

Dokument wymagań projektowych

Nazwa projektu: *Pomocnik
Profesora*

Autor:
Rozpłochowski Paweł

Wersja dokumentu:
1.0.0

Data:
05.11.2020

1. Elementy składowe projektu (produkty projektu)

W ramach projektu inżynierskiego powstała aplikacja webowa w języku Python oparta o framework Flask. Na frontendzie użyłem technologii takich jak JS, CSS, Bootstrap.

Moim celem jest stworzenie narzędzia, które ułatwi studentom zapisywanie się na dyżury do prowadzących. Aplikacja ma służyć zarówno uczącym się jak i pracownikom uniwersytetu. System udostępniony zostanie zarówno w domenie internetowej, jak i zintegrowany z systemem grupy "Wykłady Webowe". Docelowo korzystać z niego będą dziesiątki użytkowników, a w przyszłości może i zdecydowana większość na wydziale WMI. Do naszego oprogramowania będzie miał dostęp każdy student posiadający konto na platformie USOS.

2. Granice projektu

Aplikacja byłaby funkcjonalniejsza gdyby wysyłała powiadomienie do smartfonów, na które użytkownicy mogliby szybciej zareagować i dać odpowiedź drugiej stronie praktycznie od razu.

3. Lista wymagań funkcjonalnych

- Akceptowanie dyżurów przez prowadzącego
- Odmowa dyżurów przez prowadzącego
- Możliwość zapisania się na dyżur do wybranych zarejestrowanych prowadzących
- Rejestracja w zależności czy jest to student czy prowadzący
- Możliwość logowania
- Student widzi na jakie dyżury się zapisał
- Prowadzący widzi kto się do Niego zapisał
- Wejście na profil prowadzącego
- Spersonalizowane dane prowadzącego(modyfikowanie)
- Autoryzowanie dwupoziomowe użytkowników
- Wiadomości Prowadzący-Student
- Lista prowadzących z USOS
- Zmiana danych autoryzujących
- Powiadomienia na email przy zapisaniu się
-

4. Lista wymagań niefunkcjonalnych

- Przejrzysty dashboard Studenta oraz Prowadzącego

5. Mierzalne wskaźniki wdrożeniowe

System zostanie wdrożony w domenę internetową i będzie z niego korzystać około 8 użytkowników, zostanie zasilony pozyskanymi danymi między innymi

bazą użytkowników z USOS. Pod koniec drugiego semestru system zostanie dostarczony do klienta.

6. Kryteria akceptacji projektu dla I semestru prac

Kryteria wymagane:

- Możliwość zapisania się na dyżur
- Możliwość akceptacji dyżuru
- Strona wprowadzona na serwer

Kryteria oczekiwane:

- Funkcjonalności takie jak studio do zmiany kart pracy, statystyki pracowników, generowanie raportów.

7. Kryteria akceptacji projektu dla II semestru prac

Kryteria wymagane:

- Funkcjonalności wymienione w liście wymagań funkcjonalnych
- Wdrożenie aplikacji na serwer
- Dokumentacja dotycząca działania aplikacji

Kryteria oczekiwane:

- Powiadomienia na smartphona

8. Organizacja projektu

Rozpłochowski Paweł

- Reprezentatywna rola przed klientem
- Implementacja aplikacji webowej
- Zarządzanie dokumentacją
- Opracowanie prototypu projektu
- Stworzenie oprawy graficznej aplikacji webowej

- Projektowanie architektury programu

9. Ryzyka technologiczne

- Ryzyka ze względu na zasoby:
 - Odejście któregoś z członków grupy
 - Określone ramy czasowe (semestry) na stworzenie produktu
- Inne:
 - Nieporozumienia na linii zespół a promotorem wynikające z nieznamomości pojęć domenowych lub nieporozumienia dotyczące funkcjonalności czy innych rzeczy
 - Dodanie nowych wymagań przez promotora - ryzyko nieukończenia w czasie lub nie poprawna implementacja nowych wymagań
 - Nieporozumienia w zespole dotyczące rozwiązań implementacyjnych, architektury, funkcjonalności lub używanych technologii
 - Brak odpowiedniego zaangażowania w projekt ze strony zespołu, wybranych członków zespołu jak i klienta (nie dostarczanie informacji na czas, etc.)
 - Nagły brak wsparcia dla lub przerwanie rozwijania wykorzystywanej technologii lub narzędzia - konieczność implementacji od nowa, lub kontynuowanie implementacji w nieużywanym/źle zaprojektowanym narzędziu, bibliotece
 - Napięty terminu, nakład obowiązków związanych z uczelnią oraz sprawy prywatne, mogą negatywnie wpłynąć na pracę nad aplikacją
 - Trudności w organizacji pracy spowodowane pandemią

10. Kamienie milowe do osiągnięcia

- I faza, I semestr (02.2020 - 10.2020):
 - Sporządzenie prototypu aplikacji
 - Opracowanie funkcjonalności programu
 - Rozpoczęcie prac programistycznych nad projektem
 - Przygotowanie uproszczonej wersji aplikacji webowej

- Migracja aplikacji na serwer
- Ukończenie MVP projektu
- Wykonanie testów
- ...
- II faza, II semestr (10.2020 - 03.2020):
 - Ukończenie aplikacji wraz z ze wszystkimi zdefiniowanymi funkcjonalnościami
 - Zdanie produktu klientowi