

УНИВЕРЗИТЕТ У БЕОГРАДУ
ФАКУЛТЕТ ОРГАНИЗАЦИОНИХ НАУКА

**ПРОЈЕКТНА ДОКУМЕНТАЦИЈА ЗА ВЕБ АПЛИКАЦИЈУ ИЗ
ПРЕДМЕТА АЛАТИ И МЕТОДЕ СОФТВЕРСКОГ ИНЖЕЊЕРСТВА**

**ТЕМА: Веб апликација за праћење листе
најновијих филмова**

Ментор:

др Драган Ђурић

Студент

Владан Симоновић

2019/3703

Београд, 2020. године

Садржај

1	Увод	1
1.1	Вербални опис – прикупљање захтева	1
2	Опис коришћених технологија.....	2
3	Корисничко упутство	3
4	Прилог	6

1 Увод

1.1 Вербални опис – прикупљање захтева

На основу корисничких захтева, могу се дефинисати следећи случајеви коришћења:

- СК1: Случај коришћења – Креирање филма
- СК2: Случај коришћења – Претраживање филмова
- СК3: Случај коришћења – Брисање филма
- СК4: Случај коришћења – Ажурирање филма
- Позив веб сервиса за преглед свих филмова

Анализом пословног система за пружање услуга гледања филмова, уочен је неефикасан рад са корисницима, појава грешака приликом обраде информација о пројекцијама, губљење документације и спора претрага података. Како би се пословни процеси убрзали и учинили квалитетнијим, неопходно је направити софтвер који ће аутоматизовати одређене процесе и олакшати посао раднику биоскопа. Софтверски систем би требало да омогући, најпре унос нових филмова, претраживање по одређеним критеријумима и брисање филмова када они више нису на репертоару.

2 Опис коришћених технологија

За израду апликације су коришћене стандардне технологије за израду веб апликација попут HTML-а, CSS-а, и оквира за приказ Bootstrap. За обраду података и backend апликације коришћен је функционални језик опште намене, Clojure. api comjure за REST api као и ring http server апстракцију за слање захтева и примање одговора.

HTML, (енг. HyperText Markup Language), је језик који служи за структурирање текстова, медија и уграђених објеката у веб странице. Састоји се од текста и тагова који омогућавају да интернет претраживач препозна структуру документа и исправно је прикаже кориснику.

CSS, Cascading Style Sheets, омогућава раздвајање структуре странице од њеног изгледа. ЦСС фајл треба да садржи опис изгледа странице и начин на који де информације бити представљене.

Bootstrap је frontend оквир који служи за креирање веб апликација. Садржи XML и CSS шаблоне за дизајн, као и Javascript екстензије. Једна од најважнијих улога јесте омогућавање респонзивности сајта.

