

Inżynieria Oprogramowania

Łączenie ludzi poprzez pasję

Autorzy: Jakub Ziomek, Kamil Poniewierski, Tomasz Turek
Kierunek studiów: Informatyka i Systemy Inteligentne, II rok

Spis Treści

Streszczenie systemu	2
Lista obiektów	2
Udokumentowane wymagania	2
Wymagania funkcjonalne	2
Wymagania niefunkcyjne	3
Diagram przypadków użycia	4
Diagram ER	5
Architektura Systemu	6
Opis Interfejsów	7
Lista wykorzystanych technologii	8
Projekt Testów	9
Analiza Ryzyka	11
Lista Narzędzi	14
Licencja	15

Streszczenie systemu

Nasz projekt modeluje wypożyczanie pomieszczeń dla ludzi chcących wykonywać swoje pasje. Jego zadaniem jest ułatwienie kontaktu między ludźmi o podobnych zainteresowaniach oraz wynajmowania miejsc do uprawiania swojej pasji. Klient poprzez aplikację wybiera miasto oraz rodzaj aktywności jaką chce wykonywać. Klient za pomocą wiadomości może się dogadać z potencjalnym właścicielem lokum które będzie udostępniane i ustalić dokładne zasady. Po zakończeniu użytkownicy wystawiają sobie oceny i zbierają punkty, które umiejscawiają ich wyżej w rankingu.

Lista obiektów

- Użytkownik
 - Użyczacz lokum
 - Osoba która chce wynająć na jakiś czas
- Aplikacja webowa

Udokumentowane wymagania

a. Wymagania funkcjonalne

■ Dokładność

Na podstawie wpisanych miejscowości i daty dostępu lokalu, jesteśmy w stanie stwierdzić czy użytkownik nie popełnił błędu np. przy wpisywaniu kogoś na dany termin, jak ten termin już jest przez kogoś zarezerwowany na dany czas

■ Wydajność

Gdy użytkownik przejdzie do ekranu w którym wyświetla się aktywność zgodna z filtrami podanymi przez użytkownika to te dane powinny się załadować w ciągu 3 sekund.

■ Użyteczność

Aplikacja dostarcza użytkownikowi informacji czy w pobliżu są dostępne miejsca w których może uprawiać aktywność na jaką aktualnie ma ochotę i ewentualnie ją zarezerwować na przyszłość.

■ Niezawodność

Podczas gdy użytkownik np. wpisuje dane odnośnie lokacji jaką chce udostępnić i terminy, w razie nagłej utraty łącza internetowego, jego dane są tymczasowo zapisywane i po ponownym połączeniu się z internetem może kontynuować pracę bez utraty postępów.

■ Rejestracja

Na podstawie wpisywanych danych użytkownik jest dodawany do bazy danych i w ten sposób może zalogować się do systemu.

■ Logowanie

Na podstawie wpisywanych danych logowania, aplikacja sprawdza czy są one poprawne. Następnie zwraca wiadomość do użytkownika o ponownym ich wpisaniu lub przekierowuje go do strony z jego profilem.

■ Wyszukiwanie

Użytkownik dostaje informacje o dostępnych obiektach w jego okolicy na podstawie ustalonych filtrów.

■ Nawiązywanie połączeń

Użytkownik może nawiązać połączenie z hostem obejmujące umowę na użycie określonego obiektu dostarczanego przez hosta na określony czas dla określonej aktywności.

■ Usuwanie połączeń

Użytkownik oraz Host może usuwać nawiązane połączenia

■ Dodawanie, Usuwanie i Modyfikacja lokalizacji

Host może zarządzać swoimi lokalizacjami - usuwać je i modyfikować oraz dodawać nowe.

b. Wymagania niefunkcyjne

■ Interoperatywność

Ze względu na to że nasza aplikacja jest aplikacją webową, jesteśmy w stanie korzystać z niej przez telefony z oprogramowaniem Android i IOS, ale także za pomocą komputera z oprogramowaniem Linux, Windows i MacOS.

