

# Studium wykonalności

Bank prac domowych

Drużyna Pierścienia

Autorzy: Krzysztof Chołys, Natalia Stępień

Wersja 1.3

<b>Nr wersji</b>	<b>Data</b>	<b>Opis zmiany</b>	<b>Autor</b>
1.0	25.03.2022	Stworzenie dokumentu, stworzenie punktów 1.-2.	Krzysztof Chołys
1.1	25.03.2022	Stworzenie punktów 4., 5., 7.	Natalia Stępień
1.2	27.03.2022	Stworzenie punktów 6.	Natalia Stępień
1.3	27.03.2022	Stworzenie punktów: 3., 8. Uzupełnienie punktów: 1., 2., 3., 4., 6. Zredagowanie i sformatowanie tekstu.	Krzysztof Chołys

## Spis treści

Studium wykonalności .....	1
1. Założenia realizacji studium .....	4
2. Bieżąca analiza – Opis stanu istniejącego .....	5
3. Wymagania dla systemu .....	6
4. Realizacja przedsięwzięcia – warianty.....	7
5. Ocena wykonalności wariantów i wybór wariantu optymalnego.....	8
6. Plan realizacji przedsięwzięcia .....	8
7. Wstępna analiza ryzyka .....	9
8. Podsumowanie .....	9

## 1. Założenia realizacji studium

### 1.1 Zlecający

Imię i nazwisko: Doktor Paweł Dąbrowski

Firma: Uniwersytet Łódzki

Województwo: łódzkie

Miejscowość: Łódź

Adres firmy: Pomorska 149/153

Przedmiot: Inżynieria oprogramowania

### 1.2 Podstawa wykonania studium

Podstawą wykonania studium są informacje od zlecającego na temat jakiego rodzaju oprogramowania oczekuje od zespołu. Ponadto dodatkowym źródłem informacji jest plik *Tematy\_grupowe.docx* znajdujący się w zespole *1500-Inz\_oprogram\_gr2\_2022* w materiałach zespołu.

### 1.3 Temat studium

Tematem studium jest aplikacja przeznaczona do tworzenia klas, dodawania zadań z terminami końcowymi, przesyłania, wyświetlania podglądów plików i wyświetlania listy członków danej klasy

### 1.4 Cel studium wykonalności

Celem studium jest szczegółowa analiza i ocena wykonalności projektu zadanego przez zlecającego.

### 1.5 Ograniczenia

- Czas - 12 tygodni
- Liczba członków zespołu – 5
- Wykorzystywanie technologii webowych – mniejsza responsywność jak przy wykorzystywaniu technologii operujących na językach niskopoziomowych
- Umiejętności zespołu programistycznego
- Wymagania projektu

## 2. Bieżąca analiza – Opis stanu istniejącego

### 2.1 Istniejące systemy, użytkownicy, przetwarzane dane

Aktualnie pracownicy sfery edukacji posługują się głównie system assignmentów z aplikacji MS Teams. Jest to aplikacja płatna, wysoce zasobochłonna, z licznymi błędami, niską responsywnością, nieintuicyjnym interfejsem i złym UX-em. Również wykorzystywane poczty elektroniczne taka jak MS Outlook czy UsosWEB charakteryzują się nieoptymalnym systemem przesyłania zadań. Wciąż niekiedy wykorzystywana platforma edukacyjna Uniwersytetu Łódzkiego Moodle jest porażką na wielu frontach: jest nieintuicyjna, interfejsy są przebudźcowane i rozpraszaające, posiada katastrofalny UX, jest nieprzyjemna dla oka i nie spełnia dzisiejszych kryteriów wymagań w ramach funkcjonalności tego rodzaju aplikacji.

### 2.2 Podobne systemy dostępne na rynku

Systemy, które działają na zasadzie podobnej do naszej aplikacji to: MS Teams, UŁ Moodle, Wikamp, UsosWEB, Google Classroom, Canas LMS, Schoology

### 2.3 Problem i motywacja do realizacji nowego systemu

Chęć zastąpienia aktualnie wykorzystywanych technologii bardziej wydajnymi, szybszymi, bardziej intuicyjnymi i rozbudowanymi, bardziej dostosowanymi do potrzeb. Chcemy uzupełnić luki, które dostrzegamy z naszej studenckiej perspektywy. Motywuje nas chęć utworzenia dedykowanego rozwiązania stricte pod oddawanie, analizowanie, komentowanie i poprawianie prac domowych.

