



Academia de Studii Economice din București
Facultatea de Cibernetică, Statistică și Informatică Economică
Specializarea Informatică Economică

Proiectarea unei aplicații informatice pentru gestiunea produselor unei firme

lucrare de licență

COORDONATOR:
Prof. univ. dr. ION LUNGU

ABSOLVENT:
Cornel-Ionuț BACIU

București 2018

CUPRINS

INTRODUCERE	3
CAPITOLUL 1. DESCRIEREA PROBLEMEI ECONOMICE	4
1.1 PREZENTAREA ACTIVITĂȚII ECONOMICE.....	4
1.2 STOCURILE SI EVALUAREA ACESTORA.....	5
1.2.1 Înregistrarea stocurilor	6
1.2.2 Ieșirea din gestiune.....	6
1.3 PREZENTAREA ACTIVITĂȚII INFORMATIZATE	6
CAPITOLUL 2. INSTRUMENTE INFORMATICE UTILIZATE.....	7
2.2 LIMBAJUL C#	7
2.3 MICROSOFT SQL SERVER	9
CAPITOLUL 3. ANALIZA ACTIVITĂȚII DE GESTIUNE.....	10
3.1 SPECIFICAREA CERINȚELOR INIȚIALE ALE APLICAȚIEI INFORMATICE.....	10
3.2 DIAGRAMA CAZURILOR DE UTILIZARE.....	11
3.2.1 Diagrama generală a cazurilor de utilizare	11
3.2.2 Diagrama cazului de utilizare pentru comenzi.....	12
3.2.3 Diagrama cazului de utilizare pentru gestionarea depozitului	13
3.2.4 Diagrama cazului de utilizare pentru aprovizionarea depozitului.....	15
3.3 DIAGRAMA DE CLASE.....	16
3.4 DIAGrame DE ACTIVITATE.....	18
3.4.1 Diagrama de activitate pentru comanda produselor	18
3.4.2 Diagrama de activitate pentru aprovizionarea depozitului	20
3.4.3 Diagrama de activitate pentru facturare	21
3.5 DIAGrame DE STARE	22
3.5.1 Diagrama de stare pentru comanda	22
3.5.2 Diagrama de stare pentru furnizori	22
3.6 DIAGRAMA DE SECVENȚE.....	23
3.6.1 Diagrama de secvență pentru comanda produsului.....	23
3.6.2 Diagrama de secvență pentru facturarea produsului	24
3.6.3 Diagrama de interacțiune pentru aprovizionare	25
3.7 CERINȚELE DETALIAȚE ALE APLICAȚIEI INFORMATICE.....	25
CAPITOLUL 4. PROIECTAREA APLICAȚIEI INFORMATICE	27
4.1 RAFINAREA DIAGAMELOR	27
Diagrama de clase rafinată	27
4.2 ARHITECTURA SISTEMULUI	28
4.3 PROIECTAREA BAZEI DE DATE	30
4.4 PROIECTAREA INTERFETELOR UTILIZATOR	32
CAPITOLUL 5. IMPLEMENTAREA APLICAȚIEI INFORMATICE.....	33
5.1 CREAREA BAZEI DE DATE	33
5.2 PREZENTAREA FUNCȚIONALITĂȚII SISTEMULUI	34
CONCLUZIE	45
BIBLIOGRAFIE	46
ANEXE	47

Introducere

În această lucrare științifică al cărei titlu este : “PROIECTAREA UNEI APLICAȚII INFORMATICE PENTRU GESTIUNEA PRODUSELOR UNEI FIRME” o să abordez doar o anumită parte din elementele implementate într-un sistem de gestiune a bazelor de date.

Deoarece bazele de date au o importanță majoră într-o întreprindere iar activitatea acesteia depinde de modul în care aceste date sunt manipulate, am decis să implementez un sistem care să poată fi utilizat de orice firmă care are în vedere simplificarea organizării sale din depozit și obținerea rezultatelor în timp util sau chiar în timp real.

Gestionarea produselor unei firme implică mai multe operații pe care un administrator le are de făcut și de îndeplinit pentru ca totul să fie bine organizat. Ne propunem realizarea unei aplicații desktop care va fi conectată la o bază de date și prin intermediul procedurilor vor fi realizate diferite operații de gestiune cum ar fi adăugarea documentelor, NIR, facturi. Totalitatea tranzacțiilor și operațiilor vor fi implementate în baza de date iar produsele vor fi adăugate și vor putea fi modificate și șterse prin intermediul unor proceduri din bază. Furnizorii vor putea fi introduși prin intermediul interfeței grafice direct în baza de date iar datele vor putea fi modificate ulterior. Odată ce produsele vor fi vândute, vor fi generate automat facturi care ulterior sunt stocate în baza de date pentru a oferi posibilitate accesării ulterioare și creării unor rapoarte pe baza acestora.

Această lucrare este structurată în patru capitole care abordează diferite aspecte ale proiectului.

În primul capitol sunt descrise toate aspectele și detaliile legate de tema aleasă și anume gestiunea produselor unei firme adică gestiunea unor stocuri.

Al doilea capitol conține detalii despre instrumentele informatice utilizate în vederea dezvoltării produsului software de gestiune.

Al treilea capitol este dedicat analizei activității de gestiune, acest capitol fiind mai cuprinzător, în cadrul căruia se va face o analiză amănunțită a aplicației. Prin intermediul diagramelor UML vor fi analizate toate procesele și componentele sistemului dar și modul de transmitere al informațiilor în sistem urmând într-un final să se realizeze activitatea de proiectare a sistemului.

Al patrulea capitol se continuă cu proiectarea aplicației și va prezenta aplicația și de asemenea va conține rafinarea diagramelor, proiectarea bazei de date si a interfețelor utilizator.

Ultimul capitol cuprinde implementarea aplicației și toate funcționalitățile acesteia împreună cu descrierea și modul în care aplicația de gestiune a echipamentelor sportive a fost implementată.

Capitolul 1. Descrierea problemei economice

1.1 Prezentarea activității economice

Este esențial ca într-o firmă, produsele din depozit să fie gestionate astfel încât cantitățile comandate pentru aprovizionare să mențină rezervele rămase la un nivel acceptabil. Toate operațiile de intrare, păstrare și ieșire care sunt realizate în depozit trebuie să fie făcute controlat astfel încât disponibilitatea produselor să fie rezonabilă pentru a îndeplini nevoile clienților, costurile să fie reduse iar oferta să fie în echilibru cu cererea.

Compania pentru care am ales să implementez această aplicație deține mai multe depozite care se află în locații diferite și al cărei obiect de activitate este comercializarea de echipamente și articole sportive pentru o gamă largă de clienți. Fie că sunt sportivi de performanță, oameni angajați care sunt pasionați de sport, copii, studenți sau cluburi sportive și săli de fitness cu toții își pot procura echipamentul dorit într-un timp foarte scurt datorită gestionării eficiente a depozitului.

Produsele sunt achiziționate de la furnizori și producători români dar și străini datorită prețului scăzut de cumpărare și distribuite mai departe către clienți.

Furnizorii sunt stocați în baza de date sub forma unei liste accesibile administratorului care are acces la sistem. De asemenea, în sistem, sunt stocate și datele clienților pentru o gestionare mai eficientă a comenzilor și generarea unor rapoarte în funcție de comenzile acestora. Facturile emise vor fi și ele stocate prin intermediul aplicației în baza de date.

Accesul la aplicație și implicit la baza de date se poate face prin intermediul a două tipuri de autentificare. Primul, destinat administratorului depozitului, care se ocupă cu aprovizionarea acestuia și de unde poate manipula toate datele și poate avea acces la setările programului, adică are drepturi exclusive de utilizare și al doilea mod de autentificare destinat angajaților. Administratorul va adăuga un utilizator nou sau îl va șterge și se va ocupa de conturile acestora. Angajații vor avea acces doar la anumite tabele din baza de date cu scopul de a informa clientul.

“SGBD este componenta software a unui sistem de bază de date care interacționează cu toate celelalte componente ale acestuia, asigurând legătura și interdependența între elementele sistemului.”¹

O bază de date, conține date centralizate care sunt stocate cu scopul de a fi utilizate prin intermediul unei aplicații în scopul regăsirii anumitor date relevante pentru utilizator.

1.2 Stocurile si evaluarea acestora

“Stocurile reprezintă ansamblul bunurilor și serviciilor din cadrul întreprinderii deținute fie pentru a fi vândute în aceeași stare sau după prelucrarea lor în procesul de producție, fie pentru a fi consumate la prima lor utilizare”.²

Așa cum un stat își face rezerve și depozitează mâncare, medicamente pentru situațiile limită prin care poate trece o țară cum ar fi o calamitate sau un război, o întreprindere poate alege să facă rezerve de produse, adică stocuri de mărfuri pentru a minimiza cheltuielile de organizare și a asigura eficiența din punct de vedere economic.

“Pentru fiecare material sau produs stocat se întocmește o fișă pe care se înscriu pe parcurs intrările și ieșirile (precizând destinația lor pentru a cunoaște costul vizat). Aceasta permite cunoașterea nivelului existent și verificarea în vederea evitării rupturilor de stocuri.”³

Valorizarea acestor deplasări de stocuri reprezintă o problemă care necesită o atenție sporită din partea contabilității. De asemenea stocurile sunt resurse economice care sunt deținute și controlate de către entitate cu scopul de a fi comercializate către clienți.

Materiile prime și cele consumabile fac parte din categoria stocurilor precum și plățile în avans pentru achiziționarea lor dar și produsele finite și mărfurile. Mărfurile sunt reprezentate de bunuri care au fost achiziționate de o entitate pentru a fi revândute.

¹ http://bd.ase.ro/uploads/sghd_curs/materialStiduuSGBDoracle.pdf

² Conf.univ.dr.Chirița CARAIANI(coord), Conf.univ.dr.Lavinia OLIMID(coord) – “*Bazele contabilității*”

³ Maria NICULESCU, Alain BURLAUD – “Contabilitate de gestiune”, Editura ECONOMICA, 2002

1.2.1 Înregistrarea stocurilor

Înregistrarea stocurilor se face după anumite reguli, pentru a fi în conformitate cu normele și reglementările din România și pentru a păstra situația din contabilitate transparentă. La intrarea în gestiune, produsele vor intra în funcție de tipul de achiziție al acestora, la prețul de producție sau prețul de achiziție.

Evaluarea realizată la momentul intrării în gestiune se face la prețul de achiziție în cazul materialelor care sunt cumpărate din exterior și la prețul de producție pentru acele produse

1.2.2 Ieșirea din gestiune

La ieșirea din gestiune, produsele vor avea adăugat un adaos comercial la prețul de achiziție înregistrat la momentul intrării în gestiune. În acest moment, contabilitatea va avea responsabilitatea să facă scăderea din gestiune la valoarea înregistrată la momentul intrării. La ieșirea din patrimoniu scăderea din gestiune se poate face prin metode precum sunt costul mediu ponderat, metoda LIFO (ultimul intrat – primul ieșit) sau FIFO (primul intrat – primul ieșit). Metoda costului mediu ponderat este cea mai utilizată și se calculează la sfârșitul fiecărei perioade de calcul cu ajutorul valorilor stocurilor și intrărilor dar și a cantităților intrate și inițiale.

1.3 Prezentarea activității informatizate

Lucrarea își propune să trateze gestiunea stocurilor unui depozit cu articole sportive într-o manieră cât mai simplă, rapidă, eficientă și securizată în același timp. Pentru utilizarea aplicației nu este nevoie de personal specializat, perioada de învățare pentru a gestiona procesele prin intermediul acesteia fiind extrem de mică.

Persoanele care lucrează în interiorul depozitului au dreptul doar la interogarea datelor iar managerul general și administratorul de sistem au mai multe drepturi cum ar fi gestionarea furnizorilor, produselor și actualizarea stocurilor. Pentru clienții fideli care achiziționează produse pentru anumite evenimente sau pentru propria afacere managerul general are dreptul de a negocia prețul de vânzare, și a consulta furnizorii în cazul în care stocul nu permite onorarea comenzii. Pot fi aplicate reduceri în cazul în care timpii de așteptare sunt prea mari sau cantitățile sunt însemnate.

Operațiile la nivel de stoc sunt realizate doar de administratorul de sistem care are drepturi exclusive. Accesul în aplicație se realizează foarte ușor pe baza datelor de autentificare care sunt

asociate cu drepturi de utilizare în aplicație și care la un interval de timp sunt schimbate. Toate acestea sunt gestionate tot de administratorul bazei de date.

Capitolul 2. Instrumente informatice utilizate

Am ales să folosesc C# pentru că este un limbaj orientat pe obiect ,simplu, modern, care permite accesul la baza de date foarte ușor și gestionarea lor. De asemenea este integrat destul de bine în Windows ceea ce facilitează introducerea unui soft pe piață deoarece mulți dintre clienți folosesc acest sistem de operare.

Am ales să folosesc o bază de date este de tip Microsoft SQL SERVER deoarece este ideală pentru stocarea unei cantități mari de date și poate realiza un backup fără prea mult timp și efort investit.

2.2 Limbajul C#

Limbajul C# este un limbaj simplu, modern, orientat pe obiect dezvoltat în 2000 de către Microsoft. În timp acesta a căpătat popularitate, în special datorită brandului Microsoft, fiind un rival limbajului Java care inițial nu a acceptat modificări și astfel luând naștere C#. Este ușor de citit, se poate învăța ușor iar gestionarea memoriei se face automat prin intermediul garbage collectorului.

Limbajul este folosit pentru a crea jocuri video folosind motorul Unity care este cel mai popular motor pentru crearea de jocuri. Mai mult de 3 sferturi din jocuri sunt făcute cu acesta și de asemenea este folosit și pentru Virtual Reality. Este un limbaj extraordinar pentru cei care vor să învețe programare și să pătrundă mai departe în sfera dezvoltării de jocuri video și realitate virtuală.

C# este o implementare bogată a paradigmei orientate pe obiect, care include încapsulare, moștenire și polimorfism. Încapsularea înseamnă crearea unei legături în jurul unui obiect, pentru a separa comportamentul extern (public) de cel intern (private) adică detaliile de implementare. Caracteristicile distinctive ale C# din perspectiva limbajului orientat pe obiecte sunt:

- **Sistem tip unificat**

Blocul fundamental de construcție din C# este o unitate încapsulată de date iar C# are un sistem de tip unificat, unde toate tipurile în cele din urmă împărtășesc un tip de bază comun. Aceasta înseamnă că toate tipurile de date, indiferent dacă reprezintă obiecte ale clasei sau sunt tipuri primitive, cum ar fi numere, au la bază același set de funcționalitate. De exemplu, orice tip poate fi convertit într-un șir de caractere prin apelarea metodei `ToString`.

- **Clase și interfețe**

O clasă, în programarea orientată pe obiect, reprezintă o colecție de obiecte care împărtășesc aceleași atribute și operații adică implementează anumite funcții și operatori specifici.

