B&B
VIŠJA STROKOVNA ŠOLA

Diplomsko delo višješolskega strokovnega študija
Program: Logistično inženirstvo
Modul: Cestni promet

# OPTIMIZACIJA ROČNEGA VODENJA ZALOG S POMOČJO SODOBNIH INFORMACIJSKIH TEHNOLOGIJ

Mentor: spec. Jošt Šmajdek, dipl. inž. teh. prom.
Lektorica: Ksenija Pečnik, prof. slov.

Kandidat: Aljaž Mikulić

Kranj, junij 2025

## ZAHVALA

Zahvaljujem se mentorju Joštu Šmajdku, dipl. inž. teh. prom.

Zahvaljujem se tudi lektorici Kseniji Pečnik, prof. slov., ki je mojo diplomsko delo jezikovno in slovnično pregledala.

**IZJAVA**

Študent Aljaž Mikulić izjavljam, da sem avtor tega diplomskega dela, ki sem ga napisal pod mentorstvom spec. Jošta Šmajdka, dipl. inž. tehnol. prom.

Skladno s 1. odstavkom 21. člena Zakona o avtorski in sorodnih pravicah dovoljujem objavo tega diplomskega dela na spletni strani šole.

Dne: _____ Podpis: _____

## POVZETEK

Diplomsko delo raziskuje ustvarjanje programskega orodja za upravljanje zalog v skladiščih z uporabo sodobnih informacijskih tehnologij – umetne inteligence (ChatGPT). Primarni cilj je razviti uporabniku prijazno programsko opremo za poenostavitev sledenja zalog in kreiranja internih dokumentov. Programski jezik Python je bil izbran zaradi svoje vsestranskosti, preprostosti in združljivostjo z umetno inteligenco. Med pripravo naloge pridobimo praktične izkušnje pri načrtovanju arhitekture programske opreme, oblikovanju grafičnega vmesnika in upravljanju podatkov. Projekt pokaže, kako je možno z minimalnim znanjem programiranja v kombinaciji z uporabo orodij umetne inteligence učinkovito ustvariti aplikacije uporabne kakovosti. Uspešno implementiramo stabilno in funkcionalno rešitev za upravljanje zalog. Izdelava izvršljive datoteke (.exe) olajša distribucijo in praktično uporabo. Projekt poudarja dostopnost digitalne transformacije tudi za manjše organizacije z omejenimi informacijskimi viri. Navsezadnje se integracija ChatGPT izkaže za zelo koristno, saj znatno pospeši proces razvoja in izboljša splošno kakovost programa.



## KLJUČNE BESEDE

- vodenje zalog
- optimizacija vodenja zalog
- informacijske tehnologije
- umetna inteligenca

## ABSTRACT

This thesis explores the creation of a software tool for warehouse inventory management using modern information technologies – artificial intelligence (ChatGPT). The primary goal was to develop a user-friendly software to simplify inventory tracking and internal document creation. The Python programming language was chosen for its versatility, simplicity and compatibility with artificial intelligence. We gained practical experience in software architecture design, graphical interface design and data management. The project demonstrated how minimal programming knowledge combined with artificial intelligence tools can effectively create applications of usable quality. A stable and functional inventory management solution was successfully implemented. Creating an executable file (.exe) facilitated distribution and practical use. The project emphasizes the accessibility of digital transformation even for smaller organizations with limited IT resources. Ultimately, the ChatGPT integration proved to be very useful, as it significantly accelerated the development process and improved the overall quality of the program.

## KEYWORDS

- inventory management
- inventory management optimization
- information technology
- artificial intelligence

# KAZALO

## KAZALO SLIK

## POJMOVNIK

ChatGPT: je napredna umetna inteligenca, ki jo je razvil OpenAI ter lahko razume in ustvarja besedilo, podobno človeškemu. Lahko mu postavljate vprašanja, dobivate razlage, pišete dokumente ali celo ustvarjate kodo. Deluje kot pametni asistent, ki vam pomaga reševati težave, se učiti novih stvari ali avtomatizirati opravila s klepetom z vami v naravnem jeziku.

Python: je priljubljen programski jezik, enostaven za branje, odličen tako za začetnike kot za profesionalce. Uporablja se za vse od spletnega razvoja do avtomatizacije in analize podatkov.

GUI: je kratica za grafični uporabniški vmesnik (ang. Graphical User Interface). To je tisto, kar uporabnikom omogoča vizualno interakcijo s programsko opremo – z uporabo oken, gumbov in menijev – namesto tipkanja ukazov.

tkinter: je standardna knjižnica Python za ustvarjanje grafičnih uporabniških vmesnikov (GUI). S tkinterjem lahko ustvarite okna, gumbe, obrazce in druge interaktivne elemente za svoje programe.

WMS: ang. Warehouse Management System (sistem za upravljanje skladišča) je intuitivna programska rešitev, s katero lahko celovito upravljate, nadzorujete in optimizirate svoje skladišče.

Mikropodjetje: v kategoriji malih in srednjih podjetij se mikropodjetje opredeljuje kot tisto, ki ima manj kot 10 zaposlenih in ima letni promet in/ali letno bilančno vsoto, ki ne presega 2 milijonov EUR.

notacija: sistem znakov za zapisovanje kode, glasbe

implementacija: uporaba, uresničitev določenih tehničnih rešitev v informacijskem sistemu

ERP: načrtovanje virov poslovne organizacije se nanaša na vrsto programske opreme, s katero organizacije upravljajo vsakodnevne poslovne dejavnosti, kot so računovodstvo, nabava, upravljanje projektov, upravljanje tveganja in skladnost ter postopki oskrbovalne verige.

## KRATICE IN AKRONIMI

| | |
|---|---|
| EMŠO: | enotna matična številka občana |
| PE: | poslovna enota |
| SQL: | Standard query language – standardni povpraševalni jezik |
| GUI: | Graphical User Interface – grafični uporabniški vmesnik |
| JSON: | JavaScript Object Notation – notacija objektov JavaScript |

# 1 UVOD

## 1.1 Predstavitev problema

V manjšem podjetju se zaloge vodijo s pomočjo programa MS Excel, kar povzroča različne težave. Ročno vnašanje in posodabljanje podatkov v Excelovih preglednicah je zamudno in pogosto prihaja do napak zaradi človeškega dejavnika, kot so napačni vnosi ali pozabljene posodobitve. Ker preglednico uporablja več zaposlenih, pogosto prihaja do težav s sinhronizacijo, saj dokument lahko ureja le ena oseba naenkrat. Z rastjo količine podatkov preglednica hitro postane nepregledna, upravljanje pa vse bolj zapleteno in manj učinkovito. Poleg tega Excel ne omogoča sprotnega sledenja stanju zalog, kar otežuje pravočasno zaznavanje pomanjkanja ali presežkov izdelkov.

## 1.2 Cilji naloge

Cilji diplomskega dela so:

- Cilj diplomskega dela je optimizirati proces vodenja zalog s pomočjo sodobnih informacijskih tehnologij, predvsem umetne inteligence (ang. Artificial Intelligence – AI). V ta namen raziščemo dve možnosti: implementacijo sistema WMS (ang. Warehouse Management System) in razvoj lastnega programskega orodja s pomočjo AI, ki bo prilagojeno specifičnim potrebam podjetja.
- Rezultat naloge je uspešna optimizacija ročnega vodenja zalog s programskim orodjem, ki ga ustvarimo po lastnih željah in potrebah. Ustvarimo programsko orodje, ki bo izboljšalo produktivnost – čas, z njim bomo zmanjšali možnost napak pri vnosih ter imeli boljšo preglednost in sledljivost nad zalogami.

## 1.3 Predpostavke in omejitve

Predpostavke in omejitve diplomskega dela so:

- Gre za mikropodjetje, ki ima omejene finančne in človeške vire.
- Ker je podjetje majhno, ne uporablja ERP-sistema in spletne trgovine ter trguje s približno 100 artikli, se omejimo na razvoj lastne programske rešitve namesto integracije sistema WMS.

## 1.4 Metode dela

V diplomskem delu uporabimo naslednje metode dela:

- kompilacijsko metodo,
- sintezno metodo,
- praktično metodo.

V teoretičnem delu uporabimo kompilacijsko metodo, s katero povzamemo ter združimo spoznanja in opise postopkov iz različnih virov. V praktičnem delu uporabimo metodo sinteze in kompilacijsko metodo ter opišemo postopke. Nato uporabimo praktično metodo za razvoj prototipa našega programa za vodenje zalog.

# 2 TEORETIČNE OSNOVE

## 2.1 Pomen logistike

### 2.1.1 Logistika

»Logistika je skupek med seboj povezanih procesov, ki služijo za premikanje surovin, polizdelkov, ostalega materiala in gotovih izdelkov od dobavitelja do podjetja, za premikanje znotraj podjetja in od podjetja do odjemalcev oziroma kupcev« (Rak, 2011, str. 3).

»Logistika se nanaša na podrobno organizacijo in izvajanje zapletenih operacij, zlasti upravljanje pretoka blaga, storitev in informacij od točke izvora do točke porabe za izpolnjevanje zahtev strank. Zajema široko paleto dejavnosti, vključno s prevozom, upravljanjem zalog, skladiščenjem, ravnanjem z materialom in upravljanjem dobavne verige« (https://cscmp.org, 2025).

Ključne komponente logistike:

- prevoz: pretok blaga z ene lokacije na drugo, ki lahko vključuje različne načine, kot so cesta, železnica, zrak in morje;
- upravljanje in vodenje zalog: nadzor količin zalog, ki zagotavlja, da zalog ni niti preveč niti premalo;
- skladiščenje: skladiščenje blaga, dokler ga ne potrebujemo, kar lahko vključuje specializirane prostore za različne vrste izdelkov;
- rokovanje z materialom: premikanje, zaščita, shranjevanje ter nadzor materialov in izdelkov v celotni proizvodnji, skladiščenju, distribuciji, porabi in odstranjevanju;
- upravljanje dobavne verige: širok nabor dejavnosti, potrebnih za načrtovanje, nadzor in izvajanje toka izdelka od materialov do proizvodnje in distribucije na najbolj ekonomičen način.

### 2.1.2 Poslovna logistika

»Poslovna logistika je poslovna funkcija s ciljem optimalne fizične nabave, distribucije in ravnanja z logističnimi objekti. Poslovna logistika je splošno priznana delna disciplina ekonomike poslovnih sistemov. V praksi razvitih podjetij je postala pomembna dejavnost in zato sestavni del strategije podjetja. Primarni razlog je v povečanih stroških nabave in distribucije s sočasnim upoštevanjem, da so lahko vrste ter načini nabave in distribucije blaga zelo različni, zlasti v zvezi z naraščajočo globalizacijo gospodarstva in vse večjo delitvijo dela« (Logožar, 2004, str. 31).

### 2.1.3 Notranja logistika

»Notranja logistika zajema vse aktivnosti v okvirju podjetja (pretok materiala od prevzema do odpreme gotovih proizvodov). Gre za planiranje, organiziranje in kontrolo vseh aktivnosti premikanja in skladiščenja znotraj delovne organizacije z namenom optimiziranja proizvodnih procesov« (Rak, 2011, str. 4).