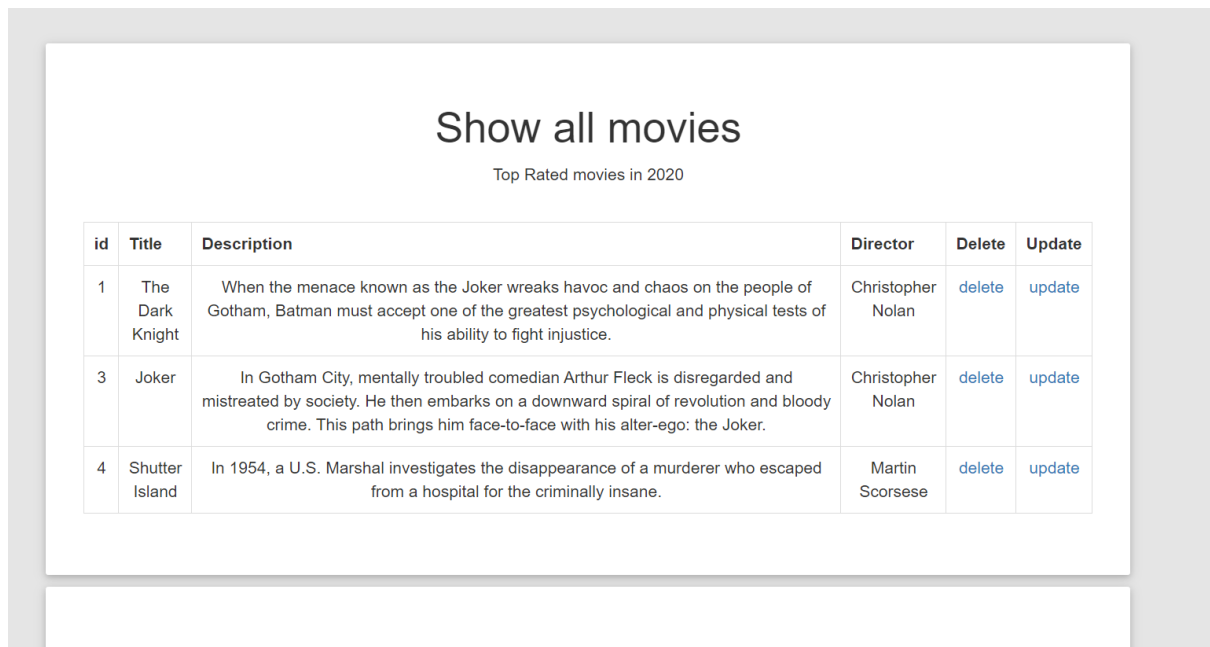
Javascript је програмски језик HTML-а и веба. Битан је део веб апликација јер омогућава интерактивност апликација. ЈаваСкрипт је скриптни језик, односно састоји се од низа команди које се читавају у интерпретору, без претходног компајлирања, и подржава објектно-оријентисано програмирање.

REST веб сервиси: REST (Representational State Transfer) представља стил архитектуре апликације, који се заснива на често коришћеним технологијама и протоколима, од којих је најважнији HTTP у комбинацији са URL-ом. Користи основне HTTP-ове методе као што су GET,POST,PUT,DELETE. Основни концепт REST-а је ресурс, који може бити информација, документ, слика, итд. Сваки ресурс има свој идентификатор, који је URL, помоћу кога се приступа и манипулише тим ресурсом. При креирању сопственог REST сервиса коришћен је библиотека comjure api.

На крају, за израду апликације коришћено је интегрисано развојно окружење Eclipse са инсталираним додатком (енг. plugin), Counterclockwise, за функционално програмирање и програмски језик Clojure.

3 Корисничко упутство

При учитавану апликације приказује се почетна страна на којој се даје приказ свих филмова, тренутно актуелних на апликацији. При дну, омогућен је унос новог филма.



Show all movies					
Top Rated movies in 2020					
id	Title	Description	Director	Delete	Update
1	The Dark Knight	When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept one of the greatest psychological and physical tests of his ability to fight injustice.	Christopher Nolan	delete	update
3	Joker	In Gotham City, mentally troubled comedian Arthur Fleck is disregarded and mistreated by society. He then embarks on a downward spiral of revolution and bloody crime. This path brings him face-to-face with his alter-ego: the Joker.	Christopher Nolan	delete	update
4	Shutter Island	In 1954, a U.S. Marshal investigates the disappearance of a murderer who escaped from a hospital for the criminally insane.	Martin Scorsese	delete	update

Слика 1 Приказ филмова

Филмови су излистани у табели при чему је приказан наслов филма, опис и режисер истог. Последње две колоне су резервисане за додатне акције над филмовима. Уколико се жели избрисати филм са репертоара потребно је кликнути на плавом бојом означени текст `delete`, док је за ажурирање филма (наслова, година издања или другог податка о филму) обезбеђена команда `update`.

За унос новог филма коришћена је форма са валидацијом за наслов као обавезним пољем при уносу филма. На следећој слици је тад приказ ове форме.



