

Universitatea Politehnica București

Facultatea de Automatică și Calculatoare



Aplicație pentru evidența medicamentelor într-o farmacie

Baze de date

Student: Tomescu Ilie Alexandru

Grupa: 332AA

Cuprins

1. Descriere	3
2. Tehnologii.....	3
3. Arhitectura bază de date	4
Tabele.....	4
Relații implementate	7
4. Definire cerințe	8
Bibliografie	9

1. Descriere

Prin acest proiect voi proiecta o bază de date simplificată care arată cum se poate face gestiunea unei farmacii, cu mai multe puncte de lucru, atât pentru medicamente, cât și pentru angajați. Astfel voi crea o aplicație desktop cu interfață grafică din care se vor putea face anumite interogări, atât simple cât și complexe.

Baza de date va stoca date despre medicamente, furnizorii de la care provin acestea, angajați și punctele de lucru la care se desfășoară activitatea.

Aplicația o să fie cât mai intuitivă de folosit pentru ca orice tip de utilizator să o poată înțelege funcționalitățile și astfel să poată obține informații relevante cu privire la evidența medicamentelor și a angajaților în funcție de punctul de lucru la care se află fiecare. Utilizatorii sunt împărțiți în două categorii (user și admin) și în funcție de aceasta au acces diferit la date.

Această aplicație desktop va fi realizată cu ajutorul limbajului de programare Java așa cum am prezentat în [Tehnologii](#).

2. Tehnologii

Ca și tehnologii pentru această aplicație voi folosi :

- Java
 - Voi realiza conexiunea la baza de date Microsoft SQL prin JDBC¹ (Java Database Connectivity) și voi folosi funcțiile din pachetul java.sql² pentru a realiza interogările
 - Voi folosi Java Swing pentru a crea interfața grafică
- Maven³
 - Pentru a folosi anumite biblioteci Java sau pluginuri precum Lombok⁴ sau Microsoft SQL Server
- IntelliJ IDEA⁵
 - Voi folosi acest IDE pentru că sunt familiar cu el și mă ajută să avansez mai repede în acest proiect
- Microsoft SQL Server⁶
 - Voi folosi Microsoft SQL pentru managementul și crearea bazei de date.

¹ (Microsoft JDBC Driver for SQL Server, 2019)

² (Package java.sql, fără an)

³ (Apache Maven Project, fără an)

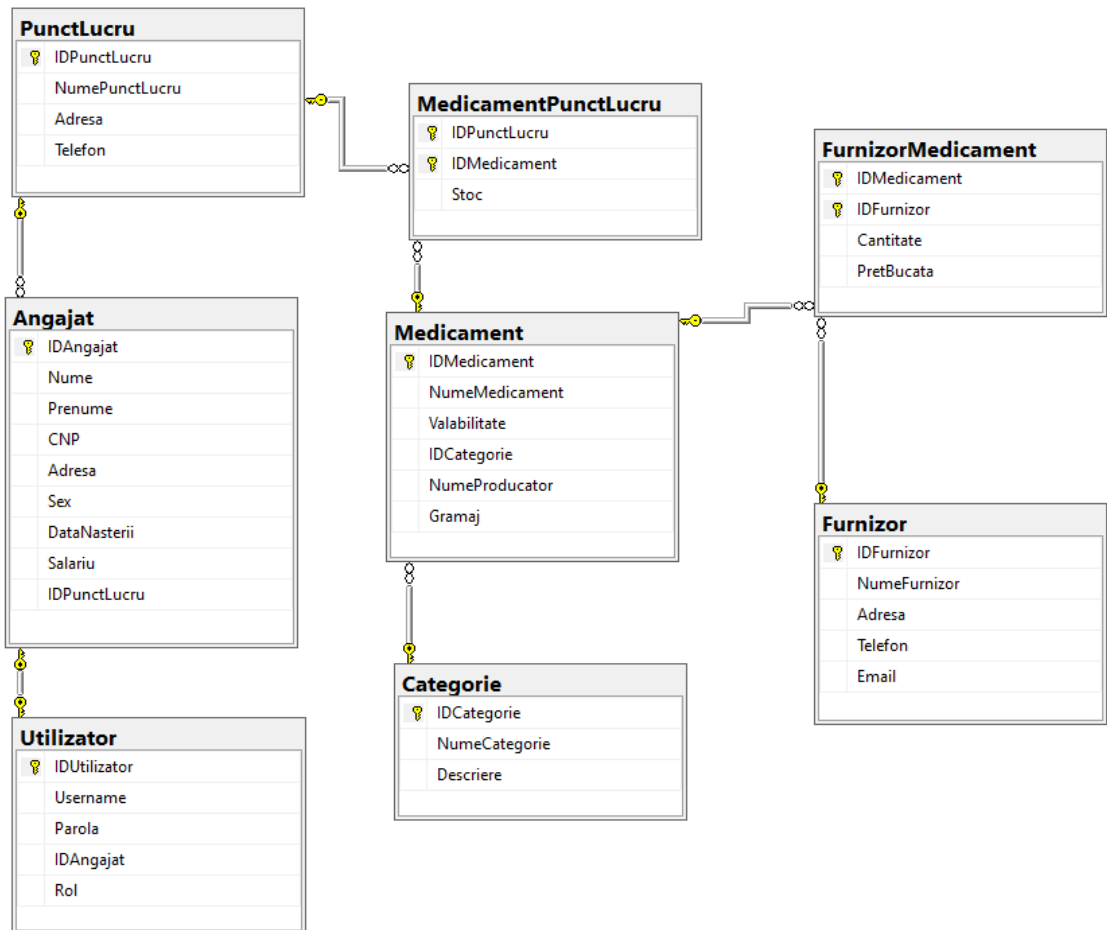
⁴ (Michael Kimberlin, 2010)