■ Bezpieczeństwo

Logowanie do naszej aplikacji jest możliwe za pomocą kluczy SSH, które generują parę kluczy (publicznego i prywatnego) na podstawie których następuje autoryzacja dostępu do serwera.

■ Konserwowalność

Nasz kod składa się z kilku tysięcy linii i posiada komentarze odpowiednie do danego modułu i metody więc jego czas naprawy jest relatywnie niski.

■ Przenośność

Nasza aplikacja jest dostępna na urządzeniach mobilnych ale i na komputerach, a do jej działania wystarczy połączenie internetowe

Diagram przypadków użycia

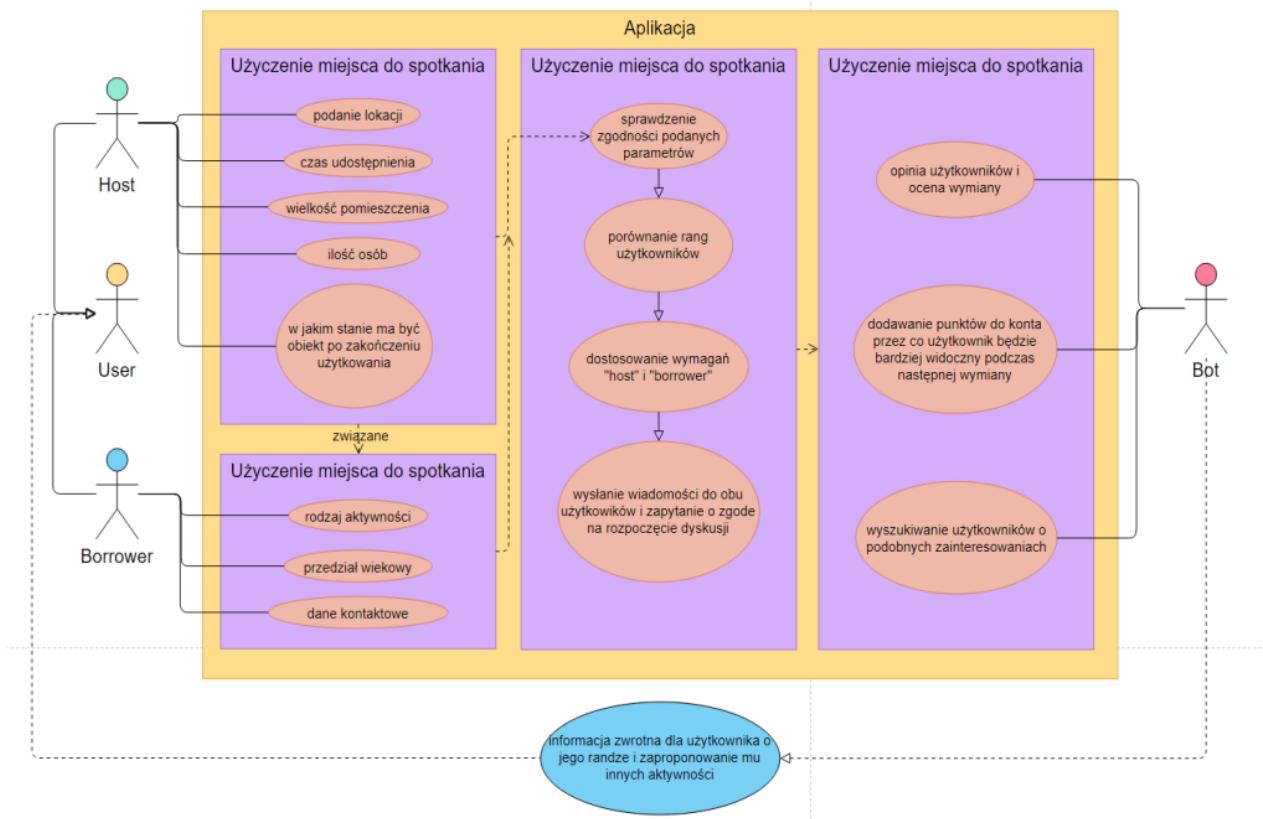
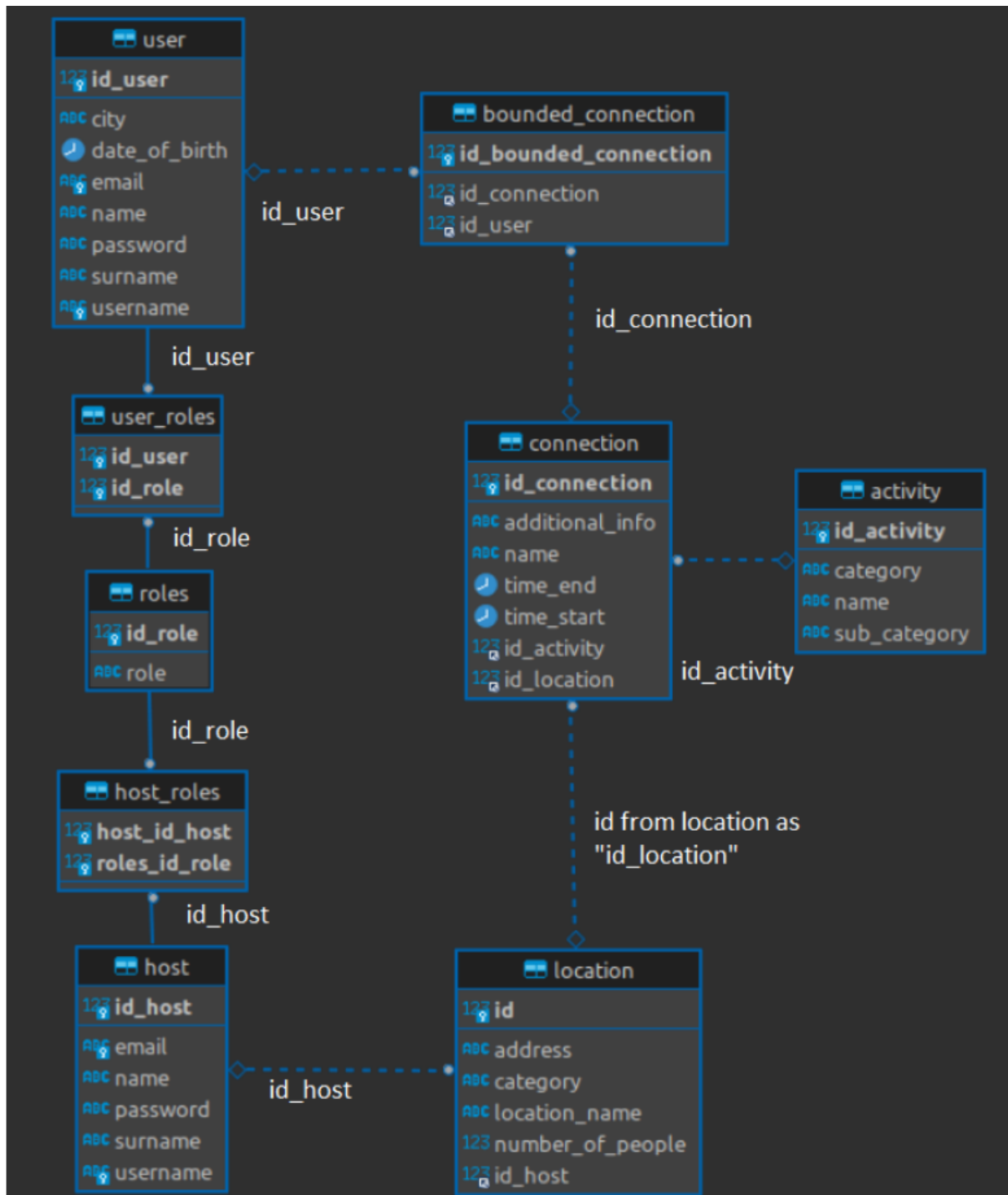
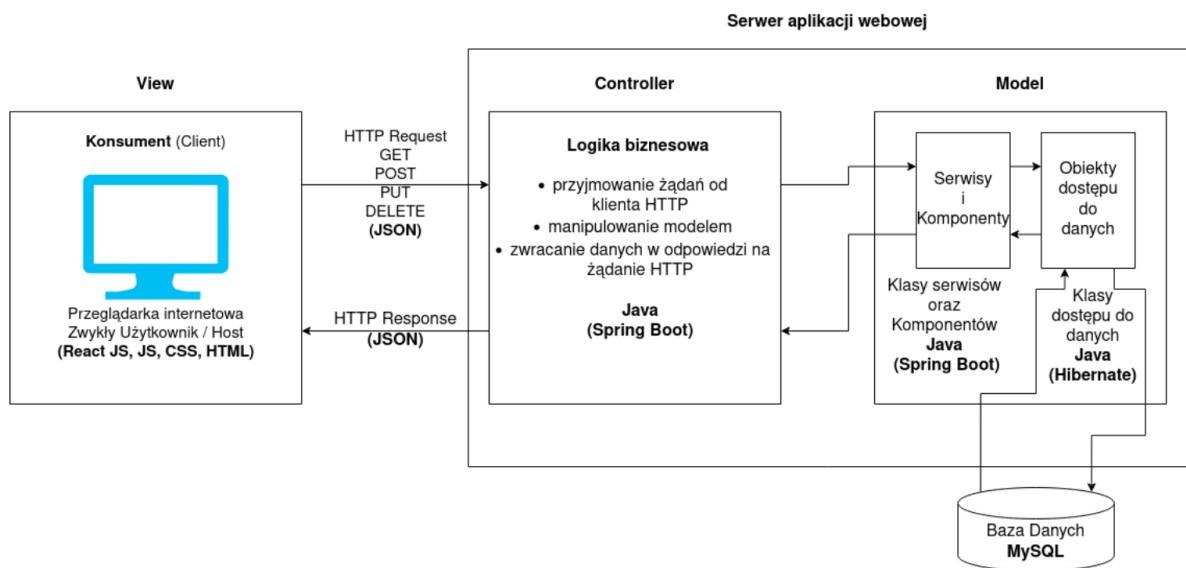


