



**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**  
**WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI**  
INSTYTUT INFORMATYKI

## **DOKUMENTACJA PROCESOWA**

*System wspomagający obsługę praktyk studenckich*

*The system supporting student internships*

Autorzy:	Igor Dzierwa, Konrad Makuch, Adrian Nędza
Kierunek:	Informatyka
Typ studiów:	Niestacjonarne
Opiekun pracy:	dr hab. inż. Rafał Dreżewski, prof. AGH

Kraków, 2021

## Spis treści

<b>1</b>	<b>Analiza ryzyka</b>	<b>4</b>
<b>2</b>	<b>Przyjęta metodyka pracy</b>	<b>5</b>
<b>3</b>	<b>Role i podział prac</b>	<b>5</b>
3.1	Komunikacja . . . . .	5
3.2	Wykorzystane narzędzia . . . . .	5
3.3	Harmonogram . . . . .	6
<b>4</b>	<b>Przebieg prac</b>	<b>6</b>
4.1	Etap 1 . . . . .	6
4.2	Etap 2 . . . . .	7
4.2.1	Sprint 1 . . . . .	7
4.2.2	Sprint 2 . . . . .	7
4.2.3	Sprint 3 . . . . .	8
4.2.4	Sprint 4 . . . . .	8
4.2.5	Sprint 5 . . . . .	9
4.3	Etap 3 . . . . .	9
<b>5</b>	<b>Dziennik projektu</b>	<b>10</b>
<b>6</b>	<b>Weryfikacja projektu</b>	<b>11</b>
6.1	Testy manualne . . . . .	11
6.2	Testy integracyjne API . . . . .	12
<b>7</b>	<b>Retrospektywna ocena przebiegu projektu i produktu końcowego</b>	<b>12</b>

## 1. Analiza ryzyka

System wspomagający obsługę praktyk wiąże się z wieloma potencjalnymi problemami w sferach bezpieczeństwa, dostępności oraz wygody korzystania z aplikacji:

- Praca projektowa wykonywana jest przez zespół trzyosobowy który nie ma doświadczenia w wykonywaniu tak dużych systemów.  
Skutki: Przesunięcie końcowego terminu oddania pracy.  
Szansa: Stosunkowo mała, wybrane zostały technologie, które znamy i czujemy się w nich komfortowo.
- Przy tak małym zespole każda osoba musi być odpowiedzialna za wiele modułów systemu, a także za przeprowadzanie analizy funkcjonalności co może powodować wydłużenie czasu wykonania modułów niż w przypadku, kiedy jedna osoba jest odpowiedzialna za jedną konkretną część systemu.  
Skutki: Przesunięcie końcowego terminu oddania pracy.  
Szansa: Nieduża, ale pod warunkiem, że projekt będzie w odpowiedni sposób zarządzany.
- Wiele decyzji dotyczących projektu nie da się w łatwy sposób przewidzieć na początku projektu, dlatego być może będą konieczne modyfikacje założeń w trakcie prac programistycznych co również może powodować opóźnienie.  
Skutki: Przedłużenie czasu wykonania projektu, potencjalnie przeprojektowanie pewnych modułów.  
Szansa: Bardzo duża.
- W związku z nadzwyczajną sytuacją epidemiologiczną istnieje ryzyko, że któryś z członków zespołu zostanie wyłączony z prac na kilka tygodni.  
Skutki: Przedłużenie czasu wykonania projektu.  
Szansa: Nie da się jej określić.
- Niezaliczenie semestru któregoś z członków postawi resztę zespołu w ciężkiej sytuacji.  
Skutki: Przedłużenie czasu wykonania projektu lub rezygnacja z pewnych modułów.  
Szansa: Bardzo mała - jak do tej pory żaden z członków zespołu nie miał problemów z zaliczeniem semestru.
- Niestabilna technologia - może okazać się, że technologia, która została wybrana nie będzie się nadawała do postawionego zadania.  
Skutki: Znaczne wydłużenie czasu wykonania.  
Szansa: Mała.

## 2. Przyjęta metodyka pracy

Prace nad projektem prowadzone były zgodnie z modelem przyrostowym. Proces został podzielony na 5 iteracji. Produktem powstałym w wyniku prac w iteracji była działająca aplikacja.

