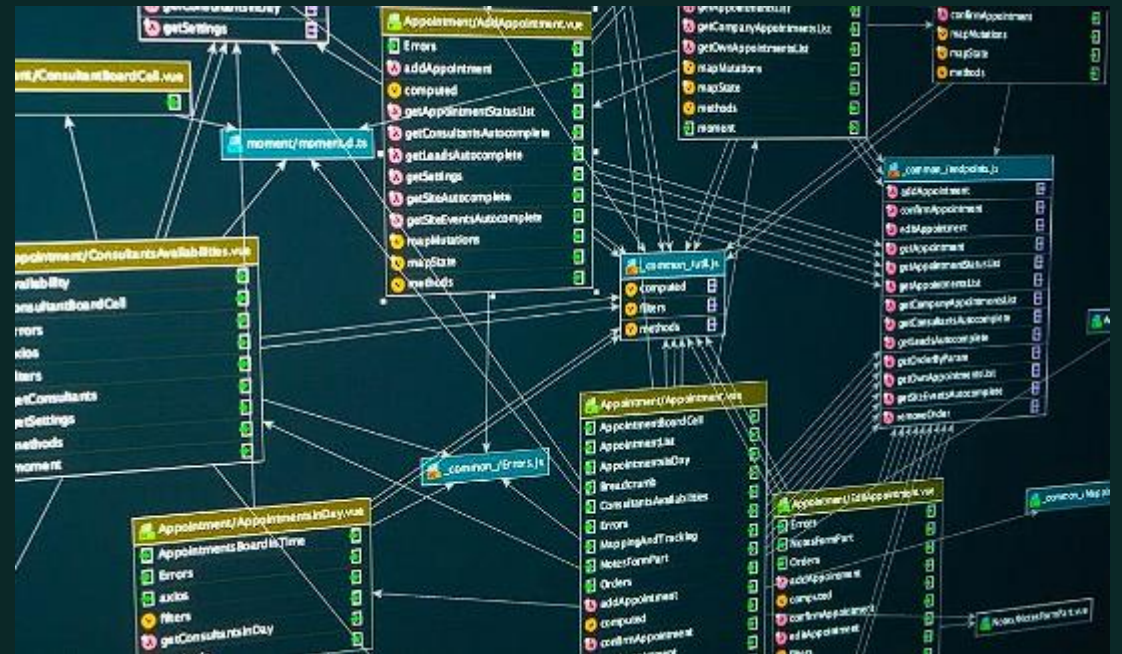




Project Final



Filip Odette-Alexandra

Cuprins

I. Partea Teoretica

II. Partea Practica

July 13, 2024



I. Partea Teoretica

THEORY

Timeline

- **1. Explicați pe scurt ce sunt cerințele de business, la ce ne folosesc și cine le creează**

- Cerințele de business reprezintă nevoile și așteptările unei organizații în ceea ce privește un produs sau serviciu. Aceste cerințe sunt esențiale pentru a defini scopul și obiectivele unui proiect.

- **2. Explicați diferența între un test condition și test case**

- **Test Condition:**

- O afirmație despre un aspect al sistemului care poate fi verificat prin testare.
- Exemple: o funcționalitate, un comportament al sistemului, o condiție de performanță.

- **Test Case:**

- Un set de condiții, inclusiv date de intrare, pași de executat și rezultate așteptate, folosite pentru a verifica un test condition.
- Conține informații detaliate pentru a verifica dacă o anumită funcționalitate a sistemului funcționează conform așteptărilor.

3. Enumerați și explicați pe scurt etapele procesului de testare



- Planificarea Testării:
Definirea strategiei de testare, obiectivelor și resurselor necesare.
- Analiza și Design-ul Testării:
Identificarea cerințelor de testare și crearea specificațiilor de testare.
- Implementarea și Executarea Testării:
Crearea și executarea testelor conform planului și design-ului.
- Evaluarea Criteriilor de Ieșire și Raportarea:
Evaluarea rezultatelor testelor pentru a determina dacă criteriile de ieșire sunt îndeplinite și raportarea defectelor.
- Activități de Închidere a Testării:
Concluzionarea testării și documentarea lecțiilor învățate. Se generează un raport de închidere a testării și se identifică dacă există riscurile de produs și se raportează către client.