Add Movie

Title

Description

Director

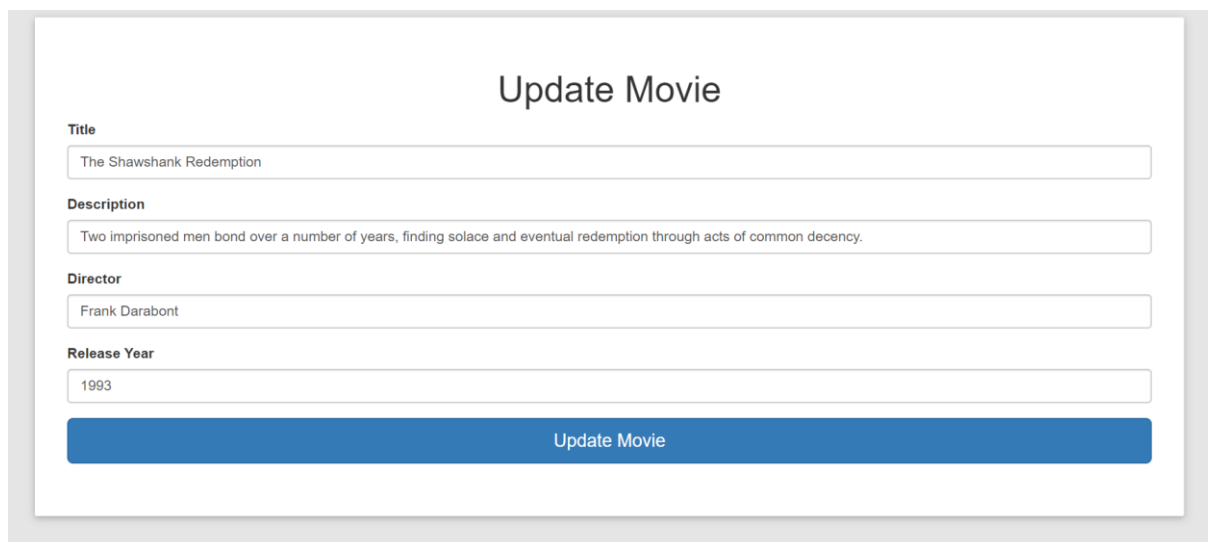
Release Year

Add Movie

Слика 2 Унос новог филма

Подаци о филму чувају се у виду текста у бази података, кликом на дугме Add Movie.

Уколико је неким случајем потребно изменити информације о филму, додати неки податак или избрисати тим функционалостима се приступа из горње табеле кликом на већ објашњена поља.



Update Movie

Title

Description

Director

Release Year

Update Movie

Слика 3 Ажурирање филма

На слици 3. може се видети нова форма на којој се могу поново уносити вредности за филм при чему, се врши ажурирање истог филма а не крерање новог, што значи да се врши само модификација истог реда у бази података.

Додатна функционалност апликације је коришћење REST веб сервиса за процесирање свих филмова тј. листање свих филмова који постоје и који су

актуелни. Сервис враћа JSON податке. У наставку је дат приказ излаза позваног REST api-ja.

```
[
  - {
    id: 1,
    title: "The Dark Knight",
    director: "Christopher Nolan",
    description: "When the menace known as the Joker wreaks havoc and chaos on the people of Gotham, Batman must accept that to become the hero Gotham deserves, but not the one it needs right now. After a long and dark journey into the depths of his psyche, the Batman must face the Joker.",
    releaseyear: "2008"
  },
  - {
    id: 3,
    title: "Joker",
    director: "Christopher Nolan",
    description: "In Gotham City, mentally troubled comedian Arthur Fleck is disregarded and mistreated by society. He then embarks on a bloody crime spree that brings him face-to-face with his alter-ego: the Joker.",
    releaseyear: "2019"
  },
  - {
    id: 4,
    title: "Shutter Island",
    director: "Martin Scorsese",
    description: "In 1954, a U.S. Marshal investigates the disappearance of a murderer who escaped from the asylum.",
    releaseyear: "2010"
  },
  - {
    id: 5,
    title: "The Shawshank Redemption",
    director: "Frank Darabont",
    description: "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.",
    releaseyear: "1993"
  }
]
```

