



**Universitatea Tehnică a Moldovei**

**GESTIONAREA DATELOR ÎN SFERA LOGISTICII  
CAMIOANELOR**

**DATA MANAGEMENT IN THE FIELD OF TRUCKING  
LOGISTICS**

**Student:**

**gr. TI-194,  
Marcu Vadim**

**Coordonator:**

**Scrob Sergiu  
asistent universitar**

**Chișinău, 2023**

## REZUMAT

### Introducere:

Logistica camionajelor joacă un rol vital în industria transporturilor și a devenit din ce în ce mai importantă în ultimii ani, datorită creșterii comerțului electronic și globalizării. Eficiența și eficacitatea logisticii de transport cu camioane au un impact direct asupra productivității și profitabilității afacerilor, făcându-l un domeniu critic de studiu.

### Analiza domeniului de studiu:

Logistica de transport implică planificarea, organizarea și controlul mișcării mărfurilor și materialelor. Este nevoie de o înțelegere a managementului lanțului de aprovizionare, a controlului stocurilor, a modurilor de transport și a rutelor. Logistica eficientă a transportului poate duce la reducerea costurilor de transport, la creșterea vitezei de livrare, la îmbunătățirea satisfacției clienților și la creșterea veniturilor.

### Modelarea și proiectarea sistemului:

Dezvoltarea unui sistem logistic de transport implică modelarea și proiectarea unui cadru care mută eficient mărfurile dintr-o locație în alta. Acest lucru necesită integrarea tehnologiei, a analizei datelor și a procedurilor operaționale. Sistemul trebuie să fie scalabil, adaptabil și capabil să facă față schimbărilor în cerere, vreme și evenimente neprevăzute.

### Realizarea sistemului:

Odată ce proiectarea sistemului este finalizată, începe faza de implementare. Aceasta implică testarea, pilotarea și reglarea fină a sistemului pentru a se asigura că îndeplinește cerințele afacerii. Sistemul trebuie să fie ușor de utilizat și intuitiv pentru a permite operațiuni fluide și eficiente.

### Documentatia produsului finit:

Trebuie implementat un proces cuprinzător de documentare pentru a se asigura că sistemul poate fi întreținut, actualizat și scalat cu ușurință. Documentația trebuie să includă specificații ale sistemului, manuale de operare și ghiduri de depanare.

### Estimarea costului proiectului:

Dezvoltarea și implementarea unui sistem de logistică pentru camioane necesită resurse și investiții semnificative. Estimarea costurilor ar trebui să ia în considerare factori precum costurile tehnologiei și echipamentelor, costurile cu forța de muncă, costurile de formare și costurile de întreținere continuă

În concluzie, logistica camioanelor joacă un rol critic în industria transporturilor, iar importanța acesteia va continua să crească. Dezvoltarea și implementarea unui sistem eficient de logistică a transportului necesită o înțelegere cuprinzătoare a domeniului de studiu, modelarea și proiectarea unui sistem robust, realizarea eficientă, documentația cuprinzătoare și estimarea precisă a costurilor proiectului.

# **ABSTRACT**

## **Introduction:**

Trucking logistics plays a vital role in the transportation industry, and it has become increasingly important in recent years due to the growth of e-commerce and globalization. The efficiency and effectiveness of trucking logistics directly impact the productivity and profitability of businesses, making it a critical field of study.

## **Analysis of the Field of Study:**

Trucking logistics involves the planning, organizing, and controlling of the movement of goods and materials. It requires an understanding of supply chain management, inventory control, transportation modes, and routes. Efficient trucking logistics can lead to reduced transportation costs, increased delivery speed, improved customer satisfaction, and increased revenue.

## **System Modeling and Design:**

Developing a trucking logistics system involves modeling and designing a framework that efficiently moves goods from one location to another. This requires the integration of technology, data analysis, and operational procedures. The system must be scalable, adaptable, and able to handle changes in demand, weather, and unforeseen events.

## **Realization of the System:**

Once the system design is complete, the implementation phase begins. This involves testing, piloting, and fine-tuning the system to ensure it meets the requirements of the business. The system must be user-friendly and intuitive to allow for smooth and efficient operations.

## **Documentation of the Finished Product:**

A comprehensive documentation process must be put in place to ensure the system can be easily maintained, updated, and scaled. The documentation should include system specifications, operation manuals, and troubleshooting guides.

## **Project Cost Estimation:**

Developing and implementing a trucking logistics system requires significant resources and investment. The cost estimate should consider factors such as technology and equipment costs, labor costs, training costs, and ongoing maintenance costs. A detailed project cost estimate can help businesses make informed decisions about the viability and feasibility of implementing a trucking logistics system.

In conclusion, trucking logistics plays a critical role in the transportation industry, and its importance will continue to grow. Developing and implementing an efficient trucking logistics system requires a comprehensive understanding of the field of study, robust system modeling and design, effective realization, comprehensive documentation, and accurate project cost estimation.

## CUPRINS

INTRODUCERE .....	14
1 ANALIZA DOMENIULUI DE STUDIU .....	15
1.1 IMPORTANȚA TEMEI .....	16
1.2 SISTEME SIMILARE CU PROIECTUL REALIZAT .....	17
1.3 SCOPUL, OBIECTIVELE ȘI CERINȚELE SISTEMULUI .....	21
2 MODELAREA ȘI PROIECTAREA SISTEMUL INFORMATIC .....	23
2.1 DESCRIEREA COMPORTAMENTALĂ A SISTEMULUI .....	24
2.1.1 IMAGINEA GENERALĂ ASUPRA SISTEMULUI .....	25
2.1.2 MODELAREA VIZUALĂ A FLUXURILOR .....	27
2.1.3 STĂRILE DE TRANZACȚIE A SISTEMULUI .....	29
2.1.4 DESCRIEREA SCENARIILOR DE UTILIZARE A APLICAȚIEI .....	31
2.1.5 FLUXURILE DE MESAJE ȘI LEGĂTURILE DINTRE COMPONENTELE SISTEMULUI.....	34
2.2 DESCRIEREA STRUCTURALĂ A SISTEMULUI .....	36
2.2.1 DESCRIEREA STRUCTURII STATICE A SISTEMULUI .....	37
2.2.2 RELATIILE DE DEPENDENȚĂ ÎNTRE COMPONENTELE SISTEMULUI .....	38
2.2.3 MODELAREA ECHIPAMENTELOR MEDIULUI DE IMPLEMENTARE.....	40
3 REALIZAREA SISTEMULUI.....	41
3.1 DESCRIEREA LA NIVEL DE COD PE MODULE .....	42
3.2 TESTAREA SISTEMULUI .....	44
CONCLUZII .....	45

## INTRODUCERE

Gestionarea datelor în orice companie este un proces complicat, luând în considerare că majoritatea bazelor de date în sfera logisticii sunt bazate pe Spreadsheet-uri, Excel-uri sau alte aplicații nesigure, aplicația dată urmează să facă acest proces de gestionare mult mai ușor și sigur.

Aplicația dată urmează să aibă un sistem de logare și înregistrare sigur, cu o parolă de unică folosință, la crearea unui cont nou sau logarea acestuia, pentru a preveni încălcarea datelor și furtul acestora.

La fel urmează să aibă o interfață plăcută la ochi, ușoară de înțeles chiar și pentru oamenii ce nu sunt pricepuți în tehnologii. Aplicația urmează să împartă procesul de gestionare a datelor: adăugarea datelor noi, modificarea acestora și ștergerea acestora, toate acestea sunt împărțite în categorii diferite.

Scopul acestui proiect este de a ușura procesul de gestionare a datelor, prin intermediul creării unui aplicații de tip GUI. Aplicația dată va conține o multitudine de ferestre de tip WindowsForm, fiecare îndeplinind o funcție diferită, cum ar fi inserarea, ștergerea sau modificarea datelor. Această separare a fost efectuată ca scop pentru a evita conglomerarea de informație vizuală și a ușura procesul de utilizare a aplicației. Design-ul aplicației va fi unul destul de simplu, luând în considerare că procesul de modificare a interfeței este cu mult mai ușor, având posibilitatea de a fi distribuit sau modificat pentru a respecta alte norme sau cerințe. Aplicația va fi legată cu o bază de date, unde se va conține toată informația ce urmează să fie gestionată.

Acest proiect este prevăzut pentru companiile ce se ocupă cu gestionarea datelor legate de importul și exportul mărfurilor sau a bunurilor, ce folosesc sistemul de operare Windows, deoarece, la fel ca orice companie această suferă de scurgeri de date, sau în cazuri și mai rele, pierderi de date în urma pierderilor acestora, unde restaurarea acestora poate dura zile întregi și pierderi de sute de mii de dolari.

Produsul dat urmează să efectueze următoarele funcții:

- adăugarea datelor noi;
- modificarea datelor existente;
- ștergerea datelor existente;
- afișarea datelor complete;
- afișarea datelor personale.

## 1 ANALIZA DOMENIULUI DE STUDIU

În urma elaborării proiectului dat, se va obține un sistem de gestionare de date cu interfață grafică și cu o conexiune la baza de date.

Logistica este procesul de planificare și execuție a transportului și depozitării eficiente a mărfurilor de la punctul de origine până la punctul de consum. Scopul logisticii este de a satisface cerințele clienților în timp util și rentabil.

Inițial, logistica a jucat rolul vital de a muta personalul militar, echipamentele și mărfurile. În timp ce logistica este la fel de importantă ca întotdeauna în armată, termenul este astăzi mai frecvent utilizat în contextul deplasării mărfurilor comerciale în cadrul lanțului de aprovizionare.

Multe companii sunt specializate în logistică, oferind serviciul producătorilor, comercianților cu amănuntul și altor industrii cu o mare nevoie de transport de mărfuri. Unii dețin întreaga gamă de infrastructură, de la avioane cu reacție la camioane, depozite și software, în timp ce alții sunt specializați în una sau două părți. FedEx, UPS și DHL sunt furnizori de logistică cunoscuți.

Deși livrarea la timp a pachetelor intacte a fost întotdeauna importantă de-a lungul lanțului de aprovizionare, aceasta a devenit și mai critică în ultimii ani, deoarece comerțul omnicanal, cu livrarea la domiciliu sau cu amănuntul în aceeași zi a produselor personalizate comandate de pe smartphone-uri, devine din ce în ce mai mult. uzual. Furnizorii, producătorii, distribuitorii și comercianții cu amănuntul au fost nevoiți să-și îmbunătățească procesele logistice pentru a satisface cererea de livrare mai rapidă și mai convenabilă a unei varietăți mai mari de mărfuri. De asemenea, au fost nevoiți să-și integreze mai bine procesele și sistemele pentru a îmbunătăți vizibilitatea lanțului de aprovizionare.

Managementul logisticii se referă la existența resurselor potrivite la momentul potrivit pentru a livra mărfurile la locația potrivită, în starea corectă.

În cea mai simplă formă, logistica este importantă deoarece este necesară pentru a furniza materii prime, bunuri și echipamente atât producătorului, cât și consumatorului.

Producătorul nu este logistician și nici consumatorul; și ambii au nevoie de intermediarul logistic care să livreze ceea ce au nevoie.

Un sistem informatic de management al logisticii (LMIS) este un sistem de înregistrări și rapoarte indiferent dacă este pe hârtie sau electronic folosit pentru a agrega, analiza, valida și a afișa datele care pot fi folosite pentru a lua decizii logistice și gestiona lanțul de aprovizionare.

Aplicatia îndeplinită va fi o aplicatie de tip GUI scrisă în limbajul C# utilizând biblioteca ADO.NET.

Aceasta aplicatie face parte din aplicațiile ce gestionează date, fie a clienților, a produselor sau a angajaților. În cazul nostru aplicația urmează să gestioneze datele legate de logistica camioanelor, pentru orice companie ce nu dispune de o modalitate comodă de a executa acest proces.

