

Academic Data Deliverer

Programowanie Obiektowe i Graficzne

Artur Bednarczyk, Dawid Grajewski, Damian Kwaśniok

Politechnika Śląska

Wydział Matematyki Stosowanej

Informatyka, semestr IV

3 czerwca 2018



**Politechnika
Śląska**

Spis treści

1	Harmonogram prac	3
2	Opis projektu	4
2.1	Opis	4
2.2	Projekt UI	4
2.2.1	Logowanie	4
2.2.2	Rejestracja	4
2.2.3	Profil	5
2.2.4	Notatka	5
2.3	Funkcjonalności	6
2.3.1	Logowanie i Rejestracja	6
2.3.2	Przypasanie do grup	6
2.3.3	Notatki	6
3	Technologie	6
3.1	Oprogramowanie	6
3.2	Technologie	6
4	Implementacja	7
4.1	Podział projektu na pliki	7
4.2	Schemat Modelu Obiektowego	8
4.3	Interfejsy	8
5	Testy	12

1 Harmonogram prac

Co	Kto	Kiedy
Projekt bazy danych	Bednarczyk	04.2018
Utworzenie bazy danych	Grajewski	04.2018
Połączenie z bazą danych	Grajewski	10.05.2018
Projekty graficzne	Bednarczyk	13.05.2018
Szkielet struktury MVP Wyśrodkowanie widoku z możliwością rozszerzenia okna	Bednarczyk	23.05.2018
Logowanie	Kwaśniok	26.05.2018
Sesja i Hash	Grajewski	26.05.2018
Przełączanie między widokami	Grajewski	28.05.2018
Wyświetlanie listy notatek Wyświetlanie notatki Zapis notatki do pliku Wyświetlanie danych użytkownika w profilu Wyświetlanie listy subskrybowanych kierunków w profilu	Bednarczyk	31.05.2018
Rejestracja	Kwaśniok	31.05.2018
Dodawanie i anulowanie subskrypcji dla kierunku	Bednarczyk	02.06.2018
Testowanie i poprawki	Bednarczyk Grajewski Kwaśniok	03.06.2018
Dokumentacja	Bednarczyk	Cały czas

2 Opis projektu

2.1 Opis

Aplikacja dla studentów umożliwiająca szybki i łatwy podgląd dostępnych materiałów! Każdy student może wybrać kierunki studiów, którymi jest zainteresowany i oglądać przypisane do nich notatki! W każdej chwili może dodać nowe lub usunąć nieinteresujące go kierunki ze swojej listy. Aplikacja jest prosta w użytkowaniu, jednak wymaga połączenia z internetem.

2.2 Projekt UI

2.2.1 Logowanie

ADD - □ X

LOGIN

HASŁO

LOGIN

REJESTRACJA

2.2.2 Rejestracja

ADD - □ X

LOGIN

HASŁO

IMIE

NAZWISKO

TEL

MAIL

UCIELENIA

DYPLOM

KIERUNKI

COFNIJ ZAREJESTRUJ

TEXTBOX PASSWORD DROPDOWN

2.2.3 Profil

ADD		SUB	
UCZELNIA	IMIE	UCZELNIA	
WYDZIAŁ	LOGIN	WYDZIAŁ	
KIEROWNIC	MAIL	KIEROWNIC	
PRZEDMIOT	TEL	CANCEL	SUB
NOTATKI			
NOWY SUB			
USUN SUB			

DROPPDOWN LIST

2.2.4 Notatka

ADD	
WLOGUJ	POBIERZ
UCZELNIA	PROFIL
WYDZIAŁ	TREŚĆ
KIEROWNIC	
PRZEDMIOT	
NOTATKA	

2.3 Funkcjonalności

2.3.1 Logowanie i Rejestracja

Założ swoje konto, a będziesz mógł korzystać z aplikacji z każdego urządzenia, na którym jest zainstalowana i posiada dostęp do internetu! Zachowaj Twoje listy kierunków na swoim koncie!

2.3.2 Przypasanie do grup

Wybierz kierunki, które Cię interesują i przeglądaj materiały z nimi powiązane. W każdej chwili możesz dopisać do swojej listy nowe kierunki lub usunąć już niepotrzebne.

2.3.3 Notatki

Przeglądaj dostępne materiały z listy kierunków, którą sam utworzyłeś. Jeśli chcesz mieć dostęp offline to pobierz materiał w formie pliku!

