|  |  |  |
| --- | --- | --- |
|  | УНИВЕРЗИТЕТ У НОВОМ САДУ  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ** |  |

Јован Гајин

**Веб сервис за управљање складиштима и логистичким процесима у оквиру њих**

ДИПЛОМСКИ РАД

- Основне академске студије -

Нови Сад, 2021

|  |  |
| --- | --- |
| Logo  Description automatically generated | УНИВЕРЗИТЕТ У НОВОМ САДУ ⚫ **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**  21000 НОВИ САД, Трг Доситеја Обрадовића 6 |
| **КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА** |

|  |  |  |  |
| --- | --- | --- | --- |
| Редни број, **РБР**: | |  | |
| Идентификациони број, **ИБР**: | |  | |
| Тип документације, **ТД**: | | Монографска документација | |
| Тип записа, **ТЗ**: | | Текстуални штампани материјал | |
| Врста рада, **ВР**: | | Дипломски (Bachelor) рад | |
| Аутор, **АУ**: | | Јован Гајин | |
| Ментор, **МН**: | | др Милан Челиковић, доцент | |
| Наслов рада, **НР**: | | Веб сервис за управљање складиштима и логистичким процесима у оквиру њих | |
| Језик публикације, **ЈП**: | | Српски / ћирилица | |
| Језик извода, **ЈИ**: | | Српски | |
| Земља публиковања, **ЗП**: | | Република Србија | |
| Уже географско подручје, **УГП**: | | Војводина | |
| Година, **ГО**: | | 2021 | |
| Издавач, **ИЗ**: | | Ауторски репринт | |
| Место и адреса, **МА**: | | Нови Сад; трг Доситеја Обрадовића 6 | |
| Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога) | | 6/76/0/103/0/0/0 | |
| Научна област, **НО**: | | Електротехника и рачунарство | |
| Научна дисциплина, **НД**: | | Примењене рачунарске науке и информатика | |
| Предметна одредница/Кqучне речи, **ПО**: | | Базе података и информациони системи | |
| **УДК** | |  | |
| Чува се, **ЧУ**: | | У библиотеци Факултета техничких наука, Нови Сад | |
| Важна напомена, **ВН**: | |  | |
| Извод, **ИЗ**: | | У овом раду описан је начин израде софтверског решења за управљање складиштеним простором и процесима који се у оквиру њега одвијају. Осмишљен је како би олакшао, убрзао и унапредио пословне процесе у оквиру складишта, али и омогућио праћење рада и отклањање евентуалних грешака у раду. Прво је конструисана база података, а затим на основу генерисаног модела и сам софтверски систем уз помоћ . Net окружења и C# и Angular програмских језика. | |
| Датум прихватања теме, **ДП**: | |  | |
| Датум одбране, **ДО**: | |  | |
| Чланови комисије, **КО**: | Председник: |  |
|  | Члан: |  | Потпис ментора |
|  | Члан, ментор: |  |  |

|  |  |
| --- | --- |
| Logo  Description automatically generated | UNIVERSITY OF NOVI SAD ⚫ **FACULTY OF TECHNICAL SCIENCES**  21000 NOVI SAD, Trg Dositeja Obradovića 6 |
| **KEY WORDS DOCUMENTATION** |

|  |  |  |  |
| --- | --- | --- | --- |
| Accession number, **ANO**: | |  | |
| Identification number, **INO**: | |  | |
| Document type, **DT**: | | Monographic publication | |
| Type of record, **TR**: | | Textual printed material | |
| Contents code, **CC**: | | Bachelor Thesis | |
| Author, **AU**: | | Jovan Gajin | |
| Mentor, **MN**: | | Milan Čeliković, Assitant Professor, Ph. D | |
| Title, **TI**: | | A Web Software Tool for supporting storekeeping buissines processes | |
| Language of text, **LT**: | | Serbian | |
| Language of abstract, **LA**: | | Serbian | |
| Country of publication, **CP**: | | Republic of Serbia | |
| Locality of publication, **LP**: | | Vojvodina | |
| Publication year, **PY**: | | 2021 | |
| Publisher, **PB**: | | Author’s reprint | |
| Publication place, **PP**: | | Novi Sad, Dositeja Obradovica sq. 6 | |
| Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes) | | 6/76/0/103/0/0/0 | |
| Scientific field, **SF**: | | Electrical and computer engineering | |
| Scientific discipline, **SD**: | | Applied computer science and informatics | |
| Subject/Key words, **S**/**KW**: | | Information systems and databases | |
| **UC** | |  | |
| Holding data, **HD**: | | The Library of Faculty of Technical Sciences, Novi Sad, Serbia | |
| Note, **N**: | |  | |
| Abstract, **AB**: | | This thesis presents the specification and implementation of a software solution for running storekeeping buisiness processes. It's designed to make process faster, easier and of better quality. Also, it could help with tracking work and prevent and resolve issues. First, database is conustrucetd and then also software solution based on model generated from database organisation. For that .Net evniroment, C# and Angular programing lenuages are used. | |
| Accepted by the Scientific Board on, **ASB**: | |  | |
| Defended on, **DE**: | |  | |
| Defended Board, **DB**: | President: |  |
|  | Member: |  | Mentor’s sign |
|  | Member, Mentor: |  |  |

Садржај

[1. Увод 11](#_Toc81952997)

[2. Опис реалног система 13](#_Toc81952998)

[2.1 Улоге корисника у информационом систему 13](#_Toc81952999)

[2.2 Чиниоци система 14](#_Toc81953000)

[2.3 Функционални захтеви 14](#_Toc81953001)

[3. Опис апликативног решења 16](#_Toc81953002)

[3.1 Почетна страница 16](#_Toc81953003)

[3.1.1 Корисничка подршка 17](#_Toc81953004)

[3.1.2 Заглавље апликације 17](#_Toc81953005)

[3.1.3 Мој профил страница 17](#_Toc81953006)

[3.2 Страница за пријаву у систем 19](#_Toc81953007)

[3.2.1 Форма за логовање у систем 19](#_Toc81953008)

[3.2.2 Заборављена лозинка 20](#_Toc81953009)

[3.3 Корисничка страница – магационер (storekeeper) 21](#_Toc81953010)

[3.3.1 Претрага робе и креирање рачуна - таб Полице (Shelves) 22](#_Toc81953011)

[3.3.2 Производи - таб Артикли (Items) 24](#_Toc81953012)

[3.3.3 Извештаји - таб извештаји (Reports) 24](#_Toc81953013)

[3.3.4 Поруџбине - таб поруџбине (Loads) 25](#_Toc81953014)

[3.4 Корисничка страница – возач (driver) 27](#_Toc81953015)

[3.4.1 Задужење и раздужење возила 27](#_Toc81953016)

[3.4.2 Преглед и преузимање проуџбина 28](#_Toc81953017)

[3.5 Корисничка страница – менаџер (admin) 30](#_Toc81953018)

[3.5.1 Управљање запосленима – Таб запослени (Employees) 30](#_Toc81953019)

[3.5.2 Управљање артиклима – таб Артикли (Items) 32](#_Toc81953020)

[3.5.3 Управљање складиштима и полицама 33](#_Toc81953021)

[3.5.4 Управљање подацима о купцима – таб Компаније (Companies) 35](#_Toc81953022)

[3.5.5 - Управљање подацима о рампама – таб рампе (Ramps) 36](#_Toc81953023)

[3.5.6 – Управљање подацима о возилима – таб возила (Vehicles) 36](#_Toc81953024)

[3.5.7 – Поруџбине и Извештаји 37](#_Toc81953025)

[4. Опис коришћених техничких решења 38](#_Toc81953026)

[4.1 Backend решење 38](#_Toc81953027)

[4.1.1 „WebApplication“ 39](#_Toc81953028)

[4.1.2 „Warehouse.Web.Model“ 39](#_Toc81953029)

[4.1.3 „Warehouse.Model“ 41](#_Toc81953030)

[4.1.4 „Warehouse.Service“ 42](#_Toc81953031)

[4.1.5 „Warehouse.Repository“ 44](#_Toc81953032)

[4.2 Frontend решење 45](#_Toc81953033)

[4.3 Комуникација са базом података 48](#_Toc81953034)

[4.4 Сигурност и безбедност 49](#_Toc81953035)

[4.5 Управљање верзијама (git) 53](#_Toc81953036)

[4.6 Ценовник 54](#_Toc81953037)

[5 Опис базе података 55](#_Toc81953038)

[5.2 Ентитети релационе базе података 56](#_Toc81953039)

[5.2.1 Ентитет „Warehouse“ – складиште 56](#_Toc81953040)

[5.2.2 Ентитет „Ramp“ - рампа 58](#_Toc81953041)

[5.2.3 Ентитет „Shelf“ - Полица 59](#_Toc81953042)

[5.2.4 Ентитет „Employee“ - Запослени 60](#_Toc81953043)

[5.2.5 Ентитет „Employee Type“ – Тип корисника 61](#_Toc81953044)

[5.2.6 Ентитет „Vehicle“ – Возило 62](#_Toc81953045)

[5.2.7 Ентитет „Load“ – Продуџбина 63](#_Toc81953046)

[5.2.8 Ентитетн „Report“ – Извештај 65](#_Toc81953047)

[5.2.9 Ентитет „Report Type“ – Тип извештаја 66](#_Toc81953048)

[5.2.10 Ентитет „Item“ - Артикал 67](#_Toc81953049)

[5.2.11 Ентитет „Price List“ – Ценовник 68](#_Toc81953050)

[5.2.12 Ентитет „Item Type“ – Тип артикла 69](#_Toc81953051)

[5.2.13 Ентитет „Item Report“ – Ентитет релације Артикал – Извештај 70](#_Toc81953052)

[5.2.14 Ентитет „Receipt“ – Рачун 71](#_Toc81953053)

[5.2.15 Ентитет „Item Receipt“ - Ентитет релације Артикал – Рачун 73](#_Toc81953054)

[5.2.16 Ентитет „Company“ – Компанија 74](#_Toc81953055)

[6 Закључак 75](#_Toc81953056)

[Речник појмова 76](#_Toc81953057)

[Скраћенице 77](#_Toc81953058)

[Литература 78](#_Toc81953059)

[Биографија 79](#_Toc81953060)

# Увод

Софтверски системи су постали саставни део готово сваког сегмента пословања. Они не само да поједностављују, већ пословне процесе чине и ефикаснијим, безбеднијим и отпорнијим на људске грешке.

Услед велике експанзије различитих врста рачунара, али и оперативних система које они користе, као и све учесталије потребе да се сисему приступи ван управљачке собе или другог места намењеног за приступ, све је већа потреба за универзалним софтверским решењима који ће у потпуности одговорити захтевима сложених пословних процеса, истовремено не оптерећујући и не ограничавајући кориснике употребом одређене врсте рачунара и/или оперативних система.

Потреба за брзим и једноставним приступом подацима у покрету, односно из било ког дела пословног објекта, возила или неке удаљене локације, отворила је низ нових питања на које су софтверски инжењери и архитекте морали дати брз одговор.

Мобилност, унвиверзалност, односно независност од специфичне врсте рачунара и оперативних система само су део проблема. Као веома важна, показала су се и питања безбедности, поузданости, отпорности на грешке, и многа друга.

Као изузетно ефикасно, а ипак довољно једноставно, доступно и „јефтино“ решење овог проблема наметнула се израда веб апликација уско специјализованих за управљање одређеним процесима, намењена и доступна веома малом броју корисника.

Овај рад представља пре свега технички опис једног оваквог решења намењеног управљању, чувању и обради података у процесу логистике, односно управљања магацинским простором. Такође, он може служити и као сасвим довољна техничка докуметација помоћу које се будући корисници система могу информисати али и у довољној мери обучити за коришћење овог система.

Поред тога, у раду ће бити изложене особине које аутор рада и софтверског решења види као главне предности и мане, али и техничке појединости и технологије без којих израда не би била могућа.

У наставку, покушаћемо да што сажетије опишено техничка и архитектонска решења која су корићена приликом израде пројекта, истовремено образлагајући разлоге да њихов одабир. Описаћемо пословни процес и начин на који се систем може користити у реалним условима.

Осим тога, одговорићемо на нека веома важна питања која са собом доноси израда једног оваквог система. То су, између осталих: безбедност, поузданост, заштита података и тако даље.