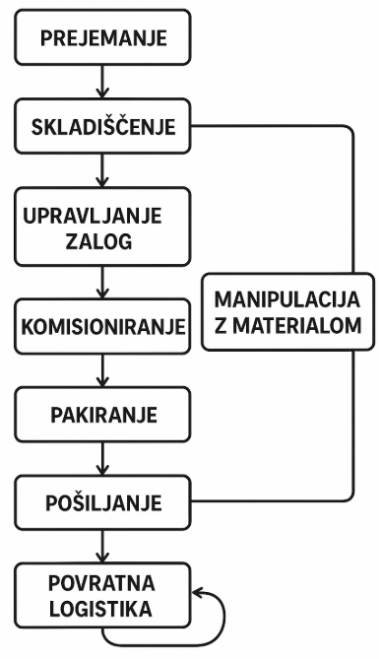
### 2.1.4 Skladiščna logistika

Skladiščna logistika se nanaša na načrtovanje, organizacijo in vodenje procesov, povezanih s skladiščenjem, manipulacijo in gibanjem blaga v skladišču. Zajema postopke in tehnologije, ki se uporabljajo za zagotavljanje učinkovitega in natančnega prejema, skladiščenja, pridobivanja in pošiljanja izdelkov. Cilj skladiščne logistike je optimizirati izkoriščenost prostora, zmanjšati stroške manipulacije in zagotoviti pravočasno dostavo blaga za potrebe kupcev.

Skladiščna logistika vključuje upravljanje fizičnega pretoka blaga v skladišču, vključno s sprejemom, skladiščenjem, komisioniranjem, pakiranjem in odpremo. Njen namen je optimizirati prostor, zmanjšati stroške in izboljšati raven storitev.

Ključne dejavnosti skladiščne logistike vključujejo:

- prejemanje: postopek sprejemanja dohodnega blaga, preverjanja njegove količine in kakovosti ter evidentiranja v sistemu zalog;
- skladiščenje: razporeditev in postavitev blaga v skladišču za čim večji izkoristek prostora in olajšanje iskanja;
- upravljanje zalog: sledenje in upravljanje ravni zalog, da se zagotovi točna evidenca zalog in prepreči izpad zalog ali prevelike zaloge;
- komisioniranje: postopek izbiranja in zbiranja artiklov iz skladišča za izpolnjevanje naročil strank;
- pakiranje: priprava predmetov za pošiljanje glede na način transporta in ustrezna identifikacija paketa;
- pošiljanje: organiziranje odhodnega prevoza blaga do strank ali drugih distribucijskih centrov;
- manipulacija z materialom: premikanje blaga v skladišču z uporabo opreme, kot so viličarji, tekoči trakovi in dvigalke za palete;
- povratna logistika: upravljanje vračila blaga, vključno s pregledom, obnavljanjem zalog ali odstranjevanjem.




*Slika 1: Diagram poteka delovnih nalog v skladišču*
(Lastni vir)

Skladiščna logistika igra ključno vlogo v celotni dobavni verigi, saj zagotavlja učinkovito skladiščenje in distribucijo blaga, skrajšanje dobavnih časov in izboljšanje zadovoljstva strank.

## 2.2 Zaloge

Pojem zaloga se navezuje na material, ki ga imamo skladiščenega v skladišču podjetja. Glavni namen zalog je zagotavljanje nemotenega poteka prodajnega oz. proizvodnega toka.

»Zaloga se razume kot količina blaga, odložena (uskladiščena) na določenem mestu (skladišču) v podjetju« (Ljubič, 2006, str. 347). Obseg zalog določata skladiščni prostor in njegova opremljenost. Pomembno je upoštevati tudi število zaposlenih v skladišču.

Zaloge niso statična količina, niso in ne morejo biti vedno enake. Potrebne zaloge so enkrat večje, drugič manjše. Spreminjajo se številni dejavniki, ki vplivajo na višino zalog in izhajajo iz možnosti proizvodnega procesa in položaja na nabavnem in prodajnem tržišču (Logožar, 2004).

Po Kaltnekarju (Kaltnekar, 1993, str. 273–274) so najpomembnejši dejavniki obvladovanja zalog:

- likvidnost podjetja in s tem v zvezi možnosti za izbiro razmer na tržišču;
- odvisnost od tržišča, predvsem od razmerja med ponudbo in povpraševanjem;
- proizvodna (predelovalna) stroka oziroma njena intenzivnost v porabi materiala;
- krajevne komunikacijske in transportne možnosti;
- individualne možnosti prodajnega in proizvodnega programa podjetja;
- cilji splošne poslovne politike podjetja.

»Zaloga je vsak neizrabljen vir, ki čaka na bodočo uporabo. Če podjetje kupuje ali izdeluje sestavne dele in proizvode, je soočeno z odločanjem o zalogah. Zaloge nastajajo, kadar se na vhodih ali izhodih materiali ne porabijo takoj, ko so na razpolago. Količina zalog predstavlja podjetju veliko finančno breme in zmanjšuje učinkovitost podjetja, zato je management (kontrola) zalog ena najpomembnejših logističnih dejavnosti v proizvodnem podjetju« (Čižman, 2001, str. 53).

## 2.3 Upravljanje zalog

Upravljanje zalog je sistematičen proces nadzora in nadzora naročanja, skladiščenja in uporabe materialov in izdelkov znotraj organizacije. Zagotavlja, da je prava količina zalog na voljo ob pravem času in na pravem mestu, da zadosti povpraševanju kupcev, hkrati pa zmanjša stroške, povezane z zadrževanjem odvečne zaloge. Učinkovito upravljanje zalog usklajuje potrebo po vzdrževanju zadostnih ravni zalog, da se izognemo izpadu zalog, in potrebo po izogibanju prevelikim zalogam, kar lahko privede do povečanih stroškov hrambe in zastarelosti.

Upravljanje zalog vključuje več ključnih dejavnosti:

- Napovedovanje povpraševanja: napovedovanje prihodnjega povpraševanja strank za določitev ustreznih ravni zalog.
- Upravljanje naročil: oddaja naročil pri dobaviteljih ter upravljanje časovnega razporeda in količine naročil za dopolnitev zalog.
- Nadzor zalog: spremljanje ravni zalog v realnem času za zagotovitev vzdrževanja optimalnih ravni zalog.
- Skladiščenje in skladiščenje: učinkovito organiziranje in shranjevanje zalog za lažji dostop in zmanjšanje stroškov ravnanja.
- Vrednotenje zalog: ocenjevanje vrednosti zalog za namene finančnega poročanja in odločanja.
- Optimizacija zalog: uporaba tehnik, kot so Just-In-Time (JIT), ekonomična količina naročila (EOQ) in analiza ABC za zmanjšanje stroškov in povečanje učinkovitosti.
- Revizija zalog (inventure): izvajanje rednih fizičnih štetij ali cikličnih štetij za zagotovitev točnosti med zabeleženimi in dejanskimi količinami zalog.

Končni cilj upravljanja zalog je zagotoviti, da ima organizacija prave izdelke v pravih količinah ob pravem času, hkrati pa zmanjšati stroške in povečati donosnost.

Na podlagi Logožarjevih (Logožar, 2004) raziskav je pomembna naloga podjetij, da ugotovijo, kakšen je optimalen obseg zalog, in da se ga poizkuša tudi doseči. Zaloge se zaradi nihanja proizvodnje, prodaje in zaradi drugih nepredvidenih dejavnikov spreminjajo, nase pa vežejo velika finančna sredstva in povečujejo stroške poslovanja, zato je relevantno, da se določi optimalno gibanje zalog. To se lahko stori z opazovanjem in ugotavljanjem količine teh zalog, ki so lahko:

- varnostne,
- operacijske,
- signalne,
- maksimalne.

Učinkovito upravljanje zalog je pogojeno s poznavanjem in razumevanjem osnovnih značilnosti sistemov zalog in njihovega delovanja. Organizacije lahko uporabljajo različne vrste sistemov zalog, kar je odvisno od njihove velikosti in dejavnosti. (Čižman, 2001).

Z vidika obnavljanja zalog razlikujemo:

- zaloge za eno obdobje,
- zaloge za več obdobij.

### 2.3.1 Sistem zalog za eno obdobje

Značilnost sistema zalog za eno obdobje je, da so predmeti uskladiščeni samo enkrat, brez ponovnega dopolnjevanja porabljene zaloge. Določanje optimalne količine naročanja se lahko oceni z uporabo mejne analize. Ta postopek temelji na konceptu pričakovane vrednosti. Mejna analiza priporoča povečanje zaloge za enoto (predmet), dokler je pričakovani dobiček večji ali enak pričakovanim stroškom. To pomeni, da se obseg naročila lahko povečuje, dokler so pričakovani stroški zaradi presežka zalog (nadzaloženosti) manjši od pričakovanih stroškov zaradi premajhnih zalog (podzaloženosti).

### 2.3.2 Sistem zalog za več obdobij

V praksi je najbolj razširjen sistem zalog za več obdobij, ki se pogosto uporablja v kosovni in serijski proizvodnji ter v trgovini (prodaja na debelo in drobno), kjer imamo opravka s standardiziranimi proizvodi, ki so namenjeni prodaji v številnih časovnih obdobjih.

Sistem zalog za več obdobij opredelimo glede na izvor zahtev po materialu (predmetih), ki je lahko zunaj ali znotraj organizacije, kot:

- Sistem vezanih zalog: nanašajo se neposredno na zahteve po materialu znotraj podjetja (kosovnica) in so odvisne od vrste izdelka. Večina predmetov, ki so v vezanih zalogah, se uporablja za izdelavo končnih izdelkov.
- Sistem nevezanih zalog: nevezane zaloge (imenovane tudi neodvisne zaloge) so zaloge artiklov, katerih povpraševanje ni vezano na povpraševanje po drugih artiklih, temveč izvira iz zunanjih zahtev trga oziroma kupcev. Klasičen primer so zaloge končnih izdelkov v trgovini: trgovsko podjetje vzdržuje določeno zalogo izdelkov zato, da lahko izpolni neposredne zahteve kupcev, pri čemer teh zahtev ni mogoče natančno izračunati, temveč jih je treba napovedovati na podlagi pretekle prodaje, tržnih trendov ipd. Nevezane zaloge torej označujemo tudi kot zaloge z neodvisnim povpraševanjem, saj zanje ni znana vnaprejšnja poraba znotraj podjetja – povpraševanje določajo zunanji odjemalci.

### 2.3.3 Metoda FIFO v skladiščnem poslovanju

FIFO (First-In, First-Out) je temeljno načelo upravljanja zalog, po katerem se prvo prispele količine blaga tudi prve izdajo iz skladišča. To pomeni, da vedno najprej uporabimo ali odpošljemo najstarejše zaloge. V praksi to zahteva, da so artikli v skladišču razporejeni in obravnavani tako, da se zagotovi odpošiljanje v enakem zaporedju, kot so bili prejeti. Na kratko: prvi noter – prvi ven.

Akademsko gledano, je FIFO v skladiščnem kontekstu način obvladovanja materialnega toka, ki določa zaporedje gibanja blaga skozi skladišče. FIFO zagotavlja, da se tok blaga odvija krožno in urejeno – starejše zaloge se rotirajo iz skladišča, medtem ko novejše ostajajo zadaj, dokler ne pridejo na vrsto. S tem se vzpostavi popoln obrat zalog, saj se prepreči kopičenje starih enot.

### 2.3.4 Načini upravljanja zalog

Za upravljanje z zalogami lahko uporabimo različne tehnike. Od tradicionalnih ročnih tehnik do naprednih, tehnološko usmerjenih rešitev. Tehnika, ki jo izberemo, je običajno odvisna od obsežnosti podjetja, kompleksnosti delovanja podjetja oz. izzivov, s katerimi se srečujemo. Z avtomatizacijo procesa upravljanja zalog lahko dosežemo, da procesi tečejo hitreje, enostavnejše in z manj napakami.