## **1.1 IMPORTANȚA TEMEI**

Logistica de transporturi cu camioane joacă un rol crucial în economia globală, permițând circulația mărfurilor și a materiilor prime între furnizori, producători, distribuitori și comercianți cu amănuntul. Iată câteva motive pentru care logistica camioanelor este importantă:

- transport eficient: logistica transporturilor asigură transportul bunurilor eficient și la timp, ajutând companiile să îndeplinească cerințele clienților și să își mențină operațiunile lanțului de aprovizionare. Acest lucru este important în special pentru produsele sensibile la timp, cum ar fi bunurile perisabile și consumabilele medicale;

- eficient din punct de vedere al costurilor: Logistica de transport prin camioane este, în general, o modalitate eficientă din punct de vedere al costurilor de a transporta mărfuri pe distanțe scurte și medii în comparație cu alte moduri de transport, cum ar fi aerul sau maritim. Acest lucru îl face o alegere populară pentru multe companii care trebuie să mute bunuri într-o anumită regiune sau țară;

- flexibilitate: logistica de transport este, de asemenea, flexibilă, deoarece se poate adapta rapid la schimbările cererii și ofertei. Acest lucru este important pentru companiile care trebuie să răspundă la schimbările în preferințele clienților, la cererea sezonieră sau la evenimente neprevăzute, cum ar fi dezastrele naturale;

- crearea de locuri de muncă: industria de camioane oferă oportunități de angajare pentru milioane de oameni din întreaga lume, de la șoferi și mecanici până la dispeceri și manageri. Acest lucru ajută la sprijinirea economiilor locale și naționale și oferă indivizilor cariere stabile și pline de satisfacții;

- impactul asupra mediului: În timp ce logistica transporturilor are o amprentă de carbon, se fac eforturi pentru a reduce emisiile prin utilizarea de vehicule mai eficiente din punct de vedere al consumului de combustibil, combustibili alternativi și practici ecologice;

În general, logistica transporturilor joacă un rol vital în economia globală, facilitând circulația mărfurilor și a materiilor prime, oferind oportunități de angajare și sprijinind întreprinderile de toate dimensiunile.

Scrierea unei aplicații de bază de date ușor de utilizat pentru logistică poate aduce mai multe beneficii unei companii de logistică. Iată câteva dintre motivele cheie pentru care este important să aveți o aplicație de bază de date ușor de utilizat:

- procese simplificate: o aplicație de bază de date ușor de utilizat poate ajuta la eficientizarea proceselor din cadrul companiei de logistică. Poate oferi o locație centrală pentru urmărirea și gestionarea datelor legate de inventar, expediere și livrare, ceea ce poate ajuta la reducerea erorilor și la îmbunătățirea eficienței;

- precizie sporită: Cu o aplicație de bază de date ușor de utilizat, companiile de logistică pot reduce erorile și pot îmbunătăți acuratețea datelor lor. Acest lucru poate ajuta la evitarea greșelilor costisitoare, cum ar fi expedierea unui produs greșit sau livrarea la o adresă greșită;

- servicii îmbunătățite pentru clienți: Având informații precise și actualizate în baza de date, companiile de logistică pot oferi clienților un serviciu mai bun pentru clienți.

## **1.2 SISTEME SIMILARE CU PROIECTUL REALIZAT**

Sisteme asemănătoare cu sistemul dat sunt:

- MSSQL;
- Microsoft Excel;
- Microsoft Access.

Microsoft SQL Server este un sistem de gestionare a bazelor de date relaționale dezvoltat de Microsoft. Ca server de baze de date, este un produs software cu funcția principală de stocare și preluare a datelor, așa cum sunt solicitate de alte aplicații software, care pot rula fie pe același computer, fie pe alt computer dintr-o rețea (inclusiv Internet).

Microsoft comercializează cel puțin o duzină de ediții diferite ale Microsoft SQL Server, destinate unor audiențe diferite și pentru sarcini de lucru, de la aplicații mici pentru o singură mașină până la aplicații mari orientate spre Internet cu mulți utilizatori concurenți.

Toate operațiunile care pot fi invocate pe SQL Server îi sunt comunicate printr-un format definit de Microsoft, numit Tabular Data Stream (TDS). TDS este un protocol de nivel de aplicație,



utilizat pentru a transfera date între un server de baze de date și un client. Pachetele TDS pot fi încadrate în alte protocoale dependente de transportul fizic, inclusiv TCP/IP, conducte numite și partajate. memorie. În consecință, accesul la SQL Server este disponibil prin aceste protocoale. În plus, API-ul SQL Server este, de asemenea, expus prin serviciile web [1].

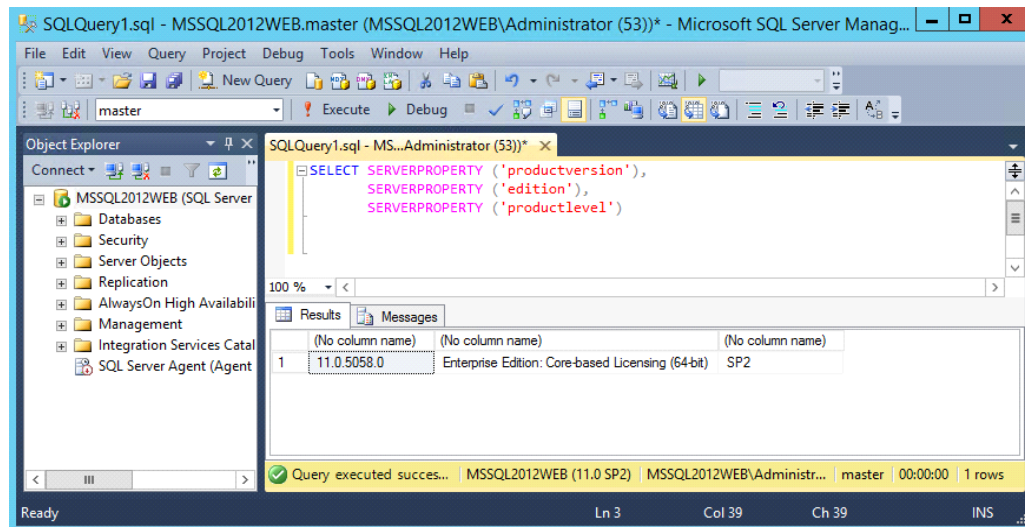


Figura 1 - Interfața aplicației Microsoft SQL Server

Microsoft Excel este o foaie de calcul dezvoltată de Microsoft pentru Windows, macOS, Android și iOS. Dispune de capacități de calcul sau de calcul, instrumente de grafică, tabele pivot și un limbaj de programare macro numit Visual Basic pentru aplicații (VBA). Excel face parte din suita de software Microsoft Office.

Microsoft Excel are caracteristicile de bază ale tuturor foilor de calcul, folosind o grilă de celule aranjate pe rânduri numerotate și coloane cu nume de litere pentru a organiza manipulările de date, cum ar fi operațiuni aritmetice. Are o baterie de funcții furnizate pentru a răspunde nevoilor statistice, de inginerie și financiare. În plus, poate afișa date sub formă de grafice cu linii, histogramme și diagrame și cu un afișaj grafic tridimensional foarte limitat. Are un aspect de programare, Visual Basic pentru aplicații, permițând utilizatorului să folosească o mare varietate de metode numerice, de exemplu, pentru rezolvarea ecuațiilor diferențiale ale fizicii matematice, și apoi raportarea rezultatelor înapoi în foaia de calcul. Are, de asemenea, o varietate de caracteristici interactive care permit interfețe cu utilizatorul care pot ascunde complet foaia de calcul de utilizator, astfel încât foaia de calcul se prezintă ca o așa-numită aplicație sau sistem de sprijinire a deciziilor (DSS), printr-o interfață de utilizator concepută personalizat, pentru de

exemplu, un analizor de stoc sau, în general, ca instrument de proiectare care pune întrebări utilizatorului și oferă răspunsuri și rapoarte. Într-o realizare mai elaborată, o aplicație Excel poate sonda automat baze de date externe și instrumente de măsurare folosind un program de actualizare, să analizeze rezultatele, să realizeze un raport Word sau o expunere de diapozitive PowerPoint și să trimită prin e-mail aceste prezentări în mod regulat unui Lista participanților. Excel nu a fost conceput pentru a fi folosit ca bază de date [2].

	A	B	C	D	E	F	G	H	I	J
	Date	Services	Products	Total	7-Day Avg	Goal				
2	2016-08-19	1345		2709	2,709	2500				
3	2016-08-20	9700		2709	2,709	2500				
4	2016-08-21		1259	-321	1,699	2500				
5	2016-08-22	4578		-27	1,268	2500				
6	2016-08-23		3336	0	1,014	2500				
7	2016-08-24		3799	0	845	2500				
8	2016-08-25	987		416	784	2500				
9	2016-08-26	8712	1205	1333	587	2500				
10	2016-08-27	4509	2424	0	200	2500				
11	2016-08-28	6312		2330	579	2500				
12	2016-08-29			2814	985	2500				
13	2016-08-30			81	996	2500				
14	2016-08-31			443	1,060	2500				
15	2016-09-01			1579	1,226	2500				
16	2016-09-02			3266	1,502	2500				

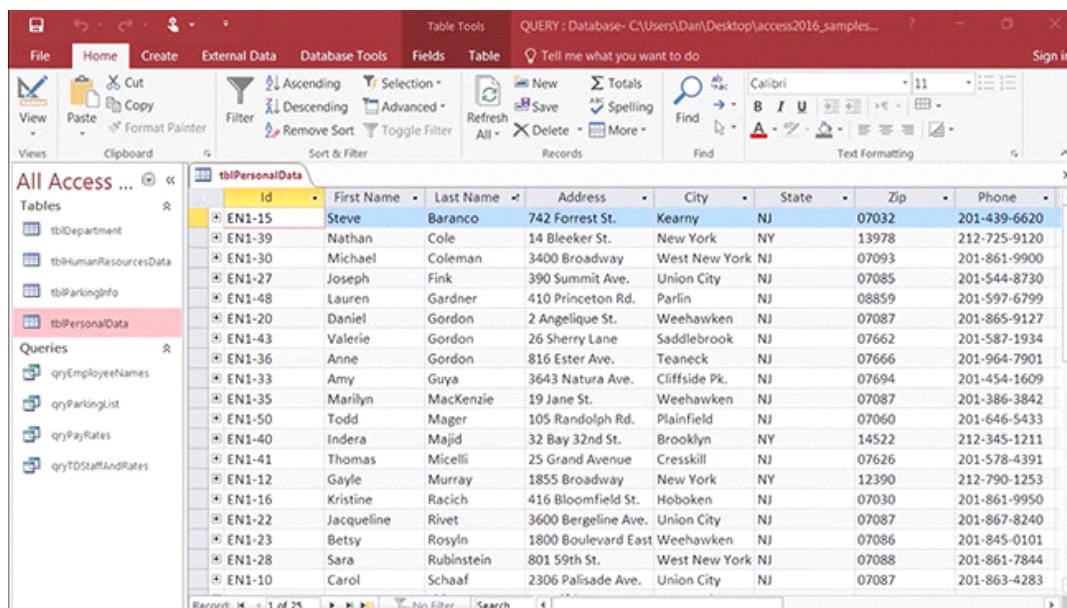
Figura 2 - Interfața aplicației Microsoft Excel

Microsoft Access este un sistem de gestionare a bazelor de date (DBMS) de la Microsoft care combină motorul de bază de date relațional Access (ACE) cu o interfață grafică de utilizator și instrumente de dezvoltare software. Este membru al suitei de aplicații Microsoft 365, incluse în edițiile Professional și edițiile superioare sau vândute separat [3].

Microsoft Access stochează datele în propriul format, bazat pe Access Database Engine (fostul Jet Database Engine). De asemenea, poate importa sau conecta direct la datele stocate în alte aplicații și baze de date

Dezvoltatorii de software, arhitecții de date și utilizatorii cu putere pot folosi Microsoft Access pentru a dezvolta aplicații software. Ca și alte aplicații Microsoft Office, Access este acceptat de Visual Basic for Applications (VBA), un limbaj de programare bazat pe obiecte care poate face referire la o varietate de obiecte, inclusiv DAO (Obiecte de acces la date), obiecte de date ActiveX

și multe alte componente ActiveX. Obiectele vizuale utilizate în formulare și rapoarte își expun metodele și proprietățile în mediul de programare VBA, iar modulele de cod VBA pot declara și apela operațiunile sistemului de operare Windows [4].



