Implementacji założen projektowych podjeto się z chwilą decyzji w sprawie wyboru technologii back-end. Wcześniej uległo też zakończeniu przygotowanie wymagań funkcjonalnych całego systemu, które były najważniejszym zbiorem warunków, jakim powinien sprostać gotowy program.

Wybór frameworka Codeigniter 4 do oprogramowania strony serwerowej, wymagał jednak przygotowania teoretycznego, z uwagi na brak doświadczenia autora w pracy z frameworkami w ogóle, a z modelem architektonicznym MVC (Model View Controller) w szczególności.

Poznawanie tej technologii odbywało się równolegle do pracy programistycznej nad procesem autoryzacji uzytkownika docelowego systemu. Na tym etapie strona kliencka aplikacji istniała tylko jako widoki tworzone w jezyku html, wywoływane bezpośrednio z odpowiednich kontrolerów frameworka. Zgodnie z początkowymi planami projektowymi, obsługa dynamicznych zachowań gotowej strony internetowej, miała opierać się o zastosowanie czystego języka JavaScript, tzw Vanilla JS, co w zamyśle autora, miało pozwolić na łatwiejsze, z zatem szybsze prowadzenie prac. Takie założenie, zdecydowało o zastosowaniu mechanizmu sesji do autoryzacji poczynań użytkownika w aplikacji internetowej Repeater.

Z chwilą zakończenia prac nad systemem rejestracji, logowania i odzyskiwania hasła dostępowego uzytkownika, rozpoczęto przygotowanie do projektowania elementów dynamicznych poszczególnych widoków. Pierwsze rozważania teoretyczne, postawiły jednak pod znakiem zapytania możliwość panowania nad organizacją kodu, przy zachowaniu pierwotnego założenia stosowania czystego JavaScript. Stosowane dotychczas w małych projektach laboratoryjnych, podobne rozwiązanie, już wtedy rodziło szereg trudności, głównie polegających na nadmiernym wzroście objętości programów.

Autor w związku z powyższym podjął decyzję o zmmianie podejścia projektowego do strony klienckiej aplikacji i oparcia jej tworzenia o gotowy framework, którego wyboru dokonano zgodnie z przyjętymi priorytetami i pomocą platformy wielokryterialnego wspomagania decyzyjnego, Promethee. Wybór technologii Vue.js 3, dawał możliwość elastycznego stosowania go w projekcie, począwsszy od prostego importowania frameworka z zewnętrznego repozutorium, aż po zastosowanie wzbogaconej o dodatkowe funkcjonalności, wersji instalacyjnej (CLI). Przewidywania autora, dotyczące konieczności posiadania kontrolera stanu zmiennych po stronie front-end, skłoniły go do użycia pełnej wersji Vue.js, posiadającej możliwość użycia Vuex, takiego właśnie managera stanu aplikacji.

Konsekwencją dokonanego wyboru, była zmiana zposobu autoryzacji użytkownika po stronie serwera z pierwotnie przygotowanej, sesyjnej, na taką, która pozwoliłaby uwiarygadniać stronę klienta, opartą na bezstannowym protokole HTTP, podczas obsugi wielokrotnych żądań wysyłanych do bazy danych.

Wybrano metodę JWT (JSON Web Token), która polega na każdorazowym dołączaniu do wysyłanego żądania na serwer, niepowtarzalnego, przygotowanego przez tenże serwer (od razu po zalogowaniu użytkownika) tokena. Na jego podstawie, każde działanie oparte o interakcję z bazą danych, jest uwierzytelnianie osobno. Szczegółowo sposób implementacji tegoż, został opisany w rozdziale poświeconym aspektom bezpieczeństwa projektowanego systemu.

Post factum, można przyznać, że obie główne decyzje dotyczące oparciu implementacji back-end o framework Codeigniter, oraz front-end o Vue CLI, okazały się z perspektywy czasu bardzo dobre, choć związane były z podjęciem dużego wysiłku w celu ich poznania.

Pierwsza z nich, ponieważ stopień trudności opanowania Codeigniter’a nie okazał się ryzykownie duży, a funkcjonalności frameworka w pełni pokryły wymagania stawiane mu przez projekt. Druga, dotycząca wyboru Vue.js do obsługi strony wizualnej, ponieważ tylko dzięki użyciu funkcji oferowanych przez jego pełną wersję CLI, udało się stworzyć uporządkowany kod, realizujący pierwotne założenia reaktywności strony internetowej Repeater.

Ostatnim etapem przed zainstalowaniem gotowej platformy na serwerze, było zastosowania stylowania wizualnego elementów poszczególnych widoków – podstron, za pomocą CSS – kaskadowych arkuszy stylów, w celu uatrakcyjnienia pracy z aplikacją i podkreślenia ważności poszsczególnych elementów podstron.

Ostatecznie, uzyskano produkt, którego funkcjonalności odpowiadają założeniom przedstawionym w dokumentacji wymagań, zawartej w załączniku A do niniejszej pracy.

Sumaryczny czas realizacji projektu od strony praktycznej, czyli jego implementacja, wliczając w to również czas poświecony na poznanie zastosowanych technologii, wyniósł blisko 2,5 miesiąca.