

Chciałbym przedstawić państwu swoją pracę inżynierską której tematem jest stworzenie aplikacji internetowej umożliwiającej wyszukiwanie ofert wynajmu oraz sprzedaży mieszkań i nieruchomości. W dzisiejszych czasach rynek nieruchomości w Polsce zaczął się bardzo szybko rozwijać, zwłaszcza w dużych miastach. Przyczyną takiego stanu jest zapewne duża ilość studentów w ośrodkach akademickich, które znajdują się zazwyczaj w wielkich ośrodkach miejskich. Jako student wiele czasu poświęciłem na szukanie odpowiedniej oferty wynajmu mieszkania i dogłębnie poznałem dobre i złe strony serwisów udostępniających ogłoszenia. Mając na uwadze problemy ze znalezieniem idealnej oferty w danej lokalizacji, w mojej głowie narodził się pomysł by stworzyć aplikację internetową która nie tylko będzie umożliwiała wyszukanie ofert, lecz również będzie proponowała oferty użytkownikowi bazując na jego preferencjach.

Celem pracy było stworzenie aplikacji internetowej która umożliwi znalezienie oferty wynajęcia lub kupna mieszkania, domu bądź też pokoju. Tworzona aplikacja ma również umożliwiać zamieszczenie oferty wynajmu lub sprzedaży. Moim głównym zadaniem będzie zaimplementowanie zupełnie nowego sposobu pokazywania ofert oraz personalizowanie ofert dla użytkownika. Jednym z celów jest również stworzenie interfejsu przyjaznego użytkownikowi, który byłby intuicyjny i prosty w użyciu zachowując przy tym wszystkie najważniejsze funkcjonalności. Dużą wagę przywiązałem również do zabezpieczenia dostępu do aplikacji przed osobami do tego niepowołanymi.

Do zaimplementowania aplikacji użyłem frameworka Ruby on Rails. Jest to framework korzystający z języka ruby powstały w roku 2005. Nie jest to zatem najnowsza technologia lecz cały czas prężnie rozwijana. Ruby natomiast powstał w roku 1995 i jest językiem interpretowanym, w pełni obiektywnym z dynamicznym typowaniem. Jest to framework wykorzystywany głównie do tworzenia niezbyt wielkich serwisów internetowych ze względu na gorszą wydajność od Javy czy C#. Jego wydajność, skalowalność jednak jest na tyle dobra, że rozpowszechniła Ruby on Rails w sektorze małych i średnich aplikacji. W Ruby on Rails duży nacisk jest kładziony na szybkość, łatwość i przyjemność pisania kodu, oraz na regułę "Don't repeat yourself". Istnieje również reguła Convention over Configuration, polegająca na sprowadzeniu do minimum niezbędnej konfiguracji przez zastępowanie jej gotowymi, domyślnymi, zalecanymi wzorcami. Ruby on Rails jest udostępniany na zasadzie open source.

Przy implementacji widoków zdecydowałem się skorzystać z map dostarczanych przez Google. Jest to rozwiązanie z którego można korzystać bezpłatnie pod warunkiem, że strona na której są wykorzystywane mapy nie pobiera opłat od użytkowników.

Żeby skorzystać z API należy dołączyć do strony internetowej skrypt udostępniany przez Google.

Google Maps API dostarcza wiele udogodnień takich jak wyświetlanie znaczników na mapie, informacji o lokalizacji.

Istnieje również możliwość wyświetlania aktualnego natężenia ruchu.

Funkcjonalności które będę wykorzystywał w swojej aplikacji to możliwość manipulacji mapą, dodawanie znaczników oraz zaznaczanie obszaru na mapie. Dużą zaletą korzystania

z tych map jest jej popularność w sieci, użytkownicy są przyzwyczajeni do wyglądu oraz obsługi tych map przez co mój serwis będzie od samego początku wyglądał znajomo a poruszanie się w nim będzie intuicyjne.

Architektura aplikacji jest typowa dla frameworku Ruby on Rails, żądania od klienta są kierowane do serwera aplikacyjnego jakim jest w moim przypadku WEBrick. Jest to serwer który jest idealny podczas tworzenia aplikacji gdyż jest bardzo szybki. Jego minusem jest jednak niezbyt wydajna obsługa wielu równoczesnych połączeń, dlatego w roli serwera dla użytkowników używa się innych rozwiązań takich jak Unicorn, Puma czy Passenger. Serwer kieruje żądanie użytkowników do dyspozytora który przekierowuje żądanie do odpowiedniego kontrolera. Kontroler korzystając z mechanizmu ActiveRecord, który jest odpowiedzialny za mapowanie obiektów z bazą danych otrzymuje konkretne dane. Następnie kontroler może przekierować żądanie oraz dane do innego kontrolera lub wysłać odpowiedź do użytkownika poprzez wyrenderowanie odpowiedzi w przeglądarce klienta lub poprzez moduł ActionMailer, który jest odpowiedzialny za wysyłanie wiadomości mailowych. Zadania które mają się wykonywać w tle trafiają do modułu ActiveJob, który jest odpowiedzialny za odpowiednie zakolejkowanie zadań i wywołanie ich w odpowiednim czasie.