Diagram ER



Architektura Systemu



Opis Interfejsów

Token wykorzystywany do autoryzacji użytkownika zwracany jest przez serwer po zalogowaniu się użytkownika i ma następujący format:

“Bearer” + spacja + 171 znaków wygenerowanych (token JWT)

Kod	Znaczenie
P1	Użytkownik dodany pomyślnie
P2	Lokalizacja zaktualizowana
P3	Nawiązanie Połączenia
P4	Pomyślne usunięcie połączenia
P5	Pomyślne dodanie lokalizacji
P6	Pomyślne usunięcie lokalizacji
E1	Nazwa użytkownika zajęta
E2	Email zajęty
E3	Niepoprawny login lub hasło
E4	Nazwa Lokalizacji zajęta
E5	Nie Udało się nawiązać połączenia
E6	Nie udało się usunąć połączenia
E7	Nie udało się zaktualizować lokalizacji
E8	Nie udało się usunąć lokalizacji

Lista wykorzystanych technologii

a. MySQL

Jest otwartą i wielowątkową bazą danych, MySQL zapewnia wysoką wydajność, skalowalność i jest dobrym wyborem dla aplikacji, które opierają się na transakcjach wielorzędowych, ale także na dla aplikacji wypożyczania miejsca na aktywności

b. JavaScript

Automatycznie animuje elementy utworzone w HTML i CSS. Jest to język skryptowy, który tworzy dynamiczną zawartość, która obejmuje między innymi pokazy slajdów ze zdjęciami, animowaną grafikę, interaktywne formularze lub sugestie autouzupełniania tekstu.

c. CSS

Formalny język opisu służący do dodawania stylu (np. Czcionek, kolorów lub odstępów) do dokumentów internetowych napisanych w HTML.

d. HTML

Język znaczników używany do tworzenia i strukturyzacji sekcji, akapitów, nagłówków i innych elementów prezentowanych na stronie internetowej.

e. Java

Współbieżny, oparty na klasach, obiektowy język programowania ogólnego zastosowania. Został przez nas wybrany przez swoją wysoką popularność, a także dużą bazę poradników i porad dostępnych w internecie.

f. Spring

Jest platformą złożoną z wielu projektów, dedykowaną do tworzenia aplikacji w Javie. Wybraliśmy go dlatego, że umożliwia tworzenie łatwych do wdrożenia aplikacji, posiada wiele podmodułów co ułatwia dostosowanie go do specyficznych potrzeb i jest łatwy do nauki.

