Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **Komunikator Tekstowy**

Autorzy: **Krzysztof Kubiś, Przemysław Kowal**

Grupa: I1-213A

Kierunek: informatyka

Rok akademicki: 2019/2020

Poziom i semestr: I/4

Tryb studiów: stacjonarne

Spis treści

[2 Odnośniki do innych źródeł 4](#_Toc1976793)

[3 Słownik pojęć 5](#_Toc1976794)

[4 Wprowadzenie 6](#_Toc1976795)

[4.1 Cel dokumentacji 6](#_Toc1976796)

[4.2 Przeznaczenie dokumentacji 6](#_Toc1976797)

[4.3 Opis organizacji lub analiza rynku 6](#_Toc1976798)

[4.4 Analiza SWOT organizacji 6](#_Toc1976799)

[5 Specyfikacja wymagań 7](#_Toc1976800)

[5.1 Charakterystyka ogólna 7](#_Toc1976801)

[5.2 Wymagania funkcjonalne 7](#_Toc1976802)

[5.3 Wymagania niefunkcjonalne 8](#_Toc1976803)

[6 Zarządzanie projektem 9](#_Toc1976804)

[6.1 Zasoby ludzkie 9](#_Toc1976805)

[6.2 Harmonogram prac 9](#_Toc1976806)

[6.3 Etapy/kamienie milowe projektu 9](#_Toc1976807)

[7 Zarządzanie ryzykiem 10](#_Toc1976808)

[7.1 Lista czynników ryzyka 10](#_Toc1976809)

[7.2 Ocena ryzyka 10](#_Toc1976810)

[7.3 Plan reakcji na ryzyko 10](#_Toc1976811)

[8 Zarządzanie jakością 11](#_Toc1976812)

[8.1 Scenariusze i przypadki testowe 11](#_Toc1976813)

[9 Projekt techniczny 12](#_Toc1976814)

[9.1 Opis architektury systemu 12](#_Toc1976815)

[9.2 Technologie implementacji systemu 12](#_Toc1976816)

[9.3 Diagramy UML 12](#_Toc1976817)

[9.4 Charakterystyka zastosowanych wzorców projektowych 12](#_Toc1976818)

[9.5 Projekt bazy danych 12](#_Toc1976819)

[9.6 Projekt interfejsu użytkownika 12](#_Toc1976820)

[9.7 Procedura wdrożenia 13](#_Toc1976821)

[10 Dokumentacja dla użytkownika 14](#_Toc1976822)

[11 Podsumowanie 15](#_Toc1976823)

[11.1 Szczegółowe nakłady projektowe członków zespołu 15](#_Toc1976824)

[12 Inne informacje 16](#_Toc1976825)

# Odnośniki do innych źródeł

* <https://github.com/asasinek123555/Project-IO>

# Słownik pojęć

* **Użytkownik** - to osoba posiadająca konto na serwerze i zalogowana do niego poprzez wpisanie odpowiednich danych logowania,
* **Pokój** - to jednoznacznie oznaczona nazwą lista użytkowników do których jest przesyłana wiadomość wysłana do tego pokoju,
* **Grupy** - to zbiory użytkowników o wspólnych uprawnieniach w pokoju,
* **Moderatorzy** - to osoba posiadająca większe uprawnienia niż pozostali użytkownicy w pokoju,
* **Administrator** - to osoba która posiada wszystkie uprawnienia do wszystkich pokoi,
* **Chat** - to zbiór wiadomości wysłanych w konkretnym pokoju.

# Wprowadzenie

## Cel dokumentacji

Celem dokumentacji jest przedstawienie specyfikacji dla czatu tekstowego zawierającego:

- Definicje,

- Cele,

- Zawartość,

- Główną funkcjonalność.

## Przeznaczenie dokumentacji

Dokumentacja jest przeznaczona celem przedstawienia jej:

1) Zaangażowanym w stworzenie i rozwój projektu:

- Menadżerom projektu,

- Programistom, którzy zaimplementują jej funkcjonalne jak i niefunkcjonalne wymagania,

- Testerom, którzy będą musieli zweryfikować spełnienie wszystkich wymagań.

2) Użytkownikom, którzy wykorzystają je:

- Do prywatnych celów celem zwiększonej integracji,

- Do łatwiejszego porozumiewania się z jednostkami administracyjnymi uczelni,

- Do łatwiejszej komunikacji z prowadzącymi poszczególne kursy w ramach nauczania.

## Opis organizacji

Organizacją jest Zachodniopomorski Uniwersytet Technologiczny w Szczecinie dla której przygotowywany jest system czatu tekstowego. Organizacja ta wykorzysta wdrożony system do systemu komunikacji pomiędzy studentami oraz na linii student-uczelnia. Na uczelni aktualnie uczy się 6000 studentów oraz 200 dydaktyków. Po wdrożeniu systemu uczelnia ta otrzyma dodatkowy sposób szybkiej komunikacji w przypadku sytuacji pracy zdalnej, który nie będzie uzależniony od zewnętrznych firm.

