MINISTERUL EDUCAȚIEI NAȚIONALE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

ASISTENT PENTRU VÂNZAREA ȘI GESTIONAREA MĂRFURILOR UNOR MAGAZINE

LUCRARE DE LICENŢĂ

Absolvent: **Sergiu-Ioan MORAR**

Coordonator ştiinţific: **As. Ing. Cosmina IVAN**

2020

Cuprins

Capitolul 1. Introducere – Contextul proiectului

Proiectul propune dezvoltarea unei aplicatii de tip client-server care vine ajutorul celor ce lucreaza in magazinele de mici dimensiuni in vanzare, managerierea stocurilor si comenzi.

1.1. Contextul proiectului

La momentul actual, in magazinele de mici dimensiuni se folosesc mult prea putin solutiile informatice pentru o mai buna organizare a functionarii intreprinderii, intrucat exista o anumita reticenta fata de solutiile de acest fel.

Scopul aplicatiei este acela de a oferi o varianta noua, simpla si intuitiva pentru activitatile zilnice dintr-un magazin, totul intr-un mod cat mai flexibil. Odata configurata aplicatia, aceasta va oferi suport in vanzari prin crearea automata a bonului fiscal si actualizarea automata a stocului, comenzi prin generarea de fisier PDF si trimiterea lui prin email catre furnizor, crearea automata de facturi sau avize de insotire a marfii.

Avantajul acestei aplicatii este acela ca vanzarea este mult mai rapida si usoara. De asemenea, monitorizarea stocurilor este mult mai simpla intrucat toate datele sunt centralizate si toate operatiile sunt simplificate.

1.2. Motivația

Accesibilitate si viteza sunt dorinte recurente in randul utilizatorilor in aceste vremuri. Toata lumea isi doreste sa primeasca informatia rapid si intr-un mod cat mai usor de inteles. Aplicatia va simplifica vanzarea si gestionarea stocurilor la vanzare, comanda, avize sau facturi.

- Procesul de vanzare va fi simplificat prin:
  - Eliminarea introducerii produs cu produs la casa de marcat pentru a fi imprimat pe bonul fiscal;
  - Folosirea unui scanner pentru coduri de bare in scopul de a micsora timpul necesar unei vanzari;
  - Actualizarea automata a stocului in momentul vanzarii
  - Posibilitatea crearii de facturi sau avize din aplicatie care pot fi imprimate ulterior.
- In ceea ce priveste gestionarea stocurilor:
  - Va exista posibilitatea trimiterii de comenzi catre furnizor prin email;
  - Crearea unei intrari in stoc pe baza unei comenzi sau produs cu produs;
  - Adaugarea de pierderi in sistem;
  - Posibilitatea monitorizarii simple si convenabile a tuturor stocurilor.

1.3. Conținutul lucrării

Acest subcapitol are rolul de a expune structura lucrarii si o prezentare scurta a informatiei prezenta in fiecare capitol.

Capitolul 1. Introducere – Scopul acestui capitol este acela de a prezenta pe scurt tema proiectului, precum si structura lucrarii, pentru a facilita intelegerea subiectului.

Capitolul 2. Obietivele Proiectului – Capitolul cu numarul doi descrie obiectivele principale, precum si cele secundare care vor fi atinse in dezvoltarea proiectului.

Capitolul 3. Studiu Bibliografic – Continutul acestui capitol va fi reprezentat de descrierea infrastructurii necesare unei bune functionari a intregului sistem, precum si un studiu comparativ cu sisteme similare.

Capitolul 4. Analiză și Fundamentare Teoretica – Capitolul patru este rezervat prezentarii cerintelor functionale si non-functionale. De asemenea, vor fi prezentate functionalitatile disponibile fiecarui tip de utilizator. Totodata, vor fi prezentate si tehnologiile utilizate in construirea aplicatiei.

Capitolul 5. Proiectare de Detaliu și Implementare – Acest capitol va contine descrierea detaliata a arhitecturii conceptuala a aplicatiei. De asemenea, tot in acest capitol vor fi prezentate si diagrama bazei de date, diagrama de deployment precum si diagrama de pachete si clase. Tot in acest capitol vor fi descrise modulele si componentele constituente.

Capitolul 6. Testare si Validare – Capitolul 6 va prezenta modurile in care buna functionare a componentelor, precum si a intregului sistem a fost verificata.

Capitolul 7. Manual de Instalare si Utilizare – Scopul acestui capitol este acela de a prezenta minimul de programe software si hardware pentru a putea instala si utiliza sistemul.

Capitolul 8. Concluzii - In acest capitol vor fi prezentate concluziile observate in urma dezvoltarii proiectului. De asemenea, vor fi descrise posibilitatile de imbunatatire a sistemului in viitor.

Capitolul 2. Obiectivele Proiectului

In acest capitol este prezentata o decriere a obiectivelor proiectului, atat cele principale cat si cele generale, care vor fi obtinute prin dezvoltare. Finalitatea este reprezentata de un sistem cu o utilizare simpla, eficienta si sigura.

2.1. Obiective principale

Un obiectiv principal al aplicatiei este acela de a fi un asistent in gestionarea magazinelor de mici dimensiuni prin simplificarea vanzarii si a gestionarii stocurilor. Obiectivele principale sunt urmatoarele:

- **Vanzarea:** vanzatorul va cauta produsul dupa codul de bare cu ajutorul scannerului pentru coduri de bare, va introduce cantitatea, iar la final va specifica metodele de plata si va finaliza vanzarea fiind imprimat si bonul fiscal;
- **Facturarea:** vanzatorul va selecta clientul dupa care va putea adauga produse la fel ca in vanzare; plata va putea fi facuta la final sau la o data de timp ulterioara din sectiunea „Vanzare” unde factura va putea fi importata;
- **Comanda:** va fi selectat un furnizor, precum si magazinul destinatie, adaugate produsele pentru comanda, urmand a fi creat un fisier PDF care va fi trimis prin email furnizorului daca utilizatorul doreste;
- **Intrarea:** odata livrata comanda, produsele primite vor putea fi introduse in sistem;
- **Avize:** vor putea fi create avize de insotire a marfii;
- **Pierderi:** in cazul in care exista produse a caror stare nu mai permite vanzarea acestea vor putea fi adaugate in sectiunea pierderi;
- **Generarea de rapoarte:** vor putea fi generate rapoarte sub format Excel din sistem pentru vanzari, facturi, comenzi, intrari, pierderi si stoc actual;
- **Intrari si iesiri ale casei de marcat:** cu ajutorul aplicatiei vor putea fi create intrari sau iesiri ale casei de marcat care reprezinta operatiuni cu banii din sertarul casei de marcat;
- **Gestionarea furnizorilor:** in sistem vor putea fi introdusi, stersi sau actualizati furnizorii;
- **Gestionarea clientilor:** in sistem vor putea fi introdusi, stersi sau actualizati clientii pentru care se genereaza factura sau aviz;
- **Gestionarea utilizatorilor:** utilizatorii cu drepturi de administrator vor putea adauga, dezactiva sau actualiza drepturile utilizatorilor, precum si reseta parolele;
- **Setari personale:** fiecare utilizator va putea sa isi modifice informatiile personale din sectiunea „setari personale”, precum si reseta parola.

Capitolul 3. Studiu Bibliografic

In acest capitol vor fi prezentate si analizate componenetele necesare sistemului, respectiv casa de marcat Datecs MP-55 si scannerul pentru coduri de bare Datalogic Heron D150. De asemenea va prezentata si arhitectura aplicatiei precum si aplicatii similare.

Avand in vedere ca lumea este intr-o continua schimbare, progresele de ordin tehnologic fiind vizibile cu ochiul liber, este necesar ca si comertul sa se foloseasca de toate resursele disponibile pentru a eficientiza procesele ce apar in magazinele mici sau mari, precum generarea de documente, interogarea stocului sau managementul clientilor si al furnizorilor.[1]

Chiar daca mecanismele de operare din cadrul comercial sunt optimizate folosind tehnologiile tot mai noi, esenta proceselor comerciala ramane neschimbata. Operatiuni precum facturarea sau vanzarea sunt la fel ca la inceputuri, respectandu-se aceleasi seturi de reglementari pentru o buna organizare si evitarea fraudei.[2]

3.1. Echipamente și software aditional

Pentru realizarea completa a sistemului au fost necesare dispozitive auxiliare, respectiv casa de marcat si scannerul pentru coduri de bare. De asemenea, pentru comunicarea dintre calculator si casa de marcat a fost necesara selectia unuia dintre programele disponibile.

3.1.1. Casa de marcat Datecs MP-55

Prima casa de marcat, care era de fapt un abac, a fost construita de catre James Ritty in anul 1879 si patentata ulterior in anul 1883. Patentul a fost vandut lui Jacob H. Eckert care a adaugat produsului un sertar pentru bani, precum si un clopotel care suna la deschiderea sertarului. Cativa ani mai tarziu, John H. Patterson se alatura lui Eckert, iar in anii urmatori vor aparea functionalitati noi, precum imprimarea unui bon.

Inventatorul Charles F. Kettering a fost cel care, odata angajat de compania producatoare de case de marcat, a creat prima casa de marcat electronica . [3]

Datecs MP-55 este un aparat de marcat electronic fiscal de dimnesiuni medii pentur magazinele cu un numar de vanzari reduse precum restaurante mici, frizerii sau alte tipuri de comert.[1]

Datecs MP-55 este un aparat de marcat electronic fiscal care poate functiona atat in regim autonom, cat si integrat cu un calculator. Principalele componente ale dispozitivului sunt:

- cele doua afisaje, unul pentru client si celalalt pentru vanzator;
- tastatura cu 33 de taste grupate in taste: numerice, de functii si departamente;
- imprimanta;
- memoria de tip CMOS RAM cu capacitatea de 128 Kbyte. Scopul este acela de acumulator tampon asigurand protectia informatiilor pentru cel putin 90

[1] https://www.ecrshop.ro/datecs-mp-55-l.html

de zile de la decuplarea tensiunii de alimentare. In aceasta memorie sunt stocate informatiile produselor, precum si totalurile acumulate;

- porturile seriale: 4 porturi RS232 pentru conexiunea la un calculator, cititor pentru coduri de bare, cantar si afisaj suplimentar. [2]

Figura 3.1 – Casa de marcat Datecs MP-55

3.1.2. Scanner pentru coduri de bare Datalogic Heron D130

In anul 1952 Norman Woodland si Bernard Silver duc la bun sfarsit constructia primului scanner pentru coduri de bare, obtinand si patentul pentru acesta in acelasi an. In anii urmatori, companii importante din Statele Unite ale Americii au inceput sa foloseasca acest produs precum Asociatia Cailor Ferate Americane (1967), General Motors (1969) sau Kroger (1972). [4]

Prima forma de cod de bare conceputa de Woodland si Silver, a fost una sub forma unor cercuri concentrice, numita „bull's eye". A doua forma, cea mai cunoscuta de altfel si in ziua de azi, a fost cea liniara (1D), aceasta permitand reprezentarea caracterelor numerice precum si a altor caractere. Ultima forma a fost cea care organiza informatia atat orizontal, cat si vertical (2D), aceasta purtand denumirea de QR code. [3]

Scannerul pentru coduri de bare Datalogic Heron D130 suporta doar codurile de tip 1D, cele de tip QR code nefiind suportate. Acesta poate fi conectat la un computer printr-o interfata USB sau RS232. De asemenea, poate fi conectat direct si la o casa de marcat tot printr-o interfata de tip RS232.

[2] https://docs.google.com/file/d/0B2dEfaLeyBiiZHBXWUxDZjdkZWM/edit
[3] https://www.barcodesinc.com/news/evolution-of-the-barcode/

Acest scanner este dedicat fiecarui utilizator, putand citi coduri de bare de la o distanta de pana la 20 cm la punctele de vanzare dedicate. Poate fi folosit atat de pe suportul dedicat cat si tinut in mana pentru o mai buna manevrabilitate.

Figura 3.2 – Scanner pentru coduri de bare Datalogic Heron D130

3.1.3. Programe conexiune casă de marcat – calculator

Pentru a reusi imprimarea bonurilor fiscale pe casa de marcat folosind calculatorul este necesar un program auxiliar care sa realizeze conexiunea dintre calculator si casa de marcat.

Programele disponibile cele mai cunoscute in vederea initierii imprimarii unui bon fiscal sunt Fprint, DatPrint si SellText. Toate aceste programe sunt construite de catre DATECS.[4][5]

Toate aceste programe necesita pentru imprimarea bonului fiscal un fisier in care sa fie specificate numele produselor, cantitatea, pretul precum si valoarea de achitat si forma in care este achitat: cash, card sau tichete de masa. Fisierul trebuie scris intr-un format anume astfel incat sa fie acceptat de casa de marcat

Fprint este cel mai simplu dintre cele trei. Este necesara configurarea prealabila a programului specificand modelul casei de marcat, portul de comunicatie, frecventa de comunicatie cu casa de marcat precum si fisierele de tip „in” si „out”. Dezavantajul este acela ca poate fi monitorizata doar ultima comanda catre casa de marcat. Alt dezavantaj este dat de faptul ca exista momente in care programul nu reuseste comunicarea cu casa de marcat iar imprimarea nu mai are loc.

[4] http://informedia.ro/download

[5] https://www.danubius-exim.ro/licenta-pentru-driverele-programele-softurile-oferite-de-datecs.html

Cel de-al doilea program este DatPrint care la fel are nevoie de setari prealabile, respectiv viteza de comunicatie, portul, fiserul de tip „in” si cel de tip „out”. Acest program ofera fata de cel precedent posiblitatea crearii mai multor fisiere de tip istoric. Insa, un dezavantaj comun cu Fprint este acela ca exista momente in care imprimarea nu are loc, comunicatia fiind fara succes intre program si casa de marcat.

Ultimul si cel mai potrivit program este SellText. Aceasta nu are nici unul dintre dezavantajele celoralte doua program prezentate. Realizeaza comunicatia cu succes cu casa de marcat, fara erori. Buna functionare este data de faptul ca spre deosebire de celelalte doua programe, acesta monitorizeaza doua foldere, unul pentru fisiere de tip „in” si unul pentru fisiere de tip „out”, spre deosebire de celelate care monitorizau doua fisiere. Astfel, se creeaza un fisier in folderul dedicat fisierelor de tip „in” cu extensia „inp” si la finalul procesarii este creat automat de catre program de tip „out” cu extensia „out”.

3.2. Sisteme similare

3.2.1. BOCP

BOCP, sau Business Online Control Panel, este o solutie de stoc-vanzari pentru diferite domenii de activitate, fiind disponibila in diferite module potrivite pentru fiecare segment din lucrul in comert.[6] Serviciile oferite sunt folosite in vaste domenii precum: produse naturiste, cosmetice, produse de design interior, electronice, librarii, service de electronice, etc.

Unul dintre module folosite este cel de gestiune si vanzari care ofera suport pentru vanzare in cadrul firmelor si intreprindelor mici si mijlocii. Acesta este folosit cand ai nevoie de raport stoc/vanzari mereu actual, vrei sa transporti produse intre punctele de lucru, ai mai multi utilizatori, se receptioneaza marfa, emiterea de facturi, etc. Modulul de facturare si automatizare facturi poate fi utilziat independent din orice locatie si la orice moment, putand conlucra cu celelalte module. De asemenea, poate fi introdus si statusul incasarilor si rambursarilor.

Magazinul online este un alt modul oferit de catre BOCP care ofera atat un website cat si partea de administrare, toate acestea oferind bineinteles o conectivitate usoara cu celelalte module. Un alt modul oferit este cel de Client Management care ofera informatiile clientilor si a furnizorilor, rapoarte referitoare la vanzari si furnizori, un istoric al achizitiilor si acces din orice locatie.

De asemenea, exista si modulul Document Management care se ocupa cu emiterea de devize, avize, fise service sau contracte. Ultimul modul oferit este cel de Work Management care este dedicat in special service-urilor electronice oferind urmarirea clientilor, crearea fiselor de reparatie, urmarirea stadiului reparatiei precum si a platilor.

3.2.2. SAGA

Exista trei produse oferite: SAGA C., SAGA B. si SAGA P.S. Primul dintre acestea este o solutie care ofera ajutor pe partea contabila pentru intreprinderile mici si mijlocii sau cabintele individuale de contabilitate. Astfel , este oferit suport in: contabilitatea financiara,

[6] https://www.bocp.eu/produse.html

stocarea informatiilor despre clienti si furnizori si a facturilor, avizelor, etc., salarizarea angajatilor, productie (in cazul producatorilor), operatii interne cu stocuri precum inventarierea sau bonurile de predare/primire, suport pentru conexiunea cu casele de marcat.

SAGA B. Este un produs care ofera informatizarea activitatii pentru institutiile bugetare. Printre facilitatile oferite: ALOP(Angajarea, Lichidarea si Ordonarea Platilor, mecanism prevazut in Legea Finantelor Publice), contabilitate financiara, clienti si furnizori, salarizarea, operatiunile interne cu stocul sau suportul pentru operatiuni in valuta.

SAGA P.S. este un produs pentru evidenta contabila si a stocurilor in partida simpla care este destinat persoanelor fizice autorizate, cabinetelor de avocatura sau celor notariale sau expertilor financiare. Acesta contine: contabilitate financiara, clienti si furnizori, salarizare, productie, operatii interne cu stocuri, urmarire centre productie, suport pentru legatura cu casele de marcat si module speciale dedicate cabinetelor de avocatura sau asociatiilor de proprietari. Faptul ca evidenta in partida simpla ofera avantajul de a reduce numarul de documente care trebuie completate.

De asemenea, sunt disponibile si extensii pentru activitati specializate. FGO(Facturezi Gratuit Online) este extensia dedicata facturarii si urmariri de pe dispozitive mobile a facturilor emise, situatia furnizorilor si statusul angajatilor. PubLine este o alta extensie disponibila atat pe PC cat si pe mobil dedicata activitatii in restaurante, cafenele sau unitati cu facilitati de cazare. MarketLine este dedicat vanzarii in magazine, iar ultima extensie este cea dedicata generarii automate de documente pentru ANAF. [7]

3.2.3. Ebriza

Ebriza ofera solutii dedicate sub forma de module specifice fiecarui tip de activitate: horeca, retail sau servicii. La fel ca in cazul celorlalte variante prezentate, achizitia poate fi facuta pe module specifice necasitatilor. [8]

Principalul produs este cel de POS care faciliteaza vanzarea in mai multe puncte, conectivitatea cu casa de marcat, emiterea facturilor, raportarea sau aplicarea reducerilor. Un alt modul oferit este cel de ECommerce unde utilizatorul poate sa isi transforme magazinul fizic in unul online, utilizatorul putand produsele disponibile la vanzare.

Kiosk ofera clientilor posiblitatea de a comanda si plati singuri preparatele dorite, fiind folosit in principal in food-court-uri sau localuri cu flux mare. Diferite module pentru asistenta in efectuarea comenzilor precum FoodPanda, comenzile telefonice, web sau curieri. In situatia in care exista un singur punct de centralizare a comenzilor si multiple locatii de productie sau livrare, exista un modul separat prin care operatorul poate plasa comanda catre locatia potrivita.

Pentru parte de lucru cu clientii, exista un modul dedicat stocarii informatiilor clientilor si operatiilor precum emiterea de facturi, fidelizare si rapoarte. De asemenea, exista un modul dedicat pentru partea de marketing putand oferi diverse reduceri clientilor . Un alt modul este cel specializat pe partea de raportare cu posibilitate de a crea rapoarte despre vanzari, facturi, stocuri, utilizatori sau clienti.

[7] https://www.sagasoft.ro/index.php
[8] https://ebriza.com/app/public/marketplace/apps?

In continuare, sunt disponibile pentru monitorizarea in detaliu a stocurilor, inventariare, registru de casa in cadrul careia fiecare operatiune cu bani este monitorizata, antifrauda pentru monitorizarea in detaliu a angajatilor sau integrarea cu POS bancar. Un ultim modul este cel de alarma, in cadrul careia un operator poate seta alarme referitor la stocuri sub un anumit numar, activitati suspecte sau comenzi, toate informatiile fiind oferite prin email, astfel fiind oferita independenta fata de locatie.

3.3. Arhitectura Client-Server

Traim in era in care tehnologia informatiei joaca un rol de maxima importanta in aplicatiile business, fiind considerata o zona in care companiile sunt dispuse sa investeasca pentru a-si largi oportunitatile disponibile.[9]

Arhitectura client-server este o arhitectura in care mai multi clienti formuleaza request-uri catre un server primind raspunsuri. Serverul asteapta request-urile pentru a trimite informatiile cerute clientilor. In general, un serviciu este o abstractie a unei resurse, iar clientul nu trebuie sa fie interesat despre modul in care este procesata cererea. Clientul doar trebuie sa fie capabil sa inteleaga raspunsul primit de la server. De obicei, serverul ofera o putere o putere computationala mult mai ridicata decat dispozitivele client. La un moment dat de timp, mai multi clientii pot accesa simultan resursele disponibile pe server.

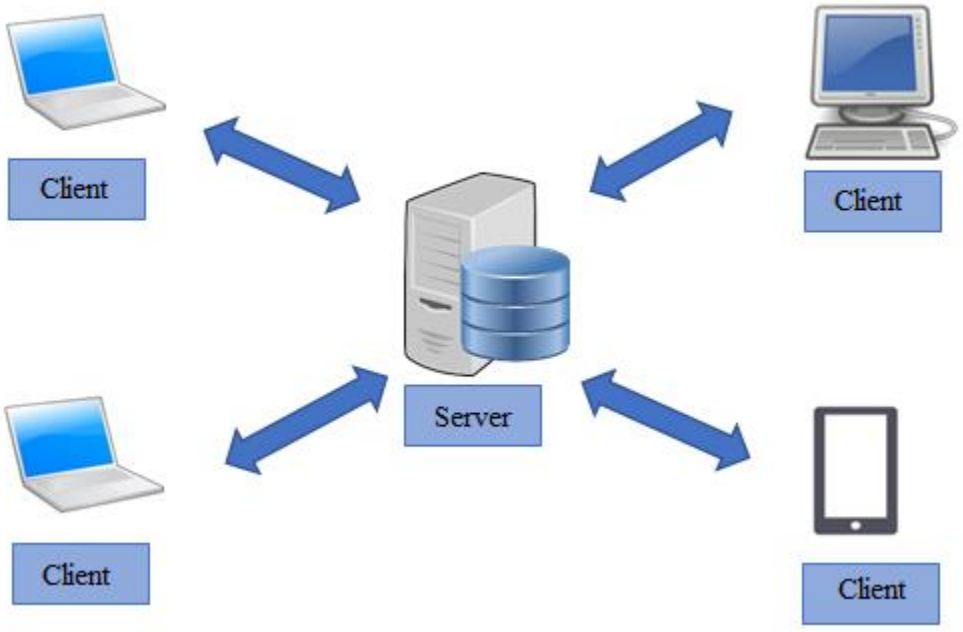

Figura 3.3 – Arhitectura Client-Server

In ceea ce priveste caracteristicile, masinile client si server au nevoie de resurse software si hardware si cantitati de resurse diferite, iar acestea pot apartine de producatori distincti. De asemenea, poate exista scalabilitate atat orizontala (cresterea numarului de

[9] https://ciowiki.org/wiki/Client_Server_Architecture#:~:text=Client%20Server%20Architecture%20is%20a,a%20network%20or%20internet%20connection.

clienti), cat si verticala (imbunatatirea serverului sau trecerea la o solutie care sa contina mai multe servere). In continuare, este necesar ca atat clientii cat si serverul sa foloseasca protocoalele de transport a informatiei pentru o comunicare cu succes a informatiei.

Exista patru mari tipuri de arhitecturi Client-Server. Prima este cea de tip One-Tier, in care exista un program care ruleaza pe un singur calculator fara a fi necesar accesul la internet. Cea de a doua este Two-Tier in care exista clientii si serverul si protocolul care leaga clientii de server. Interfetele utilizator se afla pe partea de client iar logica de procesare in server. [5]

Three-Tier este modelul in care exista clienti, server, si baza de date. La fel ca si in modelul precedent, exista un protocol care leaga clientii de server, interfetele utilizator se afla in client, logica de procesare in server, iar in baza de date sunt stocate datele. Ultimul model este cel N-Tier care imparte aplicatia in multiple niveluri care separa responsabilitatile, nivelurile fizice care ruleaza pe diferite masini, imbunatateste scalabilitatea insa creste latenta. Fiecare nivel nu poate comunica decat cu unul dintre cele doua niveluri vecine ale sale.

Capitolul 4. Analiză și Fundamentare Teoretica

Acest capitol este dedicat prezentarii cerintelor functionale si non-functionale. De asemenea vor fi expunse principalele cazuri de utilizare si tehonologiile folosite pentru a dezvolta intreaga aplicatie.

4.1. Cerințele sistemului

Cerintele functionale sunt reprezentate de activitatile intreprinse de aplicatie, in timp ce cerintele non-functionale sunt reprezentate de aspectele care influenteaza functionarea in parametrii doriti a aplicatiei.

