Dokumentacja projektu

Temat projektu: Realizacja aplikacji internetowej umożliwiającej tworzenie własnego bloga
Przedmiot: Przetwarzanie danych w chmurach obliczeniowych
Autor: Stanisław Tęczyński, WFiIS, Informatyka Stosowana, 2020/21, semestr 7

1. Wstęp

Celem projektu było utworzenie aplikacji internetowej, na której można utworzyć własne konto i dodawać wpisy, które można udostępnić innym do przeczytania. Poniżej prezentuję przypadki użycia:

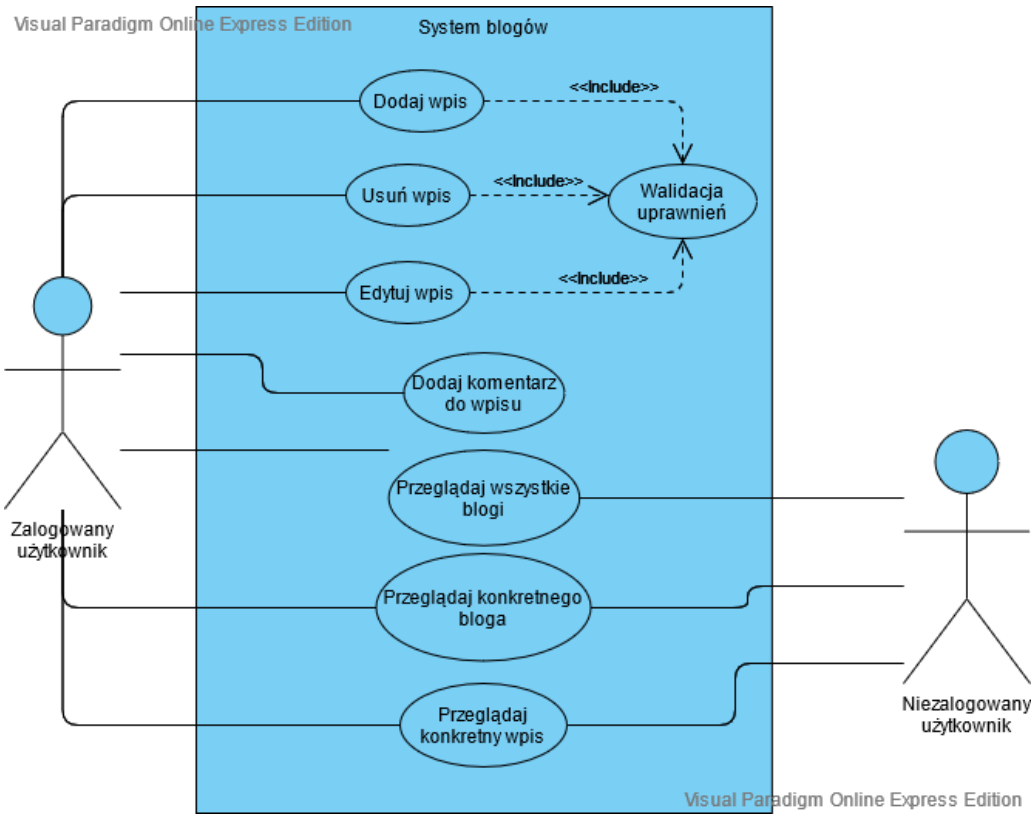
- Jako użytkownik chcę mieć możliwość logowania i rejestracji, żeby utworzyć własnego, spersonalizowanego i publicznego bloga
- Jako użytkownik chcę mieć możliwość dodawania wpisów do własnego bloga, żeby udostępniać je ludziom i dzielić się swoim życiem
- Jako użytkownik chcę mieć możliwość formatowania wpisu w określony sposób podczas jego dodawania (kursywa, pogrubienie, wcięcia), aby poprawić wrażenia wizualne podczas czytania publikacji
- Jako użytkownik chcę mieć możliwość oznaczenia wpisu jako opublikowany lub nie, aby nie wyświetlać jeszcze niedokończonych wpisów
- Jako niezalogowany użytkownik chcę mieć możliwość przeglądania blogów dostępnych na stronie, aby dowiedzieć się, co dzieje się w ich życiu
- Jako niezalogowany użytkownik chcę mieć możliwość przeglądania wszystkich wpisów w konkretnym blogu
- Jako zalogowany użytkownik chcę mieć możliwość komentowania wpisów na blogach innych ludzi, aby dzielić się swoimi opiniami

2. Technologie

- Backend: Node.js (RESTful, połączenie za pomocą AJAX)
- Frontend: React.js + biblioteka graficzna AntDesign
- Baza danych: Neo4J
- Hosting: AWS - backend: Elastic Beanstalk, frontend: AWS Amplify, certyfikat SSL dodany przez Load Balancer, baza danych postawiona na wydziałowym taurusie

3. Diagram UML

Poniżej prezentuję diagram przypadków użycia w mojej aplikacji (przygotowany z wykorzystaniem narzędzia Visual Padadigm Online).



4. Opis wdrożenia

Wdrożenie za pomocą serwisów AWS-u nie było szczególnie skomplikowane. Wykorzystałem darmowy okres próbny Amazonu AWS Free Tier na okres roku, co pozwala na darmowe hostowania aplikacji i/lub bazy danych.

W przypadku hostowania backendu skorzystałem z serwisu *Elastic Beanstalk*. Utworzyłem tam aplikację na platformie *Node.js 12 running on 64bit Amazon Linux 2*, stworzyłem paczkę .zip z moją aplikacją i przekopiowałem na serwery Amazonu. Trzeba było również wybrać strefę dostępności – w moim przypadku było to *us-east-2c*. Dodatkowo trzeba było utworzyć klucz *EC2 key pair* i uwzględnić go w konfiguracji aplikacji.

Natomiast w przypadku hostowania frontendu, skorzystałem z serwisu *AWS Amplify*. Aplikację typu React najłatwiej udostępnić poprzez przypięcie repozytorium na jednym z serwisów git – w moim przypadku był to gitlab. Mając gotowe repozytorium, przy tworzeniu aplikacji AWS Amplify wybrać odpowiednie repozytorium i branch z repozytorium, a serwis zajmie się resztą. Następnie, po każdej zmianie na wybranym branchu, aplikacja zostanie zbudowana na nowo.

Baza danych została udostępniona na serwerach uczelnianych, zatem nie była wymagana dodatkowa konfiguracja. Z racji braku ścisłej separacji schematów pomiędzy użytkownikami, nazwy labeli poprzedziłem swoim nazwiskiem, aby nie

pokrywały się z labelami innych użytkowników Czyli zamiast *User, Article, Comment* mam odpowiednio *TeczynskiUser, TeczynskiArticle, TeczynskiComment*.

Aby aplikacja backendowa była również dostępna poprzez protokół https, utworzyłem również certyfikat SSL(korzystając z tego poradnika: https://www.genesesolution.com/create-a-self-signed-ssl-certificate-and-upload-on-aws-account/) i utworzyłem Load Balancer na stronie AWS-u, wybierając właśnie ten certyfikat. Nie jest to rozwiązanie idealne, ponieważ pomimo że protokół jest https, to jednak certyfikat nie jest nigdzie potwierdzony, a więc użytkownik musi potwierdzić, że chce skorzystać z niebezpiecznej strony.

5. Linki

Link do aplikacji: https://master.d21w5jp2jd697d.amplifyapp.com/#/blogs
Link do api: https://blog-814875059.us-east-2.elb.amazonaws.com/users (trzeba potwierdzić, że chcemy skorzystać z niebezpiecznego połączenia)