3 Technologie

3.1 Oprogramowanie

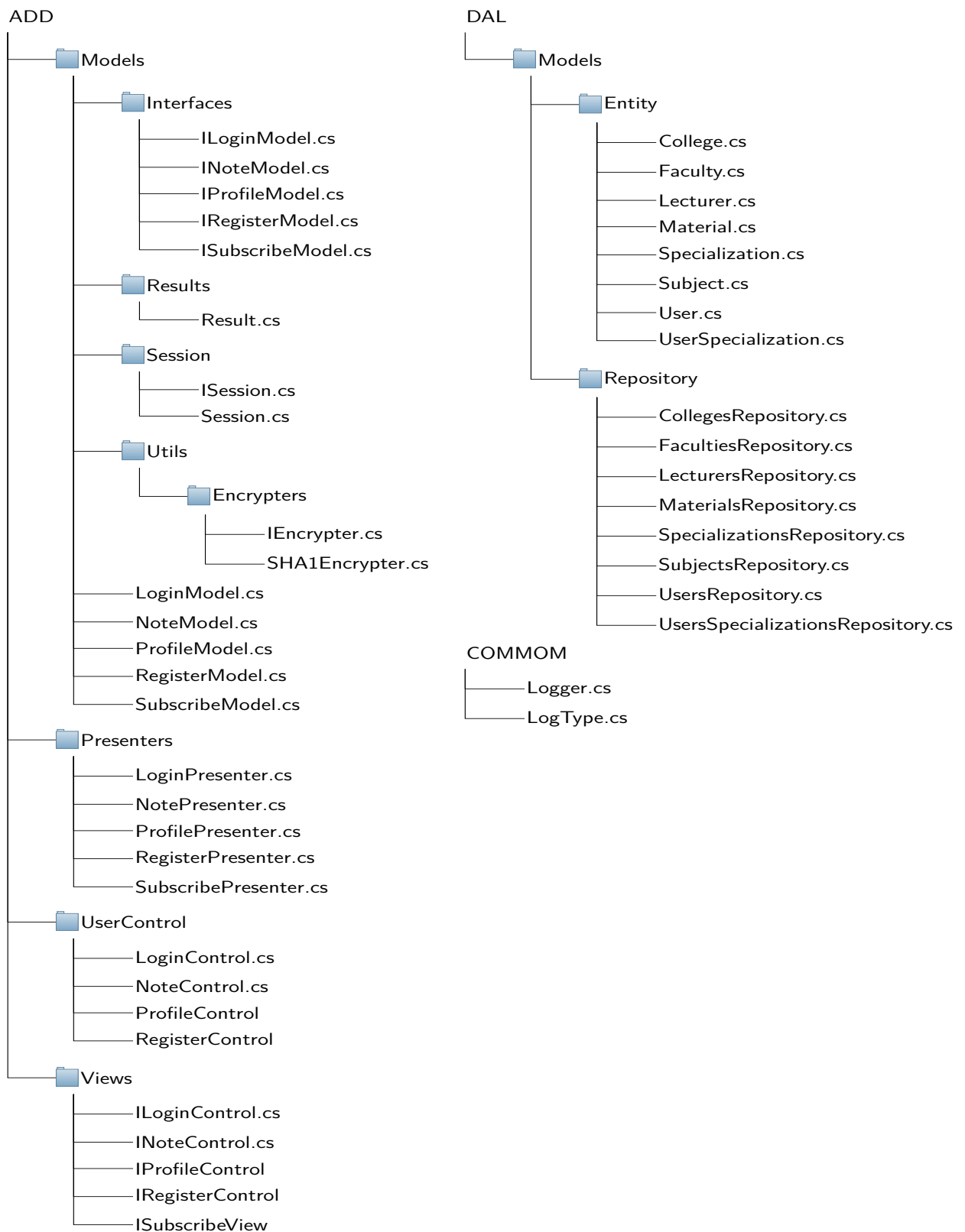
- Visual Studio 2015 - Środowisko programistyczne.
- SourceTree - Kontrola wersji.
- GitHub - Repozytorium do przechowywania wersji online.
- Heroku - Chmura, w której przechowywana jest baza danych.