2.3.4.1 Netehnološke metode upravljanja zalog:

Netehnološke metode temeljijo na ročnih procesih in fizičnem sledenju:

- ročno štetje (fizične revizije) – periodični pregledi osebja za štetje zalog,
- svinčnik-in-papir sledenje – beleženje ravni zalog v dnevnikih ali preglednicah,
- kartice zalog in knjige zalog – fizične kartice ali knjige, v katere beležimo in spremljamo gibanje zalog materiala.

2.3.4.2 Tehnološko usmerjene metode upravljanja zalog

Za uporabo tehnološko usmerjenih metod uporabljamo programsko opremo, avtomatizacijo in digitalna orodja:

- Programi za upravljanje z zalogami – najbolj razširjeni so: Zoho Inventory, TradeGecko in inFlow.
- Sistemi ERP (Enterprise Resource Planning) – SAP, Oracle, NetSuite za integriran nadzor zalog.
- Cloud-Based sistemi za upravljanje zalog – sledenje zalogam v realnem času na več lokacijah.
- Skeniranje črtnih kod – hitro skeniranje predmetov za sledenje in posodobitve.
- RFID (ang. Radio-Frequency Identification) – brezžično sledenje označenih zalog z uporabo oznak RFID.

2.3.4.3 Hibridne metode upravljanja zalog

Hibridne metode združujejo ročne in tehnološke metode:
- Preglednice (Excel/Google Sheets) – digitalni, vendar še vedno ročni vnos podatkov.
- Mobilne aplikacije za zaloge – ročne naprave za skeniranje in posodabljanje zalog.

## 2.4 Skladiščenje

Dobrega skladiščnega poslovanja si ne moremo predstavljati brez dobre organiziranosti, pomoči informacijske tehnologije za boljšo preglednost, urejene dokumentacije v zvezi s prejetim in izdanim blagom, vhodne kontrole količin in kakovosti na prevzemu, lociranja materiala izdaje in reklamacije materiala. Delovni procesi morajo bit v skladišču opravljeni natančno, hitro, sistematično in gospodarno. »Osnovna naloga skladiščne službe je spremljanje, varovanje in izdajanje surovin, polproizvodov, proizvodov in drugega blaga. Skladiščenje torej služi premagovanju časovnih neenakomernosti med raznimi dejavnostmi v podjetju. Zaradi čim bolj ekonomičnega poslovanja morajo imeti skladišča ustrezno lokacijo, poslopja pa morajo biti zgrajena tako, da čim učinkoviteje služijo svojemu namenu« (Logožar, 2004, str. 79).

Poleg navedenih nalog v skladiščih poteka tudi:
- urejanje dokumentacije v zvezi s prejetim in izdanim blagom;
- namestitev blaga v skladišča;
- nabiranje blaga (ang. picking);
- pakiranje, če je potrebno.

»Skladiščenje služi premagovanju časovnih neenakomernosti med raznimi dejavnostmi v podjetju« (Požar, 1985, str. 201). Zaradi čim bolj ekonomičnega poslovanja morajo imeti skladišča ustrezno lokacijo, poslopja pa morajo biti zgrajena tako, da čim učinkoviteje služijo svojemu namenu.

### 2.4.1 Skladiščna dokumentacija

Vsako spremembo zaloge v skladišču je treba evidentirati z dokumentom. Opišemo dva dokumenta, ki se uporabljata v našem skladišču; to sta prevzemnica in dobavnica.

2.4.1.1 Prevzemnica

S prevzemnico, ki jo prejme skladiščnik od dobavitelja, skladišče potrjuje dejanski

prevzem materiala v skladišče. V prevzemnico se vpišejo podatki: naziv materiala, šifra ali koda materiala, količina in podatki o dobavitelju. Če nastane razlika med dejansko prevzeto količino in podatki v dobaviteljevi dokumentaciji, bo nabavna služba sprožila reklamacijski postopek.

2.4.1.2 Dobavnica

Dobavnica se uporablja pri prodaji končnih izdelkov (tudi pri prodaji materiala in odpadkov). Vsebuje naslednje podatke: naslov kupca, datum dobave, naziv in kodo materiala ali blaga, enoto mere. Sodobno skladiščno evidenco omogoča računalniško spremljanje prevzema in izdaje materiala, ki zagotavlja tudi takojšnjo ugotovitev stanja zaloge in s tem pravočasno odločanje o količini novega naročila.

## 2.5 Umetna inteligenca – chatgpt

ChatGPT je napreden jezikovni model AI, ki ga je razvil OpenAI in temelji na zmogljivi Generative Pre-trained Transformer (GPT) arhitekturi. Zgrajen je na vrsti jezikovnih modelov velikega obsega, med katerimi je najpomembnejša družina GPT, ki se je začela z generacijo GPT-1 in so ji sledile GPT-2, GPT-3, GPT-3.5, GPT-4 in naprej. ChatGPT izkorišča globoke arhitekture nevronskih mrež, usposobljene z obsežnimi količinami besedilnih podatkov iz različnih internetnih virov, knjig, člankov in spletnih mest, kar modelu omogoča ustvarjanje besedila.

Zavedati se je treba, da so nevronske mreže pri enostavnejših zahtevah bolj natančne kot pri bolj kompleksnih. V našem primeru je zato pomembno, da tudi sami razumemo osnove programskega jezika in da znamo čim bolj natančno opisati zahteve, katere hočemo, da nam jih AI izpiše.

### 2.5.1 Zmogljivosti in uporaba ChatGPT

ChatGPT je mogoče uporabiti na več področjih:

- Programiranje in razvoj: pisanje, odpravljanje napak in pregledovanje izrezkov kode v različnih programskih jezikih.
- Tehnična podpora in odpravljanje težav: zagotavljanje navodil po korakih in rešitev za tehnične težave.
- Pomoč pri pogovoru: odgovarjanje na splošna vprašanja, sodelovanje v priložnostnih pogovorih in razumevanje konteksta.
- Ustvarjalno pisanje: pisanje esejev, poezije, ustvarjalnih zgodb, scenarijev, dialogov in ustvarjanje inovativne vsebine.
- Pomoč pri izobraževanju: razlaga konceptov, zagotavljanje povzetkov, preverjanje jezikov in pomoč študentom pri domačih nalogah.
- Generiranje vsebine: ustvarjanje objav na družabnih omrežjih, marketinških vsebin, člankov, e-poštnih predlogov itd.

- Avtomatizacija podpore strankam: pogovorni roboti za učinkovito in hitro pomoč strankam.

## 2.6 Osnovno razumevanje pojmov za izdelavo programske opreme

Programsko orodje (pogosto imenovano tudi »programska oprema za podporo razvoju«, angl. software tool ali software utility) je računalniški program ali skupek programov, ki je namenjen podpori, pomoči ali avtomatizaciji določenih nalog pri delu z računalnikom. Programska orodja omogočajo uporabnikom in razvijalcem učinkovitejše in bolj organizirano izvajanje določenih postopkov, analiz, razvoja ali vzdrževanja računalniških sistemov in podatkov.

Programsko orodje je lahko napisano v kateremkoli programskem jeziku, saj izbira jezika običajno temelji na:

- vrsti naloge, ki jo orodje opravlja,
- ciljni operacijskosistemski platformi (npr. Windows, Linux, splet kot porazdeljen hipertekstualni sistem medmrežja, Android za mobilne naprave),
- zahtevani hitrosti razvoja in učinkovitosti,
- znanju razvijalcev.

### 2.6.1 Python programski jezik

Python je poseben jezik, s katerim se lahko »pogovarjamo« z računalnikom. Je kot navodila ali recept, ki računalniku pove, kaj naj naredi. Če na primer želiš, da računalnik nekaj izračuna, izriše, poišče po spletu ali igra igro, mu lahko navodila napišeš v jeziku Python. Z njim lahko programirajo tako začetniki kot tudi izkušeni programerji.

Python je zelo priljubljen, ker je preprost za učenje. Zelo veliko ljudi po svetu se najprej nauči prav jezik Python, ker so njegova navodila podobna tistim, ki jih uporabljamo v vsakdanjem življenju.

»Python je interpretiran, interaktiven, objektno usmerjen programski jezik. Vključuje module, izjeme, dinamično tipkanje, zelo visokonivojske dinamične podatkovne tipe in razrede« (Lutz, July 2013, str. 5). Interpretiran pomeni, da se koda izvaja vrstico za vrstico, kar omogoča hiter razvoj aplikacij in odpravljanje napak v kodi.

Python je konec osemdesetih let prejšnjega stoletja ustvaril Guido van Rossum in je bil javno izdan leta 1991. Od takrat se je razvil v enega od najbolj razširjenih programskih jezikov na svetu.

### 2.6.2 Izdelava specifikacije (popis optimiziranih procesov – navodila za izdelavo programskega orodja)

Pisanje jasnih zahtev in seznamov procesov je ključnega pomena za izdelavo programske opreme, ki dejansko rešuje dejanski problem in izpolnjuje potrebe uporabnikov. Zagotavlja načrt za razvoj, pomaga preprečiti zmedo ter zagotavlja optimizirana in učinkovita programska orodja. Zahteve morajo biti vedno organizirane, natančne in lahko razumljive.

Ustvarjanje specifikacije pomeni naštevanje vsega, kar mora programska oprema početi in kako naj bi se obnašala. To vključuje, kar si uporabnik želi (»funkcionalne zahteve«), kot tudi to, kako naj bi sistem deloval (»nefunkcionalne zahteve«). Ko se napiše seznam optimiziranih procesov ali navodila za programska orodja, se opisujejo naloge, ki jih bo program avtomatiziral ali pri katerih bo pomagal, in natančno, kako naj bi te naloge delovale.

Pisanje specifikacije programa je pomembno za:

- Jasnost: zahteve zagotavljajo, da vsi (razvijalci, uporabniki, deležniki) razumejo, kaj naj bi programska oprema počela.
- Načrtovanje: omogočajo vam oceno časa, stroškov in virov, potrebnih za razvoj.
- Kakovost: dobre zahteve zmanjšujejo nesporazume, napake in »širjenje obsega« programa.
- Testiranje: jasne zahteve vam omogočajo, da preverite, ali končna programska oprema dejansko izpolnjuje obljubljeno.
- Optimizacijo: z naštevanjem in izpopolnjevanjem procesov zagotovite, da bo programsko orodje izboljšalo obstoječe delovne procese in prihranilo čas ali vire.

Za kakovostno napisano specifikacijo moramo v prvi vrsti dobro razumeti in poznati delovne procese ter jih proučiti. Pomembno je, da potekajo pogovori z uporabniki, ki trenutno uporabljajo še neoptimiziran način dela, da dobimo povratno informacijo, s katerimi težavami se soočajo. Postaviti je treba prioritete. Katere zahteve so nujne, da jih programska oprema izvaja. Zapisati je treba vsak korak, ki želimo, da ga programska oprema izvede. Za vsak proces je treba spisati jasna navodila.

### 2.6.3 Arhitektura programskega orodja

Dobro zasnovana arhitektura programske opreme je ključnega pomena za uspeh sistema. »Arhitektura programske opreme se nanaša na visokonivojske strukture programskega sistema, disciplino ustvarjanja takšnih struktur in dokumentacijo teh struktur. Zajema niz pomembnih odločitev o organizaciji programskega sistema, vključno z izbiro strukturnih elementov in njihovih vmesnikov, vedenjem teh elementov in sestavo sistema« (Bass, Clements, & Kazman, 2013, str. 3).