Pierwsza iteracja stanowiła największą część aplikacji, która ugruntowała ogólną architekturę systemu i nałożyła schemat tworzenia kodu. Kolejne iteracje ulepszały aplikację dodając nowe funkcjonalności i poprawiając błędy które zostały zauważone w poprzedniej.

## 3. Role i podział prac

Projekt był wykonywany przez trzy osoby. Każda osoba z uczestników projektu była odpowiedzialna za wiele elementów systemu i pracowała na wielu stanowiskach. Zadania wykonywane podczas pracy nad projektem umożliwiają wyodrębnienie następujących stanowisk:

- Analityk – kontakt z klientem, określanie wymagań, tworzenie dokumentacji funkcjonalności.
- Programista – implementacja kodu systemu.
- Projektant – projektowanie elementów interfejsu użytkownika.
- Architekt oprogramowania – zaprojektowanie architektury systemu.

### 3.1. Komunikacja

W trakcie prac nad projektem a także na etapie analizy i przygotowywania dokumentacji odbywały się regularne spotkania z klientem z częstotliwością dostosowaną do iteracji. Na etapie analizy rozwiązywane były wątpliwości dotyczące tego jak aplikacja ma działać i w jaki sposób. Na etapie prac programistycznych przedstawiane były kolejne prototypy aplikacji i na tej podstawie tworzone były wnioski jaki element aplikacji należy poprawić lub zmienić a także weryfikowany był poziom wykonania prac z założeń z harmonogramu. Dyskusje prowadzone na spotkaniach pozwoliły na bieżąco weryfikować, czy projekt zmierza w dobrym kierunku i ułatwiły weryfikacje oraz ewentualną korektę założeń.

### 3.2. Wykorzystane narzędzia

Do sprawnej komunikacji w zespole zostało użyte narzędzie MS Teams, które było używane na każdym etapie życia projektu (łącznie do komunikacji z klientem). Do organizacji pracy, zapisywania na bieżąco elementów do wykonania (planów iteracji), zarządzania zadaniami wykorzystana była zintegrowana z narzędziem MS Teams tablica Kanban. Dzięki temu w wygodny sposób było dla każdego widoczne co zostało do zrobienia, co jest w trakcie prac i co jest już wykonane. Do pracy nad kodem stosujemy IDE JetBrains (IntelliJ/Webstorm).

Oprócz tego wykorzystywane było narzędzie github.com jako repozytorium GIT, które upraszcza wiele procesów związanych z wytwarzaniem kodu (takie jak np. Wersjonowanie kodu, inspekcja kodu).

### 3.3. Harmonogram

Prace nad projektem poprzedziło spotkanie organizacyjne na którym zostały ustalone wszystkie szczegóły dotyczące harmonogramu prac. Całość pracy została podzielona wstępnie na 3 etapy:

- **Etap 1** – Podczas tego etapu przygotowane zostały wszystkie elementy merytoryczne, dokumentacje funkcjonalności, założenia tego jak system ma działać i co ma robić, przygotowywane były scenariusze użytkownika oraz przypadki użycia.
- **Etap 2** – Prace programistyczne podzielone na kilka iteracji (sprintów):
  - Sprint 1 - przygotowanie środowiska pracy, konfiguracja repozytoriów, wykonanie pierwszej wersji aplikacji wraz z konfiguracją połączenia z bazą danych, założeniami architektonicznymi, konfiguracją elementów dodatkowych takich jak SwaggerUI, SonarLint.
  - Sprint 2 - Wykonanie pierwszego prototypu aplikacji umożliwiającego wykonanie prostych operacji takich, logowania się, rejestracji, dodawania kont opiekunów, grup studenckich (moduł administratora).
  - Sprint 3 - Wykonanie modułu opiekuna ze strony uczelni, modułu opiekuna ze strony pracodawcy.
  - Sprint 4 - Wykonanie modułu praktykanta, modułu komunikatora, modułu powiadomień.
  - Sprint 5 - Wykonanie modułu generowania certyfikatu, poprawki w istniejących modułach, poprawki interfejsu, poprawki optymalizacyjne.
- **Etap 3** - Weryfikacja wyników projektu, ewentualne korekty w działaniu, wykonanie ostatecznej dokumentacji projektu.