⁵ (IntelliJ IDEA, fără an)

⁶ (SQL Server, fără an)

3. Arhitectura bază de date


Am creat baza de date în Microsoft SQL Server și are următoarea structură:



Tabele


1. Medicament
2. Furnizor
3. Categorie
4. Angajat
5. PunctLucru
6. FurnizorMedicament (tabel de legătură între Furnizor și Medicament)
7. MedicamentPunctLucru (tabel de legătură între PunctLucru și Medicament)
8. Utilizator

1. Medicament

	Column Name	Data Type	Allow Nulls
	IDMedicament	int	<input type="checkbox"/>
	NumeMedicament	nvarchar(50)	<input type="checkbox"/>
	Valabilitate	datetime	<input type="checkbox"/>
	IDCategorie	int	<input checked="" type="checkbox"/>
	NumeProducator	nvarchar(50)	<input type="checkbox"/>
	Gramaj	int	<input type="checkbox"/>


- IDMedicament – primary key, auto increment
- IDCategorie – foreign key

2. Furnizor

	Column Name	Data Type	Allow Nulls
	IDFurnizor	int	<input type="checkbox"/>
	NumeFurnizor	nvarchar(50)	<input type="checkbox"/>
	Adresa	nvarchar(50)	<input type="checkbox"/>
	Telefon	char(10)	<input type="checkbox"/>
	Email	nvarchar(50)	<input type="checkbox"/>


- IDFurnizor – primary key, auto increment

3. Categorie

	Column Name	Data Type	Allow Nulls
	IDCategorie	int	<input type="checkbox"/>
	NumeCategorie	nvarchar(50)	<input type="checkbox"/>
	Descriere	nvarchar(50)	<input type="checkbox"/>


- IDCategorie – primary key, auto increment

4. *Angajat*

	Column Name	Data Type	Allow Nulls
	IDAngajat	int	<input type="checkbox"/>
	Nume	nvarchar(50)	<input type="checkbox"/>
	Prenume	nvarchar(50)	<input type="checkbox"/>
	CNP	char(13)	<input type="checkbox"/>
	Adresa	nvarchar(50)	<input type="checkbox"/>
	Sex	char(1)	<input type="checkbox"/>
	DataNasterii	nvarchar(50)	<input type="checkbox"/>
	Salariu	int	<input type="checkbox"/>
	IDPunctLucru	int	<input checked="" type="checkbox"/>



- IDAngajat - primary key, auto increment
- IDPunctLucru - foreign key
- Constrângere - ([Sex]='F' OR [Sex]='M')
- CNP – unique key

