

# **Dokumentacja projektu zaliczeniowego**

Przedmiot: Inżynieria oprogramowania

Temat:	<b>Komunikator Tekstowy</b>
Autorzy:	<b>Krzysztof Kubiś, Przemysław Kowal</b>
Grupa:	I1-213A
Kierunek:	informatyka
Rok akademicki:	2019/2020
Poziom i semestr:	I/4
Tryb studiów:	stacjonarne

# 1 Spis treści

2	Odnośniki do innych źródeł.....	4
3	Słownik pojęć .....	5
4	Wprowadzenie .....	6
4.1	Cel dokumentacji.....	6
4.2	Przeznaczenie dokumentacji .....	6
4.3	Opis organizacji.....	6
4.4	Analiza SWOT organizacji .....	7
5	Specyfikacja wymagań .....	8
5.1	Charakterystyka ogólna.....	8
5.2	Wymagania funkcjonalne.....	9
5.3	Wymagania нефункционалне.....	18
6	Zarządzanie projektem .....	19
6.1	Zasoby ludzkie .....	19
6.2	Harmonogram prac.....	19
6.3	Etapy/kamienie milowe projektu .....	19
7	Zarządzanie ryzykiem.....	20
7.1	Lista czynników ryzyka .....	20
7.2	Ocena ryzyka.....	20
7.3	Plan reakcji na ryzyko .....	20
8	Zarządzanie jakością.....	21
8.1	Scenariusze i przypadki testowe .....	21
9	Projekt techniczny .....	25
9.1	Opis architektury systemu.....	25
9.2	Technologie implementacji systemu.....	25
9.3	Diagramy UML .....	25
9.4	Charakterystyka zastosowanych wzorców projektowych.....	25
9.5	Projekt bazy danych .....	26
9.6	Projekt interfejsu użytkownika.....	28
9.7	Procedura wdrożenia .....	29
10	Dokumentacja dla użytkownika.....	30
11	Podsumowanie .....	31
11.1	Szczegółowe nakłady projektowe członków zespołu .....	31

12	Inne informacje .....	32
----	-----------------------	----

## **2 Odnośniki do innych źródeł**

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

### 3 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 te same uprawnienia co użytkownik oraz może posiadać następujące: edycja ustawień chatu, tworzenie zaproszeń do pokoi, nakładać kary na użytkowników, czytać logi chatu,
- **Administrator** - to osoba która posiada wszystkie uprawnienia do wszystkich pokoi,
- **Chat** - to zbiór wiadomości wysłanych w konkretnym pokoju,
- **Logi chatu** – plik w którym znajdują się wypisane wszystkie akcje podjęte przez użytkowników w pokoju,
- **Partner uczelni** – konto użytkownika należące do osoby z firmy zewnętrznej,
- **Prowadzący** – konto należące do wykładowcy,
- **Pomocnik** – student wyznaczony do pomocy prowadzącemu,
- **Banowanie** – nakładanie kary trwałej lub tymczasowej na użytkownika,
- **Unbanowanie** – to operacja odwrotna do banowania.

## **4 Wprowadzenie**

### ***4.1 Cel dokumentacji***

Celem dokumentacji jest przedstawienie specyfikacji dla czatu tekstowego zawierającego:

- Definicje,
- Cele,
- Zawartość,
- Główną funkcjonalność.

### ***4.2 Przeznaczenie dokumentacji***

Dokumentacja jest przeznaczona celem przedstawienia jej:

1) Zaangażowanym w stworzenie i rozwój projektu:

- Menadżerom projektu,
- Programistom, którzy zaimplementują jej funkcjonalne jak i нефункционалне wymagania,
- Testerom, którzy będą musieli zweryfikować spełnienie wszystkich wymagań.

2) Użytkownikom, którzy wykorzystają je:

- Do prywatnych celów celem zwiększonej integracji,

### ***4.3 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. Aktualnie do komunikacji uczelnia wykorzystuje witrynę <https://webmail.zut.edu.pl/>, która pełni funkcję internetowej skrzynki pocztowej.