Vključujejo komponente (module, razrede, funkcije), povezovalnike (vmesnike, protokole) in konfiguracije (kako so komponente razporejene in medsebojno delujejo). Arhitektura programa je temelj in struktura programske opreme. Pomembna je, ker olajša gradnjo, rast in vzdrževanje programa. Za pisanje arhitekture načrtujemo komponente, definiramo njihove vloge, dokumentiramo njihove odnose in izbiramo prava orodja za delo.

Arhitektura programa se nanaša na celotno strukturo in organizacijo programske aplikacije. Je kot načrt za to, kako bo program zgrajen, kako njegove komponente medsebojno delujejo in kako bodo podatki tekli skozi sistem. Dobra arhitektura določa jasne meje med različnimi deli programa (na primer ločuje uporabniški vmesnik od upravljanja podatkov), zaradi česar je lažje za razumeti, vzdrževati in razširjati.

Arhitektura programa je pomembna za:

- Vzdrževanje: dobro strukturirana arhitektura olajša posodabljanje ali popravljanje kode. En del lahko spremenite, ne da bi pri tem pokvarili druge.
- Prilagodljivost: dobra arhitektura olajša dodajanje novih funkcij ali podporo več uporabnikom, ko vaša aplikacija raste.
- Ponovno uporabnost: jasna ločitev skrbi pomeni, da lahko kodo ponovno uporabite v drugih projektih.
- Zanesljivost: z organizirano kodo se napake manj verjetno širijo po sistemu, zaradi česar je programska oprema stabilnejša.

Arhitekturo programa napišemo tako, da:

- Določimo glavne komponente programa: odločimo se, katere dele bo imel naš program (uporabniški vmesnik – UI), upravljanje podatkov, poslovna logika, shranjevanje podatkov).
- Opišemo, kako komponente medsebojno delujejo: opišemo, kako se podatki premikajo med uporabniškim vmesnikom, logiko in shranjevanjem. Diagrami (kot so diagrami poteka ali blokovni diagrami) so koristni.
- Izberemo tehnologije: izberemo knjižnice ali ogrodja (Tkinter za grafični uporabniški vmesnik, JSON za prenos in shranjevanje podatkov).
- Dokumentiramo vse: napišemo komentarje ali dokument, ki pojasnjuje strukturo, ključne razrede in njihove odgovornosti.

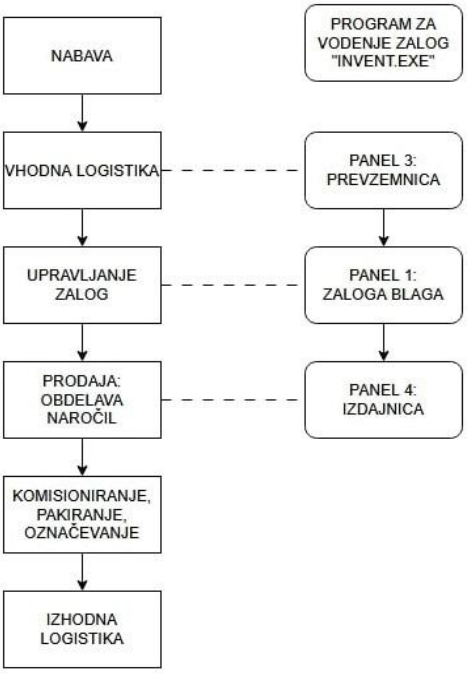
# 3 OBSTOJEČE STANJE: TRENUTNO STANJE VODENJA ZALOG V PODJETJU X

Trenutno podjetje upravlja s približno 100 različnimi izdelki, sledenje zalogam pa se izvaja ročno prek preglednic Excel – vsak izdelek se vodi na ločenem delovnem listu. Ta ročni sistem, ki temelji na preglednicah, znatno poveča tveganje za človeške napake, upočasni posodabljanje zalog in otežuje procese upravljanja zalog. Poleg tega prodajna ekipa samostojno ustvarja tako prehodno kot odhodno dokumentacijo o zalogah, kar še dodatno poveča delovno obremenitev in možnost napak.

Podjetje ne uporablja sistema za načrtovanje virov podjetja (ERP), temveč se zanaša izključno na osnovni računovodski program in standardne aplikacije Office (Excel, Word, Outlook).

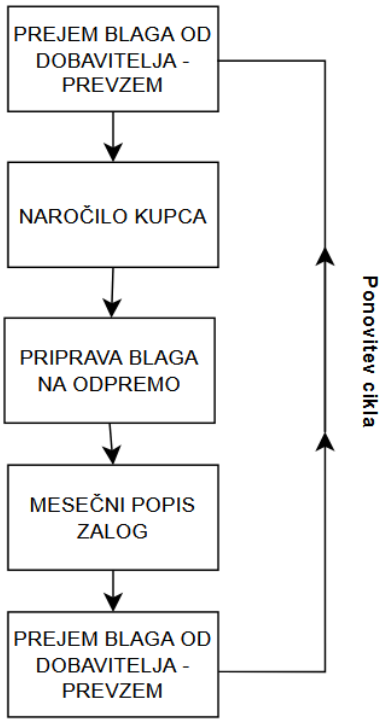
## 3.1 Logistični procesi v podjetju

Trenutno vodenje zalog poteka s pomočjo Excelovih tabel, pri čemer je vsak artikel voden na svojem listu. Poleg osnovnih postopkov, ki vključujejo prejem blaga, ročno knjiženje zaloge in prodajo artiklov, podjetje izvaja še ročno mesečno popisovanje zalog vsak konec meseca. Zaposleni opravijo fizični popis blaga na vseh lokacijah, kar zahteva veliko časa. Naročila kupcev se ročno preverjajo prek Outlooka, nato se v Excelu ročno preveri zaloga artiklov in potrdi razpoložljivost blaga, preden se naročilo potrdi in odpremi. Prodajnik samostojno ustvarja dokumente za izdajo (dobavnico, račun) in dokumente za prevzem blaga, kar dodatno obremenjuje prodajno osebje in povečuje možnost napak zaradi ročnega pretipkavanja podatkov iz različnih virov.




*Slika 2: Diagram poteka delovnih nalog v podjetju X*
(Lastni vir)

Vsi postopki potekajo ročno (brez avtomatizacije). Podatki so razpršeni na različnih izvornih dokumentih, pogosto prihaja do pretipkavanja iz elektronske pošte, vhodnih dokumentov dobaviteljev in orodja za obdelavo razpredelnic MS Excela. Obstaja velika možnost napak in zamud zaradi ročnega dela in preverjanja podatkov.




*Slika 3: Diagram poteka delovnih nalog v skladišču X*
(Lastni vir)

### 3.1.1 Prejem blaga od dobavitelja – prevzem

- Blago prispe v podjetje.
- Zaposleni ročno preveri dostavljeno blago in ga fizično vnese v skladišče.
- Skladiščnik podatke o prejetem blagu preveri s prevzemnico, nato jih ročno vnese v ustrezen Excelov list (za vsak artikel poseben list).
- Ročno se posodobi stanje zaloge v Excelovi preglednici.

### 3.1.2 Naročilo kupca

- Kupec pošlje naročilo (prek e-pošte v Outlooku).
- Kupec je pred tem od prodajnika že prejel ponudbo iskanega artikla.
- Prodajnik odpre Outlook in ročno prebere naročilo iz elektronske pošte.
- Prodajnik odpre preglednico Excel in na ustreznem listu poišče artikel ter preveri zalogo.
- Če je artikel na zalogi, prodajnik ročno potrdi razpoložljivost kupcu (ponovno preko e-pošte).
- Naročilo se potrdi.

### 3.1.3 Priprava blaga za odpremo (prodaja)

- Prodajnik ali skladiščnik fizično pripravi blago za odpremo.
- Prodajnik pripravi dobavnico in račun, podatke pretipka iz drugih virov.
- Blago se odpremi kupcu.
- V preglednici v Excelu se ročno evidentira, to je zmanjša zaloga na ustreznem listu za prodani artikel.

### 3.1.4 Mesečni popis zalog

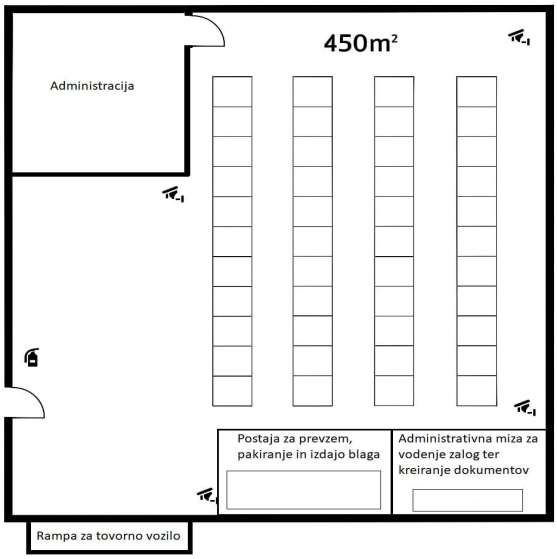
- Ob koncu meseca zaposleni podjetja fizično popišejo zalogo na vseh lokacijah.
- Popisane količine primerjajo z evidenco v Excelovi preglednici.
- Opravijo ročne popravke v Excelovi preglednici glede na dejansko stanje (izvede se vnos korekcij zaloge).

### 3.1.5 Prejem blaga od dobavitelja – prevzem (ponovitev cikla)

- Po potrebi podjetje naroči blago pri dobavitelju.
- Ob prejemu se postopek ponovi kot v koraku 1.

## 3.2 Skladiščni prostor

Skladišče podjetja X je organizirano na preprost in praktičen način. Ima skupno površino 450 m², strukturirano tako, da optimizira izrabo prostora in zagotavlja enostaven dostop do blaga.




*Slika 4: Tloris skladišča podjetja X*
(Lastni vir)

Skladišče je opremljeno z regalnim sistemom, ki ga sestavlja 100 različnih skladiščnih mest (od Loc1 do Loc100). Vsako mesto je jasno označeno za enostavno prepoznavanje.

Loc1, Loc2, Loc3 ... Loc100

*Slika 5: Način označevanja lokacij v podjetju X*
(Lastni vir)

Police v regalnem sistemu so izdelane iz trpežnih kovinskih pokončnih nosilcev s stabilnimi lesenimi policami. Vsaka polica je nastavljiva, kar omogoča prilagodljiv navpični razmik glede na velikost shranjenega blaga. Območje skladišča je pravokotne oblike, zato zagotavlja enostavne dostopne poti in prehode med vrstami polic. Zadostna širina prehoda omogoča zaposlenim enostaven in varen dostop za skladiščenje in prevzem artiklov ter nemoteno delovanje skladiščnih vozičkov ali paletnih vozičkov.

*Slika 6: Regalni sistem v skladišču X*
(Vir: Sprzedajemy.pl, 2025)

Namenski prostor v bližini vhoda je namenjen učinkovitemu ravnanju z dohodnim in odhodnim blagom. Vključuje neposreden dostop za dostavna vozila. Postaja za prevzem je opremljena z embalažnim materialom, podajalniki lepilnega traku, etiketami, tiskalnikom in mizo za varno pripravo blaga za odpremo.
Majhen pisarniški kotiček ali pisalna miza z računalnikom za upravljanje zalog, tiskanje dokumentacije in usklajevanje naročil, dopolnjen z dostopno električno in omrežno povezljivostjo.

