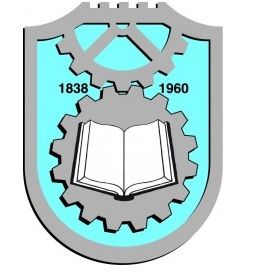
**Факултет инжењерских наука Универзитета у Крагујевцу**



**Софтверски инжењеринг 2**

**Документација пројектног задатка**

Студенти: Професор:

Никола Бабић 553/2015 Велибор Исаиловић

Јелена Глишић 566/2015

Ленка Милић 579/2015

Крагујевац, 2019.

Садржај

[1. Увод 3](#_Toc1375945)

[2. Захтеви наручиоца 4](#_Toc1375946)

[3. Архитектура програма 6](#_Toc1375947)

[4. UML дијаграми 9](#_Toc1375948)

[5. Корисничко упутство 13](#_Toc1375949)

[6. Литература 16](#_Toc1375950)

# Увод

Циљ овог пројектног задатка је апликација за управљање производима периферне рачунарске опреме (тастатуре, мишеви, подлоге, штампачи, скенери, монитори, пројектори, каблови, адаптери, слушалице, звучници, микрофони, екстерни дискови, флеш меморије и слични gadget -и).

Апликација је имплементирана као backend веб апликација. Заснована је на Angular Javascript framework-у и Codeigniter PHP framework-у. За потребе покретања оваквог система потребно је инсталирати NodeJS LTS верзију – преузети са сајта <https://nodejs.org/en/> , као и верзију PHP >5 mySQL.

# Захтеви наручиоца

Направити програм за управљање робом у продавници периферне рачунарске опреме (тастатуре, мишеви, подлоге, штампачи, скенери, монитори, пројектори, каблови, адаптери, слушалице, звучници, микрофони, екстерни дискови, флеш – меморије и слични gadget – и).

Програм треба да омогући следеће функционалности:

1. Приступ је омогућен само ауторизованим корисницима.
2. Ауторизовани корисници могу бити: Администратор, Власник, Комерцијалиста, Радник. Администратор и власник имају сва права за коришћење софтвера. Радник има ограничена права.
3. Права администратора (укључују права комерцијалисте и радника):
   1. Додавање новог производа;
   2. Уклањање производа из продајног асортимана;
   3. Промена атрибута постојећег производа (цена, опис, дужина гарантног рока, линк са описом производа на сајту произвођача, и сл.);
   4. Промена цене различитим групама производа (разне акције: по произвођачу, по врсти робе, итд.);
4. Права радника:
   1. Продаја производа;
   2. Претрага производа по имену, категорији, произвођачу, опцијама производа, итд.;
   3. Наручивање производа кога нема на стању;
   4. Замена производа;
5. Права комерцијалисте:
   1. Приступ листи наручених производа;
   2. Наручивање производа од регистрованих добављача (email);
   3. Пријем робе; промена стања залиха по наручивању производа кога није било на залихама; чување фактуре у електронском облику
   4. Приказ фактура из архиве;
   5. Промена цене производа услед промене набавне цене (цена производа у малопродаји је за 20% већа од набавне цене).
6. Аутоматизоване акције у програму:
   1. Генерисање рачуна приликом продаје и његово складиштење у дигиталној форми(pdf);
   2. Промена цене артикала када је стање њихових залиха непромењено дуже од месец дана (цена се смањује за 5% месечно, до максималних 15%). Обавештење се појављује аутоматски, непоходно је одоброње комерцијалисте, власника или администратора аручивање производа од регистрованих добављача (email);
   3. Регистровање производа са слабом продајом, складишење таквих података и могучност њиховог приказа са статистичким подацима (број продатих примерака, време када је последњи пут продат) ;
   4. Приказ производа на екрану: табела(листа и сл.) са свим производима који се могу купити у продавници, са опцијом приказа фотографија производа, линка ка страници са описом уређаја на сајту произвођача, и сл.
7. Направити алате за тестирање
8. Направити алат за помоћ кориснику програма ()
9. Написати комплетну документацију за софтвер (захтеви наручиоца, архитектура програма, техничка документација, корисничко упутство)

# Архитектура програма

Пројекат је реализован као веб апликација. Веб апликацији корисник приступа преко мреже. Backend је реализован преко Codeigniter PHP Framework-а и MySQL базе, а frontend преко Angular JavaScript Framework и за његов дизајн је употребљен Bootstrap CSS Framework. Комуникација између backend и frontend се одвија преко REST API сервиса.

