

Dokument wymagań projektowych

Nazwa projektu: „Obsługa sprawozdań z zajęć”

Autorzy: Michał Rosiński, Patryk Gałka,
Patryk Łukasiewicz, Adam Świnka

Data: 25.10.2020 r

1. Elementy składowe projektu(produkty projektu)

a) Elementy niefunkcjonalne

I semestr

- Prototypy aplikacji internetowej
- Schematy i specyfikacja odnośnie architektury wykorzystanej w projekcie(wstępne plany i schematy)
- Repozytorium Git zawierające kod aplikacji
- Tablica projektowa Jira

II semestr

- Publikacja aplikacji w domenie internetowej 'amu.edu.pl'
- Finalne produkty aplikacji internetowej
- Schematy i specyfikacja odnośnie architektury wykorzystanej w projekcie (finalna wersja i schemat)
- Repozytorium Git, API
- Tablica projektowa – Jira

b) Elementy funkcjonalne

I semestr

- Aplikacja internetowa zintegrowana z REST API
- Aplikacja serwerowa w języku Python oparta o framework Django w niepełnej funkcjonalności
- Baza danych oparta o silnik MySQL wbudowana w framework Django

II semestr

- Aplikacja serwerowa w języku Python oparta o Framework Django w pełnej funkcjonalności
- Instancja serwera bazy danych oparta o silnik MySQL

2. Granice projektu

a) Produkty zawarte:

- Aplikacja internetowa

- Publikacja aplikacji w domenie amu.edu.pl
- Aplikacja serwerowa – REST API
- Relacyjna baza danych oparta o silnik MySQL
- Platforma LDAP
- Prototypy aplikacji internetowej
- Dokumentacja dotycząca oprogramowania REST API
- Dokumentacja, schematy i specyfikacja odnośnie architektury wykorzystanej w projekcie
- Repozytorium Git zawierające kod aplikacji, API
- Tablica projektowa – Jira

b) Produkty niezawarte

- Aplikacja mobilna(w tym przypadku byłaby mało praktyczna biorąc pod uwagę np. wstawianie plików o rozszerzeniu .pml)
- Serwis internetowy USOS(brak możliwości skontaktowania się z Centrum Informatycznym UAM. W zamian za to system będzie zintegrowany z platformą LDAP)
- Aplikacja Ms Teams i platforma Moodle(występują problemy związane z zintegrowaniem aplikacji z systemem oraz połączeniem systemu z platformą)

c) Funkcjonalności niezawarte

- Możliwość oddawania zadań po terminie(klient nie życzy sobie aby oddawać zadania po terminie)
- Wykorzystywanie sieci Petriego do analizy danych
- Korzystanie z systemu przez studentów spoza wydziału(dla innych wydziałów aplikacja nie będzie potrzebna)
- Wybieranie w aplikacji uczelni do której system mógłby mieć dostęp(po co, gdy tylko studenci WMI będą korzystać z systemu)

3. Lista wymagań funkcjonalnych

- a) Sprawdzanie poprawności wykonanych zadań, takich jak sprawozdania w formacie .txt z metryczką w formacie .xml oraz programów napisanych w języku Promela
- b) Sprawdzanie poprawności formy metryczki w formacie .xml zawartej w zadaniach starego typu(sprawozdań)
- c) Wprowadzanie wyrażenia logiki temporalnej pod kodem źródłowym programu oddanego przez studenta
- d) Wstawianie kodu źródłowego Promeli do specjalnego formularza
- e) Komunikacja, wymiana informacji i pomysłów na forum internetowym
- f) Udostępnianie nowych zadań, plików, sprawozdań dla studentów
- g) Przyjmowanie rozwiązanych zadań od studentów

- h) Kontrolowanie prac nadesłanych przez studentów i wysyłanie informacji prowadzącemu o podejrzeniu plagiatu
- i) Aktualizowanie listy wszystkich studentów, wraz ze statystyką liczby punktów przez nich zdobytych
- j) Przekierowanie prowadzącego poprzez link do rozwiązania zadania oddanego przez studenta

4. Lista wymagań niefunkcjonalnych

- a) Schematy i specyfikacja odnośnie architektury wykorzystanej w projekcie
- b) Tablica projektowa – Jira
- c) Prototypy aplikacji internetowej
- d) Prototyp bazy danych
- e) Publikacja aplikacji w domenie internetowej 'amu.edu.pl'
- f) Prototyp forum internetowego
- g) Prototyp formularza
- h) Wielkość i rozszerzenie przesyłanego pliku