# 4 PRAKTIČNI DEL: IZDELAVA LASTNEGA PROGRAMSKEGA ORODJA ZA VODENJE ZALOG S POMOČJO UMETNE INTELIGENCE

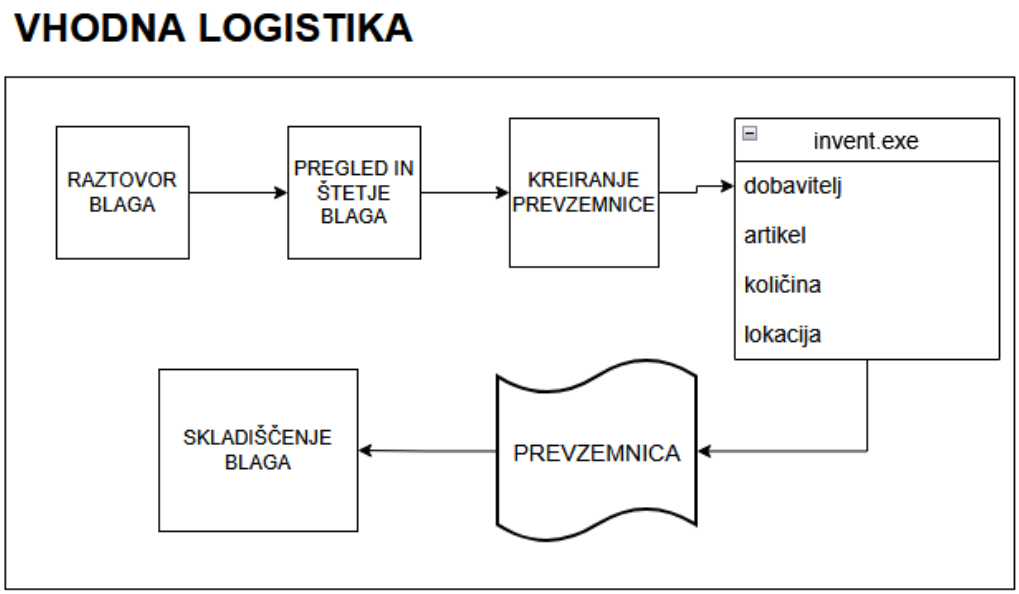
V tem poglavju bomo raziskali postopek razvoja programskega orodja za upravljanje zalog od kreiranja specifikacije programa, arhitekture programa do končnega izdelka orodja. Prikazali bomo, kako lahko optimiziramo skladiščne procese ročnega upravljanja zalog v podjetju (pero-in-papir, preglednice) ter kako s poznavanjem in opisom logističnih procesnih postopkov načrtujemo in izdelamo popolnoma delujočo programsko rešitev. S tem programskim orodjem bomo pridobili pregleden in organiziran nadzor zalog, prihranek časa in manj napak, pregled v realnem času, upravljanje dokumentov, boljši izkoristek prostora, uporabniku prijazen vmesnik in shranjevanje podatkov. Ta aplikacija poenostavlja skladiščne operacije, izboljšuje natančnost in zagotavlja jasen pregled nad zalogami, zaradi česar so vsakodnevna opravila hitrejša in zanesljivejša.

Kreirana programska oprema invent.py prikazuje, kako je mogoče programski jezik Python in njegovo knjižnico GUI tkinter za pripravo grafičnega vmesnika združiti v zmogljivo in uporabniku prijazno orodje za nadzor pretoka zalog. Ta programska oprema upravlja skladišče z do 100 skladiščnimi lokacijami, vsako s fiksno kapaciteto, kar zagotavlja pregled nad količinami izdelkov in njihovimi natančnimi lokacijami v realnem času. Uporabniki lahko hitro dodajajo ali odstranjujejo zaloge, dodeljujejo izdelke določenim skladiščnim mestom ter ustvarjajo ali izvažajo dokumente za prejemke (prejemno blago) in dobave (odhodno blago).

Ključne značilnosti programske opreme invent.py vključujejo:

- Grafični uporabniški vmesnik (ang. Graphical User Interface – GUI): jasen, preprost vmesnik, zgrajen s tkinterjem, z več okni za seznam zalog, prejemke in dobavnice, upravljanje dokumentov in prikazom vsebine.
- Upravljanje lokacij: samodejna dodelitev izdelkov po razpoložljivih skladiščnih lokacijah, sledenje količin na lokacijo in zagotavljanje, da nobena lokacija ne preseže nastavljene zmogljivosti.
- Trajno shranjevanje podatkov: vsi podatki o izdelkih, lokacijah in dokumentih so shranjeni v datotekah JSON, kar zagotavlja, da je zaloga posodobljena med sejami brez potrebe po sistemu za upravljanje baz podatkov.
- Celovito upravljanje dokumentov: možnost ustvarjanja, pregleda in izvoza potrdil o prejemu in dobavnic, vključno s popolno razčlenitvijo, kateri izdelki so bili nameščeni na katerih lokacijah ali prevzeti s katerih lokacij.
- Preverjanje in obravnavanje napak: program preprečuje pogoste napake, kot sta prenapolnjenost lokacije ali prodaja več izdelkov, kot jih je na zalogi, ter zagotavlja natančnost podatkov in celovitost zalog.

## 4.1 Diagrami optimiziranih procesov v skladišču




*Slika 7: Diagram optimiziranih delovnih nalog pri vhodni logistiki v podjetju X*
(Lastni vir)

Na sliki je prikazan proces vhodne logistike, torej postopka sprejema blaga v skladišče. Sledi opis posameznih korakov in njihovega zaporedja:

- RAZTOVOR BLAGA:

Blago, ki prispe v skladišče, se najprej raztovori z dostavnega vozila.

- PREGLED IN ŠTETJE BLAGA:

Po raztovoru sledi vizualni pregled in štetje artiklov za preverjanje količine in stanja.

- KREIRANJE PREVZEMNICE:

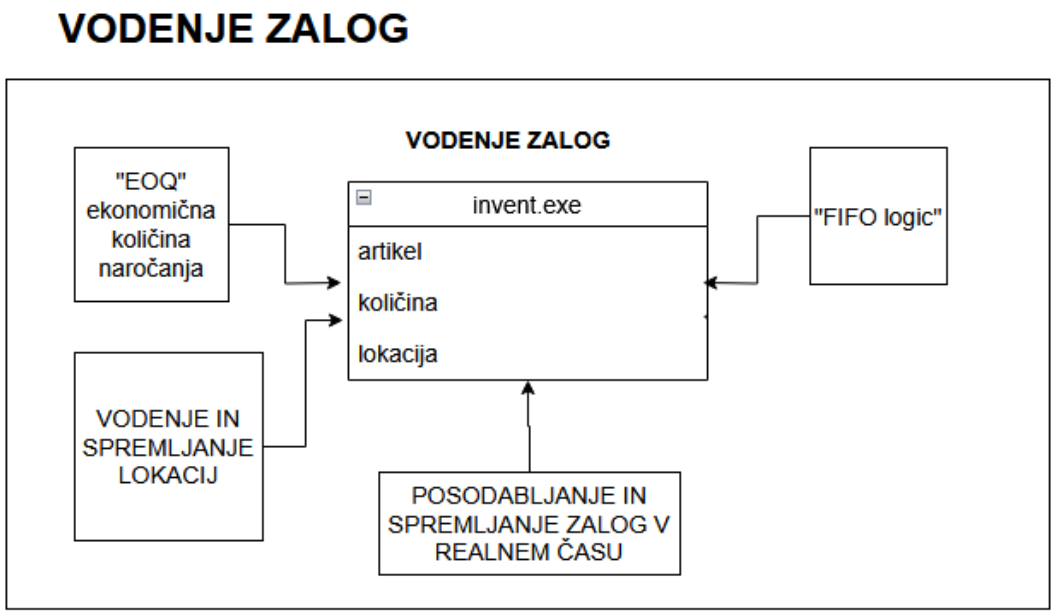
Na podlagi pregleda se v informacijskem sistemu (programu invent.exe) ustvari prevzemnica. Podatki, ki se vnesejo v programsko orodje invent, so: dobavitelj, artikel, količina in lokacija.

- PREVZEMNICA:

Ustvarjena prevzemnica služi kot dokument o prejetem blagu.

- SKLADIŠČENJE BLAGA:

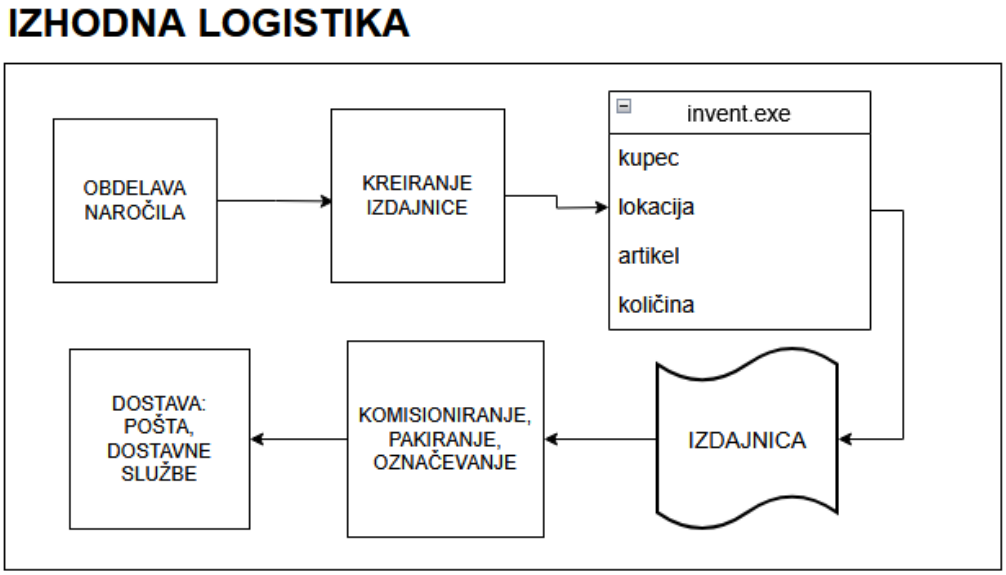
Blago se nato fizično premesti na ustrezne lokacije v skladišču v skladu z informacijami na prevzemnici.




*Slika 8: Diagram optimiziranega procesa vodenja zalog v podjetju X*
(Lastni vir)

Na sliki 8 je prikazan proces vodenja zalog, ki vključuje spremljanje, posodabljanje in optimizacijo zalog v skladišču s pomočjo programske opreme Invent. Proces vodenja zalog vključuje:

- natančno beleženje artiklov, količin in skladiščnih lokacij,
- optimizacijo naročil s pomočjo EOQ (izračun optimalne količine naročanja trenutno ni del programske opreme Invent),
- samodejno in sprotno posodabljanje zalog (samodejno posodabljaje ob vsakem premiku, prevzemu ali izdaji artiklov),
- sledenje zalogam po pravilih FIFO (ang. First In, First Out – najprej se porabijo starejše zaloge).




*Slika 9: Diagram optimiziranih delovnih nalog pri izhodni logistiki v podjetju X*
(Lastni vir)

Na sliki 9 je prikazan proces izhodne logistike, torej postopek izdaje blaga iz skladišča. V nadaljevanju so opisani posamezni koraki in njihovo zaporedje:

- OBDELAVA NAROČILA

Obdelava naročila je postopek ali potek dela, ki se zgodi po tem, ko stranka (kupec) odda naročilo. Začne se s potrditvijo, da so izdelki na zalogi, sledijo datum dostave, način transporta in lokacija dostave kupcu.