■ JPA, Hibernate

Wykorzystywane do realizacji warstwy dostępu do danych. Zapewniają przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiektywnym. Zalety rozwiązania to brak wymagania znajomości SQL, intuicyjne i proste mapowanie, kompatybilności dla szeregu baz danych.

g. React JS

Biblioteka JS, która wykorzystywana jest do tworzenia interfejsów graficznych aplikacji internetowych. Została wybrana przez nas ponieważ pozwala na ponowne wykorzystanie komponentów, kod jest stabilny, pozwala zbudować bardzo dynamiczny interfejs.

h. Apache Tomcat

Kontener dla samodzielnych aplikacji sieciowych. Wybraliśmy go dlatego, że jest to darmowe rozwiązanie niezależne od platformy, szybko i łatwo można go skonfigurować.

Projekt Testów

Chcielibyśmy jak najbardziej zautomatyzować testowanie naszej aplikacji w celu przyspieszenia prac nad nią i unikaniu regresji. W tym celu będziemy pokrywać kod możliwie szeroko (minimalny akceptowalny poziom pokrycia testami to 40%, optymalny ok. 80%, idealny - pełne pokrycie testami) i na różnych poziomach - od testów jednostkowych, które będą najliczniejsze, poprzez testy integracyjne, aż po testy end-to-end. Przykładowo, dla segmentu ostatnich aktywności chcielibyśmy przeprowadzić następujące testy:

- Funkcjonalne
 - Użytkownik powinien widzieć swoje ostatnie aktywności
 - Kafelki aktywności powinien składać się z ikony aktywności, nazwy, daty oraz wyświetlać informację o innych użytkownikach, którzy brali w niej udział
 - Naciśnięcie kafelka powinno pokazać użytkownikowi szczegóły aktywności
 - Użytkownik powinien móc za pomocą panelu przejść do sekcji, w której mógłby zobaczyć wszystkie swoje aktywności
- Niefunkcjonalne
 - Kafelki powinny wyświetlać się możliwie jak najszybciej, akceptowalny czas ładowania poniżej 3 sekund
 - System powinien obsługiwać co najmniej 20 zapytań jednocześnie
 - W celu przetestowania tego wymagania uruchomimy serwer oraz 20 instancji klientów, którzy jednocześnie wyślą zapytanie o dane potrzebne do wyświetlenia kafelków. Jeśli średni czas oczekiwania przekroczy 2 sekundy, a maksymalny 3 sekundy, test zostanie uznany za niezaliczony.

Testy które zostaną wykonane dla warstwy backendu są następujące:

- Jednostkowe
 - Testy dla poszczególnych mapperów dla obiektów z bazy danych sprawdzające poprawność mapowania:
 - Aktywności - czy poprawnie przypisane:
 - id
 - name
 - category
 - subCategory
 - Połączeń - czy poprawnie przypisane:
 - id
 - name
 - timeStart
 - timeEnd
 - additionalInfo
 - location
 - activity
 - Połączeń wraz z użytkownikiem - czy poprawnie przypisane:
 - connection