#### 4.4 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
Swoboda działania użytkowników	7	Potrzeba dodatkowych szkoleń	6
Załatwienie większości spraw uczelnianych w jednym miejscu	8	Wymagane utworzenie i utrzymanie bazy danych	8
Razem: 54		Razem: 46	

Szanse	[%]	Zagrożenia	[%]
Możliwość zwiększonej integracji pośród studentów	18	Awaryjność serwerów	11
Możliwość zwiększenia prestiżu poprzez unowocześnienie uczelni	12	Niewydolność filtrowania niewłaściwych treści	9
Możliwość zwiększenia szybkości komunikacji	14	Przeładowanie serwerów w konkretnym czasie	8
Sprawniejsze załatwianie spraw uczelnianych	16	Podatność na ataki hackerskie	13
Razem: 60		Razem: 40	

## **5 Specyfikacja wymagań**

### **5.1 Charakterystyka ogólna**

#### **5.1.1 Definicja produktu**

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

#### **5.1.2 Podstawowe założenia**

1. Aplikacja ma służyć do przesyłania wiadomości tekstowych,
2. Aplikacja pozwala na posiadanie listy znajomych,
3. Aplikacja pozwala na zakładanie pokoi publicznych i prywatnych,
4. Aplikacja pozwala na moderowanie pokoi,
5. Aplikacja pozwala na podnoszenie praw użytkowników,
6. Aplikacja pozwala na przesyłanie plików poprzez integrację pokoju z serwerem FTP,
7. Aplikacja pozwala na posiadanie wiele otwartych pokoi w zakładkach,
8. Aplikacja pozwala na wyświetlanie statusu użytkownika,
9. Aplikacja zapamiętuje które wiadomości użytkownik odczytał i umożliwia przejście do nieprzeczytanych,
10. Aplikacja pozwala na przeglądanie historii chatu,
11. Aplikacja pozwala na grupowanie użytkowników pod względem uprawnień,
12. Aplikacja pozwala na edycje danych konta,
13. Aplikacja pozwala na wysyłanie zaproszeń do pokoi, które trzeba zaakceptować,
14. Aplikacja przypisuje użytkownikom indywidualny tag pozwalając na powtarzanie się pseudonimów.

#### **5.1.3 Cel biznesowy**

Celem tworzenia aplikacji jest:

- ułatwienie komunikacji na uczelni,
- cyfryzacja placówki
- alternatywny sposób komunikacji

#### **5.1.4 Użytkownicy**

- Partner uczelni,
- Student,
- Pomocnik,
- Prowadzący,
- Moderator,
- Administrator.



### **5.1.5 Korzyści z systemu**

Wszyscy użytkownicy:

1. Mogą pisać/usuwać/edytować własne wiadomości,
2. Tworzyć pokoje,
3. Mogą dodawać i usuwać znajomych,
4. Użytkownicy mogą przeglądać historie wiadomości,
5. Użytkownicy mogą przeglądać historie wiadomości od ostatniego wylogowania,

Partner uczelni:

6. Mogą wyróżniać swoje wiadomości,
7. Mogą zakładać kanały z predefiniowanymi dla nich uprawnieniami,

Pomocnik:

8. Mogą wyróżniać swoje wiadomości,

Prowadzący:

9. Mogą założyć własny pokój prywatny/publiczny,
10. Mogą zakładać kanały z predefiniowanymi dla nich uprawnieniami,

Moderatorzy:

11. Mogą edytować ustawienia chatu,
12. Mogą karać użytkowników,

Administratorzy:

13. Mogą zarządzać serwerem aplikacji,
14. Posiadają pełną kontrolę nad uprawnieniami moderatorów i użytkowników.

### **5.1.6 Ograniczenia projektowe i wdrożeniowe**

Uczenia zastrzega sobie aby aplikacja serwera działała na systemie operacyjnym Microsoft Windows lub Linux,

Baza danych ma być realizowana na relacyjnej bazie danych mariaDB,

Aplikacja ma działać na protokole http oraz portach 80 i 8080.