Id	First Name	Last Name	Address	City	State	Zip	Phone
EN1-15	Steve	Baranco	742 Forrest St.	Kearny	NJ	07032	201-439-6620
EN1-39	Nathan	Cole	14 Bleeker St.	New York	NY	13978	212-725-9120
EN1-30	Michael	Coleman	3400 Broadway	West New York	NJ	07093	201-861-9900
EN1-27	Joseph	Fink	390 Summit Ave.	Union City	NJ	07085	201-544-8730
EN1-48	Lauren	Gardner	410 Princeton Rd.	Parlin	NJ	08859	201-597-6799
EN1-20	Daniel	Gordon	2 Angelique St.	Weehawken	NJ	07087	201-865-9127
EN1-43	Valerie	Gordon	26 Sherry Lane	Saddlebrook	NJ	07662	201-587-1934
EN1-36	Anne	Gordon	816 Ester Ave.	Teaneck	NJ	07666	201-964-7901
EN1-33	Amy	Guya	3643 Natura Ave.	Cliffside Pk.	NJ	07694	201-454-1609
EN1-35	Marilyn	MacKenzie	19 Jane St.	Weehawken	NJ	07087	201-386-3842
EN1-50	Todd	Mager	105 Randolph Rd.	Plainfield	NJ	07060	201-646-5433
EN1-40	Indera	Majid	32 Bay 32nd St.	Brooklyn	NY	14522	212-345-1211
EN1-41	Thomas	Micelli	25 Grand Avenue	Cresskill	NJ	07626	201-578-4391
EN1-12	Gayle	Murray	1855 Broadway	New York	NY	12390	212-790-1253
EN1-16	Kristine	Racich	416 Bloomfield St.	Hoboken	NJ	07030	201-861-9950
EN1-22	Jacqueline	Rivet	3600 Bergeline Ave.	Union City	NJ	07087	201-867-8240
EN1-23	Betsy	Roslyn	1800 Boulevard East	Weehawken	NJ	07086	201-845-0101
EN1-28	Sara	Rubinstein	801 59th St.	West New York	NJ	07088	201-861-7844
EN1-10	Carol	Schaaf	2306 Palisade Ave.	Union City	NJ	07087	201-863-4283

Figura 3 - Interfața aplicației Microsoft Access

Luând în considerare cât de monotonic poate fi procesul de gestionare a datelor, în special dacă utilizăm softuri cum ar fi Excel, sau SQL, am decis să creăm o aplicație care face procesul dat mult mai ușor și plăcut, în special procesul de modificare a datelor este mult mai ușor de efectuat, fără a trece prin mai mulți pași monotoni și inutili, pe lângă aceasta, mai există și o fereastră care redă informația într-un mod mult mai concis și ușor de înțeles, pentru a reduce riscul de introducere a datelor incorecte.

Cum am mai descris mai sus, proiectul dat poate fi comparat cu Microsoft Excel, Microsoft Access dar și SQL Server, toate acestea au plusurile sale dar au și niște neajunsuri, unul dintre cele mai importante puncte ale acestui proiect va fi securitatea datelor, pe lângă posibilitatea de criptare a datelor deja prezentă în SQL Server, aș vrea să ridic nivelul de securitate, ascunzând soft-ul dat sub o formă de logare, preferabil 2FA, pentru a reduce riscul de pierdere sau scurgere de date.

## 1.3 SCOPUL, OBIECTIVELE ȘI CERINȚELE SISTEMULUI

Scopul acestui proiect este de a ușura procesul de gestionare a datelor, prin intermediul creării unui aplicații de tip GUI. Aplicația dată va conține o multitudine de ferestre de tip WindowsForm, fiecare îndeplinind o funcție diferită, cum ar fi inserarea, ștergerea sau modificarea datelor. Această separare a fost efectuată ca scop pentru a evita conglomerarea de informație vizuală și a ușura procesul de utilizare a aplicației. Design-ul aplicației va fi unul destul de simplu, luând în considerare că procesul de modificare a interfeței este cu mult mai ușor, având posibilitatea de a fi distribuit sau modificat pentru a respecta alte norme sau cerințe. Aplicația va fi legată cu o bază de date, unde se va conține toată informația ce urmează să fie gestionată.

### Interfețe cu utilizatorul

Proiectul dat pus spre implementare, reprezintă o platformă desktop ce deține componente atât back-end cât și front-end. Interfața de utilizator a produsului final, luând în considerare ca aceasta urmează să fie creată cu ajutorul IDE-ului Visual Studio dar și Microsoft SQL Management Studio, va fi compatibilă cu orice S.O, cum ar fi Windows, Linux sau Ubuntu.

### Interfețe hardware

Deoarece aplicația trebuie să ruleze pe internet, tot hardware-ul va avea nevoie pentru a se conecta la internet și la fel îi va fi necesar pentru mediul de dezvoltare un sistem de operare.

### Interfețe software

Pentru asigurarea realizării cu succes a proiectului, va fi necesar de unul de sistemele de operare enumerate mai jos, mediul de dezvoltare Microsoft Visual Studio dar și Microsoft SQL Management Studio, pentru gestionarea datelor.

### Protocoale și interfețe de comunicație

Pentru asigurarea funcționalității procesului de autentificare securizată, va fi nevoie de o conexiune de internet stabilă, pentru a primi toate notificările în timp real.

### Cerințe de performanță

Cerințele minime ale sistemului sunt:

- sistemul de operare: Windows / Linux / Ubuntu;
- RAM: 4 GB;
- procesor i3 generația 6;
- prezența sistemului de gestionare a datelor MSSQL.

### Cerințe de siguranță

Platforma desktop deține un sistem de logare și înregistrare securizat prin intermediul parolei de unică folosință, adițional datele de logare sunt incryptate. În cazul bazei de date, sunt create back-up-uri zilnice, săptămânale și lunare dar și planuri de mentenanță.

### Cerințe de securitate

Platforma desktop deține 3 actori: angajații, administrator-ul bazei de date și manager. Prin urmare există posibilitatea de a gestiona datele în sfera generală, dar și sfera internă, în cazul în care apare o înregistrare ilegală în sistemul dat.

### Atribute de calitate software

- scalabilitate– platforma desktop va putea prelucra date masive, fără a degrada funcționalitatea sistemului;
- utilizabilitatea – platforma va satisface utilizatorii prin ușurarea procesului de licitare;
- disponibilitatea – platforma va fi disponibilă 24/7, excluzând lucrul de mentenanță;
- fiabilitatea – platforma va conține erori minime, iar în cazul în care sunt depistate erori cruciale,

## 2 MODELAREA ȘI PROIECTAREA SISTEMUL INFORMATIC

Proiectarea unui sistem este un proces esențial în dezvoltarea oricărui produs software sau hardware. Acest proces implică și componentele definite și cerințele sistemului, identificarea lor necesară pentru construirea sistemului, stabilitatea relațiilor și interacțiunile dintre aceste componente, precum și dezvoltarea unei arhitecturi adecvate care să permită implementarea sistemului.

Proiectarea sistemului este adesea împărțită în mai multe etape, cum ar fi cerințele, proiectarea arhitecturii și proiectarea detaliată. În etapa de analiză a cerințelor, se vor identifica obiectivele sistemului și se vor defini specificațiile funcționale și non-funcționale necesare pentru satisfacerea acestora. În următoarea etapă, se va dezvolta o arhitectură generală pentru sistem, stabilindu-se componentele și relațiile dintre acestea, iar în faza finală de proiectare detaliată, se va stabili cum vor fi implementate și integrate componentele pentru a construi sistemul final.

Proiectarea unui sistem un proces critic în dezvoltarea de software și hardware și este important să fie realizat cu atenție și meticulozitate pentru a se asigura că sistemul final este adecvat pentru nevoile și nevoile utilizatorilor.

A fost aleasă metoda dezvoltării orientate pe obiecte (OOD): Acest lucru implică dezvoltarea prin definirea obiectelor și a relațiilor dintre acestea. În acest lucru, sistemul este văzut ca un set de obiecte care comunică între ele și care pot fi reutilizate în alte aplicații.

Au fost alese următoarele limbaje de proiectare:

- diagrama de flux;
- diagrama de activitate;
- diagrama de clasă;
- diagrama de caz;
- diagrama de secvență.

Diagrama de flux este un limbaj grafic utilizat pentru a reprezenta algoritmi și procesele. Acesta este folosit pentru a descrie etapele proceselor și pentru a identifica eventualele probleme și deficiențe.

Diagrama de activitate este un limbaj grafic utilizat pentru a reprezenta fluxul de activitate într-un proces sau sistem

Diagrama de clasă este un limbaj grafic utilizat pentru a reprezenta clasele, atributele și relațiile dintre ele.

Diagrama de secvență este un limbaj grafic utilizat pentru a reprezenta secvența acțiunilor și evenimentelor dintr-un proces sau sistem.

Diagrama de caz de utilizare este un limbaj grafic utilizat pentru a reprezenta interacțiunea între sistem și utilizatorii săi.

Visual Studio este un mediu integrat de dezvoltare (IDE) dezvoltat de Microsoft, care oferă o suită de instrumente și servicii pentru dezvoltarea de aplicații software pentru diverse platforme, inclusiv Windows, Android, iOS și web.

Mediul de dezvoltare Visual Studio include un editor de cod, un compilator, un debugger, un designer de interfață grafică și alte instrumente pentru gestionarea și testarea codului sursă. Acesta oferă, de asemenea, acces la unelte de dezvoltare colaborative, cum ar fi sistemul de control al versiunilor, instrumentele de gestionare a proiectelor și instrumentele de analiză a performanței.

## **2.1 DESCRIEREA COMPORTAMENTALĂ A SISTEMULUI**

Descrierea comportamentală este o tehnică utilizată în ingineria software pentru a defini comportamentul unui sistem de software prin intermediul unui set de scenarii sau cazuri de utilizare. Aceasta descrie modul în care sistemul se comportă în diferite situații sau circumstanțe, precum și interacțiunea sa cu utilizatorii și alte sisteme.

Scopul descrierii comportamentale este de a dezvolta o înțelegere clară și precisă a modului în care sistemul trebuie să funcționeze, astfel încât să se poată crea un plan detaliat de dezvoltare și testare. Aceasta este esențială pentru a asigura că sistemul îndeplinește cerințele utilizatorilor și că este robust, fiabil și ușor de întreținut.

O descriere comportamentală poate include o varietate de artefacte, cum ar fi diagrame de cazuri de utilizare, diagrame de secvență, diagrame de stare și diagrame de activitate. Aceste artefacte sunt utilizate pentru a descrie comportamentul sistemului într-un mod vizual, ușor de înțeles și de comunicat.

În general, descrierea comportamentală este o componentă cheie a procesului de dezvoltare a software-ului și este utilizată în toate etapele, de la analiza și proiectarea până la testarea și implementarea sistemului. Prin definirea și clarificarea comportamentului sistemului, aceasta ajută la reducerea erorilor și defectelor în timpul dezvoltării și asigură că sistemul îndeplinește cerințele și așteptările utilizatorilor.

Pentru a dezvolta o descriere comportamentală precisă și completă, este important să se efectueze o analiză exhaustivă a cerințelor și să se determine funcționalitățile sistemului. Aceasta implică identificarea diferitelor cazuri de utilizare și a interacțiunilor dintre utilizatori și sistem.

Descrierea comportamentală este o componentă critică a dezvoltării software-ului și poate ajuta la asigurarea faptului că sistemul îndeplinește cerințele utilizatorilor și este capabil să funcționeze într-un mod fiabil și robust.