5. *PunctLucru*

	Column Name	Data Type	Allow Nulls
	IDPunctLucru	int	<input type="checkbox"/>
	NumePunctLucru	nvarchar(50)	<input type="checkbox"/>
	Adresa	nvarchar(50)	<input type="checkbox"/>
	Telefon	char(10)	<input type="checkbox"/>



- IDPunctLucru - primary key, auto increment

6. *FurnizorMedicament*

	Column Name	Data Type	Allow Nulls
	IDMedicament	int	<input type="checkbox"/>
	IDFurnizor	int	<input type="checkbox"/>
	Cantitate	int	<input type="checkbox"/>
	PretBucata	float	<input type="checkbox"/>


- IDMedicament – primary key / foreign key
- IDFurnizor - primary key / foreign key

7. *MedicamentPunctLucru*

	Column Name	Data Type	Allow Nulls
	IDPunctLucru	int	<input type="checkbox"/>
	IDMedicament	int	<input type="checkbox"/>
	Stoc	int	<input type="checkbox"/>

- IDMedicament – primary key / foreign key
- IDPunctLucru - primary key / foreign key

8. *Utilizator*

	Column Name	Data Type	Allow Nulls
	IDUtilizator	int	<input type="checkbox"/>
	Username	nvarchar(50)	<input type="checkbox"/>
	Parola	nvarchar(50)	<input type="checkbox"/>
	IDAngajat	int	<input checked="" type="checkbox"/>
	Rol	char(5)	<input type="checkbox"/>

- IDUtilizator – primary key, auto increment
- IDAngajat - foreign key, unique key

Relații implementate

- Categorie – Medicament (One to Many - 1:N)
- PunctLucru – Angajat (One to Many - 1:N)
- Medicament – Furnizor (Many to Many - N:N)
- Medicament – PunctLucru (Many to Many - N:N)
- Angajat – Utilizator (One to one – 1:1)

4. Definire cerințe

Intrarea în această aplicație se face pe bază de Log in, unde fiecare utilizator va avea un username/email (unic) și o parolă, care vor fi verificate în baza dacă există și dacă se potrivesc.

După ce logarea a fost realizată cu succes, utilizatorul va intra în meniul principal, din această pagină acesta poate accesa două, poate realiza ori operații de insert, update sau delete ori poate să treacă la parte de interogări, unde va avea două categorii, interogări simple și interogări complexe, iar în funcție de opțiunea selectată de acesta va primi informații specifice. În funcție de rolul pe care îl are utilizatorul acesta va avea următorul acces:

- User – poate accesa doar date referitoare la medicamente și furnizori
- Admin – are acces la toată baza de date și poate adăuga angajați care vor avea asociat un cont

Această aplicație ar putea ajuta în următoarele aspecte:

- Vizualizarea stocurilor la medicamente la fiecare punct de lucru pentru a determina ce medicamente trebuie aprovizionate și unde trebuie
- Compararea prețurilor medicamentelor oferite de mai mulți furnizori și astfel alegerea celei mai bune oferte
- Managementul angajaților în funcție de punctul de lucru
- O vizualizare eficientă cu privire la costurile pentru fiecare punct de lucru, cost medicamente + cost salarii

Bibliografie

(fără an). Preluat de pe Apache Maven Project: <https://maven.apache.org/index.html>

(fără an). Preluat de pe IntelliJ IDEA: <https://www.jetbrains.com/idea/>

Michael Kimberlin, O. S. (2010, January). *REDUCING BOILERPLATE CODE WITH PROJECT LOMBOK*.

Preluat de pe Object Computing:

<https://objectcomputing.com/resources/publications/sett/january-2010-reducing-boilerplate-code-with-project-lombok>

Microsoft JDBC Driver for SQL Server. (2019, 12 08). Preluat de pe Microsoft Docs:

<https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>

Package java.sql. (fără an). Preluat de pe OracleDocs:

<https://docs.oracle.com/javase/8/docs/api/java/sql/package-summary.html>

Package javax.swing. (fără an). Preluat de pe OracleDocs:

<https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>

SQL Server. (fără an). Preluat de pe Microsoft Data Platform: <https://www.microsoft.com/en-us/sql-server>