- KREIRANJE IZDAJNICE

Podatki, ki se vnesejo v program Invent, so: kupec, lokacija, artikel in količina.

- IZDAJNICA

Izdajnica je dokument, priložen pošiljki blaga in poslan stranki. Na njej sta navedena opis in količina blaga, ki je v pošiljki.

- KOMISIONIRANJE, PAKIRANJE, OZNAČEVANJE

S komisioniranjem pripravimo ustrezen izdelek in s pakiranjem pripravimo izdelek za ustrezen transport. Označevanje vključuje prikaz bistvenih podrobnosti, kot so ime izdelka, sestavine, naslov kupca, skladnost s predpisi in nevarnosti med transportom itd.

- DOSTAVA

Organizacija prevoza do želene lokacije kupca in priprava izdelka za ustrezen transport.

## 4.2 Izdelava specifikacije (popis optimiziranih procesov – navodila za izdelavo programskega orodja) in arhitektura programa

### 4.2.1 Izdelava specifikacije (popis optimiziranih procesov – navodila za izdelavo programskega orodja)

Pri izdelavi specifikacije za naš program vodenja zalog smo jasno opisali namen programa, kaj so značilnosti in funkcionalnosti programa, na kakšen način želimo, da program shranjuje podatke, kakšno poslovno logiko naj program uporablja ter kako želimo, da se naš uporabniški vmesnik obnaša. Na koncu dodamo omejitve in predpostavke, katero tehnologijo bomo uporabili za izdelavo (programski jezik), ter pomembne podrobnosti implementacije.

4.2.1.1 Specifikacija programa invent.py

Upravljanje zalog: omogoča ustvarjanje novih izdelkov s samodejno generiranimi 5-mestnimi številkami, ki se začnejo z 00001. Uporabnik lahko pregleduje in išče izdelke ter spremlja trenutno količino zalog, ki se samodejno posodabljajo ob vsakem prejemu ali dostavi izdelkov.

Sistem skladiščnih lokacij: vključuje 100 lokacij (Loc1–Loc100), od katerih lahko vsaka shrani do 50 enot različnih izdelkov. Lokacije se avtomatično določijo glede na prosti prostor, količine izdelkov po lokacijah pa so vedno vidne uporabniku. Ob dostavi se izdelki odstranjujejo iz skladišča samodejno po vrstnem redu lokacij (FIFO).

Upravljanje dokumentov: zajema ustvarjanje potrdil o prejemu (PREVZEMNICA) za prejeto blago od dobaviteljev in dobavnic (IZDAJNICA) za blago, dostavljeno strankam. Vse transakcije se hranijo kot dokumenti s seznamom izdelkov, datumom, časom in informacijami o poslovnih partnerjih ter jih je mogoče izvoziti v berljive .txt datoteke. Dokumente je možno iskati in pregledovati.

Grafični uporabniški vmesnik (tkinter): sestavlja šest oken z navpičnimi drsnimi trakovi. Prvo okno prikazuje seznam zalog z možnostjo iskanja in ustvarjanja izdelkov. Drugo okno je namenjeno prikazu podrobnosti o izdelkih ali dokumentih. Tretje in četrto okno sta obrazca za ustvarjanje potrdil o prejemu (s podatki o dobavitelju) in dobavnic (s podatki o stranki), kjer uporabnik lahko vnese več artiklov hkrati. Peto okno prikazuje seznam dokumentov z možnostmi iskanja in izvoza, šesto pa pregledno tabelo, ki kaže, kateri izdelki so shranjeni na posameznih lokacijah.

Shranjevanje podatkov: poteka lokalno v treh datotekah JSON: inventory.json za izdelke in skupne količine, locations.json za količine na posameznih lokacijah in documents.json za zgodovino prejemov in dobav. Vsak dokument je možno izvoziti tudi v standardizirano besedilno datoteko .txt.

Poslovna logika: avtomatično razporeja izdelke po prostih lokacijah ob prejemu blaga in opozori uporabnika, če ni dovolj prostora. Ob izdaji blaga se izdelki odvzamejo iz najnižjih številk lokacij, pri čemer sistem prikaže napako, če zalog ni dovolj. Uporabniki lahko kadarkoli pregledajo stanje zalog po izdelkih in lokacijah. Vsak dokument ima status »USTVARJENO« ali »IZVOŽENO«, ki se ob izvozu spremeni.

Uporabniški vmesnik: omogoča enostavno ustvarjanje in iskanje izdelkov prek pogovornih oken, vnos več artiklov naenkrat pri dokumentih ter samodejno posodabljanje prikazov zalog in lokacij. Potrdila in napake so prikazane v pojavnih oknih, podrobnosti o artiklih ali dokumentih pa uporabnik vidi s klikom na posamezno vrstico.

Omejitve in predpostavke: določajo največ 50 enot na posamezno lokacijo, teh pa je skupno največ 100 (Loc1–Loc100). Podatki se shranjujejo lokalno v datotekah JSON, brez možnosti omrežne povezave. Uporabniški vmesnik je prilagojen za uporabo na namiznih računalnikih z minimalno ločljivostjo zaslona 1600 x 900.

Uporabljene tehnologije: program je izdelan v jeziku Python 3 z uporabo knjižnice tkinter za grafični vmesnik in datotek JSON za trajno shranjevanje podatkov, pri čemer se uporabljajo le standardne knjižnice. Implementacija vključuje ločene razrede za upravljanje zalog, lokacij in dokumentov ter osrednji razred za grafični uporabniški vmesnik. Vseh šest vmesniških oken temelji na okvirjih in drevesnih prikazih z drsnimi trakovi. Dokumenti se izvažajo v oblikovanih besedilnih datotekah za pravne ali arhivske namene, sistem pa opozarja uporabnika v primeru preseženih omejitev skladiščenja ali zalog.

Predvideni uporabniki: program je namenjen skladiščnim delavcem, vodjem zalog in pisarniškemu osebju v malih in srednje velikih podjetjih.

Specifikacijo shranimo v beležko z imenom specification.txt. Priložimo jo v ChatGPT, kadar izvedemo kreiranje programa.

### 4.2.2 Arhitektura programa

Ta sistem je namizna aplikacija za upravljanje zalog in dokumentov, razvita z uporabo orodij Python tkinter GUI. Ponuja vmesnik za upravljanje zalog, dodeljevanje zalog fizičnim lokacijam in ustvarjanje transakcijskih dokumentov za blago: prevzem (prevzemnica) in dobava (dobavnica). Vsaka transakcija posodobi trajne podatkovne shrambe, ki temeljijo na JSON, za zaloge, lokacije in dokumente. Sistem je prilagojen primerom uporabe, kot so nadzor zalog v skladiščih ali malih podjetjih, kjer so preprosti uporabniški vmesniki dovoljšna rešitev za upravljanje z zalogami.

Program mora omogočati upravljanje seznama izdelkov z avtomatično ustvarjenimi edinstvenimi številkami in količinami. Potreben je pregleden grafični uporabniški vmesnik (GUI), izdelan z tkinterjem, ki omogoča enostavno iskanje in dodajanje izdelkov ter prikaz aktualnega stanja zalog.

Zahteva se sistem s 100 lokacijami, kjer vsaka lokacija lahko sprejme do 50 enot, pri čemer program samodejno razporeja prejete izdelke glede na razpoložljive kapacitete lokacij. Pri izdaji izdelkov mora sistem avtomatično odštevati količine z ustreznih lokacij glede na razpoložljivo zalogo.

Program naj generira dokumente za prevzeme in dobave, pri čemer vsak dokument vsebuje status »USTVARJENO« ali »IZVOŽENO« in omogoča izvoz v jasno strukturirano datoteko .txt. Prav tako mora biti na voljo jasen pregled zgodovine dokumentov, skupaj s prikazom stanja in vsebine posameznih dokumentov ter možnostjo pregleda razporeditve izdelkov po lokacijah.

Za vse opisane funkcionalnosti je potreben popolnoma delujoč grafični vmesnik, sestavljen iz šestih glavnih plošč.

V računalniškem programu za vodenje skladišč je treba podatke shranjevati na jasen in pregleden način, zato se uporablja struktura, imenovana JSON. To so besedilne datoteke, ki na enostaven način predstavljajo informacije o izdelkih, lokacijah skladišč in dokumentih, povezanih z blagom.

Sistem je zasnovan modularno in uporablja tri glavne upravljalnike podatkov: inventory.json skrbi za seznam izdelkov in njihove količine, location.json razporeja izdelke na 100 lokacij s fiksno kapaciteto, document.json pa ustvarja dokumente in spremlja njihov status.

inventory.json
Ta datoteka shrani seznam vseh izdelkov, ki jih podjetje vodi v skladišču. Vsak izdelek je opisan s številko izdelka (product_number), nazivom izdelka (name) in skupno količino (quantity), ki je trenutno na voljo.

locations.json
V tej datoteki je zapisano, kje natančno v skladišču so izdelki shranjeni. Vsaka lokacija ima svojo oznako (npr. Loc1) in seznam izdelkov z njihovimi količinami, ki so shranjeni na tej specifični lokaciji. Tako lahko hitro preverimo, na katerih lokacijah je določen izdelek in koliko ga je tam shranjenega.

documents.json
Ta datoteka shrani zgodovino dokumentov, povezanih z gibanjem izdelkov (npr. prejeta ali izdana roba). Vsak dokument ima ime (ki vključuje datum, čas in

poslovnega partnerja), vrsto dokumenta (PREVZEMNICA za prejete artikle, IZDAJNICA za izdane artikle), podatke o poslovni stranki, datum, seznam izdelkov z njihovimi količinami, kje so shranjeni izdelki in status (npr. CREATED za ustvarjen dokument). To omogoča sledenje vseh pomembnih transakcij v skladišču.

Izgled struktur treh datotek JSON – inventory.json, locations.json in documents.json:

inventory.json:

```
[
  {
    "product_number": "00001",
    "name": "Vilica",
    "quantity": 100
  }
]
```

locations.json:

```
[
  {
    "Loc1": {
      "products": {
        "00001": 20,
        "00002": 5
      }
    }
  }
]
```

documents.json:

```
[
  {
    "name": "PREVZEMNICA_DATE_TIME_SUPPLIERX",
    "type": "PREVZEMNICA",
    "party": "SUPPLIERX",
    "date": "2025-05-16 14:50",
    "items": [
      {
        "name": "Widget A",
        "qty": 50,
        "stored": 50,
        "locations": [["Loc1", 30], ["Loc2", 20]]
      }
```

```
    ],
    "status": "CREATED"
  }
]
```

Arhitekturo programa shranimo v beležko z imenom architecture.txt. Priložimo jo v ChatGPT, kadar izvedemo kreiranje programa.

## 4.3 Kreiranje orodja s pomočjo umetne inteligence – chatgpt

ChatGPT deluje kot programer. Zagotoviti moramo specifikacijo, sledimo postopku, preizkusimo kodo in dobimo takojšnjo pomoč pri vsakem koraku – od logike zaledja do uporabniškega vmesnika in dokumentacije.