## **5.2 Wymagania funkcjonalne**

### **5.2.1 Lista wymagań**

1. Konta użytkowników może dodawać administrator,
2. Konta użytkowników może usuwać administrator,
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. Pokoje mogą być ustawione jako prywatne poprzez ustawienie hasła potrzebnego na dołączenie do pokoju,
8. System zapamiętuje których wiadomości użytkownik nie przeczytał i umożliwia czytanie wiadomości od ostatniej nieprzeczytanej,
9. Użytkownik ma dostęp do historii chatu w pokoju,

10. Po stworzeniu pokoju użytkownik tworzący pokój otrzymuje status moderatora i posiada wszelkie uprawnienia pokoju,
11. Po utworzeniu pokoju tworzone są domyślne predefiniowane grupy,
12. Posiadający odpowiednie uprawnienia może ustawić czas przez jaki mają być przechowywana historia chatu w pokoju,
13. Posiadający odpowiednie uprawnienia mogą nadawać uprawnienia poszczególnym użytkownikom i grupą użytkowników,
14. Posiadający odpowiednie uprawnienia może tworzyć nowe grupy użytkowników,
15. Posiadający odpowiednie uprawnienia może dodawać i usuwać ludzi z grup,
16. 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,
17. Każdy użytkownik może usunąć / edytować własne wiadomości,
18. Pokoje posiadają logi pozwalające na odzyskanie usuniętych wiadomości,
19. Każdy użytkownik danego pokoju może przeglądać historie edycji każdej wiadomości,
20. Posiadający odpowiednie uprawnienia może pobrać / przeglądać logi pokoju,
21. Możliwość oznaczenia użytkownika w wiadomości tak by wyróżniała się ona dla oznaczonego użytkownika,
22. Możliwość oznaczenia pokoju w wiadomości tak by stał się on linkiem do niego prowadzącym,
23. Możliwość generowania linków jednoznacznie prowadzących do konkretnego pokoju,
24. Każdy użytkownik może oceniać innych użytkowników dając mu + lub – wedle uznania,
25. Posiadający odpowiednie uprawnienia może zapraszać użytkowników do pokoju,
26. Odnośnik do pokoju nie jest zaproszeniem,
27. Każde zaproszenie musi zaakceptować osoba zaproszona,
28. Każdy użytkownik ma podgląd historii Nicków użytkowników z pokojów do których ma dostęp,
29. Każdy użytkownik posiada tag jednoznacznie go identyfikujący,
30. Każdy użytkownik serwera ma możliwość tworzenia własnych pokoi chatu,
31. Posiadający odpowiednie uprawnienia może usuwać wiadomości innych użytkowników w pokoju,
32. Moderator i administrator ma mieć możliwość nakładania kary na użytkownika w formie czasowego lub permanentnej blokady dostępu do pokoju lub serwera,
33. Użytkownik może otworzyć wiele kart z pokojami.

## 5.2.2 Diagramy przypadków użycia

Diagramy są załączone w pliku programu visual paradigm.

## 5.2.3 Szczegółowy opis wymagań

- Numer 30,
- Tworzenie pokoi,
- Uzasadnienie Biznesowe:
  - Wszyscy użytkownicy mogą tworzyć pokój (5.1.5.2)
- Użytkownicy:
  - Partner uczelni
  - Student
  - Prowadzący
  - Administrator
- Scenariusze:
  - Partner uczelni:
    - Warunki początkowe:
      - Użytkownik musi być zalogowany,
    - Przebieg działań:
      1. Użytkownik powiadamia system o chęci stworzenia pokoju (klika przycisk z napisem stwórz pokój rozmów),
      2. Użytkownik wprowadza nazwę pokoju,
      3. System nadaje pokojowi id,
      4. System zapisuje nazwę i id w bazie danych,
      5. System kopiuje szablon uprawnień partnera uczelni,
      6. System nadaje uprawnienia moderatora pokoju użytkownikowi, który pokój stworzył,
    - Efekty:
      - Pokój zostaje utworzony
      - Autor pokoju ma uprawnienia moderatora w tym pokoju
    - Częstotliwość użycia: 2
    - Istotność: 3
  - Student:
    - Warunki początkowe:
      - Użytkownik musi być zalogowany,
    - Przebieg działań:
      1. Użytkownik powiadamia system o chęci stworzenia pokoju (klika przycisk z napisem stwórz pokój rozmów),
      2. Użytkownik wprowadza nazwę pokoju,
      3. System nadaje pokojowi id,
      4. System zapisuje nazwę i id w bazie danych,
      5. System kopiuje szablon uprawnień Studenta,
      6. System nadaje uprawnienia moderatora pokoju użytkownikowi, który pokój stworzył,
    - Efekty:
      - Pokój zostaje utworzony