### **2.1.1 IMAGINEA GENERALĂ ASUPRA SISTEMULUI**

În figura dată este reprezentat schema aplicației, use-case-urile și relațiile dintre ele. Relațiile de bază utilizate sunt <<include>>, <<invoke>>

În sistemul dat putem observa relațiile ce au loc între un angajat și manager, dar și funcțiile împlinite de către administrator



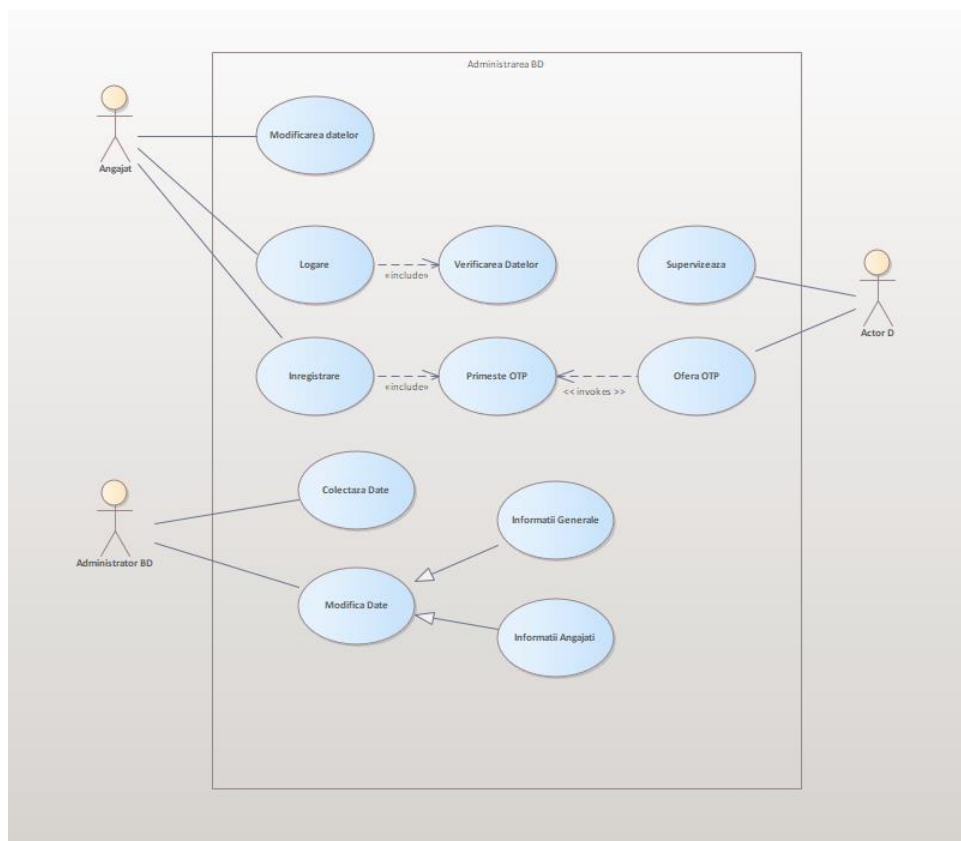


Figura 4 - Diagrama Sistemului Informational

În diagrama de mai jos este descris procesul de colectare a datelor de către administratorul bazei de date. Acesta colectează date de 3 tipuri:

- date angajați;
- date logistice;
- date șoferi.

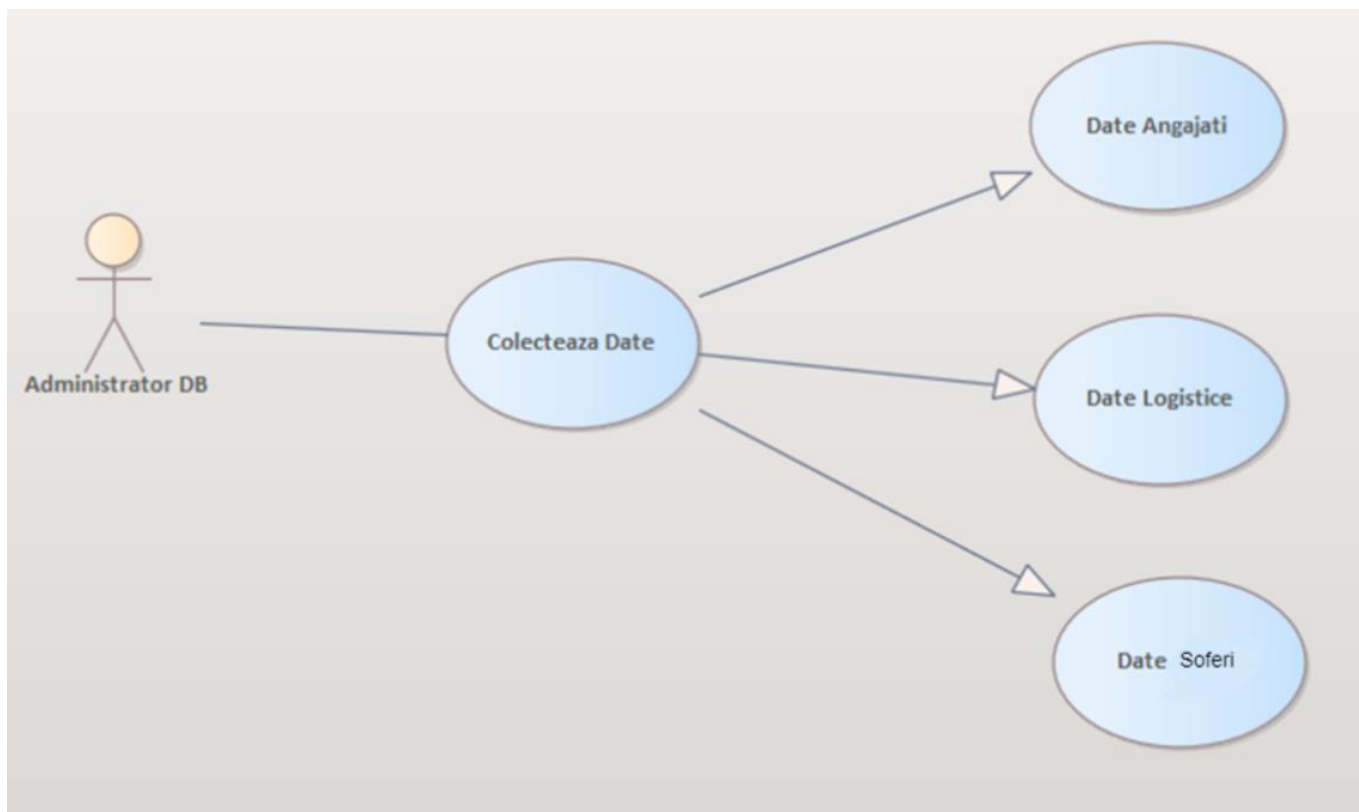


Figura 5 - Diagrama procesului de colectare a datelor

### 2.1.2 MODELAREA VIZUALĂ A FLUXURILOR

În figura de mai jos este reprezentată diagrama de activitate a afisare a datelor, în urma căreia avem activitatea inițială, urmată de logare, afisarea ferestrei principale, accesarea ferestrei de afisare a datelor.

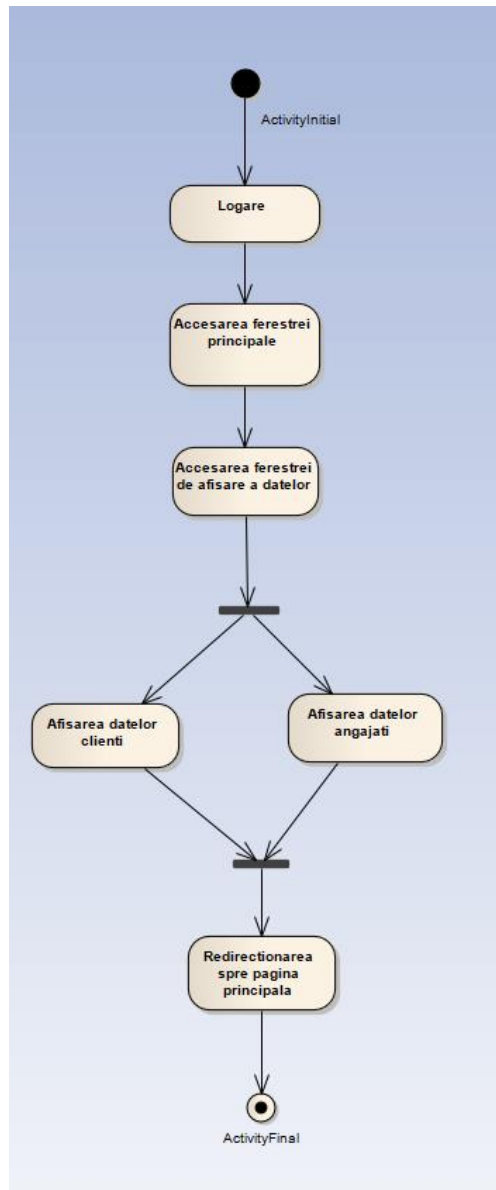


Figura 6 - Diagrama de activitate a afisare a datelor

In figura de mai jos este reprezentata diagrama de activitate reprezentata prin activitatea initiala, dupa procesul de logare, avem decizia de a accesa aplicatia ca manager sau ca angajat. In depedenta de decizia luata, putem modifica anumite date, unde apoi sunt modificate datele si sunt afisate in fereastră principala, unde ajungem la activitatea finala.

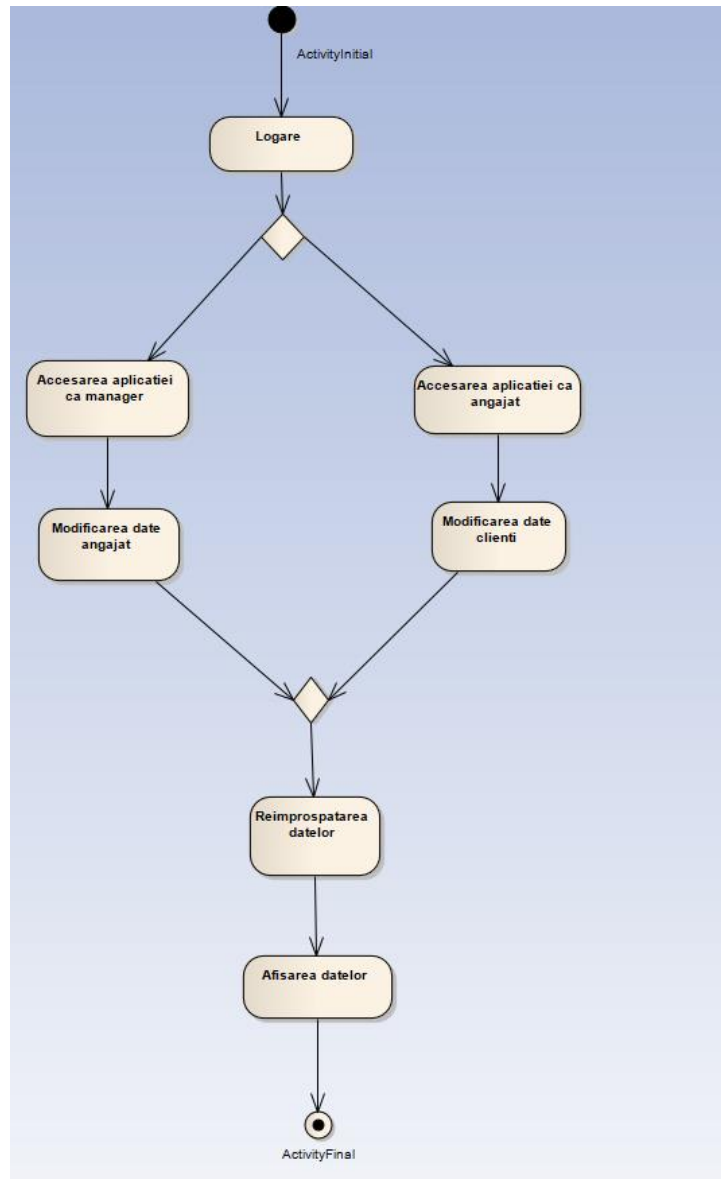


Figura 7 - Diagrama de activitate a modificarea datelor de catre manager si angajat

### 2.1.3 STĂRILE DE TRANZACȚIE A SISTEMULUI

In figura de mai jos este diagrama de stare a sistemului de modificare a datelor angajatilor.

Procesul este descris in automatul de modificare a datelor, ce incepe cu accesarea aplicatiei si modificarea urmata de acesarea ferestrei de modificare a datelor, introducerea datelor ce urmeaza sa fie modificate si salvarea acestora. In urma acestui process, fereastra de modificare a datelor se inchide, iar datele modificate sunt afisate in fereastra principala.