3.2 Technologie

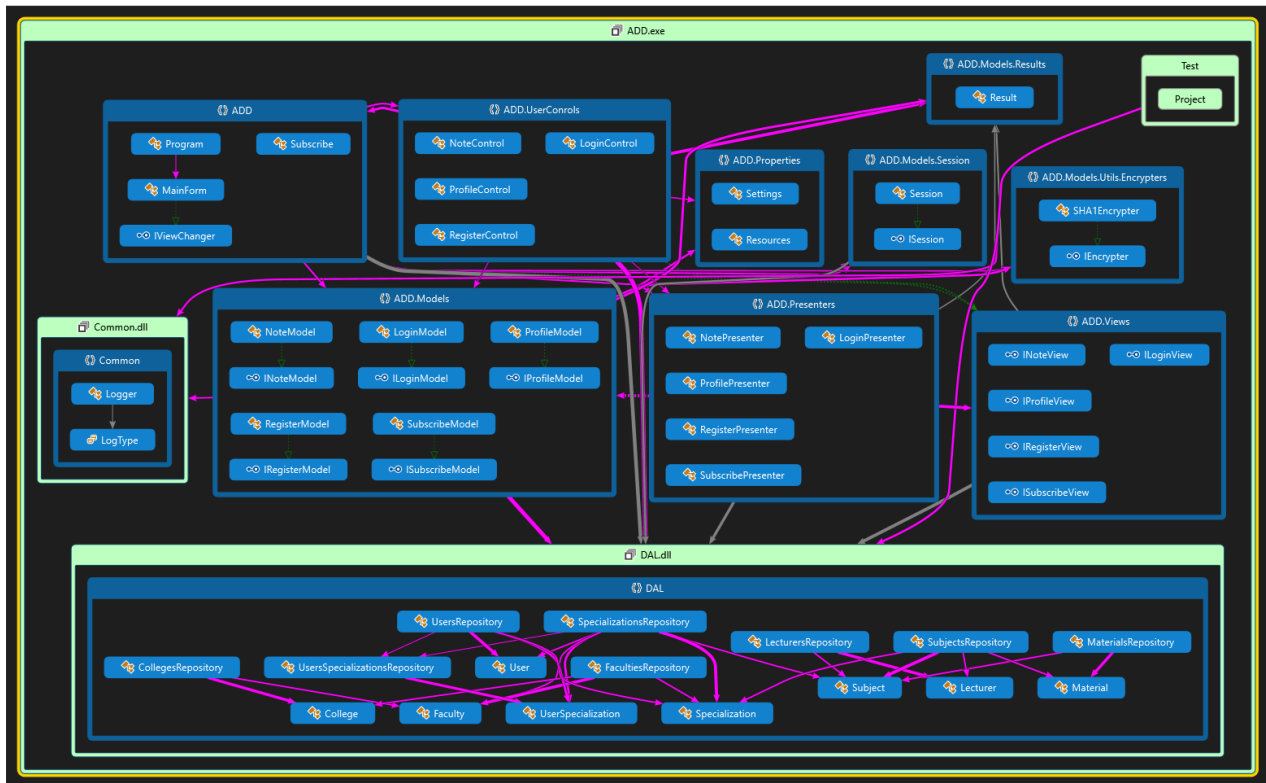
- C#
- .NET
- MySQL

4 Implementacja

4.1 Podział projektu na pliki



4.2 Schemat Modelu Obiektowego



4.3 Interfejsy

- Models

- ILoginModel

```
/* Logowanie na podstawie loginu i hasła, zwraca rezultat */
Result Login(string login, string password);
```

- INoteModel

```
/* Zwraca liste College */
ICollection<College> GetColleges();
/* Na podstawie nazwy College zwraca liste Faculty */
ICollection<Faculty> GetFacultiesInCollege(string collegeName);
/* Na podstawie nazwy Faculty zwraca liste Specialization */
ICollection<Specialization> GetSpecializationsOnFaculty(string
    facultyName);
/* Na podstawie nazwy Specialization zwraca liste Subject */
ICollection<Subject> GetSubjectsFromSpecialization(string
    specializationName);
/* Na podstawie nazwy Subject zwraca liste Material */
ICollection<Material> GetMaterialsFromSubject(string subjectName);
/* Na podstawie sciezki i zawartosci zapisuje do pliku */
bool SaveToFile(string location, string content);
```

– IProfileModel

```
/* Zwraca liste College */
ICollection<College> GetColleges();
/* Na podstawie nazwy College zwraca liste Faculty */
ICollection<Faculty> GetFacultiesInCollege(string collegeName);
/* Na podstawie nazwy Faculty zwraca liste Specialization */
ICollection<Specialization> GetSpecializationsOnFaculty(string
    facultyName);
/* Na podstawie nazwy Specialization zwraca liste Subject */
ICollection<Subject> GetSubjectsFromSpecialization(string
    specializationName);
/* Zwraca aktualnie zalogowanego uzytkownika */
User GetUser();
/* Anuluje wybrana subskrypcje */
bool CancelSub(Specialization specialization);
```

– IRegisterModel

```
/* Zwraca liste College */
ICollection<College> GetColleges();
/* Na podstawie nazwy College zwraca liste Faculty */
ICollection<Faculty> GetFacultiesInCollege(College college);
/* Na podstawie nazwy Faculty zwraca liste Specialization */
ICollection<Specialization> GetSpecializationsOnFaculty(Faculty
    faculty);
/* Rejestruje nowego uzytkownika z wybrana Specialization */
Result RegisterNewUser(User newUser, Specialization
    specialization);
```