## 4. Przebieg prac

### 4.1. Etap 1

W pierwszym etapie zostały wszystkie elementy merytoryczne, dokumentacje funkcjonalności, założenia tego jak system ma działać i co ma robić, przygotowywane były scenariusze użytkownika oraz przypadki użycia. Zdecydowane zostało, jakie narzędzia będą używane podczas prac oraz zaplanowany został harmonogram oraz wybrana metodologia tworzenia oprogramowania. Systemem kontroli wersji został GIT (github). Rozpatrzone zostały możliwości technologiczne w jakich można wykonać system oraz podjęte zostały decyzje co do architektury systemu.

**Czas trwania:** 6 tyg.

**Początek:** 2020-12-01

## 4.2. Etap 2

### 4.2.1. Sprint 1

#### Cele:

- Przygotowanie środowiska pracy.
- Konfiguracja repozytoriów.
- Konfiguracja SpringBoot.
- Wykonanie pierwszej wersji aplikacji wraz z konfiguracją połączenia z bazą danych.
- Stworzenie architektury systemu.
- Konfiguracja dokumentacji API za pośrednictwem narzędzia SwaggerUI.
- Podział prac na następne Sprints.

#### Wyniki:

- Zostało przygotowane środowisko pracy (skonfigurowana początkowa aplikacja Spring-Boot, ustawienie połączenia z bazą danych MySQL, ustawienia IDE).
- Dodane zostały wymagane zależności Maven.
- Zostały stworzone repozytoria kodu na platformie github (odzierlnie na część Backend i Frontend).
- Została wykonana początkowa architektura systemu (schemat tworzenia kolejnych kontrolerów).
- Została uzupełniona tablica Kanban o kolejne zadania oraz zostały przypisane do odpowiednich członków zespołu.

**Czas trwania:** 3 tyg.

**Początek:** 2021-03-01

### 4.2.2. Sprint 2

#### Cele:

- Wykonanie pierwszego prototypu aplikacji umożliwiającego wykonanie prostych operacji związanych głównie z modułem administratora, takich jak:
  - Logowanie się.
  - Wykonanie mechanizmu ról użytkowników.
  - Zarządzanie wydziałami uczelni przez administratora.
  - Zarządzanie kontami opiekunów przez administratora.
- Wykonanie mechanizmu pozwalającego na wysyłkę wiadomości mailowych.

**Wyniki:**

- Zostały wykonane cele wymienione powyżej. Powstał pierwszy działający prototyp aplikacji frontend, który pozwala na:
  - Logowanie się.
  - Widok po zalogowaniu – twoje konto.
  - Dodawanie wydziału i lista wydziałów.
  - Dodawanie opiekunów z uczelni.
  - Po dodaniu opiekuna przez administratora – wysyłka maila z linkiem do dokończenia rejestracji.

**Czas trwania:** 3 tyg.

**Początek:** 2021-03-22

**4.2.3. Sprint 3****Cele:**

- Umożliwienie rejestracji studenta.
- Umożliwienie zarządzania studentami.
- Umożliwienie zarządzania firmami.
- Umożliwienie zarządzania opiekunami firm.

**Wyniki:**

- Cele powyżej zostały wykonane. Powstały następujące elementy części frontend:
  - Widok rejestracji.
  - Widok dodawania firmy oraz opiekunów firm.
  - Widok firm oraz opiekunów firm pozwalające na podstawowe akcje takie jak: edycja, usuwanie.

**Czas trwania:** 3 tyg.

**Początek:** 2021-04-12

**4.2.4. Sprint 4****Cele:**

- Wykonanie modułu pozwalającego na obsługiwanie praktyk:
  - Dodawanie grup studenckich.
  - Zapisywanie studenta na praktykę.
  - Umożliwienie opiekunom (firmy i uczelni) na ocenę praktyki.
- Wykonanie modułu komunikatora.

- Wykonanie modułu powiadomień.

**Wyniki:**

- Powstał moduł pozwalający na obsługę praktyk, widoki z nim związane, takie jak:
  - Dodawanie grup studenckich.
  - Tworzenie praktyki.
  - Dodawanie danych do praktyki.
  - Ocenianie praktyk - zarówno przez opiekuna uczelni, jak i opiekuna firmy.
- Powstał w pełni działający moduł komunikatora.
- Powstał w pełni działający moduł powiadomień.