4.1.1. Cerinte functionale

Nr.	Descriere
CF 1	Logare/Iesire
CF 2	Vanzare
CF 2.1	Cautare produs dupa cod de bare si afisarea listei cu produsele regasite
CF 2.2	Modificarea cantitatii produselor in vanzare
CF 2.3	Stergerea produselor din cos
CF 2.4	Cautarea unei facturi emisa deja pe baza datei calendaristice si selectarea ei, produsele fiind adaugate in cos
CF 3	Plata
CF 3.1	Scrierea valorii pentru fiecare tip de plata: cash, card sau tichete de masa
CF 3.2	Selectarea clientului in cazul in care este generarata o factura la care sa fie anexat bonul fiscal
CF 4	Factura
CF 4.1	Selectare client
CF 4.2	Cautare produs dupa cod de bare si afisarea listei cu produsele regasite
CF 4.3	Creare factura in format PDF
CF 5	Comanda
CF 5.1	Selectare furnizor si magazin destinatie
CF 5.2	Adaugare produs in evntualitatea in care acesta nu exista in baza de date
CF 5.3	Cautare dupa nume si selectie
CF 5.4	Cautare dupa furnizor si selectie
CF 5.5	Finalizare comanda cu generare de fisier PDF si posibilitatea transmiterii ei catre furnizor
CF 5.6	Modificare cantitate comanda
CF 5.7	Stergere produs comanda
CF 6	Intrare

CF 6.1	Selectare furnizor
CF 6.2	Cautare comanda pe baza datei calendaristice si selectarea ei, produsele fiind adaugate in tabel
CF 6.3	Adaugare produs
CF 6.4	Cautare dupa nume si selectie
CF 6.5	Cautare dupa furnizor si selectie
CF 6.6	Modificare cantitate intrare
CF 6.7	Stergere produs intrare
CF 7	Aviz de insotire a marfii cu selectie tip emitere sau receptie
CF 7.1	Emitere aviz
CF 7.1.1	Selectie magazin destinatie
CF 7.1.2	Cautare produs dupa nume
CF 7.13	Modificare cantitate aviz
CF 7.1.4	Stergere produs aviz
CF 7.2	Receptie aviz
CF 7.2.1	Selectie data aviz si selectie aviz
CF 8	Pierderi
CF 8.1	Cautare produs dupa nume si magazin curent cu selectie din lista
CF 8.2	Modificare cantitate pieredere
CF 9	Rapoarte
CF 9.1	Rapoarte vanzari
CF 9.1.1	Selectie data, magazin vanzare si operator pentru cautare
CF 9.1.2	Selectia uneia dintre vanzarile care corespunde cautarii si afisarea produselor vandute
CF 9.1.3	Exportul vanzarilor conform cautarii in format .xlsx
CF 9.2	Rapoarte facturi
CF 9.2.1	Selectie data, magazin facturare, operator si client pentru cautare
CF 9.2.2	Selectia uneia dintre facturi care corespunde cautarii si afisarea produselor facturate
CF 9.2.3	Exportul facturilor conform cautarii in format .xlsx
CF 9.3	Rapoarte comenzi
CF 9.3.1	Selectie data, magazin comanda, operator si furnizor pentru cautare
CF 9.3.2	Selectia uneia dintre comenzi care corespunde cautarii si afisarea produselor comandate
CF 9.3.3	Exportul comenzilor conform cautarii in format .xlsx
CF 9.4	Rapoarte intrari
CF 9.4.1	Selectie data, magazin intrare, operator si furnizor pentru cautare
CF 9.4.2	Selectia uneia dintre intrari care corespunde cautarii si afisarea produselor din intrare
CF 9.4.3	Exportul intrarilor conform cautarii in format .xlsx
CF 9.5	Rapoarte avize
CF 9.5.1	Selectie data, magazin sursa, magazin destinatie si operator pentru cautare
CF 9.5.2	Selectia uneia dintre intrari care corespunde cautarii si afisarea produselor din aviz

CF 9.5.3	Exportul intrarilor conform cautarii in format .xlsx
CF 9.6	Rapoarte pierderi
CF 9.6.1	Selectie data si operator pentru cautare
CF 9.6.2	Cautare produs dupa nume si selectie din lista
CF 9.6.3	Exportul pierderilor conform cautarii in format .xlsx
CF 9.7	Rapoarte stoc
CF 9.7.1	Cautare produs dupa nume si selectie din lista
CF 9.7.2	Cautare produs dupa magazin si selectie din lista
CF 9.7.3	Cautare produs dupa furnizor si selectie din lista
CF 9.7.4	Exportul stocului conform cautarii in format .xlsx
CF 10	Introducerea sau retragerea de bani din sertarul casei de marcat specificand suma
CF 11	Furnizori
CF 11.1	Modificare adresa, email, telefon, banca sau cont pentru furnizorii deja introdusi in sistem
CF 11.2	Adaugare furnizor nou
CF 11.3	Export furnizori in format .xlsx
CF 12	Clienti
CF 12.1	Modificare adresa, email, telefon, banca sau cont pentru clientii deja introdusi in sistem
CF 12.2	Adaugare client nou
CF 12.3	Export clienti in format .xlsx
CF 13	Setari personale
CF 13.1	Actualizare nume, adresa, email sau telefon
CF 13.2	Modificare parola
CF 14	Gestionare operatori
CF 14.1	Modificare rol operator si status
CF 14.2	Resetare parola operator
CF 14.3	Adaugare operator

Tabel 4.1 - Functionalitati

4.1.2. Cerinte non-functionale

Cerintele non-functionale sunt cele care fac sistemul sa functioneze in modul dorit. Aceasta aplicatie beneficiaza de urmatorele aspecte non-functionale:

- **Mentenanta:** sistemul dezvoltat ofera o viziune clara asupra functionalitatii, astfel fiind usor de imbunatit sau actualizat conform cerintelor care apar;
- **Simplitate:** utilizarea sistemului este facila si chiar un necunoscator al domeniului poate sa realizeze cu usurinta care sunt functionalitatile aplicatiei si cum trebuie folosite acestea;
- **Disponibilitate:** sistemul poate fi folosit la orice ora din zi sau zi din saptamana, constrangerile de timp fiind eliminate;
- **Scalabilitatea:** aplicatia suporta scalabilizare de ordin orizontal, respectiv marirea numarului de clienti, un numar mai ridicat de utilizatori care sa

utilizeze aplicatia putand fi suportat cu succes; de asemenea, scalarea de ordin vertical poate fi realizata de catre programatori, adaugand facilitati partii de server si stocarii de date, precum si trecerea la o solutie care sa contina mai multe servere de tip cluster;

- **Performanta:** timpul de procesare reprezinta unul dintre punctele de vedere luate in considerare in vederea masurarii performantei, precum si rata de procesare; sistemul propus raspunde in timpi insesizabili utilizatorului, maxim doua secunde;
- **Utilizabilitatea:** reprezinta nivelul de dificultate pentru utilziator in vederea utilizarii; utilizatorul va reusi sa utilizeze aplicatia fara probleme daca sunt respectate anumite cerinte minime de utilizare;
- **Securitatea:** comunicarea intre aplicatiile client si server este realizata folosind protocolul https, iar parolele utilizatorilor sunt de asemenea hashuite, facand astfel imposibila coruperea informatiilor sau interceptarea comunicarii intre client si server;
- **Audit:** in urma activatilor pot fi aflate cu usurinta in urma rapoartelor oferite, precum si a istoricului facturilor si comenzilor in format PDF;

4.2. Cazuri de utilizare

Cazurile de utilizare reprezinta tipurile de actori acceptati de aplicatie si cazurile de folosinta pentru fiecare tip de utilizatori. Analiza diagramelor de utilizare are sarcina de a prezenta o imagine generala asupra utilizarii aplicatiei si a intregii aplicatiei.

Pentru o buna ilustrare a proceselor din cadrul unei aplicatii sunt folosite diferite tipuri de diagrame precum cele de secventa, tip flowchart sau cele dedicate cazurilor de utilizare, cum va fi prezentat in cele ce urmeaza.[6]

4.2.1. Actori

- Vanzator: vanzatorul este utilizatorul cu cele mai putine drepturi de uz ale aplicatiei, insa fara a-i fi impactate activitati;
- Gestionar: are mai multe facilitati puse la dispozitie fata de vanzator, avand si un rol mai important;
- Administrator: in plus fata de ceea ce poate realiza un gestionar este gestionarea tuturor operatorilor din sistem.

4.2.2. Diagrame cazuri de utilizare

Diagramele pentru cazurile de utilizare prezinta activitatile pe care le poate efectua fiecare actor.

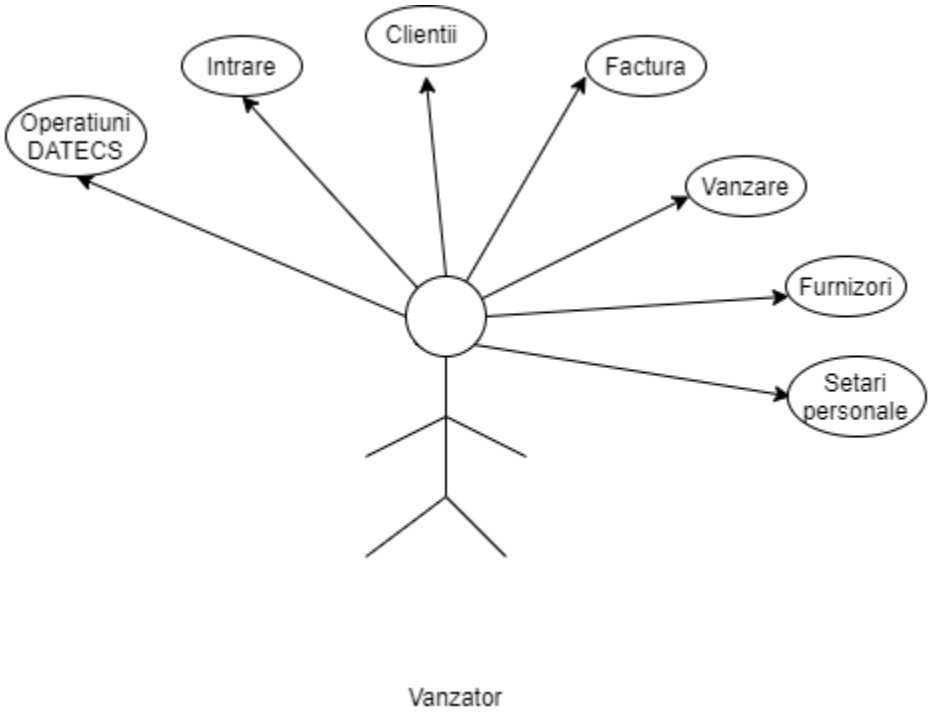

Figura 4.1 – Diagrama cazuri utilizare utilizator tip vanzator

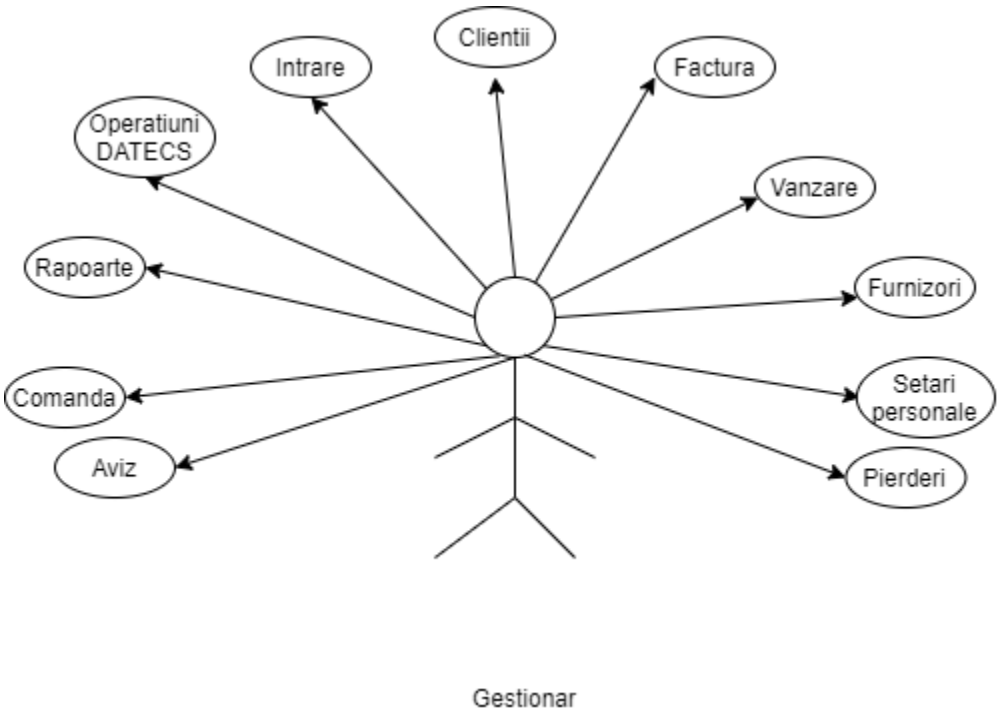

Figura 4.2 – Diagrama cazuri utilizare utilizator tip gestionar

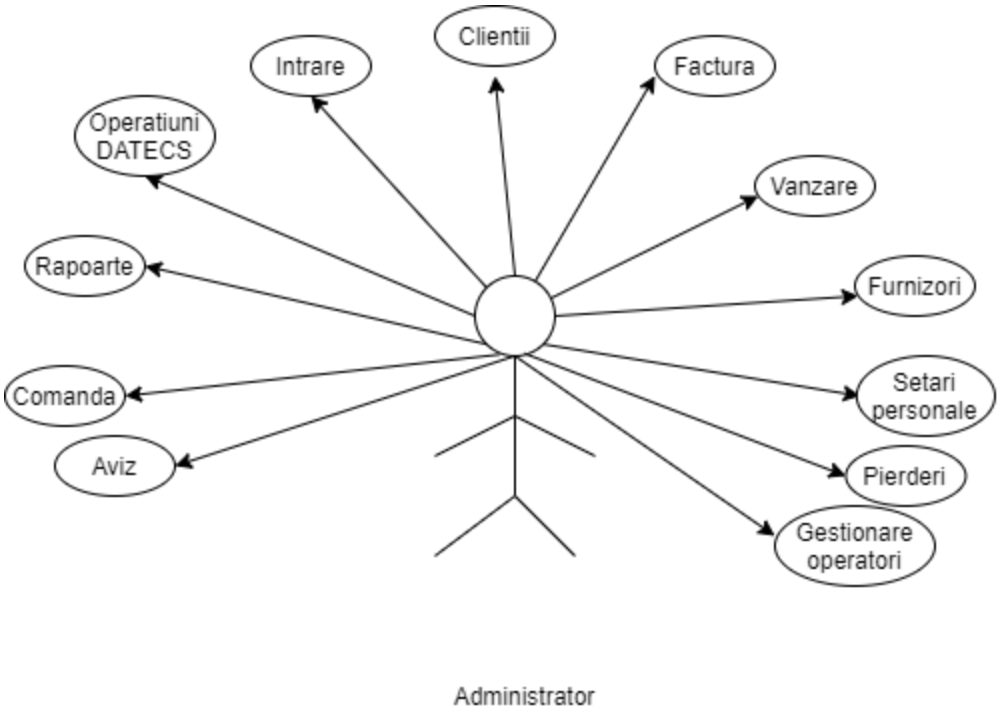

Figura 4.3 – Diagrama cazuri utilizare utilizator tip administrator

4.2.3. Descrierea cazurilor de utilizare

- Caz utilizare 1

Nume caz de utilizare: vanzare
Actor: vanzator; gestionar; admin
Descriere: acest caz de vanzare este acela de a facilita vanzarea efectuata de operator
Preconditii: utilizator este logat in aplicatie si exista produse in stoc
Postconditii: Operatorul a introdus sumele pentru fiecare tip de plata: cash, card, tichete de vanzare; selecteaza modul de finalizare: simpla sau cu factura
Scenariu principal 1:

1. Cautarea unei facturi emise anterior pe baza datei calendaristice.
2. Selectia facturii emise anterior in vederea crearii vanzarii.
3. Nu pot fi modificate cantitatile sau sterse produse din cos.
4. Finalizarea vanzarii prin specificarea metodei de plata.
5. Generarea fisierului .inp si imprimarea bonului.

Scenariu principal 2:

1. Operatorul introduce codul de bare cu ajutorul scanner-ului pentru coduri de bare
2. Selecteaza unul dintre produsele afisate in lista; motivul pentru care este necesara selectia este datorat faptului ca poate exista in stoc un produs care sa aiba diferite preturi
3. Poate fi modificata cantitatea vanduta, iar operatorulva fi avertizat in cazul in care este introdusa o cantitate indisponibila

4. Poate fi sters un produs din vanzare
5. In pasul de finalizare, este introdusa suma platita in cash, card si tichete de masa.
6. In cazul in care suma achitata de client este mai mare decat necesar, operatorul va fi notificat in legatura cu restul datorat.
7. Este apasat butonul de finalizare simpla sau cu factura, caz in care este generata automat si o factura.
8. Generarea fiserului .inp si imprimarea bonului.

Scenariu alternativ 1 – scenariu principal 1:

1. Nu este gasita nici o factura in ziua respectiva
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2 – scenariu principal 2:

1. Nu este gasit nici un produs pe baza codului de bare
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 3 – scenariu principal 2:

1. Cantitatea specificata pentru vanzare este indisponibila
2. Operatorul este notificat printr-un mesaj de eroare, iar in dreptul cantitatii este pusa valoarea maxima disponibila

Scenariu alternativ 4 – scenariu principal 1&2:

1. Valorile modalitatilor de plata nu insumeaza minimul necesar de plata
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 2

Nume caz de utilizare: factura
Actor: vanzator; gestionar; admin
Descriere: cazul acesta este dedicat procesului de facturare
Preconditii: utilizatorul este logat in aplicatie si clientul este adaugat in sistem
Postconditii: factura este generata in format PDF
Scenariu principal:

1. Selectarea clientului
2. Adaugare produs in cazul in care nu exista in stoc cu nume, unitate masura si cantitatea
3. Cautare produs dupa nume
4. Selectia produsului din lista
5. Poate fi modificata cantitatea ce urmeaza sa fie facturata, operatorul fiind atentionat in care cantitatea introdusa este indisponibila
6. Stergerea unui produs din factura
7. Finalizare factura cu generare fisier in format PDF

Scenariu alternativ 1:

1. Nu este gasit nici un produs pe baza codului de bare
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2:

1. Cantitatea specificata pentru factura este indisponibila
2. Operatorul este notificat printr-un mesaj de eroare, iar in dreptul cantitatii este pusa valoarea maxima disponibila

- Caz utilizare 3

Nume caz de utilizare: comanda
Actor: gestionar; admin
Descriere: cazul acesta este dedicat procesului de
Preconditii: utilizatorul este logat in aplicatie comanda si furnizorul este adaugat in sistem
Postconditii: comanda este finalizata, documentul PDF creat, iar mail-ul trimis catre furnizor daca operatorul doreste
Scenariu principal:

1. Selectarea furnizorului si a magazinului destinatie
2. Adaugare produs in cazul in care nu exista in stoc cu nume, unitate masura si cantitatea
3. Cautare produs dupa nume
4. Selectia produsului din lista
5. Cautarea tuturor produselor expediate de catre furnizorul respectiv
6. Selectia produsului din lista
7. Poate fi modificata cantitatea ce doreste sa fie comandata
8. Stergerea unui produs din comanda
9. Finalizare comanda cu generare de PDF si trimitere de email catre furnizor daca operatorul doreste

Scenariu alternativ 1:

1. Nu este gasit nici un produs pe baza numelui
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2:

1. Nu este gasit nici un produs pe baza furnizorului
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 4

Nume caz de utilizare: intrare
Actor: vanzator; gestionar; admin
Descriere: cazul acesta este dedicat procesului de intrare in stoc
Preconditii: utilizatorul este logat in aplicatie si furnizorul este adaugat in sistem
Postconditii: intrarea este finalizata
Scenariu principal:

1. Selectarea furnizorului
2. Cautare comanda pe baza datei calendaristice in vederea efectuarii intrarii pe baza comenzii
3. Selectarea comenzii
4. Adaugare produs in cazul in care nu exista in stoc cu cod de bare, nume, unitate masura, cota TVA, pret achizitie, pret vanzare si cantitate
5. Cautare produs dupa nume
6. Selectie produs din lista
7. Cautare produs dupa furnizor
8. Selectia produs din lista

9. Modificarea codului de bare, cotei TVA, pretului de achizitie, pretului vanzarii si a cantitatii
10. Stergerea unui produs din intrare
11. Finalizare intrarii prin apasarea butonului

Scenariu alternativ 1:

1. Nu este gasita nici o comanda pe baza datei calendaristice
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2:

1. Nu este gasit nici un produs pe baza numelui
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 3:

1. Nu este gasit nici un produs pe baza furnizorului
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 5

Nume caz de utilizare: creare aviz
Actor: gestionar; admin
Descriere: cazul acesta este dedicat procesului de creare a avizului
Preconditii: utilizatorul este logat in aplicatie
Postconditii: avizul este creat cu succes
Scenariu principal:

1. Selectarea furnizorului
2. Selectare magazin destinatie
3. Cautare produs dupa nume
4. Selectarea produsului din lista
5. Modificarea cantitatii ce urmeaza a fi livrate
6. Stergere produs din aviz
7. Finalizare aviz

Scenariu alternativ 1:

1. Nu este gasit nici un produs pe baza numelui
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2:

1. Cantitatea introdusa pentru aviz este indisponibila
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 6

Nume caz de utilizare: primire aviz
Actor: gestionar; admin
Descriere: cazul acesta este dedicat procesului de primire a avizului
Preconditii: utilizatorul este logat in aplicatie, avizul a fost creat anterior si primirea marfii se face din magazinul destinatie specificat in aviz
Postconditii: receptia avizului este finalizata cu succes
Scenariu principal:

1. Cautarea avizului pe baza datei calendaristice
2. Selectare aviz
3. Finalizare primire aviz

Scenariu alternativ:
1. Nu este gasit nici un aviz pe baza datei calendaristice
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 7

Nume caz de utilizare: pierdere
Actor: gestionar; admin
Descriere: cazul acesta este dedicat procesului de crearea unei pierderi
Preconditii: utilizatorul este logat in aplicatie si produsul exista in stoc
Postconditii: avizul este receptionat cu succes
Scenariu principal:
1. Cautarea produsului pe baza numelui
2. Selectia produsului din lista
3. Introducerea cantitatii
4. Finalizare pierderii

Scenariu alternativ 1:
1. Nu este gasit nici un produs pe baza numelui
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2:
1. Cantitatea introdusa pentru inregistrarea pierderii este mai mare decat stocul real
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 8

Nume caz de utilizare: rapoarte vanzari
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor de vanzare
Preconditii: utilizatorul este logat in aplicatie si exista vanzari
Postconditii: operatorul poate vizualiza vanzarile si le poate exporta
Scenariu principal:
1. Cautarea vanzarilor pe baza datei calendaristice, a magazinului si a operatorului care a efectuat vanzari
2. Selectia unei vanzari din lista
3. Vizualizarea produselor vandute in vanzarea respectiva
4. Exportul vanzarilor conform cautarii in format .xlsx

Scenariu alternativ:
1. Nu este gasita nici o vanzare pe baza parametrilor specificati
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 9

Nume caz de utilizare: rapoarte facturi
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor de facturi
Preconditii: utilizatorul este logat in aplicatie si exista facturi
Postconditii: operatorul poate vizualiza facturile si le poate exporta

Scenariu principal:
1. Cautarea facturilor pe baza datei calendaristice, a magazinului, a operatorului si a clientului care a efectuat facturi
2. Selectia unei facturi din lista
3. Vizualizarea produselor facturate in factura respectiva
5. Exportul facturilor conform cautarii in format .xlsx

Scenariu alternativ:
1. Nu este gasita nici o factura pe baza parametrilor specificati
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 10

Nume caz de utilizare: rapoarte comenzi
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor de comenzi
Preconditii: utilizatorul este logat in aplicatie si exista comenzi efectuate
Postconditii: operatorul poate vizualiza comenzile si le poate exporta
Scenariu principal:
1. Cautarea comenzilor pe baza datei calendaristice, a magazinului, a furnizorului si a operatorului care a efectuat facturi
2. Selectia unei comenzi din lista
3. Vizualizarea produselor comandate din comanda respectiva
4. Exportul comenzilor conform cautarii in format .xlsx

Scenariu alternativ:
1. Nu este gasita nici o comanda pe baza parametrilor specificati
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 11

Nume caz de utilizare: rapoarte intrari
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor de intrari
Preconditii: utilizatorul este logat in aplicatie si exista intrari
Postconditii: operatorul poate vizualiza intrarile si le poate exporta
Scenariu principal:
1. Cautarea intrarilor pe baza datei calendaristice, a magazinului, a furnizorului si a operatorului care a efectuat intrari
2. Selectia unei intrari din lista
3. Vizualizarea produselor din intrarea respectiva
4. Exportul intrarilor conform cautarii in format .xlsx

Scenariu alternativ:
1. Nu este gasita nici o intrare pe baza parametrilor specificati
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 12

