

Koncepcja wykonania systemu

System Tutor (Tutor.pl)

Mateusz Markiewicz

Bartosz Sobocki

User Stories

- ❖ Dodanie ogłoszenia o chęci udzielania korepetycji
 - tutor wchodzi w opcję dodawania nowego ogłoszenia
 - tutor uzupełnia formularz dodawania nowego ogłoszenia, wpisując informacje dotyczące dziedziny, terminu, ceny, lokalizacji
 - tutor potwierdza dodanie nowego ogłoszenia

- ❖ Rezerwacja spotkania z tutorem przez ucznia
 - uczeń otwiera aplikację *Tutor* lub *Tutor.pl*
 - przegląda listę ogłoszeń tutorów, ogłaszających swoje oferty oraz czas, w których mogą się one odbyć
 - wybiera ogłoszenie, które mu odpowiada
 - potwierdza zaproponowanie spotkania z tutorem w danym terminie
 - rezerwacja zostaje wysłana do zaakceptowania przez tutora
 - po zaakceptowaniu rezerwacji uczeń otrzymuje dane tutora potrzebne do kontaktu

- ❖ Akceptacja spotkania z uczniem przez tutora
 - tutor dostaje powiadomienie o nowej propozycji spotkania z uczniem, może je otworzyć od razu lub zrobić to później wybierając je z widoku terminarza
 - tutor widzi wszystkie informacje o zaproponowanym mu spotkaniu, czyli jego termin, dziedzinę korepetycji, cenę, lokalizację, informacje o uczniu (wszystkie informacje, które uczeń postanowił podać, np. płeć, wiek, szkołę)
 - tutor może zaakceptować lub odrzucić propozycję spotkania
 - po odrzuceniu do ucznia wysyłana jest informacja, że propozycja została odrzucona, spotkanie znika z terminarzy ucznia oraz tutora

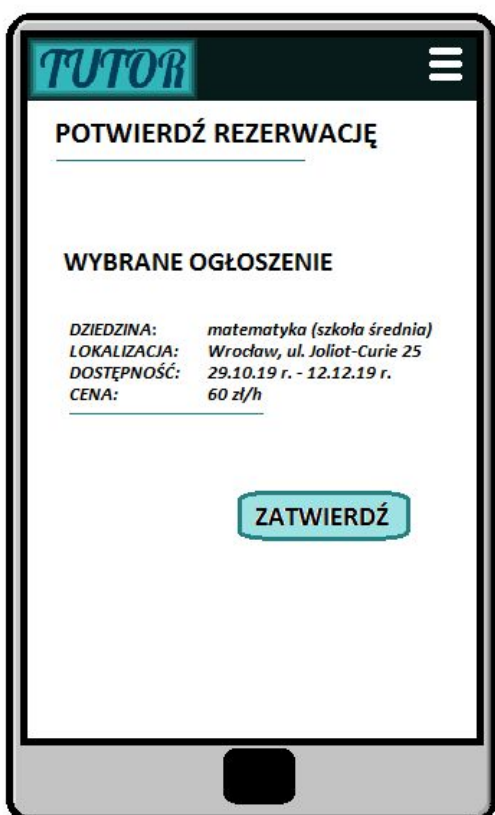
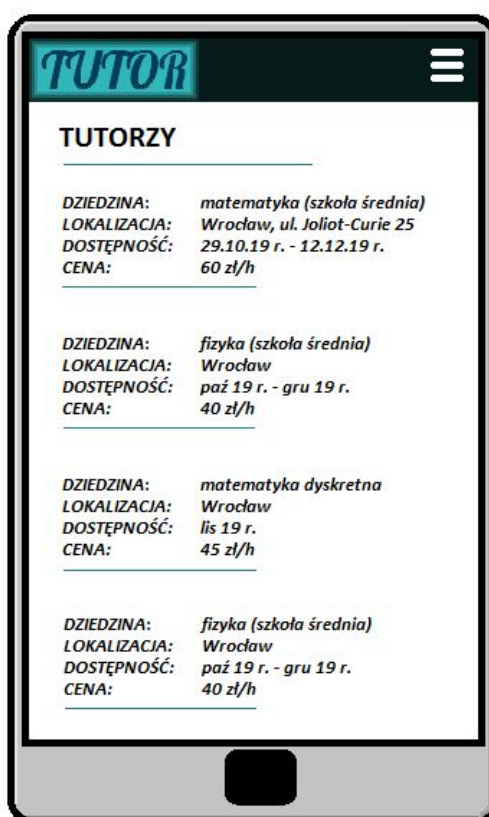
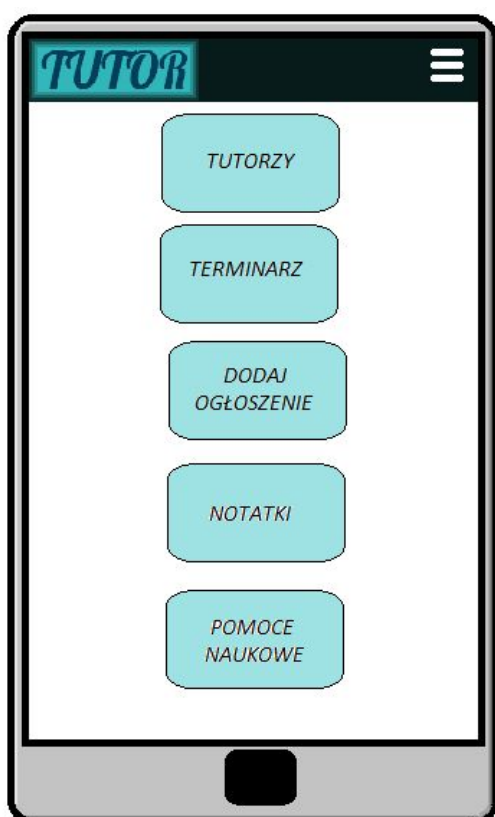
- po zaakceptowaniu uczeń również dostaje informację o zaakceptowaniu propozycji przez tutora, uczniowi wyświetlają się dane kontaktowe do tutora, zmienia się status spotkania w terminarzu ucznia i tutora
- wyświetlone zostaną dane kontaktowe do ucznia