## Analiza SWOT organizacji

|  |  |  |  |
| --- | --- | --- | --- |
| Silne strony | [%] | Słabe strony | [%] |
| Cyfryzacja placówki | 13 | Stały dostęp do Internetu | 12 |
| Alternatywny sposób komunikacji | 26 | Zapotrzebowanie na większą ilość dedykowanych serwerów | 20 |
| Razem: 39 | | Razem: 32 | |
|  |  |  |  |
| Szanse | [%] | Zagrożenia | [%] |
| Możliwość zwiększonej integracji pośród studentów | 20 | Awaryjność serwerów | 19 |
| Możliwość zwiększenia prestiżu poprzez unowocześnienie uczelni | 40 | Niewydolność filtrowania niewłaściwych treści | 20 |
| Razem: 60 | | Razem: 39 | |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

Aplikacja pozwalająca na przesyłaniu wiadomości tekstowych do wszystkich użytkowników w danym pokoju.

### Podstawowe założenia

1. Aplikacja ma służyć do przesyłania wiadomości tekstowych.

### Cel biznesowy

Celem tworzenia aplikacji jest ułatwienie komunikacji na uczelni.

### Użytkownicy

* Użytkownik,
* Moderator,
* Administrator.

### Korzyści z systemu

Użytkownicy:

1. Mogą się szybko ze sobą komunikować,
2. Mogą założyć własny pokój prywatny/publiczny

Moderatorzy:

1. Mogą edytować ustawienia chatu,

Administratorzy:

1. Mogą zarządzać serwerem aplikacji,
2. Posiadają pełną kontrole nad uprawnieniami moderatorów i użytkowników.

### Ograniczenia projektowe i wdrożeniowe

1. do działania serwera aplikacji potrzebna jest baza danych mariedb oraz interpreter php, ponieważ na tych technologiach opiera się działanie aplikacji,
2. aplikacja serwera wymaga systemu operacyjnego z rodziny Microsoft Windows lub Linux, ponieważ na te właśnie system jest możliwość zainstalowania wymagań podpunktu 1,
3. należy utrzymać porty 80 i 8080 otwarte, ponieważ do komunikacji klient serwer wykorzystywany jest protokół http,
4. należy dokupić dodatkowe serwery w celu zminimalizowania ryzyka awarii.

## Wymagania funkcjonalne

### Lista wymagań

1. Konta użytkowników można dodawać bez posiadania konta w systemie,
2. Konta użytkowników mogą usuwać użytkownicy po zalogowaniu się,
3. Dane kont użytkowników mogą wyświetlić zalogowani użytkownicy,
4. Użytkownicy mogą edytować własne dane konta,
5. Użytkownicy mogą dodawać i usuwać innych użytkowników do listy znajomych,
6. Użytkownicy mogą ustawić sobie status widoczny przez innych użytkowników w pokoju,
7. Użytkownicy mogą tworzyć pokoje rozmów,
8. Pokoje mogą być ustawione jako prywatne poprzez ustawienie hasła potrzebnego na dołączenie do pokoju,
9. System zapamiętuje których wiadomości użytkownik nie przeczytał i umożliwia czytanie wiadomości od ostatniej nieprzeczytanej,
10. Użytkownik ma dostęp do historii chatu w pokoju,
11. Po stworzeniu pokoju użytkownik tworzący pokój otrzymuje status moderatora i posiada wszelkie uprawnienia pokoju,
12. Po utworzeniu pokoju tworzone są domyślne predefiniowane grupy,
13. Posiadający odpowiednie uprawnienia r może ustawić czas przez jaki mają być przechowywana historia chatu w pokoju,
14. Posiadający odpowiednie uprawnienia mogą nadawać uprawnienia poszczególnym użytkownikom i grupą użytkowników,
15. Posiadający odpowiednie uprawnienia może tworzyć nowe grupy użytkowników,
16. Posiadający odpowiednie uprawnienia może dodawać i usuwać ludzi z grup,
17. Każdy pokój ma możliwość wpisania synchronizacji z usługą ftp pozwalając poprzez wpisanie przez moderatora/osoby posiadającej odpowiednie uprawnienie loginu i hasła do ftp na wysyłanie i pobieranie plików przez osoby w danym pokoju bez znania poświadczeń do serwera ftp,
18. Każdy użytkownik może usunąć / edytować własne wiadomości,
19. Pokoje posiadają logi pozwalające na odzyskanie usuniętych wiadomości,
20. Każdy użytkownik danego pokoju może przeglądać historie edycji każdej wiadomości,
21. Posiadający odpowiednie uprawnienia może pobrać / przeglądać logi pokoju,
22. Możliwość oznaczenia użytkownika w wiadomości tak by wyróżniała się ona dla oznaczonego użytkownika,
23. Możliwość oznaczenia pokoju w wiadomości tak by stał się on linkiem do niego prowadzącym,
24. Możliwość generowania linków jednoznacznie prowadzących do konkretnego pokoju,
25. Każdy użytkownik może oceniać innych użytkowników dając mu + lub – wedle uznania,
26. Posiadający odpowiednie uprawnienia może zapraszać użytkowników do pokoju,
27. Odnośnik do pokoju nie jest zaproszeniem,
28. Każde zaproszenie musi zaakceptować osoba zaproszona,
29. Każdy użytkownik ma podgląd historii Nicków użytkowników z pokojów do których ma dostęp,
30. Każdy użytkownik posiada tag jednoznacznie go identyfikujący,
31. Każdy użytkownik serwera ma możliwość tworzenia własnych pokoi chatu,
32. Posiadający odpowiednie uprawnienia może usuwać wiadomości innych użytkowników w pokoju,
33. Moderator i administrator ma mieć możliwość nakładania kary na użytkownika w formie czasowego lub permanentnej blokady dostępu do pokoju lub serwera.