Nume caz de utilizare: rapoarte avize
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor de avize
Preconditii: utilizatorul este logat in aplicatie si exista avize
Postconditii: operatorul poate vizualiza avizele si le poate exporta
Scenariu principal:
1. Cautarea avizelor pe baza datei calendaristice, a magazinului sursa, a magazinului destinatie si a operatorului care a efectuat avize
2. Selectia unui aviz din lista
3. Vizualizarea produselor din avizul respectiv
4. Exportul avizelor conform cautarii in format .xlsx

Scenariu alternativ:
1. Nu este gasit nici un aviz pe baza parametrilor specificati
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 13

Nume caz de utilizare: rapoarte pierderi
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor pierderilor
Preconditii: utilizatorul este logat in aplicatie si exista pierderi
Postconditii: operatorul poate vizualiza pierderile si le poate exporta
Scenariu principal:
1. Cautarea pierderilor pe baza datei calendaristice si a operatorului care a efectuat adaugarea pierderilor
2. Cautarea produselor din sectiunea de pierderi pe baza numelui
3. Exportul pierderilor conform cautarii in format .xlsx

Scenariu alternativ:
1. Nu este gasit nici o pierdere pe baza parametrilor specificati
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 14

Nume caz de utilizare: rapoarte stoc
Actor: gestionar; admin
Descriere: cazul acesta este dedicat rapoartelor referitor la stocuri
Preconditii: utilizatorul este logat in aplicatie si exista produse in stoc
Postconditii: operatorul poate vizualiza stocurile existente
Scenariu principal:
1. Cautarea produselor din stoc pe baza numelui
2. Cautarea produselor din stoc pe baza magazinului
3. Cautarea produselor din stoc pe baza furnizorului
3. Exportul stocurilor conform cautarii in format .xlsx

Scenariu alternativ 1:
1. Nu este gasit nici un produs pe baza numelui
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 2:
1. Nu este gasit nici un produs pe baza magazinului
2. Operatorul este notificat printr-un mesaj de eroare

Scenariu alternativ 3:
1. Nu este gasit nici un produs pe baza furnizorului
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 15

Nume caz de utilizare: intrare si iesire casa de marcat
Actor: vanzator; gestionar; admin
Descriere: cazul acesta este dedicat intrarilor si iesirilor din sertarul casei de marcat
Preconditii: utilizatorul este logat in aplicatie
Postconditii: operatorul efectueaza iesiri sau intrari din sertarul casei de marcat
Scenariu principal:
1. Introducerea valorii
2. Selectarea operatiunii: intrare sau iesire

- Caz utilizare 16

Nume caz de utilizare: clienti
Actor: vanzator; gestionar; admin
Descriere: cazul acesta este dedicat operatiunilor dedicate clientilor
Preconditii: utilizatorul este logat in aplicatie
Postconditii: operatorul vizualizeaza clientii si poate exporta informatiile referitoare la clienti
Scenariu principal:
1. Adaugarea unui client prin introducerea numelui, codului unic de identificare fiscala, adresei, adresei de email, numarului de telefon, bancii si contului
2. Posiblitatea modificarii adresei, adresei de email, telefonului, bancii si contului
3. Exportul clientilor in format .xlsx

- Caz utilizare 17

Nume caz de utilizare: furnizori
Actor: vanzator; gestionar; admin
Descriere: cazul acesta este dedicat operatiunilor dedicate furnizorilor
Preconditii: utilizatorul este logat in aplicatie
Postconditii: operatorul vizualizeaza furnizorii si poate exporta informatiile referitoare la furnizori
Scenariu principal:
1. Adaugarea unui furnizori prin introducerea numelui, codului unic de identificare fiscala, adresei, adresei de email, numarului de telefon, bancii si contului
2. Modificariea adresei, adresei de email, telefonului, bancii si contului
3. Exportul clientilor in format .xlsx

- Caz utilizare 18

Nume caz de utilizare: setari personale
Actor: vanzator; gestionar; admin
Descriere: cazul acesta este dedicat operatiunilor dedicate operatorilor asupra contului personal
Preconditii: utilizatorul este logat in aplicatie
Postconditii: operatorul efectueaza modificari asupra contului personal
Scenariu principal:
1. Modificarea informatiilor existente precum nume, adresa email, telefon sau adresa.
2. Posiblitatea resetarii parolei cu una dorite de el

Scenariu alternativ:
1. Parola actuala introdusa nu este cea corecta
2. Operatorul este notificat printr-un mesaj de eroare

- Caz utilizare 19

Nume caz de utilizare: gestionare operatori
Actor: admin
Descriere: cazul acesta este dedicat administratorilor in vederea operatiunilor asupra celorlalti operatori
Preconditii: utilizatorul este logat in aplicatie
Postconditii: utilizatorul efectueaza modificari asupra celorlalti utilizatori
Scenariu principal:
1. Adaugarea unui nou operator specificand numele, username-ul, adresa, adresa de email, numarul de telefon si rolul
2. Modificarea rolului unui utilizator
3. Modificarea statusului unui utilizator din activ in inactiv si viceversa
4. Resetarea parolei unui utilizator la parola predefinita “parola”

4.3. Tehnologii folosite

4.3.1. Spring

Spring este unul dintre cele mai populare framework-uri Java pentru dezvoltarea aplicatiilor, avand avantajul de a fi open-source. Motivele pentru care am optat pentru utilizarea acestei tehnologii sunt popularitatea ei, usurinta in dezvoltare precum si multitudinea de resurse disponibile.

Printre beneficiile Spring se numara:

- organizarea modulara, chiar daca numarul de pachete este unul substantial, activitatea poate fi concentrata pe cele importante, celelalte fiind ignorate
- nu este totul redefinit, in schimb Spring face uz de tehnologiile deja existente precum framework-uri ORM
- framework-ul web Spring este unul bine construit, oferind o alternativa grozava fata de alte framework-uri precum Struts
- Spring ofera o interfata consistenta de manageriere a tranzactiilor [7]

Spring este modular si ofera calitatea de alege ce module sunt necesare, fara a fi necesara utilizarea tuturor modulelor.

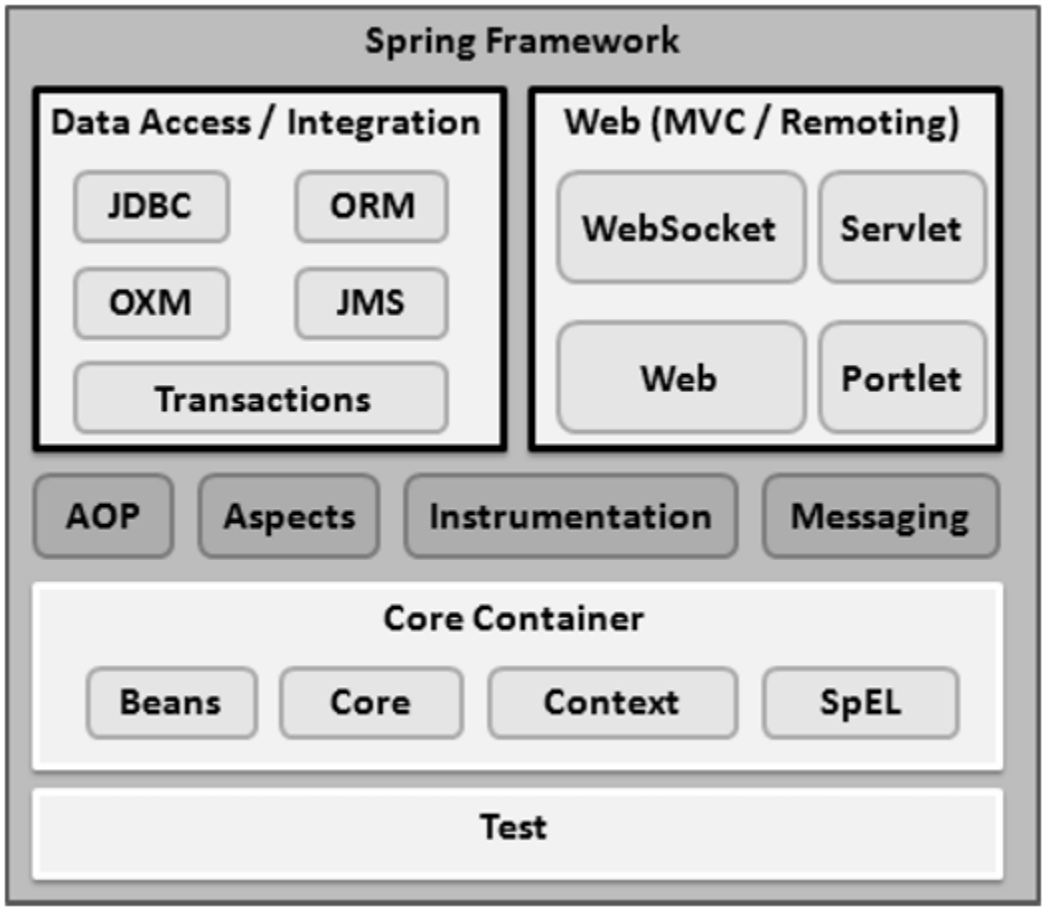

Figura 4.4 – Module Spring

ORM este o modalitate de programare ce ofera un acces si o manipulare facila a obiectelor, fara a fi necesara cunoasterea surselor obiectelor. Aceasta modalitate a aparut datorita necesitatii de a depasi diferentele de paradigma dintre modelul orientat pe obiecte(cel folosit de limbajele de programare high-level) si modelul relational(folosit de bazele de date).[10] Unul dintre cele mai populare ORM-uri este Hibernate.

Spring Data este un proiect care are la randul sau alte subproiecte sau module, avand drept scop abstractii si metode uniforme pentru layer-ul de Data Access, suportand o gama larga de baze de date si medii de stocare

Spring Data JPA este unul din multiplele proiecte Spring Data si cauta sa ofera consistenta in accesarea datelor in bazele de date relationale. Cu Spring Data JPA nu este necesara scrierea de scripturi SQL. Cu ajutorul implementarii JPA deja existente este

[10] https://www.todaysoftmag.ro/article/73/analiza-mecanismului-object-relational-mapping-orm-cu-exemplificari-hibernate

permisa utilizarea obiectelor Entity si a managerului de entitati care este responsabil de persistenta si returnarea entitatilor din baza de date.[11]

De asemenea, Spring ofera o utilizare usoara a Dependency Injection, specificand cu ajutorul adnotarii „@Component" ce componente trebuie utilizate, iar cu „@Autowired" unde trebuie utilizate pe baza tipului.[12]

Spring Boot este un microframework construit folosind Spring care ofera prin doar cateva clickuri o aplicatie functionala, singurele activitati ale utilizatorului fiind selectarea modulelor dorite in aplicatie. De asemenea, acesta reduce timpul necesar dezvoltarii prin asistenta la configurare, usurinta la testare si evitarea scrierii de cod repetitiv tip „boilerplate". Problema dependintelor este una des intalnita, iar Spring Boot vine in ajutor [13]oferind un set de dependinte automat incluse in proiect precum: spring-boot-starter-data-jpa pentru JPA si acces la baze de date, spring-boot-starter-security pentru partea de securitate, spring-boot-starter-web pentru a scrie endpoint-uri REST, spring-boot-starter-test pentru scrierea de teste.[14]

4.3.2. JavaFX

JavaFX este un framework de dezvoltare construita de Oracle care faciliteaza construirea de aplicatii desktop sau Rich Internet Applications(RIA) care pot fi accesate de o multitudine de dispozitive.[8] Scopul JavaFX este acela de inlocui Swing ca framework pentru dezvoltarea interfetelor grafice, oferind in acelasi timp mai multe functionalitati decat Swing. JavaFX a fost selectata pentru utilizare intrucat ofera multiple facilitati in dezvoltarea aplicatiilor, precum si o vedere clara si separata intre ceea ce reprezinta elementele de ordin vizual si modul in care functioneaza acestea.

Facilitati importante ale JavaFX sunt:

- JavaFX foloseste FXML, un limbaj similar cu HTML pentru descrierea usoara a interfetelor grafice
- Stilizarea interfetelor grafice cu ajutorul CSS
- Este oferita aplicatia Scene Builder pentru un design al interfetelor in mod drag&drop
- Disponibilitatea controller-elor pentru a descrie activitatea elementelor grafice descrise in FXML
- Interoperabilitate cu Swing
- Librarie pentru grafice 2D sau 3D
- Pipeline grafic[15]
- Redare audio sau video
- Web view pentru afisarea de pagini web in interiorul aplicatiei

JavaFX foloseste asa numitele stage-uri care corespund ferestrelor, fiecare stage avand un scene atasat in care vor fi afisate elementele descrise.

[11] https://www.amitph.com/jpa-and-spring-data-jpa/#1_Spring_Data

[12] https://dzone.com/articles/spring-boot-vs-spring-mvc-vs-spring-how-do-they-compare

[13] https://www.javatpoint.com/javafx-tutorial

[14]https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm#:~:text=Spring%20Boot%20is%20an%20open,and%20production%20ready%20spring%20applications.

[15] https://www.tutorialspoint.com/javafx/javafx_overview.htm

Framework-ul ofera o multitudine de elemente care pot fi adaugate in scene precum: butoane, campuri text sau parola, zone text, label-uri, checkbox-uri sau combobox-uri, meniuri, spinner-e sau slider-e.

De asemenea exista si layout-uri, care sunt componente care contin la randul lor alte componente in interiorul lor precum: grupuri, regiuni, Hbox-uri care isi expune componentele continute sub forma orizontala sau Vbox-uri care si le expune sub forma verticala.[16]

Exista si conceptul de Scene Graph care este o structura de tip arbore reprezentand continutul unei scene. Totul pleaca de la un nod sursa, copii sai fiind alte noduri care la randul lor pot avea sau nu copii. Pentru a permite extinderea arborelui este necesar ca nodul sa fie de tip grup, regiune sau webview. Nodurile finale, frunza, sunt elementele simple precum butoantele sau campurile text.[17]

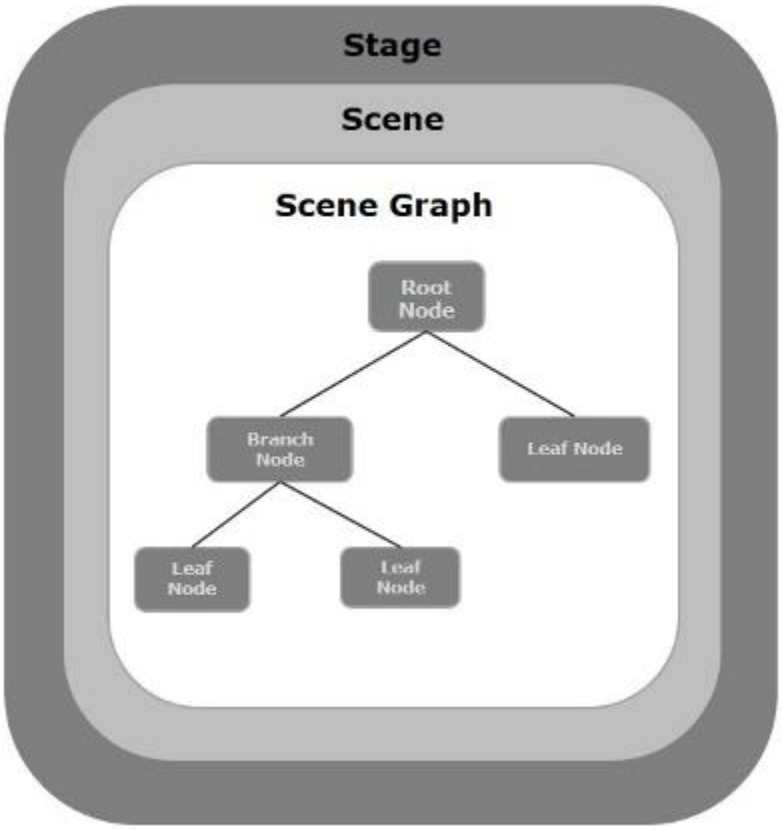

Figura 4.5 – Scene Graph in JavaFX

4.3.3. MySql

O baza de date este o colectie de date structurate. In acest loc datele sunt stocate si organizate. MySql este un sistem de manageriere a bazei de date in mod relational(RDBMS – Relational Database Management System). [9]

[16] http://tutorials.jenkov.com/javafx/overview.html
[17] https://www.tutorialspoint.com/javafx/javafx_application.htm

MySql a fost dezvoltat de o companie suedeza, MySql AB in 1994, companie achizitionata de de Sun Microsystems in 2008, companie care la randul ei a fost achizitionata de Oracle in 2010. Aceasta tehnologie a fost selectata pentru utilizare intrucat este gratuita, ofera performante ridicate, fiabilitate si este usor de folosit.

MySql si Sql nu sunt una si aceeasi, MySql fiind unul dintre cele mai populare sisteme dedicate bazelor de date, iar Sql reprezinta limbajul specific folosit, respectiv Structured Query Language.

Prin Sql se poate realiza interogarea unei baze de date, manipularea (adaugare, stergere, modificare), constituirea (definirea tipurilor datelor sau a relatiilor) precum si modificarea accesului la baza de date.

Fiecare client poate efectua request-uri catre server-ul bazei de date, iar server-ul va produce raspunsul asteptat. Utilizatorul scrie un query care este trimis catre server, procesat de catre server, raspunsul fiind trimis inapoi catre client.

Printre cele mai populare instrumente pentru client, tip GUI, se numara MySqlWorkbench, SequelPro sau DBVisualizer.

4.3.4. Maven

Maven este o puternica ustensila folosita in proiectele Java cu scopul de a automatiza tot ce tine de construirea unui proiect software.[18]

Punctul de interes pentru Maven este conceputul POM(Project Object Model). Un fisier .pom este o reprezentare XML a resurselor proiectului precum cod sursa, cod test, dependinte, etc. In acest fisier exista referinte catre toate acestea.

Unul dintre principalele scopuri ale Maven este acela de a verifica dependintele necesare rularii proiectului. Dependintele reprezinta fisiere de tip JAR(librarii Java) care sunt utilizate de catre proiect. Maven verifica daca acestea exista in depozitul local Maven, daca nu acestea sunt descarcate automat.

De asemnea, Maven ofera impartirea procesului de constructie a proiectului in cicluri, etape si scopuri. Astefel, fiecare ciclu contine o serie de etape, iar fiecare etapa la randul ei contine o serie de scopuri. Exista si posiblitatea crearii profilelor in cazul in care un proiect este necesar sa fie construit in moduri distincte, fiecare profil avand un mod.

Decizia de utiliza acest instrument a fost data de faptul ca inlesneste integrarea modulelor externe in contextul proiectului prezent, dificultatea procesului de configurare fiind foarte usoara.

4.3.5. Format fișier trimis spre casa de marcat

Pentru procesarea de catre casa de marcat a sarcinii, este necesar ca fisierul transmis sa fie intr-un format special. [19]

Structura unei comenzi din fisierul de intrare este:

<C>,<E>,<de serviciu>,<parametrii>

[18] http://tutorials.jenkov.com/maven/maven-tutorial.html

[19] http://www.aparaturafiscala.ro/design/pdf/Descrierea%20comenzilor%20pentru%20driverele%20de%20comunicatie.pdf

unde <C> reprezinta codul comenzii, <E> numarul logic al casei de marcat, iar <de serviciu> are formatul <____,_,___> avand scopul de a inregistra rezultatul operatiunii. Parametrii sunt diferiti pentru fiecare tip de comanda.

Tipurile de comenzi oferite sunt:

- Vanzarea unui articol - S
- Tiparirea unui text nefiscal - P
- Inchiderea bonului si stabilirea modului de plata – T
- Majorare/reducere procentuala - C
- Introducere/retragere sume din sertar – I
- Obtinerea numarului bonului fiscal – N
- Generarea raportului X, Z – A.

Pentru vanzare, formatul parametrilor este urmatorul:

<nume>;<pret>;<cantitate>;<dep>;<ga>;<gt>;0;0;

Numele este dedicat denumirii produsului, pretul si cantitatea sunt folosite evident pentru valoarea produsului si cantitatea vanduta, dep reprezinta departamentul din care face parte produsul, ga numarul grupei de articole, gt numarul grupei de taxe, iar ultimele doua pozitii sunt campuri rezervate cu valoare obligatorie „0”.

Pentru imprimarea unui text nefiscal vor fi doar cinci parametrii, reprezentand 5 randuri care vor fi imprimate pe bon.

Pentru finalizarea unui bon parametrii au formatul:

<cod>;<suma>;;;;;

<cod> reprezinta tipul comenzii putand primi una dintre urmatoarele valori:

0 – plata cu numerar
1 – plata cu tichet
2 – plata cu card
4 – subtotal
8 – plata in valuta alternativa si restul in valuta de baza
9 – plata in valuta alternativa si restul in valuta alternativa.

Capitolul 5. Proiectare de Detaliu și Implementare

Capitolul este dedicat prezentarii arhitecturii conceptuale a sistemului, diagramei bazei de date, diagramelor de pachete si clase. Fiecare componenta si modul vor fi descrise in detaliu pe parcursul capitolului

5.1. Arhitectura sistemului

Sistemul este impartit in doua aplicatii distincte, conform arhitecturii client-server. Aceasta impartire ofera atat o buna separare a partii de front-end fata de cea de back-end, cat si o usoara scalabalitate de ordin orizontal prin cresterea numarul de clienti. Front-end-ul are drept sarcina afisarea informatiilor catre utilizatorul aplicatiei, organizarea functionalitatilor sub o forma vizuala intuitiva si comunicarea cu back-end-ul. Back-end-ul, de cealalta parte, are drept sarcina preluarea request-urilor de la front-end, indeplinirea lor si transmiterea raspunsului inapoi catre front-end. Front-end-ul comunica cu back-end-ul cu ajutorul REST(REpresentational State Transfer).

REST este un stil arhitectural care permite comunicarea intre client si server prin protocolul HTTP sau HTTPS. Formatul mesajelor schimbate de cele doua parti poate fi printre altele JSON sau XML. De asemenea, REST folosind HTTP , foloseste pentru transmisia informatiilor metodele HTTP, precum GET, POST, PUT sau DELETE.

5.1.1. Back-end

Partea de back-end se ocupa cu receptia cerintelor de la front-end si procesarea lor. Aceasta este structurata in urmatoarele pachete:

- Controllers: in care sunt expusele controller-ele pentru fiecare tip de operatiune, metodele fiind adnotate cu tipul metodei, respectiv „@GetMapping”, „@PostMapping”, „@PutMapping” sau „@DeleteMapping”.
- Convertors: este pachetul dedicat convertirii obiectelor primite de la client, la obiecte care pot fi trimise spre baza de date, precum si viceversa
- DTOs: este pachetul dedicat obiectelor folosite in comunicarea cu partea de front-end
- Entities: este pachetul folosit pentru reprezentarea obiectelor din baza de date, fiind folosite adnotari pentru specificarea tabelei corespunzatoare din baza de date, specificarea denumirii coloanelor din tabela sau specificarea generararii id-ului obiectului ce urmeaza sa fie scris in baza de date
- Repositories: pachet dedicat repository-urilor, care sunt interfete in care vor fi specificate metodele ce pot fi efectuate, implementarea fiind generata automat de catre Spring Data
- Response: este pachetul folosit pentru construirea automata a mesajelor transmise catre front-end
- Services: este pachetul in care este efectuata procesarea datelor conform cerintei primite de la client, fie ca este vorba de scrierea informatiilor in

baza de date, aducerea informatiilor sau actualizarea lor, totul cu ajutorul repository-urilor.

- Utils: pachet pentru operatiile speciale, precum trimitearea de e-mail-uri, generarea de fisiere PDF, precum si pentru configurarile de securitate pentru a asigura o conexiune HTTPS

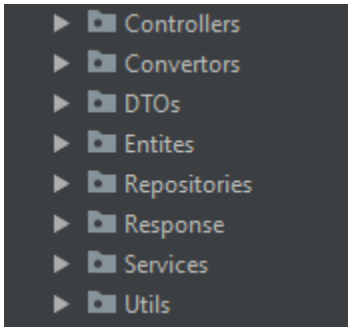

Figura 5.1 – Pachetele in aplicatia back-end

5.1.2. Front-end

Partea de front-end are drept scop, expunerea unei interfete grafice pe care sa o foloseasca utilizatorul, precum si comunicarea cu partea de back-end.

Structura acesteia va fi descrisa in cele ce urmeaza.

GUIs: este pachetul in care sunt fisiere FXML dedicate descrierii interfetelor grafice. Aceste fisiere au fost construite atat scriind cod propriu-zis, cat si folosind Scene Builder, pentru a reusi aranjarea elementelor intr-un mod placut vizual.

Controllers: este un pachet dedicat controller-elor, controller-e care efectueaza operatiile in urma interactiunii utilizatorului cu interfata grafica.

DTOs: pachet in care sunt descrise obiectele care vor fi trimise catre server sau asteptate de la acesta. De asemenea, fiecare clasa are metode de transformare a obiectelor JSON in obiecte conforme.

TableModels: in JavaFx, pentru utilizarea unui tabel este necesara folosirea unui obiect care sa aiba toate campurile ce trebuiesc afisate, care nu corespunde cu obiectele tip DTO, in consecinta fiind necesara utilizarea unor obiecte dedicate pentru aceste operatiuni.