- 
- Autor pokoju ma uprawnienia moderatora w tym pokoju
- Częstotliwość użycia 2
- Istotność 3
- Prowadzący:
  - Warunki początkowe:
    - Użytkownik musi być zalogowany,
  - Przebieg działań:
    1. Użytkownik powiadamia system o chęci stworzenia pokoju (klika przycisk z napisem stwórz pokój rozmów),
    2. Użytkownik wprowadza nazwę pokoju,
    3. System nadaje pokojowi id,
    4. System zapisuje nazwę i id w bazie danych,
    5. System kopiuje szablon uprawnień prowadzącego,
    6. System nadaje uprawnienia moderatora pokoju użytkownikowi, który pokój stworzył,
  - Efekty:
    - Pokój zostaje utworzony
    - Autor pokoju ma uprawnienia moderatora w tym pokoju
  - Częstotliwość użycia: 1
  - Istotność: 3
- Administrator:
  - Warunki początkowe:
    - Użytkownik musi być zalogowany,
  - Przebieg działań:
    1. Użytkownik powiadamia system o chęci stworzenia pokoju (klika przycisk z napisem stwórz pokój rozmów),
    2. Użytkownik wprowadza nazwę pokoju,
    3. System nadaje pokojowi id,
    4. System zapisuje nazwę i id w bazie danych,
    5. System kopiuje szablon uprawnień administratora,
    6. System nadaje uprawnienia moderatora pokoju użytkownikowi, który pokój stworzył,
  - Efekty:
    - Pokój zostaje utworzony
    - Autor pokoju ma uprawnienia moderatora w tym pokoju
  - Częstotliwość użycia: 1
  - Istotność: 2

- Numer 32,
- Banowanie,
- Uzasadnienie Biznesowe:
  - Moderator i administrator może nakładać kary na użytkowników (5.1.5.12),
- Użytkownicy :
  - Moderator,
  - Administrator,
- Scenariusze:
  - Administrator:
    - Warunki początkowe:
      - Administrator uznał że użytkownik szkodzi innym,
    - Przebieg działań:
 

Podstawowy:

      1. Administrator zgłasza do systemu chęć nałożenia kary na użytkownika (klika odpowiedni przycisk),
      2. Administrator podaje nick oraz tag użytkownika,
      3. System wyszukuje użytkownika w bazie danych,
      4. Administrator podaje rodzaj kary,
      5. System odpowiednio oznacza użytkownika w bazie danych,

Alternatywny 1:

      1. Administrator zgłasza do systemu chęć nałożenia kary na użytkownika (klika odpowiedni przycisk),
      2. Administrator podaje nick oraz tag użytkownika
      3. System wyszukuje użytkownika w bazie danych
      4. Administrator podaje rodzaj kary
      5. Administrator podaje długość kary
      6. System odpowiednio oznacza użytkownika w bazie danych,

Alternatywny 2:

      1. Administrator zgłasza do systemu chęć nałożenia kary na użytkownika (klika odpowiedni przycisk),
      2. Administrator podaje nick oraz tag użytkownika
      3. System wyszukuje użytkownika w bazie danych
      4. System zgłasza brak użytkownika w bazie danych
    - Efekty:
 

Podstawowy:

      - Wybrany użytkownik zostaje zbanowany pernamennie,

Alternatywny 1:

      - Wybrany użytkownik zostaje zbanowany tymczasowo,

Alternatywny 2:

      - Brak efektów ze względu na nie znalezienie użytkownika,
  - Częstotliwość użycia: 1,
  - Istotność: 4,

- Numer 17,
- Usuwanie/edycja wiadomości,
- Uzasadnienie Biznesowe:
  - Użytkownicy mogą usuwać/edytować swoje wiadomości(5.1.5.1),
- Użytkownicy :
  - Wszyscy(5.1.4),
- Scenariusze:
  - Wszyscy(5.1.4):
    - Warunki początkowe:
      - Wiadomość istnieje,
      - Wiadomość należy do obecnie zalogowanego użytkownika
    - Przebieg działań:
      - Podstawowy:
        1. Użytkownik zgłasza systemowi chęć edycji wiadomości,
        2. Użytkownik wysyła wyedytowaną wiadomość,
        3. System wykrywa że wiadomość jest pusta,
        4. System zapisuje starą wiadomość w logach,
        5. System usuwa wiadomość,
      - Alternatywny:
        1. Użytkownik zgłasza systemowi chęć edycji wiadomości,
        2. Użytkownik wysyła wyedytowaną wiadomość,
        3. System zapisuje starą wiadomość w logach,
        4. System zapisuje starą wiadomość w bazie danych
        5. System aktualizuje wiadomość,
    - Efekty:
      - Podstawowy
        - Wiadomość zostaje usunięta,
      - Alternatywny:
        - Stara wiadomość znajduje się w historii edycji,
        - Nowa wiadomość zastępuje starą,
    - Częstotliwość użycia: 3,
    - Istotność: 4,