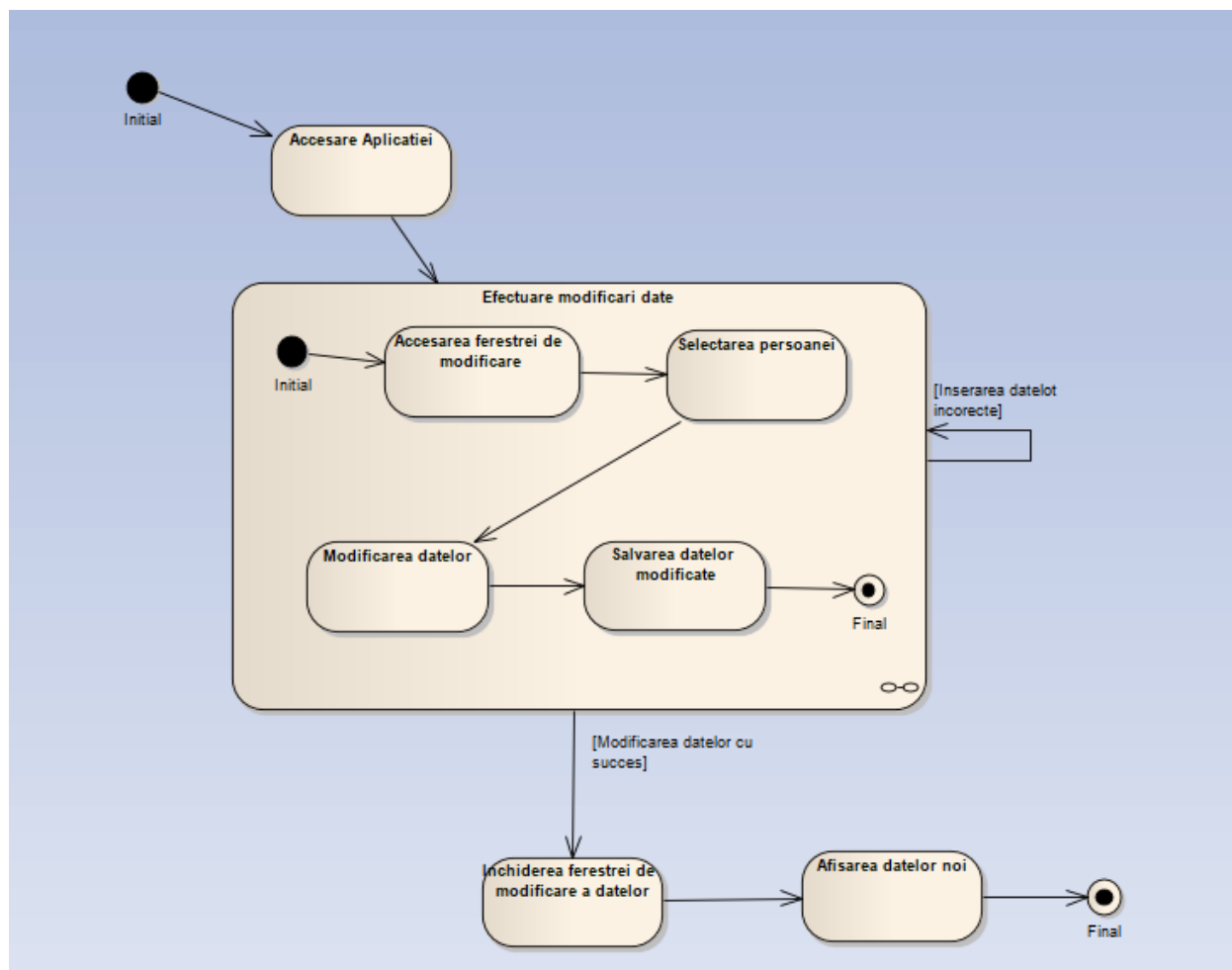


Figura 8 - Diagrama de stare a procesului de modificare a datelor

În figura alăturată este diagrama de stare a sistemului de modificare a datelor angajaților.

Procesul este descris în automatul de modificare a datelor, ce începe cu logarea în sistem și accesarea aplicației și modificare a numelui, telefonului, email-ului, zborului și adresei. Introducerea datelor ce urmează să fie modificate și salvarea acestora. În cazul în care datele sunt eronate, va apărea o eroare, în cazul în care datele au fost introduse cu succes, datele sunt modificate și procesul de modificare se încheie.

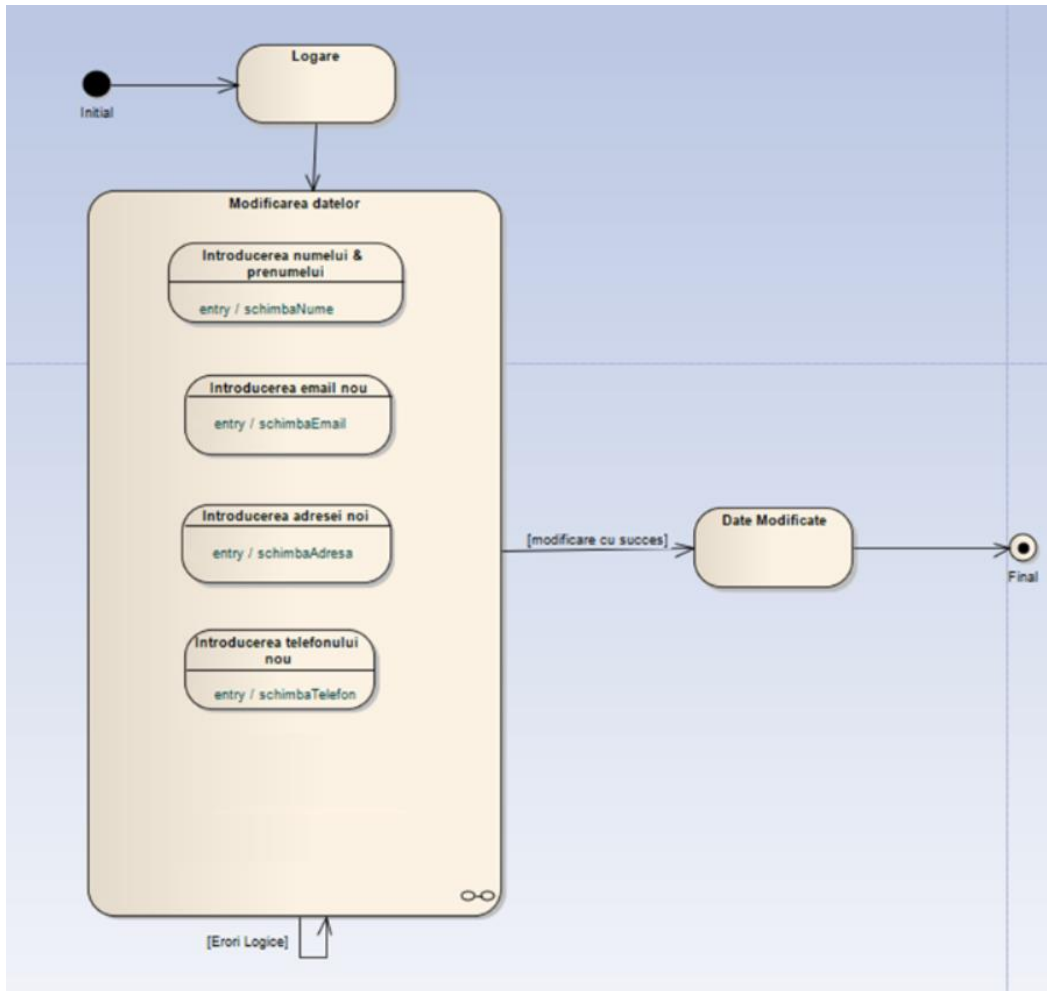


Figura 9 - Diagrama de stare a modificarea datelor clientlor

## 2.1.4 DESCRIEREA SCENARIILOR DE UTILIZARE A APLICAȚIEI

În figura de mai jos este reprezentată diagrama de secvență pentru autentificarea utilizatorului în client. Pentru accesarea paginii principale,

“Angajat”-ul fiind un actor, accesează forma de autentificare, care la rândul ei trimite un call spre server și primește un răspuns pozitiv sau negativ în funcție de datele introduse.

Dacă autentificarea a avut loc cu succes, aplicația va afișa forma principală, iar în caz contrar, un mesaj de eroare.

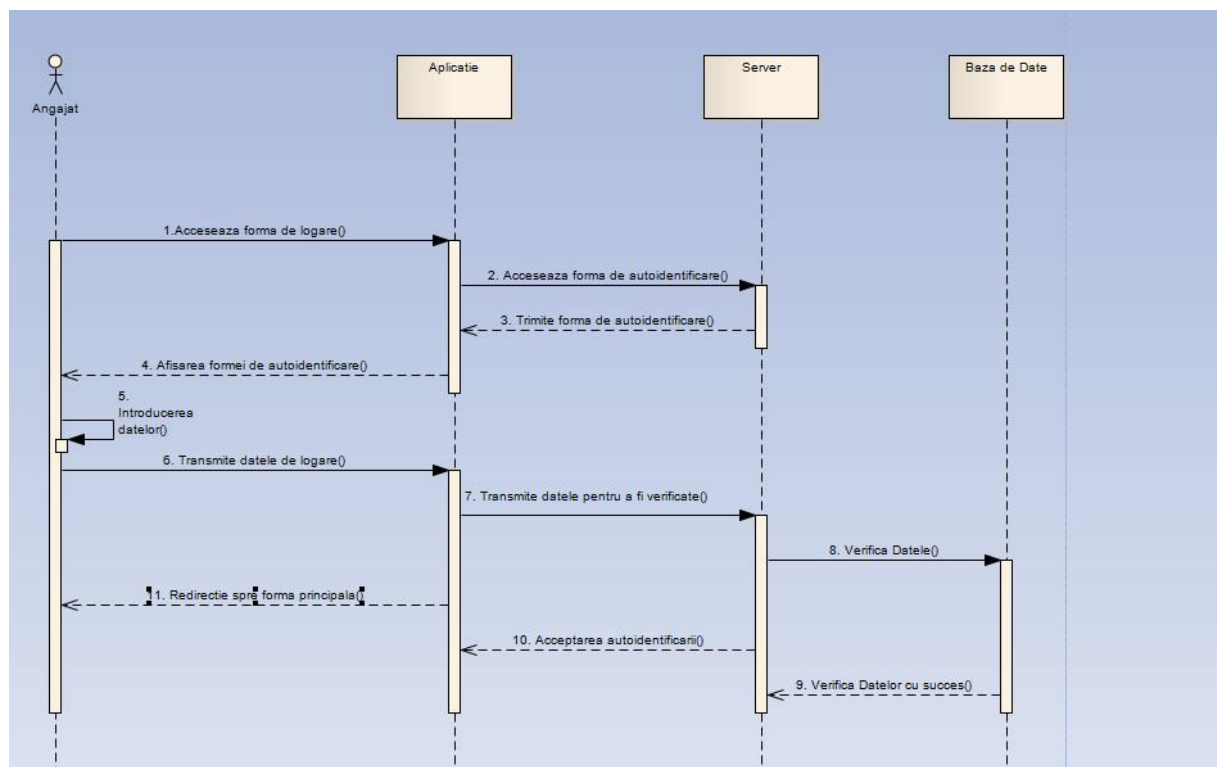


Figura 10 - Diagrama Autentificarii

In figura alaturată este reprezentata diagrama inregistrarii unui cont nou.

Actorul “Angajat” acceseaza forma de inregistrare, iar aplicatia la randul ei trimite un request spre server pentru afisarea formei de inregistrare, dupa prelucrarea requestului serverul returneaza forma care la randul ei se afiseaza pe ecranul utilizatorului. Dupa afisarea formei de inregistrare, utilizatorul completeaza toate campurile necesare, si le trimite spre server prin intermediul aplicatiei, care la fel creaza un request spre server, iar serverul spre baza de date, datele ce ajung in baza de date sunt inregistrate iar baza de date la randul ei returneaza inapoi un raspuns de success/eroare spre server, serverul primind mesajul il prelucreaza si decide ce forma sa returneze, in caz ca raspunsul este cu success, manager-ul primeste o notificare pentru a trimite o cheie de autotentificare angajatului, iar dupa introducerea cu success a cheii, din nou se trimite un request spre server care in depdenta de success/eroare va afisa alte forme.