O interfață este ca o clasă, cu excepția faptului că este doar o formă pentru o clasă și nu o implementare. Este deosebit de util în scenarii în care sunt necesare moșteniri multiple (spre deosebire de limbile cum ar fi C++ și Eiffel, C# care nu susțin moștenirea mai multor clase).

- **Proprietăți, metode și evenimente**

În paradigma pur orientată spre obiect, toate funcțiile sunt metode. În C#, metodele sunt doar un fel de membru al funcției, care, de asemenea, include proprietăți și evenimente. Proprietățile sunt membri ai funcțiilor care încapsulează o parte din starea obiectului, cum ar fi culoarea unui buton sau textul unei etichete. Evenimentele sunt membri ai funcțiilor care simplifică acțiunea pe obiect schimbări de stat.

C# se bazează pe timpul de execuție pentru a efectua gestionarea automată a memoriei. CLR are un colector de gunoi care se execută ca parte a programului reclamând memoria pentru obiecte care nu mai sunt referite. Acest lucru îi scutește pe programatori de dezalocarea explicită a memoriei pentru un obiect, eliminând problema pointerilor incorecți întâlnit în limbi precum C++.

2.3 Microsoft SQL SERVER

Între 1995 și 2016 au fost lansate 10 versiuni de SQL Server iar în decursul anilor au fost incorporate diferite unelte pentru gestionarea și analiza datelor dar și suportul pentru noi tehnologii cum ar fi cele din domeniul web, stocare de tip cloud și servicii de tip mobile. Este un produs software care stochează și interoghează datele cerute de utilizator prin intermediul unei aplicații. MS Server are un avantaj enorm spre deosebire de alte baze de date, integrându-se foarte bine cu tehnologiile Microsoft cum sunt aplicațiile dezvoltate în .NET.

SQL Server este stabil din punct de vedere al securității având o protecție sporită în momentul autentificării protejând foarte bine parolele și datele clienților precum sunt cele legate de conturi bancare, și impunând schimbarea frecventă a acestora. Deoarece orice vulnerabilitate a unei organizații poate avea urmări catastrofale, produsul pune la dispoziție două modele de autentificare, SQL și Windows. SQL Server are implementată o funcționalitate nativă pentru criptarea datelor și anume criptarea unei singure coloane dintr-o tabelă cum este spre exemplu codul numeric personal al unei persoane.

“Programul include 2 interfețe care permit utilizatorului să gestioneze și să dezvolte în SQL Server.”⁴

SQL Server pune la dispoziție patru tehnologii care asigură consistența și valabilitatea datelor.

Una dintre ele este database mirroring care menține două copii ale bazei de date care rulează pe două servere diferite. Log shipping este o tehnologie care trimite logul unei baze de date către un server secundar dinspre serverul principal.

MS SERVER a fost proiectat pentru a concura cu MySQL și baze de date Oracle și suportă ANSI SQL acesta fiind limbajul standard. Sistemul are capacitatea de a transfera informații între calculatoare care sunt conectate la aceeași rețea. Viteza cu care programul procesează informațiile este de asemenea importantă mai ales atunci când vine vorba de manipularea unor date voluminoase.

Până în anul 2016 MS Server a fost proiectat doar pentru platforma Windows iar din anul 2017 acesta a fost implementat și pentru Linux care este un sistem de operare open source cu scopul de a stabili legături cu clienții care nu folosesc Windows.

⁴ <https://mitseu.files.wordpress.com/2014/08/microsoft-sql-server-2012-step-by-step-pre-press.pdf>

Capitolul 3. Analiza activității de gestiune

3.1 Specificarea cerințelor inițiale ale aplicației informatice

Entitatea pentru care voi implementa aplicația informatică își desfășoară activitatea în comercializarea de produse și articole sportive.

1. Accesul la aplicație este implementat astfel încât autentificarea să fie făcută în sistem cu un nume și o parolă unică care este criptată înainte de a fi stocată în baza de date.

2. Produsele vor putea fi adăugate, șterse sau modificate în baza de date a firmei, această operație fiind accesibilă doar pentru manager care are drepturi asociate contului său.

3.Separat este afișată o listă cu toți furnizorii disponibili in sistem dar și o secțiune separată pentru a adăuga și modifica datele acestora.

4.Dupa adăugarea produselor într-o listă am implementat o secțiune cu un eveniment care poate genera automat o factură cu datele furnizorului sau clientului.

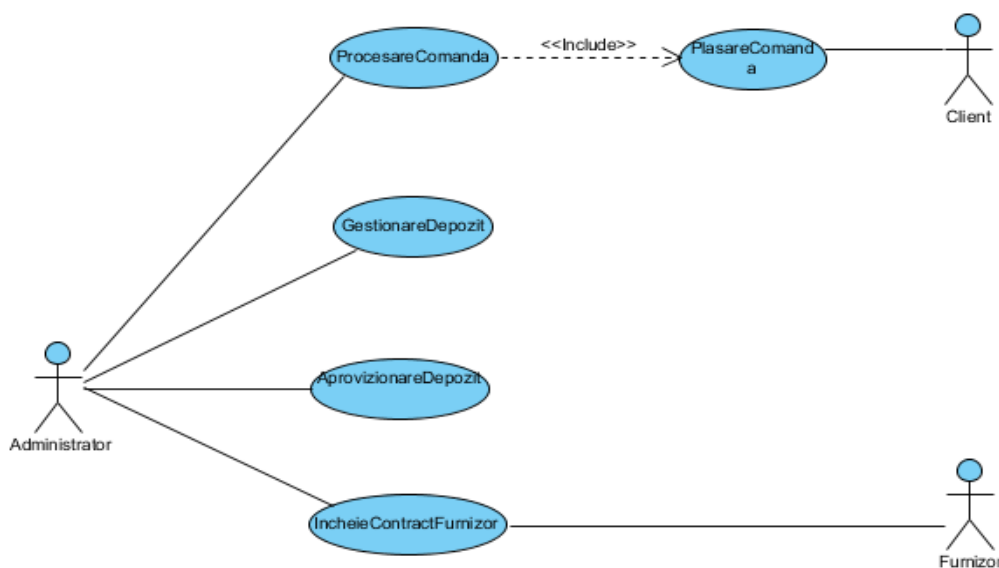
5.Totate facturile din sistem pot fi listate într-un formular de unde utilizatorul poate vedea toate detaliile acestora.

6.Vor exista rapoarte care vor pune în evidență anumite rezultate relevante pe baza datelor existente in baza de date.

3.2 Diagrama cazurilor de utilizare

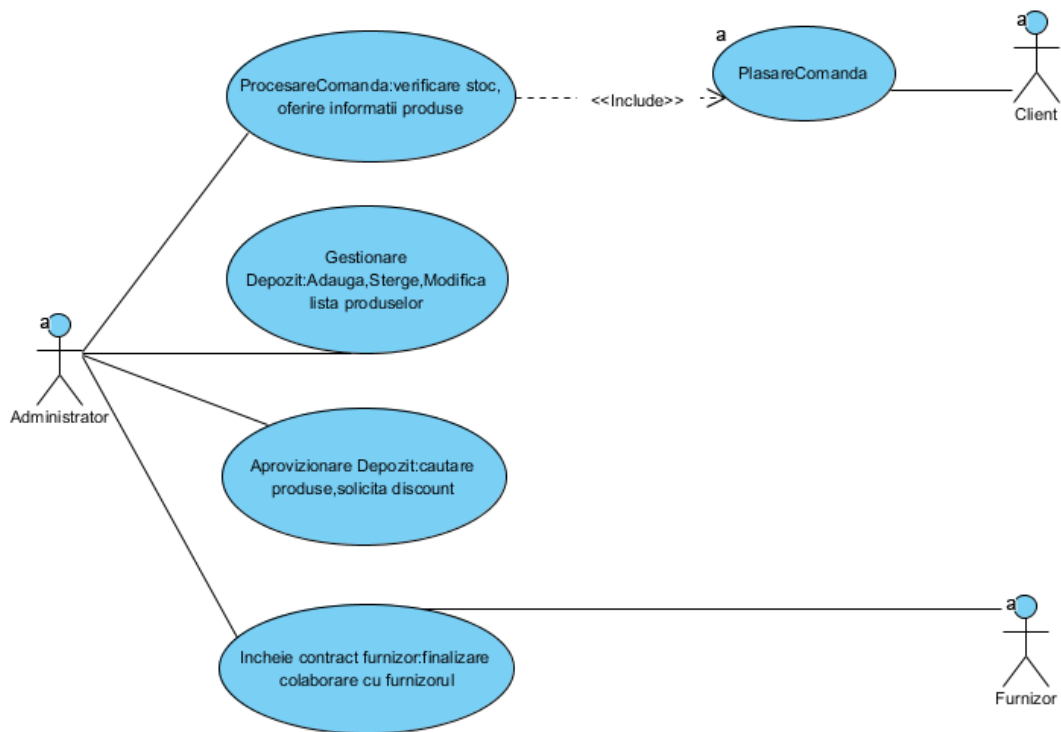
3.2.1 Diagrama generală a cazurilor de utilizare

Am indentificat si reprezentat cerințele funcționale prin intermediul diagramelor cazurilor de utilizare unde părțile interesate sunt reprezentate prin intermediul actorilor iar acțiunile care se doresc a fi întreprinse sunt exprimate prin intermediul cazurilor de utilizare.



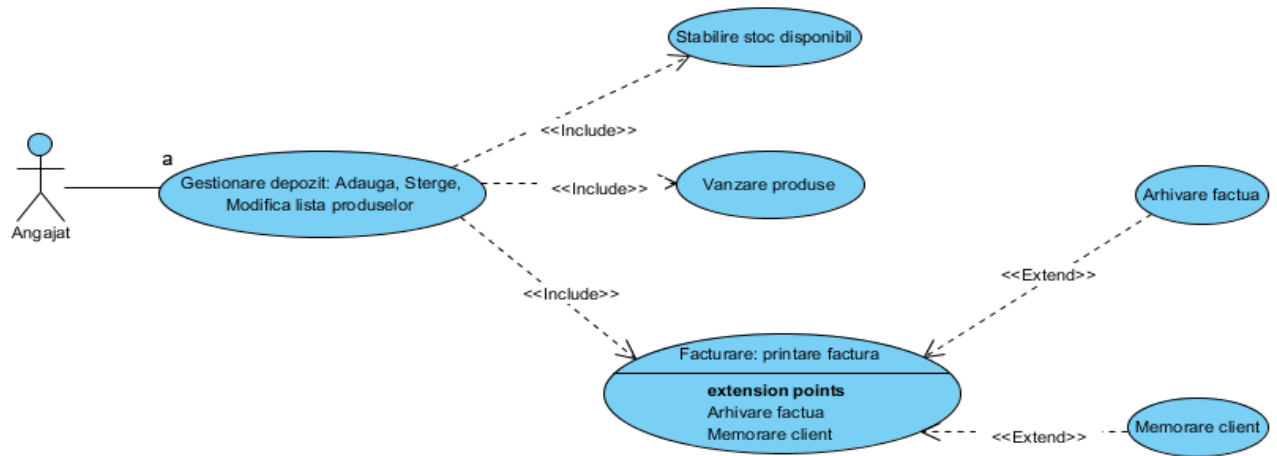
Figură 1 - Diagrama generală a cazurilor de utilizare

3.2.2 Diagrama cazului de utilizare pentru comenzi



Figură 2 - Diagrama cazului de utilizare pentru comenzi

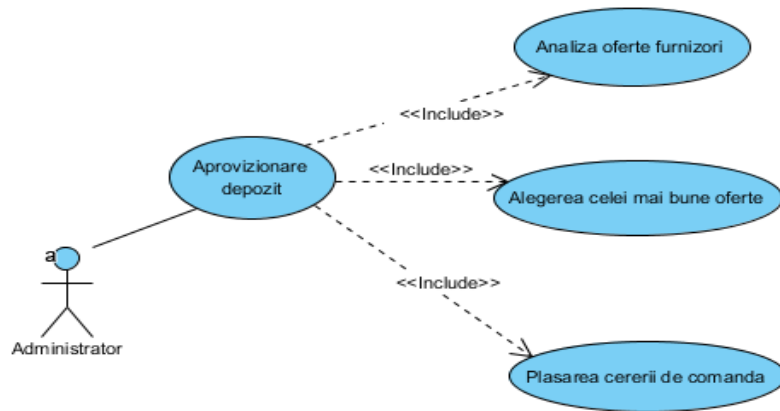
3.2.3 Diagrama cazului de utilizare pentru gestionarea depozitului



Figură 3 - Diagrama cazului de utilizare pentru gestionarea depozitului

Elemente ale cazului de utilizare	Descriere
Cod	CU01
Stare	Schiță
Scop	Gestionarea depozitului
Nume	Gestionare depozit
Actor Principal	Administratorul
Descriere	Controlul tuturor activităților de gestionare a produselor disponibile în stoc, vânzarea de produse, facturare, stocare în baza de date a tuturor tranzacțiilor
Precondiții	Administratorul trebuie să se conecteze la aplicație prin intermediul unui cont unic de utilizator și o parolă.
Postcondiții	Întocmirea facturii ce atestă încheierea unei vânzări de produse
Declanșator	Administratorul primește sarcina de a realiza o activitate ce aparține de gestionarea depozitului
Flux de baza	1)Administratorul se conectează 2)Stabilește stocul disponibil 3)Vinde produsele către client 4)Întocmește factura ce atestă încheierea tranzacției cu clientul 5)Introduce factura și clientul în baza de date
Fluxuri Alternative	Se poate încheia o comandă și dacă resursele nu sunt disponibile ,dacă clientul acceptă să intre în posesia materialelor cu o întârziere, până furnizorul va aproviziona depozitul
Relații	Include, Extinde
Frecvența Utilizării	Foarte utilizată
Reguli ale afacerii	Administratorul poate face reduceri neprevăzute dacă clientul dorește o comandă foarte mare, iar afacerea este profitabilă.