- Numer 5,
- Dodawanie i usuwanie z listy znajomych,
- Uzasadnienie Biznesowe:
  - Użytkownicy mogą dodawać i usuwać innych użytkowników do znajomych (5.1.5.3),
- Użytkownicy :
  - Wszyscy(5.1.4),
- Scenariusze:
  - Wszyscy(5.1.4):
    - Warunki początkowe:
      - Użytkownik 1 musi być zalogowany,
      - Użytkownik 2 musi istnieć,
    - Przebieg działań:
 Podstawowy:
      1. Użytkownik 1 zgłasza chęć dodania osoby do listy znajomych,
      2. Użytkownik 1 podaje nick i tag użytkownika 2,
      3. System wyszukuje użytkownika 2 w bazie danych,
      4. Użytkownik 2 otrzymuje zaproszenie do znajomych od systemu,
      5. Użytkownik 2 akceptuje zaproszenie do znajomych,
      6. System odpowiednio oznacza to w bazie danych,
 Alternatywny 1:
      1. Użytkownik 1 zgłasza chęć dodania osoby do listy znajomych,
      2. Użytkownik 1 podaje nick i tag użytkownika 2,
      3. System wyszukuje użytkownika 2 w bazie danych,
      4. Użytkownik 2 otrzymuje zaproszenie do znajomych od systemu,
      5. Użytkownik 2 odrzuca zaproszenie do znajomych ,
      6. System odpowiednio oznacza to w bazie danych,
 Alternatywny 2:
      1. Użytkownik 1 zgłasza chęć usunięcia osoby z listy znajomych,
      2. Użytkownik 1 podaje nick i tag użytkownika 2
      3. System wyszukuje użytkownika 2 w bazie danych
      4. System usuwa oznaczenia z bazy danych
    - Efekty:
 Podstawowy
      - Użytkownik 1 ma użytkownika 2 na liście znajomych,
      - Użytkownik 2 ma użytkownika 1 na liście znajomych,
 Alternatywny 1
      - Brak
 Podstawowy
      - Użytkownik 1 nie ma użytkownika 2 na liście znajomych,
      - Użytkownik 2 nie ma użytkownika 1 na liście znajomych,

- Częstotliwość użycia: 1,
- Istotność: 2,



- Numer 8,
- Wiadomości nieprzeczytane,
- Uzasadnienie Biznesowe:
  - Użytkownicy mogą przeglądać historie wiadomości od ostatniego wylogowania(5.1.5.5),
- Użytkownicy :
  - Wszyscy(5.1.4),
- Scenariusze:
  - Wszyscy(5.1.4):
    - Warunki początkowe:
      - Użytkownik musi być zalogowany,
      - Użytkownik musi mieć dostęp do historii wiadomości w danym pokoju,
    - Przebieg działań:
      1. Użytkownik loguje się do pokoju,
      2. System sprawdza jaka była ostatnia wiadomość wysłana przed wylogowaniem użytkownika z pokoju
      3. System oznacza pierwszą wiadomość wysłaną po wylogowaniu
      4. Użytkownik otrzymuje historie wiadomości
    - Efekty:
      - Historia wiadomości którą otrzymuje użytkownik zawiera oznaczenie pierwszej nieprzeczytanej wiadomości,
    - Częstotliwość użycia: 5,
    - Istotność: 1,

### **5.3 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. Operacje na bazie danych mają być realizowane poniżej 10ms
6. Hasła w bazie danych nie mogą być przechowywane otwartym tekstem
7. System ma zostać naprawiony po awarii nie krytycznej w czasie krótszym niż 10 godzin i ma w tym czasie funkcjonować dalej,
8. System ma zostać przywrócony do działania po awarii krytycznej w czasie krótszym niż 3 godzin i na ten czas ma zostać wyłączony,
9. Raz na 24 godziny ma być robiona kopia zapasowa bazy danych i utrzymywana przez 7 dób,
10. Aplikacja ma obsługiwać jednocześnie zalogowanych 2000 użytkowników.