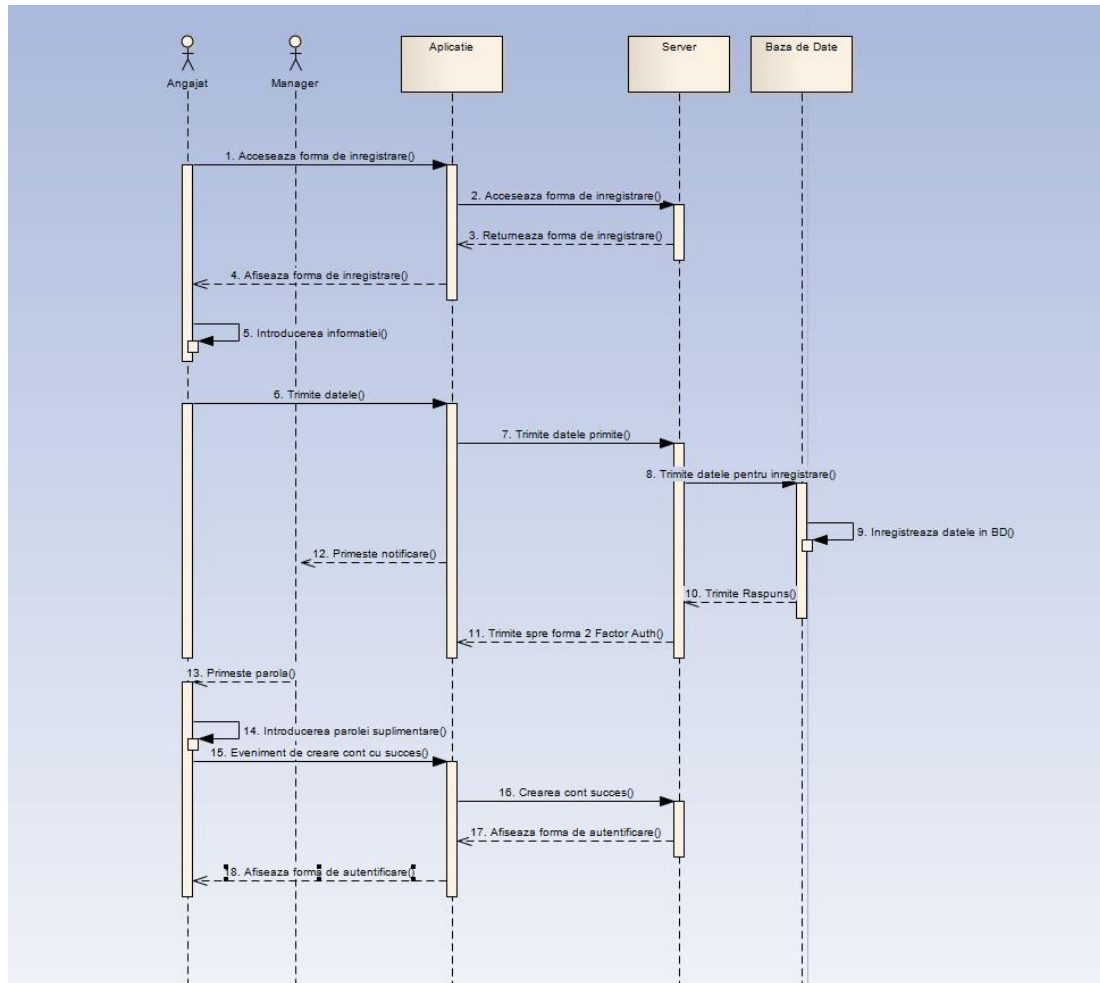


Figura 11 - Diagrama procesului de inregistrare

In figura de mai jos este reprezentata diagrama cererii informatiei a angajatilor de catre manager.

Actorul Manager, acceseaza forma data vizibila doar lui, dupa care se creaza un request de afisare a informatiei de catre server spre baza de date, aceasta in urma acestui request, trimite informatia angajatilor spre server, iar serverul la urma lui trimite aplicatiei, iar aceasta afiseaza in forma vizuala toata informatia angajatilor managerului.



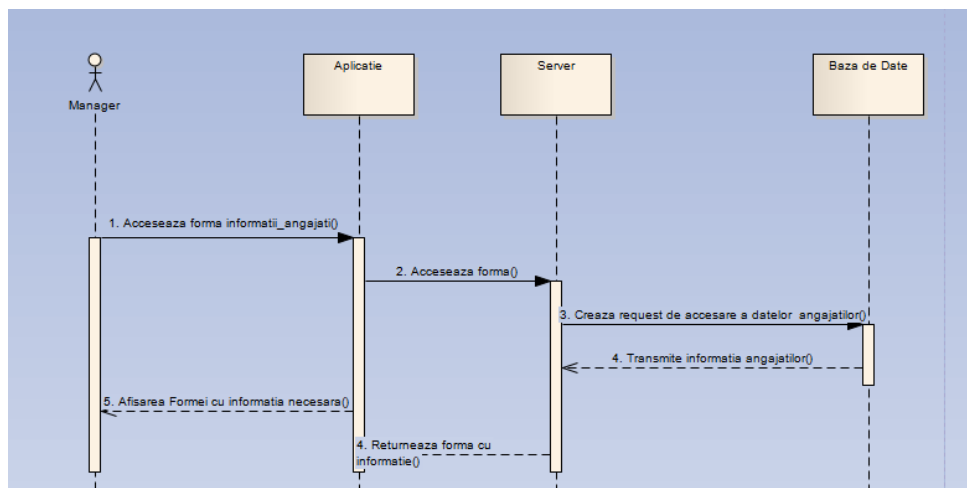


Figura 12 - Diagrama cererii informatiei a angajatilor

### 2.1.5 FLUXURILE DE MESAJE ȘI LEGĂTURILE DINTRE COMPONENTELE SISTEMULUI

În figura de mai jos este reprezentată diagrama de colaborare la nivel de exemplu pentru procesul de autentificare a managerului

În procesul de autentificare, primul pas este accesarea ferestrei de logare de către actorul “Manager”. După care aplicația trimite o cerere pentru forma de logare, după o returnează actorului spre îndeplinire. În urma îndeplinirii acestei forme, datele introduse este preluată de către obiectul “Aplicație” și trimisă la verificare spre obiectul “Baza de date” prin intermediul obiectului “Server”. După verificarea veridicității datelor, obiectul “Baza de date” returnează un mesaj cu ajutorul căruia obiectul “Server” decide să afișeze următoarea fereastră principală actorului sau nu, în cazul în care este efectuată cu succes, se deschide formularul ce cere parola de unică folosință, actorul după ce face o cerere spre aplicația de a primi parola, aceasta la rândul ei face o cerere către server, care automat creează o parola de unică folosință, care o trimite telefonului, acesta la urmă afișează parola, iar actorul introduce această parola. În urma acestora, aplicația creează o cerere de verificare a parolei de u.f. spre server, acesta verifică parola de u.f., iar în cazul în care parola este introdusă corect, acesta trimite un răspuns spre aplicație, care la rândul ei, trimite un răspuns actorului.

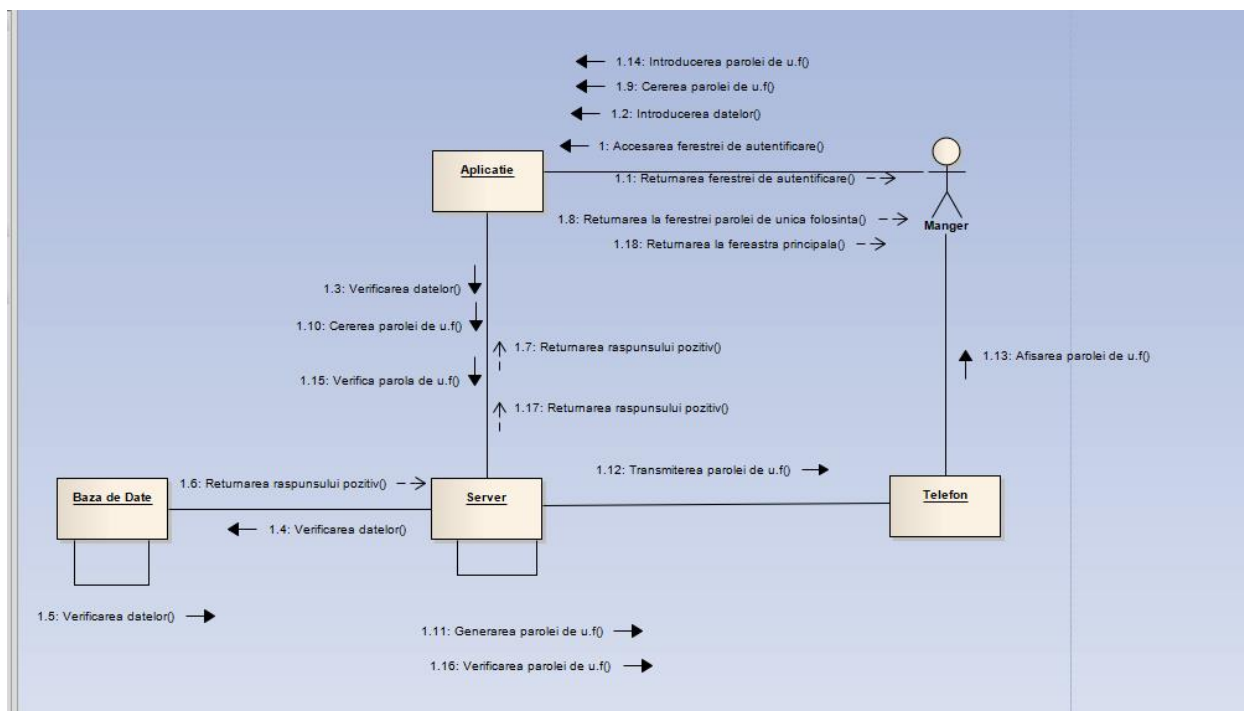


Figura 13 - Autentificarea Managerului (nivel de exemplu)

În figura alăturată este reprezentată diagrama schimbarea datelor de către administrator.

Actorul “Administrator” modifică datele angajaților, după care obiectul “Aplicatie” trimite o cerere de modificare a acestora către obiectul “Server”, care la rândul lui face aceeași chestie către obiectul “Baza de date”, aceasta verifică și schimbă datele dorite de către administrator, și după transmite un răspuns pozitiv cu datele modificate către Server. Server-ul la rândul lui transmite aceste date Aplicatiei, iar aplicația afișează informația schimbată Administratorului.

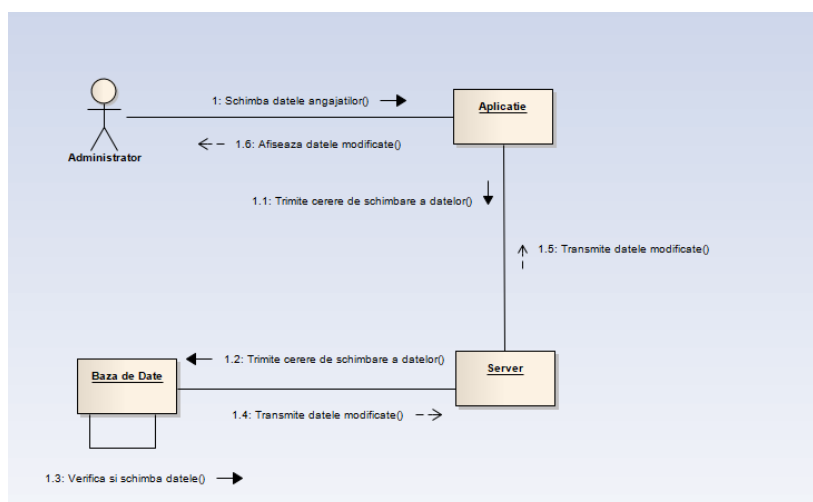


Figura 14 - Diagrama schimbării datelor (nivel de exemplu)

## 2.2 DESCRIEREA STRUCTURALĂ A SISTEMULUI

Descrierea structurală se referă la o metodă de analiză a unui sistem sau a unei entități prin examinarea componentelor sale individuale și a modului în care acestea interacționează și formează structura sa. Această abordare este folosită într-o gamă largă de domenii, cum ar fi ingineria, arhitectura, psihologia și biologia.

Descrierea structurală poate include o serie de elemente, cum ar fi:

- componente: Analiza componentelor care formează sistemul sau entitatea, precum și modul în care acestea se conectează între ele;
- relații: Analiza relațiilor dintre componentele sistemului sau entității, precum și modul în care acestea influențează comportamentul și funcționarea generală;
- funcții: Identificarea funcțiilor sistemului sau entității și modul în care acestea se îndeplinesc prin intermediul componentelor și relațiilor dintre ele;
- fluxuri: Analiza fluxurilor de informații, energie sau alte resurse prin sistemul sau entitatea în cauză;
- configurație: Identificarea configurației sistemului sau entității, inclusiv dimensiunile, orientarea și relația dintre componentele sale;
- proprietăți: Identificarea proprietăților sistemului sau entității, cum ar fi puterea, stabilitatea și eficiența.