- user
- Hostów - czy poprawnie przypisane:
 - username
 - password
 - locations
- Lokalizacji - czy poprawnie przypisane:
 - id
 - address
 - locationName
 - category
 - numberOfPeople
- Użytkowników
 - id
 - username
 - name
 - dateOfBirth
 - city
- Testy dla kontrolerów poszczególnych endpointów sprawdzające poprawność zwracanych przez nie danych lub odpowiednich kodów błędów opisanych w tabeli interfejsy:
 - Kontroler autoryzacji - /login i /singup
 - Kontroler połączeń - /connections, /connections/{id}
 - Kontroler lokalizacji - /locations, /locations/{id}
 - Kontroler strony głównej
 - Kontroler rekomendacji - /recommendation, /recommedation/{id}
- integracyjne, manualne - zarejestrowanie użytkownika, zalogowanie się, sprawdzenie wylistowanych lokalizacji i dostępnych rekomendacji, nawiązanie kilku połączeń, modyfikacja połączenia, usunięcie połączeń
 - Wykonanie wszystkiego zgodnie z prawidłowym działaniem
 - Próba usunięcia połączeń które nie istnieją
 - Próba zalogowania się użytkownika z błędnym hasłem i/lub email
 - Próba modyfikacji nieistniejących połączeń
 - Próba zarejestrowania użytkownika o takim samym usernam i/lub email

Technologie, których używamy w tym celu:

- Frontend
 - Jest
 - react-testing-library
 - Cypress
- Backend
 - JUnit
- Baza danych
 - mySQL
 - mySQL workbench diagram generator

Analiza Ryzyka

Lp.	Ryzyko	Prawdopodobieństwo	Następstwa	P * następstwa	Poziom ryzyk
1	Przekroczenie terminów realizacji funkcjonalności integracji	6	3	18	niskie
2	Nieprecyzyjna lokalizacja miejsc	6	9	54	wysokie
3	Brak strategicznych decyzji projektowych lub opóźnienia w ich uzyskaniu	3	9	27	średnie
4	Awaria serwera	3	9	27	średnie
5	Nieobecność administratora	6	3	18	niskie
6	Cel może nie zostać w pełni zrealizowany	9	6	54	wysokie
7	Zamknięcie uczelni z powodów epidemiologicznych	9	6	54	niskie
8	Organizacyjne, zarządzania i związane z czynnikiem ludzkim	6	9	54	wysokie
9	Niewłaściwe wykorzystanie aplikacji przez użytkowników w celach destrukcyjnego kontaktu z innymi	3	9	27	średnie
10	Wyciek danych	3	9	27	średnie
11	Zmiany wprowadzone w aktualizacjach programów służących do rozwijania aplikacji	3	9	27	średnie
12	Natłok innych obowiązków na uczelni - kolokwia, projekty, sprawozdania	6	6	36	średnie

Skala ryzyka	Wartość ryzyka
3	2
6	6
9	4

Macierz ryzyka				
Następstwa	W	5	2	0
	Ś	0	1	2
	N	0	2	0
		Niskie	Średnie	Wysokie
	Prawdopodobieństwo			

Lp.	Ryzyko	Sposób reagowania
1	Przekroczenie terminów realizacji funkcjonalności integracji	Gdy będzie się zbliżał termin oddania danej funkcjonalności i nie będzie ona skończona kontaktujemy się z osobą odpowiedzialną za tą część i pytamy dlaczego tak jest - w razie problemów, które nie zostały wcześniej zgłoszone staramy się pomóc by się wyrobić w czasie
2	Nieprecyzyjna lokalizacja miejsc	Walidacja adresów jest skomplikowanym zadaniem i nie będziemy w stanie wykonać go w założonym czasie, pozostawimy możliwość kontaktu z ogłaszającym, ale nie będziemy rozwiązywać tego problemu w inny sposób
3	Brak strategicznych decyzji projektowych lub opóźnienia w ich uzyskaniu	Członkowie projektu, gdy nie wiedzą co mają aktualnie robić, albo w którą stronę aktualnie iść np. czy nasza aplikacja powinna mieć większe ikonki w celu zwiększenia czytelności, ale kosztem utworzenia większej ilości podstron, to następuje konsultacja danych pomysłów
4	Awaria serwera	W tym przypadku nie jesteśmy w stanie dużo zrobić. Możemy zadzwonić bądź skontaktować się z dostawcą usług serwerowych co jest przyczyną awarii i jak długo będzie ona trwała i poinformowanie o tym użytkowników
5	Nieobecność administratora	Podczas nieobecności administratora możemy tylko poczekać aż znowu będzie dostępny i poinformować go o ewentualnych problemach typu brak łączności z serwerem, niepoprawnym wyświetlaniem zdjęcia na stronie, dziwnymi zachowaniami aplikacji.
6	Cel może nie zostać w pełni zrealizowany	Gdyby nastąpiła taka możliwość to po prostu ograniczamy funkcjonalność naszej aplikacji i staramy się dopracować elementy które mamy najlepiej rozwinięte