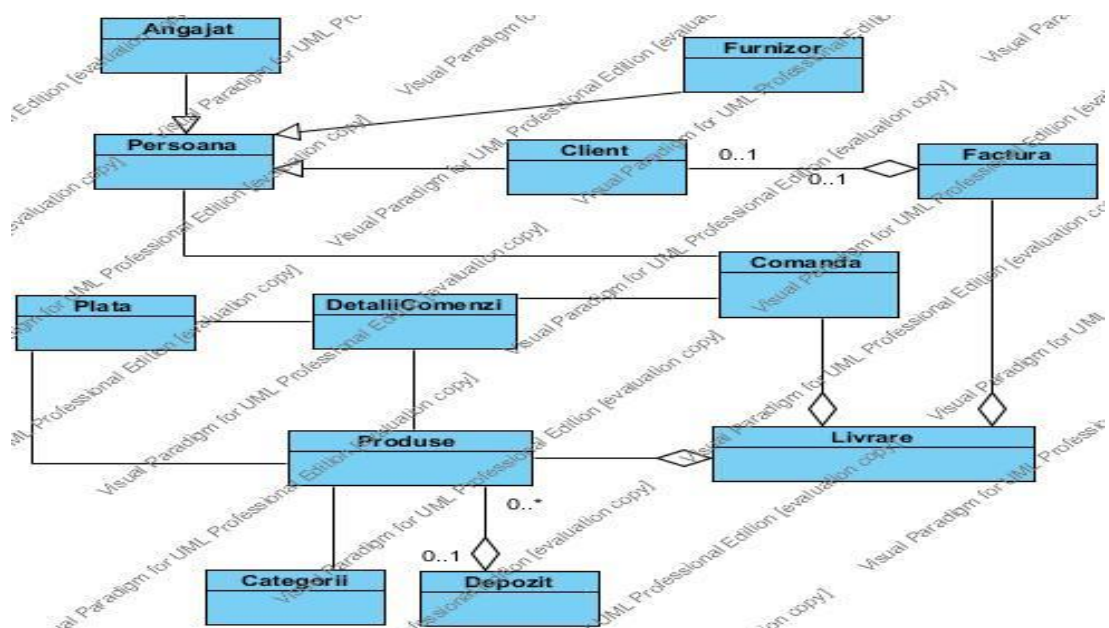
3.2.4 Diagrama cazului de utilizare pentru aprovizionarea depozitului



Figură 3 - Diagrama cazului de utilizare pentru aprovizionarea depozitului

Elemente ale cazului de utilizare	Descriere
Cod	CU02
Stare	Schiță
Scop	Aprovizionarea cu material
Nume	Aprovizionare
Actor Principal	Administratorul
Descriere	Aprovizionarea cu materii de la furnizori
Precondiții	Administratorul trebuie să se conecteze la aplicație prin intermediul unui cont unic de utilizator și o parolă.
Postcondiții	Încheierea unui contract cu furnizorul
Declanșator	Atunci când administratorul trebuie să aprovizioneze depozitul cu materiale
Flux de baza	1)Administratorul se conectează 2)Analizează ofertele furnizorilor 3)Alege cea mai bună ofertă 4)Plasează comanda
Fluxuri Alternative	Se poate realiza un contract cu plata întârziată dacă furnizorul este de acord
Relații	-
Frecvența Utilizării	Mediu
Reguli ale afacerii	Administratorul trebuie sa aleagă cea mai bună ofertă în raport calitate preț când trebuie să aprovizioneze depozitul

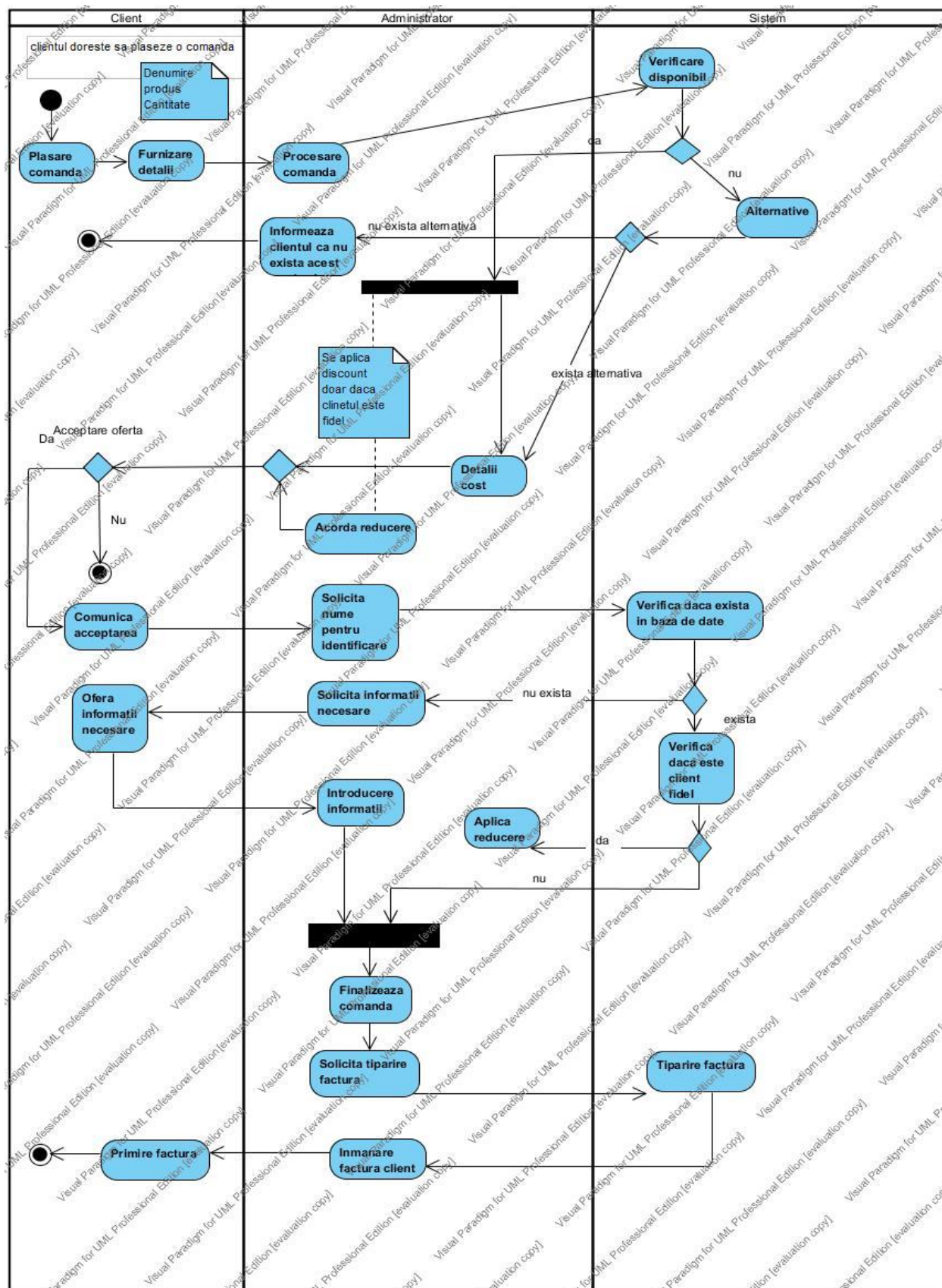
3.3 Diagrama de clase



Figură 4 - Diagrama de clase

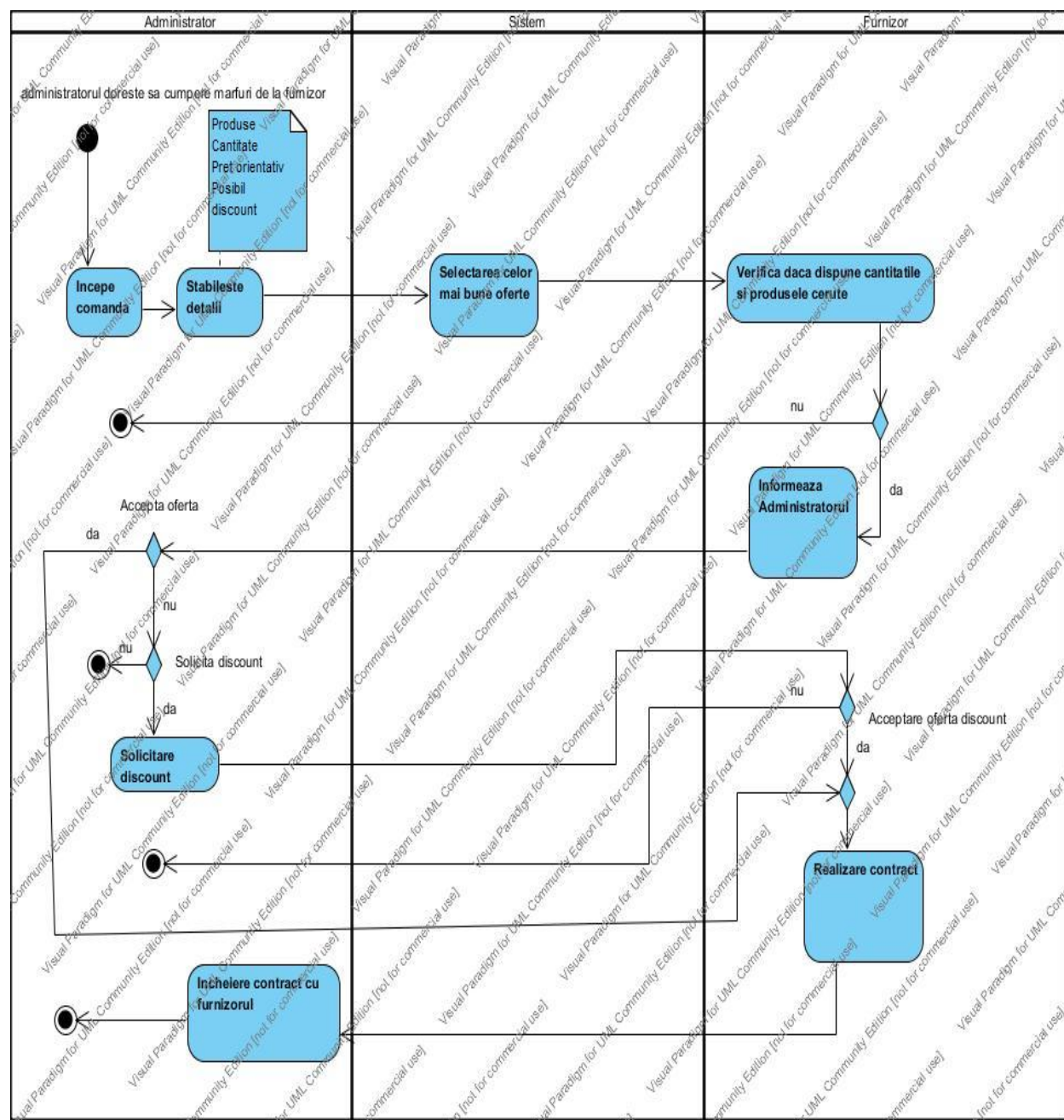
3.4 Diagrame de activitate

3.4.1 Diagrama de activitate pentru comanda produselor



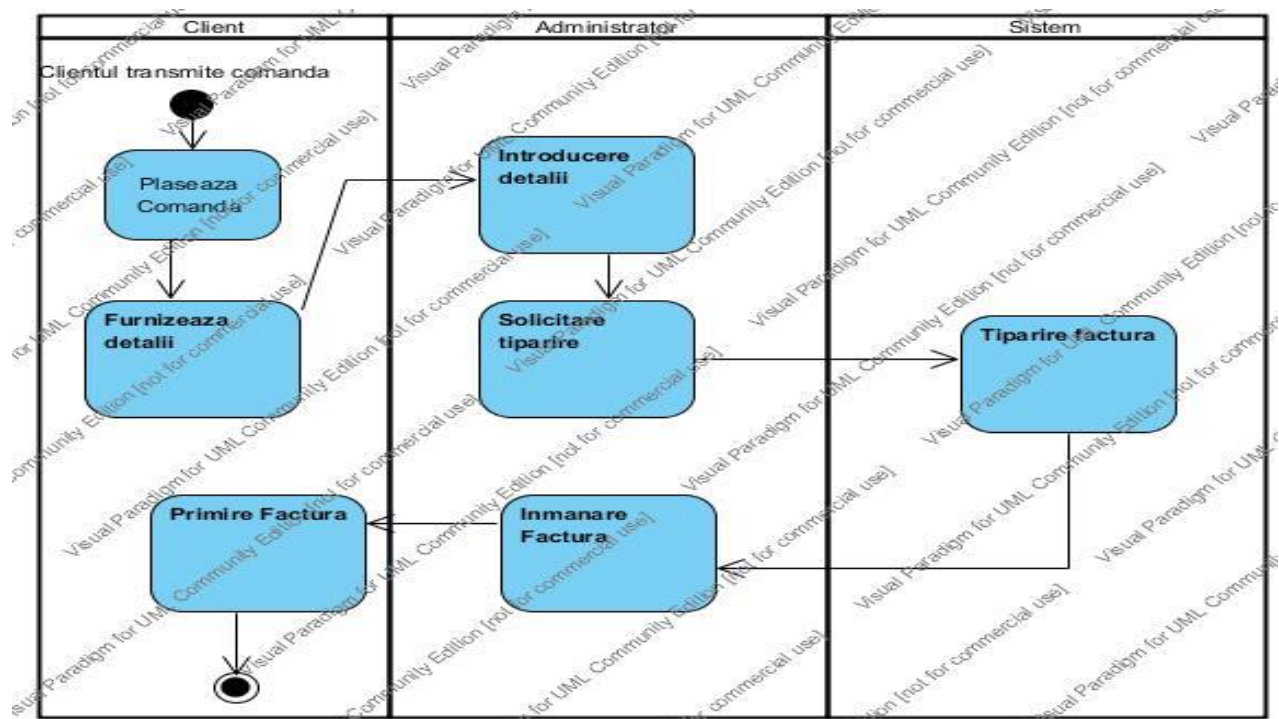
Figură 5 - Diagrama de activitate pentru comanda produselor

3.4.2 Diagrama de activitate pentru aprovizionarea depozitului



Figură 6 - Diagrama de activitate pentru aprovizionarea depozitului

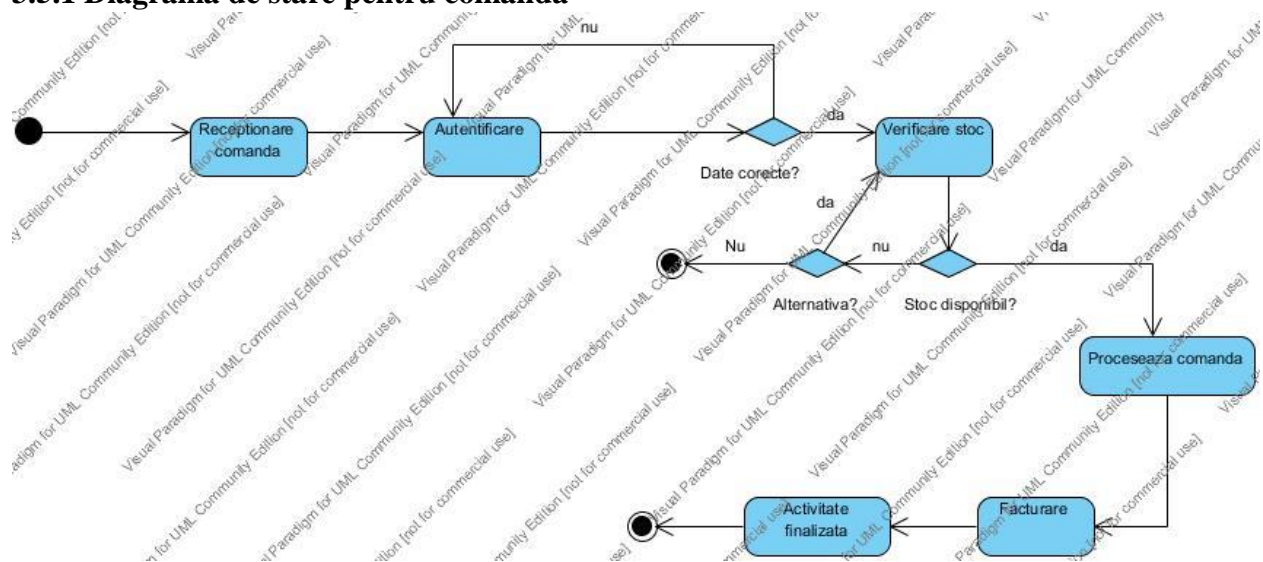
3.4.3 Diagrama de activitate pentru facturare