Рад, поред овог увода, обухвата и следеће целине, даље подељене на мања поглавља:

**Опис реалог система** - у оквиру кога ће бити приказан пословни процес као и физички чиниоци система, направљена јасна разлика између за систем битних и небитних података и дат опис функционалних захтева и изазова у процесу стварања информационог система

**Опис апликативног решења** – опис функционалности апликације помоћу којих је одговорено на функционалне захтеве реалног система са примерима

**Опис коришћених технолошких решења** – у оквиру које ћемо се осврнути на технологије помоћу којих је пројекат израђен као и на неке специфичне алгоритаме и делове кода за које је аутор сматрао да их треба приказати или образложити

**Опис базе података** – свођење скупа података на најмањи довољан скуп, дизајн и архитектура базе, опис технологија коришћених за израду и анализа шеме базе података

**Закључак** – са неким, за аутора најбитнијим предностима и потенцијалним манама система, као и простором за евентуално унапређење

# Опис реалног система

Реалан систем за управљање магацинским простором и процесима који се у њему свакодевно изводе, много је шири од оног описног и пројектованог овим системом. Ми смо се овде држали чиниоца и процеса битних за логистичке операције, пре свега пријема и издавања робе, док би други делови, попут финансија, књиговодства и тако даље, морали бити иведени као посебан део овог, или независтан систем. У складу са тим, скуп подата и операција сведен је на минимум такав да исупњава све захтеве уско специјализованог дела система, коме је овај софтвер и намењен. Примера ради, радник је у систему описан само оним подацима неопходним за функционисање система, док би подаци као што су висина плате, адреса, датум запослења, и многи други, били део горе поменутог посебног (дела) ситема у оквиру којег би они били релевантни.

У даљем тексту, укратко ћемо представити и описати класе корисника, чиниоце система, као и операције помоћу којих је дозвољена манипулација над њима. Поред тога, даћемо и кратко образложење зашто су баш ти подаци били од значаја за рад система као и зашто су остали могли бити изузети.

## Улоге корисника у информационом систему

У информационом систему за управљање складишима постоје следеће класе корисника, односно улоге у систему:

* **Магационер** – особа задужена за управљање магацинским простором, врши следеће дужности:
  + Пријем робе са пријемницом
  + Пријем захтева за издавање (продају) робе
  + Паковање
  + Издавање робе са отпремницом и рачуном
    - Претрага кроз складиште
    - Додавање робе са количином
    - Одабир компаније купца из листе сарадника
  + Додатне могућности у коришћењу система
    - Преглед извештаја
    - Претрага робе кроз сва складишта
* **Возач** – особа задужена за управљање возилом и транспорт робе од складишта до жељене локације клијента. На располагању има следеће функционалности система:
  + Избор и задуживање слободног возила
  + Повраћај односно ослобађање возила
  + Преглед робе спремне за утовар
    - Детаљи о локацији утовара (ознака и статус рампе на којој ће утовар бити извршен)
  + Преглед тренутног утовара са обавештењем о крају процеса
  + Штампање рачуна за крајњег корисника (клијента)
* **Управник** (администратор система) – особа задужена за надгледање читавог процеса, исправљање грешака насталих у раду, прегледање извештаја и измену, допуну и брисање података
  + Има приступ подацима које виде и сви остали корисници, као и могућност измене, додавања и брисања података:
    - Корисника, робе, магацина, компанија, извештаја, рачуна и возила. О овоме ће детаљније бити речи у опису апликативног решења.

## Чиниоци система

У овом поглављу даћемо кратак преглед чиниоца реалног ситема који су препознати као фундаментални за развој софтверског решења и као такви унети у инфомрациони ситем. Детаљнији опис и образложење за одабир ових чиниоца даћемо у наставку рада.

* Стовариште – магацински односно складиштени простор који има свој назив, адресу, магационера – управника, полице са робом и рампе
* Рампа – место за утовар/истовар робе са својим редним бројем
* Радник - магационер, возач или управник
* Полица – мања целина за смештање робе
* Роба – фундаменталан део система, објекат складиштења и продаје
* Рачун – физички доказ о продаји робе
* Пријемница и отпремница - физички доказ о уласку односно изласку робе из складишта
* Извештај – траг у систему са описом промене стања над подацима
* Ценовник – тренутно стање и историјат цена за сву робу

Детаљније о атрибутима, везама и улози појединачних чиниоца система позабавићемо се у поглављу описа базе података.

## Функционални захтеви

Функционални захтеви система представљају операције над чиниоцима система и подацима који су неопходни за функционисање система. На основу прегледа ових захтева, настале су операције у оквиру информационог система које су доступне корисницима, а помоћу којих корисници манипулишу подацима о чиниоцима система, на тај начин одржавајући пословни процес.

У систему урављања складиштима, препознајемо следеће процесе:

* **Пријем робе** – основна намена складиштеног простора јесте чување веће количине робе ради даље дистрибуције. Роба свакодневно долази у складиштени простор и основни задатак јесте заприма робе. Приликом пријема, дефинишу се најбитнији подаци а то су шифра, назив, врста робе, полица на коју ће роба бити смештена и количина. Овај процес обавља неки од магационера запослених у складишту у које је роба стигла. За сваку примљену робу генерише се **пријемница** (писани траг о заприми робе) .
* **Издавање робе** – пут робе из складишта ка даљим потрошачима започиње тако што купац, односно фирма сарадник потражује одређену количину робе. Тада магационер има задатак да тражену робу пронађе и припреми за транспорт, односно упакује.
  + **Паковање је под**-**процес** издавања робе

Приликом издавања робе генерише се **оптремница** (писани траг о издавању робе).

* **Утовар и превоз –** спакована роба се даље утовра на возило пристало на рампу за утовар, и даље превози на адресу фирме купца. Овај процес заједно обављају магационер и возач транспортног односно теретног возила присталог на утовар. Након утовра робе, возач добија **рачун** (легални извештај за купца о роби коју је купио).

Осим ових, за функционисање кључних операција, постоје и многе функционалности које нису део најважнијих, али су узете у обзир и пројектоване у информационом систему, јер у знатној мери олакшвају, убрзавају и повећавају квалитет самог процеса. Ту спадају следеће функционалности:

* **Претрага робе** – складиштени простори могу бити изузетно велики, па је често немогуће без претходне провере знати да ли нека роба постоји на стању, у којој количини као ни то где се она налази. Због тога је омогућена претрага система у оквиру изабраног или у свим складиштима и то на основу:
  + Назива, врсте робе и складишта у коме се налази
* **Генерисање извештаја** – како би рад био транспарентан, али и како ми могао да се прати, анализира или отклоне евентуалне грешке, за сваку промену стања робе у систему (пријем или оптремање), генерише се извештај у pdf формату спреман за штампање.
  + Извештај садржи информације о типу, датум и времену промене као и саму промену.
  + Приступ свим извештајима доступан је магационерима и управницима
  + Приказ може бити сортиран по датуму или типу
  + Типови извештаја су: **пријемница** и **оптемница**
* **Издавање рачуна –** као део законске регулативе, за сву издату робу мора да постоји рачун. Систем на крају утовара робе сам генерише и прирпема рачун за штампање или слање електронском поштом.
* **Измена података у систему –** операција доступна само управницима односно администраторима система. Подразумева додавање, измену или брисање корисника, робе, возила, купаца, рампи, складишта, полица или неких од њихових атрибута.
  + Администратор система може помоћу једноставног интерфејса да исправи грешке или забележи промену на било ком чиниоцу система. Он може, не само да додаје или брише чланове (уз одређена ограничења), већ може да изврши поромену на неком од атрибута сваког чиниоца. Тако на пример, може манипулисати количином, називом или локацијом (полицом) робе, личним подацима, адресом електронсе поште или улогом корисника. Може мењати и дефинисати нове управнике складишта, и слично.
* **Управљање корисничким налогом –** корисник система има преглед својих података у одељку „мој профил“. Овде он може видети своје податке, али и изменити корисничку шифру уколико има основа за сумњу да је она компромитовања. У случају заборављене лозинке, корисник исту може затражити на страници за пријаву у систем, и затим сачекати да му на адресу електронске поште стигне упуство за ресетовање шифре или нова, једнократна шифра.
* **Аутентификација и ауторизација помоћу логовања у систем –** како би сигурност система била на што вишем нивоу али и како би одређене операције биле досупне само корисницима који имају права да их користе, корисник пре коришћења мора да се пријави (аутентификује) помоћу адресе електронске поште и лозинке. Систем на основу ових података у бази проналази улогу корисника у систему и приказује му страницу на основу ње на којој су му доступне само опције намењене групи корисника којој он припада (ауторизација). Детаљније о овом процесу објаснићемо у опису апликативног решења.

# Опис апликативног решења

У овом одељку, бавићемо се главним функционалностима и могућностима које систем доноси различитим типовима корисника. Осим тога, описаћемо стандардан „work - flow“ односно кораке за извршавање различитих задатака у оквру пословних процеса које овај систем има задатак да испрати.

Апликативно решење овог система функционише као клијент – сервер веб апликација. Више о техлолошким карактеристикама и предностима овог система изложићемо у следећем поглављу. Разлог за одабир веб апликације огледа се пре свега у потреби корисника да систему и његовим подацима приступе било када, са било ког места и са различите врсте рачунара и оперативних система на њима.

Иако јавно доступа, апликација није намењена широком броју људи, већ веома малом броју корисника. Из тог разлога, пре почетка коришћења апликације, сваки корисник мора помоћу унапред креираног налога да потврди свој индентитет.

## 3.1 Почетна страница

Приликом покретања апликације, односно учитавања веб адресе апликације кориснику се приказује почетна страна приказана на слици 1.

Graphical user interface, application

Description automatically generated

Слика 1 – почетна страна апликације

Како је ово јавно доступна страница, на њој се не налазе нити се могу добити никакви за компанију битни или осетљиви подаци. На њој је могуће приказати лого и назив фирме, кратку статистику о укупном и дневном броју поруџбина (замишљену као део маркетиншке капање која описује ефикасност система) као и линкови ка страници за логовање у систем или евентуалну помоћ корисничке подршке.

### Корисничка подршка

Кликом на линк за помоћ корисничке подршке приказаном на слици 2, отвара се „Outlook“, „Email“ или нека друга апликација за размену електронске поште дефинисана као примарна на уређају са кога је апликација покренута.

Овај линк досупан је корисницима у сваком трненутку у „footer“ -у апликације, без обзира на страну или тренутни статус у ком се корисник налази.

Кликом на линк, у одабраној апликацији отвара се прозор за слање нове електронке поруке са већ попуњеном адресом примаоца. На овај начин, кориснику је максимално олакшана комуникација са корисничком подршком и веома једноставно је учињено да она буде на само „један клик“ од корисника.

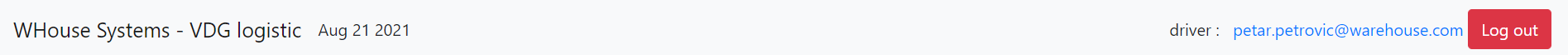
Text, letter

Description automatically generated

Слика 2 – приказ линка за слање поруке корисничкој подршци

### Заглавље апликације

У заглављу, односно „header“ – у апликације приказаном на слици 3 налазе се назив компаније, датум и у случају када је корнсик пријављен, у десном углу налазе се подаци о типу (улози, односно „role“) корисника, адреса електронске поште, чијим се кликом отвара профил страница корсиника, и дугме за одјаву односно „log out“ из система.



Слика 3 – приказ заглавља апликације

### Мој профил страница

Мој профил страница пре свега служи за преглед личних информација као што су име, презиме, број телефона, адреса електронске поште и тип корисника. Пример изледа ове странице приказан је на слици 4.

Graphical user interface, text, application

Description automatically generated

Слика 4 – приказ „Мој профил“ стране

Осим тога, веома битну улогу у систему има и опција за промену шифре, доступна кликом на дугме „change password“. Кликом на дугме, отвара се форма за измену шифре приказана на слици 5. Након уноса исправне старе лозинке и нове која испуњава услове валидације, корисник је успешно изменио своју лозинку. **Након измене лозинке, корисник ће аутоматски бити одјављен из система и потребно је пријавити се поново, овај пут користећи новодефинисану лозинку.**

**Graphical user interface, application

Description automatically generated**

Слика 5 – форма за промену лозинке

На сликама 6 и 7 приказане су поруке у случају успешне, односно неуспешне пормене лозинке.

Text

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

Слика 6 – обавештење о погрешно унтетој лозинки Слика 7 – обавештење о успешној промени лозинке

## 3.2 Страница за пријаву у систем

Кликом на “login” линк, приказан на слици 1, корисник прелази на страну намењену пријави у систем.

Уносом адресе електронске поште и лозинке, систем у бази података проналази улогу корисника и у складу са тим га преусмерава на страницу намењену улози којој корисник припада.

### Форма за логовање у систем

Форма за логовање приказана је на слици 8. У случају уноса погрешне комбинације података, приказује се обавештење приказано на слици 9.