Utils: este un pachet in care sunt clase cu un rol special. Una dintre acestea este afisarea mesajelor de informare sau eroare. Alta este cea de pregatire a apelurilor catre server, scopul ei fiind de a usura initierea de apeluri din obiectele de tip controller. Pentru efectuarea propriu-zisa a apelurilor este folosita clasa Executor, care usureaza efectuarea apelurilor in mod asincron. Datorita faptului ca fiecare fereastra are un controller dedicat, transmiterea variabilelor poate deveni problematica. In acest sens exista clasa Context, unde este folosit design pattern-ul Singleton pentru o utilizare eficienta, clasa in care sunt stocate anumite variabile care sunt necesare bunei functionari a aplicatiei precum utilizatorul logat sau magazinul curent.

Tot in pachetul Utils este si clasa ExcelExporters care se ocupa cu exportarea rapoartelor in format .xlsx din sectiunea Rapoarte a aplicatiei. De asemenea, este necesara o clasa care sa se ocupe cu generarea de fisiere care sa fie salvate in directorul prestabilit pentru imprimarea de bonuri de catre casa de marcat.

Pentru eficienta, odata reusita logarea in aplicatie, este incarcata fereastra principala, fereastra in care va fi incarcat o singura data meniul, iar pe parcursul utilizarii aplicatiei va fi schimbata doar restul ferestrei. Pentru indeplinerea acestui scop, este folosita clasa JFXUtils. Clasa JSONGetters are scopul de facilita primirea mesajelor de catre aplicatie de la server, in urma apelurilor. PasswordHash este clasa care se ocupa cu hashuirea parolelor utilizatorilor. SSLClientBuilder se ocupa cu incarcarea fisierului .p12 si pregatirea clientului SSL pentru comunicatia HTTPS.

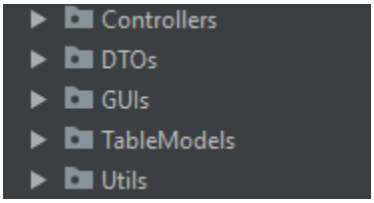

Figura 5.2 – Pachetele in aplicatia front-end

5.2. Diagramele sistemului

5.2.1. Diagrama generala a sistemului

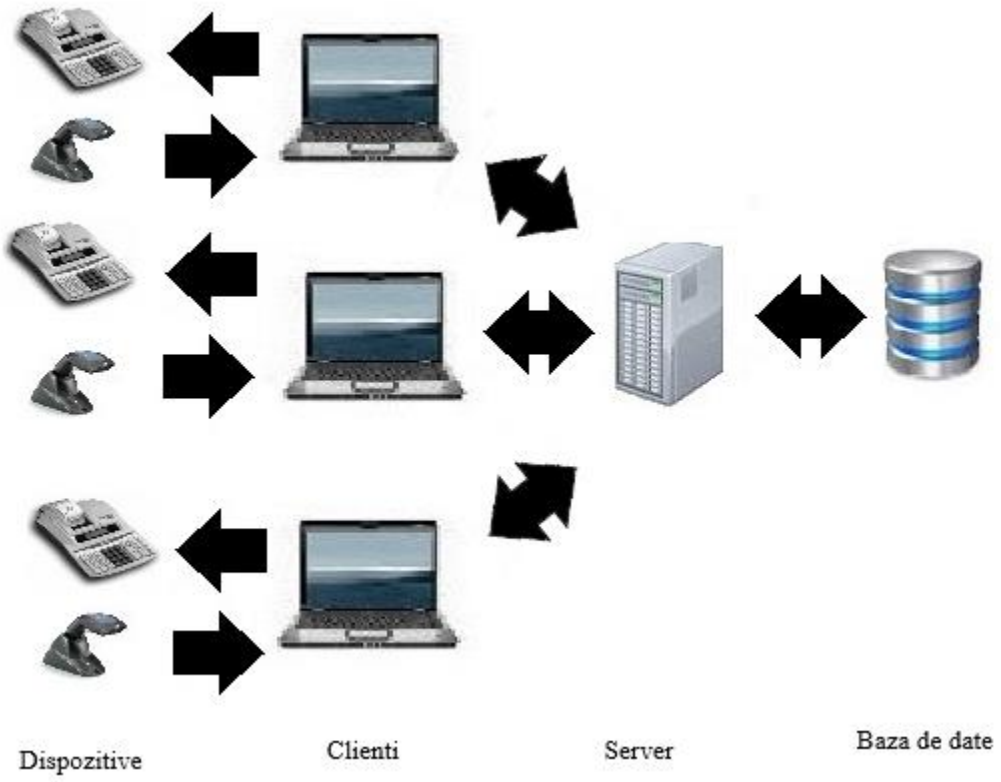

Figura 5.3 – Arhitectura generala a sistemului

Sistemul este impartit in doua aplicatii, tip client- server, putand exista mai multi clienti care sa comunice cu serverul. La randul lui serverul realizeaza conexiunea cu baza

de date. De asemenea, clientii realizeaza conexiunile cu casa de marcat si scanner-ul pentru coduri de bare.

5.2.2. Diagramele de pachete

Diagramele de pachete sunt o metoda de reprezentare a modului in care sunt grupate clasele in aplicatie, precum si a modului in care sunt folosite.

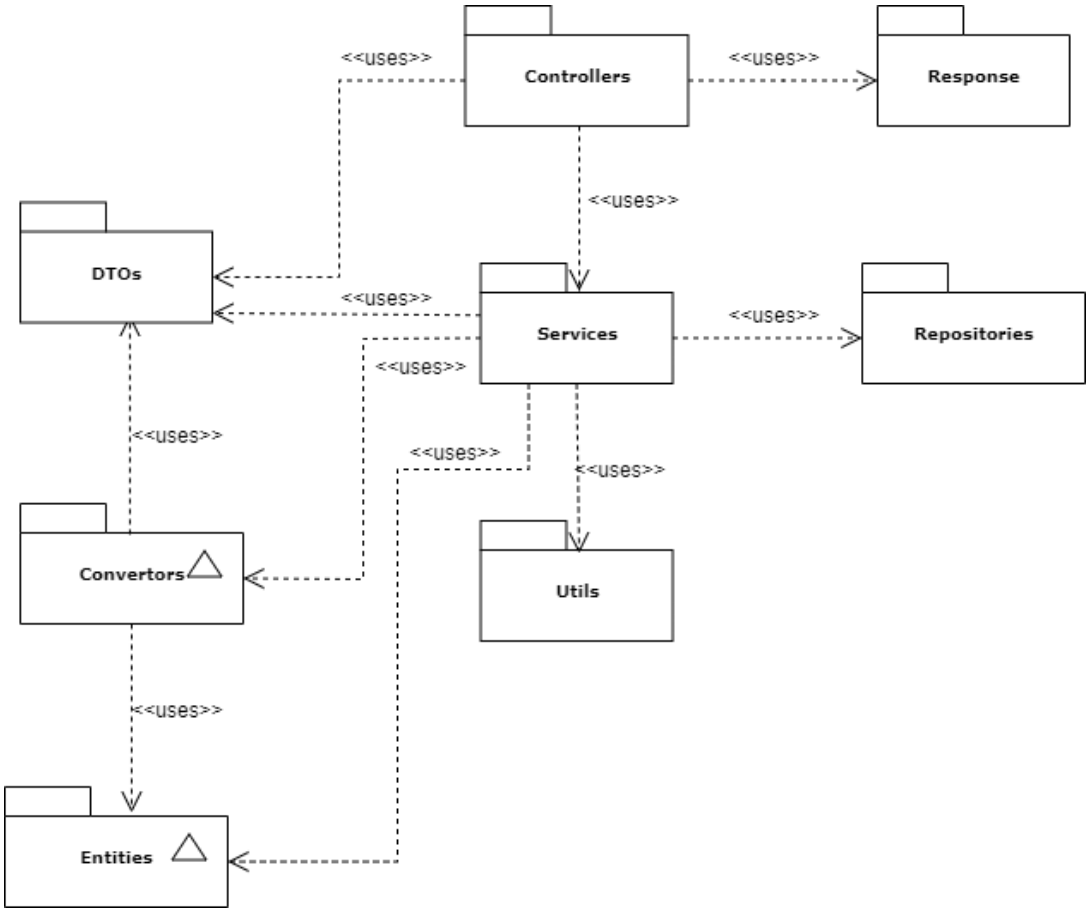

Figura 5.4 – Diagrama pachetelor pe partea de server

Dupa cum poate fi observat, in pachetul Services are loc cea mai multa activitate, fiind si pachetul care comunica cel mai mult cu celelalte pachete, fie ca este vorba de convertirea obiectelor, utilizarea repository-urilor sau a functionalitatilor din pachetul Utils.

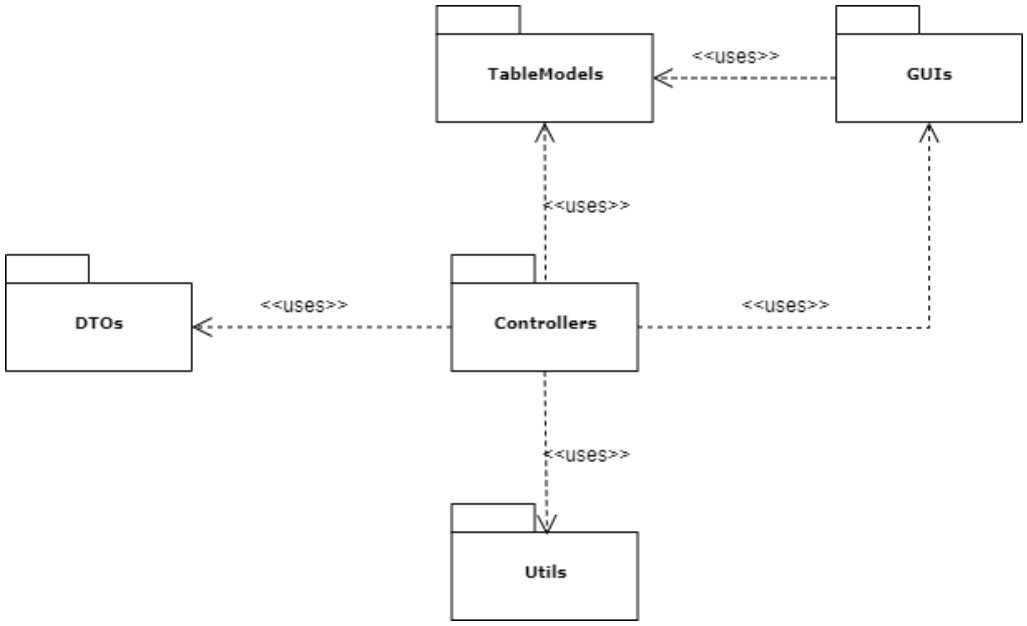

Figura 5.5 – Diagrama pachetelor pe partea de client

Pe partea de client, pachetul central este Controllers, intrucat acesta reprezinta elementul de procesare pentru interfata grafica, aici se construiesc obiectele DTO ce vor fi trimise catre server si tot de aici se initiaza activitatile din pachetul Utils.

5.3. Structura bazei de date

Datorita faptului ca baza de date este una complexa cu multe relatii intre tabele aceasta va fi prezentata in doua figuri alaturi de explicatiile aferente. In cele ce urmeaza vor fi prezentate tabelele si relatiile dintre ele conform diagramelor.

In figura 5.6 sunt prezentate tabelele care sunt folosite in cadrul functionatlitatilor de vanzare, facturare, operatii cu clientii sau modificare operatori. In figura 5.7 sunt prezentate tabelele care au relevanta in cadrul sectiunilor de comanda, intrare, operatii cu furnizorii, inregistrarea pierderilor sau activitatea cu avizele. Operatiunile de intrare si si iesire ale casei de marcat nu se reflecta si in baza de date. Pe baza tuturor tabelelor prezentate pot fi realizate rapoartele.

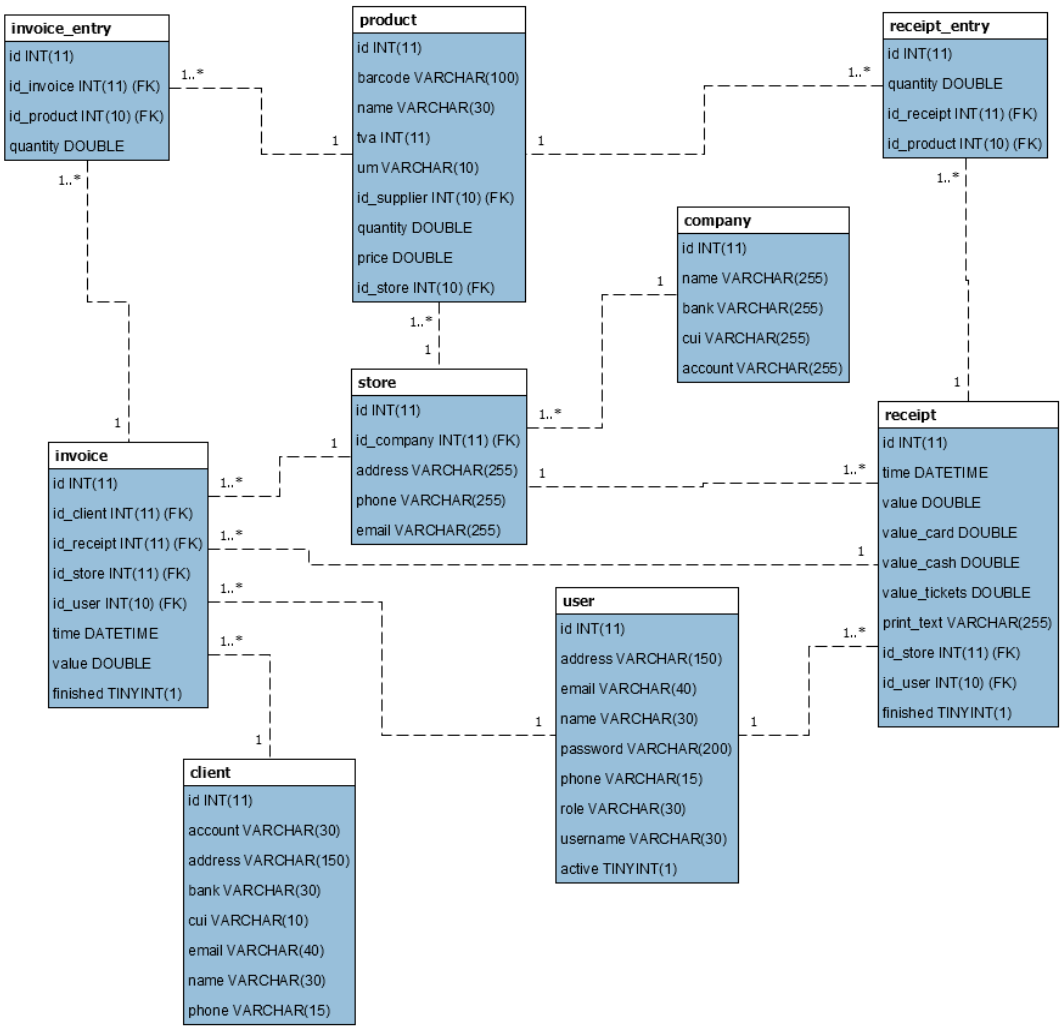

Figura 5.6 – Diagrama baza de date partea 1

Tabela User

Contine informatiile referitoare la utilizator:

- Id: cheie primara; int(11)
- Address: adresa utilizatoruluil; varchar(150)

- Email: adresa de email a utilizatorului, folosita in cadrul procesului de inregistrare in sistem, pe adresa de email fiind transmise credentialele; varchar(40)
- Name: numele utilizatorului; varchar(30)
- Password: parola hashuita a utilizatorului; varchar(200)
- Phone: numarul de telefon al utilizatorului; varchar(15)
- Role: rolul in cadrul sistemului: admin, vanzator sau gestionar; varchar(30)
- Username: folosit la logarea in aplicatie; varchar(30)
- Active: specifica daca utilizatorul este activ si are dreptul de a utiliza aplicatia; tinyint(1)

Tabela client:

Contine informatiile referitoare la client:

- Id: cheie primara; int(11)
- Account: contul bancar al clientului; varchar(30)
- Address: adresa clientului; varchar(150)
- Bank: banca folosita de client; varchar(30)
- Cui: Codul Unic de identificare al clientului; varchar(10)
- Email: adresa de email a clientului, in cazul in care utilizatorul aplicatiei doreste comunicarea cu clientul prin intermediul email-ului; varchar(40)
- Name: numele clientului; varchar(30)
- Phone: numarul de telefon al clientului; varchar(15)

Tabela company:

Contine informatiile referitoare la compania care utilizeaza aplicatia:

- Id: cheie primara; int(11)
- Name: numele companiei; varchar(255)
- Bank: banca folosita de companie; varchar(255)
- Cui: Codul Unic de identificare al companiei; varchar(255)
- Account: contul bancar al companiei; varchar(255)

Tabela store:

Contine informatiile referitoare la magazin:

- Id: cheie primara; int(11)
- Id_company: cheie straina, legata de tabela company; int (11)
- Address: adresa magazinului; varchar(255)
- Phone: numarul de telefon al magazinului; varchar(255)
- Email: adresa de email a magazinului, in cazul in care utilizatorul aplicatiei doreste comunicarea cu terti prin intermediul email-ului; varchar(255)

Tabela product:

Contine informatiile referitoare la produs:

- Id: cheie primara; int(11)

- Barcode: codul de bare al produsului, folosit la vanzare si facturare; varchar(100)
- Name: numele produsului; varchar(30)
- Tva: cota tva; int(11)
- Um: unitatea de masura a produslui; varchar(10)
- Id_supplier: cheie straina, legata de tabela supplier pentru a stii de la care furnizor este procurat produsul; int(11)
- Quantity: cantitatea disponibila in stoc; double
- Price: pretul produsului; double
- Id_store: cheie straina, legata de tabela store pentru a stii in care magazin se afla produsul; int(10)

Tabela receipt:

Contine informatiile referitoare la vanzare:

- Id: cheie primara; int(11)
- Time: momentul la care a avut loc vanzarea; datetime
- Value: valoarea totala a vanzarii; double
- Value_cash: valoarea platii in numerar; double
- Value_card: valoarea platii cu cardul; double
- Value_tickets: valoarea platii cu tichete de masa; double
- Print_text: textul ce a fost pregatit pentru imprimarea bonului fiscal; varchar(255)
- Id_store: cheie straina, legata de tabela store pentru a stii in care magazin a avut loc vanzarea; int(10)
- Id_user: cheie straina, legata de tabela user pentru a stii cine a efectuat vanzarea; int(10)
- Finished: la inceputul unei vanzari in acest camp este scrisa valoarea 0, iar in cazul in care vanzarea ajunge la bun sfarsit acesta este marcat cu 1; tinyint(1)

Tabela receipt_entry:

Contine informatiile referitoare la intrarile din vanzare:

- Id: cheie primara; int(11)
- Quantity: cantitatea vanduta; double
- Id_receipt: cheie straina, legata de tabela receipt pentru a stii in cadrul carei vanzare a avut loc intrarea; int(11)
- Id_product: cheie straina, legata de tabela product pentru a stii care produs a fost vandut; int(10)

Tabela invoice:

Contine informatiile referitoare la factura:

- Id: cheie primara; int(11)
- Id_client: cheie straina, legata la tabela client pentru a stii pentru ce client a fost creata factura; int(11)

- Id_receipt: cheie straina, legata la tabela receipt; va fi creat un rand nou in tabela receipt cu valoarea 0 pe finished pana cand clientul va plati; int(11)
- Id_store: cheie straina, legata la tabela store pentru a stii din care magazin a fost creata factura si pentru a scrie datele magazinul pe factura; int(11)
- Id_user: cheie straina, legata la tabela user pentru a stii de catre care utilizator a fost efectuata factura; int(10)
- Value: valoarea facturii; double
- Finished: la inceputul unei facturi in acest camp este scrisa valoarea 0, iar in cazul in care factura ajunge la bun sfarsit acesta este marcat cu 1; tinyint(1)

Tabela invoice_entry:

Contine informatiile referitoare la intrarile din factura:

- Id: cheie primara; int(11)
- Id_invoice: cheie straina, legata de tabela invoice pentru a stii in cadrul carei facturi a avut loc intrarea; int(10)
- Id_product: cheie straina, legata de tabela product pentru a stii care produs a fost facturat; int(10)
- Quantity: cantitatea vanduta; double

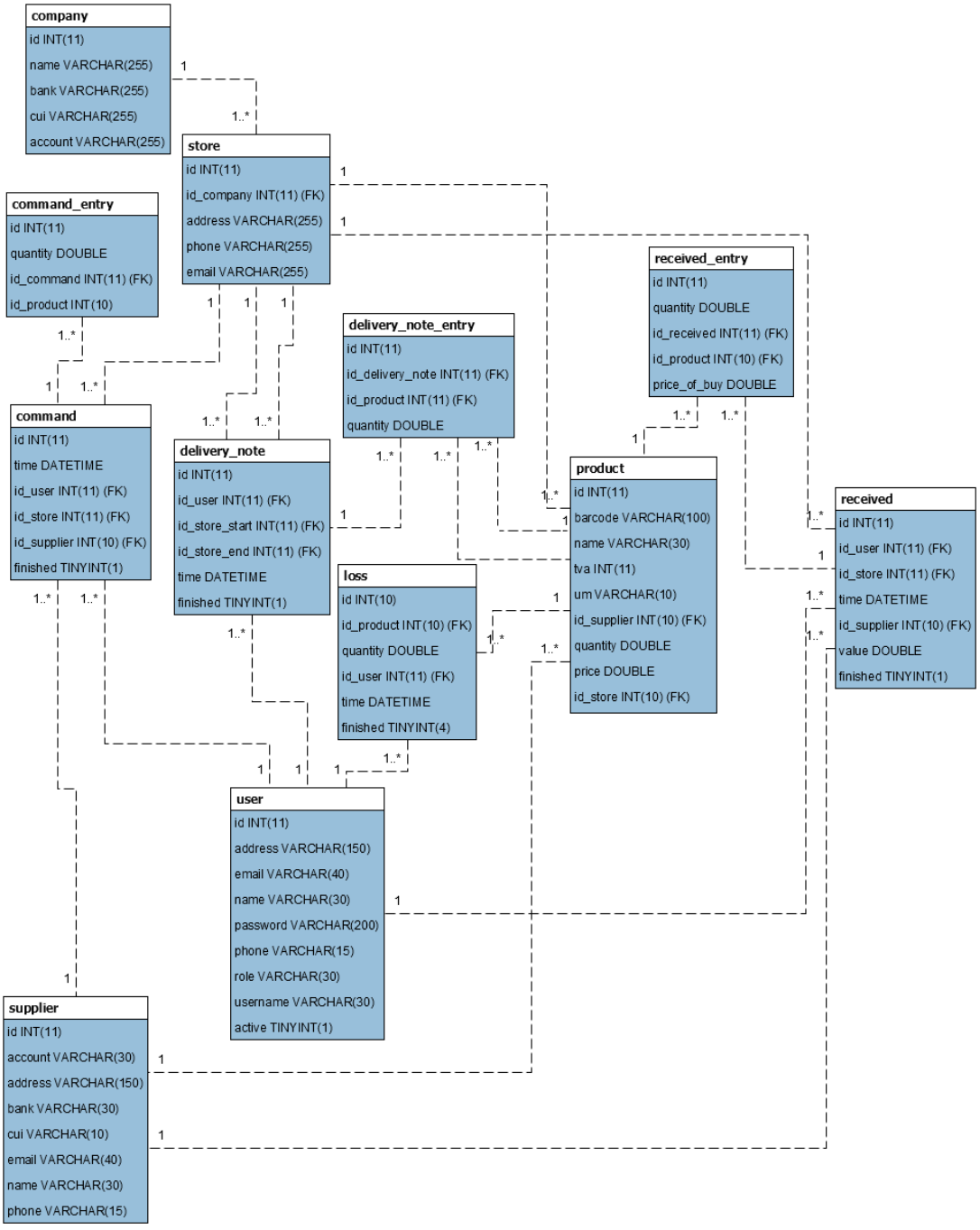

Figura 5.7 – Diagrama baza de date partea 2

Tabela supplier:

Contine informatiile referitoare la furnizor:

- Id: cheie primara; int(11)
- Account: contul bancar al furnizorului; varchar(30)
- Address: adresa furnizorului; varchar(150)
- Bank: banca folosita de furnizor; varchar(30)

- Cui: Codul Unic de identificare al furnizorului; varchar(10)
- Email: adresa de email a furnizorului, in cazul in care utilizatorul aplicatiei doreste comunicarea cu furnizorul prin intermediul email-ului precum si pentru transmiterea comenzilor catre acesta prin email; varchar(40)
- Name: numele furnizorului; varchar(30)
- Phone: numarul de telefon al furnizorului; varchar(15)

Tabela loss:

Contine informatiile referitoare la pierderile inregistrate:

- Id: cheie primara; int(11)
- Id_product: cheie straina, legate de tabela product pentru a stii care produs a inregistrat pierderi; int(10)
- Quantity: cantitatea pierduta; double
- Id_user: cheie straina, legata de tabela user pentru a stii care utilizator a inregistrat pierderea; int(11)
- Time: momentul la care a fost inregistrata piederea; datetime
- Finished: la inceputul inregistrarii unei pierderi in acest camp este scrisa valoarea 0, iar in cazul in care inregistrarea pierderii ajunge la bun sfarsit acesta este marcat cu 1; tinyint(1)