În domeniul IT, descrierea structurală poate fi utilizată pentru a analiza și proiecta sisteme software și hardware. Aceasta poate fi utilă pentru a înțelege modul în care diferitele componente ale unui sistem se leagă între ele și cum acestea contribuie la performanța generală a sistemului. De asemenea, poate fi utilizată pentru a dezvolta modele de arhitectură software și hardware, care să fie utilizate în proiectarea și dezvoltarea de soluții de IT.

În general, descrierea structurală poate fi folosită pentru a dezvolta un model de arhitectură software sau hardware care să asigure performanță, securitate și fiabilitate, iar în același timp să fie ușor de înțeles și de modificat. De asemenea, poate fi folosită pentru a identifica punctele critice ale unui sistem și pentru a îmbunătăți performanța și scalabilitatea acestuia.

## 2.2.1 DESCRIEREA STRUCTURII STATICE A SISTEMULUI

În figura de mai sus este diagrama de clasă sistemul de căutare și modificare a datelor angajaților.

Administratorul bazei de date poate accesa și actualiza datele angajaților. Acesta la fel poate face o căutare a datelor angajaților după ID-ul angajatului, numele angajatului sau prenumele angajatului. Clasa Căutare îndeplinește funcția de agerare către clasa DateAngajat. Partea DateAngajat nu poate exista fără partea întreaga BazadeDate.

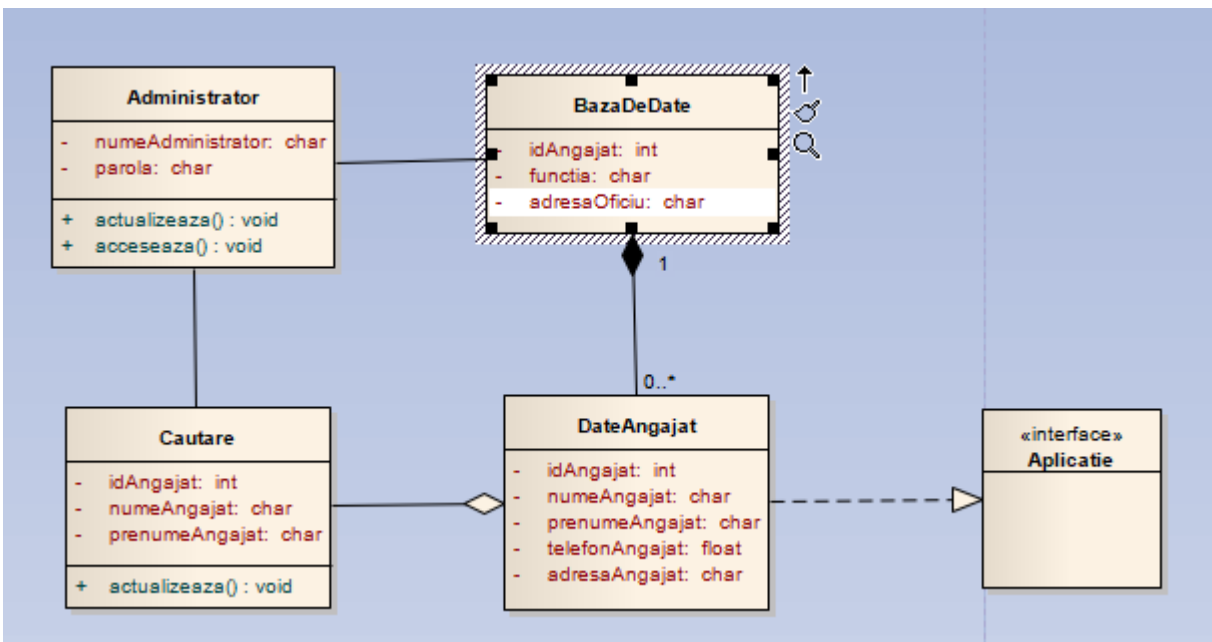


Figura 15 - Diagrama de clasă sistemul de căutare și modificare a datelor angajaților

În figura ce urmează este diagrama de clasă sistemul utilizatori.

Observăm că rolurile Angajat, Manager & Administrator sunt generalizări a clasei Utilizator, clasa Angajat moștenind atributul email, pe când Manager-ul și Administratorul moștenesc accesul privilegiat.

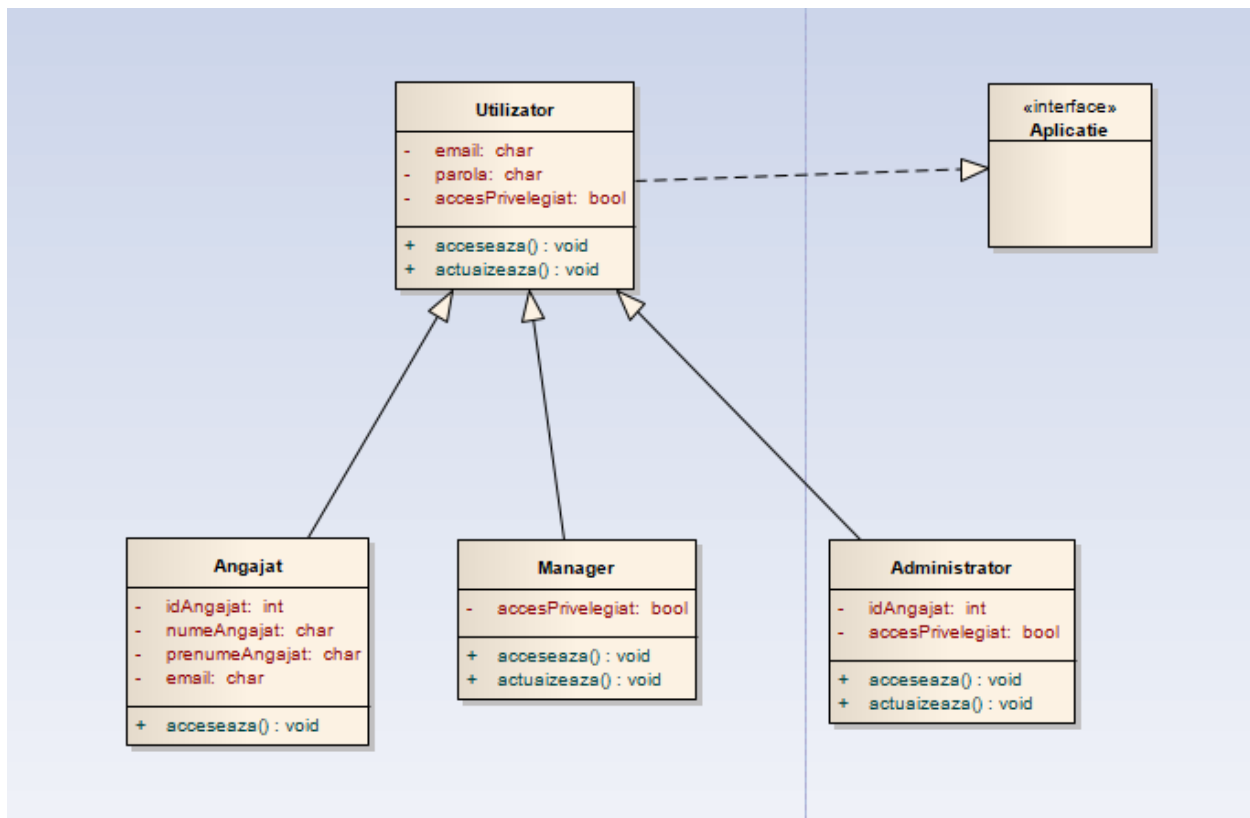


Figura 16 - Diagrama de clasa sistemul utilizatori

## 2.2.2 RELATIILE DE DEPENDENȚĂ ÎNTRE COMPONENTELE SISTEMULUI

În figura de mai jos urmeaza diagrama de componență a sistemului creat.

”Comanda” este componenta principală și este compusă din 3 sub-componente:

- țări;
- produs;
- șofer.

Sub-componenta ”Țări” se ocupă cu datele legate de țara origine și țara destinație a comenzii, sub-componenta ”Produs” se ocupă cu datele complete a produsului ce este livrat iar sub-componenta ”Șofer” se ocupă cu datele private a șoferului, cum ar fi: numele, telefon, e-mail, marca și id-ul camionului ce este manevrat.

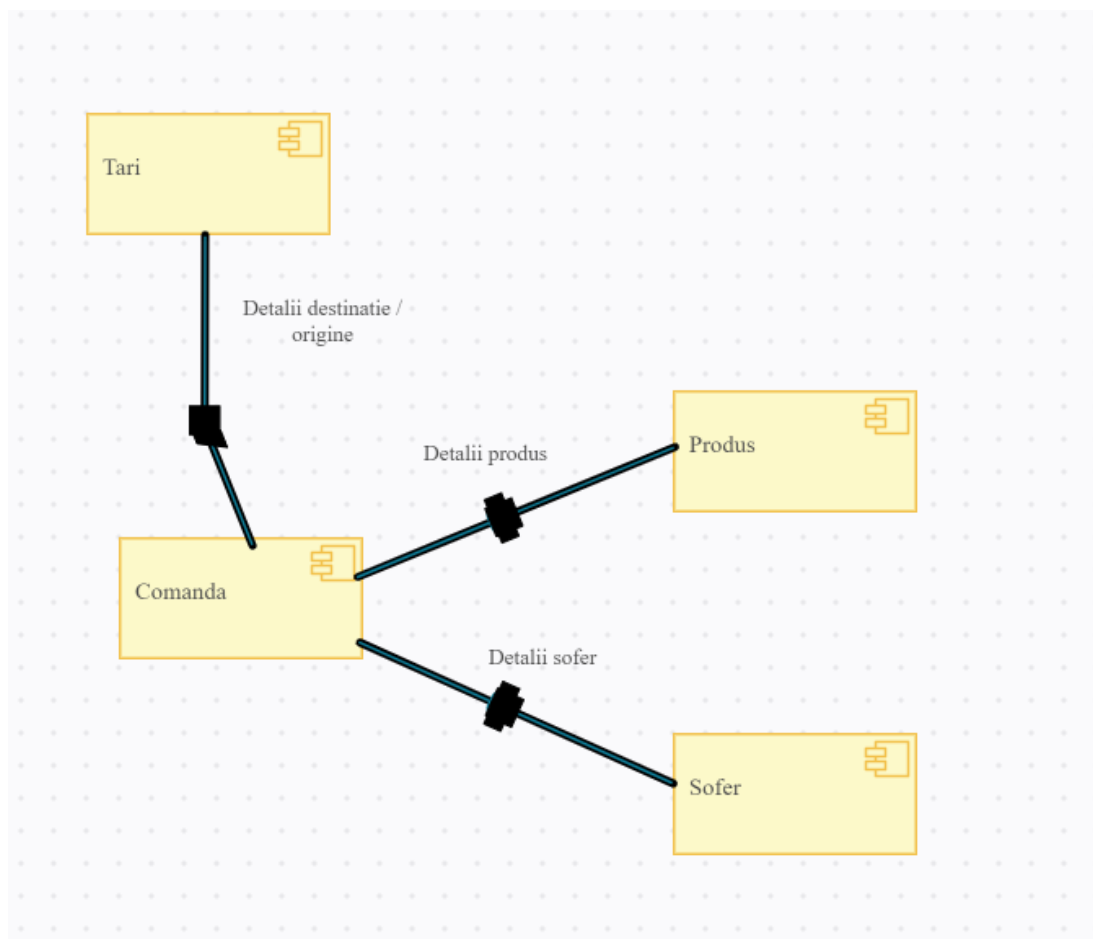


Figura 17 – Diagrama de componență a sistemului informațional

### 2.2.3 MODELAREA ECHIPAMENTELOR MEDIULUI de IMPLEMENTARE

În figura de mai jos este reprezentată diagrama deployment pentru sistemul informațional creat.

Diagrama este compusă din actorul "Utilizator", nodurile "Platforma Desktop" & "Serviciu Web" care reprezintă componentele hardware/software, componentele "Application Server" & "Baza de date", care la rândul lor conțin artefacte.