Graphical user interface, application

Description automatically generated

Слика 8 – форма за пријаву у систем

Graphical user interface, text, application

Description automatically generated

Слика 9 – форма за пријаву у систем са обавештењем о погрешно унетим подацима

Након успешне пријаве, врши се преусмеравање корисника на једну од три стране намењене приступу подацима корисника и то:

* Магационерска страна (/storekeeper-home)
* Администраторска страна (/manager-home)
* Возачева страна (/driver-home)

### Заборављена лозинка

Кликом на линк „I forgot password“ приказаном на сликама 8 и 9, корисник добија обавештење да му је нова лозинка послата на адресу електронске поште. Пример садржаја електронске поште приказан је на слици 10.

Graphical user interface, text, application, email

Description automatically generated

Слика 10 – пример поруке у случају заборављене лозинке

Шифра послата кориснику у случају заборављене лозинке аутоматки је генерисана као случајан избор караткера, и намењена је пре свега као једнократно решење за поновну пријаву на систем, па се корисницима саветује да исту, одмах након успешне пријаве промене. То обавештење је саставни део поруке која се у овом случају шаље кориснику (приказана на слици 10).

## Корисничка страница – магационер (storekeeper)

Као најчешћи начин коришћења система, замишљена је улога магационера. Као што смо у опису реалног система већ навели, магационер је особа задужена пре свега за пријем и издавање робе. У овом поглављу, представићемо опције које магационер има на располагању као и начин за њихову правилну употребу.

На слици 11 приказан је изглед странице непосредно наког пријављивања у систем.

Graphical user interface, application

Description automatically generatedСлика 11 – преглед расположивих складишта

По пријави у систем, магационер има увид у све магацине (складишта) у систему. Кликом на неки од њих, одабира се складиште над којим ће се вршити операције и отвара се нова страна за манипулацију робом, извештајима и другим подацима у оквиру одабраног складишта.

За промену складишта над којим се обављају операције, магационер не мора да се одјављује, па поново пријављује у систем већ је довољно да из падајућег менија у горњем десном углу (сликa 12) одабере име жељеног складишта.

Graphical user interface, text, application, chat or text message

Description automatically generated

Слика 12 – падајући мени за избор складишта

Операције над подацима груписане су у табове и кликом на сваки од њих отварају се форме и подаци неопходни за извршавање специфичних операција.

Graphical user interface, application

Description automatically generated

Слика 13 – Табови на страни намењеној магационерима

### 3.3.1 Претрага робе и креирање рачуна - таб Полице (Shelves)

Овај таб намењен је прегледу робе по полицама и издавању нових рачуна.

У табели приказаној на слици 14. корисник има могућност прегледа и претраге робе распоређене по полицама. Претрага је могућа по имену и типу робе. Када корисник пронађе одговарајући артикал, двоструким кликом он се додаје у „корпу“ приказану на слици 15.

A picture containing table

Description automatically generated

Слика 14 – роба приказана по полицама

Graphical user interface, application

Description automatically generated

Слика 15 – пример поруџбине

Као посебна ставка у корпи, додаје се само она роба која се већ не налази у корпи и то са количином 1. За робу која се налази у корпи, сваким следећим додавањем, не прави се нова ставка, већ се количина на постојећој повећава за 1.

Количина робе, такође се може мењати кликом на „+“ односно „-“ поред сваке ставке (слика 15). Уколико количина одређеног артикла достигне максималан број тренутно расположив за тај артикал, операција повећавања количине се блокира. Уколико корисник смањи количину за артикал на 0, артикал се уклања из корпе. Артикал се може уклонити из корпе и кликом за дугме „x“, без обзира која је тренутна количина артикла у корпи.

У доњем углу корпе, налази се информација о тренутној укупној цени за све додате артикле. Она се аутоматски ажурира било додавањем/уклањањем артикала из корпе, било променом њихове количине.

Након завршетка одабира робе, корисник из падајућег менија бира име компаније за коју се рачун издаје и притиска дугме „Сачувај (Save)“. Након тога, добија обавештење да је рачун успешно креиран. Осим рачуна, у овом тренутку у систему се креира и излазни извештај као и налог за утовар („load“) како би возач добио информацију о томе да нова испорука треба да се утовари и превезе до купца.

Graphical user interface, application, Word

Description automatically generated

Слика 16 – Обавештење о успешном чувању рачуна

Прављење заједничког рачуна за робу из више магацина није могуће. Уколико је купцу потребно издати робу чији се део налази у неком другом магацину, неопходно је направити два одвојена рачуна за сваки магацин.

### 3.3.2 Производи - таб Артикли (Items)

У овом табу магационер има могућност прегледа и претраге робе не само у складишту у ком тренутно ради, већ кроз сва складишта. Ова операција корисна је пре свега онда када магационер није сигуран у ком од магацина се нека роба налази, или да ли одређених артикала тренутно има на стању. Слика 17 приказује табелу у којој је могуће (по имену и типу) претражити робу кроз све магацине, проверити њену количину као и магацин у којој се налази.

Graphical user interface, application

Description automatically generated

Слика 17 – приказ табеле за претрагу робе кроз сва складишта

### 3.3.3 Извештаји - таб извештаји (Reports)

На табу намењеном извештајима, магационер има приступ свим извештајима о пријему односно издавању робе. Табела се може сортирати по датуму или типу кликом на имена колона („Date“ и „Type“) ради лакшег сналажења. Кликом на дугме „Download“, извештај се у PDF формату чува на рачунару корисника. На сликама 18 и 19 дати су приказ табеле извештаја сортиран по типу и пример излазног извештаја.

A picture containing graphical user interface

Description automatically generated

Слика 18 – табела извештаја сортирана по типу (излазни извештаји први)

Table

Description automatically generated

Слика 19 – пример излазног извештаја у PDF формату

### 3.3.4 Поруџбине - таб поруџбине (Loads)

Магационер има могућност провере статуса сваке креиране поруџбине. Поруџбине је могуће претражити помоћу јединственог идентификатора („Id“), или по типу.

Поруџбина може бити:

* Утоварена („Loaded“) – поруџбина која је већ у возилу или испоручена купцу
* Неутоварена(„Unloaded“) – поруџбина која је спремна и чека на утовар у возило
* На утовару („Loading“) – поруџбина која се управо утовара у возило

На слици 20 приказан је изглед стране за преглед поруџбина.

Graphical user interface, application, table

Description automatically generated with medium confidence

Слика 20 – приказ странице за преглед поруџбина

Ифнормације које корисник има на располагању о проуџбини су идентификатор, складиште у ком је проуџбина креирана, рампа на којој је извршен или се врши утовар, магационер који је проуџбину креирао, возило и возач који робу превозе (уколико су познати) као и рачун. Кликом на дугме „Download“ на рачунару се чува рачун у PDF формату чији пример је приказан на слици 21. Рачун садржи информације о купцу, издатој роби и цени, како појединачно, тако и укупно.

Table

Description automatically generated

Слика 21 – пример рачуна за издату робу

## Корисничка страница – возач (driver)

Улога возача у систему је превоз робе од складишта до купца. За обављање ове функције, возач мора да задужи возило, добије информацију о времену и локацији утовара и превезе робу (и изда рачун) на адресу купца.

У наставку овог параграфа прикзазаћемо операције које возач има на располагању и најбољи начин за њихову употребу.

### 3.4.1 Задужење и раздужење возила

Након пријаве у систем, уколико возач нема задужено возило, односно уколико не постоји возило које у том тренутку припада пријављеном возачу, на почетној страни биће приказана листа слободних возила. Возач двоструким кликом одабира возило и задужује га. Пример листе слободних возила приказан је на слици 22.

A picture containing table

Description automatically generated

Слика 22 – листа слободних возила

Возач не може да започне са процесом утовара робе, нити да види поруџбине на чекању за утовар ако нема задужено возило. Сваки возач може да задужи само једно возило, и свако возило може у једном тренутку да има само једног возача, односно, док један возач не раздужи возило, остали возачи га не виде у листи расположивих возила.

Након успешног одабира возила, у десном углу странице возач у сваком тренутку може да види податке о возилу које је задужио (слика 23).

Осим тога, кликом на дугме „Врати возило (Return vehicle)“, возач враћа задужено возило, а систем се враћа на листу слободних возила.

Graphical user interface, application

Description automatically generated

Слика 23 – информације о задуженом возилу

### 3.4.2 Преглед и преузимање проуџбина

Након успешног одабира возила, у централном делу странице возач види листу поруџбина које чекају на утовар са информацијом о складишту, рапми на којој ће се утовар извршити и томе да ли је рампа слободна.

Утовар робе почиње онда када возач својим возилом приступи слободној рампи. У систему, овај догађај дешава се онда када возач из листе поруџбина двоструким кликом одабере ону чији утовар жели да започне. На слици 24 приказан је пример листе поруџбина које чекају утовар.

Graphical user interface, text, application, email

Description automatically generated

Слика 24 – листа поруџбина које чекају на утовар

Након успешно преузете поруџбине, возач у централном делу екрана види информације о тренутном утовару у возило, приказане на слици 25.

Graphical user interface, text, application

Description automatically generated

Слика 25 – информације о тренутном утовару

У овом тренутку, возач може на свој рачунар да скине излазни извештај настао приликом креирања поруџбине како би могао да провери садржај утоварене робе. Пример овог извештаја налази се на слици 19.

Осим тога, ради лакшег проналаска поруџбине од стране магационера у систему, на овој страни **приказан је и „QR“ код јединственог идентификатора поруџбине**.

Након завршетка утовара, возач притикса дугме „Заврши утовар“ и добија информацију о успешном завршетку утовара. Тек након пријема информације, возач може да напусти рампу и започне транспорт ка адреси купца. На слици 26 приказан је очекивани изглед обавештења о успешном завршетку утовара.

**Након завршетка утовара, на раучунару возача аутоматски се чува фајл са рачуном**, приказан на слици 21.

Graphical user interface, text, application

Description automatically generated

Слика 26 – обавештење о успешном завршетку утовара

## 3.5 Корисничка страница – менаџер (admin)

Менаџер има администраторкса права над системом. У пракси, то подразумева да он не само да има увид у све податке настале у систему, већ само он има могућност да те податке проширује, мења или брише.

Слично као на страници намењеној магационерима, страница намењена менаџерима подељена је на функционалне целине, при чему сваки таб у навигационом бару представља једну функционалну целину.

### 3.5.1 Управљање запосленима – Таб запослени (Employees)

У овом табу, администратор система има преглед листе свих запослених у систему укључујући њихове личне податке и тип, односно позицију на којој раде.

Администратор на располагању има претрагу кроз табелу запослених и то по следећим критеријумима:

* Име
* Презиме
* Адреса електронске поште
* Позиција

**Кликом на дугме „очисти“, табела се враћа у почетно стање, односно приказ свих запослених.**

Пример табеле са листом заполених приказан је на слици 27.

Graphical user interface, text, application, email

Description automatically generated

Слика 27 – листа запослених

Кликом на дугме „Додај новог радника“ у горњем десном углу, отвара се форма за унос података о новом запосленом (слика 28).

**Форма се отвара на десној половини странице, док се листа запослених смањује тако да заузме преостали простор на левој половини. Сличан принцип примењен је и на осталим функционалним целинама.**

Graphical user interface, application

Description automatically generated

Слика 28 – форма за додавање новог радника

Након успешног попуњавања података, позиције и почетне шифре радника (шифре помоћу које ће се радник први пут пријавити на систем, а затим је заменити само њему познатом шифром), кликом на дугме „Сачувај“, листа радника се проширује новим уносом, и профил новог радника је спреман за употребу.

Са десне стране сваког запосленог у листи, налази се црвено „x“ дугме, чијим кликом се корисник брише из ситема. **Магационер који је наведен као управник неког од складишта не може се обрисати**. Пре брисања таквог магационера, управа над складиштем се мора пренети на неког другог расположивог магационера.

Двоструким кликом на неког од запослених отвара се иста форма за унос података као на слици 28, овај пут испуњена подацима о изабраном раднику. Администратор може изменити неки од доступних података и након чувања, они ће бити трајно промењени. На овај начин врши се манипулација подацима у оквиру свих функционалних јединица.

**Graphical user interface, text, application, email

Description automatically generated**

Слика 29 – Форма за измену података о раднику

**Background pattern

Description automatically generated**

Слика 30 – расположиве позиције при додавању / измени радника

### 3.5.2 Управљање артиклима – таб Артикли (Items)

Таб за управљање артиклима веома личи на исти таб на страници намењеној магационерима. Једина разлика огледа се у томе што администратор, осим претраге и увида у артике, има и могућност измене података. Двоструким кликом на жељени артикал, отвара се форма за измену приказана на слици 31. Принцип измене података описан је у поглављу управљање запосленима.

На овом месту, поред основних података (име, количина, цена и тип), администратор може изменити и складиште као и полицу на којој се артикал налази.

Graphical user interface, application

Description automatically generated

Слика 31 – приказ таба за преглед артикала са формом за измену артика на админ страници

**Артикал се не може обрисати из система.** Ова функционалност је са намером изостављена, ради лакшег праћења историје робе у складишту. Извештаји помоћу којих се прати рад система зависе од артикала који се на њој налазе, па би њихово брисање угрозило конзистентност података.