### 4.3.1 Specifikacija

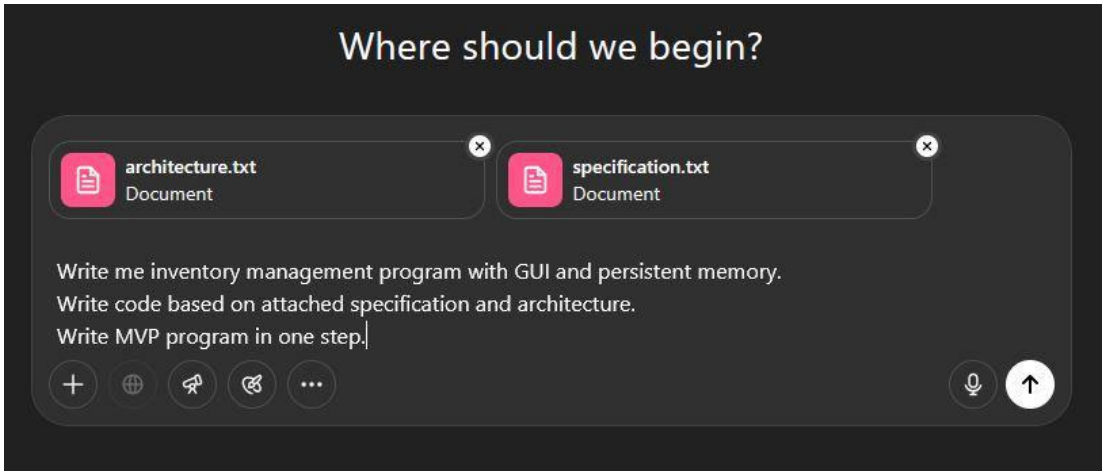
- Prepričajte se, da je specifikacija jasna, strukturirana in zajema vse ključne zahteve (funkcije, podatke, potek dela, uporabniški vmesnik itd.).
- Boljša kot je specifikacija, višja je kakovost nastalega programa.


*Slika 10: Datoteka specifikacije*
(Lastni vir)


*Slika 11: Datoteka arhitekture*
(Lastni vir)

Predložitev dokumentov v ChatGPT

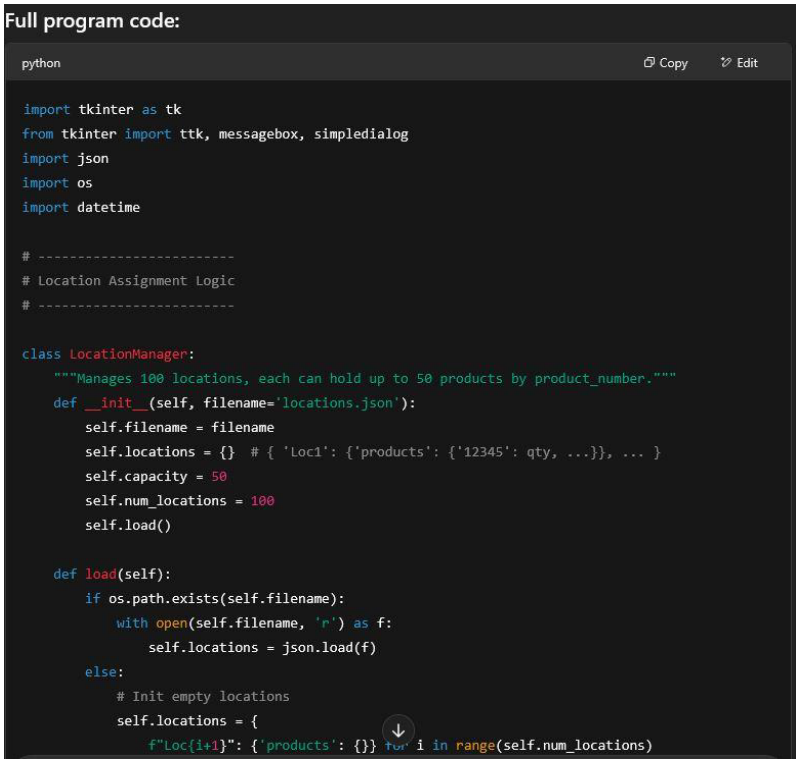
- Dodamo dokumenta specifikacija in arhitektura v klepet ChatGPT.
- ChatGPT prosimo, naj napiše kodo Python v skladu s tema dokumentoma.
- Navedemo, da želimo minimalno izvedljiv izdelek (MVP), aplikacijo s polnimi funkcijami.




*Slika 12: Navodila umetni inteligenci za kreiranje programskega opreme*
(Lastni vir)

### 4.3.2 Razvojni proces

- ChatGPT bo ustvaril kodo v skladu s podano specifikacijo.
- Po vsaki iteraciji preizkusite kodo na računalniku (zagotoviti je treba, da je nameščen Python 3.7+).




*Slika 13: Začetek (prva vrstica) programske kode*
(Lastni vir)

Umetna inteligenca nam izpiše celotno kodo programa. V našem primeru imamo 589 vrstic kode.

```
584                 ))
585
586
587 if __name__ == "__main__":
588     app = InventoryApp()
589     app.mainloop()
590
```

*Slika 14: Zadnja vrstica programske kode*
(Lastni vir)

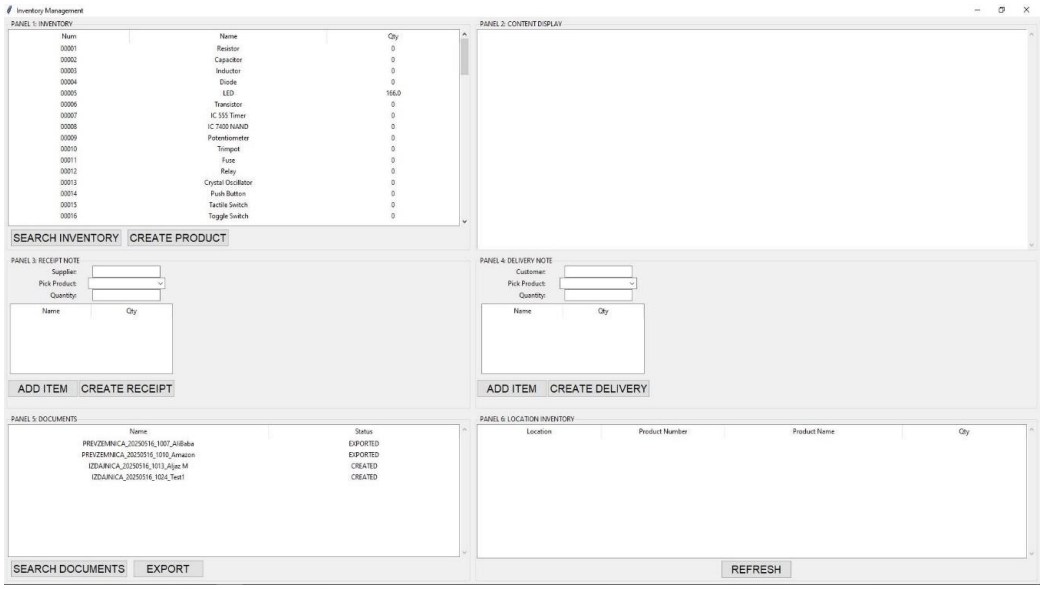
- Če imamo napake ali manjkajoče funkcije, ukažemo ChatGPT, naj nam jih popravi.
- Nadaljujemo ta cikel, dokler program ne ustreza našim potrebam.

### 4.3.3 Dokončanje

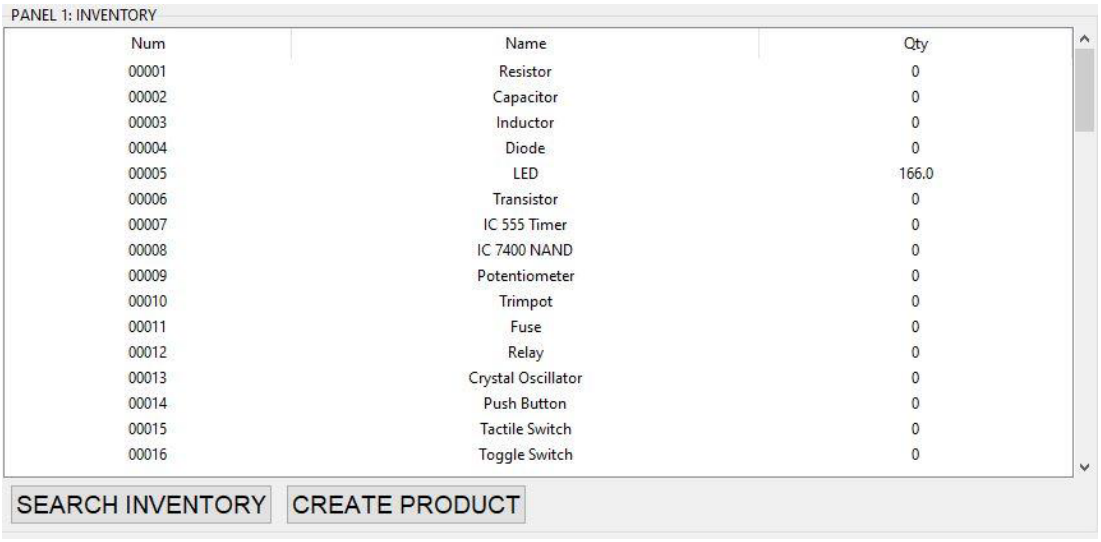
Ko program deluje po potrebi, lahko zaprosimo za pomoč pri pakiranju, navodilih za izvoz in kreiranju izvedljive datoteke.

## 4.4 Grafični uporabniški vmesnik – gui

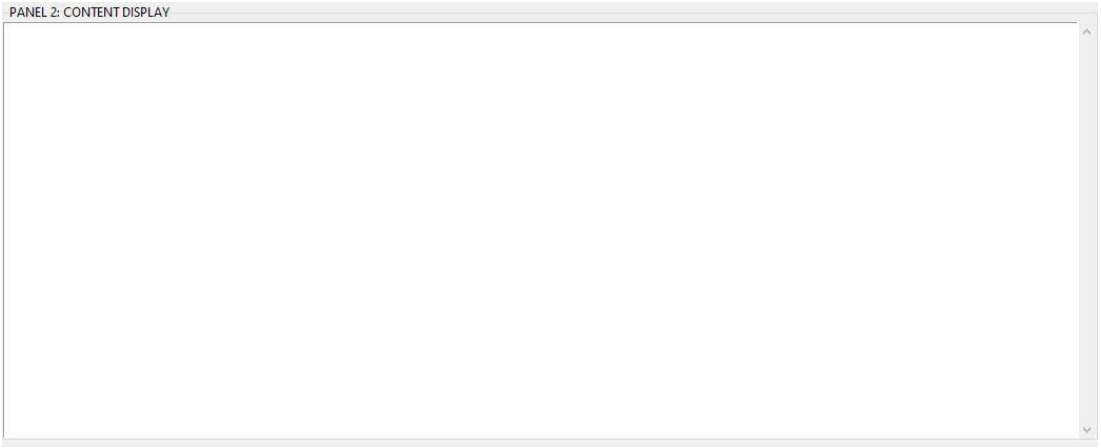
Izgled grafičnega vmesnika kreiranega programa.