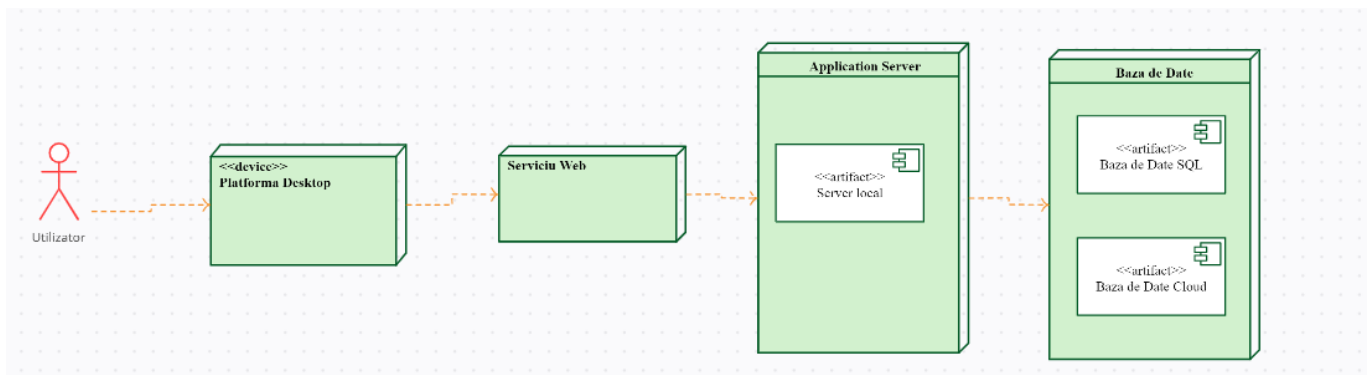


Figura 18 – Diagrama Deployment pentru sistemul informațional

### 3 REALIZAREA SISTEMULUI

SQL Server Management Studio este un instrument cuprinzător pentru gestionarea bazelor de date SQL Server și oferă o serie de funcționalități pentru a ajuta dezvoltatorii, administratorii de baze de date și analiștii de afaceri. Este o interfață grafică cu utilizatorul care permite utilizatorilor să gestioneze bazele de date SQL Server printr-un set de instrumente integrate care oferă administratorilor de baze de date o interfață ușor de utilizat pentru configurarea, gestionarea și monitorizarea instanțelor SQL Server. Instrumentul este conceput pentru a ajuta utilizatorii să creeze și să gestioneze cu ușurință obiecte de bază de date, cum ar fi tabele, vizualizări, indecși și proceduri stocate. Una dintre caracteristicile cheie ale SQL Server Management Studio este Editorul de interogări, care permite utilizatorilor să scrie și să execute interogări SQL complexe în bazele lor de date. Editorul include o serie de funcții utile, cum ar fi instrumente de evidențiere a sintaxelor, completare automată și depanare, pentru a ajuta utilizatorii să scrie interogări eficiente și fără erori.

În plus, SSMS oferă mai multe instrumente pentru administrarea bazei de date, cum ar fi funcționalitatea de backup și restaurare, managementul securității și monitorizarea performanței. Instrumentul permite utilizatorilor să vizualizeze valorile de performanță și să identifice potențiale blocaje de performanță, facilitând optimizarea performanței bazei de date.

SSMS oferă, de asemenea, suport pentru Integration Services, Analysis Services și Reporting Services, permițând utilizatorilor să creeze, să gestioneze și să implementeze pachete, modele și rapoarte. În cele din urmă, SSMS este foarte personalizabil, permițând utilizatorilor să configureze instrumentul pentru a se potrivi nevoilor lor specifice, cu opțiuni de personalizare a interfeței cu utilizatorul, comenzilor rapide de la tastatură și setărilor.

Microsoft Visual Studio este un mediu de dezvoltare integrat (IDE) de la Microsoft. Este folosit pentru a dezvolta programe pentru computer, precum și site-uri web, aplicații web, servicii web și aplicații mobile. Visual Studio folosește platforme de dezvoltare software Microsoft precum Windows API, Windows Forms, Windows Presentation Foundation, Windows Store și Microsoft Silverlight. Poate produce atât cod nativ, cât și cod gestionat. Windows Forms este o tehnologie UI pentru .NET, un set de biblioteci gestionate care simplifică sarcinile obișnuite ale aplicațiilor, cum ar fi citirea și scrierea în sistemul de fișiere. Când utilizați un mediu de dezvoltare precum Visual Studio, puteți crea aplicații inteligente pentru clienți Windows Forms care afișează informații, solicită intrări de la utilizatori și comunică cu



computerele la distanță printr-o rețea. În Windows Forms, un formular este o suprafață vizuală pe care afișați informații utilizatorului. În mod obișnuit, creați aplicații Windows Forms adăugând controale la formulare și dezvoltând răspunsuri la acțiunile utilizatorului, cum ar fi clicurile mouse-ului sau apăsarea tastelor. Un control este un element de interfață discret care afișează date sau acceptă introducerea datelor.

### 3.1 DESCRIEREA LA NIVEL DE COD PE MODULE

Aplicația data îndeplinește 3 funcționaluri:

- inserarea datelor;
- modificarea datelor;
- ștergerea datelor.

Inserarea datelor:

```
string insertsql;
insertsql = "insert into Excursii
(IDExcursie, IDTaraDestinatie, IDTaraStart, IDTruck, IDAngajat) values " +
"('"; insertsql += textBox2.Text + "',' + textBox3.Text + "',' +
textBox4.Text + "',' + textBox6.Text + "',' + textBox7.Text + "')";
SqlCommand cmd = new SqlCommand(insertsql, co);
cmd.ExecuteNonQuery();
MessageBox.Show("Inregistrare reusita !");
this.Close();
```

Este creată variabila "insertsql" de tip string, care are ca valoarea interogarea SQL "insert", care introduce datele inserate din câmpurile de text în tabelul "Excursii", după care comanda este efectuată în SQL și aplicația primește răspuns înapoi, în cazul în care interogarea a fost efectuată cu succes, se afișează fereastra de mesaj cu mesajul "Inregistrare reusita!", după care fereastra dată este închisă și revenim la fereastra principală.

Modificarea datelor:

```
String selectnume;
selectnume = "select * from Angajati where IDAngajat = '" + textBox1.Text
+ "'";
SqlCommand cmd = new SqlCommand(selectnume, f.co);
SqlDataAdapter adapter = new SqlDataAdapter(cmd);
DataSet ds = new DataSet();
```

```

        adapter.Fill(ds, "Angajati");
        label11.Text = "IDAngajat"; label12.Text = "IDTruck"; label13.Text =
"Nume";
        label17.Text = "Prenume"; label14.Text = "Telefon";

        foreach (DataRow r in ds.Tables["Angajati"].Rows)
        {
            label11.Text = label11.Text + "\n" + r["IDAngajat"] + "\n";
            label12.Text = label12.Text + "\n" + r["IDTruck"] + "\n";
            label13.Text = label13.Text + "\n" + r["Nume"] + "\n";
            label17.Text = label17.Text + "\n" + r["Prenume"] + "\n";
            label14.Text = label14.Text + "\n" + r["Telefon"] + "\n"; }
String inregsql;
        inregsql = "update Angajati set IDTruck='" + textBox3.Text + "',Nume='" +
textBox4.Text + "',Prenume='" + textBox6.Text +
        "',Telefon='" +textBox7.Text+ "' where IDAngajat='" + textBox1.Text
+ "'";
        SqlCommand cmd = new SqlCommand(inregsql, f.co);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Modificare reusita !");
        this.Close();}

```

Este creată variabila ”selectnume” de tip string, care are ca valoare interogarea SQL ”select”, care selectează toate datele din tabelul ”Angajati” după un ”IDAngajat” anumit, după care datele curente sunt afișate ca text pe fereastra dată, după care introducem datele noi în câmpurile de text, iar după apăsarea butonului ”Modify” se afiseaza fereastra de mesaj cu mesajul ”Modificare reusita!”, după care fereastra dată este închisă și revenim la fereastra principală.

Ștergerea datelor:

```

string deletesql;
        deletesql = "delete from Excursii where IDExcursie='";
        deletesql += textBox1.Text + "'";
        SqlCommand cmd = new SqlCommand(deletesql, f.co);
        cmd.ExecuteNonQuery();

        MessageBox.Show("Stergere reusita !");
        this.Close();

```

Este creată variabila ”deletesql” de tip string, care are ca valoare interogarea SQL ”delete” care șterge toate datele din tabelul ”Excursii” după un ”IDExcursie” anumit, după indicarea ID-ului și confirmarea ștergerii, se afiseaza fereastra de mesaj cu mesajul ”Ștergere reusita!”, după care fereastra dată este închisă și revenim la fereastra principală.

## 3.2 TESTAREA SISTEMULUI

Testarea a fost făcută manual, pentru a avea rezultate cât mai precise.

Luând în considerare că majoritatea operațiilor depind de baza de date, am efectuat mai multe teste, soluția acestora fiind aducând baza de date la forma normala III.

În urma efectuării acestor teste, cum ar fi:

- introducerea spațiilor libere în câmpurile de text;
- introducerea ID-urilor duble;
- introducerea caracterelor alfa în câmpurile de text cu tipul de date "int".

La fel a fost folosită biblioteca Xunit pentru crearea testelor automate, unde am testat conexiunea bazei de date. În cazul în care conexiunea este stabilită cu succes iar numărul de coloane din tabelul "Angajati" este egal cu 3, atunci testul va trece cu succes, iar în caz contrar testul va eșua.

```
using System.Data.SqlClient;
using Xunit;

public class ConnectionTest
{
    [Fact]
    public void TestSqlConnection()
    {
        public SqlConnection co = new SqlConnection(@"Data Source=.; Initial Catalog
=Transport; Integrated Security=true;");
        string sql = "SELECT COUNT(*) FROM Angajati";

        using (SqlConnection connection = new SqlConnection(co))
        {
            SqlCommand command = new SqlCommand(sql, connection);
            connection.Open();
            int count = (int)command.ExecuteScalar();
            Assert.Equal(3, count);
        }
    }
}
```

## CONCLUZII

În acest raport am studiat procesul de realizare a unei aplicații ce are ca scop ușurarea procesului de gestionare a datelor în sfera logisticii camioanelor.

Procesul de lucru a fost repartizat după capitole, fiecare din acesta având scopul său, având subcapitole ce descrie în mod desfășurat a procesului de lucru.

Primul capitol descrie importanța temei, a fost comparat sistemul efectuat cu alte trei sisteme deja existente, unde am comparat funcționalul principal al acestora, dar și prin ce se diferă sistemul efectuat și prin ce este mai bun acesta.

Sistemele existente cu care a fost comparat sistemul efectuat, au permis găsirea lucrurilor bune, dar și rele ale acestora, pentru a efectua un sistem cât mai bun și plăcut utilizatorilor. La fel am identificat scopul sistemului dat, cerințele funcționale dar și cele nefuncționale.

Scopul principal al acestui sistem a fost crearea unei aplicații cât mai ușor de înțeles și de folosit, care ușurează procesul de inserare, modificare dar și ștergerea datelor existente.

În urma definirii scopului, cercetării subiectului dat, am aflat cu adevărat cât de important este procesul de gestionare a datelor, dar și însăși importanța logisticii.

Scopul capitolului nr. 2 a fost modelarea și proiectarea sistemului, care are ca scop descrierea structurii și comportamentului sistemului efectuat prin intermediul diagramelor UML.

În capitolul nr. 3 a fost descrisă realizarea sistemului, unde s-a explicat limbajul folosit, mediul de dezvoltare și descrierea funcționalului sistemului, pe module:

- inserarea datelor;
- modificarea datelor;
- ștergerea datelor.

## **BIBLIOGRAFIE**

1. HmnWiki – Microsoft SQL Server [Resursă electronică] – Regim de acces: [https://hmn.wiki/ro/Microsoft\\_SQL\\_Server](https://hmn.wiki/ro/Microsoft_SQL_Server)
2. Wblog – Excel [Resursă electronică] – Regim de acces: [https://wblog.wiki/ro/Excel\\_spreadsheet](https://wblog.wiki/ro/Excel_spreadsheet)
3. Lwvcorg – Ce este ninox database [Resursă electronică] – Regim de acces: <https://lwvworc.org/ro/what-is-ninox-database>
4. Brainly.ro - [Resursă electronică] – Regim de acces: <https://brainly.ro/tema/8248009>