik musi podać dane, które podawał przy rejestracji: email i hasło. Jeżeli email lub hasło jest nie poprawne użytkownik zostaje o tym poinformowany i wraca do ekranu logowania, a operacja logowania nie udaje się. Jeżeli wszystkie dane są poprawne użytkownik loguje się.



*Rys. 4 Logowanie*

Dodawanie studentów

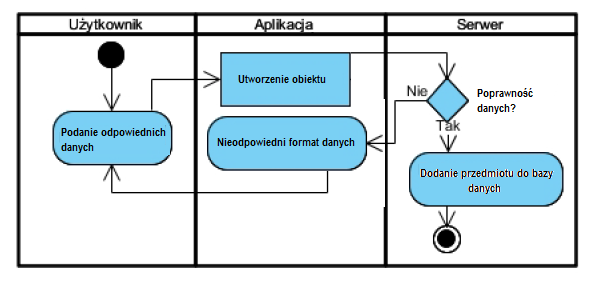
Aby dodać studenta do naszego systemu użytkownik musi podać odpowiednie dane: imię, nazwisko, numer indeksu oraz kierunek. Jeżeli wszystkie dane są poprawne użytkownik dodaje obiekt student do bazy danych.



*Rys. 5 Dodanie studenta*

Dodawanie przedmiotów

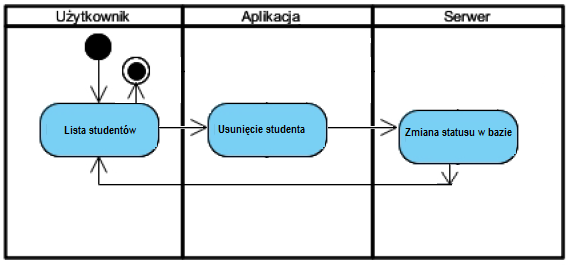
Aby dodać przedmiot do naszego systemu użytkownik musi podać odpowiednie dane: imię prowadzącego, nazwisko prowadzącego oraz nazwę przedmiotu. Jeżeli wszystkie dane są poprawne użytkownik dodaje obiekt przedmiot do bazy danych.



*Rys. 6 Dodanie przedmiotu*

Usuwanie studentów

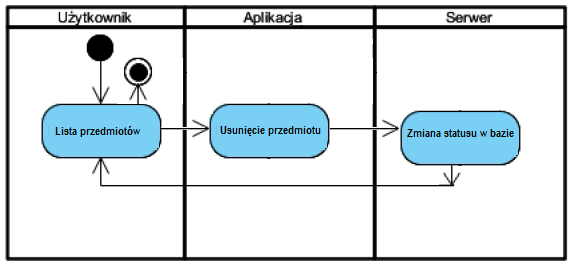
Aby usunąć studenta z naszego systemu użytkownik musi wybrać odpowiednią opcję. Status zostaje natychmiast zmieniony w bazie danych.



*Rys. 7 Usunięcie studenta*

Usuwanie przedmiotów

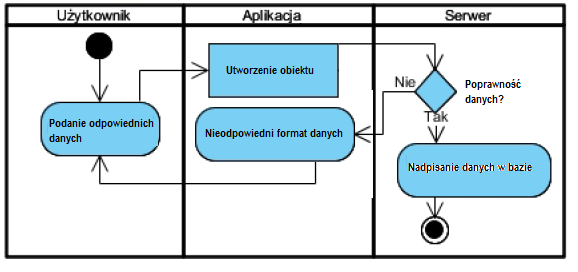
Aby usunąć przedmiot z naszego systemu użytkownik musi wybrać odpowiednią opcję. Status zostaje natychmiast zmieniony w bazie danych.



*Rys. 8 Usunięcie przedmiotu*

Edycja studentów

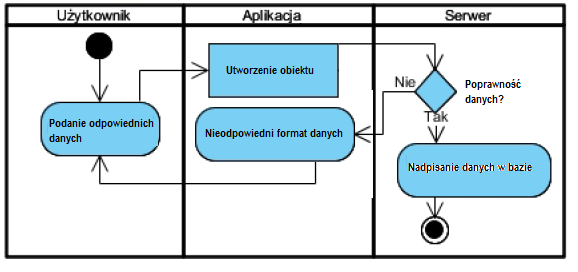
Aby zmienić dane studenta w naszym systemie użytkownik musi wybrać odpowiednią opcję, a następnie zmienić dane. Informacje zostają natychmiast zmienione w bazie danych.



*Rys. 9 Edycja studenta*

Edycja przedmiotów

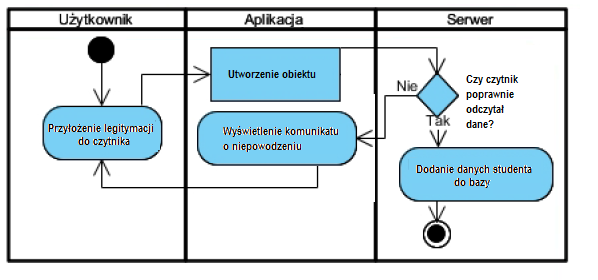
Aby zmienić dane przedmiotu w naszym systemie użytkownik musi wybrać odpowiednią opcję, a następnie zmienić dane. Informacje zostają natychmiast zmienione w bazie danych.



*Rys. 10 Edycja przedmiotu*

Dodawanie studentów za pomocą legitymacji

Aby dodać studenta do naszego systemu za pomocą legitymacji użytkownik musi przyłożyć ja do czytnika. Aplikacja pobierze odpowiednie dane: imię, nazwisko, numer indeksu oraz kierunek. Jeżeli wszystkie dane zostaną poprawnie odebrane, w bazie zostanie dodany obiekt student.



8. Napotkane problemy

W trakcie naszego projektu napotkaliśmy kilka trudności. Musieliśmy się z nimi zmierzyć i wyszliśmy zwycięsko.

Czytnik

Na początku naszej przygody zamówiliśmy czytnik widoczny na zdjęciu 17. Niestety nie spełniał on naszych wymogów i odmawiał współpracy z kodem.

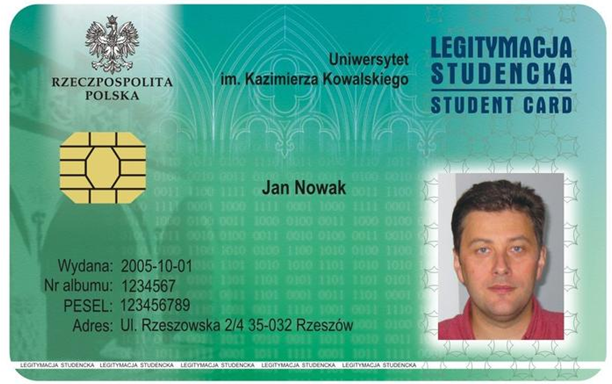


*Zdjęcie 23 Czytnik*

Rozwiązanie: Kupiliśmy nowy czytnik. Taki spełniający nasze wygórowane wymagania. Nie jesteśmy pewni, czy było to optymalne rozwiązanie.

Stare legitymacje

Podkoniec, w trakcie testów okazało się, że nie jesteśmy w stanie wczytywać danych ze starych legitymacji.



*Zdjęcie 24 Nieaktualna legitymacja*

Rozwiązanie: Nasz program nie obsługuje i obsługiwać nie będzie starych legitymacji studenckich. Jeżeli nasz system wszedłby w życie, osoby posiadające stare legitymacje musiałaby je zmienić. Albo uczelnia musiała by mieć różne rodzaje czytników i oprogramować do nich, co nie byłoby najkorzystniejszą opcją.