## 6 Zarządzanie projektem

### 6.1 Zasoby ludzkie

Dwóch studentów wydziału informatyki.

### 6.2 Harmonogram prac

l p	nazwa	dn i	data rozpoczęcia	data zakończenia	
1	Przygotowanie narzędzi	2	15.05.2020	21.05.2020	data rozpoczęcia projektu  15.05.2020
2	Layout strony	14	21.05.2020	23.05.2020	
3	Przygotowanie struktury bazy danych	2	23.05.2020	25.05.2020	
4	Funkcjonalność klienta	28	25.05.2020	22.06.2020	
5	Funkcjonalność serwera	28	22.06.2020	20.07.2020	
6	Testy	10	20.07.2020	30.07.2020	
7	Naprawa błędów	10	30.07.2020	09.08.2020	

### 6.3 Etapy/kamienie milowe projektu

- 1.Przygotowanie 1-4
- 2.Funkcjonalność 5-6
- 3.Testy 7-8

## **7 Zarządzanie ryzykiem**

### **7.1 *Lista czynników ryzyka***

Wypełniona lista kontrolna

### **7.2 *Ocena ryzyka***

prawdopodobieństwo i wpływ

### **7.3 *Plan reakcji na ryzyko***

Działania w odniesieniu do poszczególnych ryzyk.

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

## 8 Zarządzanie jakością

### 8.1 Scenariusze i przypadki testowe

- Numer 1
- logowanie
- Testy dymne
- tester – Użytkownik,
- termin – nie wcześniej niż 20.07.2020,
- narzędzia wspomagające – Brak

Przebieg działań : logowanie		
Lp	Tester/Użytkownik	system
1	wpisuje login	
2	wpisuje hasło	
3	zatwierdza	
4		sprawdza poprawność danych logowania

- założenia – Uda się poprawnie zalogować użytkownika do systemu
- środowisko – Przeglądarka internetowa
- warunki wstępne – Użytkownik musi mieć połączenie internetowe z serwerem
- zestaw danych testowych

Przebieg działań : logowanie		
lp	Tester/Użytkownik	system
1	wpisuje login = "TestUser31"	
2	wpisuje hasło = "Jaskółka33Królowa33Wiatru"	
3	zatwierdza	
4		Loguje użytkownika do systemu

Przebieg działań : logowanie (błędne dane)		
lp	Tester/Użytkownik	system
1	wpisuje login = "kp44374"	
2	wpisuje hasło = "jwvs5nhjLo"	
3	zatwierdza	
4		Wyświetla komunikat: "błędne dane"

Przebieg działań : logowanie (brak danych)		
lp	Tester/Użytkownik	system
1	wpisuje login = ""	
2	wpisuje hasło = ""	
3	zatwierdza	
4		wyświetla komunikat: "login/hasło nie może być puste"

- Numer 2
- banowanie
- Testy dymne
- tester – Moderator, Użytkownik
- termin – nie wcześniej niż po zrealizowaniu 'funkcjonalności serwera' i rozpoczęciu punktu 'Testy'
- narzędzia wspomagające – brak

Przebieg działań:	Banowanie	
lp.	Tester/Moderator	system
1	Łączy się z kanałem	
2	Wpisuje login	
3	Zatwierdza	
4		Wyszukuje w bazie danych
5	Wpisuje długość kary	
6		Oznacza w bazie danych

Przebieg działań:	Banowanie	
lp.	Tester/Użytkownik	system
1	Loguje się	
2		Wyswietla listę kanałów
3	Wybiera kanał	
4		Wyświetla komunikat o odmowie dostępu

- założenia – Uda się poprawnie nałożyć karę na użytkownika o podanym loginie
- środowisko – Przeglądarka internetowa
- warunki wstępne – Tester musi mieć połączenie internetowe z serwerem oraz posiadać konto z uprawnieniami rangi Moderator na danym kanale.