- **4. Explicați diferența între retesting și regression testing**

- **Retesting:** Procesul de testare a unui defect reparat pentru a verifica dacă problema a fost rezolvată. Se concentrează pe defectul specific care a fost remediat.
- **Regression Testing:** Testarea întregului sistem sau a unei părți a acestuia pentru a verifica dacă modificările recente nu au introdus noi defecte. Asigură că funcționalitățile existente funcționează corect după schimbări.

- **5. Explicați diferența între functional testing și non-functional testing**

- **Functional Testing:** Verifică dacă funcționalitățile sistemului funcționează conform specificațiilor. Exemple: testare de unități, testare de integrare, testare de sistem.
- **Non-Functional Testing:** Verifică aspecte care nu sunt legate de funcționalitățile specifice, cum ar fi performanța, securitatea, ușurința de utilizare. Exemple: testare de performanță, testare de securitate, testare de utilizabilitate

- **6. Explicați diferența între blackbox testing și whitebox testing**

- **Blackbox Testing:** Testare fără cunoașterea internă a codului sursă sau a structurii interne a aplicației. Se concentrează pe input-uri și output-uri.
- **Whitebox Testing:** Testare cu cunoașterea internă a codului sursă și a structurii interne a aplicației. Se concentrează pe logica internă a aplicației.

- **7. Enumerați tehnicile de testare și grupați-le în funcție de categorie (blackbox, whitebox, experience-based)**

Blackbox testing	Whitebox testing	Experienced-based Testing
<ul style="list-style-type: none">• Equivalence Partitioning• Boundary Value Analysis• Decision Table Testing• State Transition Testing• Use Case Testing	<ul style="list-style-type: none">• Statement Coverage• Branch Coverage• Path Coverage• Loop Testing	<ul style="list-style-type: none">• Exploratory Testing• Error Guessing• Checklist-Based Testing

- **8. Explicați diferența între verification și validation**

- **Verification:** Procesul de evaluare a produselor de lucru intermediare pentru a asigura că acestea respectă specificațiile. Răspunde la întrebarea "Construim produsul corect?"
- **Validation:** Procesul de evaluare a produsului final pentru a asigura că acesta satisface cerințele utilizatorului. Răspunde la întrebarea "Am construit produsul corect?"

- **9. Explicați diferența între positive testing și negative testing și dați câte un exemplu din fiecare**

- **Positive Testing:** Verifică dacă aplicația funcționează așa cum ar trebui în condiții normale.
- Exemplu: Introducerea unui nume de utilizator și a unei parole corecte pentru a te autentifica.
- **Negative Testing:** Verifică cum reacționează aplicația la date sau condiții incorecte.
- Exemplu: Introducerea unei parole greșite pentru a te asigura că sistemul nu permite accesul.

- **10. Enumerați și explicați pe scurt nivelurile de testare**

- 1. Unit Testing:** Testarea unităților individuale de cod.
- 2. Integration Testing:** Testarea interacțiunilor dintre module sau componente.
- 3. System Testing:** Testarea sistemului complet integrat pentru a verifica dacă îndeplinește cerințele specificate.
- 4. Acceptance Testing:** Testarea realizată de utilizatorii finali pentru a verifica dacă sistemul satisface nevoile și cerințele lor.

II. Partea Practica



PARTEA II – ASPECTE PRACTICE

- INSTRUCTIUNI DDL: Crearea a 3 tabele cu attributele specifice pentru fiecare si legăturile dintre acestea.

```
-- Creare baza de date:  
Create Database FarmaVetHelp;
```

```
-- Creare tabela Owners (OwnerID este primary key pentru aceasta tabela.  
-- Aceasta este foreign Key in Tabela Appointments.)  
CREATE TABLE Owners (  
    OwnerID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    PhoneNumber VARCHAR(20),  
    Email VARCHAR(100)  
);
```

```
-- Creare Tabela Pets: (OwnerID este Foreign key in tabela pets care face  
-- legatura cu primary key OwnerID din tabela owners. PetID este primary key  
-- pentru aceasta tabela. Aceasta este foreign Key in Tabela Appointments.)  
CREATE TABLE Pets (  
    PetID INT PRIMARY KEY,  
    OwnerID INT,  
    PetName VARCHAR(50),  
    Species VARCHAR(50),  
    Breed VARCHAR(50),  
    FOREIGN KEY (OwnerID) REFERENCES Owners(OwnerID)  
);
```

```
-- Creare tabela Appointments (aceasta tabela sa creaza ultima, intrucat  
-- are 2 foreign keys, prin care se face relationarea la tabelele Owners si Pets)  
CREATE TABLE Appointments (  
    AppointmentID INT PRIMARY KEY,  
    OwnerID INT,  
    PetID INT,  
    AppointmentDate DATE,  
    Reason VARCHAR(100),  
    FOREIGN KEY (OwnerID) REFERENCES Owners(OwnerID),  
    FOREIGN KEY (PetID) REFERENCES Pets(PetID)  
);
```

○ INSTRUCTIUNI DML

1. **INSERT:** Populare Valori in Tabele. Am generat 20 de entries Random pentru fiecare tabela in parte dupa cum urmeaza. Ordinea rularii lor nu este importanta.

```
INSERT INTO Pets (PetID, OwnerID, PetName, Species, Breed) VALUES
(1, 100, 'Rex', 'Câine', 'Ciobănesc German'),
(2, 101, 'Luna', 'Câine', 'Labrador'),
(3, 100, 'Miau', 'Pisică', 'Europeană Comună'),
(4, 102, 'Pufi', 'Câine', 'Bichon Maltez'),
(5, 103, 'Whiskers', 'Pisică', 'Persană'),
(6, 104, 'Max', 'Câine', 'Golden Retriever'),
(7, 105, 'Tom', 'Pisică', 'Siamieză'),
(8, 106, 'Bella', 'Câine', 'Beagle'),
(9, 107, 'Felix', 'Pisică', 'Ragdoll'),
(10, 108, 'Rocky', 'Câine', 'Rottweiler'),
(11, 109, 'Mimi', 'Pisică', 'Birmaneză'),
(12, 110, 'Coco', 'Câine', 'Cocker Spaniel'),
(13, 111, 'Tiger', 'Pisică', 'Bengaleză'),
(14, 112, 'Lucky', 'Câine', 'Ciobănesc Românesc Mioritic'),
(15, 113, 'Princess', 'Pisică', 'Angoră Turcească'),
(16, 114, 'Buddy', 'Câine', 'Bulldog Francez'),
(17, 115, 'Snowball', 'Pisică', 'Siberiană'),
(18, 116, 'Rosie', 'Câine', 'Shih Tzu'),
(19, 117, 'Oreo', 'Pisică', 'British Shorthair'),
(20, 118, 'Bailey', 'Câine', 'Boxer');
```

```
INSERT INTO Owners (OwnerID, FirstName, LastName, PhoneNumber, Email) VALUES
(100, 'Ion', 'Popescu', '0711-234-567', 'ion.popescu@example.com'),
(101, 'Maria', 'Ionescu', '0722-345-678', 'maria.ionescu@example.com'),
(102, 'Andrei', 'Popa', '0733-456-789', 'andrei.popa@example.com'),
(103, 'Elena', 'Dumitrescu', '0744-567-890', 'elena.dumitrescu@example.com'),
(104, 'Mihai', 'Constantin', '0755-678-901', 'mihai.constantin@example.com'),
(105, 'Ana', 'Radu', '0766-789-012', 'ana.radu@example.com'),
(106, 'Cristian', 'Gheorghiu', '0777-890-123', 'cristian.gheorghiu@example.com'),
(107, 'Laura', 'Stoica', '0788-901-234', 'laura.stoica@example.com'),
(108, 'Adrian', 'Munteanu', '0799-012-345', 'adrian.munteanu@example.com'),
(109, 'Simona', 'Tudor', '0710-123-456', 'simona.tudor@example.com'),
(110, 'Alexandru', 'Moldovan', '0721-234-567', 'alexandru.moldovan@example.com'),
(111, 'Andreea', 'Voicu', '0732-345-678', 'andreea.voicu@example.com'),
(112, 'Gabriel', 'Dumitru', '0743-456-789', 'gabriel.dumitru@example.com'),
(113, 'Mihaela', 'Gheorghe', '0754-567-890', 'mihaela.gheorghe@example.com'),
(114, 'Daniel', 'Stanciu', '0765-678-901', 'daniel.stanciu@example.com'),
(115, 'Raluca', 'Stefan', '0776-789-012', 'raluca.stefan@example.com'),
(116, 'Sorin', 'Marin', '0787-890-123', 'sorin.marin@example.com'),
(117, 'Irina', 'Dinu', '0798-901-234', 'irina.dinu@example.com'),
(118, 'Gabriela', 'Dobre', '0711-012-345', 'gabriela.dobre@example.com'),
(119, 'Stefan', 'Gavrilă', '0722-123-456', 'stefan.gavrilă@example.com');
```

```
INSERT INTO Appointments (AppointmentID, OwnerID, PetID, AppointmentDate, Reason) VALUES
(1, 100, 1, '2024-07-15', 'Vaccinare'),
(2, 101, 2, '2024-07-16', 'Consult anual'),
(3, 100, 3, '2024-07-17', 'Curățare dentară'),
(4, 102, 4, '2024-07-18', 'Tratament pentru paraziți'),
(5, 103, 5, '2024-07-19', 'Vaccinare'),
(6, 104, 6, '2024-07-20', 'Consult de rutină'),
(7, 105, 7, '2024-07-21', 'Control de sănătate'),
(8, 106, 8, '2024-07-22', 'Tratament pentru infecție'),
(9, 107, 9, '2024-07-23', 'Vaccinare'),
(10, 108, 10, '2024-07-24', 'Consult de urgență'),
(11, 109, 11, '2024-07-25', 'Examinare fizică'),
(12, 110, 12, '2024-07-26', 'Vaccinare'),
(13, 111, 13, '2024-07-27', 'Consult anual'),
(14, 112, 14, '2024-07-28', 'Tratament pentru alergii'),
(15, 113, 15, '2024-07-29', 'Vaccinare'),
(16, 114, 16, '2024-07-30', 'Consult de rutină'),
(17, 115, 17, '2024-07-31', 'Control de sănătate'),
(18, 116, 18, '2024-08-01', 'Tratament pentru răni'),
(19, 117, 19, '2024-08-02', 'Vaccinare'),
(20, 118, 20, '2024-08-03', 'Consult de urgență');
```

2. **UPDATE** – Modificarea valorilor dintr-o tabela existenta.

```
-- Actualizarea Prenumelui pentru userul cu Idul de owner 100 : Ion -> Mihai
update owners set FirstName = 'Mihai' where Ownerid = 100;
```

3. **DELETE** – Stergea unei inregistrari dintr-o tabela.

```
-- Stergea intrarii din tabela appointments, care are PetID = 1
delete from appointments where PetId = '1';
```

○ INSTRUCTIUNI DQL

1. **Select *** - selectarea tuturor datelor dintr-o tabela.

```
-- Returnarea tuturor datelor din tabela:
select * from owners;
select * from pets;
select * from appointments;
```



	Appointm	OwnerID	PetID	AppointmentDate	Reason
▶	2	101	2	2024-07-16	Consult anual
	3	100	3	2024-07-17	Curățare dentară
	4	102	4	2024-07-18	Tratament pentru paraziți
	5	103	5	2024-07-19	Vaccinare
	6	104	6	2024-07-20	Consult de rutină
	7	105	7	2024-07-21	Control de sănătate
	8	106	8	2024-07-22	Tratament pentru infecție
	9	107	9	2024-07-23	Vaccinare
	10	108	10	2024-07-24	Consult de urgență
	11	109	11	2024-07-25	Examinare fizică
	12	110	12	2024-07-26	Vaccinare
	13	111	13	2024-07-27	Consult anual
	14	112	14	2024-07-28	Tratament pentru alergii
	15	113	15	2024-07-29	Vaccinare
	16	114	16	2024-07-30	Consult de rutină
	17	115	17	2024-07-31	Control de sănătate
	18	116	18	2024-08-01	Tratament pentru răni
	19	117	19	2024-08-02	Vaccinare
	20	118	20	2024-08-03	Consult de urgență

2. Selectarea doar anumitor coloane.


```
-- Returnarea unor anumite coloane din tabela.
-- (am folosit functia limit pentru a afisa doar primele 5 rezultate din tabela):
select FirstName, LastName, Email from owners limit 5;
select PetName, Species, Breed from pets limit 5;
select OwnerID, AppointmentDate, Reason from appointments limit 5;
```



	PetName	Species	Breed
▶	Rex	Câine	Ciobănesc German
	Luna	Câine	Labrador
	Miau	Pisică	Europeană Comună
	Pufi	Câine	Bichon Maltez
	Whiskers	Pisică	Persană