Tabela command:

Contine informatiile referitoare la comanda catre furnizor:

- Id: cheie primara; int(11)
- Time: momentul la care a avut loc comanda; datetime
- Id_user: cheie straina, legata de tabela user pentru a stii cine a efectuat comanda; int(11)
- Id_store: cheie straina, legata de tabela store pentru a stii catre care magazin este dorita livrarea a avut loc vanzarea; tabela care la randul ei este legata de tabela company pentru a scrie in fisierul comanda de catre care companie a fost efectuata comanda; int(11)
- Id_supplier: cheie straina, legata de tabela supplier pentru a stii catre care furnizor este legata comanda, precum si pentru a stii adresa de email catre care sa fie trimisa comanda; int(10)
- Finished: la inceputul unei comenzi in acest camp este scrisa valoarea 0, iar in cazul in care comanda ajunge la bun sfarsit acesta este marcat cu 1; tinyint(1)

Tabela command_entry:

Contine informatiile referitoare la intrarile din comanda:

- Id: cheie primara; int(11)
- Quantity: cantitatea comandata; double
- Id_command: cheie straina, legata de tabela command pentru a stii in cadrul carei comenzi a avut loc intrarea; int(11)
- Id_product: cheie straina, legata de tabela product pentru a stii care produs a fost comandat; int(10)

Tabela received:

Contine informatiile referitoare la intrarea de marfa in stoc:

- Id: cheie primara; int(11)
- Id_user: cheie straina, legata de tabela user pentru a stii care utilizator a efectuat preluarea marfii; int(11)
- Id_store: cheie straina, legata de tabela store pentru a stii in care magazin a avut loc intrarea in stoc; int(11)
- Time: momentul la care a avut loc intrarea; datetime
- Id_user: cheie straina, legata de tabela user pentru a stii cine a efectuat comanda; int(11)
- Id_supplier: cheie straina, legata de tabela supplier pentru a stii de la care furnizor a fost primita; int(10)
- Value: valoarea totala a intrarii; double
- Finished: la inceputul unei intrari in acest camp este scrisa valoarea 0, iar in cazul in care intrarea ajunge la bun sfarsit acesta este marcat cu 1; tinyint(1)

Tabela received _entry:

Contine informatiile referitoare la intrarile din comanda:

- Id: cheie primara; int(11)
- Quantity: cantitatea intrata in stoc; double
- Id_received: cheie straina, legata de tabela received pentru a stii in cadrul carei intrari in stoc a avut loc intrarea; int(11)
- Id_product: cheie straina, legata de tabela product pentru a stii care produs a intrat in stoc; int(10)
- Price_of_buy: pretul achizitiei produsului; double

Tabela delivery_note:

Contine informatiile referitoare la avizele de insotire emise pentru transportul marfii intre magazine:

- Id: cheie primara; int(11)
- Id_user: cheie straina, legata de tabela user pentru a stii care utilizator a efectuat avizul; int(11)
- Id_store_start: cheie straina, legata de tabela store pentru a stii in care magazin a avut emiterea avizului; int(11)
- Id_store_end: cheie straina, legata de tabela store pentru a stii in care magazin a urmeaza sa ajunga marfa; int(11)
- Time: momentul la care a avut loc avizul; datetime
- Finished: la inceputul unei aviz in acest camp este scrisa valoarea 0, iar in cazul in care avizul ajunge la bun sfarsit acesta este marcat cu 1; tinyint(1)

Tabela delivery_note _entry:

Contine informatiile referitoare la intrarile din aviz:

- Id: cheie primara; int(11)
- Id_delivery_note: cheie straina, legata de tabela delivery_note pentru a stii in cadrul carui aviz a avut loc intrarea; int(11)
- Id_product: cheie straina, legata de tabela product pentru a stii care produs a fost inscris in aviz, urmand a fi actualizat stocul din magazinul sursa la emitere si la fel actualizat in magazinul destinatie la livrare; int(11)
- Quantity: cantitatea trecuta in aviz; double

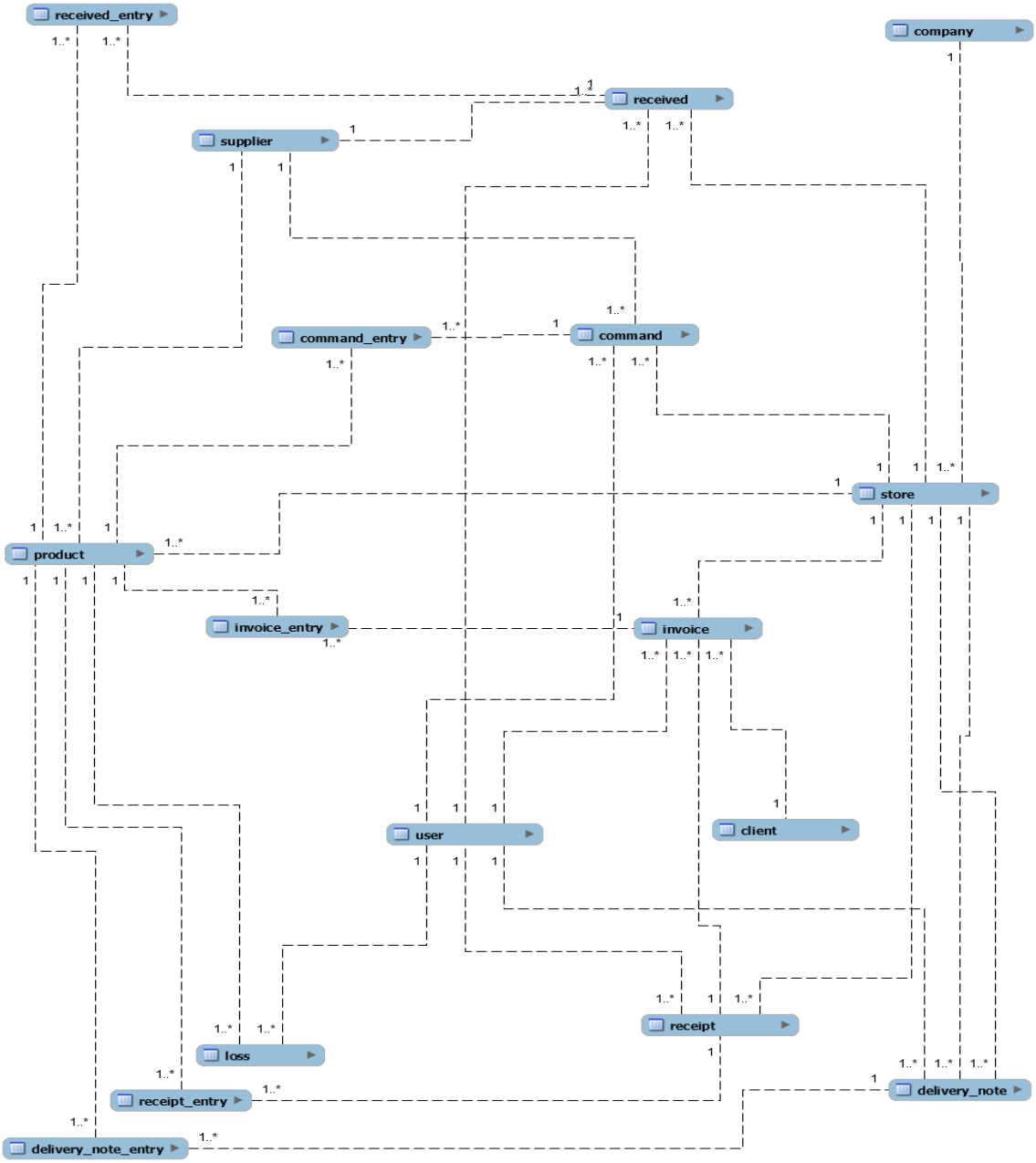

Figura 5.8 – Diagrama baza de date completa

In implementarea bazei de date s-a folosit normalizarea pentru a avea o baza de date corect si consistent construita.

Astfel, pentru ca forma normala 1 sa fie indeplinita fiecare tabela are o cheie primara. Pentru forma normala 2, pe langa forma normala 1, elementele ce apartin unei tabele trebuie sa fie dependete total de cheile primare, forma fiind indeplinita datorita faptului ca fiecare tabele are o cheie primare.

Forma normala 3 a fost indeplinita fiind indeplinita forma normala 2 si eliminate dependentele tranzitive prin care un atribut este dependent de cheia primara prin intermediul altui atribuit prin crearea de tabele. Forma normala Boyce-Codd a fost obtinuta prin indeplinirea formei normale 3, precum si a faptului ca orice determinant din orice relatie este cheie candidat.

5.4. Elemente de implementare

In aceasta parte vor fi prezentate functionalitatile reprezentative ale aplicatiei.

5.4.1. Vânzare

In cadrul procesului de vanzare din aplicatia client la momentul accesarii functiei de vanzare, este initiat un apel catre partea de back-end pentru crearea unei vanzari, avand in corpul apelului informatiile necesare crearii vanzarii. Mai departe urmeaza ca partea de back-end sa se ocupe de validarile necesare si sa trimita inapoi catre client informatiile despre vanzare.

Pasul urmator este cautarea produsului dupa codul de bare. Cu ajutorul scanner-ului pentru coduri de bare va fi citit codul produsului, iar datorita faptului ca scanner-ul emuleaza functionarea unei tastaturi va atasa la finalul codului de bare tasta Enter, astfel fiind initiat apelul catre back-end pentru cautarea produsului. In caz de insucces va fi returnat un mesaj cu produs neregasit sau cantitate indisponibila. In cazul in care sunt regasite produse in magazinul din care are loc vanzarea, vor fi returnate catre client, iar dintr-o fereastra dedicata, operatorul va putea selecta produsul ce trebuie adaugat in cos. Odata selectat produsul, va fi initiat un apel catre back-end pentru a crea o noua intrare in vanzare. Motivul pentru care exista mai multe produse este acela ca la un moment de timp pot exista in stoc produse de exact acelasi fel insa la preturi diferite, iar responsabilitatea selectarii produsului corect ii revine operatorului.

Odata adaugat un produs, campul dedicat cantitatii in tabel poate fi modificat cu valoarea ce se doreste a fi vanduta, fiind initiat un apel catre back-end pentru verificarea faptului daca cantitatea dorita este disponibila si actualizarea intrarii de vanzare. In cazul in care cantitatea ceruta este indisponibila, operatorul va fi notificat.

Daca clientul se razgandeste si nu mai doreste, operatorul poate elimina din cos produsul cu un simplu click pe butonul „Sterge” din dreptul produsului. In acest moment va fi initiat un apel catre back-end pentru stergerea intrarii din vanzare.

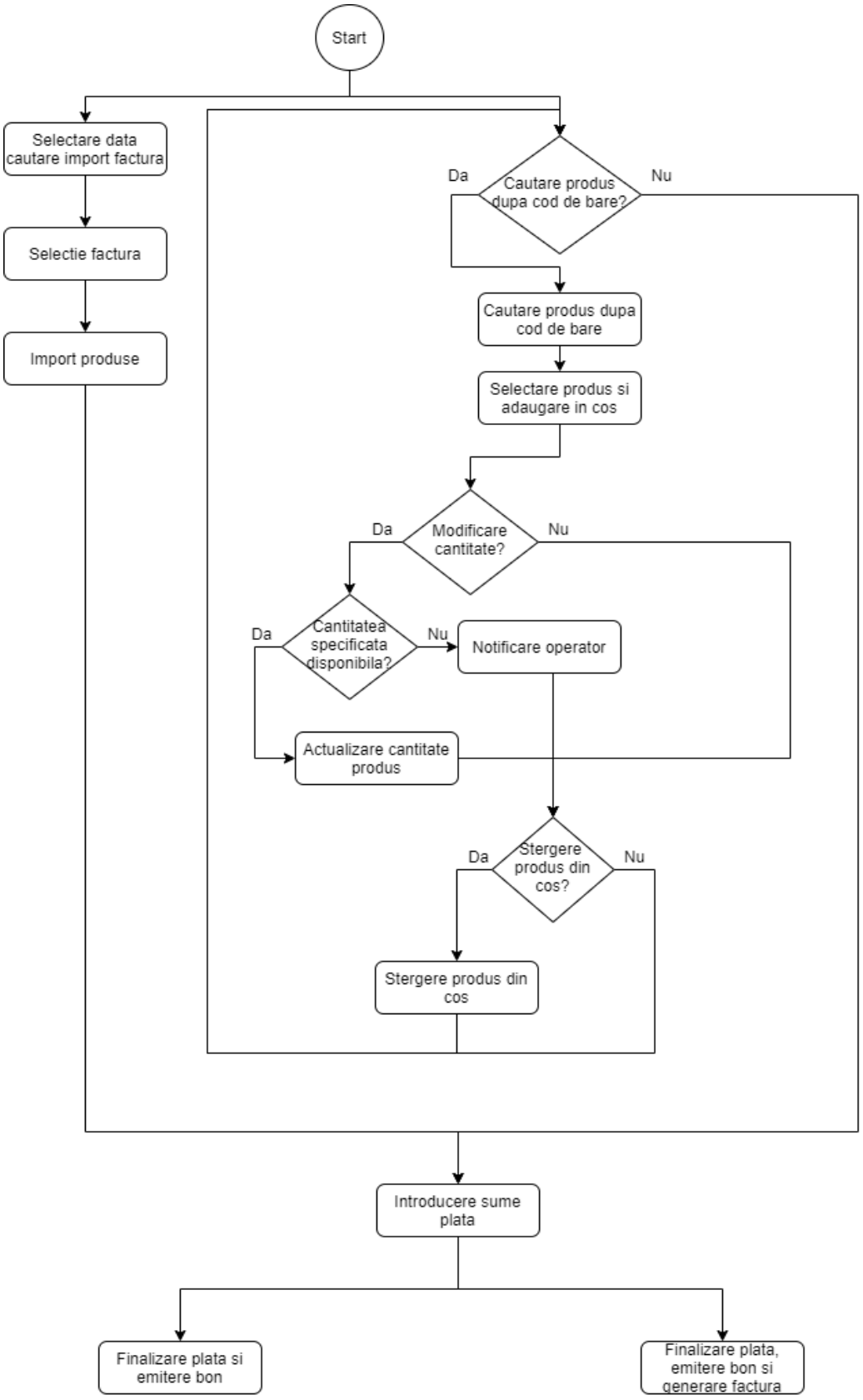

Figura 5.9 – Organigrama vanzare

Odata finalizat procesul de adaugare in cos, se trece la etapa de finalizare. In cadrul acestei etape, operatorul poate introduce in campurile de dedicate sumele fiecarui tip de plata, respectiv cash, card sau tichete de masa. Nu este obligatoriu ca plata sa fie efectuata cu o singura modalitate de plata, acestea putand fi combinate. Odata finalizata vanzarea, va fi actualizat stocul, precum si marcat campul finished din tabela receipt cu 1. In cazul in care suma achitata de client depaseste suma necesare, operatorul va fi notificat cu suma care trebuie returnata clientului drept rest. Textul pentru imprimarea bonului va fi construit in cadrul serverului si trimis catre aplicatia client, care il va salva sub format „.inp” in folderul dedicat, casa de marcat urmand sa il imprime.

In cazul in care clientul doreste si factura pentru produsele achizitionate, operatorul are la dispozitie un combo-box de unde poate selecta clientul pentru care sa fie creata factura. Odata selectat clientul, prin apasarea butonului „Finalizare cu factura”. Va fi finalizata atat vanzarea, cat si creata factura cu intrarile necesare, generat fisierul PDF cu toate datele completate(companie, magazin, client, produse) si trimis catre aplicatia client, de unde operatorul va putea sa il imprimeze si sa il inmaneze clientului.

O situatie speciala este cea in care exista o factura emisa anterior, iar clientul s-a prezentat pentru achitarea facturii. In acest caz, poate fi selectata data emiterii facturii, trimisa catre server aceasta cerinta, care va cauta printre facturile emise in toate magazinele si returna facturile regasite. Operatorul va selecta factura de platit, moment in care produsele vor fi adaugate in cos si de asemenea si intrarile aferente in baza de date corespunzatoare bonului de marcat, insa vor fi dezactivate functionalitatile de cautare dupa cod de bare, modificare cantitate si stergere, intrucat bonul fiscal trebuie sa corespunda cu factura. In continuare, se va trece la etapa de finalizare in care vor fi specificate modalitatile de plata si sumele aferente. Si in aceasta fereastra va fi dezactivat butonul de „Finalizare cu factura”, intrucat exista o factura emisa anterior. Odata apasat butonul „Finalizare”, va fi trimisa catre client sarcina, care va verifica daca exista o factura emisa deja pentru acest bon, caz in care nu va actualiza stocul, intrucat acesta a fost actualizat la facturare. In continuare va fi trimis textul necesar imprimarii bonului si imprimat bonul.

5.4.2. Facturare

Odata accesata aceasta sectiune, operatorul trebuie sa selecteze clientul catre care se face facturarea, dupa selectare fiind creata factura cu finished marcat cu 0.

Functionalitate similara cu cea de vanzare. Cautarea produselor, actualizarea cantitatii sunt identice. Procesul de finalizare este chiar mai simplu decat la vanzare, intrucat nu trebuie specificata metoda de plata, iar odata apasat butonul de finalizare, este creat fisierul PDF si trimis catre client, precum si marcat campul finished cu 1.

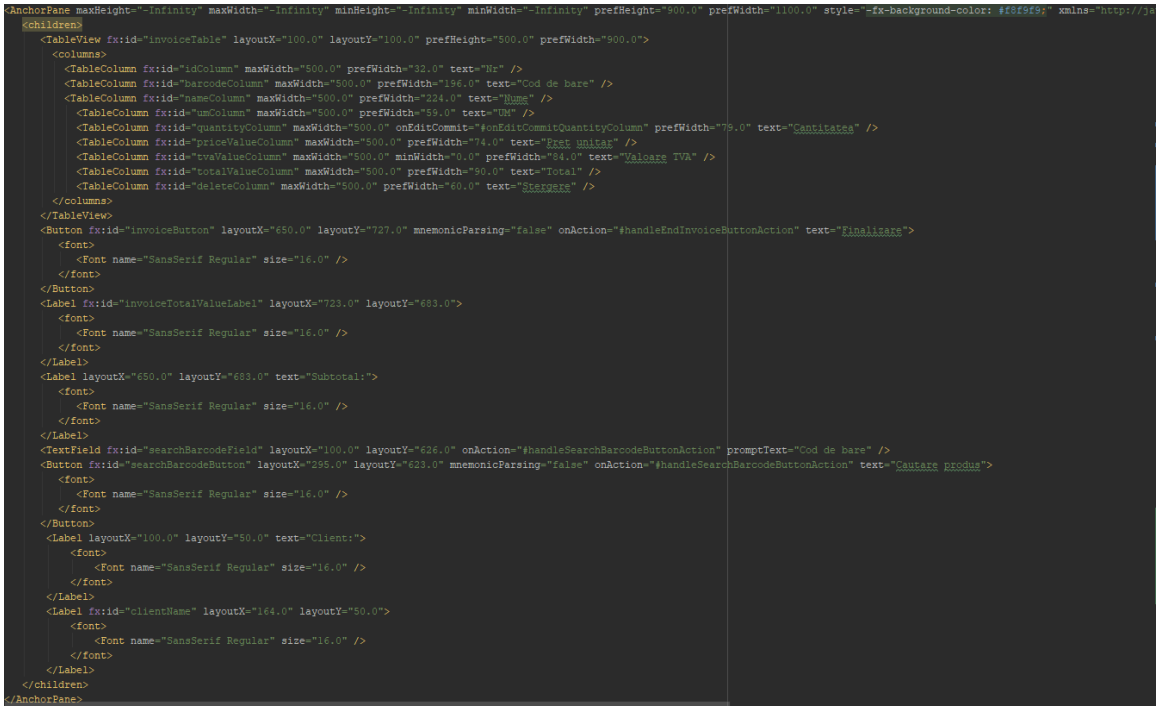

Figura 5.10 – Descrierea in FXML a ferestrei de factura

5.4.3. Comandă

In cadrul aceste sectiuni, operatorul va trebui sa selecteze in prima faza furnizorul catre care se face comanda si magazinul catre care sa fie facuta livrarea. Odata selectata va fi creata comanda si marcat campul finished cu 0.

Aplicatia doreste sa fie flexibila, din acest considerent ofera multiple variante de adaugare a produselor in comanda. Una dintre ele este aceea in care cel care efectueaza comanda a fost informat de catre un agent de vanzari despre existenta unui nou produs care poate fi comandat. In aceasta eventualitate, operatorul poate introduce acest produs in comanda, specificand numele produsului, unitatea de masura si cantitatea. La adaugarea produsului, este creat noul produs si in tabela product.

De asemenea, operatorul poate cauta un produs dupa sau dupa furnizor. Situatii in care acesta va putea vizualiza produsele cu stocul total pe fiecare dintre magazinele disponibile. Odata selectat un produs, acesta va fi adaugat in comanda si initiat apelul catre sever pentru adaugarea produsului in comanda.

De asemenea, operatorul poate modifica cantitatea ce doreste sa fie comandata din campul dedicat sau poate sterge produsul din comanda. In fiecare dintre cazuri, este initiat un apel catre back-end pentru actualizarea intrarilor din comanda.

Daca au fost adaugate toate produsele in comanda, operatorul poate selecta „Finalizare comanda”, caz in care va fi finalizata comanda, marcat campul finished cu 1 si generat fisierul PDF. Daca doreste, operatorul poate opta pentru varianta cu trimitere de email catre furnizor cu comanda atasata.

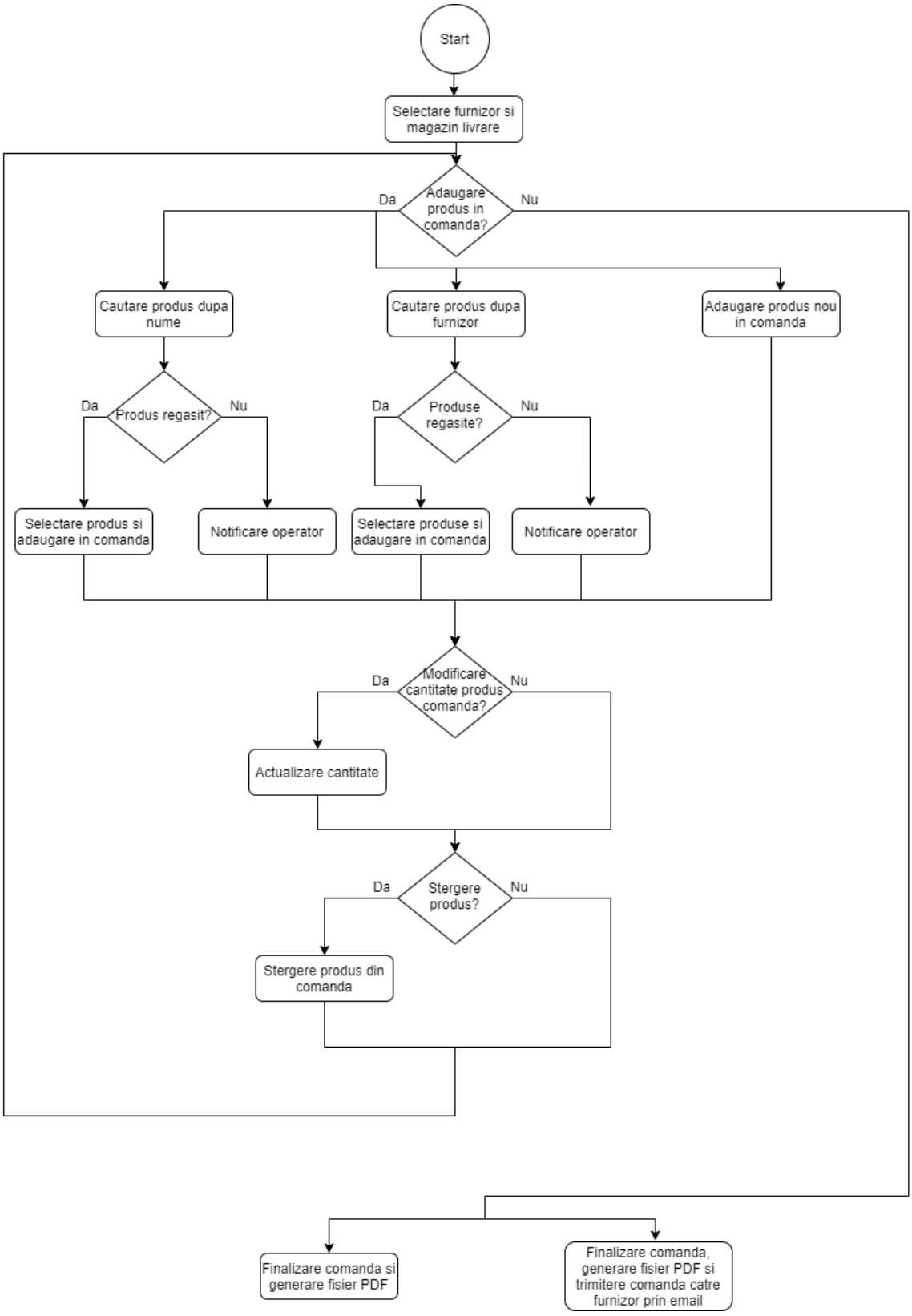

Figura 5.11 – Organigrama comanda

5.4.4. Intrarea

Primul pas este acela de selectie a furnizorului, fiind creata astfel intrarea in baza de date si marcat campul finished cu 0.