5. Mierzalne wskaźniki wdrożeniowe

- a) System zostanie udostępniony w domenie internetowej amu.edu.pl i będą z niego korzystać osoby zapisane na zajęcia 'Algorytmy rozproszone'
- b) System zostanie przetestowany co najmniej 50 rozwiązanymi zadaniami, zarówno dotyczącymi sprawozdań, jak i programów zaimplementowanych w języku Promela, aby sprawdzić działanie modułu składającego się z

- systemu sprawdzającego poprawność rozwiązanych zadań i systemu antyplagiatowego
- c) System zostanie poddany testom obciążeniowym, aby zbadać szybkość odpowiedzi serwera w przypadku gdy z systemu będzie korzystać określona liczba użytkowników
 - d) Na koniec pierwszego semestru system zostanie dostarczony do klienta w wersji testowej alpha (rozumianej zgodnie z metodyką Agile)
 - e) Na koniec drugiego semestru klient dostanie system w wersji testowej beta
 - f) Produkt zostanie zaakceptowany ustnie przez klienta

6. Kryteria akceptacji projektu dla I semestru prac

- Ukończenie i wdrożenie podstawowych funkcjonalności do aplikacji internetowej
- Przetestowanie zaimplementowanych funkcjonalności pod kątem użytkowym przez klienta
- Wykonanie testów jednostkowych i sprawdzenie czy spełniają swoje zadanie
- Wdrożenie prostego systemu antyplagiatowego
- Wykonanie testów dla sprawdzenia poprawności działania systemu antyplagiatowego

- Dostarczenie prototypu aplikacji internetowej w wersji testowej alfa
- Spełnienie częściowych wymagań klienta
- Akceptacja wersji alfa produktu
- Dostarczenie wstępnych prototypów składowych produktu
- Planowana integracja systemu z platformą LDAP
- Planowane wdrożenie systemu sprawdzającego rozwiązania zadań

7. Kryteria akceptacji projektu dla II semestru prac

- Ukończenie wszystkich funkcjonalności
- Wdrożenie systemu sprawdzającego poprawność wykonanych sprawozdań oraz wyników działania programów w języku Promela
- Dostarczenie i zaakceptowanie wersji beta aplikacji internetowej
- Dostarczenie wszystkich prototypów składowych produktów
- Pozytywnie zakończone testy działania systemu
- Integracja z uczelnianym systemem LDAP
- Wdrożenie aplikacji na serwerze wskazanym przez klienta
- Brak błędów utrudniających korzystanie z aplikacji

- Wdrożenie aplikacji internetowej dla użytkownika końcowego w domenie internetowej 'amu.edu.pl'
- Produkt końcowy oddany w czasie

8. Organizacja pracy zespołu

- Strona klienta:
 - a) Dr Michał Hanćkowiak:
 - Product Owner
 - Komunikacja Product Owner z zespołem wykonawczym
 - Promotor zespołu
- Strona zespołu projektowego:
 - a) Michał Rosiński
 - Utworzenie interfejsu użytkownika
 - Utworzenie front-endu aplikacji wraz z mechanizmem logowania i rejestracji na stronie
 - Wylistowanie studentów wraz ze statystyką łącznej liczby punktów na stronie prowadzącego oraz informacji o plagiacie
 - Zaimplementowanie możliwości przekierowania prowadzącego do rozwiązanego zadania poprzez kliknięcie w link
 - Stworzenie bazy danych na potrzeby aplikacji API
 - b) Patryk Gałka
 - Implementacja aplikacji serwerowej API
 - Projektowanie architektury aplikacji

- Wdrożenie mechanizmu wysyłania i wstawiania plików na serwerze
- Zaimplementowanie mechanizmu pobierania zadań ze strony prowadzącego
- Zaimplementowanie systemu sprawdzającego poprawność nadesłanych rozwiązań zadań

c) Patryk Łukasiewicz

- Implementacja aplikacji serwerowej API
- Projektowanie architektury aplikacji
- Projektowanie prototypu bazy danych
- Utworzenie forum internetowego dla aplikacji
- Wdrożenie mechanizmu sprawdzającego poprawność i formę metryczki w formacie .xml w sprawozdaniach

d) Adam Świnka

- Reprezentatywna rola i zarządzanie komunikacją z klientem
 - Pełnienie funkcji lidera zespołu
 - Projektowanie architektury aplikacji
 - Projektowanie prototypu aplikacji internetowej
 - Implementacja aplikacji serwerowej API
 - Zarządzanie dokumentacją
 - Implementacja systemu antyplagiatowego
- Przyjęta metodyka to Scrum. Została ona zaakceptowana przez wszystkich członków zespołu. Odrzucona została metodyka RUP ze względu na zbyt małą liczbę osób w grupie