Figură 7 - Diagrama de activitate pentru facturare

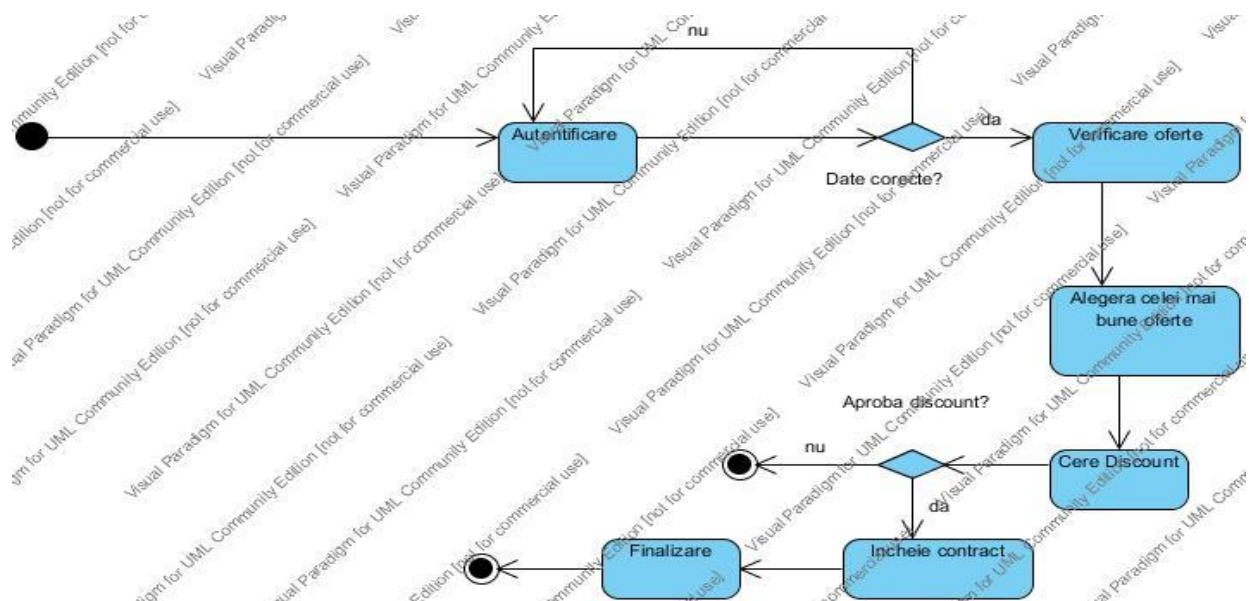
3.5 Diagrame de stare

3.5.1 Diagrama de stare pentru comanda



Figură 8 - Diagrama de stare pentru comanda

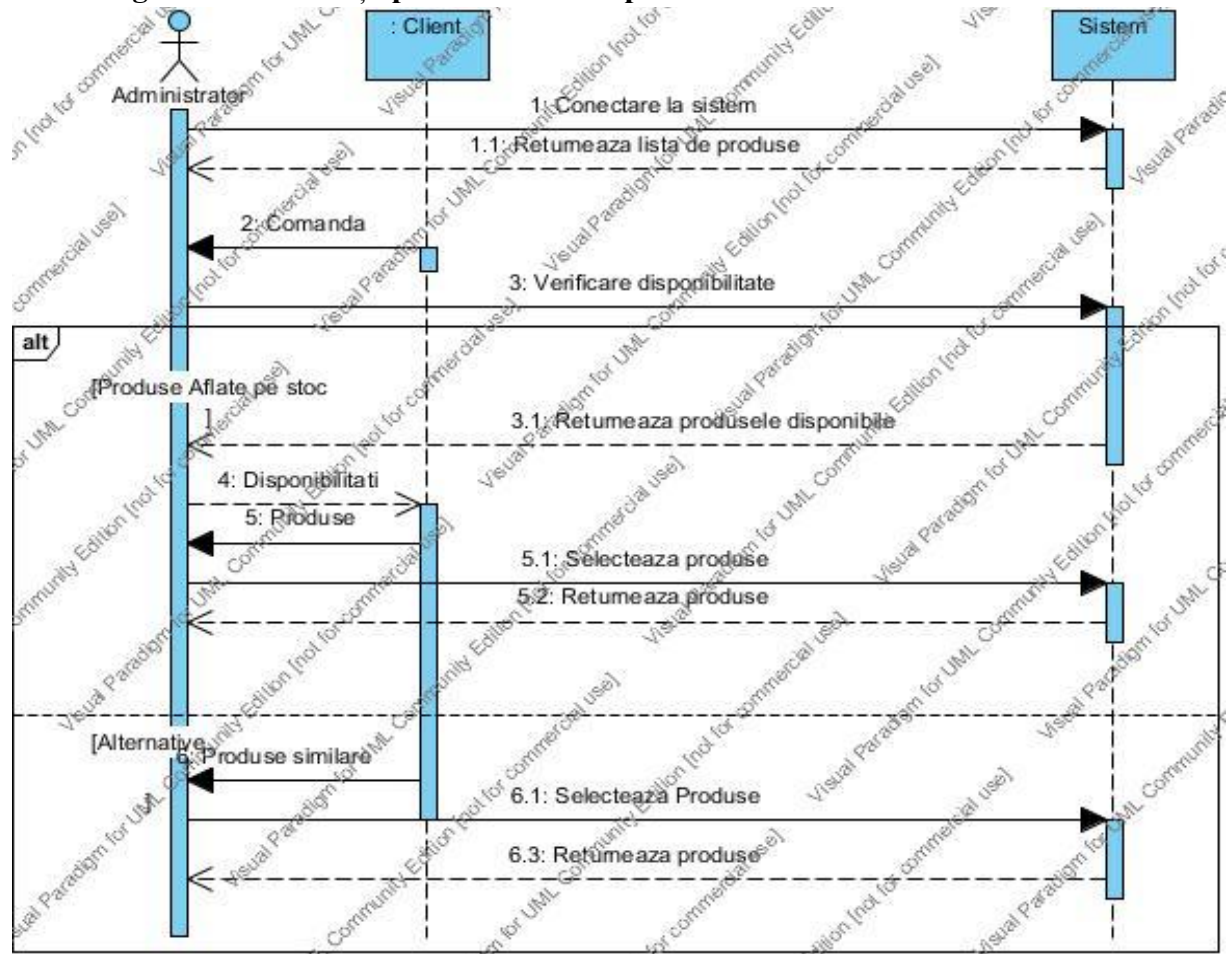
3.5.2 Diagrama de stare pentru furnizori



Figură 9 - Diagrama de stare pentru furnizori

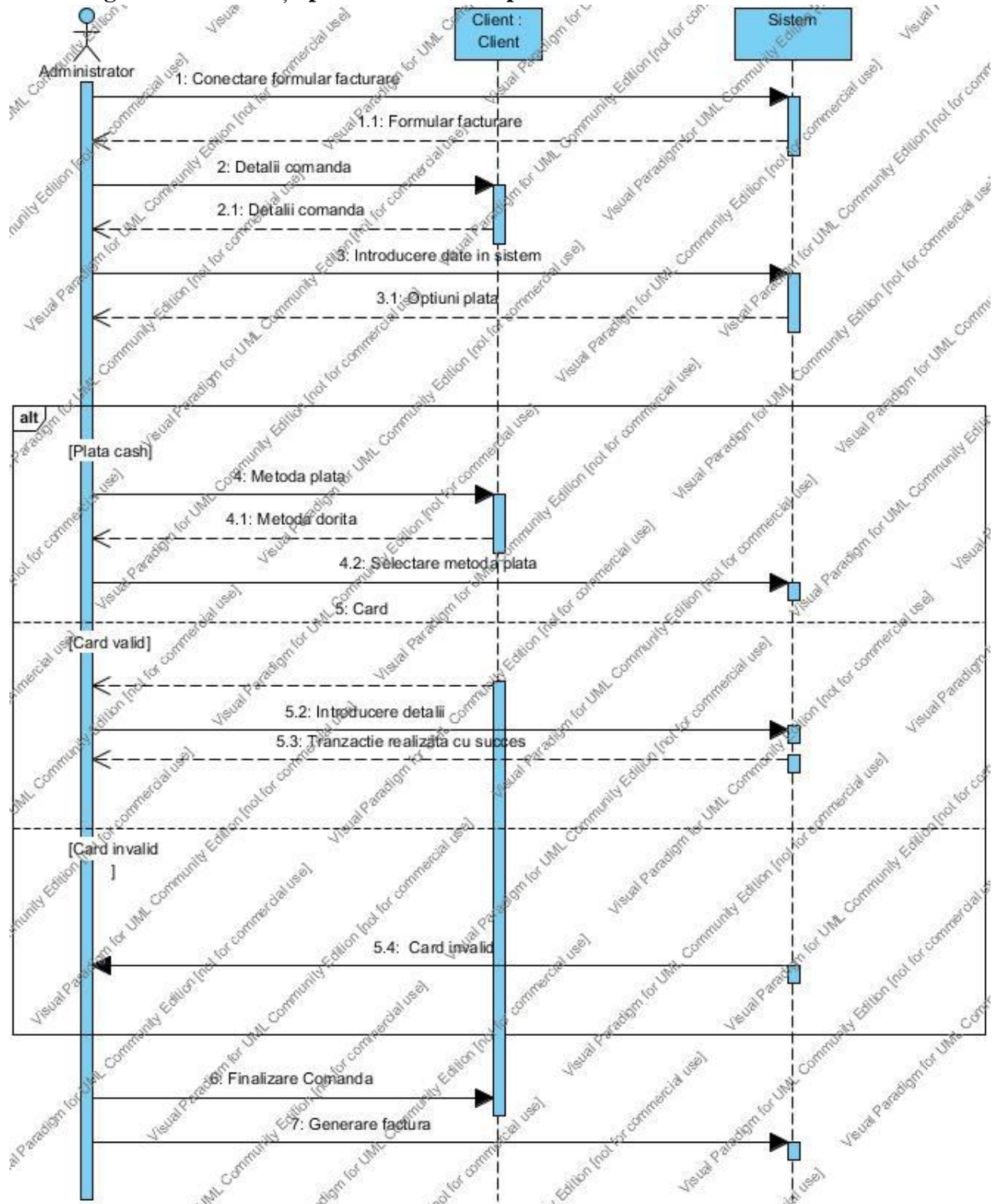
3.6 Diagrama de secvențe

3.6.1 Diagrama de secvență pentru comanda produsului



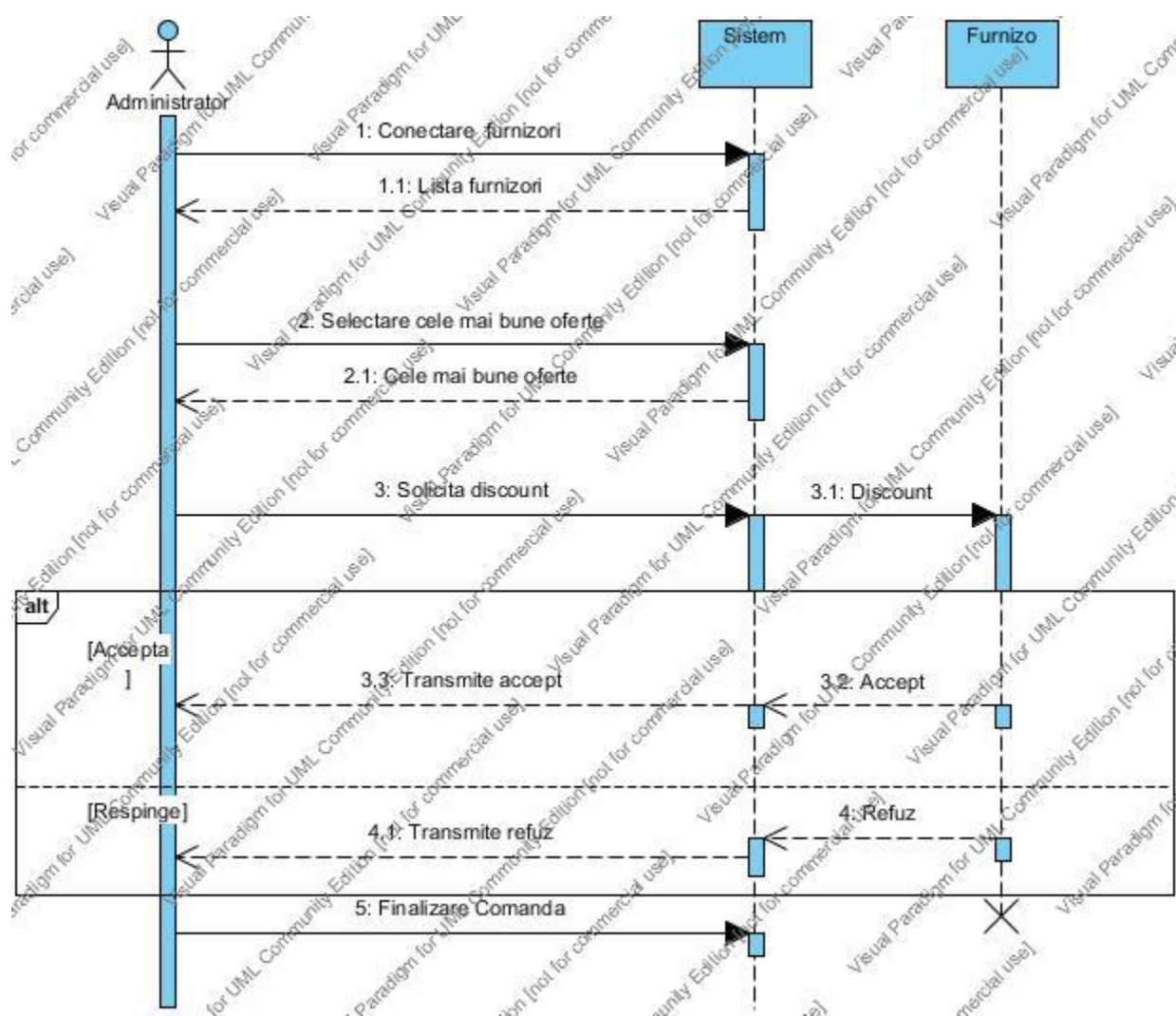
Figură 10 - Diagrama de secvență pentru comanda produsului

3.6.2 Diagrama de secvență pentru facturarea produsului



Figură 11 - Diagrama de secvență pentru facturarea produsului

3.6.3 Diagrama de interacțiune pentru aprovizionare



Figură 11 - Diagrama interacțiune pentru aprovizionare

3.7 Cerințele detaliate ale aplicației informatice

Aplicația trebuie să implementeze module pentru gestiunea stocurilor, modificarea acestora, formulare pentru ieșirea mărfurilor, afișarea stocurilor pentru a ști dacă se poate îndeplini necesarul minim și diverse rapoarte pe baza datelor existente.