**Czas trwania:** 3 tyg.

**Początek:** 2021-05-03

#### 4.2.5. Sprint 5

**Cele:**

- Wykonanie modułu związanego z generowaniem certyfikatów ukończenia praktyk.
- Testy manualne aplikacji.
- Poprawki wizualne.

**Wyniki:**

- Powstał moduł generujący certyfikat w formacie PDF.
- Testy manualne wykazały potrzebę poprawek wizualnych w zakresie smartfonów.
- Zostały wprowadzone poprawki wizualne w zakresie smartfonów.

**Czas trwania:** 3 tyg.

**Początek:** 2021-05-24

#### 4.3. Etap 3

Celem etapu trzeciego jest weryfikacja wyników projektu, tj. czy były one zgodne ze wstępnymi założeniami i czy nie należy wprowadzić pewnych poprawek z tym związanych. Następnym celem jest utworzenie ostatecznej wersji dokumentacji projektowej.

**Czas trwania:** 3 tyg.

**Początek:** 2021-05-24



## 5. Dziennik projektu

### Marzec:

- 03/21/2021 - Utworzenie repozytoriów - Adrian Nęcza
- 03/25/2021 - Konfiguracja SpringBoot – Igor Dzierwa

### Kwiecień:

- 2021-04-15 - Rozpoczęto i zakończono prace nad logowaniem, rolami użytkowników. Utworzono strukturę projektu – Konrad Makuch
- 2021-04-20 - Wstępna konfiguracja części frontend – Konrad Makuch
- 2021-04-20 – Rozpoczęto prace nad modulem administratora – Konrad Makuch
- 2021-04-23 – Zakończono prace nad modulem administratora: zarządzanie opiekunami, zarządzanie wydziałami – Konrad Makuch
- 2021-04-23 - Rozpoczęcie prac nad systemem powiadomień - Adrian Nęcza, Konrad Makuch
- 2021-04-30 - Rozpoczęcie prac na dokumentacją API (Swagger), organizowanie struktury projektu - Igor Dzierwa

### Maj:

- 2021-05-04 - Zakończenie prac na dokumentacją API (Swagger) - Igor Dzierwa
- 2021-05-04 - Rozpoczęcie prac nad grupami studenckimi, firmami, opiekunami firm, wydziałami – Igor Dzierwa
- 2021-05-10 - Kontynuacja prac nad grupami studenckimi – Igor Dzierwa
- 2021-05-11 - Dodanie listowania firm – Igor Dzierwa
- 2021-05-11 – Przebudowa kodu związanego z generowaniem tokenów walidacyjnych – Adrian Nęcza
- 2021-05-11 - Naprawa wysyłanych danych do części frontend – Igor Dzierwa
- 2021-05-15 - Rozpoczęcie pracy nad modulem studenta: rejestracja – Adrian Nęcza
- 2021-05-15 - Kontynuacja pracy nad modulem studenta i administratora: zarządzanie firmami oraz zarządzanie studentami – Igor Dzierwa
- 2021-05-18 - Utworzono możliwość dodawania opiekuna firmy – Igor Dzierwa
- 2021-05-19 - Kontynuacja prac nad modułami studenta oraz opiekuna uczelni: pobieranie grup studenta oraz walidacja konta opiekuna uczelni – Adrian Nęcza
- 2021-05-20 - Usuwanie opiekunów uczelni – Igor Dzierwa
- 2021-05-24 - Rozpoczęto prace nad zarządzaniem opiekunami firm – Igor Dzierwa

- 2021-05-25 - Zakończono prace nad zarządzaniem opiekunami firm – Igor Dzierwa
- 2021-05-25 - Dodanie możliwości ponownego wysyłania tokenu walidacyjnego – Igor Dzierwa
- 2021-05-25 - Zakończenie prac nad modulem administratora – Igor Dzierwa
- 2021-05-26 - Rozpoczęcie prac nad modulem komunikatora – Igor Dzierwa
- 2021-05-27 - Dodanie możliwości zarządzania grupami studenckimi – Adrian Nęcza