3. Przykładowe ekrany aplikacji

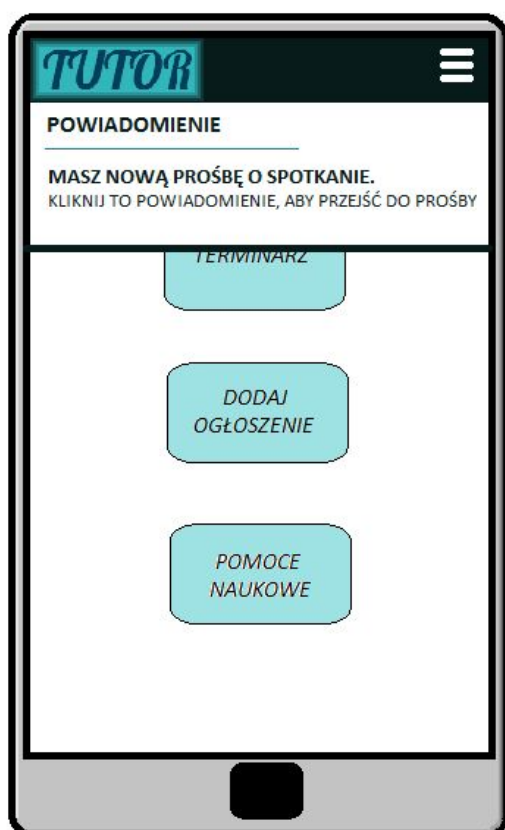
3.1)



3.2)



3.3)