- zestaw danych testowych

Przebieg działań:	Banowanie(czasowe na 30 dni)	
lp.	Tester/Moderator	system
1	Łączy się z kanałem="Testowy"	
2	Wpisuje login="TestUser31"	
3	Zatwierdza	
4		Wyszukuje w bazie danych
5	Wpisuje długość kary="30"	
6		Oznacza w bazie danych na 30 dni

Przebieg działań:	Banowanie(czasowe na 30 dni)	
lp.	Tester/Użytkownik	system
1	Loguje się jako "TestUser31"	
2		Wyświetla listę kanałów
3	Wybiera kanał="Testowy"	
4		System wyświetla komunikat "Brak dostępu. Została na ciebie nałożona kara 30 dni"

Przebieg działań:	Banowanie(permanentne)	
lp.	Tester/Moderator	system
1	Łączy się z kanałem="Testowy"	
2	Wpisuje login="TestUser31"	
3	Zatwierdza	
4		Wyszukuje w bazie danych

5	Wpisuje długość kary=""	
6		Oznacza w bazie danych jako NULL, co jest równoznaczne z czasem nieokreślonym

Przebieg działań:	Banowanie(permanentne)	
lp.	Tester/Użytkownik	system
1	Loguje się jako "TestUser31"	
2		Wyświetla listę kanałów
3	Wybiera kanał="Testowy"	
4		System wyświetla komunikat "Brak dostępu. Została na ciebie nałożona kara permanentna"

Przebieg działań:	Banowanie(błędne dane)	
lp.	Tester/Moderator	system
1	Łączy się z kanałem="Testowy"	
2	Wpisuje login="admin"	
3	Zatwierdza	
4		Wyszukuje w bazie danych
5		Wyświetla komunikat "Błędne dane"



## 9 Projekt techniczny

### 9.1 Opis architektury systemu

Użytkownik będzie łączył się z systemem poprzez Internet protokołem http/https.

A sam system można podzielić na aplikację, która będzie przetwarzać te zapytania i wysyłać adekwatne odpowiedzi. Bazę danych, z którą aplikacja będzie się łączyć przy wykorzystaniu SQL.

### 9.2 Technologie implementacji systemu

Technologia	Uzasadnienie
SQL	Obsługa bazy danych
PHP	Wygodny do obsługi zapytań http/https
JS	Wykonywanie skryptów po stronie klienta
HTML i CSS	Formatowanie tekstu w odpowiedziach na zapytania http/https
Bootstrap	Ułatwia obsługę HTML, CSS, JS

### 9.3 Diagramy UML

#### 9.3.1 Diagram(-y) klas

Diagramy są załączone w pliku programu visual paradigm.

#### 9.3.2 Diagram(-y) czynności

Diagramy są załączone w pliku programu visual paradigm.

#### 9.3.3 Diagramy sekwencji

Diagramy są załączone w pliku programu visual paradigm.

#### 9.3.4 Inne diagramy

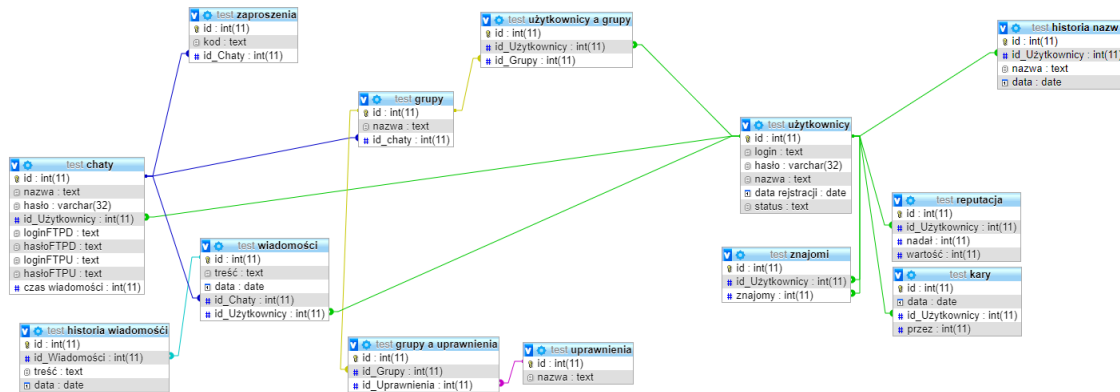
co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