#### **Czerwiec:**

- 2021-06-01 - Dodawanie studentów do grup studenckich – Adrian Nęcza
- 2021-06-01 – Rozpoczęcie i zakończenie prac nad modulem statystyk - Konrad Makuch
- 2021-06-02 - Rozpoczęcie prac nad modulem praktyki – Adrian Nęcza
- 2021-06-02 - Zakończenie praktyk nad modulem komunikatora – Igor Dzierwa
- 2021-06-03 - Umożliwiono tworzenie praktyki oraz dodawanie danych do praktyki – Adrian Nęcza
- 2021-06-04 – Kontynuacja prac nad modulem praktyk – realizacja procedury akceptacji praktyk przez opiekunów – Konrad Makuch
- 2021-06-04 – Rozpoczęcie i zakończenie prac nad modulem generowania certyfikatu ukończenia praktyki w PDF - Konrad Makuch
- 2021-06-04 - Zakończono pracę nad modulem powiadomień mailowych - Konrad Makuch
- 2021-06-06 – Kontynuacja prac nad modulem praktyk - lista praktyk do oceny dla opiekunów – Adrian Nęcza
- 2021-06-06 - Zakończono pracę nad modułami: studenta, opiekuna firmy oraz opiekuna uczelni, oraz ograniczono dostęp do poszczególnych funkcji dla odpowiednich ról w systemie – Adrian Nęcza

## **6. Weryfikacja projektu**

### **6.1. Testy manualne**

Regularnie po zaimplementowaniu każdej z funkcjonalności wykonywano testy manualnie (na przykład po zaimplementowaniu rejestracji studenta sprawdzano poprawność działania wszystkich związanych z nią funkcjonalności). Uzyskane w trakcie testów informacje wykorzystano do poprawiania błędów lub po prostu ulepszania danej funkcjonalności.

Dużą pomocą przy testowaniu stanowiła biblioteka do generowania dokumentacji tworzonego API - Swagger. Pozwala ona na zwizualizowanie zasobów serwera, które dzielone są na kontrolery. Rezultatem użycia jest znaczne przyspieszenie pracy nad serwerem, ułatwienie testów, zaoszczędzenie czasu na korzystaniu z zewnętrznych aplikacji takich jak Postman czy Insomnia.

## 6.2. Testy integracyjne API

Do automatycznych testów integracyjnych poszczególnych modułów wchodzących w skład API, wykorzystaliśmy Javascriptowy framework - Cypress. Jest to narzędzie, które pozwala na testy zarówno po stronie frontendowej, jak również backendowej. Pozwala na wygodne przejście wyników testów po stronie przeglądarki, co jest o wiele wygodniejsze od bazowania na wynikach zwracanych w konsoli. Zostały przetestowane następujące rzeczy:

- Czy odpowiedzi są poprawne z dokumentacją.
- Czy odpowiedzi zawierają odpowiednie kody HTTP w zależności od scenariusza.
- Czy użytkownik z daną rolą ma dostęp tylko do odpowiednich endpointów.

## 7. Retrospektywna ocena przebiegu projektu i produktu końcowego

Rozpoczynając od przebiegu projektu - należy go ocenić pozytywnie. Każdy z członków zespołu miał okazję się wykazać, nie było żadnych problemów przy komunikacji, praktycznie wszystkie funkcjonalności były dostarczane na czas (czasami należało przesunąć jakąś funkcjonalność ze względu na powiązanie z inną). Jeśli któryś z członków zespołu potrzebował pomocy, to zawsze ją otrzymywał od innego członka zespołu.

Jeśli natomiast chodzi o kwestię produktu, to należy uznać, że jest to pełny sukces, oczywiście biorąc pod uwagę skalę projektu oraz brak doświadczenia członków zespołu. Prawie wszystkie funkcjonalności zostały dostarczone w stanie, jaki został założony przy ustalaniu wymagań lub odpowiadającym. Powstała poprawnie działająca aplikacja.

Mimo wszystko należy wziąć pod uwagę, że zespół miał możliwość zmian założeń projektowych. Jeśli pewne elementy sprawiały zbyt dużo problemów lub okazywały się niespójne logicznie, to mogliśmy samodzielnie podjąć decyzję o uproszczeniu danej funkcjonalności lub zmianie jej działania.