**Codeigniter PHP Framework** је PHP Framework чија је сврха да учини заједничке операције лаким олакшавајући дебаговање структурног кода. Прву јавну верзију CodeIgniter објавила је лабораторија EllisLab 28. фебруара 2006.

Codeigniter PHP Framework треба одабрати из следећих разлога:

* Лаган је
* Нема инсталације, само треба укључити Codeigniter library и кренути са радом
* Широка заједница, заједница његових корисника је активна
* Обезбеђује детаљну документацију са примерима
* Флексибилан је, омогућава да се лако прошири и развије базна архитектура



*Слика 1. Codeigniter PHP Framework*

**MySQL** је RDBMS (Relational Database Management System) систем за манипулисање релационим базама података. Понаша се као сервер, са вишекорисничком функцијом, односно дозвољава приступ више корисника. Свака MySQL база може имати неколико корисника који јој могу приступити а сваки корисник има предефинисане могућности за рад, односно овлашћења. Овај приступ знатно умањује грешке. На серверу може постојати већи број база података које су потпуно самосталне, али унутар једног пројекта се може баратати подацима из више база на серверу. Сваком корисничком рачунару на серверу је могуће доделити разна административна права на цео софтвер или поједине базе. Нека од права су стварање нових база, право приступа постојећим базама, право уређивања (унос или измена података) постојећих база итд.

**Angular JavaScript Framework** (често називан и „Ангулар“) је фрамеворк отвореног кода који подржава Google и заједница појединачних програмера и корпорација за решавање многих изазова приликом креирања једностраничних апликација. Angular JS библиотека ради тако што прво прочита HTML стране, који има уграђене додатне нестандардне таг атрибуте. Ангулар те атрибуте интерпретира као директиве да би везао улазне или излазне делове странице за модел који је представљен стандардним ЈavaScript променљивим. Вредности тих JavaScript променљивих се могу ручно подесити у коду, или могу бити преузете од статичних и динамичних JSON извора. Имплементира two-way везивање података, везујући HTML(view) са JavaScript објектима (models) на веома лак начин. Једноставно то значи да ће се било каква промена на моделу сместа рефлектовати на view страни без потребе за било DOM каквом манупулацијом или руковањем догађаја (помоћу JQuery-a). Ангулар је врло флексибилан што се тиче серверске комуникације.

Као и већина JavaScript framework-a допушта рад са било којом server-side технологијом док год може да опслужује апликацију преко REST API-ja.



*Слика 2. Angular JavaScript Framework*

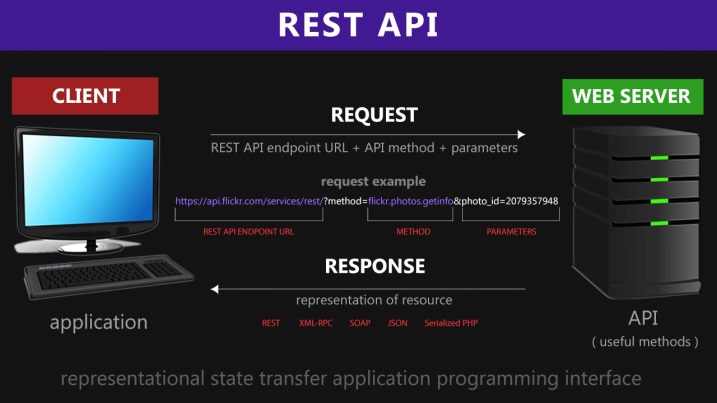
**Bootstrap CSS Framework** представља веб фрамеворк отвореног кода за креирање веб сајтова и веб апликација. Базиран је на HTML и CSS шаблонима за типографију, креирање формулара, дугмади, навигационим и осталим компонентама интерфејса, као и опционим JavaScript додацима. Циљ Bootstrap CSS Framework-a је олакшавање програмирања за веб. Такође, Bootstrap је фрамеворк за веб аплиакције тј. софтверски фрамеворк који је дизајниран да подржи развој динамичких веб сајтова и веб апикација.