4. Zaprojektowanie architektury

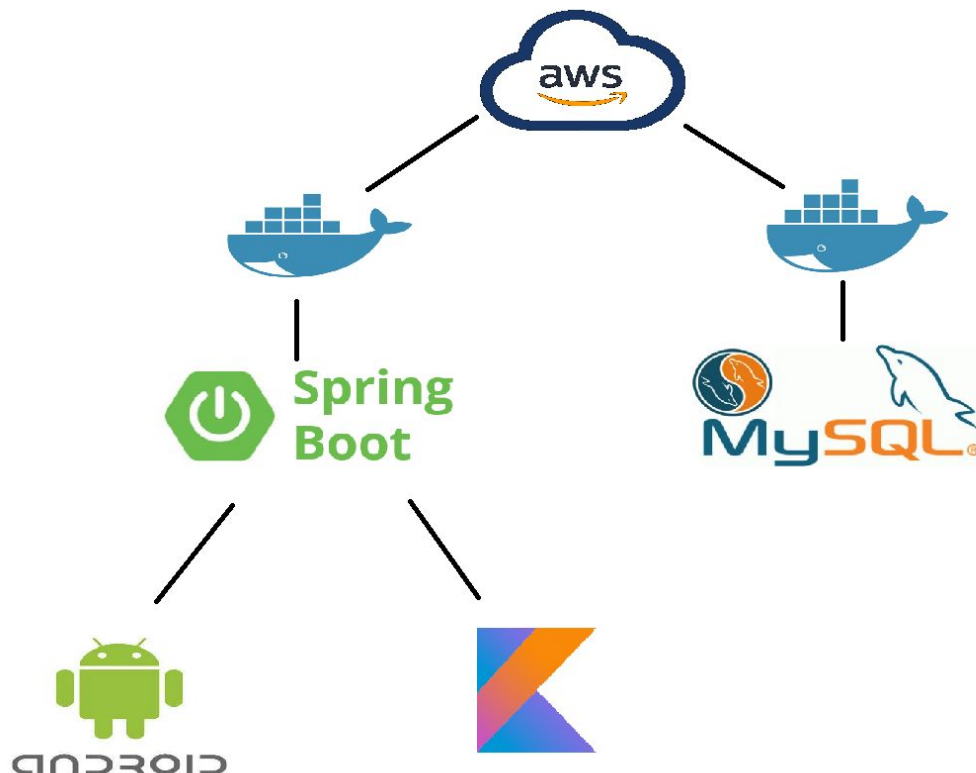
Aplikacja będzie posiadała wersję mobilną i webową. Wersja mobilna będzie dostępna na system operacyjny Android. Wersja webowa będzie obsługiwała następujące przeglądarki: IE9-11, Microsoft Edge, Google Chrome, Firefox, Safari, Opera.

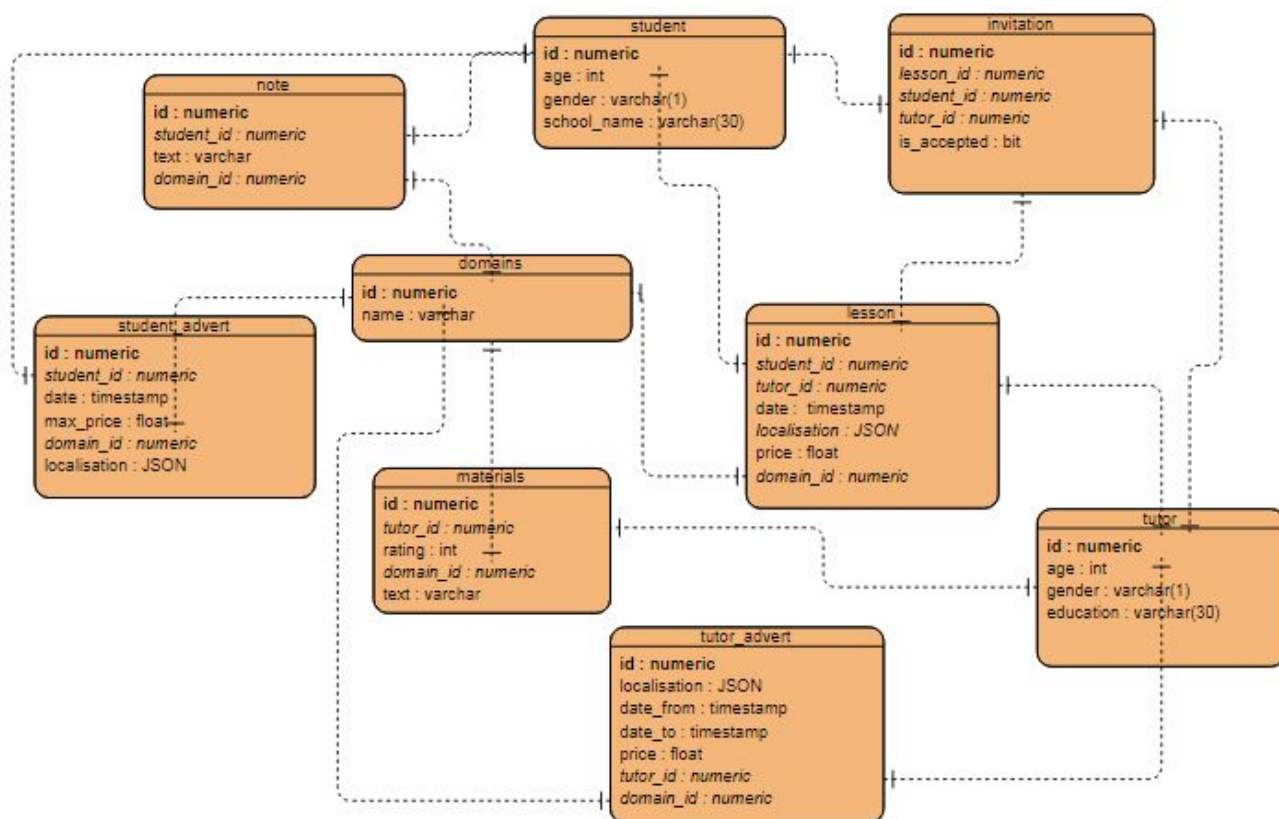
Interfejs będzie prosty i czytelny. Będzie zaprojektowany w sposób nowoczesny i minimalistyczny. Będzie posiadał dominujące kolory nawiązujące do kolorów z logo naszej aplikacji. Projekt interfejsu będzie spełniał aktualne najnowsze przyjęte standardy.