### 3.5.3 Управљање складиштима и полицама

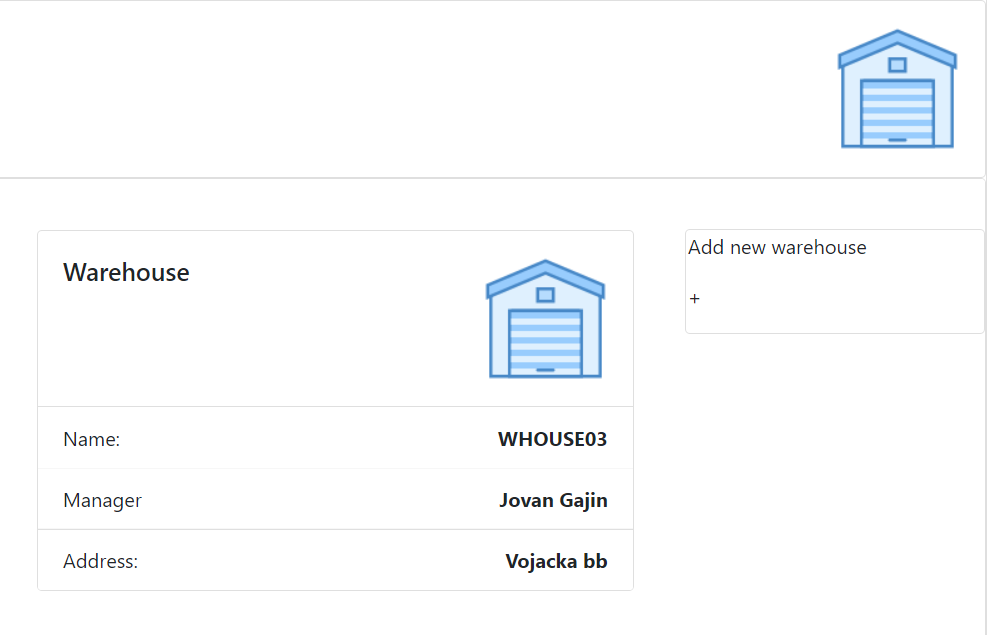
У табу за управљање складиштима и полицама, администратору система на располагању је преглед информација о свим складиштима система. На овом месту видљиве су информације о називу, адреси и управнику сваког складишта у систему.

Кликом на дугме „+“, отвара се форма за додавање новог складишта (слика 32).

Иста форма, попуњена подацима отвара се и двоструким кликом на неко од складишта. На овај начин врши се измена над подацима о складиштима.

**На овом месту, администратор може изменити податке о управнику складишта, како би омогућио брисање корисника.**

Осим података о складишту, на овом месту видљиве су и полице које тренутно постоје у сваком од складишта (слика 35).



Слика 32 – Информације о складишту и дугме за додавање новог складишта

Кликом на дугме „додај полицу (add shelf)“, отвара се форма за додавање нове полице приказана на слици 33, а кликом на дугме „x“, полица се брише из одабраног складишта.

Table

Description automatically generated with medium confidence

Слика 33 – Форма за унос нове полице

Полица се може обрисати само ако се на њој не налази ни један артикал. У супротном, артикли са полице се прво морају распоредити на неку другу полицу у табу за управљање артиклима.

Graphical user interface, text, application

Description automatically generated

Слика 34 – Обавештење о неуспешном брисању полице

**Дугме затвори („Close“) затвара отворену форму у оквиру које се налази. Ова функционалност иста је за све делове система.**

Додата складишта не могу се обрисати из система због конзистентности података који од њих зависе.

Graphical user interface

Description automatically generated

Слика 35 – форма за додавање новог складишта / измену складишта са прегледом полица

### 3.5.4 Управљање подацима о купцима – таб Компаније (Companies)

У оквиру овог таба, амдинистратору су на располагању:

* преглед података,
* претрага компанија,
* бирсање,
* додавање нових и
* измена инфорација о посојећим компанијама.

Подаци дуступни о свакој компанији су назив, ПИБ, адреса, број рачуна и информација о томе да ли је компанија приликом регистровања уплатила договорени износ као депозит.

**Претрага по компанијама је могућа на основу имена или ПИБ-а.**

На слици 36 приказан је пример странице за преглед информација о компанијама. На десној половини странице, налази се табела са листом компанија које одговарају критеријуму унетом у поље за претграгу. Како је ПИБ јединствени идентификатор сваке компаније, само једна компанија у табели је очекиван резултат.

На левој половини странице налази се форма за измену или унос нових података. Она се отвара кликом на дугме „Додај нову компанију (Add new company)“ у горњем десном углу, или двоструким кликом на компанију чије податке желимо да изменимо и у том случају је већ испуњена тренутним подацима о компанији.

Кликом на дугме „x“, изабрана компанија се брише из система.

Graphical user interface, application

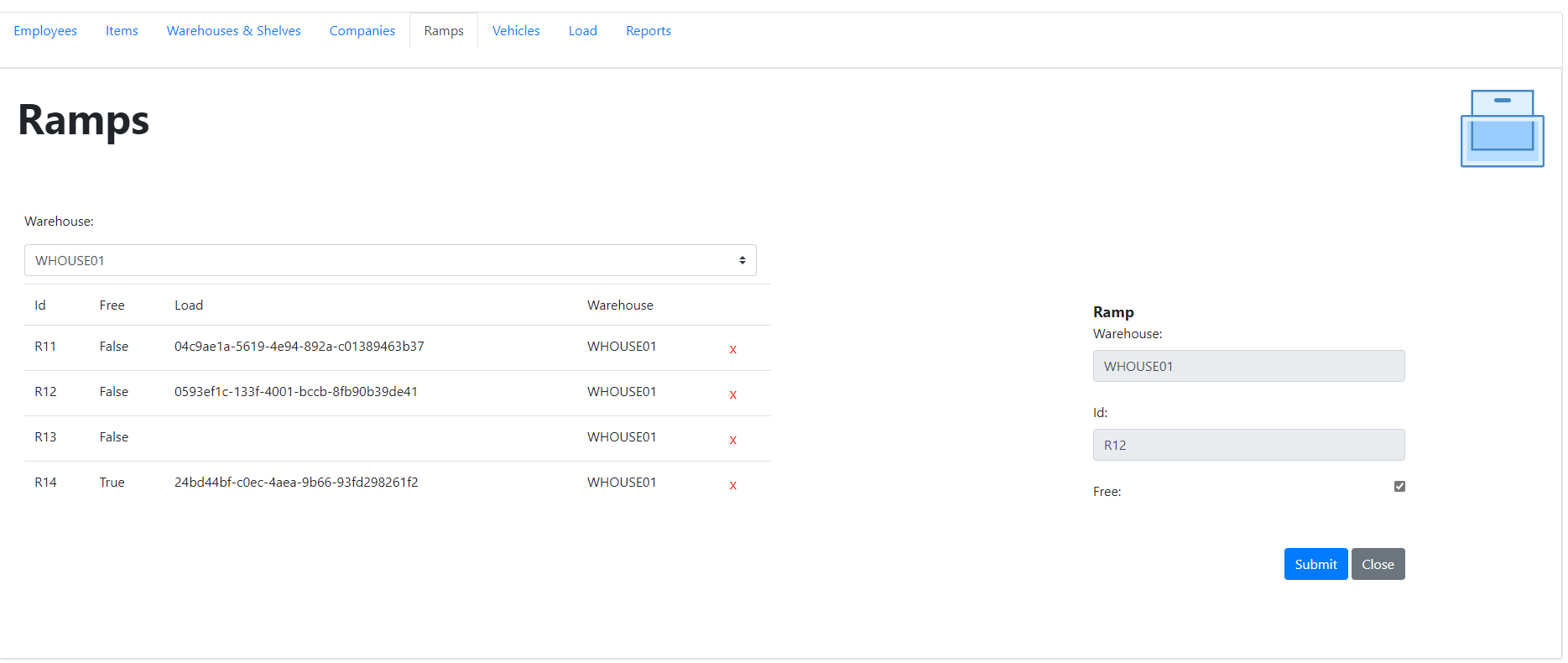
Description automatically generated

Слика 36 – листа компанија са функциом претраге по ПИБ-у и форма за унос/измену података

### 3.5.5 - Управљање подацима о рампама – таб рампе (Ramps)

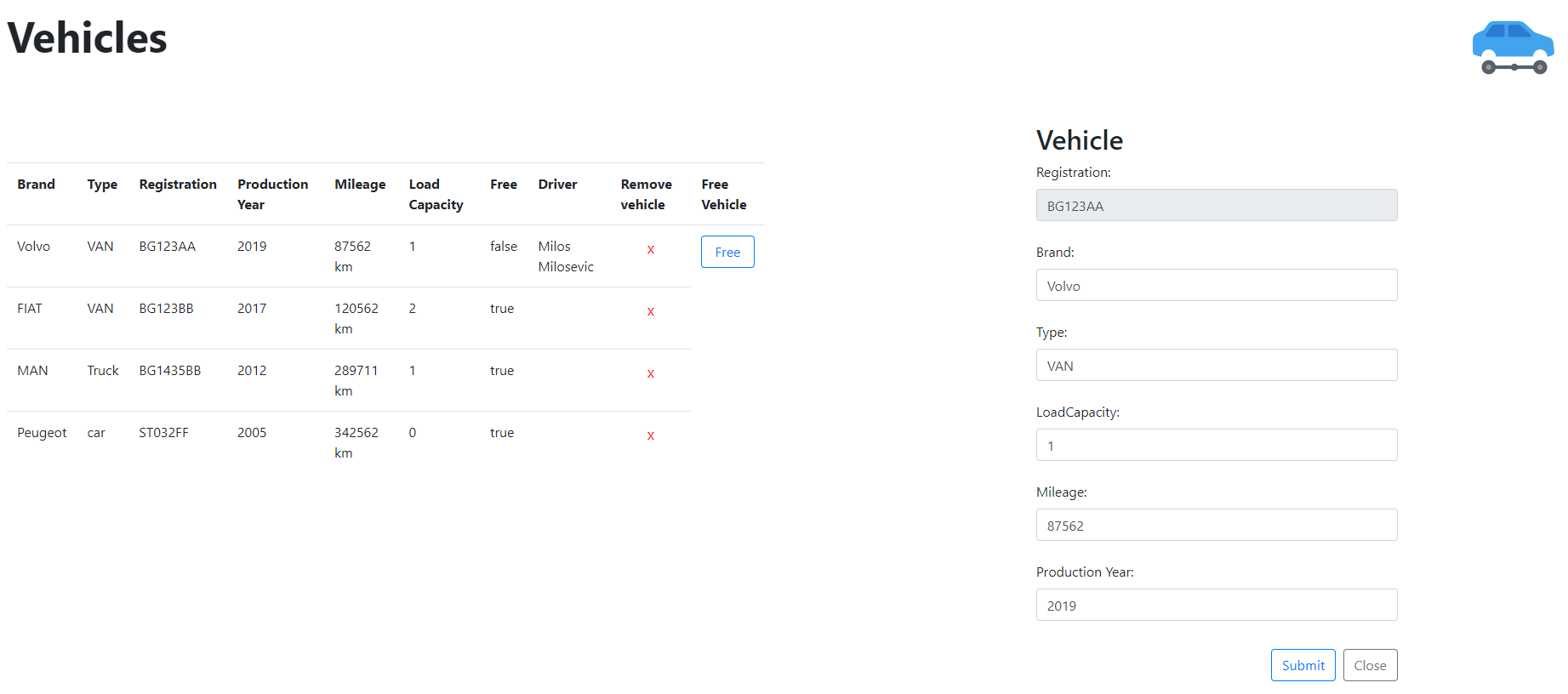
У табу за управљање рампама могуће је ивршити увид у податке рампи за утовар, по складиштима. Одабиром складишта из падајућег менија приказују се подаци о рампама за одабрано складиште.

Осим тога, и идентификационе ознаке рампе, доступне су и информације о томе да ли је нека рампа слободна, и уколико није, ознаке поруџбине која се на тој рампи тренутно утовара.Кликом на дугме „Додај нову рампу (Add new ramp)“ или двоструким кликом на неку од рампи, отвара се форма за унос нове, или измену података постојеће рампе. Пример прегледа рампи са формом за додавање односно измену дат је на слици 37.



Слика 37 - Преглед информација о рампама и форма за унос нове / измену постојеће рампе

### 3.5.6 – Управљање подацима о возилима – таб возила (Vehicles)



Слика 38 – приказ листе за преглед возила и форме за измену / унос

На слици 38 приказан је изглед странице за приказ и манипулацију информацијама о возилима у систему. У табели је приказана листа возила са основним информацијама.