****

*Слика 3. Bootstrap CSS Framework*

**REST API** је једноставан сервис интегрисан са HTTP-ом који не захтева XML поруке или WSDL описе сервиса. Данас се Rest API издвојио као доминантан мрежни сервис. REST API је концепт који се базира на промени стања клијента и све акције се обављају у тренутку промене тог стања с могућношћу враћања на неко од претходних стања. REST API архитектура се типично састоји од клијента и сервера. Клијент иницира захтев серверу, сервер процесира захтев и враћа одговор клијенту. Подаци се најчешће пребацују у JSON формату мада је доступан и XML и YAML формат. Може бити извршен на било ком клијенту или серверу који има HTTP/HTTPS подршку. Предности коришћења REST API-ја су:

* Једноставност
* Флексибилност формата враћених података
* Коришћење постојеће мрежне инфраструктуре
* Брзо савладавање технике



*Слика 4. REST API сервис*

# UML дијаграми

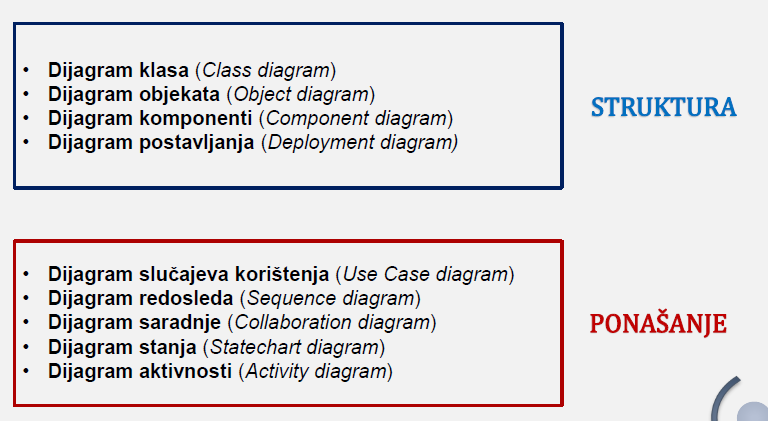
**U**ndefined **M**odeling **L**anguage је стандардни графички језик за моделовање објектно-орјентисаног софтвера. Undefined значи да уједињује све постојеће нотације. Modeling значи да се користи за моделовање софтверских елемената и Language да представља средство комуникације. Основни елементи UML-а су:

* Дијаграми (груписање међусобно повезаних колекција ствари и релација)
* Елементи (битни концепти – класе, објекти, поруке)
* Релације (повезивање индивидуалних ствари)

Дијаграми могу бити:

* Дијаграми понашања (Behaviour diagram) који обухватају дијаграм случајева коришћења (use case diagram), активности (activity diagram), машине стања (state machine diagram), као и дијаграме интеракције (interaction diagram) унутар којих се налазе дијаграми комуникација (communication diagram), редоследа/секвенци (sequence diagram) и времена (timing diagram)
* Дијаграми структура (Structure diagram) унутар којих се налазе дијаграми класа (class diagram), објеката (object diagram), компоненти (component diagram), постављања (deployment diagram), пакета (package diagram), профила (profile diagram), композитне структуре (composite structure diagram)

Њихова детаљна подела дата је на слици 5.



*Слика 5. Подела UML дијаграма*

Сви UML дијаграми коришћени за овај пројекат креирани су у програму за израду дијаграма – Modelio. То су:

* Дијаграм случајева коришћења (Use Case diagram)
* Дијаграм активности (Activity diagram)
* Дијаграм стања машине (State Machine diagram)