7	Zamknięcie uczelni z powodów epidemiologicznych	Przejsie na tryb całkowicie zdalny z wykorzystaniem platform takich jak m.in. MS Teams
8	Organizacyjne, zarządzenia i związane z czynnikiem ludzkim	Właściwa działalność informacyjna o pracach w projekcie, odciążenie członków zespołu którzy mają mniejszy bufor czasowy
9	Niewłaściwe wykorzystanie aplikacji przez użytkowników w celach destrukcyjnego kontaktu z innymi	Wychwytywanie podejrzenia działających użytkowników i zawieszenie lub całkowita blokada ich kont.
10	Wyciek danych	Nie będziemy podejmować działań w tym przypadku, jeśli taka sytuacja wystąpi postaramy się dociec jak do tego doszło
11	Zmiany wprowadzone w aktualizacjach programów służących do rozwijania aplikacji	Ustalamy jakich wersji oprogramowania będziemy używać i nie zmienianie jej. Gdy pojawią się dodatkowe funkcje, które mogłyby się przydać w łatwiejszym rozwijaniu aplikacji, to przedyskutujemy to w zespole czy aktualizujemy do danej wersji.
12	Natłok innych obowiązków na uczelni - kolokwia, projekty, sprawozdania	Podzielimy zadania w inny sposób, aby odciążyć członka zespołu, w ostateczności zrezygnujemy z niektórych funkcjonalności

Lista Narzędzi

a. Github

Serwis wykorzystywany jest przez nas do przechowywania repozytorium z kodem, wykorzystujący system kontroli wersji git, umożliwia nam w łatwy sposób śledzenie zmian w kodzie jak również zdalną współpracę nad projektem, stanowi również kopię zapasową względem lokalnych wersji projektu.

b. IntelliJ Idea

Zintegrowane środowisko programistyczne wykorzystywane przez nas do tworzenia warstwy logiki biznesowej naszej aplikacji.

c. WebStorm

Zintegrowane środowisko programistyczne wykorzystywane przez nas do tworzenia warstwy frontendu naszej aplikacji.

d. Discord

Aplikacja wykorzystywana przez nas do komunikacji w formie tekstowej jak również wideo rozmów.

e. MySQL Workbench

Narzędzie wykorzystywane przez nas do projektu bazy danej.

f. Docker

Narzędzie wykorzystywane do tworzenia kontenera z serwerem bazy danych.

g. Google Chrome

Przeglądarka wykorzystywana do testowania funkcjonalności aplikacji.

h. Postman

Aplikacja wykorzystywana do testowania stworzonych przez nas endpointów.

i. SonarQube

Platforma do ciągłej kontroli jakości kodu w celu wykonywania automatycznych przeglądów ze statyczną analizą kodu. Pozwala nam na wykrycie błędów, zapachów kodu i luk w zabezpieczeniach.

Licencja

Licencja MIT

Najprostsza z licencji. Wymaga zachowania informacji o licencji danego kawałka kodu oraz zachowania informacji o prawach autorskich. Ktokolwiek wykorzysta kawałek kodu objęty tą licencją nie będzie mógł wymagać jakichkolwiek gwarancji od autora kodu. Licencja pozwala na wykorzystanie kodu w projektach prywatnych jak i komercyjnych. Pozwala na dokonywanie zmian w kodzie źródłowym i dalsze rozpowszechnianie kodu.

Zezwolenia:

- Użytek komercyjny,
- Użytek prywatny,
- Modyfikowanie kodu,
- Dalsze rozpowszechnianie.

Warunki:

- Zachowana informacja o licencji kodu,
- Zachowana informacja o autorstwie kodu.

Ograniczenia:

- Brak odpowiedzialności za skutki użycia kodu,
- Brak gwarancji na działanie kodu.