Modelu koncepcyjnego rzeczywistości:

Uczeń	Tutor	Notatka	Pomoc_naukowa	Spotkanie	Zaproszenie
ID Wiek Płeć Szkoła	ID Wiek Płeć Wykształcenie	ID ID_uczen tekst	ID ID_tutor ocena dziedzina tekst	ID ID_uczen ID_tutor termin lokalizacja cena dziedzina	ID ID_spotkanie Od_ID Do_ID zaakceptowane

Powiązania między podstawowymi elementami. Użyjemy chmury Amazon Web Services, za pomocą której postawimy dockery dla serwera oraz bazy danych. Użyjemy bazy MySQL.





5. Przedstawienie głównych zasad kodowania

Wersja mobilna aplikacji zostanie zaprogramowana w języku Java za pomocą Android Studio.

Wersja webowa aplikacji zostanie zaprogramowana w języku Kotlin. Strona serwerowa zostanie zaprogramowana za pomocą Spring Boot. Frontend zostanie stworzony za pomocą frameworka KVision. Jako IDE użyjemy IntelliJ IDEA. Będziemy przestrzegać oficjalnych standardów kodowania w Kotlinie.

Będziemy wykorzystywali repozytorium na GitHubie, by programiście w łatwy sposób mogli wymieniać się kodem, panować nad wersją.

Będziemy regularnie wykonywać code review, by upewnić się, że zachowane są wszystkie standardy wykonywania oprogramowania.

6. Identyfikacja ryzyka i zasady jego zarządzaniem

Stojące przed nami ryzyka możemy podzielić na dwie grupy - biznesowe i techniczne.

Ryzyka biznesowe to:

1. Brak zainteresowania ze strony uczniów - w przypadku wystąpienia można rozpocząć kampanię reklamową w szkołach, lub serwisach popularnych wśród uczniów, np YouTube.com. Można oferować pokrycie części kosztów pierwszych kilka korepetycji.
2. Brak zainteresowania ze strony tutorów - w przypadku wystąpienia można rozpocząć kampanię reklamową wśród studentów oraz nauczycieli, można również oferować dopłatę do przeprowadzonych kilku pierwszych korepetycji
3. Zły model płatności - w przypadku wystąpienia można próbować zbalansować liczbę darmowych miesięcznych propozycji spotkań oraz cenę zakupu dodatkowych. Można również położyć większy nacisk na reklamy wyświetlane w niektórych momentach pracy aplikacji, bądź w niektórych miejscach strony webowej.

Ryzyka techniczne to:

1. Mało czytelny interfejs - by zapobiec takiej sytuacji zatrudnimy grafików mających doświadczenie w projektowaniu aplikacji webowych i mobilnych
2. Mała skuteczność algorytmu dopasowującego uczniów i tutorów - by zapobiec tej sytuacji zatrudnimy specjalistę z data science, potrafiącego skonstruować odpowiedni model
3. Zbyt mała responsywność aplikacji - w przypadku wystąpienia można przenieść się na wydajniejszy serwer

7. Zmiany względem tablicy koncepcyjnej

Postanowiliśmy zrezygnować z przeglądania ogłoszeń za pomocą mapy. Mapa nie będzie częścią aplikacji, użytkownik precyzuje lokalizację za pomocą adresu, a zadaniem serwera jest wyznaczenie odległości pomiędzy dwoma podanymi lokalizacjami. Naszym zdaniem nie zmniejsza to znacząco funkcjonalności naszej aplikacji, a ogranicza problemy, które możemy napotkać.

Postanowiliśmy również zatrudnić na pewien czas osobę odpowiedzialną za stworzenie algorytmu dopasowywania uczniów i tutorów.