Similar cu comanda si in cazul intrarii aplicatia doreste sa fie cat de flexibila posibil. In acest sens, daca comanda a fost creata fara ajutorul aplicatiei(prin telefon sau interactiune directa cu un agent de vanzari) aceasta poate fi adaugata manual fara probleme.

Daca exista un produs care nu este prezent in baza de date, acesta poate fi adaugat specificand codul de bare al produsului, numele, cantitatea, unitatea de masura, cota TVA, pretul de achizitie si pretul de vanzare.

Daca produsul exista, acesta poate fi cautat dupa nume sau dupa furnizor. Operatorul va putea selecta dintr-o lista produsul cautat, iar acesta fi adaugat la intrare.

In toate situatiile prezentate anterior, daca exista modificari la nivelul codului de bare, cotei TVA, precum si pentru specificarea pretului de achizitie, pretului de vanzare si a cantitatii receptionate, campurile pot fi modificate in interiorul tabelului cu intrari. De asemenea, in cazul in care un produs a fost adaugat accidental acesta poate fi sters.

Odata adaugate toate produsele in intrare, va putea fi apasat butonul „Finalizare intrare”. In server vor fi verificate urmatorele situatiii:

- Daca unul dintre produsele adaugate la intrare apartine de alt magazin, iar produsul respectiv nu se regaseste in stocul magazinului curent, acest va fi creat pentru magazinul curent.
- Daca au intervenit modificare la nivelul codului de bare, cotei TVA sau a pretului de vanzare fata de ce se regaseste in baza de date, atunci va fi creat un produs nou.
- In toate celelalte situatii, va fi actualizat corespunzator stocul, precum si finalizata intrarea si marcat campul finished cu 1.

In eventualitatea in care intrarea este conform unei comenzi, comanda poate fi cautata dupa data si apoi selectata, urmand ca produsele sa fie adaugata in intrare. Chiar daca produsele primite sunt in urma unei comenzi, exista posiblitatea in care cantitatea primita este alta fata de cea comandata, exista produse lipsa sau in plus(in cazul in care un agent de vanzari a comunicat cu un operator dupa efectuarea comenzii). In toate aceste situatii, operatorul are libertatea de a efectua operatiile necesare pentru a crea intrarea conform cu realitatea.

5.4.5. Aviz

O alta sectiune importanta este cea dedicata avizelor. Un aviz de insotire a marfii este necesar pentru ca un operator sa poate transfera marfa intre un magazin si altul.

In primul rand operatorul va trebui sa selecteze tipul operatiuni cu aviz: creare sau primire.

Pentru cazul de creare, operatorul va selecta magazinul destinatie iar avizul va fi creat cu campul finished 0. In continuare, operatorul va putea cauta produse din magazinul curent dupa, selectand dintr-o lista produsele ce se doresc a fi adaugate in aviz.

Poate fi modificata cantitatea ce se doreste a fi transferata, operatorul fiind notificat in cazul in care cantitatea inscrisa este indisponibila. De asemenea, poate fi sters un produs din aviz.

Odata finalizata adaugarea produselor, prin apasarea butonul „Creare aviz” va fi creat un fisier PDF reprezentand avizul de insotire, actualizat corespunzator stocul si marcat campul finished cu 1.

Pentru cazul de primire, operatorul va selecta data emiterii avizului, iar dintr-o lista va selecta avizul corect. Produsele vor fi adaugate automat in tabel, iar functionalitatea de adaugare produse va fi dezactivata, precum si cele de modificare cantitate sau stergere. Prin finalizare, stocul va fi actualizat corespunzator in magazinul destinatie.

5.4.6. Raporte

In cadrul aceste sectiuni, operatorul va putea vizualiza raporte despre vanzari, facturi, comenzi, intrari, avize, pierderi si stoc. Pentru acestea, el poate efectua cautari dupa client, furnizor, magazin, operator, data sau nume produs, depinzand de tipul operatiunii. Odata efectuat raportul, acesta poate fi exportat sub format .xlsx.

5.4.7. Operațiuni DATECS, Clienți și Furnizori

Prima dintre aceste functionalitati este dedicata intrarilor sau iesirilor din sertarul casei de marcat.

Cea de a doua este pentru crearea de clienti specificand informatiile necesare, dar si pentru efectuarea de modificari in cazul in care situatia o cere, putand fi efectuate modificari la nivelul adresei, email-ului, numarului de telefon, bancii si contului. Toate informatiile despre clienti pot fi exportate in format .xlsx.

Pentru furnizori sunt oferite aceleasi facilitati precum si pentru clienti.

5.4.8. Setări personale

Sectiunea este dedicata setarilor pe care fiecare utilizator sa le faca pentru contul personal. Astfel, acesta poate sa isi modifice numele, adresa, adresa de email sau numarul de telefon, username-ul si rolul ramanand neschimbate. De asemenea, acesta poate sa isi schimbe parola.

```
public ChangePasswordDTO changePassword(ChangePasswordDTO changePasswordDTO) {

    UserEntity userEntity = userRepository.findByUsernameAndPasswordAndActive(changePasswordDTO.getUsername(), changePasswordDTO.getCurrentPassword(), (byte) 1);

    if(userEntity != null)
    {
        userEntity.setPassword(changePasswordDTO.getNewPassword());
        userRepository.saveAndFlush(userEntity);
        return changePasswordDTO;
    }
    else {
        return null;
    }
}
```

Figura 5.12 – Metoda care realizeaza schimbarea parolei unui utilizator

5.4.9. Gestionare operatori

Sectiune dedicata exclusiv administratorilor, de unde acestia pot sa adauge noi operatori specificand campurile necesare, operatorul urmand sa primeasca un email cu credentialele pentru logare. De asemenea, poate fi modificat rolul unui operator,

modificarea statusului din activ in inactiv caz in care respectivul operator nu va mai avea acces la aplicatie, precum si resetarea parolei in eventualitatea in care operatorul nu si-o mai aminteste.

Toate documentele in format PDF vor fi generate pe server si transmise catre client. Indiferent ce se intampla cu fisierele de pe computerele pe care ruleaza aplicatia client, cele de pe server vor ramane pentru a avea o evidenta a documentelor.

La deschiderea aplicatiei, intr-un fisier dedicat este inscris id-ul magazinului, fiind efectuat un apel catre server pentru aducerea informatiilor referitoare la magazinul curent

5.4.10. Securizare

Pentru ca o aplicatie nu este completa fara a asigura integritatea datelor, siguranta comunicatiei client-server si evitarea diferitelor atacuri malitioase, este necesara si implementarea diferitelor sisteme de securitate.[10]

Fie ca este vorba de un certificat de tip self-signed sau emis de o autoritate competenta, Spring poate fi configurat cu usurinta sa accepte HTTPS in locul HTTP. Primul pas este acela de a adauga fisierul keystore in folderul resources, urmand configurarea serverului pentru pentru utilizarea keystore-ului si activarea HTTPS.

In fisierul aplication.properties sunt definite urmatoarele proprietati:

```
server.port = 8443

server.ssl.key-store-type = PKCS12
server.ssl.key-store = classpath:keystore.p12
server.ssl.key-store-password = sergiumorar
server.ssl.key-alias = SM
```

Figura 5.13 – Setari aplication.properties pentru HTTPS

In continuare, in cadrul clasei SecurityConfig, este setat sa blocheze automat request-urile de tip HTTP, precum si csrf este dezactivat. De asemenea, requestu-rile care vin prin HTTP sunt redirectionate catre HTTPS cu ajutorul clasei ServerConfig.[20]

```
private Connector getHttpConnector() {
    Connector connector = new Connector(TomcatServletWebServerFactory.DEFAULT_PROTOCOL);
    connector.setScheme("http");
    connector.setPort(8080);
    connector.setSecure(false);
    connector.setRedirectPort(8443);
    return connector;
}
```

Figura 5.14 – Redirectionare HTTP catre HTTPS

De partea cealalta, a aplicatiei client, este incarcat la fel fisierul .p12, iar apoi este construit contextul SSL care va fi utilizat de catre CloseableHttpClient pentru efectuarea apelurilor.

De, asemenea inainte ca parolele sa fie transmise catre server, acestea sunt hashuite in aplicatia client.

[20] https://www.thomasvitale.com/https-spring-boot-ssl-certificate/

5.4.11. Utilitare

In cadrul pachetului „Utils” din server se afla implementarile pentru generarea fiserelor de tip PDF, fie ca este vorba de factura, comanda sau aviz. De asemenea, aici este pregatita configurarea pentru trimiterea unui email, fie ca este catre furnizor sau operator.

```
public static boolean sendMail(String from, String to, String subject, String text, String password, String path) {


    final Properties props = new Properties();
    props.put("mail.smtp.auth", "true");
    props.put("mail.smtp.starttls.enable", "true");
    props.put("mail.smtp.host", "smtp.gmail.com");
    props.put("mail.smtp.port", "587");
    props.put("mail.smtp.ssl.trust", "smtp.gmail.com");

    Session session = Session.getInstance(props,
            getPasswordAuthentication() → {
                    return new PasswordAuthentication(from, password);});

    Message msg = new MimeMessage(session);

    try {
        msg.setFrom(new InternetAddress(from));

        msg.setRecipients(Message.RecipientType.TO,
                InternetAddress.parse(to,  strict: false));

        msg.setSubject(subject);

        BodyPart messageBodyPart = new MimeBodyPart();
        messageBodyPart.setText(text);
        Multipart multipart = new MimeMultipart();
        multipart.addBodyPart(messageBodyPart);
        messageBodyPart = new MimeBodyPart();
        if(path != null) {
            DataSource source = new FileDataSource(path);
            messageBodyPart.setDataHandler(new DataHandler(source));
            messageBodyPart.setFileName(path.substring(path.lastIndexOf( str: "\\")+1));
            multipart.addBodyPart(messageBodyPart);
        }
        msg.setContent(multipart);

        SMTPTransport t = (SMTPTransport) session.getTransport( protocol: "smtp");
        t.connect(from, password);
        t.sendMessage(msg, msg.getAllRecipients());
        t.close();
        return true;
    } catch (MessagingException e) {
        e.printStackTrace();
        return false;
    }
```

Figura 5.15 – Metoda pentru trimiterea unui email

De partea cealalta, si in aplicatia client este un pachet „Utils” in care se afla implementarea pentru exportarea in format .xlsx a rapoartelor, efectuarea de apeluri sau modificarea scenelor afisate in interiorul aplicatiei.

Capitolul 6. Testare şi Validare

Acest capitol este dedicat testarilor si validarilor proiectului propus.

Testarea este modalitatea prin care este verificat daca functionalitatile unui produs indeplineste cerintele cerute. Presupune executarea de teste manuale sau automate pentru a evalua punctele de interes. Scopul testarii este acela de a detecta erorile, lipsurile sau implemntarile neconforme cu cerintele.

Testarea poate fi de trei feluri:

- Functionala, in care sunt efectuate teste precum UAT(User Acceptance Testing) sau de interoperabilitate
- Non-functionala, in care printre altele este testata performanta, scalabilitatea, utilizabilitatea sau securitatea
- De mentenanta, print regression testing sau maintenance testing.[21]

Testarea unui scenariu presupune parcurgerea unui flux complet al operatiei respective. Pe parcursul testarii scenariului poate fi observat modul in care aplicatia raspunde la un comportament inadecvat precum introducerea de valori incorecte in campurile dedicate, cautarea dupa valori inexistente sau intreruperea fluxului la momente neasteptate. De asemenea, poate fi observata si comunicarea intre aplicatia client si server. De cealalta parte, in cazul in care utilizarea este cea corecta, aplicatia va furniza rezultatul dorit.

Caz de testare 1: vanzarea
Preconditii: operatorul sa fie logat si sa existe produse in stoc
Rezultatul final asteptat: generarea bonului fiscal si actualizarea stocului

Actiune	Rezultat asteptat
Cautare produs dupa cod de bare	Afisarea listei cu produsele regasite
Selectie produs	Adaugarea produsului in cosul de vanzare
Modificarea cantitatii	Actualizarea cantitatii in cos
Stergerea unui produs	Stergerea produsului din cos
Finalizare	Deschiderea ferestrei dedicate platii
Scrierea metodelor de plata si a sumelor	Actualizarea conforma a metodelor de plata
Finalizare plata	Inchiderea ferestrei de plata, precum si generarea bonului fiscal si actualizarea stocului

Tabel 6.1 – Caz testare vanzare

[21] https://www.guru99.com/software-testing-introduction-importance.html

Caz de testare 2: comanda
Preconditii: operatorul sa fie logat
Rezultatul final asteptat: generarea comenzii in format PDF

Actiune	Rezultat asteptat
Selectie furnizor si magazin livrare	Crearea comenzii cu informatiile corecte
Adaugare produs	Adaugarea produsului in comanda
Cautarea produsului dupa nume	Afisarea listei cu produsele regasite
Selectie produs	Adaugarea produsului in comanda
Cautarea produsului dupa furnizor	Afisarea listei cu produsele regasite
Selectie produs	Adaugarea produsului in comanda
Modificarea cantitatii	Actualizarea cantitatii din comanda
Stergerea unui produs	Stergerea produsului din comanda
Finalizare comanda	Generarea fisierului PDF si trimiterea acestuia catre furnizor daca operatorul doreste

Tabel 6.2 – Caz testare comanda

Capitolul 7. Manual de Instalare si Utilizare

Pe parcursul acestui capitol, vor fi prezentate resursele de tip hardware, precum si cele de tip software pentru instalarea si utilizarea aplicatiei. De asemenea, va fi prezentata si o descriere a modului in care poate fi utilizata aplicatia.

7.1. Resurse necesare

Ca aplicatia sa poata fi instalata si functionala trebuie indeplinite urmatoarele cerinte. Vor fi prezentate intai cele de ordin hardware, mai apoi cele de ordin software.

7.1.1. Resurse hardware

Computer pe care sa ruleze aplicatia server trebuie sa fie dotat cu urmatoarele specificatii tehnice:

- Cel putin 4 GB memorie RAM
- Frecventa procesor cel putin 2 GHz
- Sistem operare: Windows

Computerul pe care ruleaza aplicatia client are nevoie de urmatoarele specificatii:

- Cel putin 2 GB memorie RAM
- Frecventa procesor cel putin 2 GHz
- Sistem operare: Windows

Pe langa toate acestea este necesara conexiune la internet atat pentru server cat si pentru aplicatia client.

7.1.2. Resurse software

Pentru server este necesar sa fie disponibile urmatoarele programe:

- MySql instalat si configurat corespunzator
- Intellij instalat si configurat corespunzator
- JDK(Java Developement Kit)

De cealalta parte, computerul pe care ruleaza aplicatia client are nevoie de Intellij si JDK.

7.2. Manual de utilizare

In cele ce urmeaza va fi prezentat sumar modul in care trebuie utilizata aplicatia.

7.2.1. Configurări

Pentru a realiza comunicatia cu casa de marcat, este necesar ca casa de marcat sa fie setata ca imprimanta fiscala. Pentru realizarea conexiunii intre casa de marcat si

computer este necesar un cablu serial si un adaptor usb serial. Cablul se conecteaza la portul PC al casei de marcat, celalt capat la adaptorul usb iar mai apoi la computer.[22]

Pentru setarea casei de marcat, se apasa tasta OFF, se trece in meniul de programare apasand tasta 4. Pentru viteza de comunicatie de 4800 se realizeaza urmatoare succesiune de taste: 1-30-TOTAL-X-PRC de 7 ori-STL-3-PRC-3-TOTAL.

Din aplicatia SellText este specificat portul COM pe care se afla casa de marcat, precum si viteza de 4800.

7.2.2. Autentificare

Primul ecran care ii va fi prezentat operatorului este cel in care acestra trebuie sa isi introduca credentialele pentru autentificarea in aplicatie. Operatia poate fi efectuata de orice tip de operator.

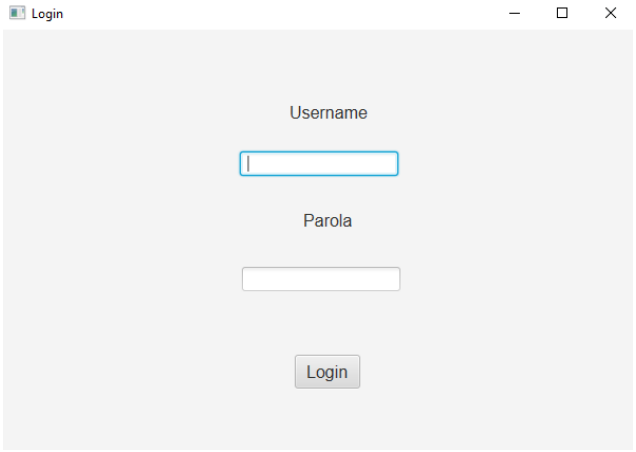

Figura 7.1 – Ecran login

[22] https://www.youtube.com/watch?v=GbxT4egq9Ig

7.2.3. Pagina principală

Odata reusita logarea in aplicatie, operatorul ii va fi prezentata pagina principala.

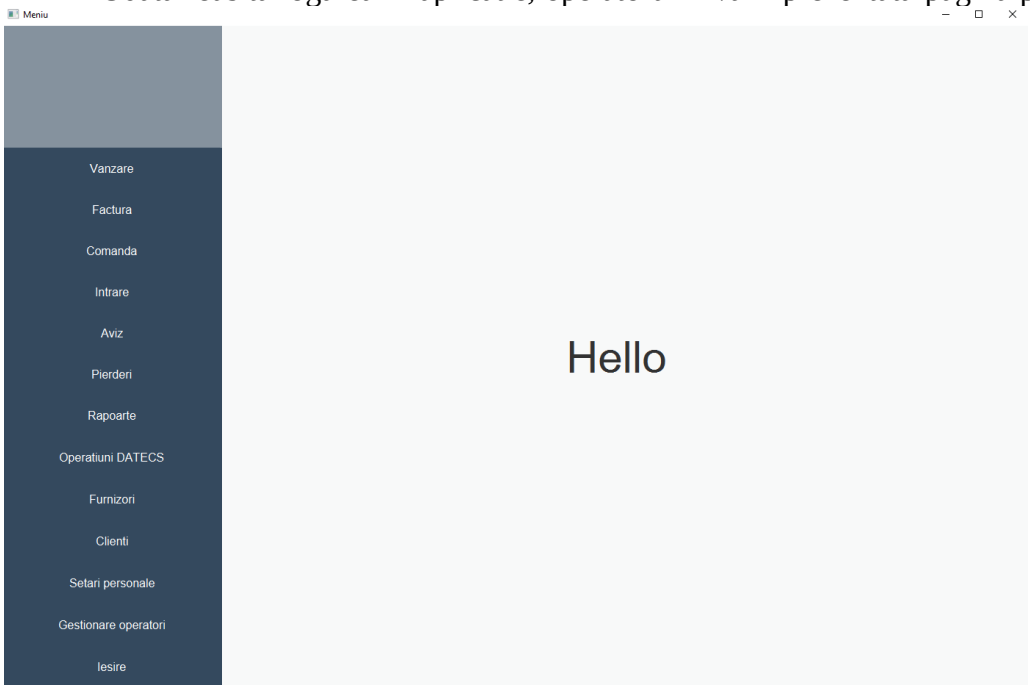

Figura 7.2 – Ecran principal

Pagina prezentata este cea dedicata administratorilor, iar in cazul in care operatorul logat are functia de vanzator sau gestionar acesta va avea mai putine functionalitati disponibile.

7.2.4. Vânzare

Prin selectia butonului de vanzare operatorului ii este deschisa fereastra dedicata vanzarii.

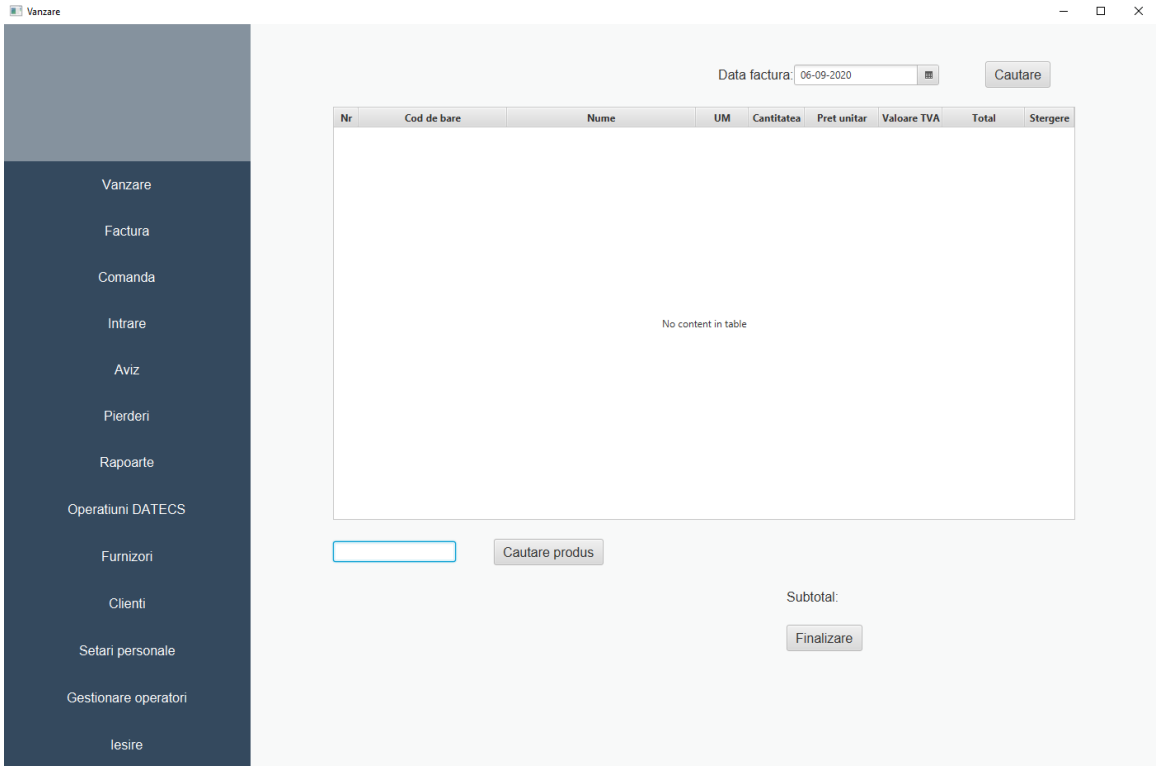

Figura 7.3 – Ecran vanzare

In urma introducerii codului de bare, operatorul va selecta produsul dorit din fereastra prezentata in figura 7.4

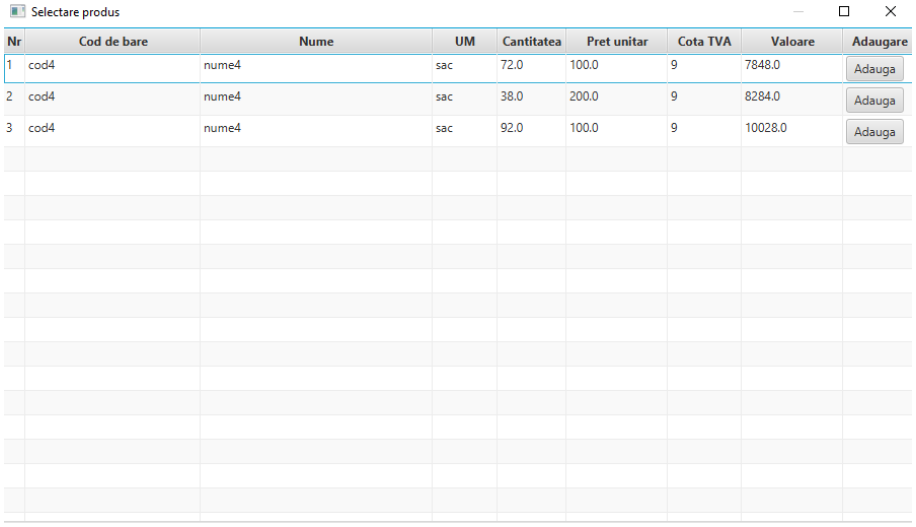

Figura 7.4 – Ecran selectie produs

In coloana pentru cantitate, este prezentata cantitatea disponibila in stoc pentru informa operatorul despre acest aspect.

Dupa ce operatorul a adaugat produsele dorite, poate efectua operatiile referitoare la modificarea cantitatii in vanzare sau eliminare din vanzare a produselor.

Odata finalizat procesul de adaugare a produsului, se trece la finalizare.

Figura 7.5 – Ecran finalizare vanzare

Operatorul poate modifica sumele in modalitatile de plata sau poate opta pentru finalizare a vanzarii cu factura specificand clientul.

In situatia in care exista o factura emisa anterior, operatorul poate selecta data emiterii facturii, iar din ecranul prezentat in Figura 7.7 poate selecta factura. Produsele vor fi incarcate in cos, iar operatorul va putea finaliza vanzarea.

Operatia poate fi efectuata de orice tip de operator.