### 9.4 Charakterystyka zastosowanych wzorców projektowych

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

## 9.5 Projekt bazy danych

### 9.5.1 Schemat



### 9.5.2 Projekty szczegółowe tabel

```
CREATE TABLE IF NOT EXISTS `Uzytkownicy`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `login` TEXT,  
  `haslo` VARCHAR(32),  
  `nazwa` TEXT,  
  `data rejestracji` DATE,  
  `status` TEXT)
```

```
CREATE TABLE IF NOT EXISTS `Reputacja`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `id_Uzytkownicy` INT,  
  `nadal` INT,  
  `wartosc` INT,  
  FOREIGN KEY(`id_Uzytkownicy`) REFERENCES `Uzytkownicy`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Historia nazw`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `id_Uzytkownicy` INT,  
  `nazwa` TEXT,  
  `data` DATE,  
  FOREIGN KEY(`id_Uzytkownicy`) REFERENCES `Uzytkownicy`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Znajomi`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `id_Uzytkownicy` INT,  
  `znajomy` INT,  
  FOREIGN KEY(`id_Uzytkownicy`) REFERENCES `Uzytkownicy`(`id`),  
  FOREIGN KEY(`znajomy`) REFERENCES `Uzytkownicy`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Chaty`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `nazwa` TEXT,  
  `hasło` VARCHAR(32),  
  `id_Użytkownicy` INT,  
  `loginFTPD` TEXT,  
  `hasłoFTPD` TEXT,  
  `loginFTPU` TEXT,  
  `hasłoFTPU` TEXT,  
  `czas wiadomości` INT,  
  FOREIGN KEY(`id_Użytkownicy`) REFERENCES `Użytkownicy`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Wiadomości`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `treść` TEXT,  
  `data` DATE,  
  `id_Chaty` INT,  
  `id_Użytkownicy` INT,  
  FOREIGN KEY(`id_Chaty`) REFERENCES `Chaty`(`id`),  
  FOREIGN KEY(`id_Użytkownicy`) REFERENCES `Użytkownicy`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Historia wiadomości`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `id_Wiadomości` INT,  
  `treść` TEXT,  
  `data` DATE,  
  FOREIGN KEY(`id_Wiadomości`) REFERENCES `Wiadomości`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Kary`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `data` DATE,  
  `id_Użytkownicy` INT,  
  `przez` INT,  
  FOREIGN KEY(`id_Użytkownicy`) REFERENCES `Użytkownicy`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `zaproszenia`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `kod` TEXT,  
  `id_Chaty` INT,  
  FOREIGN KEY(`id_Chaty`) REFERENCES `Chaty`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Grupy`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `nazwa` TEXT,  
  `id_chaty` INT,
```

```
FOREIGN KEY(`id_chaty`) REFERENCES `chaty`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Uprawnienia`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `nazwa` TEXT)
```

```
CREATE TABLE IF NOT EXISTS `Grupy a Uprawnienia`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `id_Grupy` INT,  
  `id_Uprawnienia` INT,  
  FOREIGN KEY(`id_Grupy`) REFERENCES `Grupy`(`id`),  
  FOREIGN KEY(`id_Uprawnienia`) REFERENCES `Uprawnienia`(`id`))
```

```
CREATE TABLE IF NOT EXISTS `Użytkownicy a Grupy`(  
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  `id_Użytkownicy` INT,  
  `id_Grupy` INT,  
  FOREIGN KEY(`id_Użytkownicy`) REFERENCES `Użytkownicy`(`id`),  
  FOREIGN KEY(`id_Grupy`) REFERENCES `Grupy`(`id`))
```

## ***9.6 Projekt interfejsu użytkownika***

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

### **9.6.1 Lista głównych elementów interfejsu**

okien, stron, aktywności (Android)

### **9.6.2 Przejścia między głównymi elementami**

### **9.6.3 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.

### ***9.7 Procedura wdrożenia***

1	Przygotowanie domyślnej konfiguracji	6	09.08.2020	15.08.2020
2	Konfiguracja serwera	6	15.08.2020	21.08.2020
3	Beta testy	7	21.08.2020	28.08.2020
4	testy na użytkownikach końcowych	7	28.08.2020	04.09.2020
5	wprowadzenie poprawek	7	04.09.2020	11.09.2020

## **10 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.

## **11 Podsumowanie**

### ***11.1 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

## **12 Inne informacje**

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach