**6.Opis aplikacji**

Zdefiniowanie wymagań funkcjonalnych oraz diagramów opisujących architekturę systemu, pozwala na rozpoczęcie implementacji aplikacji. Tworzenie kodu opierającego się o wcześniej stworzony projekt, pozwala na uniknięcie dużej ilości błędów oraz stworzenia optymalnego rozwiązania. Spójna wizja jest bardzo pomocna w pisaniu kodu dobrze poukładanego oraz pozwalającego na przyszłe rozszerzenie funkcjonalności, dzięki wzorcom projektowym, zastosowanym w implementacji. By aplikacja mogła dobrze działać, potrzeba stworzyć środowisko wykonawcze, łatwe do zarządzania oraz starające się działać z jak najmniejszą ilością konfiguracji, przeprowadzonej przez administratora, potrzebnej do uruchomienia danego systemu. Bardzo ważnym punktem, z perspektywy użytkownika, jest nie tylko dobrze działająca aplikacja, ale interfejs, pozwalający na intuicyjne korzystanie z funkcjonalności. W tym celu, w projekcie aplikacji, został stworzony prototyp interfejsu, na podstawie którego, zaimplementowane zostały widoki w warstwie klienckiej. Ostatnim elementem opisanym w tym rozdziale są testy stworzone i przeprowadzone w ramach sprawdzania działania systemu. Istnieją różne rodzaje testów, które są istotnym czynnikiem wpływającym na kontrolę nad błędami pojawiającymi się w trakcie działania. Testy jednostkowe z łatwością mogą nam pokazać miejsce, w którym zmiana mogła spowodować błąd, natomiast testy akceptacyjne, przeprowadzone w formie testów interfejsowych, sprawdzają główne funkcjonalności systemu. Rozdział ten opisuje wszystkie, wyżej wymienione fazy tworzenia aplikacji.

**6.1. Środowisko wykonawcze**

Biorąc pod uwagę, że aplikacja tworzona w ramach tej pracy, jest to aplikacja webowa, jej środowisko wykonawcze można podzielić na dwie grupy – środowisko wykonawcze po stronie klienta oraz serwera.

Środowiskiem wykonawczym po stronie klienta dla aplikacji webowej, jest dowolna przeglądarka internetowa, posiadająca silnik, uruchamiający kod pisany w Javascripcie. Najpopularniejszymi przeglądarkami są Mozilla, Chrome, Internet Explorer oraz Safari. Ważnym aspektem dobrze działającej aplikacji webowej, jest możliwość uruchomienia jej na każdej z tych przeglądarek. Różnice jakie można napotkać w interpretowaniu kodu przeznaczonego dla warstwy klienckiej, może tworzyć problem w napisaniu rozwiązania działającego na wszystkich dostępnych interpreterach. Jednym z dużych plusów użycia frameworka Javascriptowego jest zapewnienie, że ten problem nie wystąpi dla tworzonej aplikacji. Angular, który został użyty przy implementacji, zapewnia spójny wygląd oraz prawidłowe działanie funkcjonalności na każdej z wyżej wymienionych przeglądarek.

Środowisko wykonawcze, którego wybranie jest zadaniem dla osoby tworzącej aplikację, jest to środowisko po stronie serwera. Biorąc pod uwagę silne połączenie serwerów z wybraną technologią w jakiej napisana jest warstwa serwerowa, wybór narzędzia jest dokonany wcześniej. Implementacja serwera używając platformy .NET, kieruje do użycia serwera IIS w celu hostowania tworzonej aplikacji webowej.

IIS jest to serwer webowy stworzony przez firmę Microsoft, działający na systemach Windows, akceptujący zdalne żądania klientów, wysyłając do nich odpowiednią treść odpowiedzi. Odpowiedzią na żądanie klienta może być statyczna strona HTML, tekst, obrazki oraz inne zasoby dostępne w internecie. Jednakże IIS dostarcza mechanizmy nie tylko generujące odpowiednie odpowiedzi, ale także zapewniające bezpieczeństwo oraz optymalne działanie serwera. Biorąc pod uwagę, że każde żądanie jest obsłużone przez IIS, administrator jest w stanie sprawdzić tożsamość klienta oraz jego uprawnienia do zasobów, wymienionych w żądaniu. W celu ułatwienia zarządzania serwerem, wprowadzone zostały pojęcia puli aplikacji oraz procesu wykonawczego. Proces wykonawczy odpowiada za wygenerowaniu wszystkich żądań i odpowiedzi pochodzących z serwera, będąc głównym punktem działania aplikacji serwowanej przez IIS. Pula aplikacji jest to zbiór procesów wykonawczych, pozwalających na organizacje oraz łatwiejsze przypisanie uprawnień dla danego systemu działającego w ramach IIS’a.

Ostatnim elementem, wchodzącym w skład środowiska wykonawczego, jest Docker. Docker jest to open-sourcowy projekt, stworzony w 2014 roku. Pozwala on na stworzenia środowiska uruchomieniowego dla aplikacji, jako przenośne kontenery, posiadające całą konfigurację zdefiniowaną wewnątrz. Kontenery stworzone za pomocą Dockera, uruchomione na różnych maszynach, działają w ten sam sposób. W aplikacji tworzonej w ramach tej pracy, Docker został użyty do uruchomienia serwera bazodanowego oraz stworzenia architektury Lambda za pomocą tego serwera oraz Sparka. Dzięki takiemu podejściu, możliwe jest użycie stworzonej architektury bazodanowej dla różnych aplikacji oraz na różnych maszynach, nie potrzebując zbędnej konfiguracji.

**6.2. Najważniejsze rozwiązane problemy**

**6.3. Manual**

**6.4. Testy jednostkowe, akceptacyjne oraz środowiskowe**