Уколико је возило заузето, односно задужено од стране неког возача, име и презиме возача је видљиво у табели, а на десном крају табеле доступно је дугме „Ослободи“ помоћу којег се возило може раздужити од стране администратора.

**Возило се не може обрисати уколико је задужено од стране возача (слика 39).**

Graphical user interface, text, application

Description automatically generated

Слика 39 – обавештење о неуспешном брисању заузетог возила.

### 3.5.7 – Поруџбине и Извештаји

Табови за преглед поруџбина и извештаја су у потпуности исти као на страници намењеној магационеру и детаљно описани у одељку за странице магационера, па због тога, опис нећемо овде понављати.

# Опис коришћених техничких решења

У овом одељку представићемо основна техничка решења коришћена за израду апликације. Даћемо кратак опис програмских језика и оквира (framework-a) за израду backend и frontend делова апликације, начин њихове комуникације као и комуникације са базом података. Осим тога, представићемо и основну архитектуру оба софтверска решења, начин њиховог повезивања у једну целину односно функционалан систем али и најважније алате, алгоритме и специфичности које ће читаоцу овог рада помоћи да што боље разуме начин израде, функционисања и комуникације између компоненти система.

Софтверско решење система може се поделити у три функционалне целине:

* backend – позадински код – пословна логика и операције над подацима
* frontend – изглед и начин приказа података и интеракција са корисницима
* база података – чување података о чиниоцима и догађајима у систему

У основи, решење је дизајнирано као веб серивс, односно клијент – сервер апликација, при чему сервер подразумева backend део софтверског решења које се извршава на физичком серверу или у cloud-у, док су клијенти рачунари из чијег се веб претраживача приступа подацима.

## 4.1 Backend решење

Backend решење подразумева софтверско решење које као основни задатак има извршавање пословне логике, операције над подацима и извршавање алгоритама неопходних за функционисање система. У перспективи наше апликације, основни задатак backend дела апликације огледа се у приступу и манипулисању над подацима смештеним у бази података, али и извршавању специфичних операција и алгоритама које као резултат дају очекивано понашање апликације и податке које корисник очекује.

Backend део апликације написан је уз помоћ .NET Framework окружења помоћу програмског језика С#.

Решење подразумева веб сервис архитектонску организацију, односно организацију која омогућава комуникацју са клијентским рачунарима путем веб претраживача. Backend решење представља серверски део апликације и заснива се на REST API приступу, који на располагање клијентској страни апликације ставља велики број HTTP метода за приступ, промену, брисање, додавање и измену података у бази (CRUD методе).

Решење је организовано по пројектима и то на начин приказан на слици 40. У наставку изложићемо основну сврху и организацију сваког од пројеката.

Graphical user interface, text, application

Description automatically generated

Слика 40 – пројекти backend решења

### 4.1.1 „WebApplication“

„WebApplication“ је пројекат који садржи контролере који прате функционалне целине података. Сваки контролер излаже низ метода за манипулацију подацима путем HTTP захтева.

На слици 41 приказана је листа контролера. Сваки је намењен приступу и манипилацији над подацима једног чиниоца система.

Text

Description automatically generated with medium confidence

Слика 41 – класе контролера

Методе које контролер излаже представљају REST API методе, односно методе које помоћу HTTP захтева омогућавају манипулацију над подацима. Операције расположиве над подацима прате CRUD принцип односно постоје:

* Метода за добављање података са сервера (HTTP get)
* Метода за додавање нових података у базу (HTTP post)
* Метода за измену постојећих података (HTTP put)
* Метода за брисање података из базе (HTTP delete)

### 4.1.2 „Warehouse.Web.Model“

У оквиру овог пројекта налазе се модели података за комуникацију са клијентима.

Сваки захтев упућен од стране клијента, између осталог садржи и податке који се са клијентске стране шаљу ка серверу. У случају захтева за читањем или брисањем података, то је углавном идентификациона ознака чиниоца односно ентитета у бази података, али у случају писања новог, или измене постојећег ентитена, количина података и њихова сложеност могу бити знатно веће.

Иста логика примењује се и приликом слања одговора са серверске стране, у случају враћања података одређеног чиниоца система.

Због тога, а ради лакше организације и коришћења података, али и ради веће сигурности, сви сложенији захтеви или одговори дефинисани су у облику посебне класе – модела података чија се поља даље користе за приступ и манипулацију подацима.

Пројекат је организован у две целине – директоријума

* Захтеви (Request) – Модели (класе) параметара које серверска страна очекује од клијента при слању одређеног захтева (слика 42)
* Одговори (Response) – Модели (класе) резултата односно одговора сервера на захтев клијента коју клијентска страна може очекивати као одговор на послати захтев за подацима (слика 43)

Text

Description automatically generated Graphical user interface, text

Description automatically generated

Слика 42 – модели захтева Слика 43 – модели одговора на захтеве

На слици 44 приказан је пример једне класе – модела података из система. По истом шаблону формиране су све остале класе.

Text

Description automatically generated

Слика 44 – пример модела захтева (параметара) очекиваних од стране сервера

### 4.1.3 „Warehouse.Model“

У оквиру модел пројекта налазе се класе генерисане на основу базе података, приказ шеме базе података као и логика за приступ подацима.

Како је за комуникацију између базе података и апликације изабран Entity Framework, и то database – first приступ (о овоме ћемо детаљније причати у одељку 4.3), у оквиру овог пројекта извршено је повезивање базе података и апликације, на основу које су аутоматски изгенерисане класе – модели приказани на сликама 45 и 46.

Text

Description automatically generated with medium confidence

Слика 45 – структура аутоматски креираних класа на основу табела у бази

Text

Description automatically generated

Слика 46 – пример класе – модела насталог на основу табеле у бази података

Осим тога, у оквиру овог пројекта налази се и логика помоћу које је омогућен приступ подацима. На слици 47. приказан је код класе „AccessDB“, чијим се инстанцирањем добија веза ка подацима у бази.

Класа представља контекст базе података (па у скалду са тим и наслеђује класу DbContext) и нуди низ „DbSet“ пропертија за приступ подацима сваке од табела у бази.

Text

Description automatically generated

Слика 47 – изглед класе за приступ подацима у бази

### 4.1.4 „Warehouse.Service“

У оквиру сервис пројекта смештена је логика извршавања операција потребна ради исупуњења специфичних захтева неопходних за функционисање система.

Text

Description automatically generated

Слика 48 – преглед сервиса у систему

На слици 48 приказани су сервиси система и то су:

* Сервис са генерисање подразумеване лозинке - у оквиру овог сервиса налази се алгоритам за креирање пордазумеване лозинке по принципу случајног избора 10 караткера. Алгоритам за одабир приказан је на слици 49. Овај сервис користи се за генерисање нове, у случају заборављене лозинке.

A screenshot of a computer

Description automatically generated with medium confidence

Слика 49 – алгоритам за генерисање подразумеване лозинке

* Сервис за слање електронске поруке - на слици 50 приказан је део алгоритма за слање електронске поруке који се користи за комуникацију са корисницима путем електронске поште (слање лозинке или других података на адресу електронске поште).

Алгоритам користи унапред дефинисану адресу и лозинку елетрконске поште и на адресу прослеђену кроз параметре шаље поруку дефинисану помоћу HTML-a.

Text

Description automatically generated

Слика 50 – део алгоритма за слање HTML електронске пруке

* Сервис за креирање докумената – логика за креирање pdf докумената коришћена за креирање извештаја и рачуна, за потребе пројекта коришћена је „third party“ библиотека „Syncfusion.Pdf.Grid“ чија је основна намена динамичко креирање докумената.
* Сервис за претрагу – логика која на основу улазног параметра враћа само оне податке који садрже улазни параметар у предефинисаним пољима
* Сервис са хеширање лозинке – логика хеширања лозинке, помоћу које се додатно штити лозинка корисника, о овом сервису и алгоритму више информација ћемо навести у одељку 4.3

### 4.1.5 „Warehouse.Repository“

У овом пројекту налазе се репозиторијуми, односно класе које садрже логику за приступ подацима.

Text

Description automatically generated with medium confidence

Слика 51 – преглед репозиторијума система

На слици 52 налази се пример једног таквог репозиторијума. Његова основна намена је да омогући и на располагање кориснику стави методе за читање, упис, измену и брисање над подацима у бази. За сваку табелу од интереса у бази података постоји класа репозиторијума која на располагање ставља ове функционаслности.

Text

Description automatically generated

Слика 52 – декларација метода које излаже сваки репозиторијум

## Frontend решење

Frontend део апликације развијен је помоћу Angular 8 програмског језика.

Организација пројекта је таква да веб апликација није „single - page“ у правом смислу те речи, већ се састоји из неколико различитих страница, при чему се странице намењене корисницима система (магационеру, возачу и менаџеру) понашају као „single - page“ странице.

Пројекат је организован кроз компоненте. Свака компонента представња један логички део система и садржи следеће фајлове:

* Html – структура странице
* Css – изглед странице
* Ts – функционалност

Graphical user interface, text

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Слика 53 – компоненте пројекта Слика 54 –структура компоненте

Свака компонента представља функционални сегмент система и може се употребити односно приказати на било ком месту у систему.

У класи „app-routing.module.ts“ дефинисане су све компоненте и путање до њих. На основу тога, претраживач свуда где пронађе таг (<>) са именом компоненте (слика 55), рендерује, односно умеће елементе наведене компоненте.



Слика 55 – додавање компоненте помоћу HTML тага

Такође, у оквиру класа чувара („guard“) дефинисана су права приступа, па тако:

* driver.guard – омогућава приступ компонентама само корисницима чија је улога **возач**
* manager.guard – омогућава приступ компонентама само корисницима чија је улога **менаџер**
* storekeeper.guard – омогућава приступ компонентама само корисницима чија је улога **магационер**
* login.guard – омогућава приступ компонентама **само улогованим (пријављеним) корисницима**

Text

Description automatically generated

Слика 55 – пример „guard“ класе за ограничење права приступа

Осим компоненти, пројекат садржи и низ сервиса чији је основни задатак комуникација са сервером. Сваки сервис нуди низ метода које помоћу HTTP захтева комуницирају, односно шаљу или примају податке са серверске стране.

Graphical user interface, text, application

Description automatically generated

Слика 56– сервиси Frontend пројекта

Text

Description automatically generated

Слика 57 – пример сервиса са HTTP методама за комуникацију са сервером

## Комуникација са базом података

База података, као једна од важнијих целина овог решења, биће детаљније описана и обајшњена у одељку 5. У овом параграфу бавићемо се пре свега начином на који је база повезана са остатком система као и процесима који су ту комуникацију омогућили.

Како је база креирана уз помоћ Microsoft SQL језика, те представља релациону базу података смештену на Microsoft SQL server- у, она се може налазити на истом (као у нашем случају, за потребе тестирања), или на одвојеном (много чешће у пракси) хардверу, односно рачунару или серверу.

Комунијацију са базом података омогућила нам је употреба **Entity Framework** решења. Ово решење омогућило нам је да управљамо и комуницирамо базом података користећи .NET објекте.

Приступ који смо одабрали, и већ раније поменули у тексту је „прво база података (**database first**)“ решење. У пракси, то значи да смо прво креирали релациону базу података, а затим помоћу операција Entity Framework-а изгенерисали објекте помоћу којих тој бази можемо приступати, њом управљати или добављати, мењати и брисати податке.

Diagram

Description automatically generated

Слика 58 – дијаграм употребе Entity Framework -а у

Више о употреби изгенерисаних објеката можете пронаћи у одељцима 4.1.3 и 4.1.5.

## Сигурност и безбедност

Пошто смо за развој овог софтверског решења одбрали веб технологију, што практично значи да је читава апликација видљива на интернету потенцијално и људима који нису запослени у фирми, односно онима за које је она намењена, па тако и евентуалним злонамерним корисницима интернета, сигурност података и безбедност саме апликације били су један од најважнијих аспеката приликом израде пројекта.

Као прво питање током размишљања на ову тему намеће се ограничавање приступа систему на само оне кориснике којима је систем и намењен.

За ове потребе искористили смо врло познату логику аутентификације уз помоћ унапред регистроване адресе електронсе поште (као јавног дела аутентификационог кључа), и лозинке познате само кориснику система.

У тренутку приступа систему, корисник види само почетну страну на којој се не налазе никакви поверљиви подаци. Тек након уноса аутектификационих података он се преусмерава на ону страну која је намењена и прилагођена његовој улози у систему и на располагању има оне податке који су намењени тој улози.

Са техничке стране, овај процес подразумева неколико логичких целина.