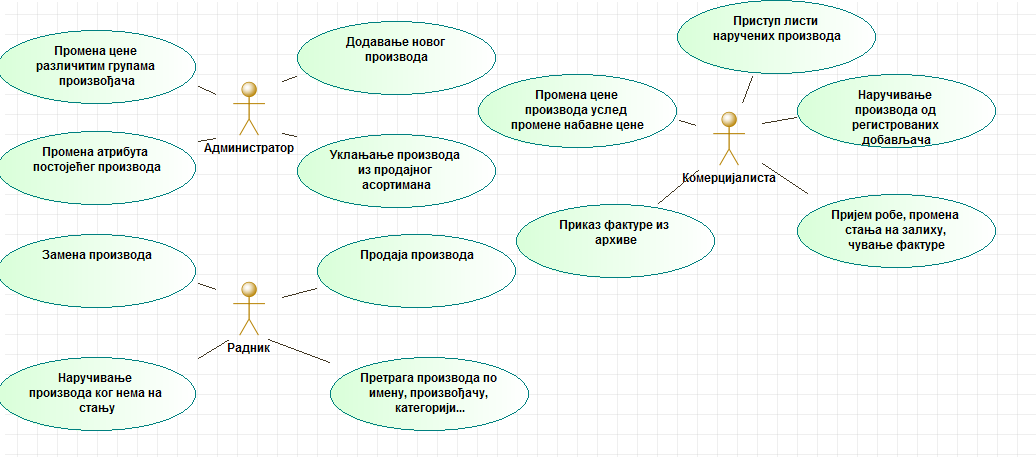
**Дијаграм случајева коришћења (use case diagram)**

Дијаграм случајева коришћења приказује скуп случајева коришћења и актера. Типично се користи да специфицира неку функционалност и понашање неког субјекта. Дијаграм визуелизује понашање система, подсистема или чак класе и интерфејса

Елементи дијаграма су:

* Случајеви коришћења
* Актери
* Релације:
  + Асоцијације(комуникација)
  + Зависности (укључивање и проширивање)
  + Генерализација
* Пакети

На слици 6 дат је изглед дијаграма коришћења случајева примењен на пројектни задатак.



С*лика 6. Изглед дијаграма коришћења случајева*

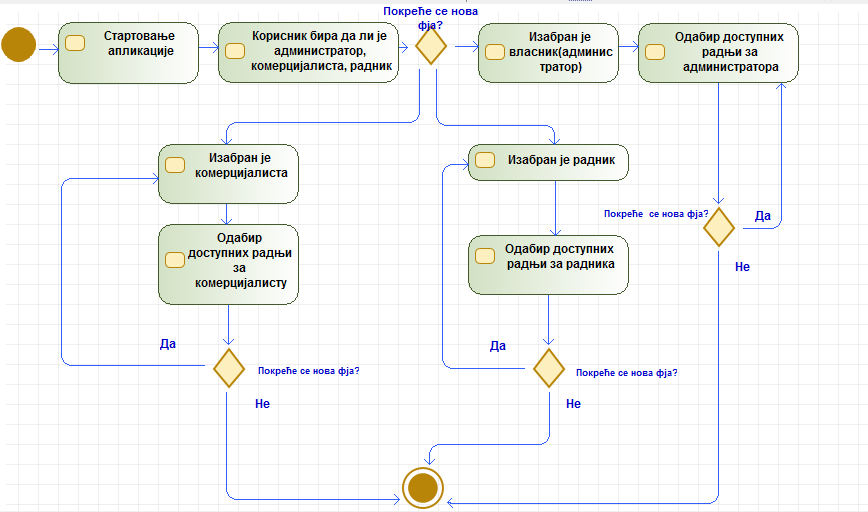
**Дијаграм активности (Activity diagram)**

Дијаграми активности су намењени моделирању динамичких аспеката (понашања) система. Дијаграм приказује:

* Ток активности коју извршавају објекти
* Евентуално и ток објеката између корака активности

Активност је спецификација понашања која се изражава кроз ток извршења преко секвенцирања и конкурисања подактивности. Акција је основна јединица спецификације понашања која репрезентује неку трансформацију или обраду у моделираном систему.

На слици 7 је дат изглед дијаграма активности примењен на пројектни задатак.

