\section{Odjemalec}

Aplikacijo smo napisali v programskem jeziku JavaScript in Vue Frameworku. Gre za moderen, odziven in reaktiven vmesnik, ki deluje v realnem času. Vue je eden izmed mnogih kot npr. Angular, Ember, React,… poznan pa je predvsem zaradi enostavnosti za upravljanje in izvajanje testov. Vsem je skupna točka reaktivnost, vendar v drugačnem pomenu besede. Gre za to, da se aplikacija postopno prilagaja glede na vrednosti podatkov. To prednost smo izkoristil in dodali še knjižnico Vuex, ki služi kot centralizirana baza podatkov za vse komponente v aplikaciji.

Aplikaciji so dve in predstavljajo tri poglede. Ena aplikacija je namenjena samo gostu, druga pa natakarju in kuharju. Pri drugi aplikaciji ločimo poglede glede na prijavo, ki jo more izvesti uporabnik pred uporabo.

Da bi to dosegli ne bi potreboali JavaScript zaradi RESTful arhitekutre, ki omogoča katerokoli platformo na strani odjemalca.

\subsection {Kaj je reaktivnost?}

Reaktivnost \cite{reaktivnost}, je programska paradigma, ki nam omogoča, da se na deklarativni način prilagodimo spremembam. Dober primer reaktivnosti je npr. funkcija SUM, ki jo uporabljamo v Excelu. Slika~\ref{Excel-SUM} prikazuje primer v Excelu.

\begin{figure}[h]

\begin{center}

\includegraphics[width=7cm]{Excel-SUM.png}

\end{center}

\caption{Primer fukcije SUM v programu Excel}

\label{Excel-SUM}

\end{figure}

Če vstavimo številko 2 v prvo celico in številko 3 v drugo celico ter izberemo funkcijo SUM teh dveh celic. Kot rezultat dobimo vsoto obeh številk skupaj, kar ni nič posebnega. Vendar če bomo spremenili vrednost prve celice, bo funkcija SUM avtomatsko posodobila skupno vrednost. Tako deluje tudi reaktivnost v aplikacijah za razliko, da je podatek lahko vezan na več funkciji oziroma delov programske kode, ki se ob spremembi vrednosti posodobijo.

\subsection {Delovanje reaktivnost v Vue}

Vue se torej v primerjavi z navadnim JavaScriptom sprehodi skozi podatke in njihove lastnosti (angl. Properties) pretvori v fukciji Getter in Setter, ki sta nevidni uporabniku \cite{delovanjeReaktivnost}. Poglejte si sliko~\ref{VueReacitivity} za lažjo predstavo.

Torej funkcija Getter pokliče instanco Watcher z namenom odvisnosti do drugih komponent. To pomeni, če je podatek označen kot odvisen (angl. Dependency) bodo nekateri deli programske kode oziroma funkcije poklicane vsakič, ko se spremeni vrednost podatka. Funkcija Setter pa obvesti instanco Watcher, vsakič ko se podatku spremeni lastnost. Ta poskrbi, da se pokliče funkcija upodabljanja (angl. Render) tiste komponente, ki potem prikaže spremembe v samem pogledu aplikacije.

\begin{figure}[h]

\begin{center}

\includegraphics[width=13cm]{Vue-reactivity.jpg}

\end{center}

\caption{Kateri dialekt uporabljati?}

\label{VueReacitivity}

\end{figure}

\subsection {Knjižnice in dodatki}

Tako kot vsak programski jezik ima tudi Vue svoje dodatke, ki pomagajo pri razvoju aplikaciji. Spodnje smo uporabljali in so vredni omembe.

\begin{description}

\item[Vue CLI] velja kot standardno orodje za ekosistem Vue \cite{VueCLI}. Zagotavlja, da že pri gradnji novega projekta poveže različne dodatke med seboj. To omogoča razviljacu, da se bolj osredotoči na programiranje in ne na povezovanje njih v projekt. Zadeva izgleda nekako tako, da preko CLI vmesnika izbereš kakšen projekt želiš. Imaš seveda že privzete nastavitve, vendar omogoča tudi nastavljanje po meri. Sam sem uporabil Vuex, Vue-Router, ESLint in Vuetify.

\item[Vuex] je knjižnica za shranjevanje vrednosti v aplikacijah Vue.js \cite{Vuex}. Služi kot centralizirana baza podatkov za vse komponente v aplikaciji.

\item[Vue-Router] je uradni usmerjevalnik za Vue.js \cite{VueRouter}. Integrira se globoko z jedrom Vue.js, tako da poenostavi izdelavo SPA aplikacij. Usmerjevalki je mišljen v smislu usmerjanja na druge komponente (angl. Component), ki v Vue.js predstavljajo druge poglede, lahko bi rekli podobno kot podstrani.

\item[ESLint] je orodje za prepoznavanje in poročanje o popravkih v programski kodi \cite{ESLint}. Cilje je narediti kodo bolj pregledno in urejeno, kar pripomore k izogibanju napak.

\item[Vuetify] je eden izmed mnogih uporabniških vmesnikov, ki je zgrajen na vrhu Vue.js \cite{Vuetify}. Za razliko od drugih vmesnikov je Vuetiy enostaven za učenje z več stotimi komponentami izdelanih po specifikacijah Material Design.

\item[Vue-devtools] je zgolj dodatek v brskalniku, ki omogoča lažje sledenje delovanja aplikacije in odpravljanju napak.

\end{description}