Пре свега, у бази података чувају се информације о корисницима, односно налозима регистрованих у систем који имају право коришћења и приступа поверљивим подацима. Ради што веће безбедности све корисничке лозинке се у бази података чувају као **„hash“ вредности**. На сликама 59 и 60 приказан је део алгоритма који вредност лозинке пристигле са клијентске стране претвара у хеш вредност, однсоно обрнутог процеса који приликом провере исправности лозинке проверава подаударност хеш вредности унете и сачуване лозинке.

Text

Description automatically generated

Слика 59 – алгоритам за хеширање лозинке

Text

Description automatically generated

Слика 60 – алгоритам за упоређивање хеш вредности

Након успешног и сигурног чувања података о кориснику, друго питање на које је било потребно дати одговор односи се на начин на који ће се клијентском делу апликације проследити информација о томе да корисник има права приступа систему, али и о његовој улози, односно информација о томе ком делу система и којим подацима одређени корисник има права приступа.

За потребе решавања овог проблема користили смо „**OWIN token based authentication**“, алгоритам за слање и чување потребних аутентификационих података у складишту претраживача.

Употребом **токена**, на клијентској страни, односно у меморији претраживача након успешне пријаве у систем чувају се инфомрације о идентитету и улози корисника, као и специјалан код (**jwt**) који јединствено аутентификује, односно потрвђује идентитет особе која приступа систему.

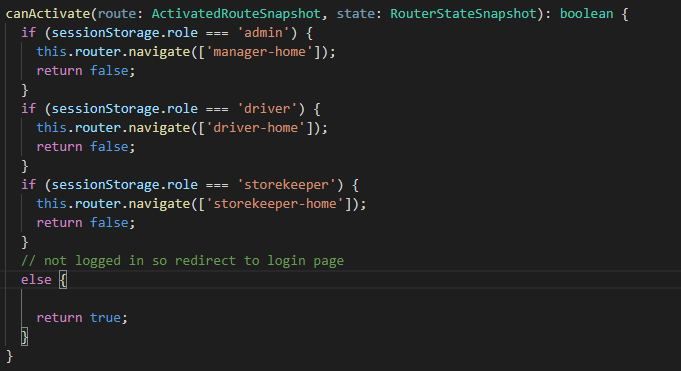
На слици 61 приказан је изглед токена у складишту претраживача. За потребе чувања ових информација изабрали смо складиште сесије (session storage) - једну од неколико врста складишта на претраживачу чија је главна особина то да податке чува само док траје сесија (у пракси, сесија на претраживачу траје док је отворен и активан један таб односно картица претраживача).

Table

Description automatically generated

Слика 61 – приказ аутентификационих подтака који се чувају у меморији претраживача

Употребом података сачуваних у „role“ варијабли, клијентска страна одлучује на коју страну треба проследити корисника након пријаве у систем. Логика за одлучивање о даљем преусмеравању корисника приказан је на слици 62.



Слика 62 – логика за преусмеравање корисника након пријаве у систем

Осим тога, веома битан део процеса представља и „доказивање“ серверској страни да корисник који приступа систему и захтева неке податке има права да те податке од сервера и добије.

Како бисмо били потпуно сигурни да ће корисницима бити омогућен приступ само оним подацима који су њима и намењени, однсоно, како бисмо потпуно искључили могућност не само неовлашћеног приступа систему од стране неидентификованог корисника, већ и приступ делу података од стране корисника чија улога система не одговара тим подацима, дефинисали смо јасна **права приступа**.

Свака од улога у систему (магационер, возач и менаџер) има низ функција којима може да приступи, али су оне недоступне осталим корисницима система.

Провера права приступа врши се на серверкој страни, али је клијентска страна осмишљена тако да се са странице намењене одређеном кориснику система може доћи само до оних података и функција које су њему и намењене, а **корисник са једном улогом не може приступити страницама намењеним осталим корисницима**.

Логика за проверу права приступа одређеној станици на основу улоге корисника у систему приказана је на слици 63. На слици 64 дат је пример алгоритма за утврђивање права приступа на основу улоге корисника.

Text

Description automatically generated

Слика 63 – ограничење приступа станицама на оснву улоге

Text

Description automatically generated

Слика 64 – алгоритам за проверу права приступа на основу улоге корисника

Како би провера права приступа била могућа на серверској страни, сваки захтев упућен од стране клијента ка серверу мора да садржи и jwt токен, на основу кога се идентитет и улога корисника утврђују на серверској страни. Слика 65 садржи логику за престретање и додавање токена сваком HTTP захтеву послатом са клијентске стране.



Слика 65 – логика са пресретање и додавање токена у заглавље HTTP захтева

На серверу, једноставном употребом атрибута на свакој од метода са ограниченим правом приступа, приказаном на слици 66, осигурава се да ће подацима који се као резултат те методе шаљу ка клијенту, право приступа имати само они корисници којима су ти подаци и намењени.

Text

Description automatically generated

Слика 66 – атрибут за проверу права приступа на основу улоге корисника на серверској страни

## 4.5 Управљање верзијама (git)

Управљање верзијама је саставни део процеса развоја софтвера данас. То је процес односно скуп алата и метода који програмеру, или тиму програмера омогућава да безбедно и у великој мери отпорно на грешке, мења, проширује и унапређује изворни код, али и друге податке који су саставни део пројекта.

То се постиже тако што се као почетна тачка узмима тренутак односно стање над фајловима чијим верзијама желимо да управљамо, у ком смо сигурни да су сви подаци ваљани односно исправни.

Свака наредна измена система почиње тако што се од те исправне верзије прави засебна грана на којој се врше измене, а затим се, када се са изменама заврши, та грана спаја у претходно сачувану почетну тачку, стварајући нову верзију (ону која укључује и старе и измењене вредности), која постаје нова почетна тачка.

На овај начин у случају грешке или пропуста програмера, стање фајлова се може вратити на тренутак пре спајања измена које су грешку изазвале. Осим тога, на овај начин се читав процес може рашчланити на мање, једноставније целине, али и пратити ток развоја посматраног система.

У нашем случају, као алат за управљање верзијама изабрали смо један од најпознатијих алата – **git**.

Конкретна имлементација реализована је помоћу вероватно најпопуларнијег бесплатног git алата –<https://github.com/>.

Изворни код и сви остали пропратни подаци овог пројекта налазе се на приватном репозиторијуму до ког се може доћи помоћу следећег линка: <https://github.com/gajin7/web2Project>

На слици 67 приказан је преглед фајлова чијим верзијама се управља на поменутом репозиторијуму.

Graphical user interface, text, application, email

Description automatically generated

Слика 67 – git преглед репозиторијума

## 4.6 Ценовник

Управљање производима као основном јединицом а уједно и најважнијим објектима у систему представљало је веома велики изазов пре свега због порменљивих цена производа као и чињенице да једном промењена цена не постаје ирелевантан појам у систему јер и у случају промене цене, рачун и извештаји морају садржати ону цену артикла која је важила у тренутку њиховог (првог) креирања.

Овај проблем решен је увођењем засебне табеле базе података, као и логике за чување, промену и добављање цена сваког појединачног производа.

У овом одељку даћемо кратак преглед организације и процеса који су коришћени како би се осигурало да ће историјат цена сваког пројекта бити запамћен и по потреби добављен из базе.

О табели „Ценовник“ биће више речи у одељку о бази података, ипак, ради лакшег разумевања и овде ћемо укратко обајаснити њен садржај:

* Id – јединстевни идентификатор
* ValidFrom – датум од кад унета цена важи
* ValidTo – датум до кад унета цена важи
* ItemId – идентификатор производа чија се цена чува
* Value – вредност у подразумеваној валути

У тренутку креирања новог производа у бази, у табелу Ценовник додаје се нови објекат који чува његову у том тренутку унету цену.За датум од када унета цена важи узима се тренутак креирања, а датум до када она важи остаје празан. **Тек у случају промене цене, датум до када стара цена важи се мења, и то датумом промене односно уносом нове цене.**

На овај начин тренутна цена неког производа може се препознати по томе што нема датум до када важи (ValidTo = null).

Осим тога, у случају накнадног креирања извештаја (ако он више не постоји у директоријуму извештаја) или рачуна, за цене артикала узимају се оне цене које су важиле у тренутку првог креирања извештаја. Логика за проверу однсоно проналазак цене која је важила у одређеном тренутку у прошлости, приказана је на слици 68. Валидна цена је она која је важила у тренутку креирања извештаја (или рачуна), односно треба пронаћи ону цену чији је датум од када важи већи, а датум до када важи мањи од датума креирања извештаја (или рачуна).

A screen shot of a computer

Description automatically generated with low confidence

Слика 68 – логика за проналазак одговарајуће тренутне или цене у прошлости

**Корисници система у сваком тренутку виде тренутно важећу цену за сваки артикал.**

# Опис базе података

У ранијим деловима овог рада, већ смо, више пута помињали и делимично објашњавали делове базе података, њихову улогу, функционалности које оне омогућавају, али и начин комуникације и приступа, промене и брисање података, прва приступа, сигурност података, и друго.

У овом поглављу даћемо кратак преглед релационе базе података, њених ентитета и њихових међусобних веза.

На слици 69 дат је приказ енититета, односно табела у релационој бази података креираних у оквиру Microsoft SQL Server Management Studio алата.

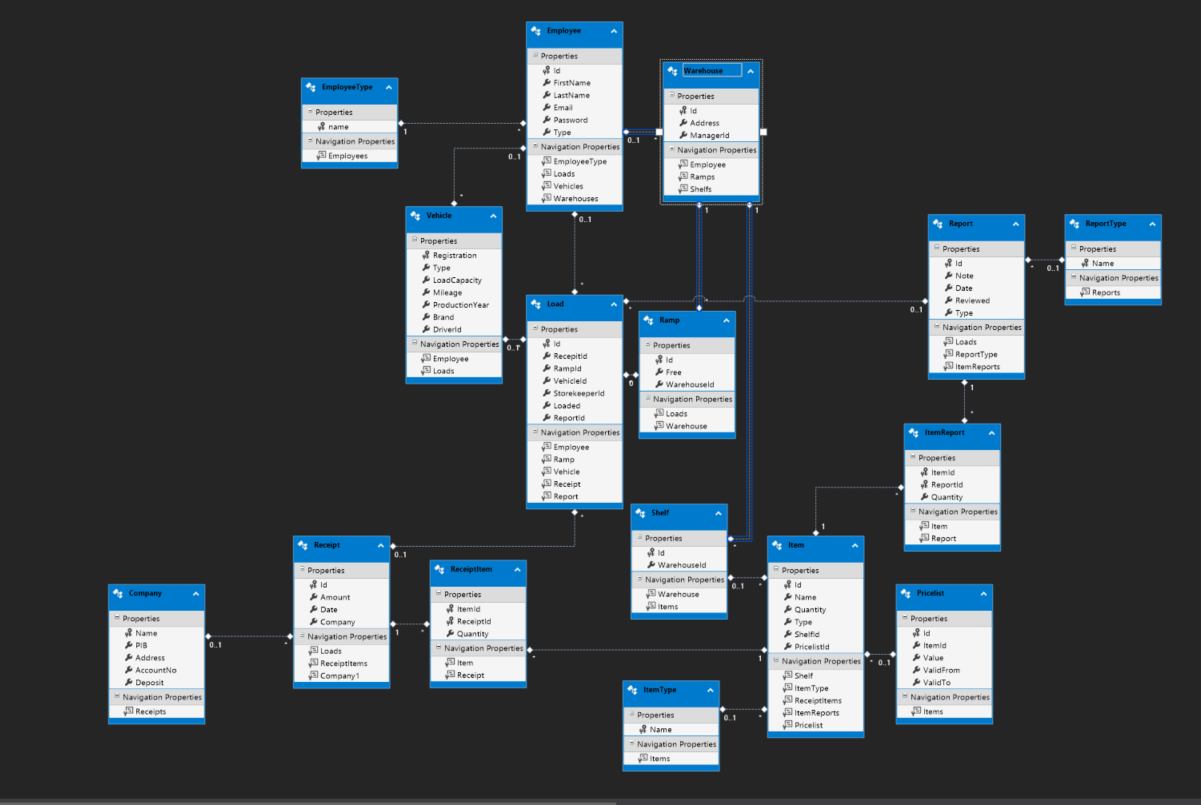
A picture containing table

Description automatically generated

Слика 69 – Прегед табела у бази података

На слици 70 приказана је шема базе података са ентитетима, њиховим обележјима и идентификаторима као и везе иземђу њих. Приказана шема базе података настала је на основу већ креираних табела релационе базе података уз помоћ Entity Framework-а и на основу ње креиран је модел података раније помињан у овом раду.

У наставку текста, даћемо кратак опис сваког од ентитета, укључујући његова обележја и релације (везе) са осталим ентитетима, али и његову улогу у систему.



