```
        }
    }
};
```

Sl.6d Funkcija za kreiranje i prikaz grafikona

Primetimo da za instancu EventSource interfejsa vežemo tri osuškivača događaja kao što je prikazano na slici 7.

```
source.addEventListener('message', function (e){
     // code }, false);
source.addEventListener('open', function (e) {
     // code}, false);
source.addEventListener('error', function (e) {
     // code }, false);
```

Sl.7. Osluškivači događaja

U trenutku kada dođe do promene akcije na serveru, aktivira se osluškivač događaja koji reaguje na poruke *("message")* i izvršava se metod koji se navodi kao njegov drugi argument, dok se u slučaju otvaranja konekcije aktivira metod koji se navodi kao drugi argument za osluškivač koji se aktivira na događaj otvaranja konekcije *("open").* Takođe, kao što se vidi, definisan je i osluškivač koji se aktivira u slučaju pojave greške. U okviru metoda koji se navodi kao drugi argument osluškivača na događaje tipa *"message"* navodi se logika za kreiranje i izmenu grafikona na osnovu prispelih podataka. Incijalno javascript varijabla *optionsChanged* se postavlja na vrednost nula, što znači da se prilikom izbora kompanije za kreiranje grafikona njenih akcija koristi blok kodau okviru *if* naredbe koja se izvršava u slučaju kada ta varijabla ima vrednost nula. Prispeli JSON niz sa servera se parsira i čuva u objektu, koji se prosleđuje *update* funkciji koja postavlja nove podatke u grafikon i menja teksta u HTML paragraf elementu sa id-jem *placeHolder* imenom izabrane kompanije. U slučaju da korisnik izabere ulaganje novca, odnosno klikom na dugme *"Uloži"*(slika 2),vrednost *optionsChanged*varijable se postavlja na jedinicu i aktivira se blok koda u okviru if narebe koja se izvršava u tom slučaju. U okviru toga bloka naredbi vrši se prihvatanje unete sume novca i vremena i kreira se novi niz sa podešavanjima za prikaz grafikona kojima se kreira horizontalna linija paralelna sa x osom grafikona na visini prognozirane cene akcija. Konačno, varijabla *optionsChanged* se postavlja na vrednost dva nakon isteka vremena koje je korisnik prognozirao. U tom slučaju, dolazi do restartovanja

podešavanja za prikaz grafikona na ona koja su bila pre pritiska na dugme *"Uloži"*.Za kreiranje grafikona korišćena je *jQuery flot*[6] biblioteka.

V ZAKLJUČAK

U ovom radu dat je kratak uvod u reaktivnu programsku paradigmu, korišćenjem *SpringWebFlux* tehnologije za razvoj veb aplikacije za pregled stanja na berzi. Kao i sve ostale tehnologije, Spring *WebFlux* (i generalno reaktivna programska paradigma) ima svoje prednosti i nedostatke. *WebFlux* donosi benefite u pogledu performansi aplikacije, kao i pružanja boljeg korisničkog iskustva. S druge strane, reaktivni način programiranja dolazi sa novim načinom pisanja, testiranja i debagovanja koda, tako da tranzicija sa tradicionalnog načina pisanja koda na reaktivni nije uvek brza i jednostavna. Činjenica da opisani model koristi ne-blokirajući tok izvršavanja (*eng."non-blocking execution flow")*, razlikuje ga od tradicionalnog načina razvoja gde je svaki poziv blokiran dok ne dodje odgovor, čini ga pogodnim za razvoj brojih aplikacija koje se izvršavaju u realnom vremenu (eng."real time applications"), kao što je aplikacija opisana u ovome radu. Generalno, ne postoji pravilo prilikom izbora i korišćenja tehnologija u projektu, osim da te tehnologije daju dobra biznis rešenja, lako održiva, elastična i skalabilna.

LITERATURA

[1] Ranga Rao Karanam, "Mastering Spring 5.0"

[2] Craig Walls, "Spring in Action, 4th Edition"

[3] http://www.reactive-streams.org/

[4] https://maven.apache.org

[5] Spring Framework reference Documentation,https://docs.spring.io/spring-framework/docs/current/spring- frameworkreference/html/index.html

[6] http://www.flotcharts.org/

ABSTRACT

In order to acquire a comprehensive understanding of the mutual similarities, differences, advantages and disadvantages of the traditional and reactive approach to solving some of the problems encountered by modern web applications, as well as the possibilities offered by the reactive module of the

popular Spring framework, this article will demonstrate the development of the web applications that manage companies' shares on the stock market, using both of the aforementioned approaches.

INTRODUCTION TO REACTIVE PROGRAMMING PARADIGM BY USING SPRING FRAMEWORK

Nemanja Zirojević