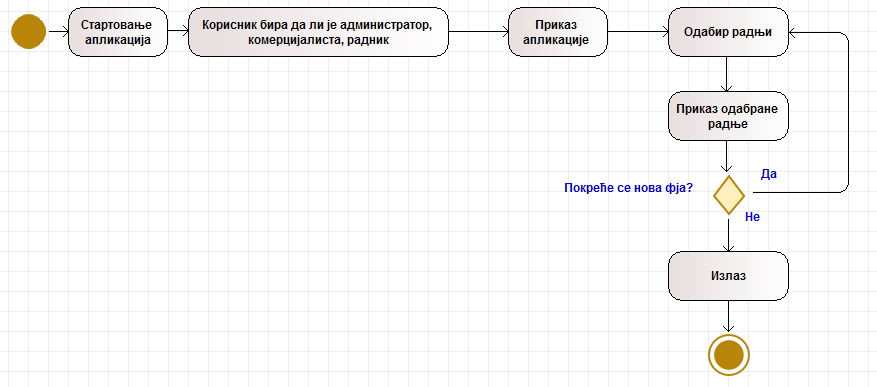


*Слика 7. Изглед дијаграма активности*

**Дијаграм стања (State Machine Diagram)**

Дијаграм стања приказује понашање које специфицира секвенца стања кроз коју пролази и моделира понашање неког ентитета или протокол интеракције. Они се креирају за ентитете које показују динамичко понашање унутар система. Дијаграм стања је граф који приказује аутомат стања (State Machine) где су чворови стања а гране су пролази.

На слици 8 је дат изглед дијаграма стања примењен на пројектни задатак.



*Слика 8. Изглед дијаграма стања машине*

# Корисничко упутство

**Упутство за админа**

Као администратор или власник апликације имате могућност да на странама одржавате своју онлине продавницу. Стране су подељене у следеће групе:

Производи:

* Табеларни преглед производа
* Брзу претрагу производа по групи и називу
* Додавање нових производа (кликом на дугме "Додај нови производ" отвара се форма за уписивање свих потребних података о производу.
* Измена производа (кликом на опцију у табели отвара се форма за измену производа и његовог стања)
* Брисање производа (кликом на опцију "Обриши" у табели)
* Наручивање производа којих нема на стању (кликом на опцију у табели отвара се форма за наручивање)
* Продаја производа (кликом на опцију у табели производ се додаје у корпу)

Групе производа:

* Табеларни преглед свих група
* Додавање група попуњавањем форме и кликом на дугме "Додај групу"
* Измену група
* Брисање група

Опције производа:

* Табеларни преглед свих опција
* Додавање опција
* Брисање опција

Произвођачи:

* Табеларни преглед произвођача
* Додавање произвођача
* Измену произвођача
* Брисање произвођача

Продати производи:

* Преглед и преузимање ПДФ рачуна продатих производа

Наручени производи:

* Табеларни преглед наручених производа

Корпа:

* Табеларни преглед производа у корпи
* Брисање производа из корпе (кликом на опцију "Ukloni")
* Продаја производа (кликом на дугме "Prodaja" производи из корпе прелазе у продате производе, након чега се преузима PDF рачун.)

**Упутство за комерцијалисту**

Као комерцијалиста имате могућност да на странама одржавате своју онлине продавницу. Стране су подељене у следеће групе:

Производи:

* Табеларни преглед производа
* Брзу претрагу производа по групи и називу
* Измена производа (кликом на опцију у табели отвара се форма за измену производа и његовог стања)
* Наручивање производа којих нема на стању (кликом на опцију у табели отвара се форма за наручивање)

Продати производи:

* Преглед и преузимање PDF рачуна продатих производа

Наручени производи:

* Табеларни преглед наручених производа

**Упутство за радника**

Као радник имате могућност да на странама одржавате своју онлине продавницу. Стране су подељене у следеће групе:

Производи:

* Табеларни преглед производа
* Брзу претрагу производа по групи и називу
* Брисање производа (кликом на опцију "Obrisi" у табели)
* Наручивање производа којих нема на стању (кликом на опцију у табели отвара се форма за наручивање)
* Продаја производа (кликом на опцију у табели производ се додаје у корпу)

Корпа:

* Табеларни преглед производа у корпи
* Брисање производа из корпе (кликом на опцију "Ukloni")
* Продаја производа (кликом на дугме "Prodaja" производи из корпе прелазе у продате производе, након чега се преузима PDF рачун.)

# Литература

1. <https://en.wikipedia.org/wiki/Unified_Modeling_Language>
2. <https://en.wikipedia.org/wiki/Activity_diagram>
3. <https://en.wikipedia.org/wiki/Use_case_diagram>
4. <https://en.wikipedia.org/wiki/UML_state_machine>
5. <https://www.w3schools.com/>