Autentificarea trebuie să se facă astfel încât utilizatorul să poată avea diferite drepturi de utilizare a funcționalităților din aplicație. Parolele vor fi criptate înainte de a fi stocate în baza de

date. Va exista posibilitate de a afișa în momentul autentificării versiunea bazei de date curentă, aceasta modificându-se de fiecare dată când este instalată o actualizare de pe server. Administratorul și managerul vor avea dreptul de a modifica produse și de a gestiona conturile utilizatorilor. In acest fel administratorul poate adăuga, șterge sau modifica datele unui utilizator și de asemenea poate realiza o căutare după adresa de email sau numele acestuia. Pot fi verificate oricând date despre depozit astfel încât administratorul poate lua decizii din timp pentru a evita eventuale decizii care pot conduce organizația către o catastrofă.

Datele despre produsele achiziționate de la furnizori vor putea fi importate printr-un modul de import a datelor din excel direct în baza de date pentru a evita timpii mari pentru introducerea datelor. Fișierele de tip csv sunt puse la dispoziție de fiecare furnizor în vederea facilitării operațiilor de distribuție și achiziție a mărfii.

Filtrarea produselor în funcție de disponibilul stocului oferă posibilitatea utilizatorului de a alege produsele după anumite criterii. De asemenea acesta poate realiza o căutare după nume, denumire sau preț. Există o secțiune de unde utilizatorul poate raporta eventualele probleme către firma care se ocupă de suportul aplicației.

Firma curentă poate fi adăugată, modificată sau ștearsă din baza de date. În momentul adăugării vor exista validări pentru datele de tip întreg, codul numeric personal care trebuie să aibă 13 caractere și numele care trebuie să fie de tip caracter.

Salvarea datelor din formulare se face prin intermediul unor proceduri din baza de date care permit simplificarea prelucrării datelor. Se va implementa un buton pentru afișarea cursului valutar util atunci când se lucrează cu alte monede.

Va exista o secțiune în meniu unde utilizatorul va găsi un manual cu documentația aplicației și datele de contact ale firmei care a dezvoltat aplicația. Vor putea fi emise documente prin intermediul unui formular care conține datele furnizorului, date despre firmă și câmpuri unde pot fi introduse date despre factură precum sunt seria, numărul, valoarea reducerii produselor, data scadentă. Vor putea fi introduse și datele privind expediția precum sunt data de expediere, numele delegatului, mijlocul de transport dar și seria și numărul BI/CI.

În acest formular produsele vor fi adăugate prin intermediul unui buton și vor fi alese dintr-o listă unde sunt încărcate toate produsele. Acestea vor conține date precum sunt denumirea, valoarea tva, unitate de măsură, cantitatea, prețul unitar și valoarea.

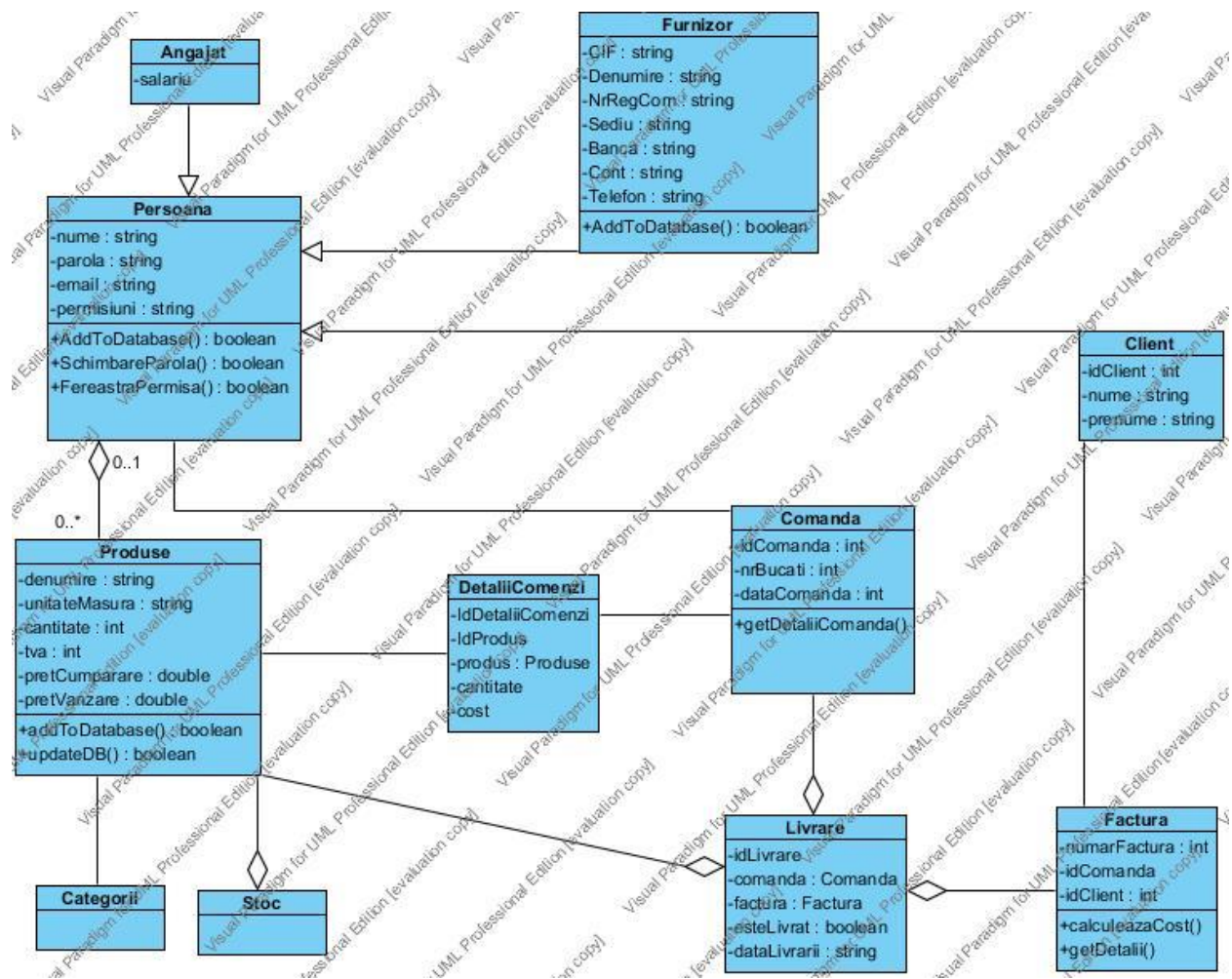
Vizualizarea stocului va permite listarea acestuia și posibilitatea de a le modifica prin intermediul unor corecții. Tot în acest formular există stocată o listă cu toate corecțiile făcute și o secțiune care permite realizarea unei căutări după denumire sau cod.

Capitolul 4. Proiectarea aplicației informatice

4.1 Rafinarea diagramelor

Diagrama de clase rafinată

Diagrama de clase este detaliată și prezintă clasele aplicației cu atributele și metodele aferente acestora. Aici este prezentat în detaliu modul în care clasele se comportă iar diagrama permite o mai bună înțelegere a tuturor funcționalităților pe care aplicația le implementează.

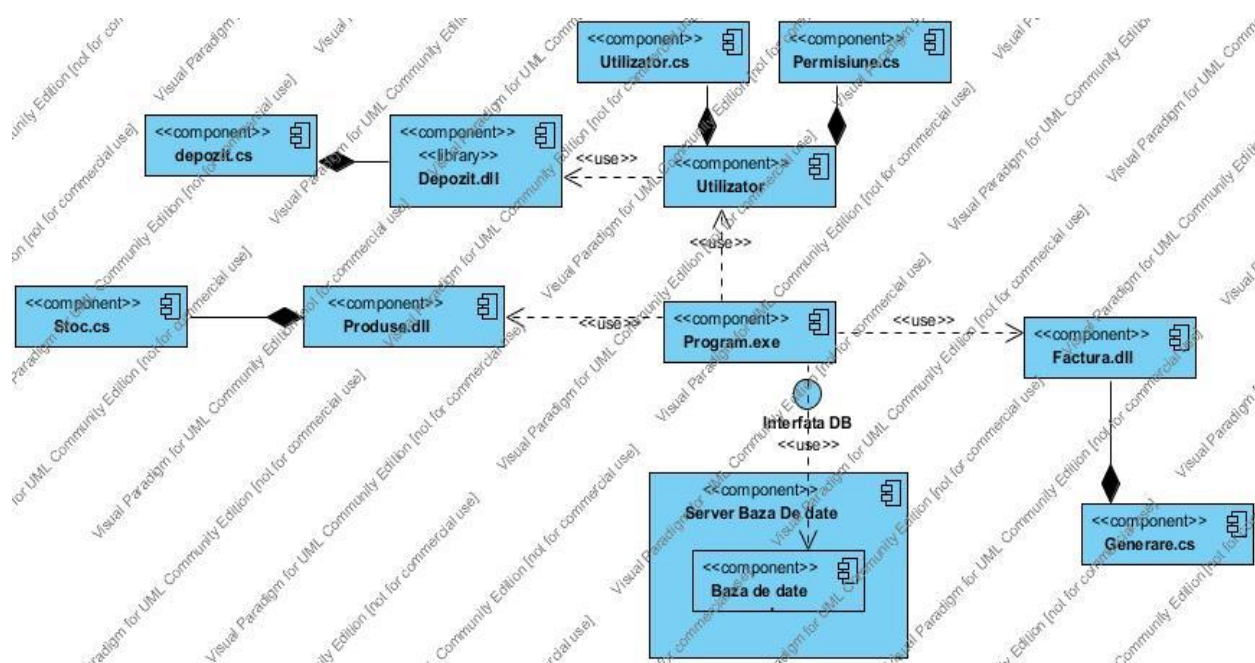


Figură 12 - Diagrama de clase rafinată

4.2 Arhitectura sistemului

Diagrama de componente

“O diagramă de componente prezintă dependențele existente între diverse componente software ce compun un sistem informatic. O componentă este un modul soft(cod sursă, cod binar, dll, executabil etc) cu o interfață bine definită.⁵”

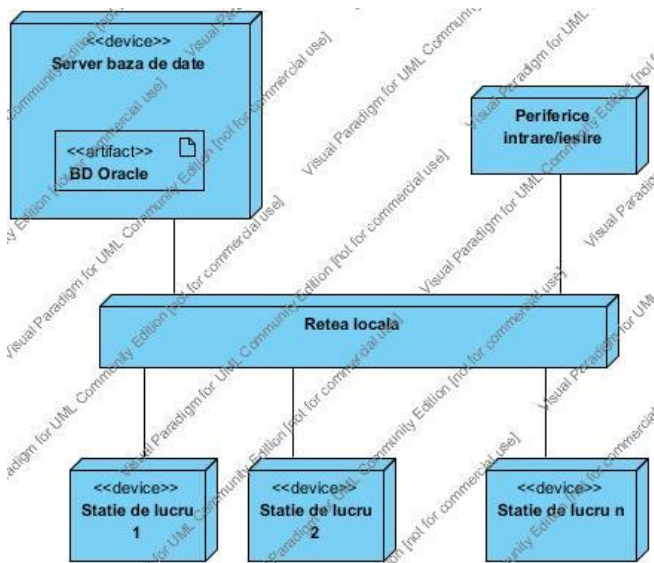


Figură 15 – Diagrama de componente

Diagrama de desfășurare

Diagrama de desfășurare prezintă structura sistemului pe care acesta o are la momentul execuției. Aceasta are în componența sa noduri de tip medii de execuție și dispozitive, fiind componente hardware fără putere de calcul sau componente capabile să execute programe.

⁵ Prof.univ.dr. Ion Lungu, Conf.univ.dr. Anca Andreescu, Conf.univ.dr. Ramona Bologa, Lect.univ.dr. Alexandra Florea – **Proiectarea sistemelor informatice**

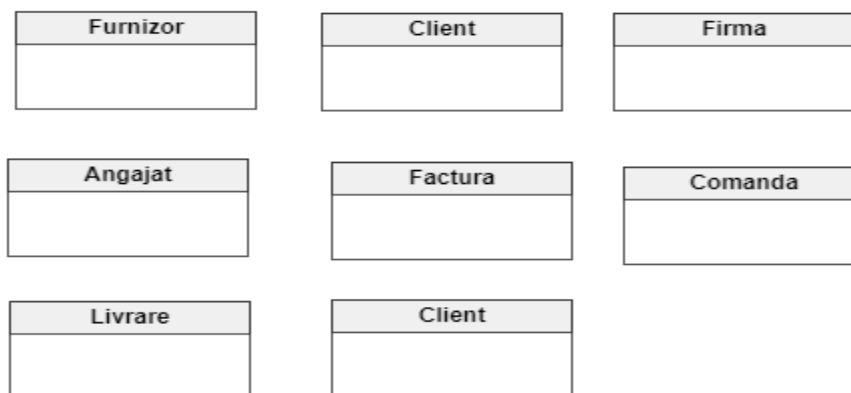


Figură 16 – Diagrama de desfașurare

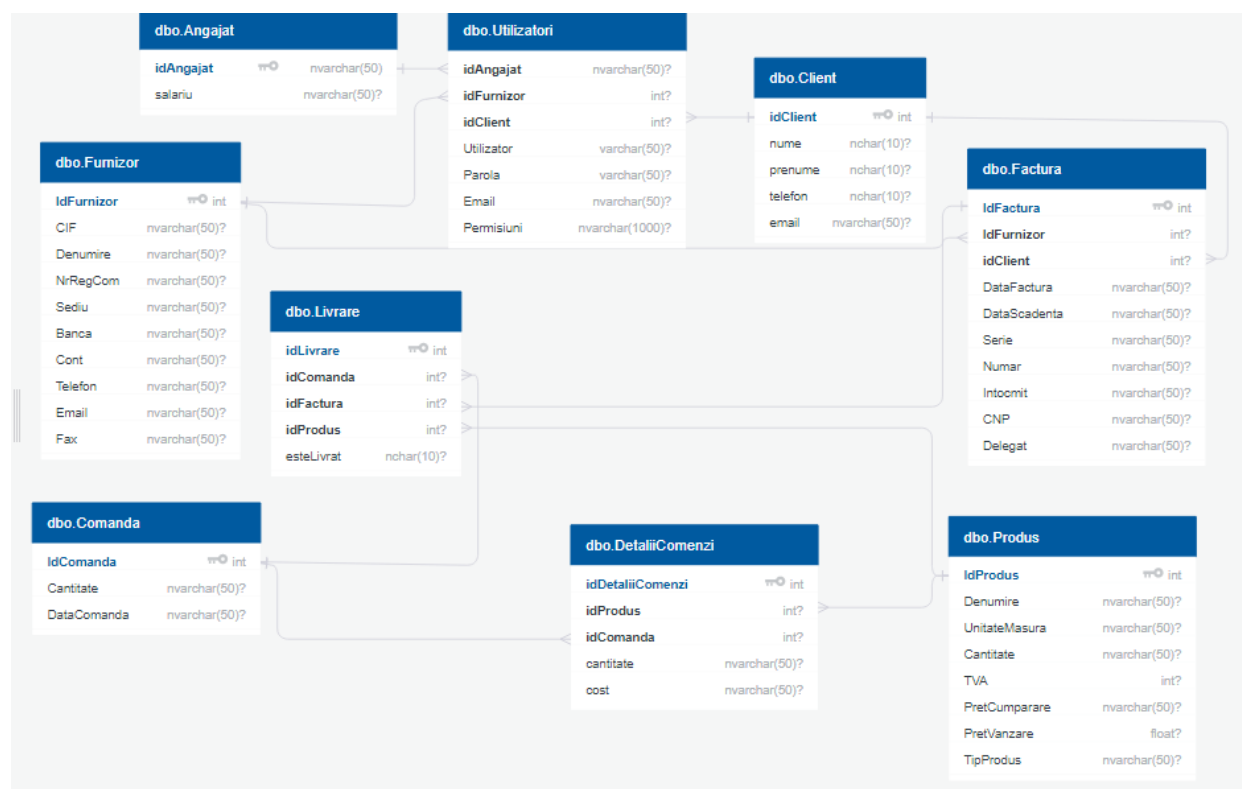
4.3 Proiectarea bazei de date

În proiectare se modelează cerințele sistemului și informațiile care urmează sa fie utilizate de către aplicație. Aici se va ține cont de identificarea entităților, determinarea atributelor, identificarea relațiilor, schema logică etc.