### 3. Wymagania dla systemu

#### 3.1 Funkcjonalne

- Rejestracja
- Wybór rodzaju konta: uczeń lub nauczyciel
- Logowanie
- Tworzenie klas
- Generowanie unikalnego kodu dołączenia do klasy
- Dołączanie do klasy za pomocą kodu
- Tworzenie zadań
- Wyświetlanie listy klas
- Wyświetlanie zadań ucznia
- Dodawanie terminów końcowych do przesyłania rozwiązań zadań
- Dołączanie plików do zadań przy ich tworzeniu
- Dołączanie plików do rozwiązań zadań
- Wyświetlanie godziny odesłania rozwiązania zadania przez ucznia
- Wyświetlanie podglądu przesłanych plików bezpośrednio w aplikacji
- Pobieranie przesłanych plików w aplikacji
- Zatwierdzanie i odrzucanie przesłanych rozwiązań zadań
- Przesyłanie komentarza do rozwiązań zadań
- Zablokowanie możliwości usuwania przesłanego rozwiązania zadania przez określony czas
- Wyświetlanie stanu przesłań rozwiązań zadań w terminie (zaliczona/niezaliczona/nieprzesłana)

#### 3.2 Niefunkcjonalne

Aplikacja będzie spełniać wymogi bezpieczeństwa w zakresie dostępu użytkowników do zasobów systemu, poprzez zastosowanie mechanizmów uwierzytelniania użytkownika. Ponadto aplikacja będzie dostępna jedynie w języku polskim.

#### 3.3 Jakościowe

- Użyteczność – Aplikacja rozwiąże szereg problemów występujących w jej rynkowych odpowiednikach i pojawiających się przy niestosowaniu żadnej podobnej aplikacji
- Wydajność – Aplikacja będzie działać w czasie rzeczywistym. Ponadto cechuje się wysokim stosunkiem zalet do pracy włożonej w jej obsługę.
- Intuicyjność – Idea aplikacji jak i jej interfejsy nie wymagają poświęcenia dużej ilości czasu na zapoznanie się, by móc korzystać z pełni jej funkcjonalności
- Prostota – Aplikacja specjalizuje się w wąskim zakresie zadań, dzięki czemu jej użytkowanie jest nieskomplikowane i oczywiste

### **3.4 Architektura**

Aplikacja będzie działała na modelu klient-serwer, gdzie połączenie będzie odbywało się na bazie sieci internetowej. Łączenie z bazą danych będzie leżało po stronie serwera.

### **3.5 Wymagania technologiczno-techniczne**

Komputer z dostępem do internetu z systemem Windows 7 lub nowszym, MacOS 10.12 lub nowszy, Linux, telefon z systemem Android 9.0 lub iOS 8 dostępem do internetu.

Przeglądarki: Chrome, Firefox, Safari, Opera, Brave

## **4. Realizacja przedsięwzięcia – warianty**

### **4.1 Wariant 1.**

Aplikacja desktopowa – Aplikacja będzie dostępna na komputery z systemem Windows. Aplikacja będzie posiadać różne zestawy funkcjonalności dla odpowiednich rodzajów użytkowników: nauczycieli i uczniów. Jako frontend wykorzystany zostanie React. Poza tym do frontendu i backendu wykorzystany zostanie framework Electron.JS.

### **4.2 Wariant 2.**

Aplikacja webowa - Aplikacja będzie dostępna w przeglądarkach: Mozilla Firefox, Google Chrome, Opera, Safari, Brave, MS Edge. Aplikacja będzie posiadać różne zestawy funkcjonalności dla odpowiednich rodzajów użytkowników: nauczycieli i uczniów. Frontend będzie opierał się na EJS - Embedded JavaScript templates. Backend z kolei na NodeJS.

### **4.3 Wariant 3.**

Aplikacja webowa - Aplikacja będzie dostępna w przeglądarkach: Mozilla Firefox, Google Chrome, Opera, Safari, Brave, MS Edge. Aplikacja będzie posiadać różne zestawy funkcjonalności dla odpowiednich rodzajów użytkowników: nauczycieli i uczniów. Frontend będzie opierał się na React'cie, backend na Express'ie i NodeJS'ie. Wykorzystany zostanie system baz danych MongoDB.