Слика 4 REST api приказ свих филмова

4 Прилог

```
(ns movieapp.core
  (:use compojure.core
        ring.util.json-response
        ring.adapter.jetty)
  (:require [compojure.core :refer [defroutes GET POST]]
            [clojure.string :as str]
            [ring.util.response :as ring]
            [movieapp.view :as view]
            [movieapp.db :as db]
            [ring.middleware.resource :refer [wrap-resource]]
            [ring.middleware.defaults :refer [wrap-defaults site-defaults]])

(defn display-all-movies []
  (view/index-page (db/get-all-movies)))

(defn create-movie [title description director releaseYear]
  (when-not (str/blank? title)
    (db/create-movie title description director releaseYear))
  (ring/redirect "/"))

(defn delete-movie [id]
  (when-not (str/blank? id)
    (db/delete-movie id))
  (ring/redirect "/"))

(defn show-update-view [id]
  (view/update-movie-form (db/get-movie-by-id id)))

(defn update-movie [id title description director releaseYear]
  (when-not (str/blank? id)
    (db/update-movie id title description director releaseYear))
  (view/index-page (db/get-all-movies)))

(defroutes my_routes
  (GET "/" [] (display-all-movies))
  (POST "/" [title description director releaseYear] (create-movie title
description director releaseYear))
  (POST "/update-movie" [id title description director releaseYear] (update-movie
id title description director releaseYear))
  (GET "/update/:id" [id] (show-update-view id))
  (GET "/delete/:id" [id] (delete-movie id))
  (GET "/api/all" [] (json-response (db/get-all-movies))))

(def app
  (wrap-defaults my_routes site-defaults))

(ns movieapp.db
  (:require [clojure.java.jdbc :as sql]))

(def connection
  {:classname "com.mysql.cj.jdbc.Driver"
   :subprotocol "mysql"
   :subname "//127.0.0.1:3306/movie_v2"
   :user "root"
   :password "vlajce"
  })
```



```

(defn create-movie [title description director releaseYear]
  (sql/insert! connection :movie [:title :description :director :releaseYear]
    [title description director releaseYear]))

(defn delete-movie [id]
  (sql/delete! connection :movie
    ["id = ?" id]))

(defn get-all-movies []
  (into [] (sql/query connection ["select * from movie"])))

(defn get-movie-by-id [id]
  (into [] (sql/query connection ["select * from movie where id = ?" id])))

(defn update-movie [id title description director releaseYear]
  (sql/update! connection :movie {:id id :title title :description description
    :director director :releaseYear releaseYear} ["id = ?" id]))

(ns movieapp.handler
  (:use ring.middleware.json
    ring.adapter.jetty)
  (:require [compojure.core :refer [defroutes ANY]]
    [ring.adapter.jetty :as ring]
    [compojure.route :as route]
    [movieapp.layout :as layout]
    [ring.middleware.defaults :refer [wrap-defaults site-defaults]]
    [movieapp.core :as core]
  ))

(defroutes routes
  core/my_routes
  (route/resources "/")
)

(def app
  (wrap-defaults routes site-defaults))

(ns movieapp.layout
  (:use [hiccup.element :only (link-to)])
  (:require [hiccup.page :as h]
    [hiccup.page :refer [html5 include-css]]))

(defn common-layout [header & body]
  (h/html5
    [:head
      [:title "Movie"]
      (include-css "/css/bootstrap.css")]
    [:body
      [:div {:class "header"}
        [:h1 {:class "container"} header]]
      [:div {:id "content" :class "container"} body]]))

(defn not-found []
  (common-layout "NOT FOUND" [:div {:id "error"} "The page you requested could not
be found"])))