În figura 17 sunt prezentate entitățile identificate după realizarea etapei de analiză. Acestea sunt Angajat, Furnizor, Produs, Utilizator, Factura, Comanda, Client, DetaliiComenzi.



Figură 17 – Identificarea entităților modelului de date



Figură 18 – Schema logică a bazei de date

Aplicația rulează în prezent cu 9 tabele legate între ele prin anumite relații iar acestea sunt: Angajat, Utilizator, Client, Furnizor, Factura, Livrare, DetaliiComenzi, Comanda, Produs.

Tabela UTILIZATORI a fost definită pentru a gestiona datele utilizatorilor și este o tabelă independentă deoarece nu conține restricții de tip cheie externă. Cheia tablei este câmpul idUtilizator și identifică în mod unic un utilizator. Câmpurile nume utilizator, parolă și telefon sunt obligatorii.

Tabela PRODUS gestionează detaliile despre produsele aflate pe stoc. Aceasta conține un id care este de tip cheie primară iar denumirea, categoria și prețul sunt câmpuri care trebuie completate obligatoriu.

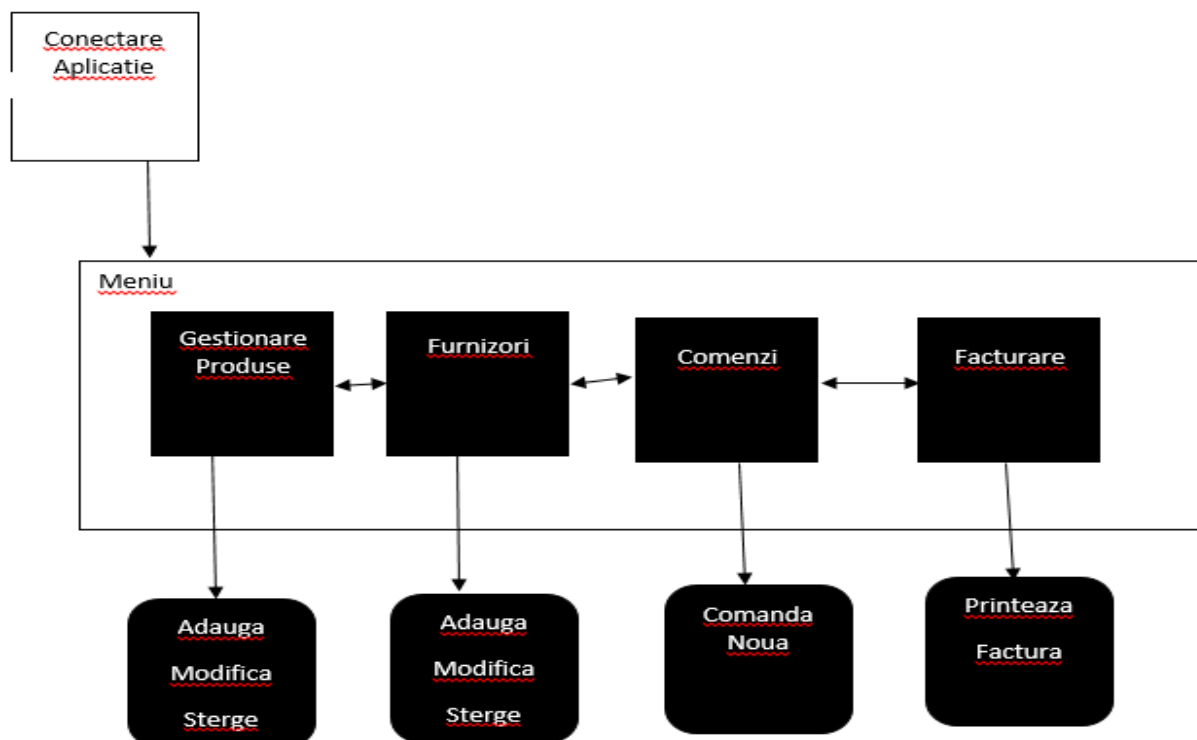
Tabela FURNIZORI conține detalii despre furnizorii care aprovizionează depozitul. Este o tabelă independentă care nu conține câmpuri de tip cheie externă.

Tabela CLIENT conține date despre clienții depozitului și este o tabelă independentă pentru că nu conține restricții de tip cheie externă. Id client este un atribut de tip cheie primară, numărul de telefon și numele fiind câmpuri obligatorii.

Tabela FACTURA conține detalii legate de facturile emise și tranzacțiile din depozit. De asemenea conține restricții de tip cheie externă din tabele precum sunt tabela client și furnizor.

Tabela COMANDĂ gestionează datele care corespund comenzilor din depozitul de articole sportive. Această tabelă conține restricții de tip cheie externă din tabela PRODUSE iar produsele, prețul și cantitatea sunt obligatorii.

4.4 Proiectarea interfetelor utilizator



Figură 19 – Interfețele utilizator

CAPITOLUL 5. Implementarea aplicației informatice

5.1 Crearea bazei de date

Baza de date a fost creată, builduită și actualizată prin intermediul visual studio și table designer. Datele au fost introduse prin modulele implementate în aplicație și modificate ulterior în funcție de operațiile de gestiune.

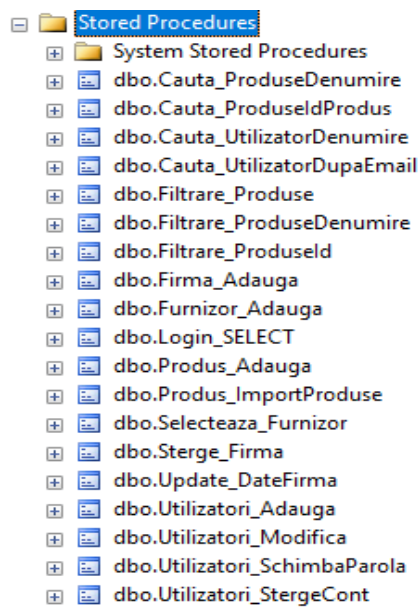
```
GO
CREATE TABLE [dbo].[Produs] (
    [IdProdus] [int] IDENTITY(1,1) NOT NULL,
    [Denumire] [nvarchar](50) NULL,
    [UnitateMasura] [nvarchar](50) NULL,
    [Cantitate] [nvarchar](50) NULL,
    [TVA] [int] NULL,
    [PretCumparare] [nvarchar](50) NULL,
    [PretVanzare] [float] NULL,
    [TipProdus] [nvarchar](50) NULL,
    PRIMARY KEY CLUSTERED
(
    [IdProdus] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object: Table [dbo].[Utilizatori]    Script Date: 6/19/2018 2:45:18 PM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Utilizatori] (
    [idAngajat] [nvarchar](50) NULL,
    [idFurnizor] [int] NULL,
    [idClient] [int] NULL,
    [Utilizator] [varchar](50) NULL,
    [Parola] [varchar](50) NULL,
    [Email] [nvarchar](50) NULL,
    [Permisuni] [nvarchar](1000) NULL
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
ALTER TABLE [dbo].[DetaliiComenzi] WITH CHECK ADD CONSTRAINT [FK_DetaliiComenzi_Comanda] FOREIGN KEY([idComanda])
REFERENCES [dbo].[Comanda] ([IdComanda])
GO
ALTER TABLE [dbo].[DetaliiComenzi] CHECK CONSTRAINT [FK_DetaliiComenzi_Comanda]
GO
ALTER TABLE [dbo].[DetaliiComenzi] WITH CHECK ADD CONSTRAINT [FK_DetaliiComenzi_Produs] FOREIGN KEY([idProdus])
REFERENCES [dbo].[Produs] ([IdProdus])
```

Figură 20 – Script creare bază de date

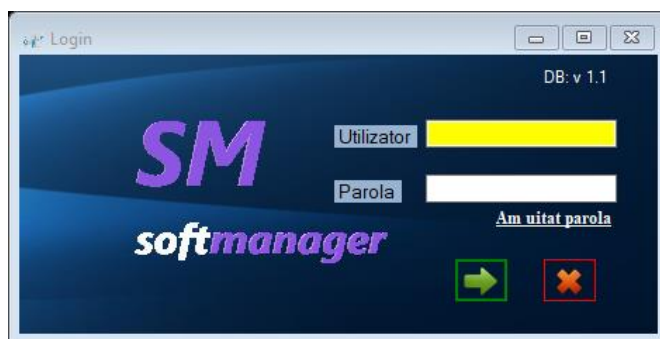
Toate acțiunile pentru manipulare a datelor din baza de date sunt realizate prin intermediul unor proceduri din baza de date. Acestea preiau parametrii trimiși din cod, care reprezintă niște variabile, în funcție de care sunt retrase datele din bază.



Figură 21 – Proceduri

5.2 Prezentarea funcționalității sistemului

La rularea aplicației SoftManager va fi deschisă o activitate unde utilizatorul va putea să își introducă datele de autentificare. În cazul în care un utilizator și-a uitat una din datele de autentificare, acesta are posibilitate de a schimba numele de utilizator sau parola. De asemenea ori de câte ori aplicația este actualizată, varianta bazei de date va fi



Figură 22 Fereastră Autentificare

și ea actualizată și afișată în partea din dreapta sus a ferestrei de autentificare.

În momentul în care au fost furnizate datele corespunzătoare, utilizatorul poate accesa fereastra principală printr-un simplu click pe butonul care conține o imagine cu o săgeată sugestivă, sau prin apăsarea tastei enter. În spatele acestei operații se află o procedură aflată în baza de date și apelată din cod.

```

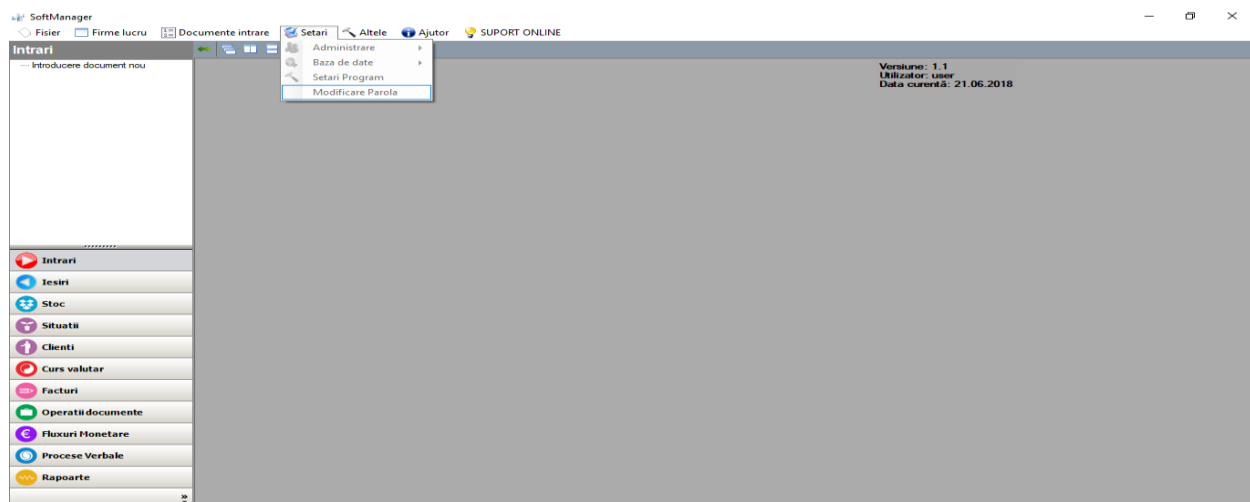
try
{
    con.Open();
    SqlCommand com = new SqlCommand();
    com.Connection = con;
    com.CommandText = "Login_SELECT";
    SqlDataReader r = com.ExecuteReader();
    int count = 0;
    string Utilizator = "";
    if (r.HasRows)
    {
        while (r.Read())
        {
            Utilizator = r["Utilizator"].ToString();
            string Parola = r["Parola"].ToString();

            if (tb_Utilizator.Text.ToLower() == Utilizator && tb_Password.Text.ToLower() == Security.Decrypt(Parola))
            {
                Settings.permisiuni = r["Permiuni"].ToString();
                Settings.utilizator_Logat = Utilizator;
                Settings versiuni = versiune;
            }
        }
    }
}

```

Figura 23 Codul pentru interogarea tabeli LOGIN

Imediat după ce autentificarea a fost realizată cu succes se va deschide o fereastră principală (Figura 25) care este de tip MDI (o interfață care conține mai multe documente) și în care sunt afișate detalii despre utilizator, versiunea bazei de date și data curentă. Aici utilizatorul are acces la o parte din opțiunile implementate din meniu, în funcție de drepturile acordate contului său cum ar fi raportarea unei probleme către suportul online al aplicației, listarea produselor din baza de date, gestionarea datelor de utilizator proprii. Poate obține rapoarte pe baza datelor înregistrate în baza de date, emite o factură către un client care va trebui revizuită de către manager și poate vedea toate produsele existente în stoc.



Figură 24 Fereastră Meniu Principal

Prima secțiune din meniu permite iesirea din aplicație și delogarea utilizatorului. În cazul în care este aleasă opțiunea pentru delogare un alt utilizator va avea posibilitatea de a se loga în aplicație.