Aplikacja została wykonana w konwencji Single Page Application, konwencja ta polega na wyświetlaniu odpowiedzi użytkownikowi bez konieczności przeładowywania strony. Jest to możliwe do osiągnięcia poprzez wysyłanie żądań oraz odbierania żądań w skryptach javascriptowych poprzez bibliotekę AJAX. Dzięki zastosowaniu tego mechanizmu strona znacząco podnosi swoją responsywność i sprawia wrażenie bardziej płynnej w działaniu. Po lewej stronie ekranu można dojrzeć pasek który po naciśnięciu się wysuwa z lewej strony listę filtrów ale to pokażę w następnym slajdzie. Na górze ekranu po prawej stronie widoczne są dwa przyciski służące do rejestracji i logowania. Po naciśnięciu jednego z tych przycisków, na stronie renderowane jest dodatkowe okno, które umożliwia zalogowanie lub rejestrację. Na mapie są widoczne znaczniki, każdy znacznik odpowiada ogłoszeniu.

Po zalogowaniu się, można dostrzec, że lista filtrów automatycznie wysunęła się lewej strony oraz zostały zmienione przyciski. Wśród filtrów możemy zaobserwować suwaki do określenia przedziałów paramterów nieruchomości które interesują użytkownika np. wielkość mieszkania, cena, ilość pokoi. Znajdują się także parametry do wyboru z listy takie jak typ nieruchomości albo czy to jest dom czy mieszkanie lub pokój, rodzaj ogłoszenia, czy to jest sprzedaż czy wynajem. Można też zauważyć tutaj cheboxy dzięki który możemy zaznaczyć czy np. dana oferta jest skierowana do ludzi palących lub posiadających zwierzęta czy też nie. Na samym dole znajduje się ikonka znacznika którym można wybrać interesujący nas punkt na mapie oraz suwak którym można ustalić jak wielki obszar nas interesuje. Po wybraniu zasięgu oferty zostaną okrojone tylko do tych które leżą w środku tego okręgu. Wśród przycisków można zauważyć te do edytowania profilu, wylogowania się, dodania nowego ogłoszenia oraz wyświetlenie obserwowanych ogłoszeń. Dodanie nowego ogłoszenia wiąże się z wypełnieniem formularza który jest walidowany pod kątem poprawności danych w celu uniemożliwienia dodawania zbyt ogólnych ogłoszeń. Podczas przeglądania ogłoszeń istnieje możliwość dodania ogłoszenia do obserwowanych w celu ich szybszego odszukania.

Istotnym elementem aplikacji jest zapisanie filtrów przez użytkownika. Na podstawie tych danych system jest w stanie przefiltrować nowo dodane ogłoszenia pod kątem zgodności z kryteriami wyznaczonymi przez użytkownika. Jeżeli dana oferta odpowiada preferencjom danego użytkownika jest on o niej powiadamiany poprzez maila. Aby uniknąć sytuacji, że po dodaniu każdej oferty jest sprawdzane czy odpowiada ona każdemu użytkownikowi, został zaimplementowany proces w tle który sprawdza raz dziennie dla każdego użytkownika czy oferty dodane z ostatniego dnia są dla niego interesujące.

Podsumowując, udało mi się zrealizować wszystkie założenia które zdefiniowałem w fazie analizy aplikacji. Podczas procesu tworzenia aplikacji nasunęły się na myśl dodatkowe rozszerzenia, których nie uwzględniłem w planach. Podczas dalszego rozwoju aplikacji możnaby uwzględnić kwestię wymiany wiadomości między zainteresowanymi a ogłaszającymi. Dodatkowo byłaby możliwość wprowadzanie opłat za wyróżnienie ofert, byłaby to propozycja skierowana do osób którym zależy na tym by jak najszybciej wynająć mieszkanie. Następnym rozszerzeniem mogłaby być możliwość pobierania danych z innych serwisów na przykład Facebook na którym istnieją specjalne grupy służące do zamieszczania ogłoszeń. W ramach testowania aplikacji w środowisku docelowym, umieściłem ją na darmowym serwerze o nazwie Heroku. Podczas testów do których wykorzystałem kilka kont nie stwierdziłem żadnych odstępstw od zamierzonego działania serwera.

Dziękuję za uwagę