Слика 70 -шема базе података

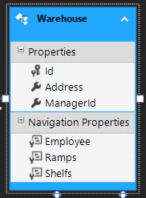
## Ентитети релационе базе података

### Ентитет „Warehouse“ – складиште

Ентитет складиште представља основни ентитет чија је улога опис сваког појединачног складишта система. Постојање овог ентитета неопходно је пре свега ради лакше организације. У реалним условима, једна компанија може имати више складишта која такође могу бити на потпуно различитим локацијама. Због тога је неопходно знати у ком од магацина се роба тачно налази, али и у који магацин и где (на коју адресу) возач треба да дође по робу.

Обележја ентитета Складиште и њихова улога су:

* Id – јединствени идентификатор, такође и примарни кључ ентитета
* Address – адреса на којој се складиште налази
* ManagerId – идентификатор запосленог у компанији који се налази на позицији менаџера посматраног предузећа

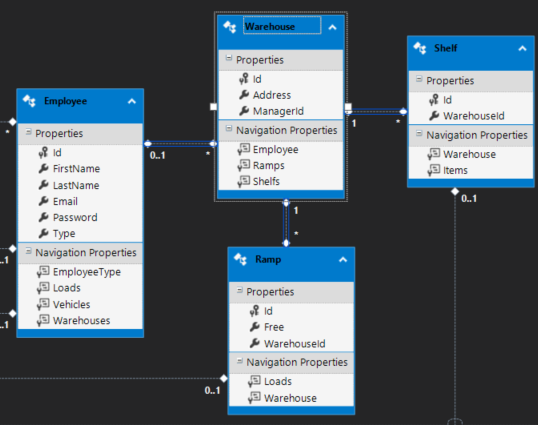


Слика 71 – Ентитет Складиште

Релације са осталим енититетима ентитета складиште и њихови кардиналитети су:

* Складиште – Рампе – свако складиште може имати више рампи, а свака рампа мора припадати тачно једном складишту (N:1)
* Складиште – Полице - свако складиште може имати више полица, а свака полица мора припадати тачно једном складишту (N:1)
* Складишта – Запослени – Складиште може имати највише једног, а не мора имати ни једног управника, а један запослени може бити управник већег броја складишта (0..1 : N)

**Неки кардиналитети, попут везе „Складишта – Запослени“ прилагођени су функционисању система**, па тако, у пракси свако скалдиште треба увек да има свог управника али ипак, због могућности брисања података из базе, уведен је специјалан случај у коме он није познат (неопходно да би скалдишта могла и даље да постоје у бази, чак и у случају да су сви запослени обрисани из базе).



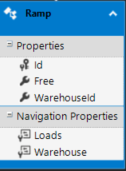
Слика 72 – везе ентитета Складиште са кардиналитетима

### Ентитет „Ramp“ - рампа

Рампа у контексту нашег система представља место на коме је могуће преузети и утоварити робу у возило. Свако складиште може имати већи број рампи ради што брже испоруке робе, па је то управо главни разлог због кога је неопходно водити њихову евиденцију. Свака поруџбина на себи садржи и ознаку рампе, како би магационери и возачи знали на коју од рампи треба да донесу робу односно довезу возило. Осим тога, она садржи информацију о томе да ли је рампа тренутно слободна, како би возач знао у ком тренутку може да приступи рампи ради утовара поручене робе.

Обележја ентитета Рампа

* Id – јединствени идентитификатор (примарни кључ)
* Free – информација о томе да ли је рампа тренутно слободна или не
* WarehouseId – идентификатор складишта у коме се рампа налази

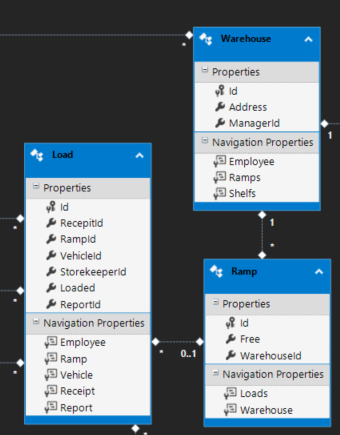


Слика 73 – Ентитет Рампа

Релације ентитета Рампа

**У наставку овог одељка описиваћемо само оне везе које нису претходно већ описане приликом описа неког од већ обрађених ентитета. Ипак, ради лакшег сналажења, такве везе ћемо навести без даљег обајшњења.**

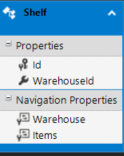
* *Складиште - Рампа*
* Рампа – Поруџбина – Поруџбина може имати највише једну рампу, али једна рампа може бити локација утовара за више различитих поруџбина (0..1:N)



Слика 74 – везе ентитета Рампа са кардиналитетима

### Ентитет „Shelf“ - Полица

Сваки артикал мора бити смештен на неку од полица које се налазе у складишту. Полица је моделована у оквиру базе података пре свега како би олакшака сналажење магационера у нарочито великим складиштима и убрзала проналажење потребног артикла



Слика 75 – Енитет Полица

Идентификатори ентитета Полица:

* Id – јединствени идентитификатор (примарни кључ)
* Warehouse Id – идентификатор складишта у коме се рампа налази

Релације ентитета Полица:

* *Складиште – полица*
* Полица - Роба – роба се може налазити на тачно једној полици, на једној полици може бити више артикала (N:0..1)

Graphical user interface, application, Teams

Description automatically generated

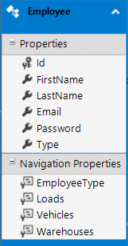
Слика 76 – Везе ентитета Полица са кардиналитетима

### Ентитет „Employee“ - Запослени

Ентитет Запослени неопходан је ради чувања података о запосленом. Ови подаци користе се за идентификацију особе која врши неке промене у систему, задужена је за одређене процесе или приликом аутентификације у случају пријаве у систем.

Обележја ентитета Запослени:

* Id – јединствени идентитификатор (примарни кључ)
* First Name – име запосленог
* Last Name – презиме заполсеног
* Email – адреса електронске поште
* Password – хеш вредност лозинке за пријаву на систем
* Type – тип корисника неопходан за утврђивање права приступа и ограничења



Слика 77 – Енитет Запослени

Релације ентитета Запослени:

* *Складиште – Запослени*
* Запосленои – Тип заполсеног – Сваки радник мора имати тачно један тип, а исти тип радника може имати већи број запослених (1:N)
* Запослени – Возило – Запослени чији је тип однсоно улога у систему возач може задужити више возила, док свако возило може бити задужено од стране само једног возача у посматраном тренутку, или може бити слободно (N:0..1)
* Запослени – Поруџбина – Запослени чија је улога магационер може бити задужен за више поруџбина, док свака поруџбина има једног магационера који је за њу задужен (N:0..1)

Graphical user interface, application, Teams

Description automatically generated

Слика 78 – Везе енитета Запослени са кардиналитетима

### Ентитет „Employee Type“ – Тип корисника

Тип запосленог моделован је као посебан ентитет због своје важности за сигурност система. На основу овог типа креиране су улоге корисника у систему а на основу њих дефинисана права приступа.

Типови корисника тренутно обухваћени системом одговарају улогама корисника и то су:

* Магационер
* Возач
* Менаџер (администратор)

Обележја ентитета тип корисника

* Name - назив типа корисника (примарни кључ)

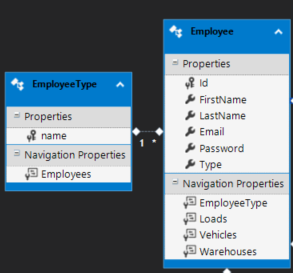
Релације ентитета Тип корисника

* *Запосленои – Тип заполсеног*

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Слика 79 – Ентитет Тип корисника



Слика 80 – Везе ентитета Тип корисника

### Ентитет „Vehicle“ – Возило

Ентитет Возило садржи податке неопходне за идентификацију возила којим одређени возач управља или чека да га неки од возача задужи. Основни мотив за моделовање овог ентитета одгледа се у потреби да се прате информације о свим возилима у компанији, али и лакше утврди да ли неких возила има премало или превише како би се на прави начин реаговало (уколико су, на пример, возила одређене категорије често слободна и то у већој количини, можда би њихов број могао да се редукује).

Обележја ентитета Возило:

* Registration – регистрациона ознака возила (примарни кључ)
* Type – тип возила
* Load Capacity – капацитет
* Milage – пређена километража
* Production Year – година производње
* Brand – марка возила
* Driver Id – идентификатор возача уколико је возило задужено

Graphical user interface, text, application

Description automatically generated

Слика 81 – ентитет Возило

Релације ентитета Возило:

* *Запослени – возило*
* Возило – поруџбина – Свака поруџбина може имати тачно једно возило у које треба да се утовари, а једно возило може бити предвиђено за утовар више различитих поруџбина (N:0..1)

Graphical user interface, text, application

Description automatically generated

Слика 82 – везе ентитета Возило

### Ентитет „Load“ – Продуџбина

Ентитет Поруџбина дефинише скуп информација неопходних за креирање пошиљке робе од магацина до купца одабране робе.

Као један од најкомплекснијих ентитета у систему, Поруџбина игра централну улогу у функционисању читавог процеса па је због тога повезана са великим бројем ентитета у бази.

Обележја ентитета Поруџбина:

* Id – јединствени идентитификатор (примарни кључ)
* Receipt Id – идентификатор рачуна који се издаје за посматрану поруџбину
* Ramp Id – идентификатор рампе на којој се поруџбина утовара
* Vehicle Id – идентификатор возила које је транспортовало или ће транспортовати поруџбину
* Storekeeper Id – идентификатор магационера задуженог за паковање робе за поруџбину
* Loaded – индикатор да ли је роба утоварена (користи је пре свега возач да би знао када може да крене)
* Report Id – идентификатор извештаја који је генерисан као евиденција излаза робе приликом утовра поруџбине

Graphical user interface, application

Description automatically generated

Слика 83 – ентитет Поруџбина

Релације ентитета Поруџбина

* *Запослени – поруџбина*
* *Рампа – поруџбина*
* *Возило – поруџбина*
* Проуџбина – рачун – свака поруџбина има један рачун, а на рачуну се може наћи више поруџбина (0..1: N)
* Поруџбина – извештај - свака поруџбина има један ивештај, а у ивештају се може наћи више поруџбина (0..1: N)

Graphical user interface, application

Description automatically generated

Слика 84 – Везе ентитета Поруџбина

### Ентитетн „Report“ – Извештај

Ентитет Извештај чува најважније податке о сваком генерисаном извештају у току рада система. На овај начин омогућена је накнадна провера поступака корисника у систему и праћење уласка, односно изласка робе.

Обележја ентитета Извештај

* Id – јединствени идентитификатор (примарни кључ)
* Note – евентуална забелешка (настаје најчешће приликом прегледа)
* Date – датум креирања
* Reviewed - инидкатор да ли је извештај прегледан од стране администратора
* Type – тип (улазни/излазни) извештаја

Graphical user interface, text, application, chat or text message

Description automatically generated

Слика 85 – Ентитет Извештај

Релације ентитета Извештај

* *Поруџбина – Извештај*
* Ивештај - Тип извештаја – Сваки извештај има само један тип, док више извештаја може бити истог типа (0..1:N)
* Ивештај - Ентитет релације Артикал – Извештај – Сваки извештај садржи листу артикала на које се односи. Како артикли који се налазе на извештају могу имати количину која је већа од 1, веза између извештаја и артикала представљена је новим ентитетом. Извештај може имати више различитих скупова ариткала на које се односи, сваки скуп артикала се односи само на један извештај (N:1)

A screenshot of a computer

Description automatically generated

Слика 86 – Везе ентитета Извештај

### Ентитет „Report Type“ – Тип извештаја

Енититет Тип извештаја дефинише различите врсте извештаја у систему.

У тренутној имплементацији систем подржава

* Улазне извештаје – извештај о заприми робе
* Излазне извештаје – извештај о издавању робе

Обележја ентитета Тип Извештаја

* Name - назив типа извештаја (примарни кључ)

Graphical user interface, text, application

Description automatically generated

Слика 87 – Ентитет Тип извештаја

Релације ентитета Тип извештаја

* *Извештај – Тип извештаја*

Graphical user interface, text, application

Description automatically generated

Слика 88 – Везе ентитета Тип извештаја

### Ентитет „Item“ - Артикал

Ентитет Артикал описује једну јединицу робе – производ. Артикал представља основну јединицу и централни објекат система.