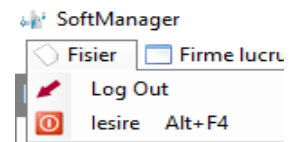


Figura 25 Log Out

Meniul “Modificare Date Utilizator” permite accesul printr-o activitate la formularul unde orice utilizator își poate modifica informațiile contului și datele de acces. În partea de implementare, din cod se va apela o procedură din baza de date unde comanda update va realiza o schimbare direct în baza datelor utilizatorului.

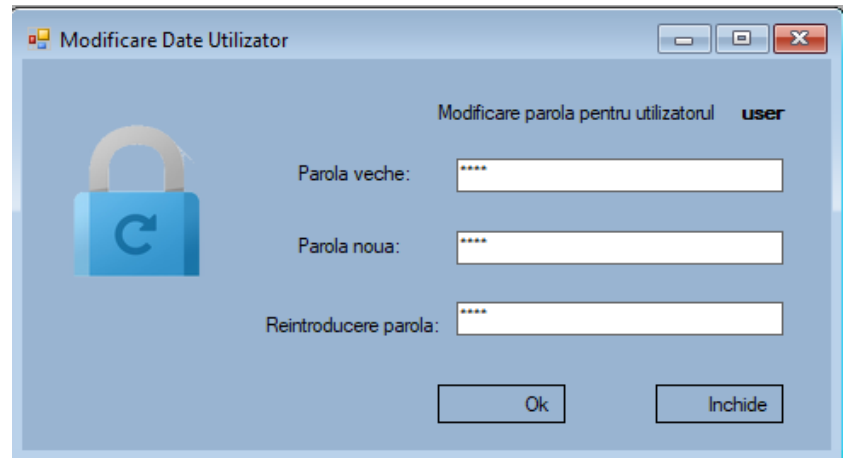


Figura 26 Modificare Date Utilizator

Administratorul are posibilitatea de a crea un cont, de a șterge un cont, de a modifica datele acestuia și de asemenea poate căuta un utilizator mult mai ușor prin intermediul căutare cont.

Pentru adăugarea unui cont nou, administratorul trebuie să adauge în formular numele utilizatorului, parola care va fi schimbată la fiecare 3 luni și adresa de email a utilizatorului.

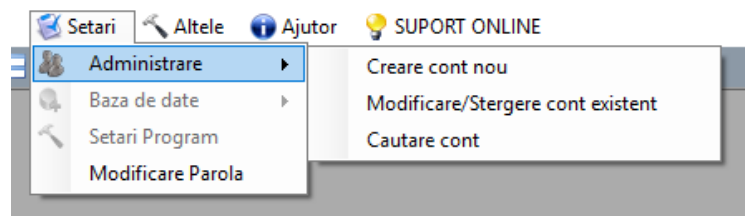


Figura 27 Adăugare cont nou

Pentru a efectua o căutare a unui utilizator în baza de date, administratorul trebuie să aleagă un filtru de căutare după nume sau adresa de mail. Astfel vor fi afișate detalii despre contul angajatului cum sunt permisiunile, parola care este criptată și numele utilizatorului.

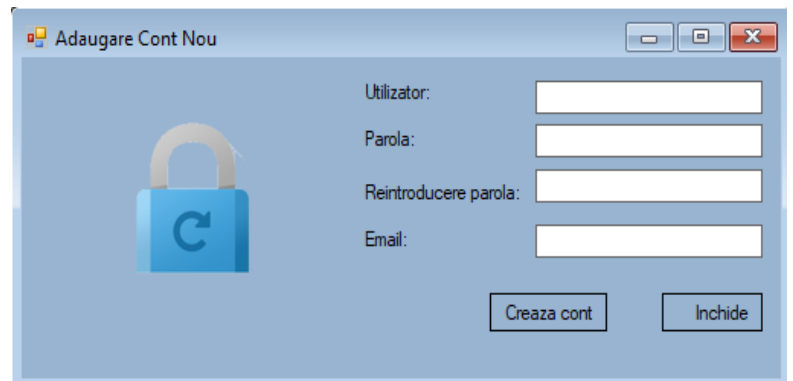


Figura 28 Adăugare cont nou

Cautare Cont

Cauta dupa
☒ Utilizator
☐ Email

Nume utilizator:

	idAngajat	idFurnizor	idClient	Utilizator	Parola	Email	Permisuni
▶				sa	c2E=	office@gestsoft.ro	mnuListaFirme,mnuAdaugaFirma,mnuAjutor,fmEmitA...

Figura 29 Căutare cont

Formularul pentru adăugarea firmei care utilizează aplicația permite apelul unei proceduri care adaugă date relevante precum sunt codul de identificare fiscal, sediul, banca și datele de contact. Va exista posibilitatea de a fi adăugată persoana care a întocmit firma dar și CNP-ul acesteia. Pentru fiecare câmp există verificări realizate cu ajutorul unui `errorProvider` care va înștiința utilizatorul în cazul în care va introduce caractere în câmp destinat introducerii numărului de telefon sau mai mult de 13 caractere în secțiunea CNP.

Adaugă Firmă

Date firma

CIF:

Denumire:

Nr.Reg.Com.:

Sediul:

Banca:

Cont:

Telefon:

E-mail:

Fax:

☐ Platitor TVA

Intocmitor

Nume:

CNP:

Figura 30 Adăugare firmă

Introducere document nou permite adăugarea unui document de către orice utilizator atunci când se creează o comandă. Aici sunt selectate datele furnizorului de la care se face aprovizionarea, datele firmei în numele căreia se face comanda, detalii despre factură și despre produsele comandate.

Date furnizor
Denumire:
[Adauga furnizor](#) [Modifica furnizor](#)

CIF: 25514899
Sediu: PLOIESTI STR. GAGENI
Banca: BRD
Cont: RO77BRDE450SV6568304
Telefon: 0726274621
E-mail: office@lore.com

Document intrare
Tip document:
Serie:
Numar:
Data:
Discount: %
Scadenta:
Mod plata:

Date firma
Denumire: UNIX AUTO SRL (Fil)
CIF: 10542416
Sediu: CLUJ NAPOCA STR.FABR
Banca: BRD
Cont: RO77BRDE450SV6568304
Telefon: 0722534852
Fax: office@unix.com
E-mail: 0316200595
[Firma...](#)

Produse/Servicii
Denumire produs: [Nomenclator produse/servicii](#) TVA(%): U.Masura: Cantitate: Pret unitar(RO) fara TVA: [Mondea facturi: RON](#) [Adauga](#)

NrCrt	DenumireProdus	TVA(%)	U/M	Cantitate	Pret unitar	Valoare	T.V.A
117	PANTALONI DAMA M...	24	Buc	2	138	276	66.24
124	TRICOU MACRON TO...	24	Buc	20	85	1700	408
126	TRICOU DAMA MACR...	24	Buc	2	79	158	37.92
125	SORT GIOVA FRIEND	24	Buc	15	59	885	212.4
TOTAL 3019							

Date privind expeditia
Data expediere: Nume delegat:
[Inregistreaza document](#) [Anuleaza modificari](#)

Figura 31 Document intrare

Există posibilitatea adăugării, ștergerii și modificării unei firme în baza de date și ale cărei detalii vor apărea în formularul de introducere date. Au fost implementate validări pentru fiecare câmp pentru a evita anumite situații cum ar fi introducerea unui produs fără cantitate. Astfel dacă o cantitate este mai mică decât 1 va afișa o eroare prin care utilizatorul va fi informat și îndrumat să introducă datele corect.

Exista o verificare în momentul adăugării unui produs în listă pentru a genera o factură care verifică dacă a fost introdusă cantitatea și aceasta este mai mare strict ca 0. Nomenclator produse va permite verificarea produselor existente în baza de date și va permite modificarea detaliilor acestora dar și adăugarea unui nou produs în cazul în care nu există.

După ce toate câmpurile au fost completate se va înregistra documentul și se va emite factura. Produsele sunt încărcate din baza de date într-un dataGridView și în funcție de cantitate și preț se va calcula fiecare valoare a comenzii unui produs și mai departe totalul tuturor valorilor.

În momentul în care un document a fost generat, datele vor fi stocate în baza de date iar pe ecran va fi afișată factura cu toate detaliile selectate până la momentul respectiv. Această

Factura seria AAAA nr. 2 din 25/06/2018

Furnizor:

MAGNICOM SRL

Cod Identificare Fiscală: RO 5427038

Nr. Registrul Comerțului: 340/5494/1994

Sediu: BUCUREȘTI

Județ, Țară: , Romania

Banca: BRD

IBAN: RO77BRDE450SV65683044500

Cumpărător:

SoftManager

Cod Identificare Fiscală: 123456789

Nr. Registrul Comerțului: 123695

Sediu: Titan

Banca: BRD

IBAN: RO34 6524 7127 4547

+40722 534 789

office@fax

office@softmanager.ro

Pagina: 1 / 1

Nr.crt.	Denumire produs/serviciu/lucrare	TVA (%)	U.M.	Cantitate	Pret unitar (RON)	Valoare (RON)	T.V.A. (RON)
1	ADIDAS Incaltaminte ELEMENT RACENEW	0	BUC	20,000	78,00	1.560,00	0,00
2	GEACA IMPERMEABILA	0	BUC	32,000	24,00	768,00	0,00
3	NIKE Geaca RS WIND 3KT WNEW	0	BUC	50,000	125,00	6.250,00	0,00
Total:						8.578,00	0,00
Discount: 10.00%						857,80	0,00

Factura conține detalii despre Furnizor, Cumpărător adică propria firmă în cazul nostru dar și detalii despre produsele selectate în vederea cumpărării lor. Sunt afișate date precum sunt denumirea produsului, valoarea TVA-ului, unitatea de măsură și cantitatea introdusă, prețul unitar și valoarea produsului în funcție de cantitate dar și valoarea totală a facturii.

— □ ×

Listă Firme

	IdFirma	Denumire	CIF	NrRegCom	Sediu	Banca	Cont
▶	102	UNIX AUTO SRL (Fili	10542416	J29/806/2009	CLUJ NAPOCA STR.FABR	BRD	RO77BRDE450SV6568
	108	GLOBESPED INTERNATI	RO 24136434	J08/1828/2008	STR. FANULUI 51 Cod	BCR	RO77BRDE450SV6865
	109	GLOBAL TEAM SRL	23939206	J12/2314/2008	CLUJ NAPOCA	ING	RO77BRDE450SV6568

< >

Figura 33 Lista cu firme

Aplicația conține și o listă cu firme deja predefinite unde utilizatorul poate selecta o firmă pentru a seta datele acesteia în aplicație și poate edita datele acestora. Există un buton pentru a reactualiza datele din bază și un buton pentru a șterge o firmă în cazul în care firma nu mai există sau din anumite motive nu se mai dorește păstrarea acestor date.

O altă funcționalitate este reprezentată de Nomenclator produse care permite administratorului să modifice orice produs din bază sau să adauge un produs nou. Tot aici există posibilitatea de a face o căutare după codul sau denumirea unui produs și de a salva produsul respectiv.

Nomenclator Produse

Cautare produs

☐ Denumire:

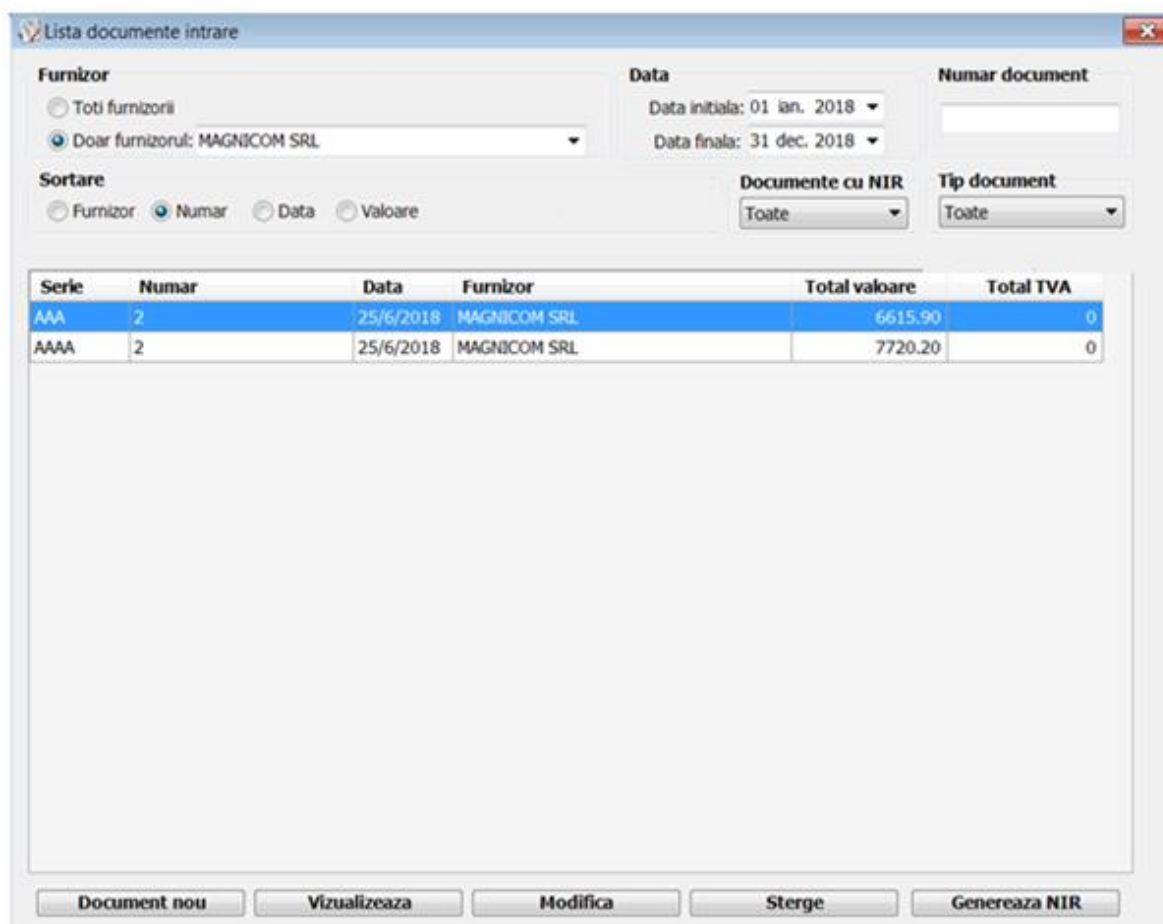
☒ Cod: 128

	IdProdus	IdFactura	Denumire	UnitateMasura	Cantitate	TVA	PretCumparare	PretVanzare
▶	128		GEANTA MACR...	Buc	20	19	156	178
*								

Salveaza

Figura 34 Nomenclator produse

În meniu la secțiunea Intrari există o listă cu toate documentele de intrare care afișează toate facturile, avizele și bonurile de transfer din gestiune. Există posibilitatea de a selecta doar un anumit furnizor sau toți furnizorii și se poate face o sortare după număr, data, valoare sau furnizor. Se poate pune un filtru pe data initiala și finală iar documentele pot fi filtrate în funcție de existența NIR. Datele afișate conțin detalii despre totalul TVA, furnizor, valoare totală, dată, număr.