## 5. Ocena wykonalności wariantów i wybór wariantu optymalnego

### 5.1 Porównanie wad i zalet, kosztów i korzyści

Niektóre z zaprezentowanych przez nas wariantów obejmują technologie, które wymagają dodatkowego zapoznania się z nimi zespołu programistycznego. Mając również na uwadze czynniki takie jak: kompatybilność, wieloplatformowość oraz development, najbardziej korzystny jest wybór 3. wariantu.

Pozostałe warianty posiadają wiele ograniczeń pod względem ciągłego dostępu do Internetu czy braku zewnętrznej przeglądarki. Również kluczowa jest kwestia wydajności. Pod tym względem zdecydowaną przewagę wykazuje aplikacja desktopowa.

### 5.2 Wybór najkorzystniejszego wariantu

Wybrany został wariant 3.

## 6. Plan realizacji przedsięwzięcia

### 6.1 Strategia

Po zapoznaniu się z tematem projektu wybranym przez członków zespołu, przystępujemy do implementacji modelu kaskadowego. W fazie planowania ustalimy wspólne środki komunikacji i współpracy, podzielimy zespół na część odpowiedzialną za programowanie i dokumentację, i zaczniemy zapoznawać się z wymaganymi technologiami. Ponadto napiszemy raport nt. Studium wykonalności projektu. Następnym krokiem będzie przejście do analizy – fazy, w której określimy wymagania dotyczące projektu. Ciągła analiza będzie dynamicznie określała wymagania produktu, które będą z kolei wymagały ponownej analizy, co tworzy swojego rodzaju pętlę. W następującym etapie projektowania skupimy się na tworzeniu schematów user flow i wireframe'ów aplikacji. W fazie implementacji rozpoczniemy fizyczną budowę produktu – powstaną pierwsze prototypy stron, podstron, funkcjonalności i poszczególne systemy. Przejdą one następnie do etapu testowania, gdzie będą po pomyślnej weryfikacji zostaną wdrożone i będą stale utrzymywane.

### 6.2 Wstępny harmonogram

- Określenie wymagań: marzec
- Analiza: kwiecień
- Projektowanie: kwiecień
- Implementacja: maj
- Testowanie: maj
- Konserwacja: koniec maja



## 7. Wstępna analiza ryzyka

### 7.1 Wstępna ocena ryzyka

Jednym z głównych zagrożeń dla projektu jest niedoszacowanie czasu realizacji poszczególnych etapów. Aby temu zapobiec, podzieliliśmy nasz zespół na 2 pomniejsze grupy odpowiedzialne za programowanie i pisanie dokumentacji.

Mając na uwadze krótki czas na wykonanie projektu, niezwłocznie ustaliliśmy narzędzia komunikacji, współpracy i organizacji pracy. Wdrożenie tego typu rozwiązań zdecydowanie usprawni realizację poszczególnych zadań, a w konsekwencji – faz projektów. Konsekwentnie określamy przez nas priorytet poszczególnych zadań zapobiega ewentualnym odstępstwom czasowym.

Ze względu na występowanie osób ze słabym zapleczem programistycznym zdecydowaliśmy się na zwiększenie naszego nakładu pracy. Poznanie nowych technologii przez większą liczbę członków umożliwi nam wzrost tempa pracy.

Uwzględniliśmy również scenariusz zakładający nieobecność jednego bądź większej liczby członków naszego zespołu. Przy ograniczonej liczbie osób zdecydowaliśmy się na dynamiczny podział obowiązków niezależny od wewnętrznego podziału na podgrupy.

### 7.2 Ryzyka zaniechania realizacji

Projekt oraz zespół nastawiony jest na następujące ryzyka:

- Określenie celów niemożliwych do osiągnięcia
- Nieokreślone oczekiwania zleceniodawcy
- Nadmierna drobiazgowość projektu
- Nieodpowiednio oszacowany czas wykonania projektu
- Nieodpowiednie decyzje podejmowane przez liderów grupy

Wszystkie wspomniane czynniki mogą przyczynić się do wystąpienia wewnętrznych konfliktów w zespole. W najgorszym możliwym przypadku doprowadzi to do nagłego zaprzestania pracy nad projektem bądź rezygnacją ze studiów.

## 8. Podsumowanie

Podsumowując wszystkie dostępne możliwości realizacji projektu, dostępne narzędzia, ograniczenia, wymagania dotyczące systemu, wady i zalety wariantów rozwiązań, możemy wysnuć wniosek, że jesteśmy w stanie wykonać zleczone nam zadanie. Dysponując obecnymi zasobami w postaci wiedzy, ilości osób, narzędzi i czasu projekt powinien zostać dobrze zrealizowany.