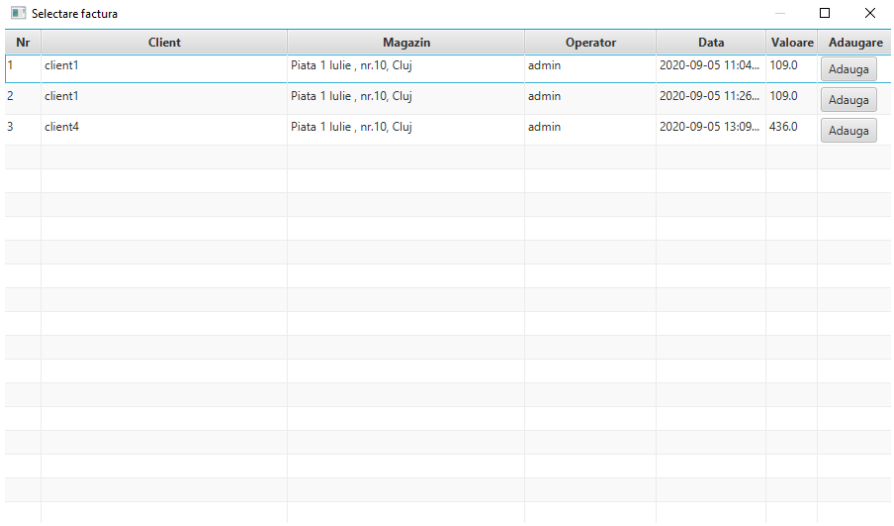

Figura 7.6 – Ecran selectie factura

7.2.5. Factură

Procesul de facturare este similar cu cel de vanzare, chiar mai simplu intrucat nu exista situatii precum importarea unei facturi sau plata. Din aceasta cauza, nu va fi prezentat decat ecranul dedicat selectiei clientului la inceputul facturii. Operatia poate fi efectuata de orice tip de operator.

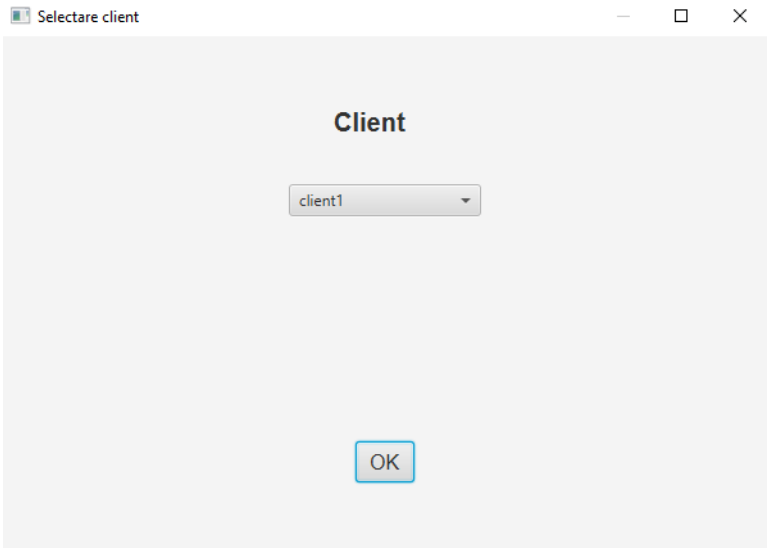

Figura 7.7 – Ecran selectie client

7.2.6. Comandă

Din meniu, poate fi selectata functionalitatea de comanda, urmand apoi ca operatorul sa selecteze furnizorul catre sa fie efectuata comanda precum si magazinul la care sa fie livrata comanda.

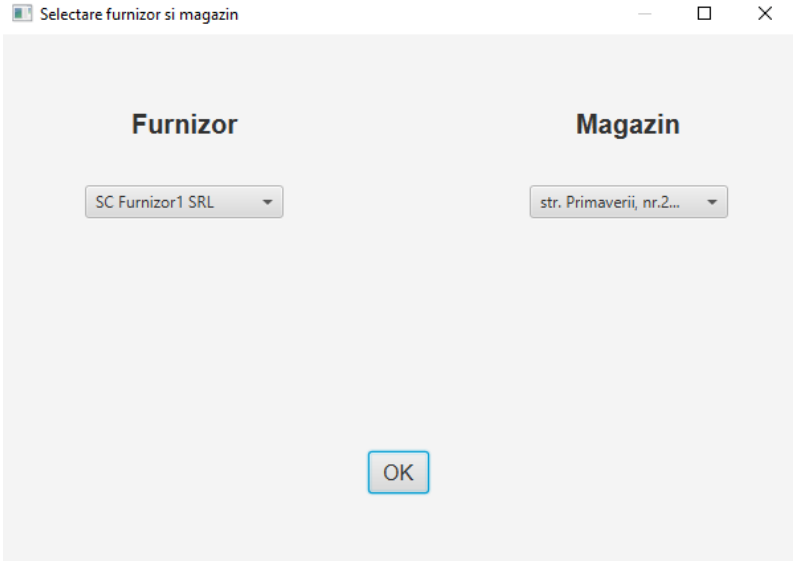

Figura 7.8 – Ecran selectie furnizor si magazin

Odata selectate aceste informatii, operatorul va putea efectua operatiunile de cautare si adaugare in comanda a produselor din ecranul de comanda.

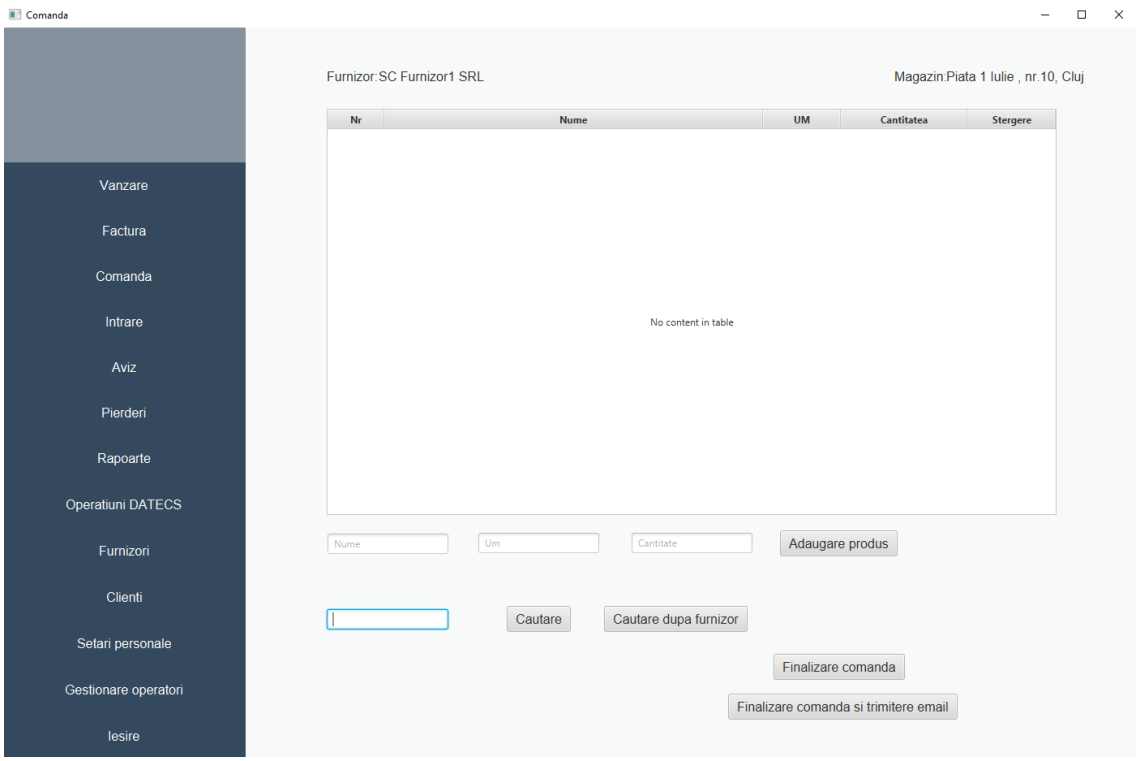

Figura 7.9 – Ecran comanda

In cazul cautarii dupa nume sau furnizor, operatorul va putea selecta dintr-o fereastra conform figurii 7.4 produsele de adaugat in comanda, el putand vizualiza produsele regasite in toate magazinele conform cautarii cu cantitatile cumulate.

Mai departe, operatorul va putea modifica cantitatile produselor din comanda, precum si stergerea lor.

Odata finalizat procesul de adaugare a produselor in comanda, operatorul poate finaliza comanda cu generare de fisier PDF sau generare de fisier PDF si trimiterea catre furnizor.

Operatia poate fi efectuata doar de catre gestionar sau administrator.

7.2.7. Intrare

Similar cu comanda ca si flux, insa cu un alt scop. Este necesara specificarea furnizorului, mai apoi se poate trece la adaugarea produselor efectiv in intrare.

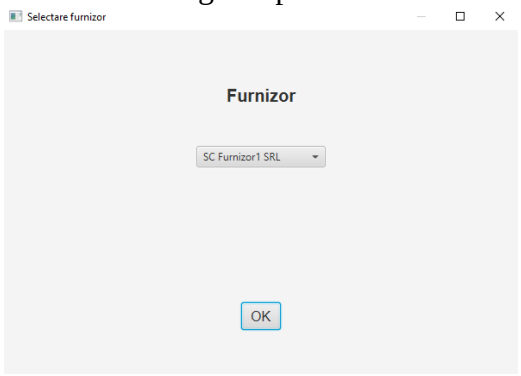

Figura 7.10 – Ecran selectie furnizor

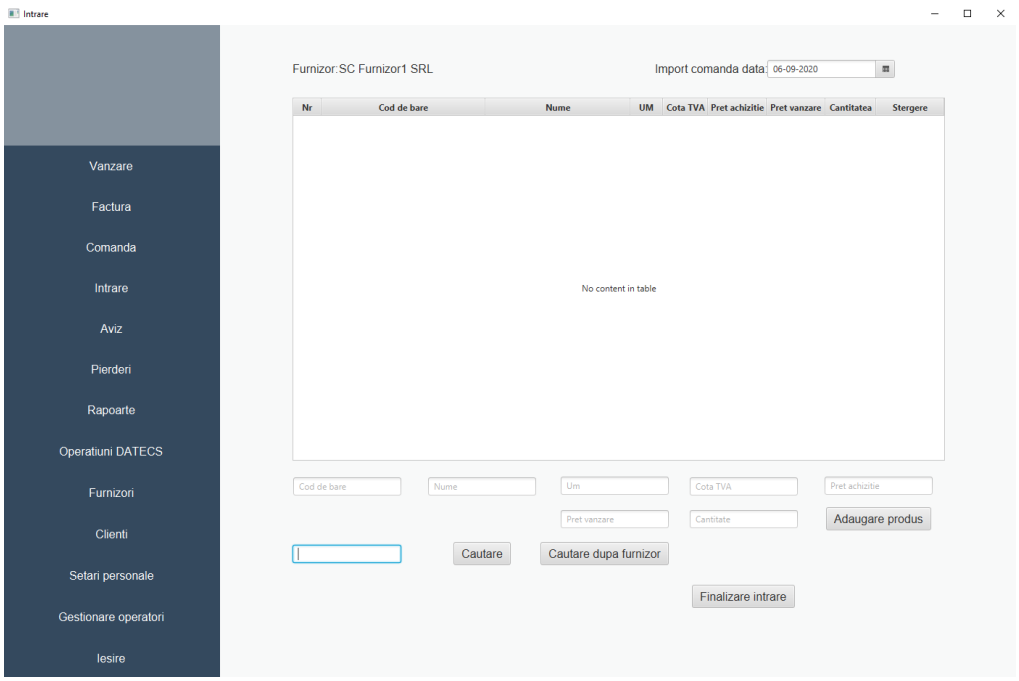

Figura 7.11 – Ecran intrare

Operatorul poate adauga produsul in cazul in care acesta nu exista, iar daca exista poate face cautare dupa nume sau furnizor urmand sa selecteze produsul din fereastra prezentata in figura 7.4. De asemenea, poate modifica cantitatea sau sterge produse din intrare.

In cazul in care exista o comanda care precede intrarea, aceasta poate fi cautata dupa data trimiterii, urmand ca operatorul sa selecteze comanda dorita din fereastra.

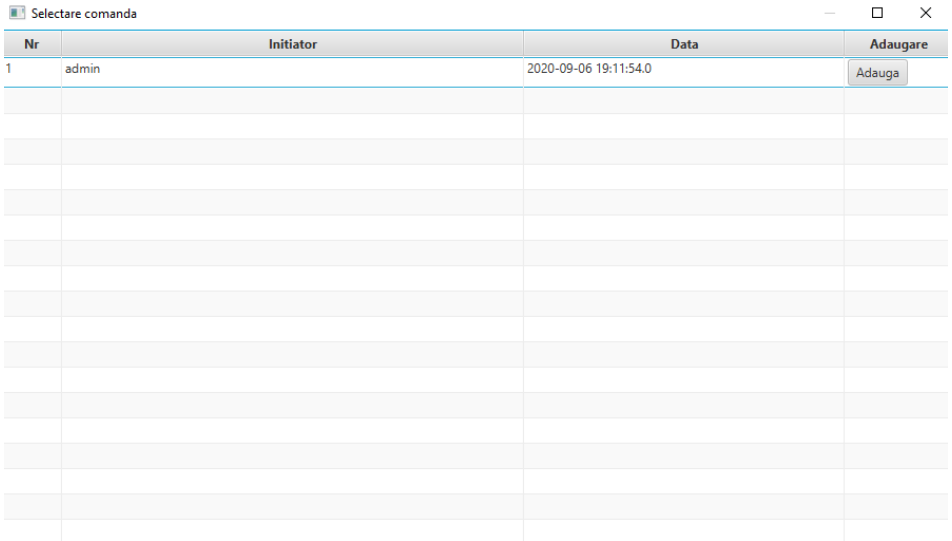

Figura 7.12 – Ecran selectie comanda

Odata selectata comanda, produsele sunt adaugate la intrare, iar operatorul poate efectua operatiile aditionale asupra intrarii. Daca totul este incheiat, prin apasarea butonul „Finalizare intrare”, intrarea este incheiata si stocurile sunt actualizate.

Operatia poate fi efectuata de orice tip de operator.

7.2.8. Aviz

In cadrul acestei sectiuni poate a fi atat generat un aviz, cat si primit.

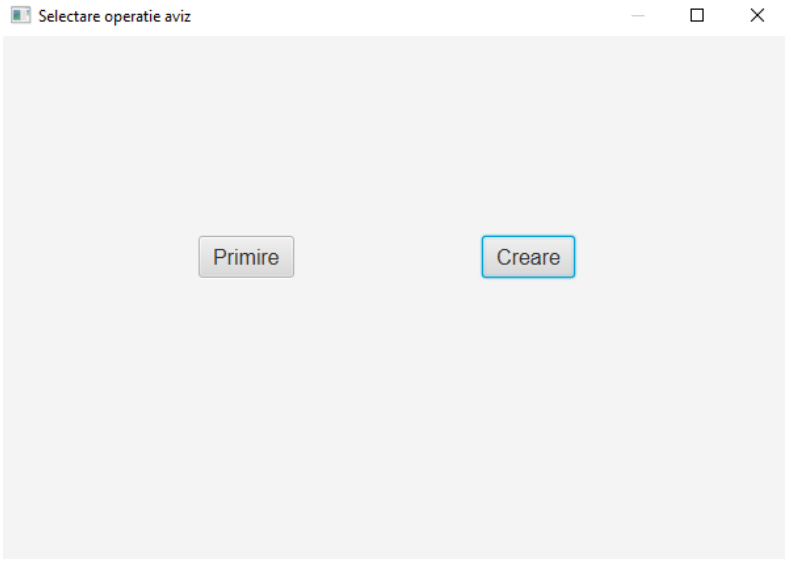

Figura 7.13 – Ecran selectie tip operatie aviz

Pentru cazul in care operatorul doreste sa creeze, acesta va trebui sa specifice magazinul destinatie, magazinul spre care se face transferul de marfa.

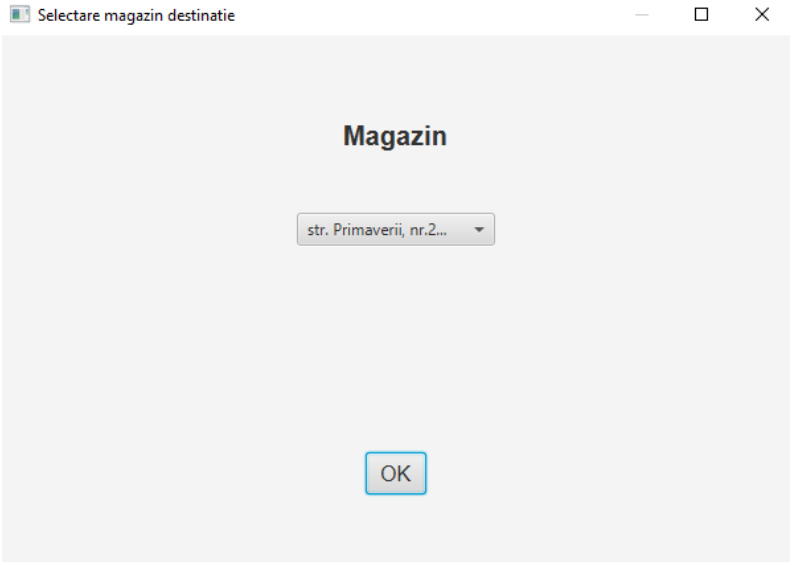

Figura 7.14 – Ecran selectie magazin destinatie

In continuare, operatorul poate incepe procesul propri-zis de adaugare a produselor in aviz prin cautarea dupa nume si selectia din fereastra descrisa in figura 7.4.

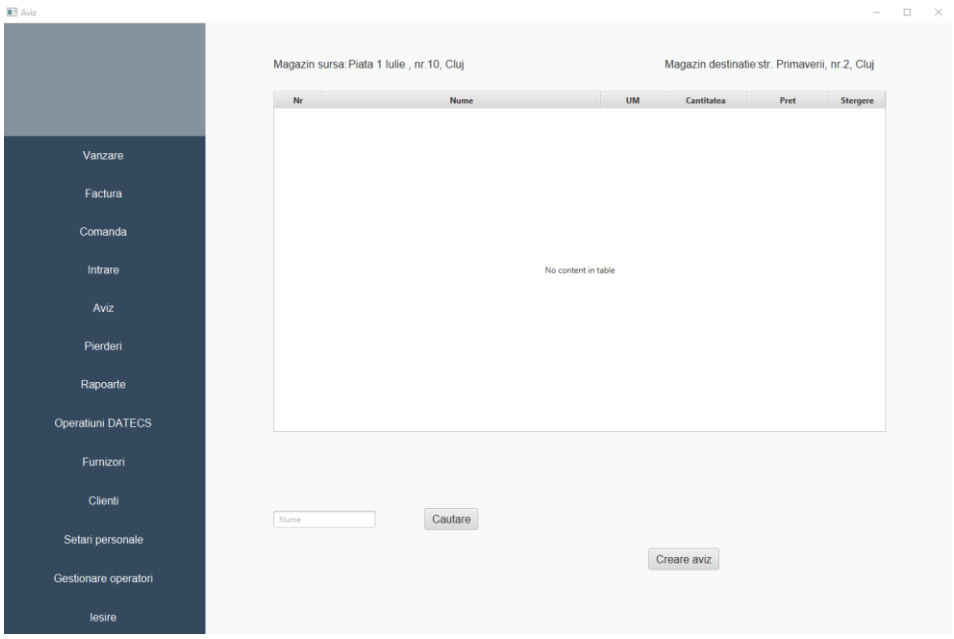

Figura 7.15 – Ecran aviz

Bineinteles, operatorul poate modifica cantitatile pentru livrare in limita stocului disponibil, precum si stergerea din aviz. Odata adaugate produsele, se finalizeaza avizul prin apasarea butonului „Creare aviz”, moment in care este generat si fisierul PDF.

Daca este specificata optiunea de primire aviz, in continuare se face o cautare dupa data, ca mai apoi sa fie specificat avizul dintr-o lista. Produsele vor fi adaugate in lista, insa nu vor putea fi modificate cantitatile sau ster, precum nici nu vor putea fi efectuate cautari dupa numele produselor.

Operatia poate fi efectuata doar de catre gestionar sau administrator.

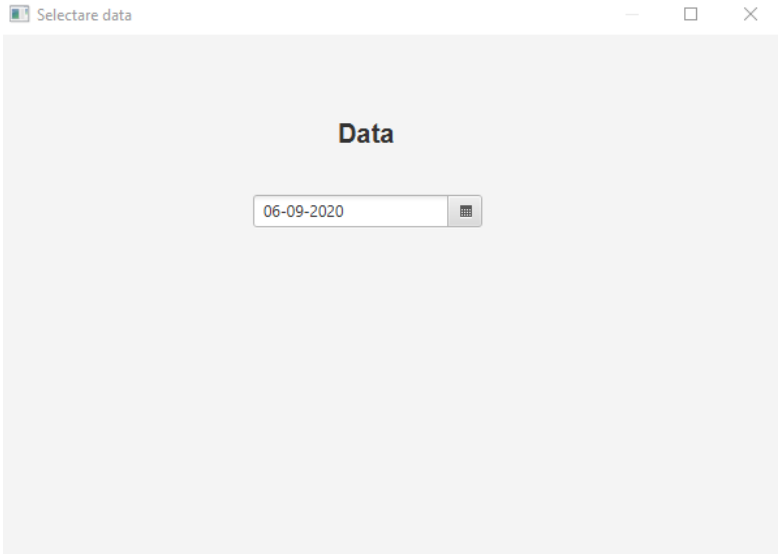

Figura 7.16 – Ecran selectare data

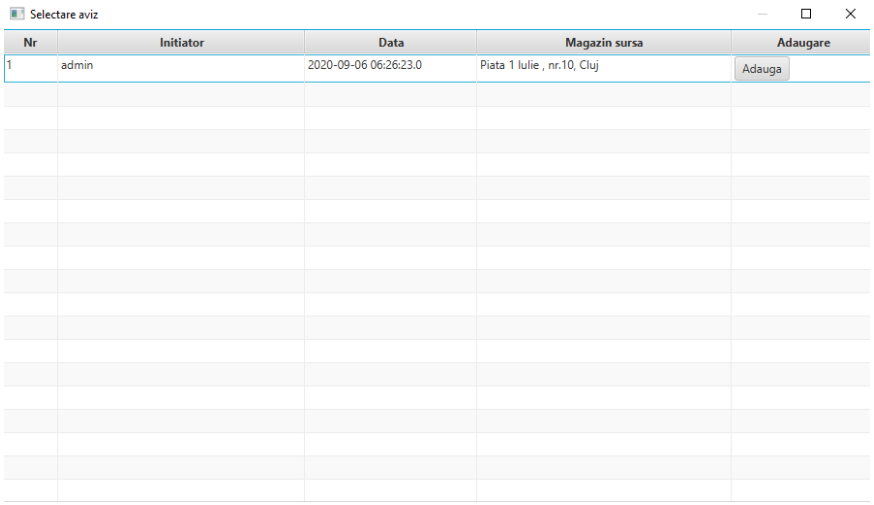

Figura 7.17 – Ecran selectare aviz

7.2.9. Pierderi

In cadrul acestei sectiuni, vor putea fi cautate produse dupa nume din magazinul curent si selectate dintr-o fereastra precum cea prezentata in figura 7.4, mai apoi fiind specificata cantitatea din pierderi. Odata selectat produsul si cantitatea, pierderea poate fi finalizata. Operatia poate fi efectuata doar de catre gestionar sau administrator

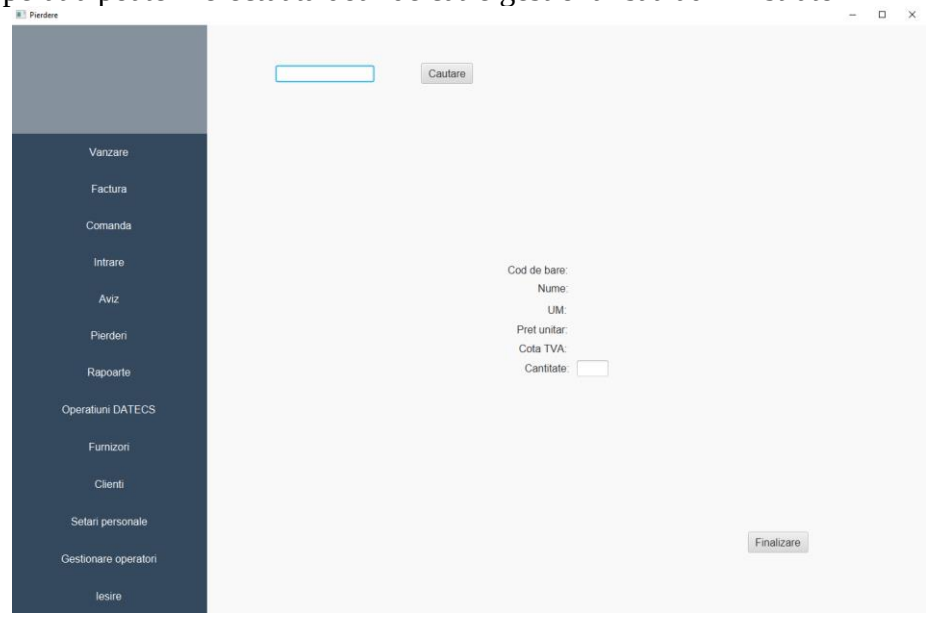

Figura 7.18 – Ecran pierderi

7.2.10. Rapoarte

In cadrul aceste sectiuni, operatorul poate selecta raportul dorit si efectua cautarile conform operatiunilor.