- Do wspomagania prac projektowych wykorzystywany jest system JIRA. Kod źródłowy jest zarządzany za pomocą repozytorium Git
- Zadanie do wykonania w określonym przyroście miało swój termin wykonania przez okres dwóch tygodni i zwykle dwa dni przed terminem wykonania zostało skończone

9. Ryzyka technologiczne

- Ryzyka ze względu na zasoby:
 - a) Odejście członka implementującego stronę serwerową lub zrezygnowanie z wykonywania przez niego tej czynności – trudność w ukończeniu serwera
 - b) Określone ramy czasowe – dwa semestry na wykonanie działającego produktu
 - c) Ograniczona znajomość technologii zastosowanych przy budowie projektu, członkowie zespołu za mało czasu poświęcili na stosowanie tych technologii w praktyce, niemożliwe do przewidzenia problemy oraz uwidaczniające się przeszkody podczas pracy nad projektem
 - d) Odejście członka implementującego określone funkcjonalności – trudność w całkowitym ukończeniu architektury systemu
- Inne:
 - a) Liczne nieporozumienia na linii zespół-klient, związane z implementacją określonych funkcjonalności, wyglądem końcowym poszczególnych stron internetowych czy lokalizacją pól do wprowadzania tekstu, niektórych wyrażień, kodu programów itp.

- b) Nagła zmiana wymagań klienta, która wiąże się z ryzykiem nieukończenia w czasie niektórych elementów zawartych w architekturze projektu
- c) Planowana jest implementacja antyplagiatu metodą porównywania ze sobą takich samych i podobnych łańcuchów znaków. Jeśli okaże się, że jest zbyt wolna lub w trakcie testów pojawią się błędy, zostanie zastosowana szybsza i lepsza metoda
- d) Zaproponowana integracja systemu z aplikacją Microsoft Teams oraz platformą Moodle może okazać się niemożliwa do wykonania gdyż mogą wystąpić problemy z działaniem aplikacji albo korelacja systemu z zewnętrznymi aplikacjami może przynosić niewłaściwe rezultaty.
- e) Dodanie nowych wymagań przez klienta – możliwe wydłużenie czasu pracy związane z rekonstrukcją kodu, niepoprawna implementacja niektórych wymagań funkcjonalnych, brak możliwości dodania nowych wymagań
- f) Nieporozumienia w zespole dotyczące implementacji funkcjonalności, architektury w określonym czasie
- g) Napięty termin oraz duża ilość obowiązków związanych z uczelnią i sprawy prywatne mogą negatywnie wpłynąć na zrealizowanie aplikacji
- h) Brak odpowiedniego zaangażowania w projekt poszczególnych członków zespołu oraz klienta
- i) Konflikty i brak właściwej organizacji w zespole

10. Kamienie milowe

- I faza, I semestr (02.2020 – 09.2020):
 - a) Przygotowanie prototypu aplikacji

- b) Stworzenie backlogu dla projektu „Obsługa sprawozdań z zajęć” w systemie JIRA – opracowanie funkcjonalności oraz user stories
- c) Rozpoczęcie prac programistycznych nad projektem
- d) Testowanie aplikacji przez zespół
- e) Ukończenie MVP aplikacji
- f) Wdrożenie MVP aplikacji na serwer lokalny
- g) Poddanie MVP testom funkcjonalnym
- II faza, II semestr(10.2020 – 01.2021):
 - a) Reorganizacja i zdefiniowanie backlogu
 - b) Rozpoczęcie prac programistycznych nad projektem
 - c) Testowanie aplikacji przez zespół
 - d) Ukończenie aplikacji wraz ze wszystkimi wcześniej założonymi funkcjonalnościami
 - e) Wdrożenie aplikacji w domenę internetowej „amu.edu.pl” dla użytkowników końcowych
 - f) Przekazanie produktu klientowi