Обележја ентитета Артикал

* Id - јединствени идентитификатор (примарни кључ)
* Name – назив производа
* Quantity – количина прозивода тренутно расположива
* Type – тип производа
* Shelf Id – идентификатор полице на којој се произод налази
* Price List Id – идентификатор помоћу којег се проналази тренутно важећа цена производа у ценовнику

Graphical user interface, text, application

Description automatically generated

Слика 89 – ентитет Артикал

Релације ентитета Артикал

* *Артикал – Полица*
* *Артикал – Ивештај*
* Артикал – Тип артикла – сваки артикал има тип и више артикала могу припадати истом типу (0..1: N)
* Артикал – Ценовник – Сваки артикал може имати више цена (од тога само једну важећу), а сваки члан у ценовнику се односи на тачно један артикал (N:0..1)
* Артикал – Ентитет релације артикал рачун – на рачуну се налази листа артикала од којих сваки има своју количину, због тога се, баш као и у случају извештаја уводи нови ентитет који представља везу између артикла и ивештаја, односно скуп артикала који се налазе на извештају (N:1)

Graphical user interface, text, application, chat or text message

Description automatically generated

Слика 90 – везе ентитета Артикал

### Ентитет „Price List“ – Ценовник

Као што смо већ навели у одељку посвећеном ценовнику, тема чувања цена кроз историју промене цена је веома битна за правилно функционисање система. Због тога, али и звог велике количине података која би се потенцијално могла наћи у бази (услед честог мењања цена артикала), било је неопходно моделовати ценовник као посебан ентитет.

Обележја ентитета Ценовник

* Id - јединствени идентитификатор (примарни кључ)
* Item Id – идентификатор артикла на који се ценовник односи
* Valid From – датум почетка важења цене
* Valid To – датум завршетка важења цене
* Value – контректна вредност цене у подразумеваној валути

Graphical user interface, text, application

Description automatically generated

Слика 91 – Ентитет Ценовник

Релације ентитета Ценовник

* *Артикал - Ценовник*

Graphical user interface, text, application, chat or text message

Description automatically generated

Слика 92 – Везе ентитета Ценовник

### Ентитет „Item Type“ – Тип артикла

Ентитет Тип артикала садржи информације о различитим врстама артикала у систему. На овај начин направљена је класификација артикала и олакшана претрага у оквиру система (на пример, много је лакше претраживати систем на основу типа „аутомобилска гума“, него појединачно проверавати који све артикли заправо јесу аутомобилске гуме):

У тренутној имплементацији постоје следећи типови артикала (слика 93):

* Акумулатор
* Ауто – део
* Боја
* Аутомобилска гума
* Алат
* Непознато

Table

Description automatically generated

Слика 93 – Типови артикала у тренутној имплементацији система

Обележја ентитета Тип артикла

* Name – назив типа артикла (примарни кључ)

Graphical user interface, text, application

Description automatically generated

Слика 94 – Ентитет Тип артикла

Релације ентитета Тип артикла

* *Артикал – Тип артикла*

Graphical user interface, application

Description automatically generated

Слика 95 – Везе ентитета Тип артикла

### Ентитет „Item Report“ – Ентитет релације Артикал – Извештај

Овај ентитет настао је на основу везе Артикал – Извештај. Како се на сваком извештају налази, односно може налазити низ различитих артикала, али истовремено сваки од њих има своју количину која не мора бити једнака 1, било је неопходно дефинисати нови ентитет који ће моделовати ову везу, како бисмо сачували информацију о томе која количина сваког артикла се налази на извештају.

Обележја ентиета релације Артикал – Извештај

* Item Id – идентификатор артикла на који се односи
* Report Id – идентификатор извештаја на који се односи
* Quantity – количина артикла која се налази у извештају

**Item Id и Report Id заједно чине примарни кључ овог ентитета**

Graphical user interface, text, application, chat or text message

Description automatically generated

Слика 96 – Ентитет релације Артикал – Извештај

Релације ентитета релације Артикал – Ивештај

* *Артикал – Ентитет релације Артикал – Извештај*
* *Извештај - Ентитет релације Артикал – Извештај*

Graphical user interface, application

Description automatically generated

Слика 97 - Везе Ентитета релације Артикал – Извештај

### Ентитет „Receipt“ – Рачун

За сваку проуџбину, однсоно сваки издат (продат) артикал, компанија је у обавези да изда фискални рачун и фактуру (рачун у оквиру нашег система). Осим тога, комапија је дужна да извесни период чува и по потреби прикаже све издате рачуне.

Због свега овога, али и због потребе да се води евиденција и евентуално генеришу финансијкси извештаји, појавила се потреба за моделовањем ентитета Рачун.

Обележја ентитета Рачун

* Id - јединствени идентитификатор (примарни кључ)
* Amount – укупна цена робе на рачуну
* Date – датум издавања рачуна
* Company – идентификатор компаније која купује робу (којој се рачун издаје)

Graphical user interface, text, application

Description automatically generated

Слика 98 – Ентитет Рачун

Релације ентитета Рачун

* *Поруџбина – рачун*
* Рачун – Ентитет релације Артикал – Рачун (N:1)
* Рачун – Компанија – Сваки рачун намењен је само једној компанији, док једна компанија може направити више поруџбина, а самим тим имати и више рачуна (0..1: N)

Graphical user interface, application, Teams

Description automatically generated

Слика 99 – Везе ентитета рачун

### Ентитет „Item Receipt“ - Ентитет релације Артикал – Рачун

Из идентичних разлога као и Ентитет релације Артикал – Извештај, настао је и Ентитет релеације Артикал Рачун. Сваки рачун садржи листу артикала на које се односи, али сваки од тих артикала има и своју количину. Из тог разлога било је неопходно моделовати нови ентитет од релације Артикал – Рачун.

Обележја ентитета релације Артикал – Рачун

* Item Id – идентификатор артикла на који се односи
* Receipt Id – идентификатор рачуна на који се односи
* Quantity – количина прозивода

Graphical user interface, text, application, chat or text message

Description automatically generated

Слика 100 – Ентитет релације Артикал – Рачун

**Item Id и Receipt** **Id заједно чине примарни кључ овог ентитета**

Релације ентитета релације Артикал – Рачун

* *Артикал - Ентитет релације Артикал – Рачун*
* *Рачун – Ентитет релације Артикал - Рачун*

Graphical user interface, application

Description automatically generated

Слика 101 – Везе ентитета релације Артикал – Рачун

### Ентитет „Company“ – Компанија

Овај ентитет садржи податке о компанији – купцу робе. Моделовање овог ентитета изазвано је потребом да се подаци о сваком купцу чувају на једном месту, како би били лако доступни али и како би могло да се провери да ли одређени купац има права куповине, да ли је упратио депозит и друго.

Обележја ентитета Компанија

* Name – назив компаније
* PIB – идентификациони број (примарни кључ)
* Address – адреса компаније
* Account No – број жиро рачуна
* Deposit – индикатор да ли је компанија уплатила обавезан депозит

Graphical user interface, text, application

Description automatically generated

Слика 102 – Ентитет Компанија

Релације ентитета Компанија

* *Рачун - Компанија*

Graphical user interface, text, application

Description automatically generated

Слика 103 – Везе ентитета Компанија

# Закључак

Увођењем система за управљање складиштима и процесима у оквиру њих олакшао би се и убрзао процес уноса, прегледа и обраде података како у већим тако и у мањим складиштима. Ипак, акценат овог рада био је пре свега на поједностављивању и убрзавању логистичких процеса оних компанија чији су складиштени капацитети веома велики, па је самим тим време и број запослених неопходних за функционисање процеса у знатној мери смањен.

Имплеметацијом решења описаног у овом раду, такође, добио би се и алат уз помоћ кога би велики системи могли да испрате и исконтролишу рад система, али и уклоне или у значајној мери смање евентуалне грешке.

Као главне препреке у реалној имплементацији овог система видимо пре свега инфраструктурална улагања која подразумевају улагање у сервере, људство за њихово одржавање али и уређаје намењене запосленима.

Осим тога, за успешну имплементацију оваквог система потребно је како време, тако и средства али и кадар чији би примарни задатак била обука нових али и постојећих корисника, као и извесна доза прилагођавања постојећих процеса и навика запослених, како би они одговарали раду овог система.

Са друге стране, систем би, након имплементације омогућио јасније дефинисан начин извршавања неопходних операција, али и у значајој мери скратио време за њихово извршавање. Осим тога, све промене у систему би могле бити испраћене, прегледане и евентуално кориговане, па би се тако грешке и евентуални губици настали као њихова директна последица могли у значајној мери смањити.

Највећи простор за унапређење система огледа се у скалирању захтева за приступ подацима (у случају веома великих система), као и евентуалну интеграцију са осталим технолошким решењима која већ постоје у оквирима пословног процеса.

Тако на пример, систем би било могуће повезати са фискалним касама, па би се већ постојећем рачуну могао додати и фискални исечак. Повезивање је могуће и са штампачима, ради штампања генерисаних извештаја и фактура, читачима бар кодова ради лакшег проналаска робе или спремљене пошиљке (поруџбине), и друго.

Осим тога, као веома битан део система навели смо генерисање извештаја помоћу којих се бележе и прате поступци и процеси који се одвијају у самом систему. За потребе овог пројекта, у систем извештаја интегрисани су само најважнији процеси, односно улаз и излаз робе из система, али, ради добијања веће контроле и јасније слике догађаја у читавом систему, систем извештавања би било могуће или чак пожељно проширити на што већи број процеса. У идеалном случају, сваки „клик“ у систему би био обухваћен и забележен у неком од извештаја.

Ипак, за проналазак додатних простора за унапређивање, али и скривених мана система, један овакав систем, пре употребе у реалним пословним условима, било би неопходно веома темељно тестирати на стварним захтевима неке од компанија којима је намењен.

# Речник појмова

[Microsoft](https://sr.wikipedia.org/wiki/Majkrosoft) **.NET Framework**  - [софтверска платформа](https://sr.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D0%B2%D0%B5%D1%80%D1%81%D0%BA%D0%B0_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0) доступна на [Windows](https://sr.wikipedia.org/wiki/Microsoft_Windows) [оперативном систем](https://sr.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B8_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC)у. Укључује велики број готових библиотека за решавање проблема у програмирању и [виртуелну машину](https://sr.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B5%D0%BB%D0%BD%D0%B0_%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%B0) која управља извршавањем програма писаних специјално за .NET Framework. .NET подржава више [програмских језика](https://sr.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%81%D0%BA%D0%B8_%D1%98%D0%B5%D0%B7%D0%B8%D0%BA) ([VB.NET](https://sr.wikipedia.org/w/index.php?title=VB.NET&action=edit&redlink=1), [C#](https://sr.wikipedia.org/wiki/C_Sharp)...).

Репозиротијум (Repository) – шаблон који омогућава мапирање података добијених из базе података у модел потребан за даљи рад програма.

Сервис (Service) – шаблон који омогућава извршавање одређене логике обраде над подацима, и поновну употребу на различитим местима.

Хeширање (Hash) - **Хеш функција** је сваки алгоритам који подацима произвољне дужине додељује податке фиксне дужине. Вредност коју враћа хеш функција назива се **хеш вредност** или **хеш код**.

Entity Framework – скуп алата који омогућавају лаку комуникацији апликације и базе података, као и креирање модела на основу већ креиране базе, и обрнуто.

Релациона база података – један од типова базе података у коме је организација података заснована на релационом моделу. Подаци су организовани у скуп релација између којих се дефинишу одређене везе.

Microsoft SQL – релациона база података развијена од стране Microsoft-а

Microsoft SQL server – софтверски алат за развој и упраљање Microsoft SQL релационим базама података

OWIN token-based authentication – стандардна .NET логика за сигурну и идентификовану комуникацију између различитих веб апликација

Примарни кључ – Обележје, однсоно колона у бази података која јединствено идентификује сваку ставку (врсту) у табели

# Скраћенице

ПИБ – порески идентификациони број

SQL – Structured Query Language

OWIN - Open Web Interface for .NET

HTTP - Hypertext Transfer Protocol

HTML - Hypertext Markup Language

CSS - Cascading Style Sheets

TS - TypeScript

Jwt - JSON Web Token

# Литература

Документација .NET framework-a - <https://docs.microsoft.com/en-us/dotnet/>

Документација C# програмског језика - <https://docs.microsoft.com/en-us/dotnet/csharp/>

Документација Entity Framework-а - <https://docs.microsoft.com/en-us/ef/>

Документација ASP.NET - https://docs.microsoft.com/en-us/aspnet/web-api/

Документација Microsoft SQL-a - <https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>

Документација Microsoft SQL Server-a - <https://docs.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver15>

Документација SyncFusion, библиотеке за генерисање PDF докумената - <https://help.syncfusion.com/>

TypeScript званична документација - <https://www.typescriptlang.org/docs/>

OWIN аутентифијација, документација - <https://docs.microsoft.com/en-us/aspnet/aspnet/overview/owin-and-katana/>

# Биографија

Јован Гајин рођен је 11. маја 1996. године у Београду. Завршио је основну школу „Милан Хаџић“ у Војки, а затим друштвено – језички смер гимназије у „Земунској гимназији“ у Београду. Школске 2015/2016. године уписује Факултет техничких наука у Новом Саду, на студијском програму Примењено софтверско инжењерство. Положио је све испите прописане планом и програмом.