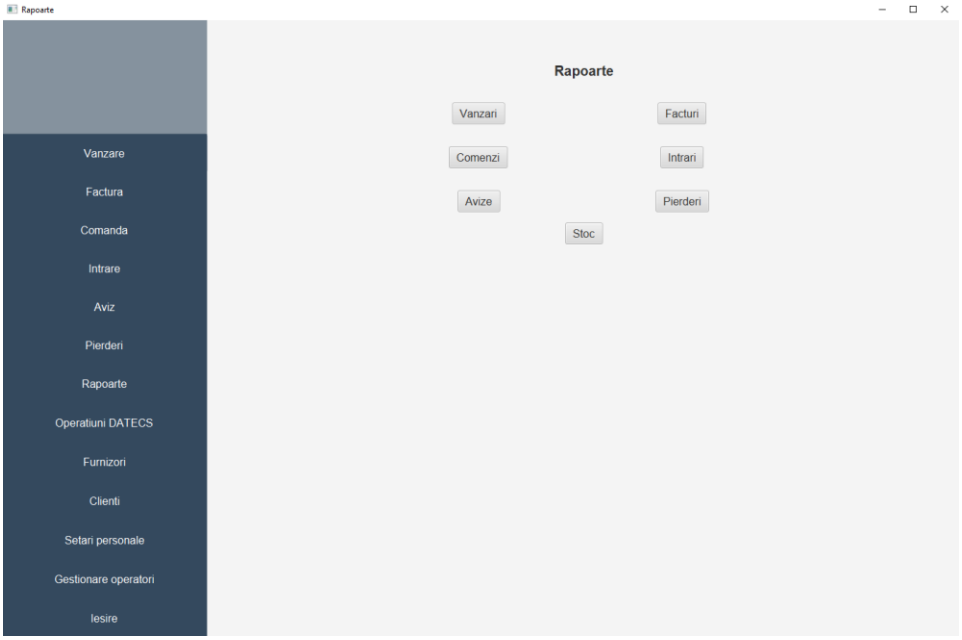

Figura 7.19 – Ecran selectare raport

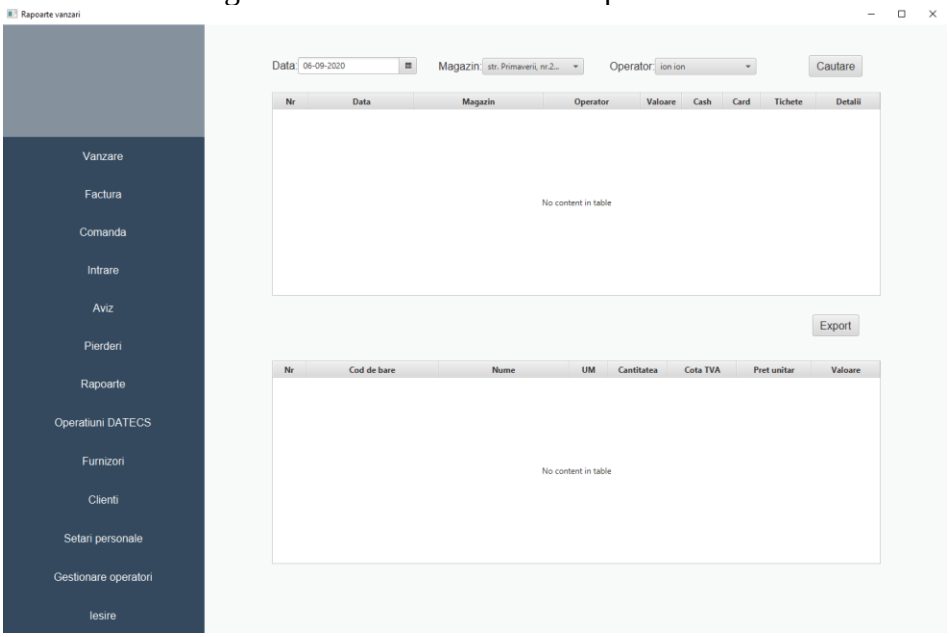

Figura 7.20 – Ecran raport vanzari

In cadrul raportelor de vanzare, poate fi efectuata cautarea dupa data, magazin si operator urmand sa fie afisate in cadrul primului tabel. Daca este apasat butonul detalii pentru o vanzare, produsele din vanzare vor fi afisate in cel de-al doilea tabel. Daca se doreste, vanzarile conform cautarii pot fi exportate in format .xlsx cu ajutorul butonului Export.

Functionalitatile sunt similare si pentru rapoartele de factura, comanda, intrare si avize, din acest motiv nu vor mai fi prezentate.

In cadrul rapoartelor pierderilor, poate fi efectuata cautarea dupa data inregistrarii pierderii si operatorul care a inregistrat pierderea precum si dupa numele produsului. La fel, raportul poate fi exportat in format .xlsx.

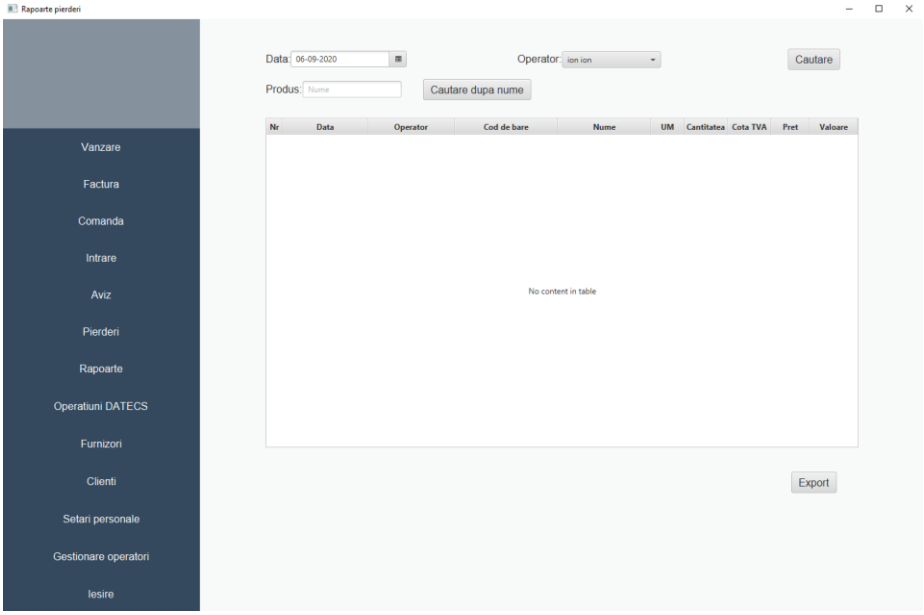

Figura 7.21 – Ecran raport pierderi

Pentru rapoartele dedicate stocului, cautarea poate fi facuta dupa numele produsului, magazin sau furnizor. Raportul poate fi exportat in format .xlsx, bineinteles. Rapoartele pot fi accesate si generate doar de catre gestionar si administrator.

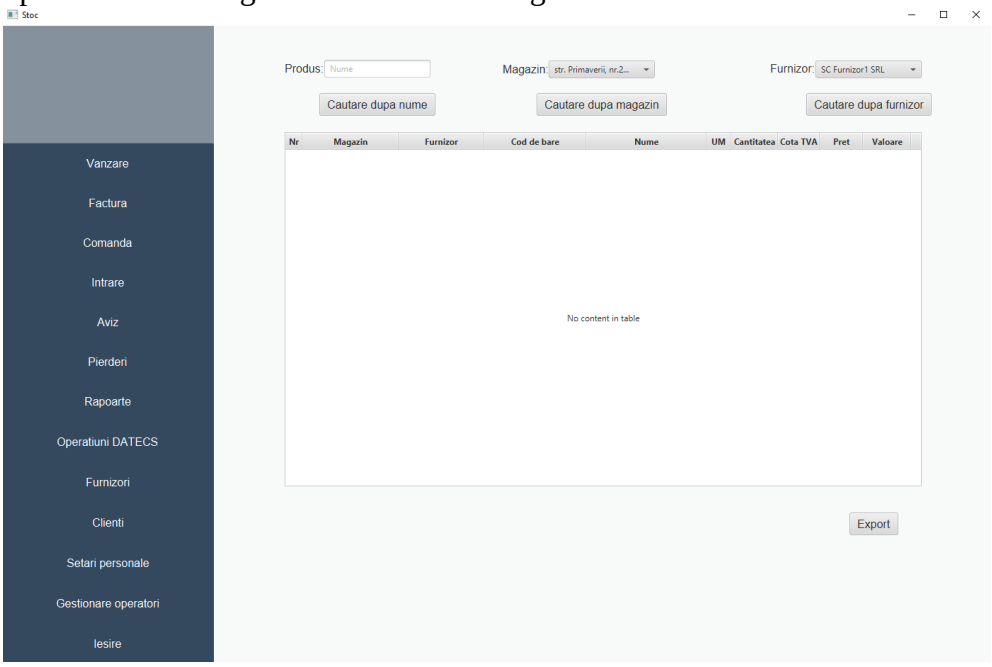

Figura 7.22 – Ecran raport stoc

7.2.11. Operaţiuni DATECS

In aceasta sectiune, operatorul poate introduce suma dorita si printr-un click specifica oepratiunea dorita: intrare sau iesire din casa de marcat. Operatia poate fi efectuata de orice tip de operator.

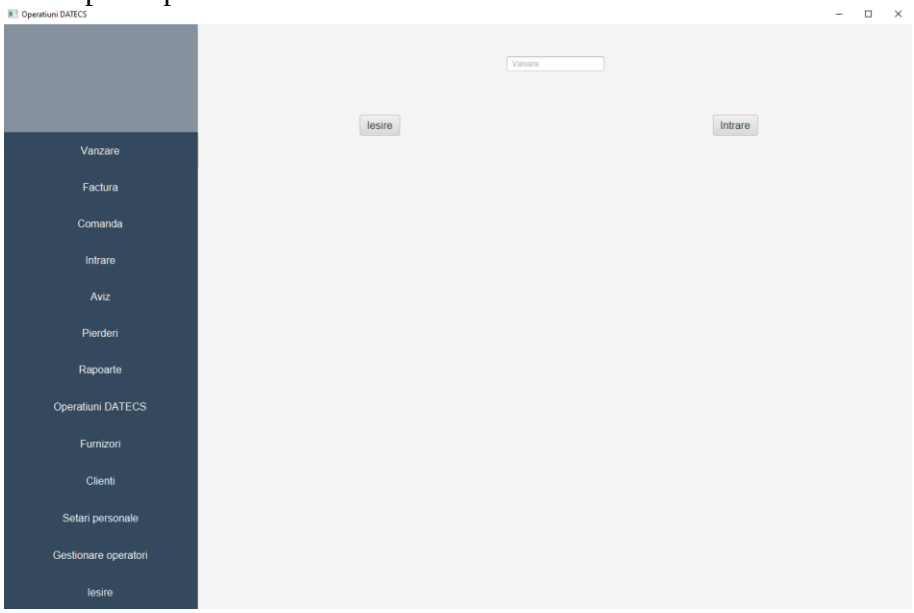

Figura 7.23 – Ecran operatiuni DATECS

7.2.12. Furnizori&Clienţi

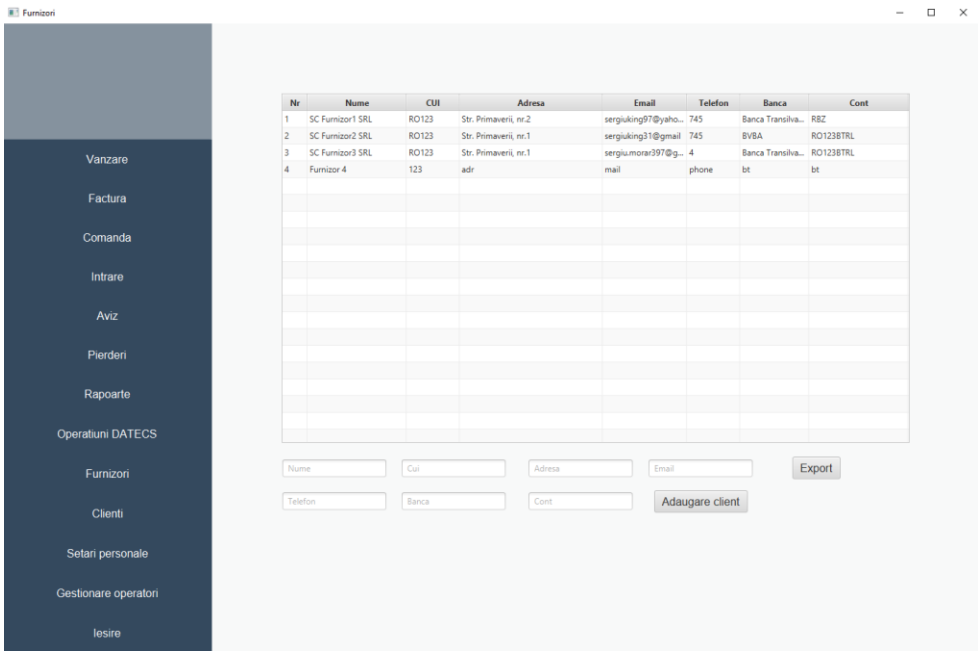

Figura 7.24 – Ecran furnizori

Operatorul poate vizualiza furnizorii, modifica anumite informatii, adauga furnizori noi sau exporta in format .xlsx furnizorii actuali.

Pentru clienti, operatiunile sunt similare si nu vor mai fi prezentate. Ambele operatiuni pot fi realizate de catre orice tip de operator.

7.2.13. Setari personale

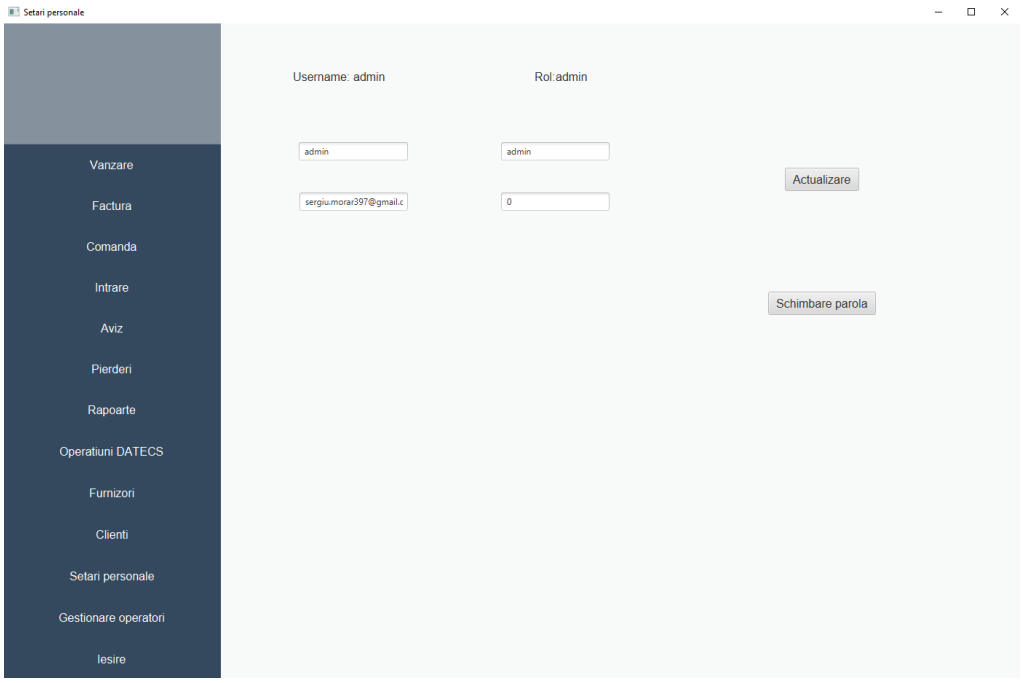

Figura 7.25 – Ecran setari personale

Utilizatorul aplicatiei isi poate modifica numele, adresa, adresa de email sau numarul de telefon. De asemenea isi poate schimba parola. Operatia poate fi efectuata de orice tip de operator.

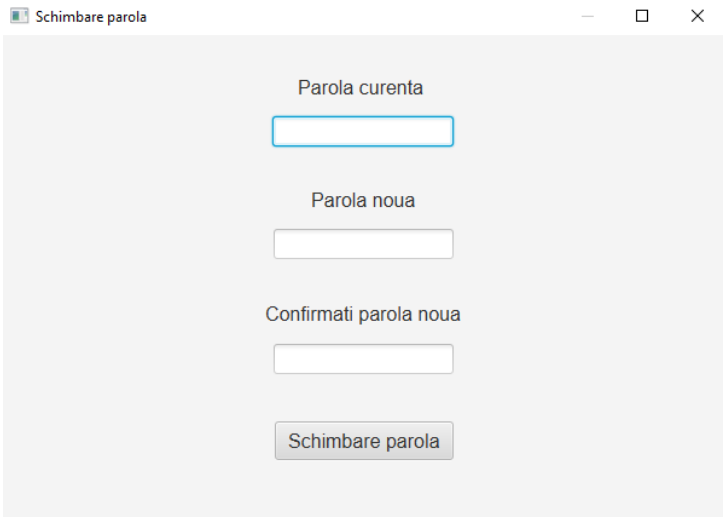

Figura 7.26 – Ecran schimbare parola

7.2.14. Gestionare operatori

Sectiune dedicata administratorilor pentru adaugarea de operatori noi, modificarea rolurilor, setarea statusului operatorului sau resetarea parolei. Operatia poate fi efectuata doar de catre administrator.

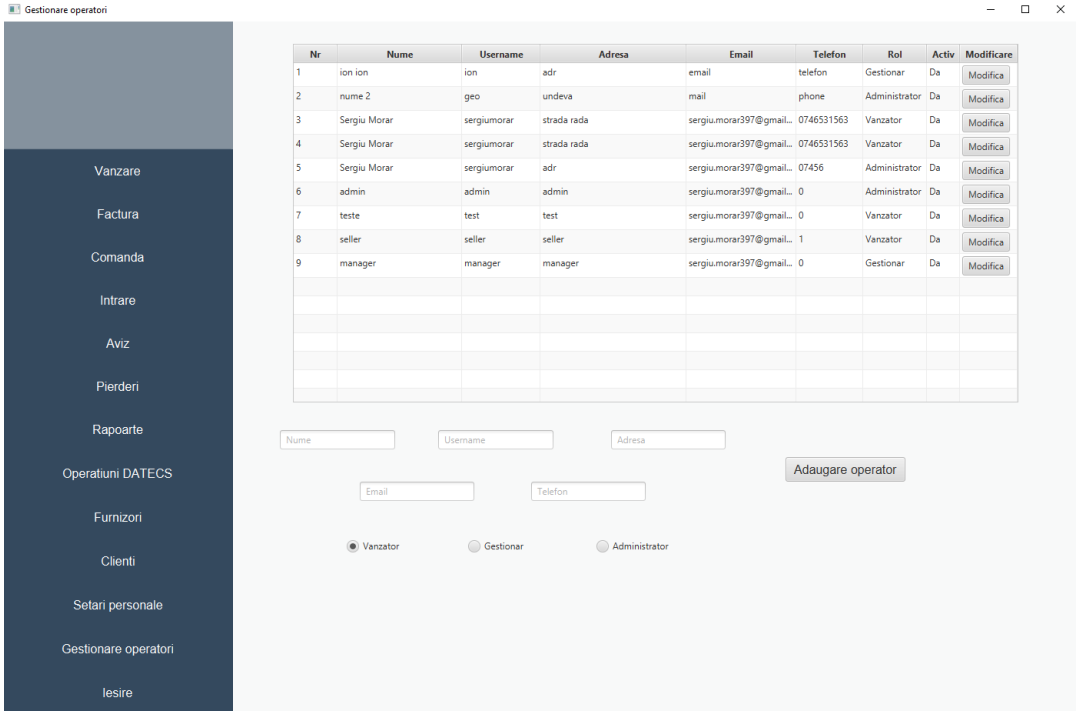

Figura 7.27 – Ecran gestionare operatori

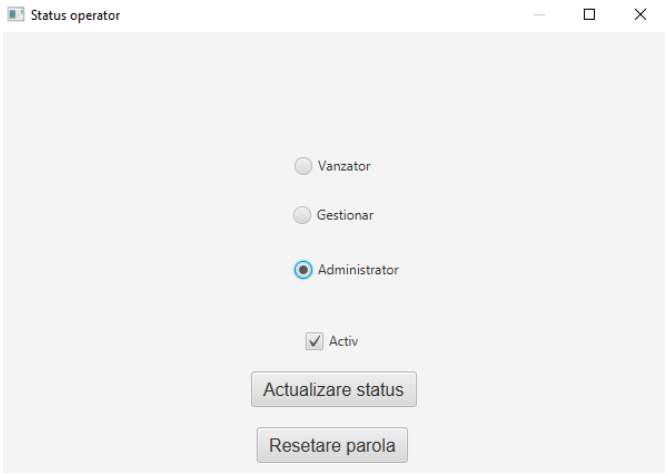

Figura 7.28 – Ecran modificare operator

Capitolul 8. Concluzii

Capitolul cu numarul 8 este dedicat concluzionarii lucrarii prezente, fiind ilustrata o imagine de ansamblu asupra aplicatiei dezvoltate. De asemenea, vor fi propuse si dezvoltari ulterioare.

8.1. Contribuție personală si rezultate

Ideea de a creea un astfel de sistem a izvorat in urma activitatii mele in cadrul magazinelor parintilor mei. Avand ocazia sa observ indeaproape tot ceea ce inseamna munca intr-un astfel mediu, am putut sa aflu informatii despre modul in care decurge activitatea, problemele care pot aparea, activitatile care consuma mult prea mult timp si multe altele.

De asemenea, in urma discutiilor avute am putut sa imi formez o idee despre ceea ce inseamna, care sunt neajunsurile si ce ar fi de dorit sa existe.

8.2. Dezvolăari ulterioare

In urma studiului bibliografic, discutiilor avute cu persoanele care lucreaza in domeniul vanzarilor, cat si a implementarii propriu-zise, printre imbunatatirile care pot fi aduse aplicatiei se numara:

- O sectiune dedicata alarmelor, sectiune in care operatorul poate crea o alarma referitoare la un produs din stoc, elementul declansator al alarmei putand fi momentul in care cantitatea din stoc ajunge sub un nivel mentionat, ori o anumita data calendaristica.
- Extinderea si spre platforma mobila, cu functionalitati reduse, de unde sa poata fi vizualizate rapoarte, create comenzi sau efectuare operatii referitoare la utilizatori.
- Oferirea de ajutor in ceea ce priveste partea de contabilitate, prin completarea automata de rapoarte sau declaratii.
- Introducerea in aplicatia a procedeului de plata in avans si stornare.
- Eliberarea de facturi proforme.

Bibliografie

[1] Mukesh Lal, „Study of Efectiveness of POS Data in Managing Supply Chain”, 2018.
Disponibil la: https://www.researchgate.net/publication/328406416_Study_of_Effectiveness_of_POS_Data_in_Managing_Supply_Chain

[2] Gheorghe Pistol, „Bazele Comertului”, 2004.
Disponibil la: https://www.academia.edu/18456775/120622917_Bazele_comertului

[3] Anabel Daniel, „The history of Cash Registers”, 2017.
Disponibil la: https://silo.tips/download/the-history-of-cash-registers

[4] Alfred Whitehead, „Evolution of barcode”, 2016.
Disponibil la: https://silo.tips/download/evolution-of-barcode

[5] Borrie Helen, „Introduction to Client/Server Architecture”, 2004.
Disponibil la: https://link.springer.com/chapter/10.1007/978-1-4302-0743-6_5

[6] Jim Arlow, „UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design”, 2002.
Disponibil la: https://www.researchgate.net/publication/234797700_UML_2_and_the_Unified_Process_Practical_Object-Oriented_Analysis_and_Design

[7] Craig Walls, „Spring in Action”, 2019.
Disponibil la: http://mocom.xmu.edu.cn/home/project/soft/Spring/Spring%20in%20Action,%205th%20Edition.pdf

[8] Herbert Schildt, „Introduction JavaFX 8 Programming”, 2015.
Disponibil la: https://link.springer.com/chapter/10.1007/978-1-4302-0743-6_5

[9] Stack Overflow Contributors, „Learning MySql”, 2019.
Disponibil la: https://www.computer-pdf.com/database/888-tutorial-learning-mysql.html

[10] Scott Oaks, „Java Security”, 2001.
Disponibil la: https://www.amazon.com/Java-Security-2nd-Scott-Oaks/dp/0596001576/ref=sr_1_1?dchild=1&keywords=Java+Security&qid=1599529081&sr=8-1

Anexa 1: Lista de figuri

Anexa 2: Lista de tabele