– ISubscribeModel

```
/* Zwraca liste College */
ICollection<College> GetColleges();
/* Na podstawie nazwy College zwraca liste Faculty */
ICollection<Faculty> GetFacultiesInCollege(College college);
/* Na podstawie nazwy Faculty zwraca liste Specialization */
ICollection<Specialization> GetSpecializationsOnFaculty(Faculty
    faculty);
/* Subskrybuje dla danego uzytkownika wybrana Specialization*/
bool Subscribe(User user, Specialization specialization);
```

– ISession

```
/* Uzytkownik */
User User { get; set; }
/* Konczy sesje uzytkownika */
void EndSession();
```

– IEncrypter

```
/* Zwraca hash na podstawie podanego ciagu znakow */
string Encrypt(string text);
```

- Views

- ILoginView

```
/* Login */
string Login { get; }
/* Haslo */
string Password { get; }
/* Logowanie na podstawie nazwy i hasla uzytkownika */
event Func<string, string, Result> LoginClick;
```

- INoteView

```
/* Pobieranie listy College */
event Func<ICollection<College>> CollegeGetItems;
/* Pobieranie listy Faculty na podstawie wybranego College */
event Func<string, ICollection<Faculty>> FacultyGetItems;
/* Pobieranie listy Specialization na podstawie wybranego Faculty
*/
event Func<string, ICollection<Specialization>>
    SpecializationsGetItems;
/* Pobieranie listy Subject na podstawie wybranej Specialization
*/
event Func<string, ICollection<Subject>> SubjectGetItems;
/* Pobieranie listy Material na podstawie wybranego Subject */
event Func<string, ICollection<Material>> MaterialGetItems;
/* Zapis do pliku wybranego materialu do wybranej sciezki */
event Func<string, string, bool> SaveToFile;
```

- IProfileView

```
/* Pobieranie listy College */
event Func<ICollection<College>> CollegeGetItems;
/* Pobieranie listy Faculty na podstawie wybranego College */
event Func<string, ICollection<Faculty>> FacultyGetItems;
/* Pobieranie listy Specialization na podstawie wybranego Faculty
*/
event Func<string, ICollection<Specialization>>
    SpecializationsGetItems;
/* Pobieranie listy Subject na podstawie wybranej Specialization
*/
event Func<string, ICollection<Subject>> SubjectGetItems;
/* Usuwanie subskrypcji */
event Func<Specialization, bool> DeleteSubscription;
/* Pobieranie aktualnie zalogowanego Usera */
event Func<User> GetUser;
/* Przeladowanie listy subskrybowanych */
void Reload();
/* Imie */
string FirstName { get; }
/* Nazwisko */
string LastName { get; }
/* Mail */
string Mail { get; }
/* Numer Telefonu */
string Phone { get; }
```

– IRegisterView

```
/* Wartosci do formularza rejestracyjnego */
string Login { get; }
string Password { get; }
string FirstName { get; }
string Surname { get; }
string PhoneNumber { get; }
string Email { get; }
string CollegeName { get; }
string FacultyName { get; }
string SpecializationName { get; }
/* Czy te wartosci sa poprawne */
bool IsLoginValid { get; }
bool IsPasswordValid { get; }
bool IsFirstNameValid { get; }
bool IsSurnameValid { get; }
bool IsPhoneNumberValid { get; }
bool IsEmailValid { get; }
bool IsCollegeNameValid { get; }
bool IsFacultyNameValid { get; }
bool IsSpecializationNameValid { get; }
/* Czy wszystkie sa poprawne */
bool AreAllValuesValid();
/* Pobierane listy College */
event Func<ICollection<College>> CollegeComboBoxDropDown;
/* Pobieranie listy Faculty na podstawie wybranego College */
event Func<College, ICollection<Faculty>> FacultyComboBoxDropDown;
/* Pobieranie listy Specialization na podstawie wybranego
Faculty*/
event Func<Faculty, ICollection<Specialization>>
SpecializationComboBoxDropDown;
/* Rejestracja nowego uzytkownika i zapis jego subskrypcji */
event Func<User, Specialization, Result> RegisterClick;
```

– ISubscribeView

```
/* Pobieranie listy College */
event Func<ICollection<College>> CollegeComboBoxDropDown;
/* Pobieranie listy Faculty na podstawie wybranego College */
event Func<College, ICollection<Faculty>> FacultyComboBoxDropDown;
/* Pobieranie listy Specialization na podstawie wybranego Faculty
*/
event Func<Faculty, ICollection<Specialization>>
SpecializationComboBoxDropDown;
/* Subskrypcja wybranej Specialization dla danego uzytkownika */
event Func<User, Specialization, bool> SubscribeClick;
```

5 Testy

Aplikacja została przetestowana manualnie. Przy testach zostały uwzględnione warunki szczególne, takie jak brak dostępu do internetu lub utrata połączenia w czasie działania aplikacji.