### Diagramy przypadków użycia

Diagramy są załączone w pliku programu visual paradigm.

### Szczegółowy opis wymagań

każde na nowej stronie wg następujących punktów:

* Numer – jako ID
* Nazwa
* Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w 5.1.5. (id i treść elementu, do którego się odwołujemy)
* Użytkownicy
* Scenariusze, dla każdego z nich:
* Warunki początkowe
* **Przebieg działań –** numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
* Efekty – warunki końcowe
* Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
* Częstotliwość - na skali 1-5 lub BN-BW
* Istotność – inaczej: zależność krytyczna, znaczenie - na skali 1-5 lub BN-BW

***Ważne!***

*Elementy od warunków początkowych do końca mogą być grupowane, tj. specyfikacja pojedynczego przypadku użycia może zawierać:*

*- pojedynczy przebieg działań (scenariusz główny) oraz ew. scenariusze alternatywne, albo*

*- wiele przebiegów głównych wraz z ew. scenariuszami alternatywnymi – wtedy każdy z przebiegów głównych powinien być opisany wg tych punktów (od warunków początkowych do końca).*

## Wymagania niefunkcjonalne

1. Aplikacja ma mieć interfejs w stylu minimalistycznym
2. Aplikacja ma posiadać tryb w którym przeważają jasne kolory
3. Aplikacja ma posiadać tryb w którym przeważają ciemne kolory
4. Wiadomości nie mogą pobierać się powyżej 0,5s
5. Gdzie to możliwe operacje na bazie danych mają być poniżej 10ms
6. Hasła w bazie danych nie mogą być przechowywane otwartym tekstem

# Zarządzanie projektem

## Zasoby ludzkie

(rzeczywiste lub hipotetyczne) – przy realizacji projektu

Należy założyć, że projekt byłby realizowany w całości jako projekt komercyjny a nie tylko częściowo w ramach zajęć na uczelni

## Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)

## Etapy/kamienie milowe projektu

dla głównych etapów projektu

# Zarządzanie ryzykiem

## Lista czynników ryzyka

Wypełniona lista kontrolna

## Ocena ryzyka

prawdopodobieństwo i wpływ

## Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

# Zarządzanie jakością

## Scenariusze i przypadki testowe

szczegółowy plan testowania systemu – głównie testowanie funkcjonalności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

* numer – jako ID
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* kategoria – poziom/kategoria testów
* opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
* tester - konkretna osoba lub klient/pracownik,
* termin – kiedy testowanie ma być przeprowadzane,
* narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu
* założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań
* *przebieg lub zestaw danych testowych musi zawierać jawną informację o warunku zaliczenia testu*

# Projekt techniczny

## Opis architektury systemu

z ew. rysunkami pomocniczymi

## Technologie implementacji systemu

tabela z listą wykorzystanych technologii, każda z uzasadnieniem

## Diagramy UML

każdy diagram ma mieć tytuł oraz ma być na osobnej stronie

diagramy przypadków użycia umieszczone w punkcie 5.2.2, a nie tutaj.

### Diagram(-y) klas

### Diagram(-y) czynności

### Diagramy sekwencji

co najmniej 5, w tym co najmniej 1 przypadek użycia zilustrowany kilkoma diagramami sekwencji

### Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

## Charakterystyka zastosowanych wzorców projektowych

informacja opisowa wspomagana diagramami (odsyłaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich

## Projekt bazy danych

### Schemat

w trzeciej formie normalnej; jeśli w innej to umieć uzasadnić wybór

### Projekty szczegółowe tabel

## Projekt interfejsu użytkownika

Co najmniej dla głównej funkcjonalności programu – w razie wątpliwości, uzgodnić z prowadzącym zajęcia

### Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

### Przejścia między głównymi elementami

### Projekty szczegółowe poszczególnych elementów

każdy element od nowej strony z następującą minimalną zawartością:

* numer – ID elementu
* nazwa – np. formularz danych produktu
* projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi informacjami (nie pusty!!!)
* opcjonalnie:
* opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco czytelna
* wykorzystane dane – jakie dane z bazy danych są wykorzystywane
* opis działania – tabela pokazująca m.in. co się dzieje po kliknięciu przycisku, wybraniu opcji z menu itp.

## Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

# Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

* pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
* może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

# Podsumowanie

## Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – udział każdej osoby w skali całego projektu

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach