

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Рачунарска техника и софтверско
инжењерство

Ниво студија: Основне академске студије

Предмет: Програмирање интернет апликација

Број индекса: 561/2015

Коста М. Ерић

Развој *web* апликације за управљање магацинским пословањем дипломски рад

Комисија за преглед и одбрану:

1. Проф. др Миладин Стефановић - ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____



Универзитет у Крагујевцу
Факултет инжењерских наука



Основне студије: Рачунарска техника и софтверско инжењерство

Име и презиме: Коста Ерић

Број индекса: 561/2015

ПРИЈАВА ДИПЛОМСКОГ РАДА

Тема рада: **Развој *web* апликације за управљање магацинским системима**

Задатак: Задатак дипломског рада је развој и имплементација веб апликације која треба да омогући управљање магацинским системима у реалном времену. За израду апликације потребно је опишти реалан систем, дефинисати одговарајућу архитектуру, одабрати софтверске алате као подршку развоја, описати развијене моделе, описати развијени софтвер и могућности имплементације, на крају описати и фазу тестирања софтвера.

Крагујевац, 29.05.2019

Ментор:

Проф. др Миладин Ж. Стефановић



Abstract

The structure of supply chain consists of companies, branches, warehouses and goods that are managed. Each company (firm) has its branches (offices) in different cities. For each branch there are multiple warehouses managed by that branch. Within each warehouse there are various products, accompanied by formal documentation (a receive document when the goods are received in warehouse, a dispatch document when the goods are dispatched from warehouse, order document when the ordering of goods is needed). Application manages all of mentioned elements in the supply chain. Application is built as web application in PHP programming language through Laravel framework with MySQL database system and represents complete web application with all necessary elements for its functioning.

Key words: *inventory management, optimiyation, web, databases, PHP, MySQL*

Резиме рада

Структура ланца набавке се састоји од предузећа, пословница, магацина и робе којом се управља. Свако предузеће поседује своје пословнице у различитим градовима. За сваку пословницу су везана складишта којима дата пословница управља. У оквиру сваког складишта се врши размена производа уз пратећу формалну документацију (пријемница уколико је реч о пријему робе, отпремница уколико је реч о отпреми робе, наруџбеница уколико је неопходно наручивање робе). Апликација води евиденцију и омогућава управљање свим поменутих елементима ланца снабдевања. Израђена је као веб апликација у програмском језику *PHP* кроз *framework Laravel* уз *MySQL* базу података и тиме представља комплетну веб апликацију са свим неопходним елементима за њено функционисање.

Кључне речи: *управљање залихама, оптимизација, веб програмирање, базе података, PHP, MySQL*

САДРЖАЈ

1. УВОДНЕ НАПОМЕНЕ	1
2. МАГАЦИНСКО ПОСЛОВАЊЕ: ТРЕНУТНИ ИЗАЗОВИ	2
2.1. Залихе и управљање залихама	2
2.2. Трошкови у процесу управљања залихама	5
2.3. Типови управљачких модела за управљање залихама	8
2.4. Полазни модел реалног система (управљање магацинским пословањем)	17
3. РАЗВОЈНО ОКРУЖЕЊЕ И ТЕХНОЛОГИЈЕ	21
3.1 WampServer 3.1.0	21
3.1.1 Apache HTTP Server	22
3.1.2 MySQL	22
3.1.3 PHPMyAdmin	24
3.1.4 PHP	25
3.2 Laravel 5.8 – PHP Framework	26
3.3 NGROK	28
3.4 Visual Studio Code	30
4. МОДЕЛ БАЗА ПОДАТАКА И UML ДИЈАГРАМИ	33
4.1 Модел база података	33
4.2 Идентификација учесника (<i>Actors</i>) и случајева коришћења (<i>Use case</i>)	39
4.3 Дијаграм сарадње (<i>Communication diagram</i>)	40
5. ДИЗАЈН И ОПИС СОФТВЕРСКОГ РЕШЕЊА	41
6. ЗАКЉУЧАК	54
ЛИТЕРАТУРА	55

1. УВОДНЕ НАПОМЕНЕ

Тема дипломског рада је развој веб апликације која треба да омогући управљање магацинским пословањем у реалном времену, применом *MVC* архитектуре софтвера.

Главни циљ развоја веб апликације је да се кроз функционалности апликације омогући овлашћеном особљу фирме, кориснику, да води прецизну евиденцију о стању залиха складишта (магацина) фирме и одабиром модела за анализу и оптимизацију залиха дефинише оптималне наруџбе према добављачима и тиме обезбеди успешно управљање магацинским пословањем. Свако предузеће (фирма) поседује своје пословнице у различитим градовима. За сваку пословницу су везана складишта којима дата пословница управља. У оквиру сваког складишта се врши размена производа уз пратећу формалну документацију (пријемница уколико је реч о пријему робе, отпремница уколико је реч о отпреми робе, наруџбеница уколико је неопходно наручивање робе). Апликација води евиденцију и омогућава управљање свим поменутих елементима ланца снабдевања.

С обзиром да се ради о *web* апликацији, неопходно је коришћење неке врсте сервера на који ће бити постављена апликација. У овом раду је коришћен *WampServer* који се извршава на виртуелном серверу рачунара (*localhost*) али уз подршку апликације *NGROK* је могуће публиковати виртуелни сервер на *web* и тиме обезбедити да датој апликацији може да се приступи директно преко интернета са било ког уређаја. Што се тиче самих технологија, изворни код целокупне апликације је написан у програмском језику *PHP* кроз *PHP framework Laravel* који подстиче имплементацију *MVC* архитектуре софтвера, за базу података је коришћен *MySQL* систем за управљање релационим базама података, док је за уређивач изворног кода изабран *Visual Studio Code*.

2. МАГАЦИНСКО ПОСЛОВАЊЕ: ТРЕНУТНИ ИЗАЗОВИ

2

Пратити ток магацинског пословања (уопштено и пословања целог предузећа) у реалном времену и успоставити контролу над истим је и данас прави изазов. Магацинско пословање прво мора да обезбеди контролу залиха, посебно код система (предузећа) са великим пројем производа различитих категорија и модела. Након што се успостави контрола над магацином, следи други корак, оптимизација залиха. Смањење потребних залиха на оптималне количине, креирање процеса наручивања по основу оптималних количина, продаја по методу FIFO, креирање система који ће омогућити да се производ испоручи купцу у што краћем времену су предуслови дугорочне уштеде.

2.1. Залихе и управљање залихама

Залихе су све количине материјала (сировина, полупроизвода, готових производа), енергије и информација које су одређено време искључене из процеса производње или употребе, а са циљем да се у датом тренутку указане потребе могу користити.²

Значај функције управљања залихама произилази из вредности залиха којима предузеће располаже. Управљању залихама се поклања значајно већа пажња у индустријски развијеним, него у индустријски мање развијеним земљама. Проблем који се решава управљањем залихама јесте временски несклад између тражње и снабдевања, односно несклад између потреба и расположивости материјала који се користи у производњи. Захтев који треба да испуни управљање залихама јесте обезбеђење потребних материјала, тако да се процес производње може одвијати равномерно и непрекидно. Постављени захтев треба испунити уз што је могуће ниже укупне трошкове, који настају у управљању залихама.²

У пословању са купцима може се само предвиђати шта купци желе да купе, односно које производе желе да купе и када ће одређени производ купити, под условом да се производ већ налази на тржишту. У циљу превазилажења неизвесности, држање залиха производа омогућава задовољење тражње купаца, а залихе представљају резерву која треба да надомести неизвесност између интензитета снабдевања и интензитета тражње.¹

Могу се навести следеће основне функције залиха:

- Задовољавање неизвесности и промена у тражњи за производима;
- Обезбеђивање непрекидног тока производње, односно задовољење флексибилних производних планова;
- Заштита система од недостатка залиха;
- Обезбеђивање стабилности у ланцу снабдевања, односно обезбеђивање стабилног циклуса набављања и производње;
- Заштита система од последица повећања цена залиха на тржишту, куповином већих количина залиха по нижим ценама;

- Омогућавање повећања продаје, кроз давање одређених попушта на количину;
- Стварање бољих услова приликом набављања, остварујући попусте на купљену количину;
- Смањење организационих напора у планирању у ланцу снабдевања;
- Омогућавање апсорпције кашњења учесника у операцијама ланца снабдевања.

Управљање залихама је један од најважнијих задатака менаџмента, зато што залихе представљају велики део уложеног капитала и директно утичу на испоруке робе купцима. Управљање залихама има велики утицај на остале пословне функције у компанији, посебно на логистику, маркетинг, финансије и итд. Залихе се такође дефинишу као количина материјала која се користи за одржавање процеса производње или за задовољење тражње купаца. Ако се посматрају залихе у једном пословном систему, може се рећи да оне укључују залихе сировина, залихе недовршене производње или полупроизвода и залихе готових производа. У случају сагледавања операција једног производног трансформационог процеса, може се рећи да залихе прелазе из једног процеса у други процес, мењајући своје трансформационе карактеристике, тако сировине улазе или чекају на процес производње, у процесу производње сировине постају залихе у производњи, а на крају трансформације представљају готове производе. Залихе се дефинишу као скупови сировина, компонената, недовршене производње и готових производа, које се појављују на више места у оквиру производње и логистике. У компанијама залихе се налазе на различитим местима, као што су: затворена складишта, отворена складишта, продајна места, у транспорту, на продајним полицама малопродајних објеката. Процењује се да држање залиха може да кошта годишње између 20% и 40% њихове вредности, стога, пажљиво управљање нивоом залиха омогућава смањење трошкова ангажованог капитала. Поједини аутори дефинишу залихе као потенцијалну економску додатну вредност, али већина аутора и компанија, у пракси, сматра да капитал у залихама, представља погрешно везан или заробљен капитал. Ово се посебно дешава у случајевима када је трошак држања залиха већи од профита оствареног продајом истих. У таквим ситуацијама, компаније усвајају стратегију недостатка залиха, које омогућавају да потребе за производима од стране купца буду задовољене тек после одређеног временског периода, након што је поруџбина за производима послата од стране купца. Оваква стратегија управљања залихама омогућава мање везивање капитала компаније у залихама, али и претпоставља чињеницу да је купац вољан да чека одређени временски период испоруку производа.

У литератури се могу пронаћи различите поделе (класификације), односно типови залиха. Тако према степену обрађености робе могу бити залихе:

- сировина,
- полупроизвода,

- финалних проивода,
- отпада.

Према карактеру залиха:

- заштитне,
- антиципативне,
- калкулативне,
- шпекулативне.

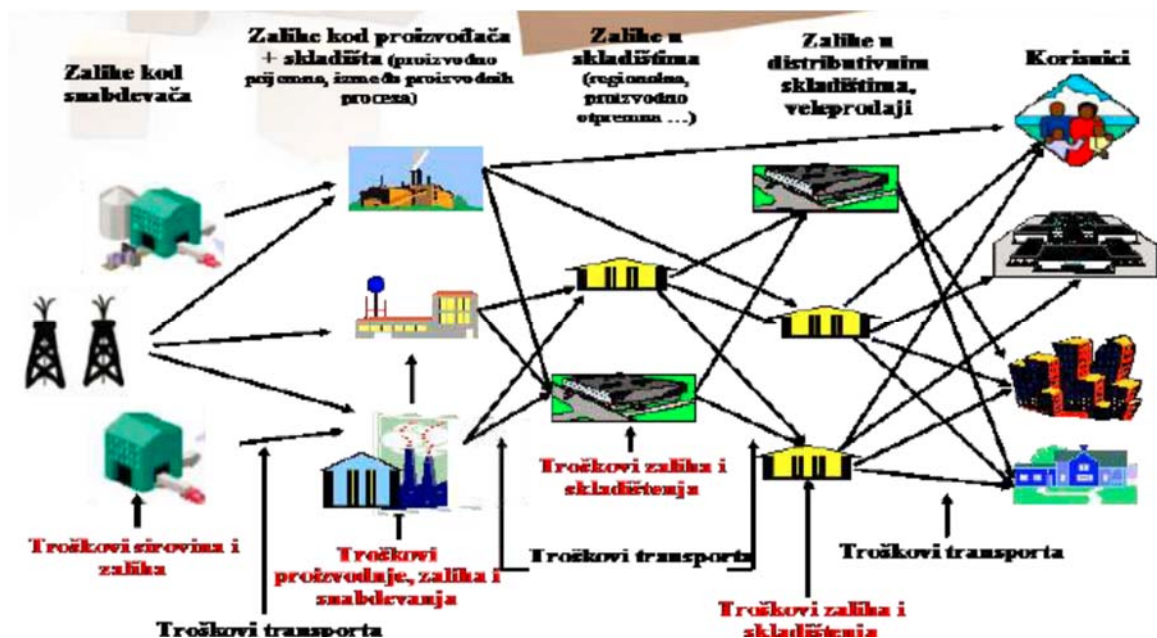
Према количини робе на залихама:

- максималне,
- просечне,
- минималне,
- оптималне,
- заштитне.

Према величини обрта:

- курентна роба,
- некурентна роба,
- средње курентна роба.

Слика 2.1 илуструје где се све могу наћи залихе.



Слика 2.1 Где се све могу наћи залихе

Основна сврха управљања залихама и контроле залиха је да се осигура да тачна количина тачно одређеног производа буде поручена у право време. Да би ова претпоставка била испуњена неопходно је одговорити на следећа питања:

1. Које производе наручити;
2. Када направити поруџбину;
3. Колико производа наручити.

Прво питање указује на важност избора производа односно сировина које треба поручити. Основни разлог држања одређених производа на залихама је:

- Омогућавање неометаног и ефикасног процеса производње;
- Добијање попушта на куповину веће количине залиха производа;
- Заштита од могућности настанка пробоја залиха услед већих флукуација на страни тражње;
- Прављење резерви залиха услед сезонских осцилација у набавци производа;
- Заштита од инфлације и промена цена на тржишту.

Као одговор на друго питање, могу се дефинисати два основна приступа:

- Према првом приступу, се у сваком поступку набавке поручује иста количина залиха. Било каква варијација у тражњи за производима или сировинама се превазилази променом времена између поруџбина, јер се увек наручују исте количине;
- Према другом приступу, поруџбине се лансирају у фиксним временским интервалима у зависности од количине залиха производа или сировина на стању. У случају било какве варијације у тражњи за производима или сировинама мења се величина поруџбине. Било каква варијација у тражњи за производима или сировинама се превазилази мењањем наручене количине, односно величине поруџбине, јер су периоди између два наручивања увек исти.

Одговор на треће питање је значајно теже одредити, јер је потребно имати модел на основу којег ће се израчунати оптимална количина залиха за поручивање. Постоје различити математички модели за одређивање оптималне величине наруџбине са најнижим трошковима. Најпознатији математички модел за одређивање оптималне величине наруџбине по најнижим трошковима је EOQ модел или модел економичне количине наручивања залиха (*Economic Order Quantity*).

2.2. Трошкови у процесу управљања залихама

Трошкови залиха представљају битан део укупних трошкова производње. Ефикасност управљања залихама значајно се огледа у ефикасном мерењу и контроли трошкова

држања залиха. Када су у питању проблеми управљања залихама, могуће је дефинисати различите врсте трошкова. У анализи трошкова залиха разматрају се четири групе трошкова.

Трошкови набавке залиха – обухватају све трошкове који настају од тренутка утврђивања потребе за производом, до тренутка добијања производа на складишту предузећа. У ширем смислу, ови трошкови обухватају и набавну цену самог производа. Већина модела у оквиру овог трошка не урачунава цену производа, него се само разматрају трошкови припремања наруџбине, њене реализације и транспорта производа до складишта предузећа. У моделима који се односе на сопствену производњу у самој организацији, уместо трошкова набавке, рачунају се и трошкови припреме производње. Као најзначајнији трошкови у оквиру овог трошка могу се навести: плате запослених у набавној служби, птт услуге, канцеларијски материјал и транспорт. Обрачун трошкова набавке се врши тако, што се сви саставни трошкови саберу за један обрачунски период, тако да се добију укупни трошкови наручивања у обрачунском периоду. Када се ови трошкови поделе са бројем наруџбина у датом периоду, добијају се просечни јединични трошкови по наруџбини.

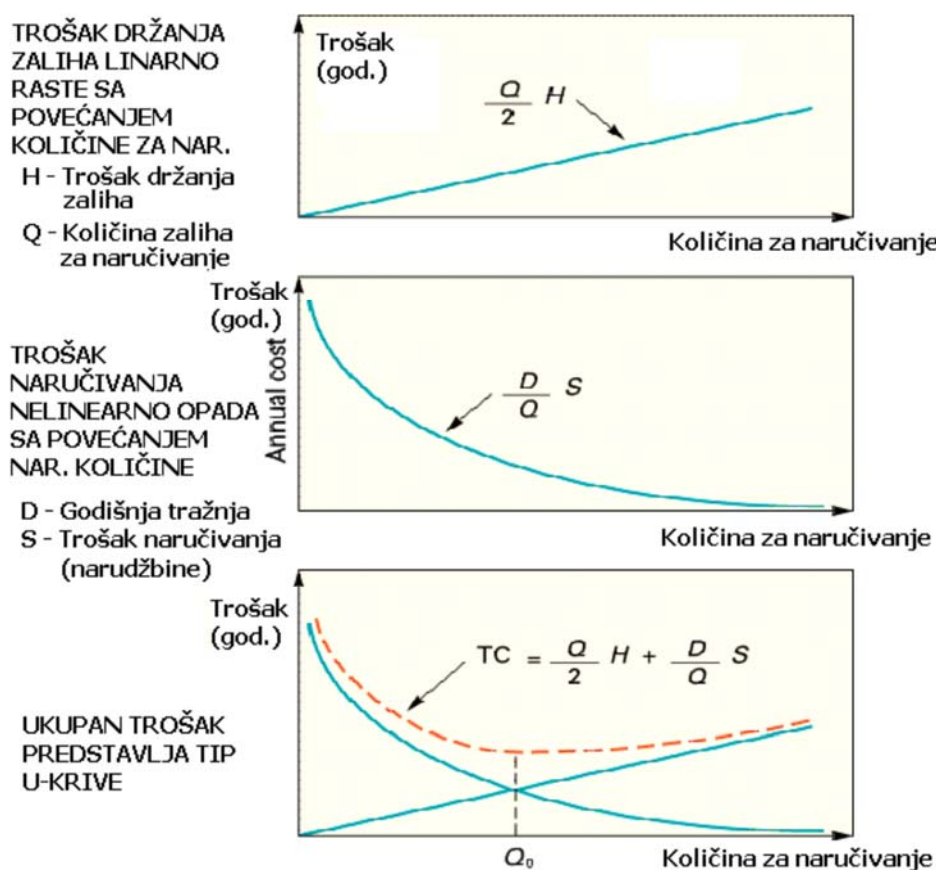
Трошкови држања залиха – настају из чињенице да предузеће поседује залихе. У најзначајније ставке овог трошка спадају: трошкови капитала, трошкови простора и трошкови запослених који раде у складишту. Трошкови држања залиха се обично изражавају процентом од просечне вредности залиха. Да би се добио овај податак потребно је знати вредност просечних залиха у току обрачунског периода и све трошкове који настају услед држања залиха. Такође, јединични трошкови држања залиха се изражавају у вредности по јединици залихе у јединици времена.

Трошкови недостатка залиха – обично се класификују у две групе: трошкови услед прекида рада и застоја у производњи и трошкови услед ванредних поступака због недостатка залиха. Застој у производњи услед недостатка залиха има као последицу пропуштање шансе да се заради. Недостатак залиха изазива слабљење поверења и добре воље код клијената. Ванредни поступци и активности могу створити додатне трошкове у вези са технолошким процесима, односно трошкове хитне набавке, ванредног наручивања, хитне израде и итд. Трошкови недостатка залиха се обично изражавају у вредности по јединици робе и јединици времена. Ову врсту трошкова је изузетно тешко мерити, па се они по правилу процењују. Нарушавање угледа фирме, услед недостатка залиха је изузетно тешко мерити и изразити квантитативно путем износа трошка. Стога се у математичким моделима уместо трошкова недостатка залиха користе други показатељи у вези са снабдевеношћу складишта потребном робом. То су коефицијент снабдевености,

као вероватноћа да ће на складишту бити расположива роба у тренутку када се тражи и очекивани број незадовољених наруџбина у посматраном временском периоду.

Остали трошкови – су додатни трошкови, који се тешко разврставају у једну од горе наведених категорија. Типичн пример ових трошкова су трошкови расхоровања застарелих залиха, трошкови флексибилности производње услед неочекиване промене тражње и тд.

Поред приказане, постоје и многе друге поделе трошкова управљања залихама које се могу наћи у релевантној литератури. Да би се што боље разумели укупни трошкови залиха, неопходно је сагледати понашање ових трошкова у односу на количину производа који се наручује за залихе, односно у односу на величину наруџбине, као што је приказано на Слици 2.2.



Слика 2.2 Графички приказ релевантних трошкова залиха у зависности од величине наруџбине

Основни циљ управљања залихама јесте уравнотежење расположивости производа или тражње купаца са трошковима, који истовремено овај процес омогућавају. Задовољавање потреба купаца, може се остварити на велики број начина, али треба изабрати онај који

минимизира трошкове држања залиха, а истовремено максимално задовољава тражњу купаца.¹

2.3. Типови управљачких модела за управљање залихама

Савремени приступи управљању залихама значајно се ослањају на интензивно коришћење информационих система и погодних математичких модела. Моделирање залиха може бити усмерено ка развоју рачунарских метода и алгоритама које се могу користити у конкретним системима залиха, за решавање појединачних практичних проблема, а такође може бити усмерено ка изучавању теорије залиха, у циљу изучавања односа и једначина које описују систем. Постоји велики број типова модела залиха и различити критеријуми њихове класификације. Фактори који су кључни приликом одређивања типа и структуре модела су:

1. Надгледање односно контрола залиха – *непрекидно или периодично*;
2. Карактер тражње – *детерминистичка, случајна или непозната тражња*;
3. Стационарност – *да ли се параметри расподеле вероватноћа, које описују случајне величине у моделу мењају у времену*;
4. Период планирања – *само један период, коначан временски хоризонт подељен на више периода, бесконачан временски хоризонт*;
5. Незадовољена тражња – *да ли се незадовољена тражња губи или нагомилава*;
6. Трошкови – *просечни или дисконтовани*;
7. Испорука – *тренутна или постоји време испоруке, које може бити константно или случајно*;
8. Трајност залиха – *неограничена или ограничена*;
9. Број артикала – *модел за један артикал или више артикала*.

Модели управљања залихама могу бити подељени на:

Детерминистичке и пробабилистичке моделе залиха. Када се говори о детерминистичким моделима залиха, претпоставка је да су сви улази у модел, посебно тражња купаца, познати у тренутку доношења одлука. У реалним ситуацијама, компанија мора предвиђати будуће потребе купаца, одређеним моделом предвиђања тражње. Излазни резултат из ових модела предвиђања је просечна тражња и стандардна девијација тражње. Међутим, у детерминистичким моделима се користе само просечне вредности и искључују се било какве информације о неизвесности, као што су у овом случају стандардна девијација. Овакво посматрање детерминистичких модела има за резултат стварање једноставнијих модела, али обично мање реалних. Пробабилистички модели за разлику од детерминистичких модела користе информације о неизвесности. Ови модели су типично знатно тежи за анализу, али за резултат имају боље одлуке, посебно када је ниво неизвесности у процесу одлучивања висок.

Модели са независном (екстерном) и зависном (интерном) тражњом. Независна тражња се јавља када компанија продаје производе крајњем купцу и када не може директно контролисати величину и време настајања тражње купаца. Типичан пример је малопродаја која наручује производе од добављача и онда чека да види колико ће купаца тражити те производе у екстерној тражњи. У овом случају одлука о набавци не утиче на ниво тражње купаца. Супротно од независне тражње јавља се зависна тражња или интерна тражња претежно у монтажним и производним процесима. Као пример, у циљу лакшег разумевања, наводи се компанија која производи фрижидере. За фрижидерима, као готовим производима постоји независна (екстерна) тражња купаца, али за саставним деловима који улазе у готов производ – фрижидер постоји зависна тражња. Када компанија изврши предвиђање тражње за будући период и дефинише планове производње у броју фрижидера, које је неопходно направити, приступа се набавци тачно одређеног броја залиха компоненти и склопова, у тачно потребним количинама, за производњу планираног броја фрижидера. Основни проблем модела за зависном тражњом је осигурати да све компоненте буду на правом месту у право време.

Модели за наручивање и за производњу. Модели за управљање залихама се разликују и према томе да ли компанија залихе наручује од добављача или исте производи интерно у сопственој производњи. Ако се залихе производа наручују екстерно постоји тзв. време кашњења у наручивању залиха, које представља време које протекне од тренутка када је наруџбина лансирана ка добављачу до тренутка пријема залиха у складиште. У моделима наручивања залиха постоји фиксни трошак наручивања залиха, који настаје сваки пут када је наруџбина направљена. Овај трошак је независан од наручене количине производа и представља трошак припреме сваке наруџбине. Супротно, ако се производи производе интерно, постоји време потребно за производњу једне јединице производа или једне серије производа. Ово време представља стопу продуктивности производње. Такође, ово време може да представља време припреме производње, односно време за подешавање опреме односно машина да би се произвео специфичан производ. Као у моделима наручивања, могу такође постојати трошкови припреме производње сваки пут када је потребно произвести одређену серију производа. Овај трошак је независан од величине серије која се производи.

Модели континуалне (непрекидне) контроле (надгледања) залиха и периодичне контроле (надгледања) залиха. У моделима континуалне контроле залиха, залихе се контролишу непрекидно и наруџбину је могуће направити у било ком тренутку времена. Постоји тачка (тренутак) поновног наручивања залиха, односно специфичан ниво залиха, који залихе временом достижу и при коме се врши поновно наручивање залиха. Активност наручивања залиха се може десити у било ком дану или делу одређеног дана током месеца. У супротном, периодична контрола (надгледање) залиха подразумева

стандардна времена за наручивање залиха, као што су једанпут месечно или почетком сваке седмице. У пракси, модели непрекидне контроле залиха се могу применити, уколико компанија поседује компјутеризован приступ надгледању залиха у реалном времену. Ови модели, сигурно, могу да допринесу снижавању нивоа залиха у компанији, јер су знатно флексибилнији на промене у окружењу компаније. Међутим, када компанија складишти хиљаде артикала на залихама, понекад је погодније користити периодичне моделе контроле залиха и залихе поручивати на пример једном недељно.

Модел за један артикал и за више артикала. Један од фактора према коме се могу поделити залихе су број производа које модел третира у наручивању. Модел који су прављени за наручивање или управљање залихама једног производа су концептуално и математички једноставнији. У реалности, већина компанија истовремено управља залихама са више различитих производа, односно од једног или више добављача поручује више производа одједном. Овакав начин наручивања више артикала одједном или синхронизација више наруџбина, доводи до смањења трошкова наручивања производа.

У детерминистичке моделе са независном тражњом се убрајају: модел економичне количине наручивања (EOQ), модел са попустом у зависности од набављене количине залиха (*Quantity Discount Model*), модел са планираним пробојем односно недостатком залиха (*Planned Shortages Model*). У овим моделима тражња за једном компонентом је константна и позната, али и независна од тражње за другим компонентама. У пробабилистичке моделе са независном тражњом спадају следећи модели: проблем продавца новина (*The Newsboy Problem*), модели нивоа услуге (*Service Level Models*) и модели периодичне контроле, надгледања залиха (*Periodic Review Systems*).

MRP (*Material Requirements Planning*) и JIT (*Just-In-Time*) су модели са зависном тражњом код којих тражња за одређеном компонентом је у зависности од постојећих планова производње.

Модел економичне количине наручивања залиха (EOQ модел)

Најпознатији математички модел за одређивање оптималне величине наручивања (наруџбенице) по најнижим трошковима је EOQ модел или модел економичне количине наручивања залиха. Када је тражња константна и позната, класични модел за одређивање економичне количине наручивања залиха (EOQ модел) представља основни модел за разумевање теорије залиха. Модел економичне количине наручивања залиха одређује оптималну количину за наручивање која минимизира укупне трошкове држања залиха. Као резултат излаза из модела за један временски период (на пример једна година) добијају се подаци о оптималној количини наручивања и броју наруџбина током године.

Оптималне величине набавки и оптималан број наручивања производа током године битне су величине с обзиром на осигурање континуалне производње, односно снабдевања купаца, укупне трошкове набавке и складиштења производа. За израчунавање оптималне величине (количине) набавки и оптималног броја наручивања користи се формула:

$$Q_{opt} = \frac{\sqrt{200 \cdot Q_{pl} \cdot T}}{c_n (k_{\%} + t_{\%})}$$

где је:

Q_{opt} — оптимална количина набавке производа,

Q_{pl} — годишњи промет производа,

T — трошкови појединачне набавке, *дин*

c_n — набавна цена по јединици производа, *дин*

$t_{\%}$ — стопа трошкова складиштења, %

$k_{\%}$ — каматна стопа, %

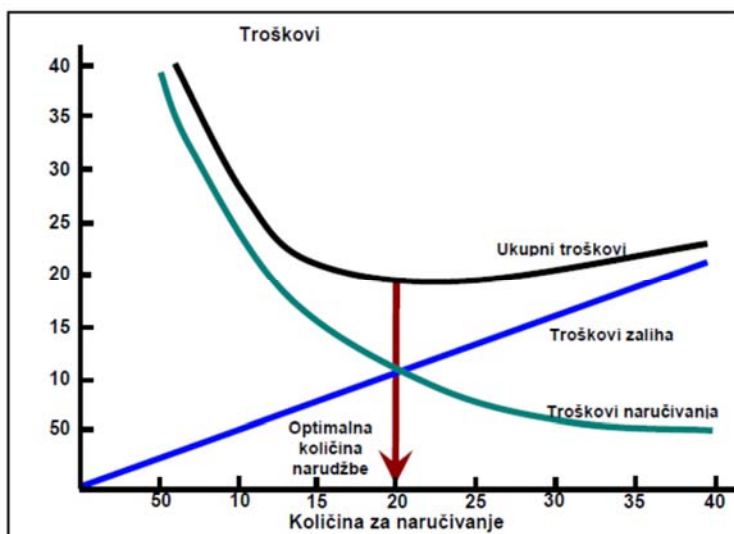
Оптимални број наручивања током године, израчунава се на начин да се планирана годишња потрошња или промет подели са оптималном количином набавке:

$$N_{opt} = \frac{Q_{pl}}{Q_{opt}}$$

где је:

N_{opt} — оптимални број наручивања у току године.

Може се рећи да крива укупних трошкова достиже свој минимум у пресеку кривих укупних годишњи трошкови складиштења (залиха) и укупних годишњих трошкова наручивања производа, као што је приказано сликом 2.3.



Слика 2.3 Графички приказ релевантних трошкова залиха

Модел минималних залиха

С обзиром да се израчунавање минималних залиха базира на просечној дневној потрошњи или продаји производа, држање минималних залиха има смисла, само, уколико се ради о производном или трговачком предузећу, које у пословању нема сезонских осцилација и има поуздане добављаче, да на њих може увек рачунати по питању сигурности испоруке наручених производа.

Минималне залихе се израчунавају по обрасцу:

$$Z_{\min} = Q_{dn} \cdot V_{nab}$$

или

$$\overline{Z}_{\min} = \frac{Q_{god} \cdot V_{nab}}{D}$$

Q_{dn} – дневна (просечна) потрошња (промет),

Q_{god} – годишња (просечна) потрошња (промет),

V_{nab} – време набавке,

D – број радних дана у години.

Модел максималних залиха

13

Максималне залихе представљају горњу границу количине производа у складишту изнад које се у одређеном периоду не набављају производи јер је то економски неоправдано. Држање максималних залиха има смисла када производња или потражња купаца, мање или више осцилује током године, па се предузеће политиком држања максималних залиха осигурава од несташице производа.

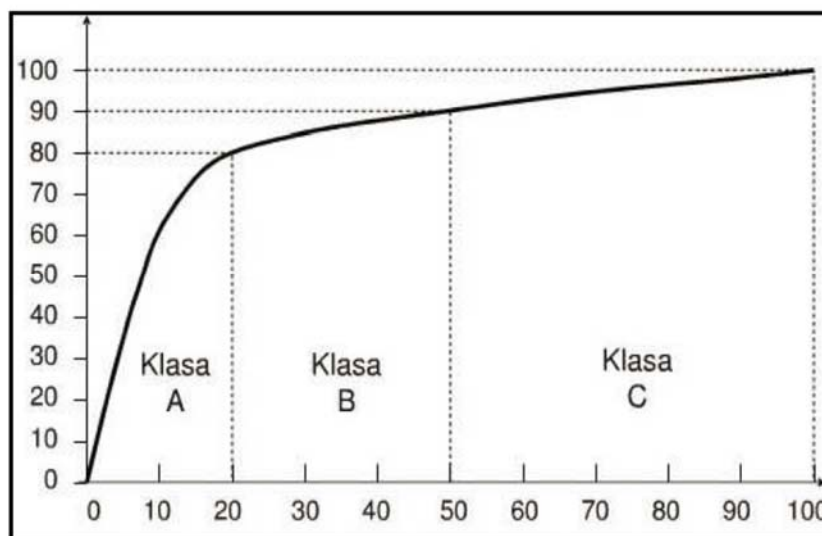
Постоји више могућности израчунавања количине производа која представља норматив за максималну вредност залиха, а најчешће се користи начин по којем се вредност највеће планиране продаје подели са данима одабраног или планираног периода и резултат помножи са норма данима (број дана у којима предузеће мора имати одговарајућу количину залиха производа), односно:

$Z_{max} = (\text{Вредност највеће планиране продаје} / \text{Број дана периода за који се тражи норматив максималне залихе}) * \text{Норма дани}$

АВС анализа залиха

АВС анализа представља методу груписања, која полази од претпоставке да се производи на залихама могу поделити у групе у зависности од њиховог релативног значаја. Наиме, сви производи или одређене врсте производа немају исти значај посматрано са аспекта профитабилности компаније, у складу са тим се креира политика управљања залихама, којом се утврђује расположивост производа за подмиривање тражње. Диференцирање подразумева претходно дефинисање значаја појединих производа и њихово рангирање, оно се може вршити према различитим критеријумима, а најчешће је то према годишњој употребној вредности. Стога, што трошкови управљања и контроле производа мале вредности могу бити знатно већи од држања нешто већих залиха тих производа. Ова метода подразумева да производи које карактерише већа остварена годишња употребна вредност завређују и већу пажњу менаџмента, обзиром да могу донети већи профит. Диференцијација најчешће узима у обзир три групације производа односно три нивоа приоритета:

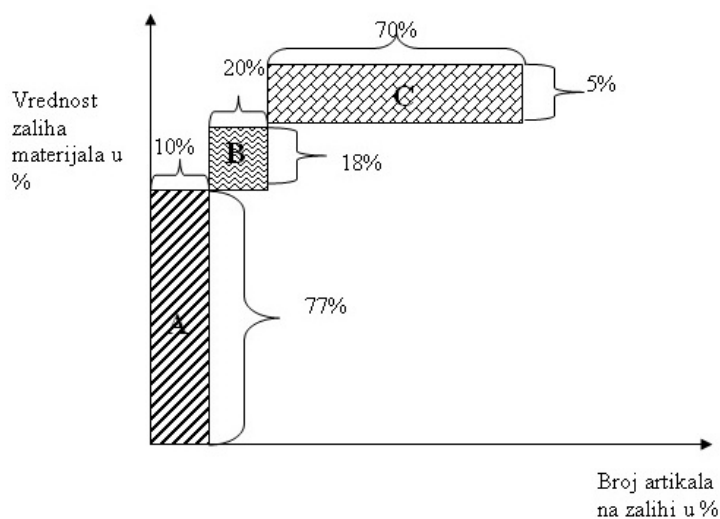
- Класа А – најважнији артикли; приближно 20% највредније робе која чини око 80% вредности свих залиха;
- Класа В – средње важни артикли; око 30% робе средње вредности која чини око 10% укупне вредности залиха;
- Класа С – најмање важни артикли; остатак од око 50% мало вредне робе које чине око 10% вредности укупних залиха.



Слика 2.4 Крива дистрибуције за ABC анализу

Анализом графика датог сликом 2.4, који иначе изражава Паретово правило, може се уочити да је велики проценат годишње употребне вредности, од чак приближно 80% заправо концентрисан у релативно малом броју производа, око 20%. Логично је зато да А класа захтева највећу пажњу и контролу од стране менаџмента, посебно стратешког, у смислу помног праћења као основе за што прецизнију процену будуће тражње и, консеквентно, обима потребних залиха. В класа производа се третира слично као и А групација, уз акценат на аутоматизацији одлучивања, јер је у питању велики број рутинских одлука. Да би пажња могла бити фокусирана на класу А, менаџмент, корективно реагује само у изузетним случајевима. Услед изузетно мале вредности, уместо праћења тражње, за артикле који припадају С класи најчешће се утврђује виши ниво сигурносних залиха, циљ је минимизирање административних трошкова и укупног броја наруџби према добављачима. Ипак, у савременим условима, захваљујући високом степену развоја информационих технологија, рачунарски ресурси пружају могућност да се и производи из ове класе све чешће третирају као они из В групације.

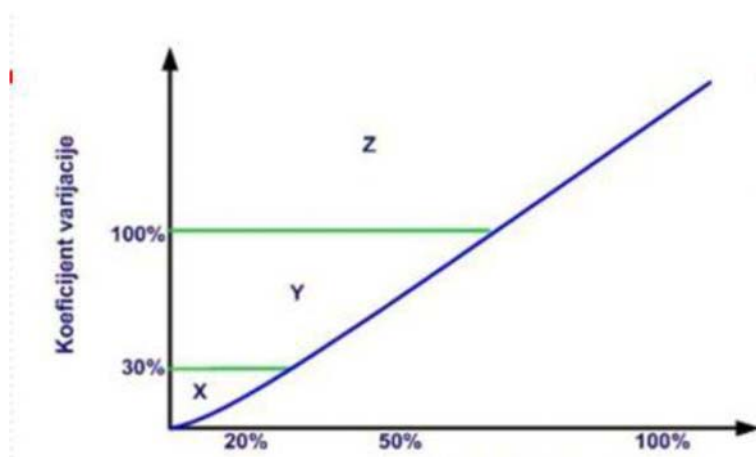
Генерално правило је да за класу А производа треба радити детаљну анализу, и често и детаљно ажурирати податке о потражњи, за класу В производа обављати само рутинску контролу и рутинско ажурирање. Класу С производа анализирати и вршити ажурирање података периодично. Слика 2.5 приказује однос структуре залиха по вредности и броју производа.



Слика 2.5 Однос структуре залиха по вредности и броју производа (артикала, материјала)

XYZ анализа залиха

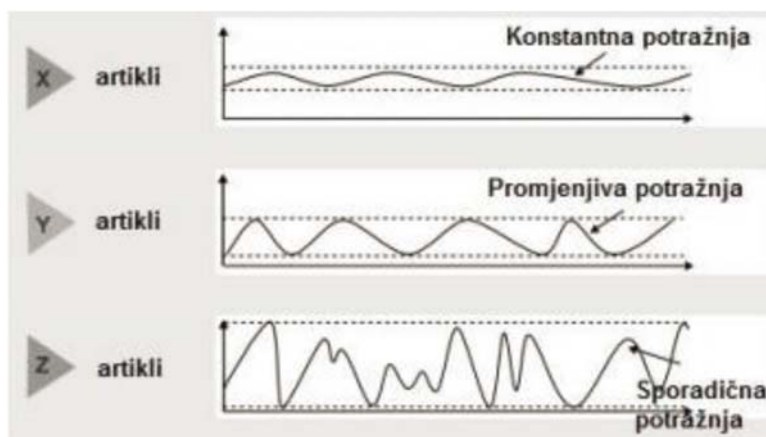
XYZ анализа је секундарна анализа која говори о стабилности потражње за неким производима који се сврставају у три групе, X, Y и Z, применом критеријума варијабилности потражње у односу на просечну потражњу (слика 2.6). Мерење варијабилности продаје врши се коефицијентом варијације који се рачуна на начин да се у однос поставе: одступања од просека (стандардна девијација) и просечна продаја (аритметичка средина).



Слика 2.6 Графички приказ XYZ анализе

XYZ анализа је и метода која обухвата и другу димензију залиха односно варијабилност потражње појединих производа. Сврха примене ове класификације као и код ABC анализе

је успостављање оптималног система набавног, продајног и складишног пословања а све у циљу смањења трошкова залиха, набавке и складишта, што је један од основних циљева логистике. Објекти XYZ анализе могу бити различити економски индикатори предузећа: продаја, приходи, материјални трошкови, број добављача итд. Једна од најчешћих области примене ове методе је дефиниција производа који имају сталну потражњу (константна потражња), сезонску (променљиву потражњу) и случајну (спорадичну потражњу), што је илустровано сликом 2.7.



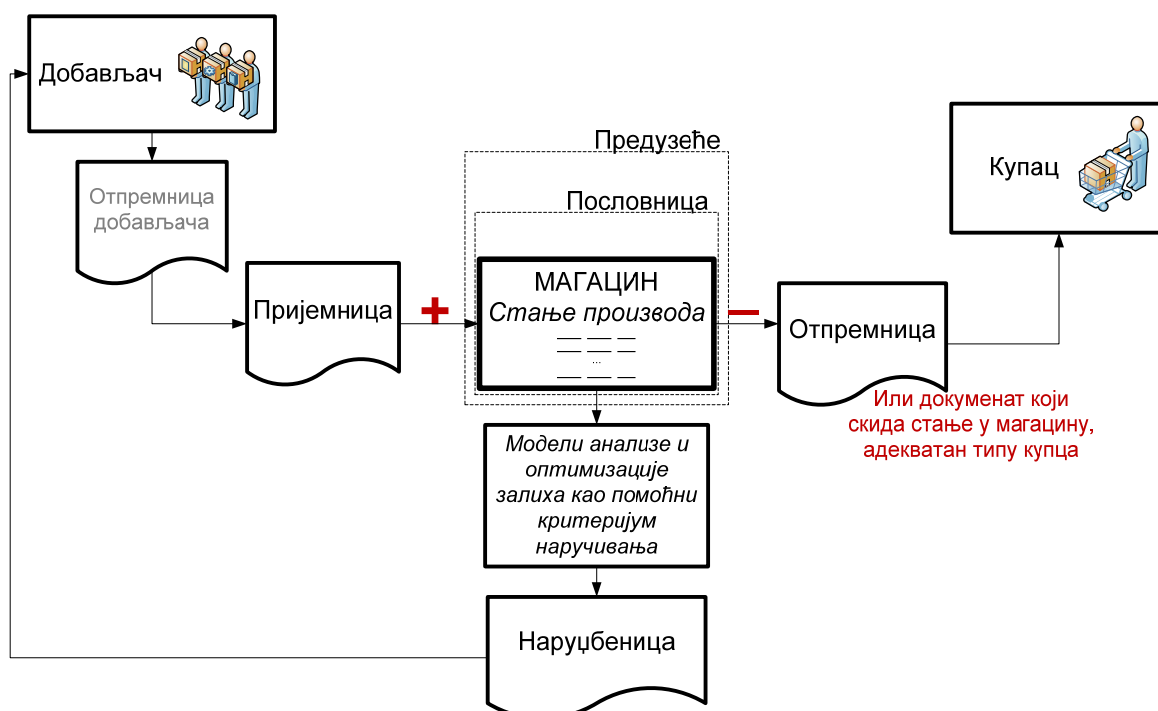
Слика 2.7 Одступање у потражњи по групама

XYZ анализа распоређује производе у три групе према континуитету потрошње/потреба и сигурности потрошње:

- X група (континуирана потражња): производи за које постоји континуирана потражња или се у њиховој потражњи јављају мање осцилације (до 10%) и на тај начин се постиже велика тачност прогнозе потражње. X групу чини 50% врста производа.
- Y група (дисконтинуирана, променљива потражња): производи чија је потражња променљива. Колебања потражње у појединим временским периодима су до 60%, чиме се постиже само средња тачност прогнозе потражње. Y групу чини 20% врста производа.
- Z група (повремена потражња) : производи чија је потражња повремена уз велике размаке и одступања у количини потражње (преко 60%) па је тешко дефинисати потражњу, за ове производе се постиже мала тачност прогнозе потражње. Z групу чини 30% врста производа.

2.4. Полазни модел реалног система (управљање магацинским пословањем)

На основу претходног описа реалног система, односно математичких модела, активности у оквиру функције управљање магацинским пословањем, пратеће документације, формиран је полазни модел управљања магацинским пословањем (слика 2.8) за дефинисање логичког и физичког модела база података и израду апликације.



Слика 2.8 Модел магацинског пословања

На основу два основна документа, пријемнице (изглед приказан на слици 2.9) и отпремнице (изглед приказан на слици 2.10), се врши промена стања производа у магацину. Пријемницом се стање производа повећава а отпремницом смањује. Поред ова два документа која прате активности пријема и отпреме, стање производа у магацину могу променити и активности повраћаја производа као и из неког законски одобреног разлога отпис производа. Наружбеница, са могућим изгледом као на слици 2.11, не мења стање производа у магацину, се формира по добављачу на основу оптимизације и анализе залиха по неком од понуђених модела који су претходно описани.

Prodavac:
Adresa:
Mesto:
PIB:

Kupac:
Adresa:
Mesto:
PIB:

OTPREMNICA BR. _____

Datum izdavanja: _____

Redni broj	Naziv robe	Jed. Mere	Količina	Cena	Iznos

Robu izdao

Robu primio

mp

Слика 2.10 Могући изглед отпремнице

Анализом документа "отпремница", уочавају се објекти (ентитети):

- ОТПРЕМНИЦА са атрибутима: број отпремнице, датум издавања (књижења), ...
- АРТИКАЛ (ПРОИЗВОД) са атрибутима: шифра производа, назив производа, јединица мере, количина, цена, износ,
- ДОБАВЉАЧ са атрибутима: шифра добављача, назив добављача, место и адреса, ПИБ добављача,
- КУПАЦ са атрибутима: шифра купца, назив купца, место и адреса, ПИБ купца, ...

PORUČILAC: Mesto: _____ Ulica i br.: _____ Telefon: _____ Web sajt: _____ Imejl: _____ Br. Žiro rač.: _____ PIB: _____	DOBAVLJAČ: Mesto: _____ Ulica i br.: _____ Telefon: _____ Web sajt: _____ Imejl: _____ Br. Žiro rač.: _____ PIB: _____
--	--

NARUDZBENICA br. _____ Datum: _____

Na osnovu: _____

Red. br.	Šifra artikla	Naziv robe	Jed. mere	Kolicina	Cena
1.					
2.					

NAČIN I USLOVI NABAVKE:

NAČIN I USLOVI PLAĆANJA:	_____
MESTO ISPORUKE ROBE:	_____
NAČIN ISPORUKE:	_____
VREME ISPORUKE:	_____
MESTO DOSTAVE RAČUNA:	_____
OSTALI USLOVI:	_____ / _____

Prodavac

 (ovlašćeni radnik)

(M.P.)

Poručilac

 (ovlašćeni radnik)

Слика 2.11 Могући изглед наруџбенице

Анализом документа "наруџбеница", уочавају се објекти (ентитети):

- НАРУЏБЕНИЦА са атрибутима: број наруџбенице, датум издавања, ...
- АРТИКАЛ (ПРОИЗВОД) са атрибутима: шифра производа, назив производа, јединица мере, количина, цена,
- ДОБАВЉАЧ са атрибутима: шифра добављача, назив добављача, место и адреса, ПИБ добављача, имејл добављача, телефон, веб сајт,
- ПОРУЧИЛАЦ са атрибутима: шифра поручиоца (купца), назив купца, место и адреса, ПИБ купца, имејл, веб сајт, телефон, ...

3. РАЗВОЈНО ОКРУЖЕЊЕ И ТЕХНОЛОГИЈЕ

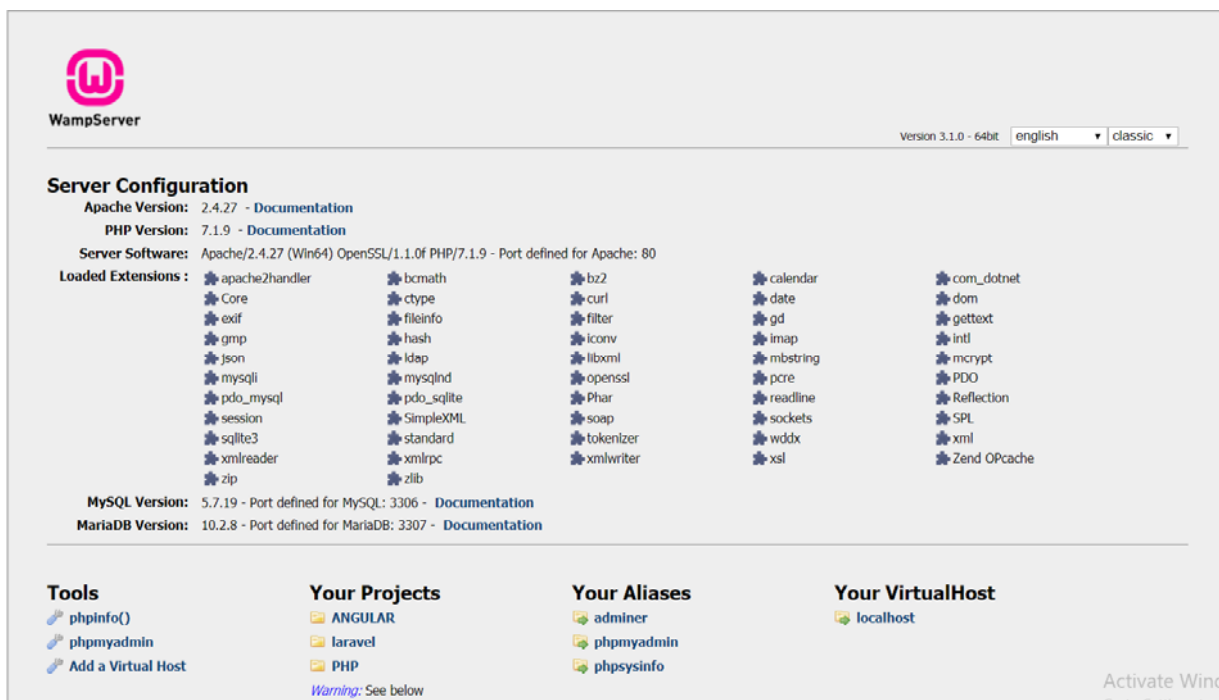
21

Технологије коришћене у дипломском раду су:

- *WampServer 3.1.0*
- *Laravel 5.8 – PHP Framework*
- *NGROK*
- *Visual Studio Code*

3.1 WampServer 3.1.0

За "Windows, Apache, MySQL и PHP", WAMP је варијација LAMP-а за Windows системе и инсталира се као пакет софтвера (Apache, MySQL и PHP). Најчешће се користи за развој интернет апликација и интерно тестирање на Localhost серверу. Слика 3.1 илуструје један од почетних конфигурационих панела WampServer-а.



Слика 3.1 Конфигурациони панел WampServer-а

3.1.1 Apache HTTP Server

Најважнији део *WAMP* пакета је *Apache* (или „*Apache HTTP Server*“) који се користи за покретање веб сервера на *Windows* оперативном систему. Покретањем локалног *Apache* веб сервера на *Windows* машини, веб програмер може тестирати веб странице у веб претраживачу без објављивања истих на Интернету. Пројекат *Apache HTTP Server* је настао као продукт колаборативног рада на развоју софтвера који има за циљ стварање робусне, комерцијалне, функционалне и доступне (*open-source*) имплементације изворног кода *HTTP* (Веб) сервера. Пројектом заједнички управља група волонтера широм света, који користе Интернет за комуникацију, планирање и развој сервера као и креирање пратеће документације. Овај пројекат је део *Apache* софтверске фондације. Поред тога, стотине корисника су допринеле развоју идеје, кода и пројектне документације.

WAMP такође укључује *MySQL* и *PHP*, које су две најчешће коришћене технологије за креирање динамичких веб страна. *MySQL* је база података велике брзине, док је *PHP* скриптни језик који се може користити за приступ подацима из базе података. Инсталирањем ове две компоненте локално, програмер може креирати и тестирати динамичну веб страну пре објављивања на јавни веб сервер.

3.1.2 MySQL

MySQL је софтвер отвореног кода који представља систем за управљање релационим базама података (*Relational database management system (RDBMS)*). Његово име је комбинација „*My*“, имена ћерке кооснивача *Michael Widenius*-а, и „*SQL*“, скраћенице за *Structured Query Language*.

MySQL је бесплатни софтвер који је издат под условима *GNU General Public Licence*, а доступан је и под разним власничким лиценцама. *MySQL* је био под власништвом и спонзорством шведске компаније *MySQL AB*, коју је купила компанија *Sun Microsystems* (данас *Oracle Corporation*). 2010. године, када је *Oracle* купио *Sun*, *Widenius* је форсирао *MySQL* пројекат отвореног кода и створио *MariaDB* систем за управљање базама података.

MySQL је компонента софтверског пакета *LAMP(WAMP)*, софтвера за развијање веб апликација (и других), што је акроним за *Linux(Windows)*, *Apache*, *MySQL*, *Perl /PHP/ Python*. *MySQL* користе многе веб апликације које су засноване на коришћењу база података, укључујући, *Drupal*, *Joomla*, *phpBB* и *WordPress*. *MySQL* такође користе и многи популарни веб-сајтови, међу којима су *Facebook*, *Flicker*, *Twitter*, и *YouTube*. На слици 3.2 приказана је конзола *MySQL*-а.

```

c:\wamp64\bin\mysql\mysql5.7.19\bin\mysql.exe
+-----+
15 rows in set (0.00 sec)

mysql> use test
Database changed
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| books           |
| borrow          |
| users           |
+-----+
3 rows in set (0.00 sec)

mysql> select * from books;
+-----+-----+-----+-----+-----+-----+
| book_id | name                | author                | published_at | quantity | available |
+-----+-----+-----+-----+-----+-----+
| 1       | The Testaments      | Margaret Atwood       | 2019-09-01  | 20       | 20       |
| 2       | Talking to Strangers | Malcolm Gladwell      | 2019-07-03  | 10       | 10       |
| 3       | The Water Dancer    | Ta-Nehisi Coates     | 2018-02-19  | 30       | 30       |
| 4       | Things You Save in a Fire | Katherine Center    | 2016-04-25  | 50       | 50       |
| 5       | Naslov1             | Autor1, Autor2, Autor3 | 2019-09-11  | 100      | 100      |
| 6       | Naslov2             | Autor1               | 2019-09-28  | 20       | 20       |
| 7       | Naslov3             | Autor2, Autor3       | 2019-09-01  | 10       | 10       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>

```

Слика 3.2 Конзола MySQL-a

MySQL је написан у C и C++ програмском језику. Ради на многим системским платформама, укључујући AIX, BSDi, FreeBSD, HP-UX, IRIX, Linux, macOS, Microsoft Windows, NetBSD, OpenBSD, OpenSolaris, Oracle Solaris, Simbian, SunOS. У MySQL софтверу постоји и порт за конфигурисање *OpenVMS* виртуалног оперативног система.

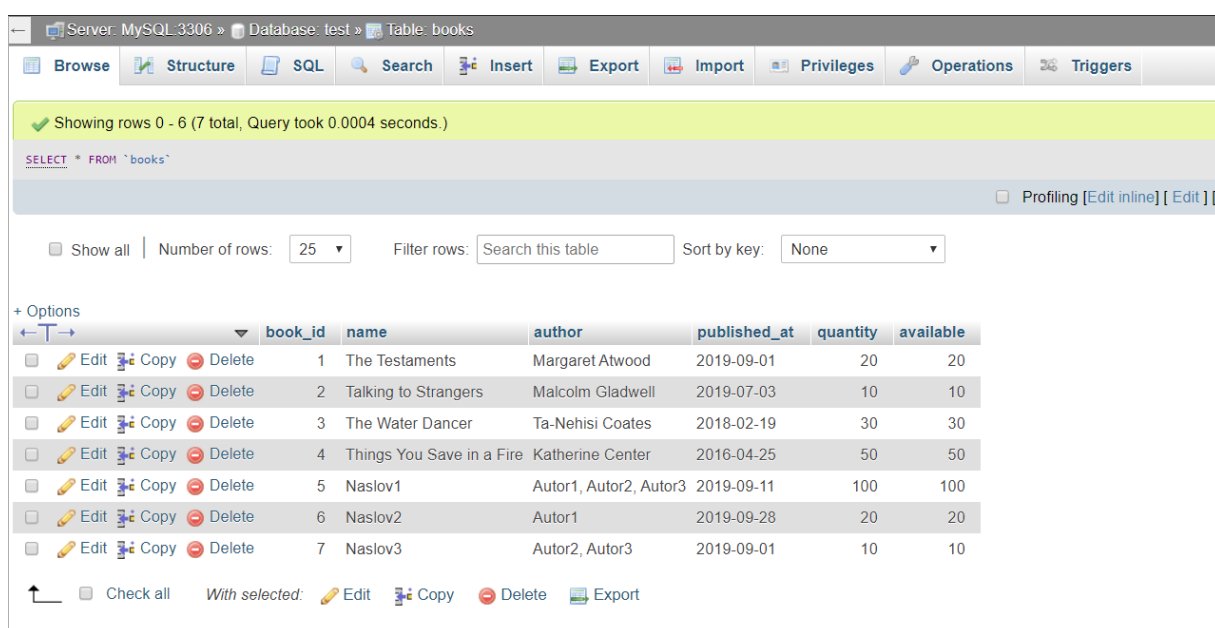
Сам софтвер MySQL сервера као и клијентске библиотеке користе дистрибуцију двоструког лицензирања. Нуде се под *GPL* лиценцом верзије 2, или власничком лиценцом. Подршка се може добити преко званичног приручника. Бесплатна подршка је такође доступна на различитим *IRC (Internet Relay Chat)* каналима и форумима. *Oracle* нуди плаћену подршку путем својих *MySQL Enterprise* производа. Они се разликују по обиму услуга и цени. Поред тога, постоји низ организација трећих страна које пружају подршку и услуге, међу којима су најпопуларнији *MariaDB* и *Percona*.

MySQL је добио позитивне критике и рецензенти су приметили да „изузетно добро функционише у просечном случају“ и да „развојни интерфејси постоје (попут *phpMyAdmin* окружења), а документација је веома добро написана“. Такође је тестирано да је „брз, стабилан и прави вишекориснички сервер са више слојева *SQL* базе података“.

3.1.3 PHPMyAdmin

У склопу *WampServer*-а се налази и *PhpMyAdmin*, интуитивни интерфејс за комуникацију са базом података. *PhpMyAdmin* је бесплатни софтвер написан у *PHP*-у, намењен за управљање администрацијом *MySQL* базе података преко Интернета.

Подржава широк спектар операција над *MySQL* и *MariaDB* системима за управљање базама података. Операције које се често користе (управљање базама података, табелама, колонама, релацијама, индексима, корисницима, дозволама итд.) могу се изводити преко корисничког интерфејса, а поред тога постоји могућност и да се директно извршава било који *SQL* упит. Слика 3.3 илуструје интерфејс апликације.



Слика 3.3 Интерфејс апликације

PhpMyAdmin долази уз обимну документацију, а његови корисници су добродошли да ажурирају све информативне странице како би разменили идеје и начине рада за разне операције. *PhpMyAdmin* је такође врло дубоко документован у књизи “*Mastering phpMyAdmin for Effective MySQL Management*”, која је доступна на енглеском језику. Да би се олакшала употреба широком кругу људи, *PhpMyAdmin* се преводи на 72 језика и подржава *LTR* (*Left-to-right*) и *RTL* (*Right-to-left*) језике. *PhpMyAdmin* је успешан пројекат са стабилном и флексибилном базом кодова. Пројекат *PhpMyAdmin* је део и организације *Software Freedom Conservancy (SFC)*. *SFC* је непрофитна организација која помаже у промовисању, унапређењу, развоју и заступању, *Free, Libre and Open-Source (FLOSS)* пројеката.

3.1.4 PHP

25

PHP: Hypertext Preprocessor је програмски језик опште намене који је првобитно дизајниран за развој интернет апликација. Пројекат је први створио и покренуо *Rasmus Lerdorf* 1994. године док се сада целокупна имплементација изводи од стране "*The PHP Group*" групе. *PHP* је првобитно представљао скраћеницу за личну почетну страницу "*Personal Home Page*", али сада представља рекурзивну скраћеницу *PHP: Hypertext Preprocessor*.

PHP код може да се изврши помоћу интерфејса командне линије (*CLI*), убацивања у *HTML* код или се може користити у комбинацији са различитим системима веб шаблона (*WEB Template Systems*), системима за управљање садржајем на Интернету као и мрежним оквирима (*Web Framework*). *PHP* код обично обрађује *PHP* интерпретер имплементиран као модул на веб серверу или као извршни *Common Gateway Interface (CGI)*. Веб сервер приказује резултате интерпретираног и извршеног *PHP* кода, који могу бити било које врсте података, попут генерисаног *HTML* кода или података бинарне слике. *PHP* се може користити за многе програмске задатке изван веб контекста, као што су самосталне графичке апликације и роботска контрола беспилотних дрона.

Стандардни преводилац *PHP* -а, произведен од стране *Zend Engine*-а, је бесплатни софтвер објављен под *PHP* лиценцом. *PHP* је врло распрострањен програмски језик, користе га многи програмери и може бесплатно да се имплементира и користи на већини веб сервера на скоро сваком оперативном систему и платформи.

Као програмски језик, *PHP* је еволуирао без писане формалне спецификације или стандарда све до 2014. године, при чему се оригинална имплементација користи као стандард који су преузеле друге имплементације. Од 2014. године наставља се рад на стварању формалне *PHP* спецификације.

До септембра 2019. године преко 60% веб-сајтова које користе *PHP* и даље користе застареле верзије 5.6 или старије. Верзије *PHP* програмског језика до 7.1. се више званично не подржавају од стране *The PHP Group* развојног тима, али безбедносну подршку пружају треће стране, као што је *Debian*.

PHP је скриптни језик опште намене који је посебно погодан за веб развој на страни сервера (*Server-side web development*), где *PHP* обично ради на веб серверу. Било који *PHP* код у траженој датотеци извршава се током рада *PHP* -а, обично за креирање динамичког садржаја веб страница или динамичних слика коришћених на веб локацијама. Такође се може користити за скрипте командне линије и апликације графичког корисничког интерфејса (*GUI*) на страни клијента. *PHP* се може имплементирати на већини веб сервера, на многим оперативним системима и платформама, а може се користити са

многим системима за управљање релационим базама података (*RDBMS*). Већина провајдера за веб-хостинг подржава *PHP* како би их њихови клијенти стално користили. Доступан је бесплатно, а *The PHP Group* корисницима пружа комплетни изворни код за изградњу, прилагођавање и проширење за сопствену употребу.

PHP делује углавном као филтер, узимајући улаз из датотеке или ток који садржи текст и/или *PHP* инструкције и емитује други ток података. Излаз ће најчешће бити *HTML* фајл, мада то могу бити *JSON*, *XML* или бинарни подаци попут слике или аудио формата. Од *PHP* -а 4.0, *PHP* парсер компајлира улаз да би произвео бајт код за обраду од стране *Zend Engine*-а, дајући побољшане перформансе у односу на претходне верзије овог интерпретера.

3.2 Laravel 5.8 – PHP Framework

Laravel је бесплатни, *open-source PHP* веб оквир (*framework*), креиран од стране *Taylor Otwell*-а и предвиђен је за развој веб апликација које се заснивају на принципима архитектонског обрасца модел-приказ-контролер (*Model-View-Controller*).

Неке од карактеристика Ларавела су модуларни систем структурирања и паковања ресурса са уграђеним менаџером за управљање зависностима (*Dependency Manager*), могућност различитих начина приступа релационим базама података, помоћни програми који помажу у активирању и одржавању апликација и његова оријентација ка имплементирању најједноставније и најинтуитивније могуће синтаксе програмског језика. Изворни код Ларавела налази се на *GitHub*-у и лиценциран је под условима *MIT* лиценце.

Следеће карактеристике представљају кључне тачке дизајна Ларавела:

- **Bundles** пакети пружају модуларни систем паковања од верзије Ларавела 3.0, а у пакетима су већ доступне функције за једноставно додавање додатних функционалности апликацијама. Поред тога, од верзије Ларавела 4.0 па на даље, користи се *Composer* као менаџер за управљање зависностима (*Dependency Manager*) помоћу ког се додају *PHP* пакети доступни из *Packagist* репозиторијума.
- *Eloquent ORM* (објектно-релацијско пресликавање) је напредна *PHP* имплементација обрасца активног снимања (*active record pattern*), која истовремено нуди уграђене методе за постављање ограничења на односе између објеката базе података. Према принципима обрасца активног снимања, *Eloquent ORM* приказује табеле из база података као класе, а инстанце њиховог објекта везане су за појединачне редове тих табела.
- *Query builder*, доступан од верзије Ларавела 3.0, пружа директнију алтернативу приступу базама података *Eloquent ORM*-у. Уместо да захтева директно писање

SQL упита, Ларавелов “градитељ упита” пружа скуп класа и метода који могу програмски да креирају SQL упите. Такође омогућава селективно кеширање резултата извршених упита.

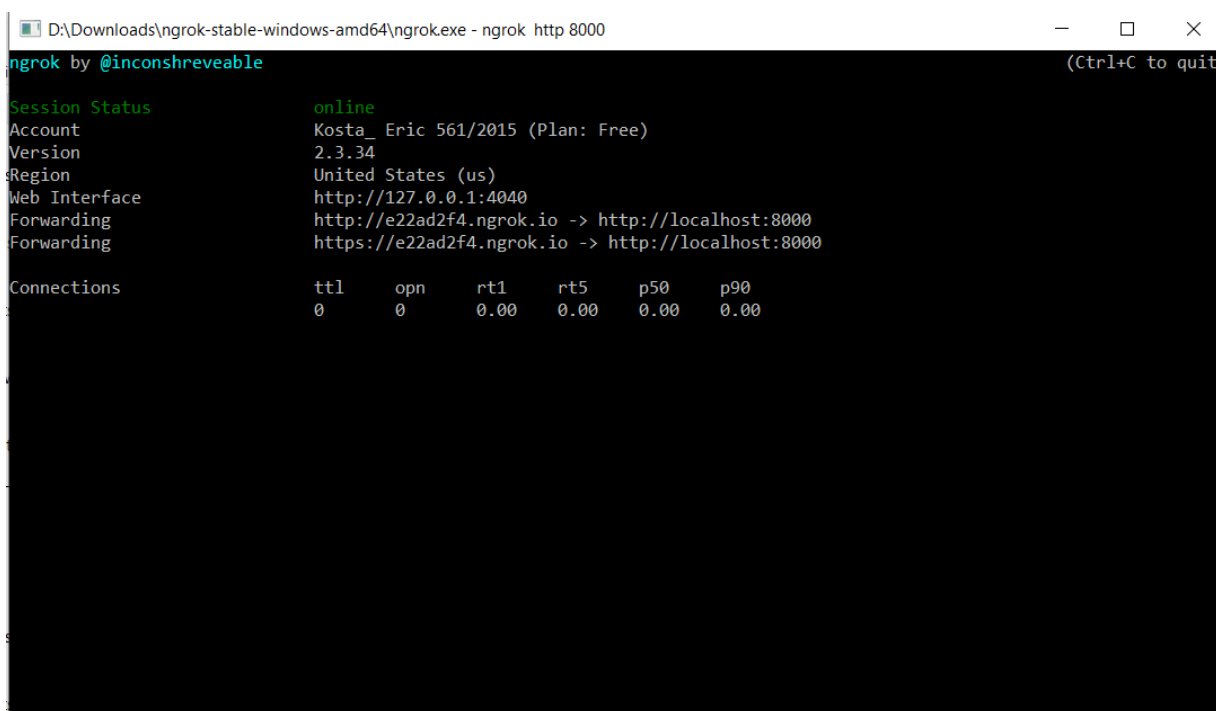
- Логика апликације је саставни део развијених апликација, имплементираних било коришћењем контролера или као део декларације руте.
- *Reverse routing* (обрнуто усмеравање) дефинише однос између веза (линкова) и рута, омогућавајући тако да се евентуалне касније промене рута аутоматски изврше и над одговарајућим линковима. Када се линкови створе помоћу имена постојећих рута, Ларавел аутоматски креира одговарајуће јединствене идентификаторе ресурса (*Uniform resource identifiers - URI*).
- *Restful controllers* (контролери засновани на принципима *REST* модела), пружају опционални начин за одвајање позадинске логике која служи за опслуживање *HTTP GET* и *POST* захтева.
- Аутоматско учитавање класе омогућава аутоматско учитавање *PHP* класа без потребе за ручним одржавањем путања. Учитавање на захтев спречава укључивање непотребних компоненти, тако да се учитавају само компоненте које се заправо користе.
- *View composers* (композитори приказа) служе као прилагодљиве јединице логичког кода које се могу извршити када се учита приказ.
- *Blade templating engine* комбинује један или више образаца са моделом података како би се добили одговарајући прикази, тако што преводи образац у кеширани *PHP* код и тиме омогућава побољшане перформансе. *Blade* такође пружа скуп сопствених контролних структура као што су *if* услови и петље, које се интерно мапирају на одговарајуће *PHP* делове кода. Такође, Ларавел услуге могу да се позивају из *Blade* шаблона, а сам уређај за обраду шаблона може да се прошири кориснички дефинисаним директивама.
- *IoC containers* омогућавају стварање нових објеката применом принципа инверзије контроле (*Inversion of Control*), у ком *framework* позива код који се односи на извршавање апликације или задатка, са опционим инстанцирањем и референцирањем нових објеката на начин који прати *singleton* образац.
- *Migrations* пружају систем контроле верзија за шеме база података, што омогућава једноставно извршавање промена у кодној бази апликације и потребних промена у изгледу базе података. Као резултат, ова функција поједностављује употребу и ажурирање апликација заснованих на Ларавелу.
- *Database seeding* омогућава начин попуњавања табела база података одабраним подразумеваним подацима који се могу користити за тестирање апликација или се изводе као део почетног подешавања апликације.

- *Unit testing* (модуларно тестирање) обезбеђено је као саставни део Ларавела, који сам по себи садржи модуларне тестове који откривају и спречавају регресију у оквиру самог *framework*-а. Модуларни тестови могу се изводити кроз обезбеђени алат за командну линију.
- Аутоматска пагинација поједностављује задатак имплементације пагинације, замењујући уобичајене ручне приступе са имплементацијом аутоматизованим методама интегрисаним у Ларавел.

3.3 NGROK

NGROK омогућава да се на интернету изложи веб сервер који ради на локалној машини. Та функционалност се обезбеђује креирањем тунела (*tunneling*), где се у самој конзоли *NGROK*-а подешава порт који слуша веб сервер.

NGROK пружа веб окружење које ради у реалном времену на којем је могуће увидети сав *HTTP* саобраћај који пролази кроз генерисане тунеле. Када се *NGROK* покрене, на интернет страни *localhost:4040* се приказују детаљи захтева, укључујући време, трајање, *IP* извор, заглавља, параметре упита и као и оптерећење захтева. На слици 3.4 приказана је конзола *NGROK* –а.



```

D:\Downloads\ngrok-stable-windows-amd64\ngrok.exe - ngrok http 8000
ngrok by @inconshreveable (Ctrl+C to quit)

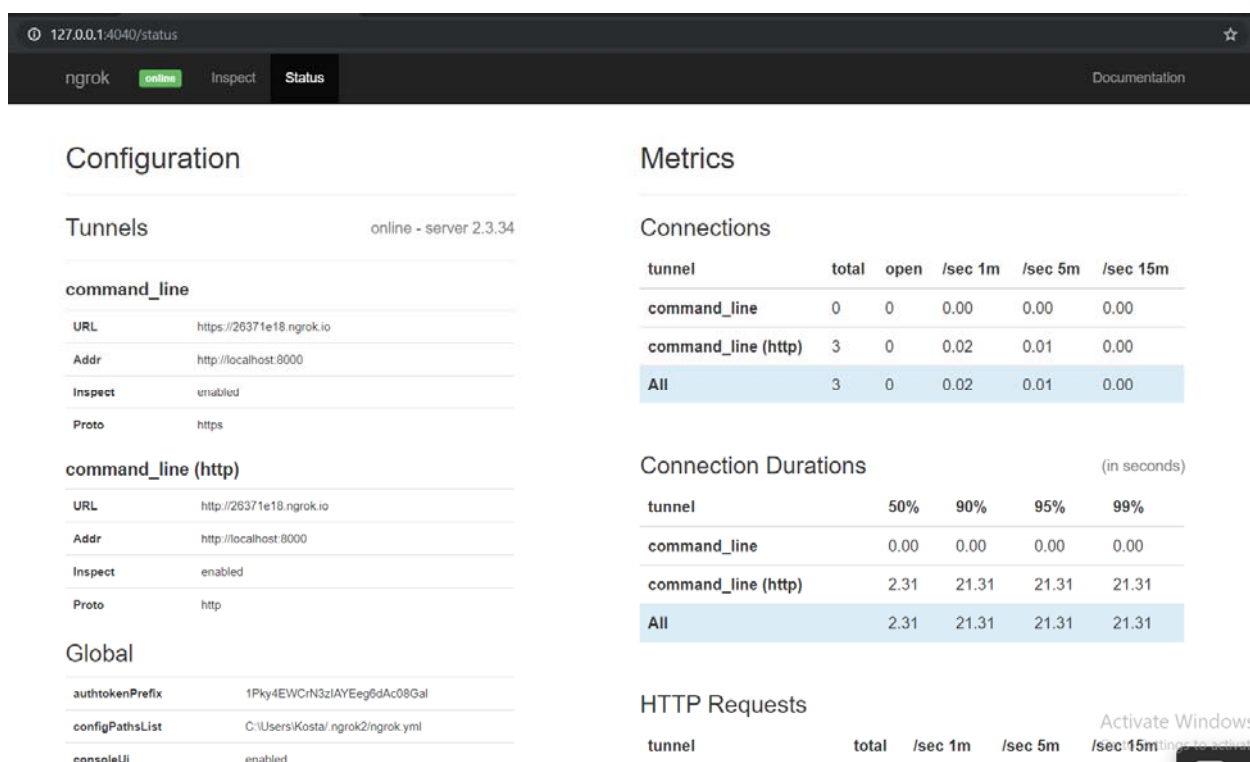
Session Status      online
Account             Kosta_Eric 561/2015 (Plan: Free)
Version             2.3.34
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://e22ad2f4.ngrok.io -> http://localhost:8000
Forwarding           https://e22ad2f4.ngrok.io -> http://localhost:8000

Connections
t1l   opn   rt1   rt5   p50   p90
0     0     0.00  0.00  0.00  0.00
    
```

Слика 3.4 Конзола *NGROK*-а

NGROK додељује случајна хексадецимална имена *HTTP* тунелима који се отварају приликом коришћења, што задовољава једнократну личну употребу. Али ако се *URL* интегрише са веб-куком (*webhook*) треће стране, може бити веома проблематично ако се име тунела промени или га је тешко прочитати. Помоћу команде *-subdomain* могуће је да се одреди прилагођени поддомен за УРЛ тунел.

NGROK клијент испоручује моћан интерфејс за преглед у реалном времену који омогућава прегледање саобраћаја који се шаље серверу апликације и које одговоре сервер враћа. Сваки *HTTP* захтев који се шаље кроз тунеле се приказује у инспекцијском интерфејсу. Интерфејс за инспекцију има неколико ограничења. Ако је тело ентитета предуго, *NGROK* може хватати само почетни део тела захтева. Поред тога, *NGROK* не приказује привремених 100 одговора од сервера. Слика 3.5 илуструје *NGROK* конфигурацију.



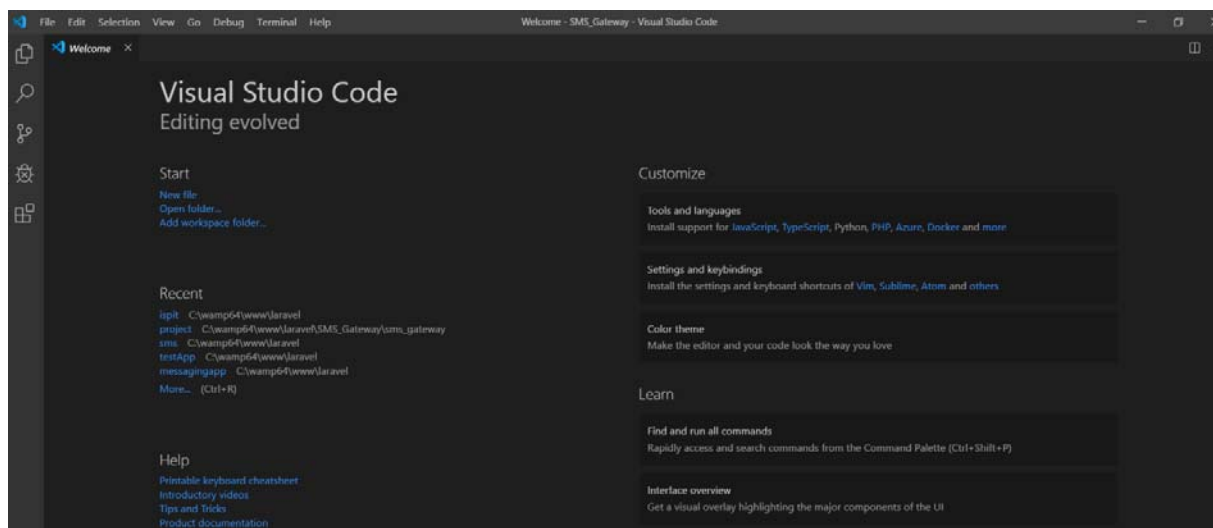
Слика 3.5 *NGROK* конфигурација

NGROK има посебну подршку за најчешће формате размене података који се користе на вебу. Било који *XML* или *JSON* податак у телима за захтев или одговор аутоматски се штампају где се и проверавају грешке у синтакси. *SSH reverse tunneling* је алтернативни механизам за покретање *NGROK* тунела без потребе чак и за преузимањем или покретањем *NGROK* клијента. Тунели се могу покренути путем *SSH*-а без преузимања *NGROK* клијента покретањем *SSH* наредбе за повратни тунел.

Функцију *SSH* пролаза не треба мешати са излагањем *SSH* сервера преко *NGROK*-а. Популарни веб сервери као што су *MAMP* и *WAMP* ослањају се на технику "*Virtual Hosting*" што значи да они консултују заглавље *HTTP Host* захтева да одреде коју од својих локација треба да послужују. Да би се изложила локација, могуће је и захтевање од *NGROK* -а да напише *Host* заглавље свих захтева који се шаљу преко тунела да се подударају са оним што веб сервер очекује. То може да се учини помоћу опције *-host-header* где се специфицира који виртуелни *Host* се циља.

3.4 Visual Studio Code

Visual Studio Code је уређивач изворног кода (*source-code editor*) који је развио *Microsoft* за *Windows*, *Linux* и *macOS* оперативне системе. Укључује подршку за уклањање грешака, уграђену *Git* и *GitHub* контролу, истицање синтаксе, интелигентно довршавање кода, шаблоне и рефакторисање кода. Веома је прилагодљив, омогућава корисницима да мењају тему, пречице на тастатури, поставке и инсталирају додатке који додају додатну функционалност. Изворни код је бесплатан и отворен је и пушта се под *MIT* лиценцом. *Visual Studio Code* заснован је на *Electron*-у, *framework*-у који се користи за креирање *Node.js* апликација, чије се функционисање заснива на *Blink Layout Engine* прегледачкој машини. На слици 3.6 приказан је кориснички интерфејс *Visual Studio Code*-а.

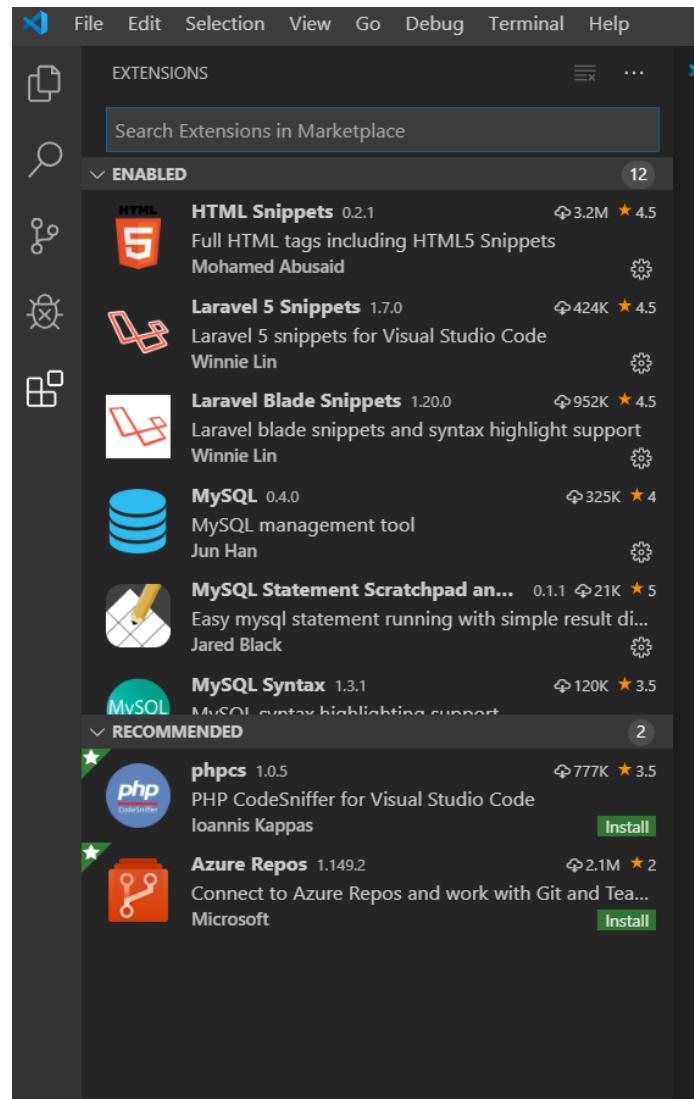


Слика 3.6 Кориснички интерфејс *Visual Studio Code*-а

У анкети за програмере на *Stack Overflow* форуму 2019. године, *Visual Studio Code* рангиран је као најпопуларније окружење за програмере, а 50,7% од 87.317 испитаника је тврдило да га користи. *Visual Studio Code* је уређивач изворног кода који се може

користити са различитим програмским језицима. Уместо пројектног система омогућава корисницима отварање једног или више директоријума, који се затим могу организовати у радне просторе за будућу употребу. То му омогућава да ради као језично-агностички уређивач кода за било који језик, супротно *Miscrosoft Visual Studio* окружењу које користи *.sln* фајлове и пројектне фајлове који су стриктно везани за дати пројекат.

Подржава бројне програмске језике и скуп функција које се разликују по језику. Нежељене датотеке и мапе могу се искључити из стабла пројекта путем поставки. Многе функције *Visual Studio Code*-а нису изложене кроз меније или корисничко окружење, већ им се може приступити путем палете наредби. *Visual Studio Code* може да се прошири помоћу додатака који су доступни и могу да се преузму са централног репозиторијума. Ово укључује додатке за уређивач и подршке за програмске језике. Значајна карактеристика је могућност креирања додатака који додају подршку за нове језике, теме, исправљаче грешака као и извршавање статичке анализе кода (слика 3.7).



Слика 3.7 Кориснички интерфејс за додатне пакете

Visual Studio Code укључује вишеструка проширења за *FTP (File Transfer Protocol)*, омогућавајући да се софтвер користи као бесплатна алтернатива за веб развој. Код се може синхронизовати између уређивача и сервера, без преузимања додатног софтвера. Такође омогућава корисницима да поставе кодну страницу на којој је сачуван активни документ, знак нове линије за *Windows / Linux* и програмски језик активног документа. То му омогућава да се користи на било којој платформи, у било којем локалитету и за било који одређени програмски језик.

4. МОДЕЛ БАЗА ПОДАТАКА И UML ДИЈАГРАМИ

4.1 Модел база података

Модел база података и базе података представљају структуру и стање реалног систем у неком тренутку времена садашњег и прошлог, које такође мора да задовоље и неке будуће захтеве. Из база података се користе подаци односно информације које су по обиму, садржају и начину приказивања у највећој мери прилагођене корисницима. Приступ подацима се остварује на различите начине и треба да омогући брзо проналажење и обраду у редоследу који најчешће не одговара редоследу уноса. На слици 4.1 приказан је модел база података за систем управљања магацинским пословањем. Релације (везе) између појединих табела су приказане на моделу, а структура појединих табела је приказана кроз следећи sql код креирања истих, са примарним и секундарним индексима и ограничењима:

```
-- phpMyAdmin SQL
-- version 4.7.4
-- https://www.phpmyadmin.net/
-- Host: 127.0.0.1:3306
-- Server version: 5.7.19
-- PHP Version: 7.1.9
-- Database: `storage_management`
-- Table structure for table `branches`
CREATE TABLE IF NOT EXISTS `branches` (
  `branch_id` bigint(20) UNSIGNED NOT NULL,
  `company_id` bigint(20) UNSIGNED NOT NULL,
  `branch_name` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`branch_id`),
  KEY `branches_company_id_foreign` (`company_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `companies`
CREATE TABLE IF NOT EXISTS `companies` (
  `company_id` bigint(20) UNSIGNED NOT NULL,
  `company_name` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `street_address` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `zip_code` varchar(5) COLLATE utf8mb4_unicode_ci NOT NULL,
  `bank_account` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `phone_number` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `PIB` varchar(9) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`company_id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `documents`
CREATE TABLE IF NOT EXISTS `documents` (
  `document_id` bigint(20) UNSIGNED NOT NULL AUTO_INCREMENT,
  `supplier_customer_id` bigint(20) UNSIGNED NOT NULL,
  `document_number` int(11) NOT NULL,
  `total` decimal(10,2) NOT NULL,
  `document_type` enum('prijemnica','otpremnica','narudzbenica') COLLATE
utf8mb4_unicode_ci NOT NULL,
  `source_document_number` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL,
  `document_date` date NOT NULL,
  `entry_date` date NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`document_id`),
  KEY `documents_supplier_customer_id_foreign` (`supplier_customer_id`),
  KEY `documents_document_type_foreign` (`document_type`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `document_types`
CREATE TABLE IF NOT EXISTS `document_types` (
  `document_type` enum('prijemnica','otpremnica','narudzbenica') COLLATE
utf8mb4_unicode_ci NOT NULL,
  `document_number` int(11) NOT NULL,
  `description` varchar(150) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`document_type`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `items`
CREATE TABLE IF NOT EXISTS `items` (
  `document_id` bigint(20) UNSIGNED NOT NULL,
  `storage_id` bigint(20) UNSIGNED NOT NULL,
  `product_id` bigint(20) UNSIGNED NOT NULL,
  `quantity` decimal(12,3) NOT NULL,
  `price_on_invoice` decimal(15,2) NOT NULL,
  `discount` decimal(8,2) NOT NULL,
  `invoice_price` decimal(15,2) NOT NULL,
  `retail_price` decimal(15,2) NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  KEY `items_document_id_foreign` (`document_id`),
  KEY `items_product_id_foreign` (`product_id`),
  KEY `items_storage_id_product_id_foreign` (`storage_id`,`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `migrations`
CREATE TABLE IF NOT EXISTS `migrations` (
  `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
```

```

`migration` varchar(191) COLLATE utf8mb4_unicode_ci NOT NULL,
`batch` int(11) NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=146 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `products`
CREATE TABLE IF NOT EXISTS `products` (
  `product_id` bigint(20) UNSIGNED NOT NULL,
  `tax_id` bigint(20) UNSIGNED NOT NULL,
  `product_name` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `product_type` enum('gotov_proizvod','repro_materijal') COLLATE utf8mb4_unicode_ci NOT
NULL,
  `unit_of_measure` varchar(6) COLLATE utf8mb4_unicode_ci NOT NULL,
  `transport_package` int(11) NOT NULL,
  `invoice_price` decimal(10,2) NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`product_id`),
  KEY `products_tax_id_foreign` (`tax_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `stamps`
CREATE TABLE IF NOT EXISTS `stamps` (
  `stamp_id` bigint(20) UNSIGNED NOT NULL,
  `stamp_name` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `tax_id` bigint(20) UNSIGNED NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`stamp_id`),
  KEY `stamps_tax_id_foreign` (`tax_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `storages`
CREATE TABLE IF NOT EXISTS `storages` (
  `storage_id` bigint(20) UNSIGNED NOT NULL,
  `branch_id` bigint(20) UNSIGNED NOT NULL,
  `storage_name` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
  `storage_type`
enum('skladiste_gotovih_proizvoda','skladiste_repro_materijala','medjuskладiste') COLLATE
utf8mb4_unicode_ci NOT NULL,
  `special_requirements` varchar(150) COLLATE utf8mb4_unicode_ci NOT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`storage_id`),
  KEY `storages_branch_id_foreign` (`branch_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Table structure for table `supplier_customers`
CREATE TABLE IF NOT EXISTS `supplier_customers` (
  `supplier_customer_id` bigint(20) UNSIGNED NOT NULL,
  `supplier_customer_name` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,

```

```
`supplier_customer_type` enum('dobavljac','kupac') COLLATE utf8mb4_unicode_ci NOT NULL,
`street_address` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
`zip_code` varchar(5) COLLATE utf8mb4_unicode_ci NOT NULL,
`bank_account` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
`phone_number` varchar(15) COLLATE utf8mb4_unicode_ci NOT NULL,
`email` varchar(30) COLLATE utf8mb4_unicode_ci NOT NULL,
`PIB` varchar(9) COLLATE utf8mb4_unicode_ci NOT NULL,
`created_at` timestamp NULL DEFAULT NULL,
`updated_at` timestamp NULL DEFAULT NULL,
PRIMARY KEY (`supplier_customer_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Table structure for table `s_p_s`
CREATE TABLE IF NOT EXISTS `s_p_s` (
  `storage_id` bigint(20) UNSIGNED NOT NULL,
  `product_id` bigint(20) UNSIGNED NOT NULL,
  `quantity` decimal(12,3) NOT NULL,
  `z_min` decimal(12,2) DEFAULT NULL,
  `q_dn` decimal(12,2) DEFAULT NULL,
  `q_god` decimal(12,2) DEFAULT NULL,
  `v_nab` int(11) DEFAULT NULL,
  `d` int(11) DEFAULT NULL,
  `z_max` decimal(12,2) DEFAULT NULL,
  `p_max` decimal(12,2) DEFAULT NULL,
  `d_max` int(11) DEFAULT NULL,
  `nd` int(11) DEFAULT NULL,
  `q_opt` decimal(12,2) DEFAULT NULL,
  `q_p1` decimal(12,2) DEFAULT NULL,
  `t` decimal(12,2) DEFAULT NULL,
  `st` decimal(5,2) DEFAULT NULL,
  `ks` decimal(5,2) DEFAULT NULL,
  `n_opt` int(11) DEFAULT NULL,
  `vrednost` decimal(12,2) DEFAULT NULL,
  `kumulativa` decimal(12,2) DEFAULT NULL,
  `procenat` decimal(5,2) DEFAULT NULL,
  `abc_rang` varchar(1) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `starting_supply` decimal(12,2) DEFAULT NULL,
  `created_at` timestamp NULL DEFAULT NULL,
  `updated_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`storage_id`,`product_id`),
  KEY `s_p_s_product_id_foreign` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Table structure for table `taxes`
CREATE TABLE IF NOT EXISTS `taxes` (
  `tax_id` bigint(20) UNSIGNED NOT NULL,
  `start_date` date NOT NULL,
  `tax1` decimal(6,2) NOT NULL,
```

```

`tax2` decimal(6,2) NOT NULL,
`tax3` decimal(6,2) NOT NULL,
`created_at` timestamp NULL DEFAULT NULL,
`updated_at` timestamp NULL DEFAULT NULL,
PRIMARY KEY (`tax_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
-- Constraints for table `branches`
ALTER TABLE `branches`
  ADD CONSTRAINT `branches_company_id_foreign` FOREIGN KEY (`company_id`) REFERENCES
`companies` (`company_id`) ON DELETE CASCADE ON UPDATE CASCADE;
-- Constraints for table `documents`
ALTER TABLE `documents`
  ADD CONSTRAINT `documents_document_type_foreign` FOREIGN KEY (`document_type`)
REFERENCES `document_types` (`document_type`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `documents_supplier_customer_id_foreign` FOREIGN KEY
(`supplier_customer_id`) REFERENCES `supplier_customers` (`supplier_customer_id`) ON
DELETE CASCADE ON UPDATE CASCADE;
-- Constraints for table `items`
ALTER TABLE `items`
  ADD CONSTRAINT `items_document_id_foreign` FOREIGN KEY (`document_id`) REFERENCES
`documents` (`document_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `items_product_id_foreign` FOREIGN KEY (`product_id`) REFERENCES
`products` (`product_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `items_storage_id_product_id_foreign` FOREIGN KEY
(`storage_id`, `product_id`) REFERENCES `s_p_s` (`storage_id`, `product_id`) ON DELETE
CASCADE ON UPDATE CASCADE;
-- Constraints for table `products`
ALTER TABLE `products`
  ADD CONSTRAINT `products_tax_id_foreign` FOREIGN KEY (`tax_id`) REFERENCES `taxes`
(`tax_id`) ON DELETE CASCADE ON UPDATE CASCADE;
-- Constraints for table `stamps`
ALTER TABLE `stamps`
  ADD CONSTRAINT `stamps_tax_id_foreign` FOREIGN KEY (`tax_id`) REFERENCES `taxes`
(`tax_id`) ON DELETE CASCADE ON UPDATE CASCADE;
-- Constraints for table `storages`
ALTER TABLE `storages`
  ADD CONSTRAINT `storages_branch_id_foreign` FOREIGN KEY (`branch_id`) REFERENCES
`branches` (`branch_id`) ON DELETE CASCADE ON UPDATE CASCADE;
-- Constraints for table `s_p_s`
ALTER TABLE `s_p_s`
  ADD CONSTRAINT `s_p_s_product_id_foreign` FOREIGN KEY (`product_id`) REFERENCES
`products` (`product_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `s_p_s_storage_id_foreign` FOREIGN KEY (`storage_id`) REFERENCES
`storages` (`storage_id`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

```



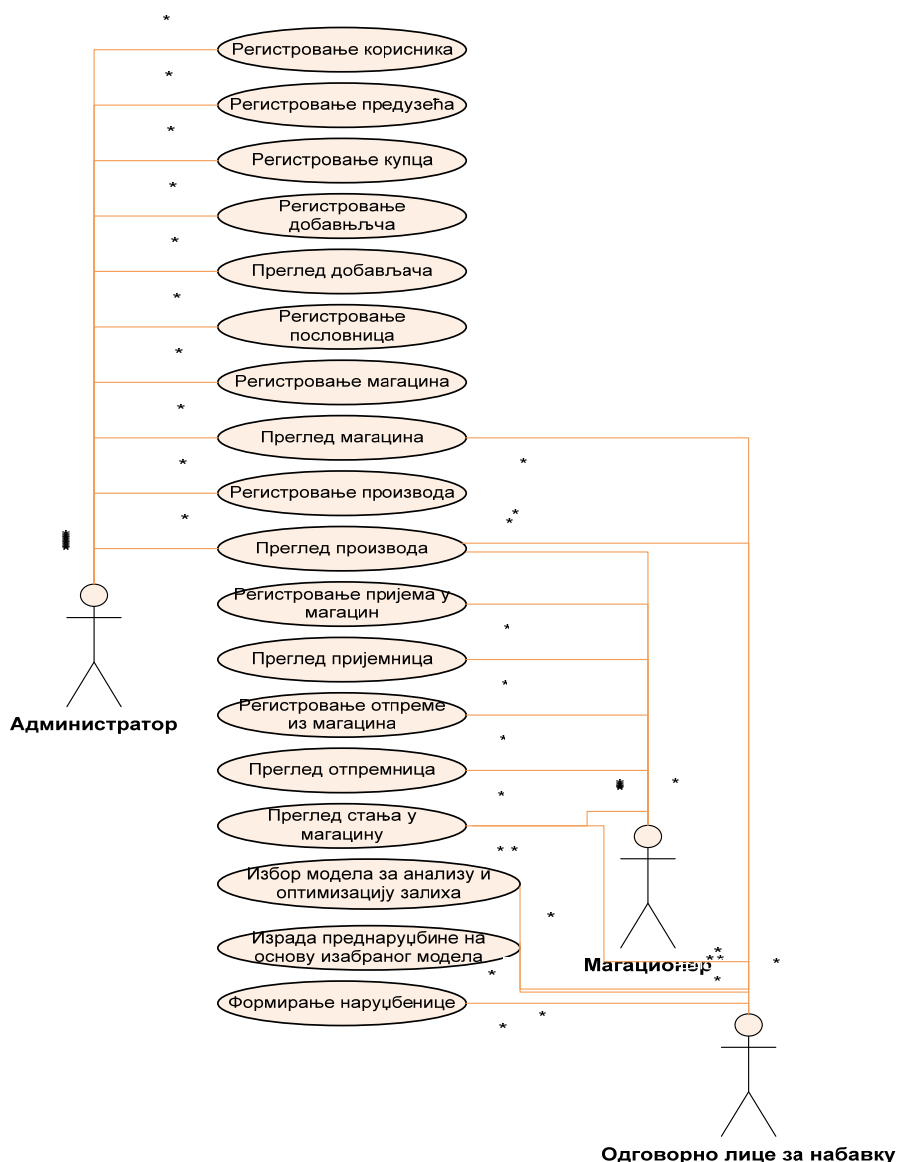
Activate Windows

Go to Settings to activate Windows.

4.2 Идентификација учесника (Actors) и случајева коришћења (Use case)

39

Концепт сценарија коришћена система приказује скуп случајева коришћења и актера (учесника) односно моделира дијалог између учесника и система. Дијаграм случајева коришћења визуелизује понашање система, подсистема, класа и интерфејса и специфицира њихове функционалности. Скуп случајева коришћења за систем управљања магацинским пословањем приказан на слици 4.2, успоставља све дефинисане начине на које систем може бити коришћен. За остварење циљеви система управљање магацинским пословањем дефинисани су следећи учесници који ће интерактивно комуницирати са системом: администратор, магационер, одговорно лице за набавку.

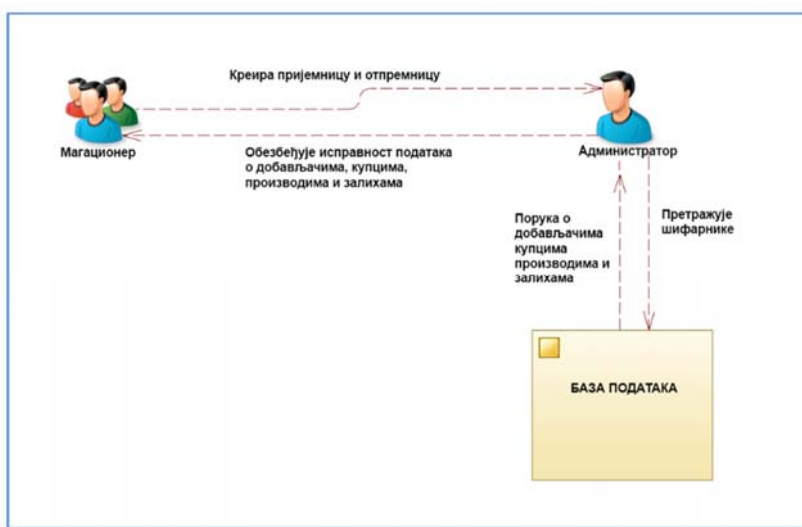


Слика 4.2 Use case дијаграм

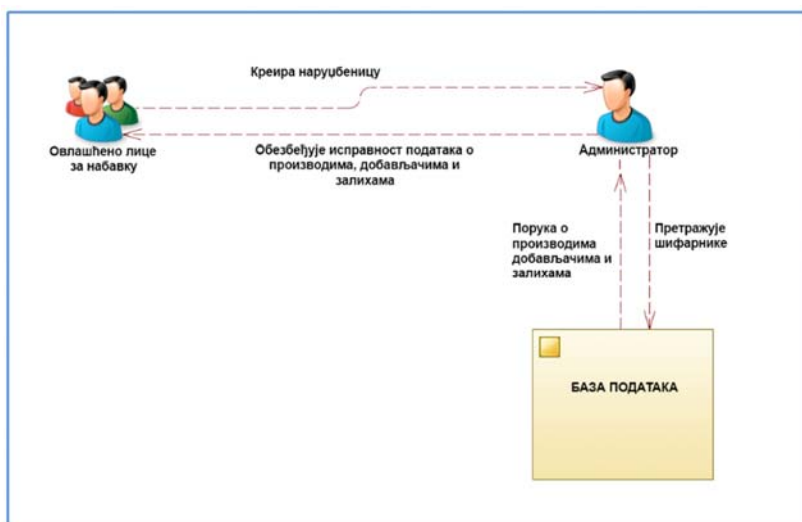
4.3 Дијаграм сарадње (Communication diagram)

40

Дијаграм сарадње приказује интеракцију објеката који остварују понашање одређеног случаја коришћења или дела случаја коришћења. Користе га пројектанти ради дефинисања и објашњавања улога објеката који извршавају одређени ток догађаја случаја коришћења. Они су примарни извор информација за одређивање одговорности класа и њихових односа. Због свог формата погоднији су за описивање једноставних интеракција малог броја објеката. Како број објеката и порука почне да расте, тада дијаграм сарадње постаје тешко читљив. Дијаграм сарадње за систем управљања магацинским пословањем приказан је сликама 4.3 и 4.4.



Слика 4.3 Дијаграм сарадње – одржавање стања у магацину

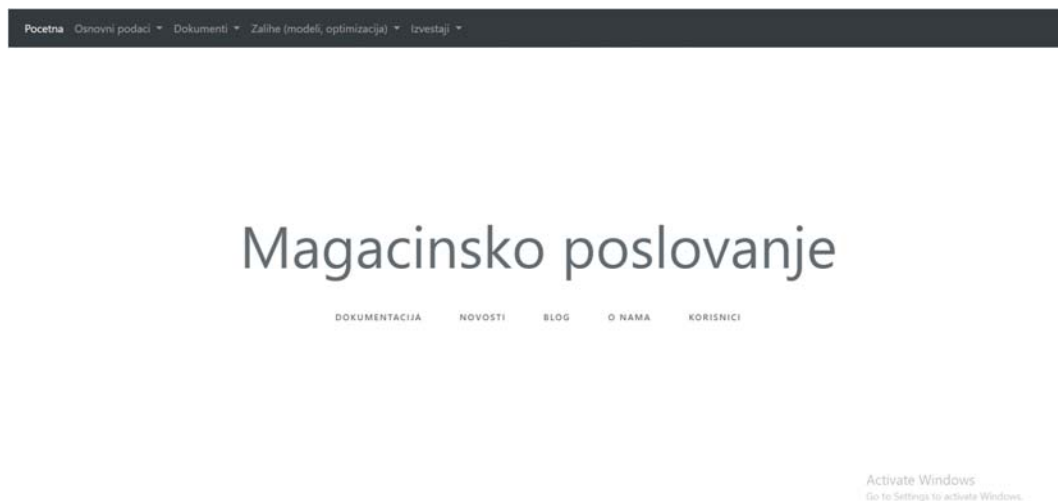


Слика 4.4 Дијаграм сарадње – креирање наруџенице

5. ДИЗАЈН И ОПИС СОФТВЕРСКОГ РЕШЕЊА

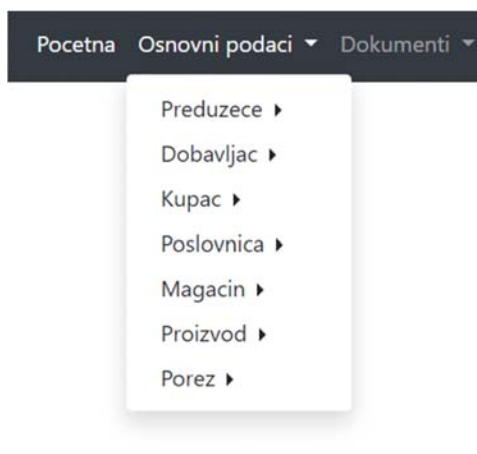
41

Приказ почетне стране веб апликације се налази на слици 5.1. На страни се налази назив апликације као и корисни линкови, док се на врху ове, као и свих осталих страна налази заглавље које води корисника кроз целокупну апликацију и обезбеђује пуну функционалност.



Слика 5.1 Приказ почетне стране апликације

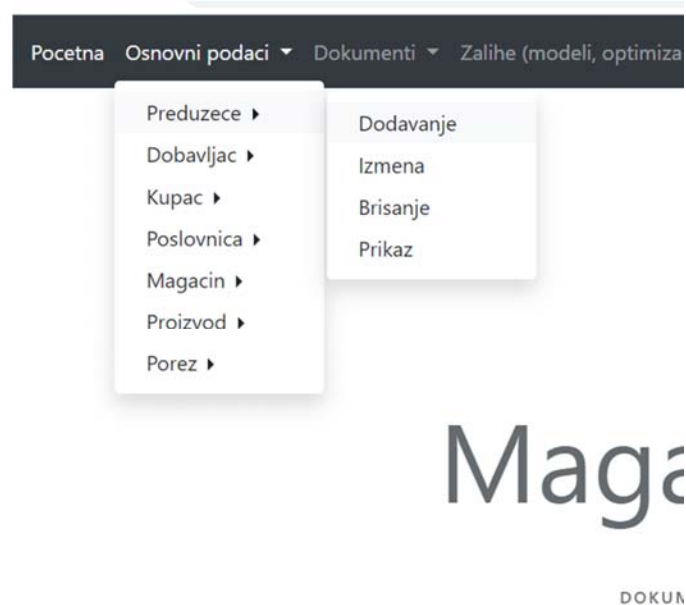
У заглављу се налазе четири падајуће листе које представљају главне секције апликације. На првој падајућој листи су приказани сви основни модели (подаци) који се користе у апликацији (слика 5.2).



Слика 5.2 Приказ картице основних података

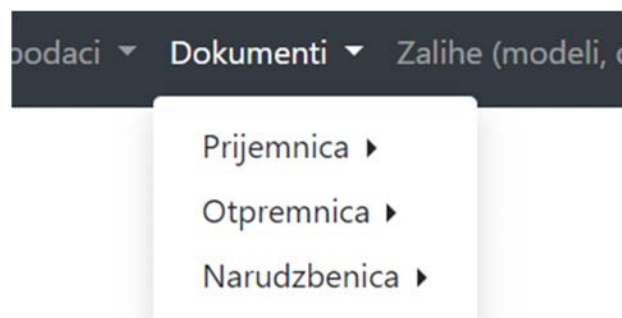
Над сваким моделом је могуће извршити све операције предвиђене *CRUD*-ом, а то су додавање, ажурирање, брисање и приказ (слика 5.3).

42



Слика 5.3 Операције над моделима

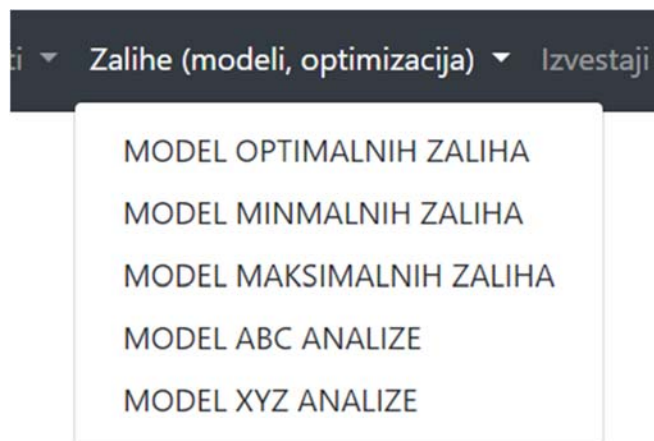
Друга секција (падајућа листа) води корисника ка подсекцијама које омогућавају извршавање свих претходно поментуиx операција над главним документима који се користе приликом управљања магацинским пословањем, а то су пријемница, отпремница и наруџбеница (слика 5.4).



Слика 5.4 Приказ основних докумената

Трећа секција излистава неке од поменутих модела прорачуна, где се кликом на један (или више) модел покреће позадински процес који врши израчунавање оптималних количина за дати модел над постојећом базом производа (слика 5.5).

43



Слика 5.5 Картица модела прорачуна

На слици 5.6 је приказана форма за унос предузећа.

A screenshot of a web form for entering company data. At the top is a dark navigation bar with 'Dokumenti', 'Zalihe (modeli, optimizacija)', and 'Izvestaji'. The form itself has several input fields: 'Sifra preduzeca' (containing 'Sifra preduzeca'), 'Ime preduzeca' (containing 'Ime preduzeca'), 'Adresa' (containing 'Karadjordjeva 43/1 Kragujevac'), 'Postanski broj' (containing '34000'), 'Broj ziro racuna' (containing 'xxx-yyy-yyyy-yyyy-yy-zz'), 'PIB' (containing 'PIB'), 'Broj telefona' (containing '+381698288758'), and 'E-mail adresa' (containing 'kgkosta360@gmail.com'). At the bottom left is a dark button labeled 'Unesi preduzece'.

Слика 5.6 Форма за унос предузећа

Након попуњавања форме (слика 5.7) и клика на дугме унеси предузеће,

44

podaci ▾ Dokumenti ▾ Zalihe (modeli, optimizacija) ▾ Izvestaji ▾

Sifra preduzeca: 541326 Ime preduzeca: Preduzece 1

Adresa: Ulica 456 Postanski broj: 34000

Broj ziro racuna: 123-1231231231234-25 PIB: 456859652

Broj telefona: +381652565885 E-mail adresa: preduzece1@gmail.com

Unesi preduzece

Activate

Слика 5.7 Попуњена форма за унос предузећа

приказује се порука о статусу уноса, позитивна уколико је предузеће успешно унето у базу и негативна уколико је дошло до неке грешке приликом уноса (слика 5.8).

Preduzece uspesno uneto

Sifra preduzeca: Sifra preduzeca Ime preduzeca: Ime preduzeca

Adresa: Karadjordjeva 43/1 Kragujevac Postanski broj: 34000

Broj ziro racuna: xxx-yyy-yyyy-yyyy-yy PIB: PIB

Broj telefona: +381698288758 E-mail adresa: kgkosta360@gmail.com

Unesi preduzece

Activa

Слика 5.8 Форма са приказаним повратним статусом

Идентичне форме се налазе и на странама за измену и брисање предузећа, као и на странама за обрађивање осталих основних објеката модела. Одабиром падајуће листе на другој картици заглавља, кориснику се даје могућност да врши додавање, измену, брисање или приказ неког од докумената. На слици 5.9 је приказана веб страна за додавање документа пријемнице.

45

Слика 5.9 Форма за додавање пријемнице

Од корисника се очекује да прво унесе информације о заглављу пријемнице, након чега се откључава форма за унос артикала (слика 5.10).

Слика 5.10 Форма за унос артикала

Након уноса свих артикала са изворног документа, кликом на дугме *унеси пријемницу* се дате информације уносе у базу података (слика 5.11).

46

Artikli

Sifra proizvoda Kolicina

Cena na fakturi Popust Nabavna cena

Unesi artikal

#	Naziv	Kolicina	Cena	Ukupno
10	Proizvod 10	500	200	100000
7	Proizvod 7	300	150	45000
4	Proizvod 4	1200	120	144000

TOTAL

289000

Unesi prijemnicu

Activate 1
Go to Setting

Слика 5.11 Унос пријемнице

Одабиром стране за приказ пријемнице (слика 5.12),

Dokumenti Zalihe (modeli, optimizacija) Izvestaji

Prijemnica ▶

Otpremnica ▶

Narudzbenica ▶

Dodavanje

Izmena

Brisanje

Prikaz

Sifra prijemnice

Sifra prijemnice

Prikazi prijemnicu

Слика 5.12 Форма за приказ пријемнице

приказује се форма која са састоји од падајуће листе свих постојећих пријемница где се кориснику даје могућност да одабере одређену за приказ (слика 5.13).

Sifra prijemnice

Broj dokumenta: 3 / Broj prijemnice: 2 ▼

Prikazi prijemnicu

Dobavljaц: Dobavljaц 4
Mesto i adresa: Adresa 4 Postanski broj: 18000
PIB dobavljaца: 000000004
Tekuci racun: 111-1111111111111-14
Email adresa: dobavljaц4@gmail.com
Broj telefona: +381611111114

Kupac: Preduzece 4
Mesto i adresa: Save Kovacevica 35
Postanski broj: 34000 PIB dobavljaца: 811552349
Tekuci racun: 363-9863365281953-69
Email adresa: preduzece4@hotmail.com
Broj telefona: +381645246726

PRIJEMNICA Br: 2
Datum izdavanja: 2019-09-11

#	Sifra artikla	Naziv	Jedinica mere	Kolicna	Cena	Iznos
1	10	Proizvod 10	Jm2	500.00	200.00	100000
2	7	Proizvod 7	Jm2	300.00	150.00	45000
3	4	Proizvod 4	Jm1	1200.00	120.00	144000
TOTAL:						289000.00

Robu izdao: _____ Robu primio: _____

Слика 5.13 Приказ пријемнице

Идентична процедура се извршава приликом додавања и приказа документа отпремнице. Приказ једног таквог документа је дат на слици 5.14.

Sifra otpremnice
Broj dokumenta: 2 / Broj otpremnice: 1

Prikaži otpremnicu

Dobavljaц: Preduzece 4

Mesto i adresa: Save Kovacevica 35 Postanski broj: 34000

PIB dobavlјaca: 811552349

Tekuci racun: 363-9863365281953-69

Email adresa: preduzece4@hotmail.com

Broj telefona: +381645246726

Kupac: Dobavlјac 1

Mesto i adresa: Adresa 1 Postanski broj: 11000

PIB dobavlјaca: 000000001

Tekuci racun: 111-1111111111111-11

Email adresa: dobavlјac1@gmail.com

Broj telefona: +381611111111

OTPREMNICA Br: 1

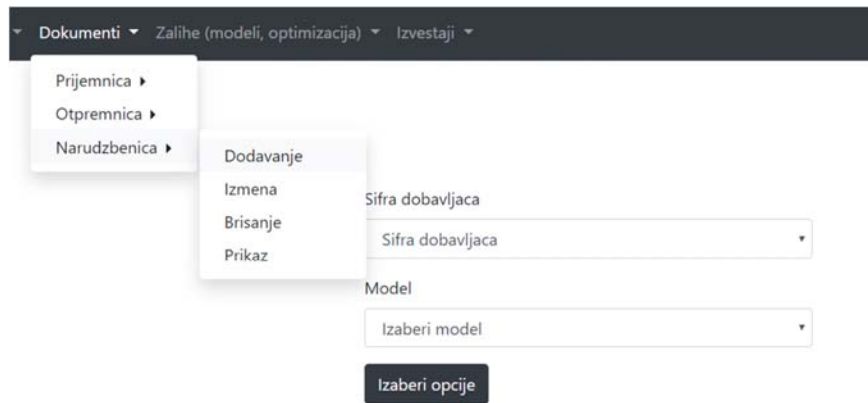
Datum izdavanja: 2018-06-21

#	Sifra artikla	Naziv	Jedinica mere	Kolicna	Cena	Iznos
1	1	Proizvod 1	Jm1	500.00	73.00	36500
2	2	Proizvod 2	Jm1	1000.00	37.00	37000
3	3	Proizvod 3	Jm1	1500.00	52.00	78000
4	4	Proizvod 4	Jm1	2000.00	93.00	186000
5	5	Proizvod 5	Jm1	2500.00	87.00	217500
TOTAL:						555000.00

Robu izdao: _____
Robu primio: _____

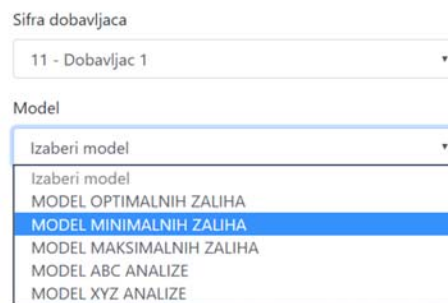
Слика 5.14 Приказ отпремнице

Одабиром стране за додавање наруџбенице из картице документи, кориснику се представља форма у овирu које треба попунити два поља. Прво поље је одабир добављача коме ће бити послата наруџбеница, а друго поље представља избор модела по ком корисник жели да добије предложене количине производа које треба наручити (слика 5.15).



Слика 5.15 Одабир информација о наруџбеници

У опцијају су модели који су претходно описани у раду (слика 5.16).



Слика 5.16 Одабир модела

Након одабира поља, кориснику се издаје табела (листа) производа са специфичним вредностима колоне за дати модел, као и предвиђена количина за наручивање. Предвиђене количине корисник може да модификује у табели уколико сматра да вредности нису адекватне (слика 5.17).

Sifra dobavlјaca

11 - Dobavlјac 1

Model

MODEL MINIMALNIH ZALIHA

Izaberi opcije

#	Sifra artikla	Naziv	Kolicina	Z_min	Kolicina za narucivanje
1	1	Proizvod 1	772.00	3306.19	3306.19
2	2	Proizvod 2	700.00	2991.40	2991.40
3	3	Proizvod 3	1954.00	2832.67	2832.67
4	6	Proizvod 6	1680.00	2361.50	2361.50
5	8	Proizvod 8	867.00	3709.35	3709.35
6	9	Proizvod 9	1082.00	2756.46	2756.46
7	11	Proizvod 11	621.00	1867.41	1867.41

Слика 5.17 Модел минималних залиха

Одабиром модела минималних залиха, у табели се представља колона Z_{min} , која специфицира минималну количину производа која треба да се у сваком тренутку налази у складишту, на основу чега модел предвиђа количину за наручивање (слика 5.17). Слично моделу минималних залиха, модел максималних залиха уводи колону Z_{max} која представља критичну количину за набавку (слика 5.18).

Sifra dobavlјaca

11 - Dobavlјac 1

Model

MODEL MAKSIMALNIH ZALIHA

Izaberi opcije

#	Sifra artikla	Naziv	Kolicina	Z_max	Kolicina za narucivanje
1	1	Proizvod 1	772.00	40083.67	40083.67
2	2	Proizvod 2	700.00	18593.91	18593.91
3	3	Proizvod 3	1954.00	67564.32	67564.32
4	4	Proizvod 4	3362.00	53760.12	53760.12
5	5	Proizvod 5	2592.00	34427.59	34427.59
6	6	Proizvod 6	1680.00	22463.00	22463.00
7	7	Proizvod 7	2725.00	67923.68	67923.68

Слика 5.18 Модел максималних залиха

Модел оптималних залиха уводи вредност Q_{opt} која представља оптималну количину за наручивање производа приликом једне набавке, док вредност параметра N_{opt} представља колико ће таквих набавки морати да се изврши за дати производ у року од годину дана (слика 5.19).

51

Sifra dobavlјaca

11 - Dobavlјac 1

Model

MODEL OPTIMALNIH ZALIHA

Izaberi opcije

#	Sifra artikla	Naziv	Kolicina	Q_opt	N_opt	Kolicina za narucivanje
1	1	Proizvod 1	772.00	1165.33	22	1165.33
2	2	Proizvod 2	700.00	2531.78	11	2531.78
3	6	Proizvod 6	1680.00	1726.10	15	1726.10
4	9	Proizvod 9	1082.00	2115.22	12	2115.22
5	11	Proizvod 11	621.00	2285.38	12	2285.38
6	12	Proizvod 12	755.00	797.68	32	797.68

Слика 5.19 - Модел оптималних залиха

Модел ABC анализе уводи рангирање производа по томе колики је процентуални удео датог производа у укупној продаји, на основу чега се производи сврставају у три групе, А В и С (слика 5.20, слика 5.21).

Sifra dobavlјaca

11 - Dobavlјac 1

Model

MODEL ABC ANALIZE

Izaberi opcije

#	Sifra artikla	Naziv	Kolicina	Procenat	ABC_rang	Kolicina za narucivanje
1	1	Proizvod 1	772.00	13.53 %	A	1544
2	2	Proizvod 2	700.00	26.04 %	A	1400
3	3	Proizvod 3	1954.00	37.18 %	A	3908
4	4	Proizvod 4	3362.00	47.71 %	A	6724

Слика 5.20 Модел ABC анализе

14	14	Proizvod 14	2998.00	90.40 %	B	5996
15	15	Proizvod 15	562.00	92.75 %	B	1124
16	16	Proizvod 16	3252.00	94.78 %	B	6504
17	17	Proizvod 17	839.00	96.50 %	C	1678
18	18	Proizvod 18	3951.00	97.98 %	C	7902
19	19	Proizvod 19	3011.00	99.21 %	C	6022

Слика 5.21 Модел ABC анализе

По одабиру модела и потврди о изабраним количинама (слика 5.22),

8	12	Proizvod 12	755.00	2874.91	2874.91
9	13	Proizvod 13	2217.00	3183.47	3183.47
10	14	Proizvod 14	2998.00	3726.92	3726.92
11	15	Proizvod 15	562.00	2170.17	2170.17
12	17	Proizvod 17	839.00	1570.59	1570.59
13	4	Proizvod 4	1200.00	2347.15	2347.15
14	7	Proizvod 7	300.00	2105.20	2105.20
15	10	Proizvod 10	500.00	1727.53	1727.53

Potvrdi količine

Слика 5.22 Потврда количина

кориснику се представља приказ наруџбенице (слика 5.23),

Dobavljaц: Dobavljaц 1
Mesto i adresa: Adresa 1 Postanski broj: 11000
PIB dobavljaца: 000000001
Tekuci racun: 111-1111111111111-11
Email adresa: dobavljaц1@gmail.com
Broj telefona: +381611111111

Kupac: Preduzeće 4
Mesto i adresa: Save Kovacevica 35 Postanski broj: 34000
PIB dobavljaца: 811552349
Tekuci racun: 363-9863365281953-69
Email adresa: preduzeće4@hotmail.com
Broj telefona: +381645246726

NARUDZBENICA Br: 1
Datum izdavanja: 2019-09-24

#	Sifra artikla	Naziv	Jedinica mere	Kolicna	Cena
1	1	Proizvod 1	Jm1	3306.19	46.00
2	2	Proizvod 2	Jm1	2991.40	23.00
3	3	Proizvod 3	Jm1	2832.67	47.00

Nacin i uslovi nabavke:

Nacin i uslovi placanja: _____

Mesto isporuke robe: _____

Nacin isporuke: _____

Vreme isporuke: _____

Mesto dostave racuna: _____

Ostali uslovi: _____

Dobavljaц: _____
Narucilac: _____

Слика 5.23 Приказ наруџбенице

а након тога и могућност уноса (регистравања у бази података) исте (слика 5.24).

Nacin i uslovi nabavke:

Nacin i uslovi placanja: _____
Mesto isporuke robe: _____
Nacin isporuke: _____
Vreme isporuke: _____
Mesto dostave racuna: _____
Ostali uslovi: _____

Dobavljaц: _____
(ovlasćeni radnik)

Narucilac: _____
(ovlasćeni radnik)

Unesi narudzbenu

Слика 5.24 Унос наруџбенице

6. ЗАКЉУЧАК

Веб апликација за управљање магацинским пословањем испуњава све захтеве наведене у спецификацији проблема. Израђена је у програмском језику *PHP* кроз *framework Laravel* уз *MySQL* базу података и тиме представља комплетну веб апликацију са свим неопходним елементима за њено функционисање. Апликација омогућава корисницима брз и лак приступ, преглед и управљање свим елементима у ланцу набавке, омогућава креирање потребних докумената за управљање складишним пословањем. Даљи развој апликације би могао да води у смеру креирања корисничког интервејса за регистрацију и приступ апликацији, коришћење баркода за читавање производа приликом његовог транспорта у и из магацина као и могућност обраде додатних модела који би кориснику омогућили још разноврснији сет (распон) предлога за количине производа који треба да се нађу на креираној наруџбеници у оквиру апликације. Треба нагласити да се ради о веб апликацији, што значи да приступ истој није зависан од коришћења одређених рачунара, већ је једини услов постојање интернет мреже преко које би било који уређај могао да се повеже са апликацијом (пре свега *pc* рачунари, а уз веома малу модификацију изворног кода апликације и сви мобилни телефони и други паметни уређаји).

ЛИТЕРАТУРА

1. Антић С. *Модели и методе управљања залихама засновани на метахеуристикама*, Докторска дисертација, Факултет организационих наука, Београд, 2014.
2. Вукићевић С. *Складишта*, Универзитет у Београду, Превинг, Београд, 1995.
3. Diffusion of D-glucose measured in the cytosol of a single astrocyte - Marko Kreft (pristupljeno 24.08.2019)
4. In Vivo Evidence for a Lactate Gradient from Astrocytes to Neurons - Philipp Mächler, Matthias T. Wyss, Maha Elsayed, ..., Pierre J. Magistretti, L. Felipe Barros, Bruno Weber (pristupljeno 24.08.2019)
5. Крчевинац С., Чангаловић М., Ковачевић-Вујчић В., Мартић М., Вујошевић М., *Операциона истраживања*, Факултет организационих наука, Београд, 2004.
6. L. Shea, Development of an Automated Storage and Retrieval System in a Dynamic Knowledge Environment, MSc, Waterford institute of Technology, Waterford, 2007.
7. Naiburg E., Maksimchuk R., UML for Database Design, Addison-Wesley, Boston, 2002.
8. С. Аћимовић, В. Божић, Маркетинг логистика, Економски факултет Београд, 2012.
9. Williams H., Lane D.: Web Database Applications with PHP & MySQL, O'Reilly, 2004.
10. <https://www.mathworks.com/help/matlab/> (приступљено 14.08.2019)
11. <https://www.sciencedirect.com/topics/computer-science/sigmoid-function> (приступљено 14.08.2019)