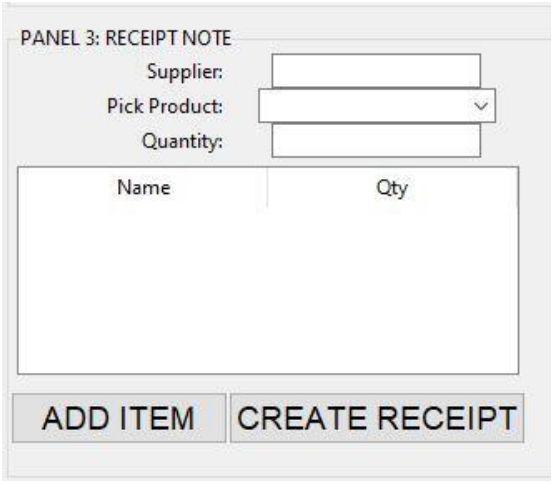

*Slika 15: GUI ali grafični vmesnik programske opreme Invent*
(Lastni vir)



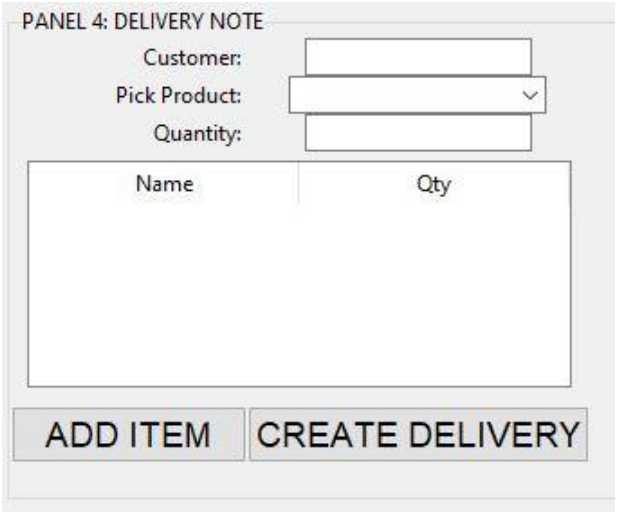

*Slika 16: Okno, kjer vidimo stanje naše zaloge*
(Lastni vir)



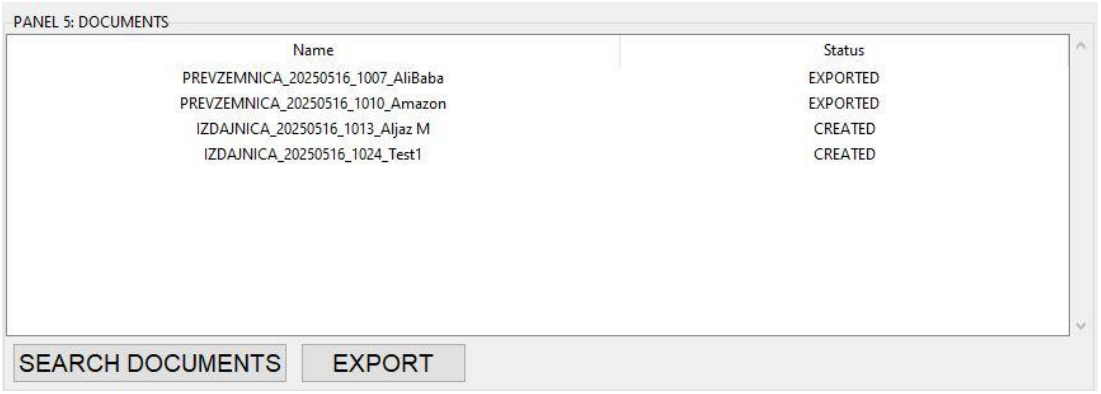

*Slika 17: Okno za prikaz vsebine: prikaže lastnosti izbranega artikla, dokumenta ali lokacije*
(Lastni vir)



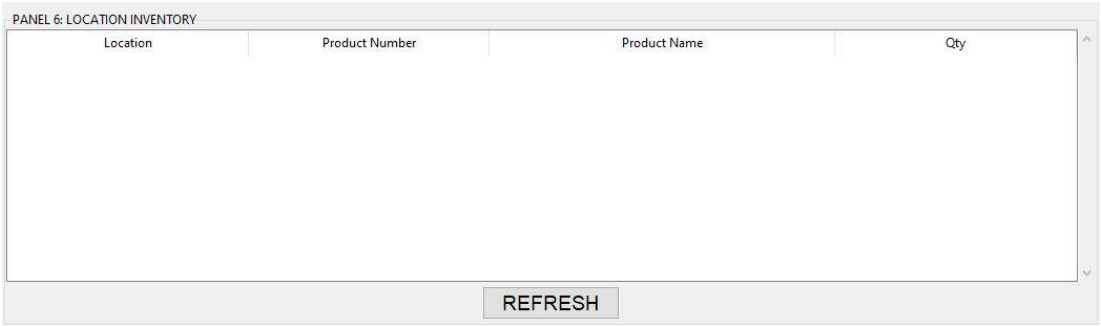

*Slika 18: Okno za prevzem, v katero vpišemo dobavitelja, artikel in količino artikla*
(Lastni vir)




*Slika 19: Okno za kreiranje izdajnice, v katero vpišemo kupca, izberemo artikel in vpišemo prodano količino*
(Lastni vir)




*Slika 20: Okno, v katerem so prikazani ustvarjeni dokumenti s statusom*
(Lastni vir)




*Slika 21: Okno, ki prikazuje status lokacij*
(Lastni vir)

## 4.5 Postopek izdelave izvršljive (.exe) datoteke iz python kode s pomočjo powershella

Za distribucijo Python programov na računalnikih, kjer Python ni nameščen, je smiselno izvorno kodo pretvoriti v samostojno izvršljivo datoteko (.exe). To omogoča, da lahko končni uporabniki program zaženejo neposredno z namizja. To storimo s pomočjo orodja PyInstaller in okolja PowerShell v operacijskem sistemu Windows.

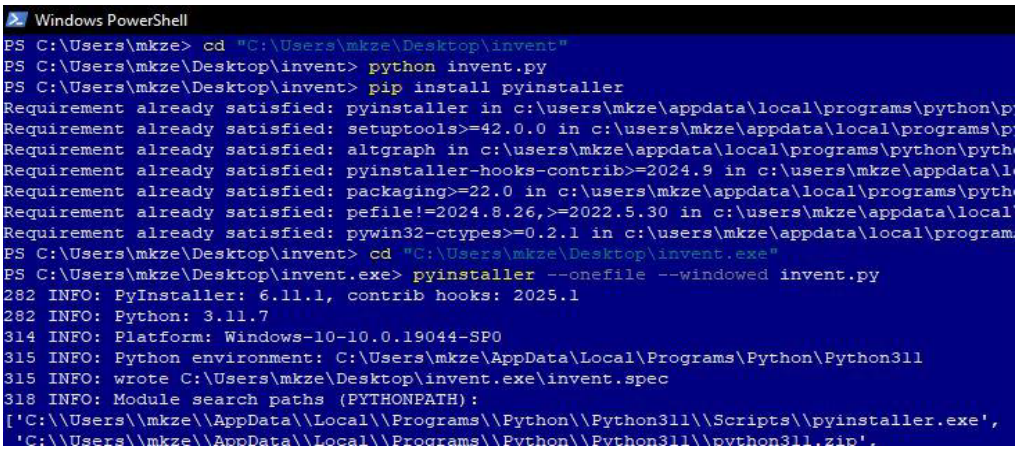
Najprej mora biti na računalniku nameščen Python. Namestitveni program je na voljo na uradni spletni strani python.org. Pri namestitvi je priporočljivo izbrati možnost »Add Python to PATH«, saj to omogoča izvajanje ukazov iz katerekoli mape v ukazni vrstici oziroma PowerShellu.

Za izdelavo datoteke .exe iz Python kode bomo uporabili orodje PyInstaller. Namestimo ga prek ukazne vrstice (PowerShell) z naslednjim ukazom:
**»pip install pyinstaller«**

Ta ukaz bo prenesel in namestil potrebno orodje.
V PowerShellu se nato premaknemo v mapo, kjer se nahaja izvorna Python datoteka (.py), ki jo želimo pretvoriti v .exe. Če imamo na primer datoteko »invent.py« na namizju, uporabimo ukaz:
**cd »C:\USers\mkze\Desktop\invent«**

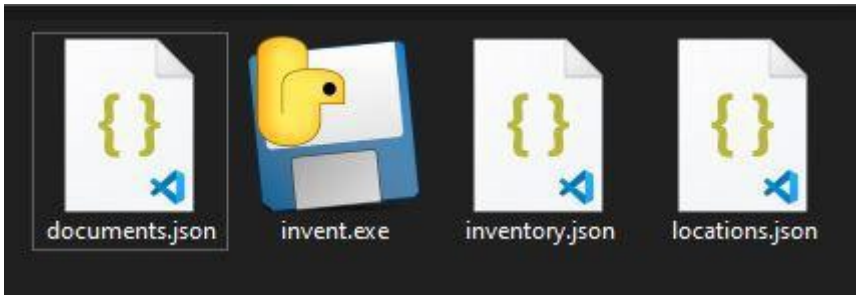
V isti PowerShell seji nato izvedemo naslednji ukaz:
**»pyinstaller --onefile moj_program.py«**

Pri tem parameter --onefile poskrbi, da bo vsebina programa zapakirana v eno samo .exe datoteko, kar olajša distribucijo.




*Slika 22: PowerShell po vpisanih zahtevah za kreiranje izvršljive datoteke Invent*
(Lastni vir)


*Slika 23: Izgled izvršljive datoteke .exe*
(Lastni vir)




*Slika 24: Poleg izvršljive datoteke .exe se ustvarijo še datoteke .json za shranjevanje podatkov*
(Lastni vir)

# 5 ZAKLJUČEK

Cilj diplomskega dela je bil razviti program za vodenje zalog s pomočjo sodobnih informacijskih tehnologij – umetne inteligence (ChatGPT). Program smo izdelali v programskem jeziku Python, kar nam je omogočilo hitro in učinkovito implementacijo vseh potrebnih funkcionalnosti. Ugotovili smo, da lahko tudi z osnovnim predznanjem s področja informacijske tehnologije v kombinaciji z znanjem z drugih strokovnih področij razvijemo programsko rešitev, ki bistveno izboljša in optimizira naše poslovne procese. Diplomsko delo dokazuje, da je digitalizacija skladiščnih procesov dosegljiva in izvedljiva tudi za posameznike ali manjša podjetja, ki se želijo podati na pot razvoja lastnih informacijskih rešitev.

# 6 LITERATURA IN VIRI

Bass, L., Clements, P., & Kazman, R. (2013). *Software Architecture in Practice, 3rd Edition.* Addison-Wesley.

Čižman, A. (2001). *Upravljanje logističnih procesov v organizaciji.* Kranj.

https://cscmp.org. (2025). Management Professionals Council of Supply Chain.

Kaltnekar, Z. (1993). *Logistika v proizvodnem podjetju.* Kranj: Moderna organizacija.

Klemen, K. (1975). *Management oskrbovalnih verig in model taktnega časa.* Koper: Univerza na Primorskem.

Ljubič, T. (2006). *Operativni management proizvodnje.* Kranj: Založba Moderna organizacija.

Logožar, K. (2004). *Poslovna logistika: Elementi in podsistemi - 1. natis.* Ljubljana: GV Izobraževanje.

Lutz, M. (July 2013). *Learning Python: Powerful Object-Oriented Programming, 5th Ed.* O'Reilly Media.

Požar, D. (1985). *Teorija in praksa transporta in logistike.* Maribor: Obzorja.

Rak, G. (2011). *Logistika notranjega transporta in skladiščenja.* Ljubljana: Konzorcij višjih strokovnih šol za izvedbo projekta IMPLETUM.