Lista documente intrare

Furnizor
☐ Toti furnizorii
☒ Doar furnizorul: MAGNCOM SRL

Data
 Data initiala: 01 ian. 2018
 Data finala: 31 dec. 2018

Numar document

Sortare
☐ Furnizor ☒ Numar ☐ Data ☐ Valoare

Documente cu NIR

Tip document

Serie	Numar	Data	Furnizor	Total valoare	Total TVA
AAA	2	25/6/2018	MAGNCOM SRL	6615.90	0
AAAA	2	25/6/2018	MAGNCOM SRL	7720.20	0

Document nou Vizualizeaza Modifica Sterge Genereaza NIR

Figura 35 Lista documente intrare

La secțiunea ieșiri se poate face transferul între gestiuni prin selectarea gestiunii sursă și destinație. În momentul în care este selectată gestiunea sursă, vor apărea toate produsele care corespund acestei gestiuni și date despre produse precum sunt unitatea de măsură, cantitatea care se vrea a fi transferată în gestiunea destinație, prețul unitar și cantitatea disponibilă. După ce au fost adăugate toate produsele care urmează să fie transferate într-o listă, este calculat totalul și este emis bonul de transfer (figura 35). Tot aici este implementat și un buton pentru anularea modificărilor în cazul în care s-a făcut o greșeală sau nu se mai dorește transferul respectiv.

Transfer gestiuni

Gestiuni

Gestiune sursă: Berceni

Gestiune destinație: Pipera

[Adauga/modifica gestiuni](#)

Date generale

Numar: 1

Data: /06/2018

Produse stoc

Denumire produs: [Nomencator produse](#)

ADIDAS Incaltaminte ELEMENT RACENEW

U/M: BUC Cantitate: 20 Pret unitar (RON): 78

[Adauga](#)

ADIDAS Incaltaminte ELEMENT RACENEW Cantitate disponibila: BUC

NrCrt	Cod produs	Denumire produs	U/M	Cantitate	Pret unitar	Valoare
1	2	ADIDAS Incaltaminte ELEMENT RACENEW	BUC	20	78	1560

Total valoare transferat: 1560

[Inregistreaza bon transf](#) [Anuleaza modificari](#)

Figura 36 Transfer gestiuni

Pentru a evita situațiile în care comenzile efectuate nu pot fi onorate din cauza insuficienței stocurilor există un modul de afișare a detaliilor despre cantitățile aflate în depozit. Sunt afișate din baza de date toate produsele împreună cu detaliile aferente acestora, tot aici fiind posibilă corecția lor. Într-o fereastră diferită sunt afișate toate corecțiile făcute.

Unitate: SoftManager

Cod Identificare fiscala: 123456789

Nr. Registrul comertului: 123695

Adresa: Titan

Gestiune: Berceni

Bon de transfer nr. 1 din 28/06/2018

Catre gestiunea:Pipera

Nr. Crt.	Denumire produs	U.M.	Cantitate	Pret unitar	Valoare
1	GEACA IMPERMEABILA	BUC	20,00	24,00	480,00
TOTAL					480,00

Figura 37 Bon de transfer

Vizualizare Stoc

Cautare produs

☐ Doar produsul ☒ Denumire ☐ Cod

Vizualizare stoc

☒ Cantitativ ☐ Cantitativ-valoric

Gestiune

	PretUnitar	IdProdus	Denumire	UnitateMasura	Cantitate	TVA	PretCumparare	PretVanzare	TipProdus
▶	128	116	PANTALONI DA...	Buc	2	24	128	138	Marfa
	128	117	PANTALONI DA...	Buc	2	24	128	138	Marfa
	156	118	GEANTA MACR...	Buc	20	19	156	178	Marfa
	84	119	GEACA IARNA M...	Buc	17	24	84	98	Marfa
	185	120	SET ECHIPAME...	Buc	23	24	185	191	Marfa
	152	121	HANORAC DAM...	Buc	34	24	152	169	Marfa
	75	122	TRICOU POLO ...	Buc	56	24	75	89	Marfa
	99	123	TRICOU MACRO...	Buc	78	24	99	109	Marfa
	77	124	TRICOU MACRO...	Buc	56	24	77	85	Marfa
	55	125	SORT GIOVA F...	Buc	12	24	55	59	Marfa
	65	126	TRICOU DAMA ...	Buc	89	24	65	79	Marfa
*									

Corectie Lista corectii

Total valoare stoc: 1204

Figura 38 Vizualizare stoc

Evoluțiile cantitative și valorice ale produselor din stoc sunt evidențiate prin intermediul unor grafice unde sunt afișate detalii despre firmă cum ar fi adresa, CUI, denumirea firmei și numele gestiunii în cazul în care avem mai multe gestiuni.

Unitate: SoftManager

Cod Identificare fiscala: 123456789

Nr. Registrul comertului: 123695

Adresa: Titan

Gestiune: 1

**Istoric stoc pentru produsul GEACA IMPERMEABILA
pe perioada 01/06/2018 - 29/06/2018**

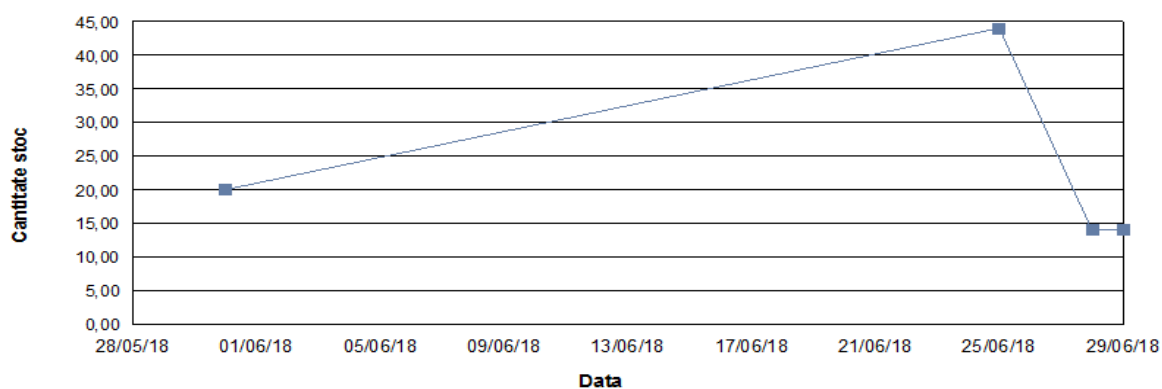


Figura 39 Istoric Stoc

Concluzie

În concluzie, această aplicație desktop dezvoltată cu ajutorul tehnologiei Windows Forms, este destinată clienților a căror afacere este la început și în plină dezvoltare iar modul în care datele sunt prelucrate este dinamic și într-o continuă schimbare.

În primul rând aplicația a fost proiectată pentru gestiunea unui depozit de dimensiuni reduse unde numărul de date este relativ mic și implicit tabelele și procedurile nu implică un grad foarte mare de dificultate. Pe viitor intenționez să dezvolt programul sau chiar să-l reprojectez, și să-l îmbunătățesc astfel încât să fie adaptat cât mai ușor oricărui depozit.

În al doilea rând atât aplicația cât și baza de date vor primi îmbunătățiri și modificări substanțiale pentru a face față unui flux de date mult mai mare și unor proceduri mai avansate care să satisfacă cererile de pe piață și cererile clienților, acestea fiind într-o continuă schimbare.

Aplicația a fost gândită astfel încât utilitatea acesteia să fie maximă, costurile de implementare și instalare să fie minime iar modul în care este utilizată să fie cât mai simplu și ușor fără să implice o instruire îndelungată și personal specializat pentru a utiliza aplicația. Deoarece costul achiziției unui PC este mic și orice firmă și-l poate permite, iar sistemul Windows este foarte popular și accesibil tehnologiile au fost special alese pentru a facilita tot acest proces de dezvoltare, implementare și instalare a sistemului.

Tot datorită tehnologiei aplicația este rapidă iar totul este făcut în timp util fără să testeze prea mult răbdarea clientului. Intenționez să lucrez în continuare la securizarea acesteia pentru a fi în concordanță cu normele cerute de ASF și a îndeplini cerințele impuse de GDPR.

Sistemul este încă în faza de dezvoltare și intenționez să dezvolt noi module pentru a acoperi cât mai multe nevoi ale clienților și a concura cu softuri deja existente pe piață dar care nu reușesc să facă față cererii din partea clienților.

Consider că această aplicație poate fi folosită de foarte multe întreprinderi pentru gestiunea produselor din depozitele proprii și cu dezvoltări ulterioare prin optimizarea fluxurilor și informațiilor poate reprezenta o soluție viabilă pentru majoritatea depozitelor în vederea gestionării inteligente a stocurilor și maximizării profitului.

Bibliografie

1. Albahari Joseph and Ben Albahari - *C# 4.0 IN A NUTSHELL Fourth Edition*, Fourth Edition, 2010, pp 1,2
2. Chirita CARAIANI(coord), Conf.univ.dr.Lavinia OLIMID(coord) - *Bazele contabilitatii*
3. Lungu Ion (coord), Adela Bara, Constanta Bodea, Iuliana Botha, Vlad Diaconita, Alexandra Florea, Anda Velicanu – *Tratat de Baze de Date*, vol. I,Editura ASE, Bucuresti, p.140
4. Lungu Ion prof.univ.dr., Conf.univ.dr. Anca Andreescu, Conf.univ.dr. Ramona Bologa, Lect.univ.dr. Alexandra Florea – **Proiectarea sistemelor informatice**
5. Smeureanu Ion, Dârdala Marian, Reveiu Adriana – *Visual C#.NET*, Editura CISON, 2004
6. Smeureanu Ion, Dârdală Marian. (2002) *Programarea orientată obiect în limbajul C++*, București Editura: CISON
7. http://bd.ase.ro/uploads/sgbd_curs/materialStudiuSGBDoracle.pdf
8. <https://mitseu.files.wordpress.com/2014/08/microsoft-sql-server-2012-step-by-step-pre-press.pdf>

ANEXE

Lista Figuri:

Selectarea unui produs din baza de date pentru crearea unei facturi

```
string UnitateMasura, TipProdus = "";
int IdProdus, TVA, Cantitate = 0;
double PretVanzare = 0.0;

if (Convert.ToInt32(tbCantitate.Text) > 0)
{
    Cantitate = Convert.ToInt32(tbCantitate.Text);
    string Denumire = cmbDenumireProdus.SelectedItem.ToString();
    int index = (int)cmbDenumireProdus.SelectedIndex;

    SqlConnection con = new SqlConnection(Settings.Provider);
    con.Open();
    SqlCommand com = new SqlCommand("Select * from Produs", con);
    DataSet ds = new DataSet();
    SqlDataAdapter da = new SqlDataAdapter(com);

    try
    {
        da.Fill(ds);
        IdProdus = (int)ds.Tables[0].Rows[index]["IdProdus"];
        Denumire = ds.Tables[0].Rows[index]["Denumire"].ToString();
        TVA = (int)ds.Tables[0].Rows[index]["TVA"];
        tbTVA.Text = (Convert.ToString(TVA));
        UnitateMasura = ds.Tables[0].Rows[index]["UnitateMasura"].ToString();
        PretVanzare = (double)ds.Tables[0].Rows[index]["PretVanzare"];

        double Valoare = PretVanzare * Cantitate;
        TipProdus = ds.Tables[0].Rows[index]["TipProdus"].ToString();
        double ValoareTVA = Valoare * TVA * 0.01;
    }
}
```

Modificarea datelor pentru un furnizor în baza de date

```

private void btnModificaFirma_Click(object sender, EventArgs e)
{
    SqlConnection conexiune = new SqlConnection(Settings.Provider);
    conexiune.Open();
    foreach (DataGridViewRow row in dgListaFirma.Rows)
    {
        if (dgListaFirma.Rows.Count > 0)
        {
            int nRowIndex = dgListaFirma.Rows.Count - 2;

            if (dgListaFirma.Rows[nRowIndex].Cells[0].Value != null)
            {
                string query = "Update_DateFirma";
                using (SqlCommand cmd = new SqlCommand(query, conexiune))
                {
                    cmd.CommandType = CommandType.StoredProcedure;

                    cmd.Parameters.AddWithValue("@IdFirma", row.Cells[0].Value);
                    cmd.Parameters.AddWithValue("@Denumire", row.Cells[1].Value);
                    cmd.Parameters.AddWithValue("@CIF", row.Cells[2].Value);
                    cmd.Parameters.AddWithValue("@NrRegCom", row.Cells[3].Value);
                    cmd.Parameters.AddWithValue("@Sediu", row.Cells[4].Value);
                    cmd.Parameters.AddWithValue("@Banca", row.Cells[5].Value);
                    cmd.Parameters.AddWithValue("@Cont", row.Cells[6].Value);
                    cmd.Parameters.AddWithValue("@Telefon", row.Cells[7].Value);
                    cmd.Parameters.AddWithValue("@Email", row.Cells[8].Value);
                    cmd.Parameters.AddWithValue("@Fax", row.Cells[9].Value);
                    cmd.Parameters.AddWithValue("@Nume", row.Cells[10].Value);
                    cmd.Parameters.AddWithValue("@CNP", row.Cells[11].Value);
                }
            }
        }
    }
}

```

Autentificare

```

con.Open();
SqlCommand com = new SqlCommand();
com.Connection = con;
com.CommandText = "Login_SELECT";
SqlDataReader r = com.ExecuteReader();
int count = 0;
string Utilizator = "";
if (r.HasRows)
{
    while (r.Read())
    {
        Utilizator = r["Utilizator"].ToString();
        string Parola = r["Parola"].ToString();

        if (tb_Utilizator.Text.ToLower() == Utilizator && tb_Password.Text.ToLower() == Security.Decrypt(Parola))
        {
            Settings.permisiuni = r["Permiuni"].ToString();
            Settings.utilizator_Logat = Utilizator;
            Settings.versiuni = versiune;
            count = count + 1;
        }
    }
}

```

Funcțiile de criptare si decriptare


```

public static string Encrypt(string Data)
{
    SHA1Managed shaM = new SHA1Managed();
    Convert.ToBase64String(shaM.ComputeHash(Encoding.ASCII.GetBytes(Data)));
    byte[] eNC_data = ASCIIEncoding.ASCII.GetBytes(Data);
    string eNC_str = Convert.ToBase64String(eNC_data);
    return eNC_str;
}

public static string Decrypt(string Data)
{
    try
    {
        byte[] dEC_data = Convert.FromBase64String(Data);
        string dEC_Str = ASCIIEncoding.ASCII.GetString(dEC_data);
        return dEC_Str;
    }
    catch (Exception ex)
    {
        return "";
    }
}

```