```

```

(ns movieapp.view
  (:use hiccup.page hiccup.element)
  (:require
    [hiccup.core :refer [h]]
    [hiccup.form :as form]
    [clojure.string :as str]
    [movieapp.layout :as layout]
    [ring.util.anti-forgery :as anti-forgery]))

(defn display-all-movies [movies]
  [:div {:class "card text-center"}
   [:h1 "Show all movies"]
   [:p "Top Rated movies in 2020"]
   [:br]
   [:table {:class "table table-bordered"}
    [:th "id"]
    [:th "Title"]
    [:th "Description"]
    [:th "Director"]
    ; [:th "Release Year"]
    [:th "Delete"]
    [:th "Update"]
    (map
      (fn [movie]
        [:tr
         [:td (h (:id movie))]
         [:td (h (:title movie))]
         [:td (h (:description movie))]
         [:td (h (:director movie))]
         ; [:td (h (:releaseYear movie))]
         [:td [:a {:href (str "/delete/" (h (:id movie)))} "delete"]]
         [:td [:a {:href (str "/update/" (h (:id movie)))} "update"]]
        ]) movies))]

  )

(defn add-movie-form []
  [:div {:class "form-group card"}
   [:h1 {:class "text-center"} "Add Movie"]
   (form/form-to [:post "/"]
     (anti-forgery/anti-forgery-field)
     [:div {:class "form-group"}
      (form/label "title" "Title ")
      (form/text-field {:class "form-control"} "title")]
     [:div {:class "form-group"}
      (form/label "description" "Description ")
      (form/text-field {:class "form-control"} "description")]
     [:div {:class "form-group"}
      (form/label "director" "Director ")
      (form/text-field {:class "form-control"} "director")]
     [:div {:class "form-group"}
      (form/label "releaseYear" "Release Year ")
      (form/text-field {:class "form-control"} "releaseYear")]
     (form/submit-button {:class "btn btn-primary btn-lg btn-block"}
       "Add Movie")
     [:br]])

  )

(defn index-page [movies]
  (layout/common-layout ""
    [:div {:class "col-lg-1"}]
    [:div {:class "col-lg-10"}
     (display-all-movies movies)]
  )
)

```

```

        (add-movie-form)]
        [:div {:class "col-lg-1"}]
    ))

(defn update-movie-form [movie]
  (layout/common-layout ""
    [:div {:class "form-group card"}
      [:h1 {:class "text-center"} "Update Movie"]
      (map
        (fn [movie]
          (form/form-to [:post "/update-movie"]
            (anti-forgery/anti-forgery-field)
            (form/hidden-field "id" (:id movie))
            [:div {:class "form-group"}
              (form/label "title" "Title ")
              (form/text-field {:class "form-control"} "title" (:title
movie)))])
          [:div {:class "form-group"}
            (form/label "description" "Description ")
            (form/text-field {:class "form-control"} "description"
(:description movie))]
          [:div {:class "form-group"}
            (form/label "director" "Director ")
            (form/text-field {:class "form-control"} "director" (:director
movie))]
          [:div {:class "form-group"}
            (form/label "releaseYear" "Release Year ")
            (form/text-field {:class "form-control"} "releaseYear"
(:releaseYear movie))]
          (form/submit-button {:class "btn btn-primary btn-lg btn-block"}
"Update Movie")
          [:br])) movie])
    ))

(defproject movie "0.1.0-SNAPSHOT"
  :description "Simple Clojure CRUD application with REST API"
  :url "http://example.com/FIXME"
  :license {:name "Eclipse Public License"
            :url "http://www.eclipse.org/legal/epl-v10.html"}

  :dependencies [[org.clojure/clojure "1.8.0"]
                 [hiccup "1.0.5"]
                 [org.clojure/java.jdbc "0.4.2"]
                 [mysql/mysql-connector-java "8.0.17"]
                 [ring/ring-defaults "0.1.2"]
                 [ring/ring-core "1.4.0"]
                 [ring/ring-json "0.4.0"]
                 [ring/json-response "0.2.0"]
                 [ring/ring-jetty-adapter "1.4.0"]
                 [compojure "1.4.0"]]

  :plugins [[lein2-eclipse "2.0.0"]
            [lein-ring "0.8.10"]]

  :ring {:handler movieapp.handler/app})

```