3. WHERE - selectarea tuturor datelor dintr-o tabela.


```
-- Filtrare valori folosind WHERE
-- Returnarea datelor unde FirstName este Mihai;
select * from owners where FirstName = 'Mihai';
```




	OwnerID	FirstName	LastName	PhoneNumber	Email
▶	100	Mihai	Popescu	0711-234-567	ion.popescu@example.com
	104	Mihai	Constantin	0755-678-901	mihai.constantin@example.com
*	NULL	NULL	NULL	NULL	NULL

4. LIKE - selectarea tuturor datelor dintr-o tabela.

```
-- Filtrare valori folosind LIKE
-- Returnarea datelor unde PetName incepe cu litera M
select * from pets where PetName like 'M%';
-- Returnarea datelor unde PetName se termina cu litera M
select * from pets where PetName like '%A';
```




	PetID	OwnerID	PetName	Species	Breed
▶	3	100	Miau	Pisică	Europeană Comună
	6	104	Max	Câine	Golden Retriever
	11	109	Mimi	Pisică	Birmaneză
*	NULL	NULL	NULL	NULL	NULL



	PetID	OwnerID	PetName	Species	Breed
▶	2	101	Luna	Câine	Labrador
	8	106	Bella	Câine	Beagle
*	NULL	NULL	NULL	NULL	NULL

5. AND si OR.

```
-- Filtrare valori folosind AND si OR
-- Returnare datelor unde PetName incepe cu M si Species este Pisica.
select * from pets where PetName like 'M%' and Species = 'Pisica';
-- Returnare datelor unde PetName incepe cu M sau Species este Pisica.
select * from pets where PetName like 'M%' or Species = 'Pisica';
```



	PetID	OwnerID	PetName	Species	Breed
▶	3	100	Miau	Pisică	Europeană Comună
	11	109	Mimi	Pisică	Birmaneză
*	NULL	NULL	NULL	NULL	NULL



	PetID	OwnerID	PetName	Species	Breed
	3	100	Miau	Pisică	Europeană Comună
	5	103	Whiskers	Pisică	Persană
	6	104	Max	Câine	Golden Retriever
	7	105	Tom	Pisică	Siamieză
	9	107	Felix	Pisică	Ragdoll
	11	109	Mimi	Pisică	Birmaneză
	13	111	Tiger	Pisică	Bengaleză
	15	113	Princess	Pisică	Angoră Turcească
	17	115	Snowball	Pisică	Siberiană
	19	117	Oreo	Pisică	British Shorthair



6. ORDER BY - ordonarea rezultatelor cautate intr-o ordine dorita.

```
-- Filtrare valori folosind ORDER BY
-- Returnarea valorilo din col. PetName, Species, Breed
-- si ordonarea lor in ordine alfabetica (asc-> de la A la Z)
select PetName , Species, Breed from pets order by PetName asc;
```

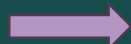


PetName	Species	Breed
Bailey	Câine	Boxer
Bella	Câine	Beagle
Buddy	Câine	Bulldog Francez
Coco	Câine	Cocker Spaniel
Felix	Pisică	Ragdoll
Lucky	Câine	Ciobănesc Românesc Mioritic
Luna	Câine	Labrador
Max	Câine	Golden Retriever
Miau	Pisică	Europeană Comună
Mimi	Pisică	Birmaneză
Oreo	Pisică	British Shorthair
Princess	Pisică	Angoră Turcească
Pufi	Câine	Bichon Maltez
Rex	Câine	Ciobănesc German
Rocky	Câine	Rottweiler
Rosie	Câine	Shih Tzu
Snowball	Pisică	Siberiană
Tiger	Pisică	Bengaleză
Tom	Pisică	Siamieză
Whiskers	Pisică	Persană



7. JOIN

```
-- Filtrare valori folosind JOIN
-- Returnarea tuturor valorilor din ambele tabele (owners si pets)
-- a si b sunt „nick-names” pentru a fi mai usor de inteles joinul
select * from owners a
join pets b on a.OwnerID = b.OwnerID limit 5;
```



OwnerID	FirstName	LastName	PhoneNumber	Email	PetID	OwnerID	PetName	Species	Breed
100	Mihai	Popescu	0711-234-567	ion.popescu@example.com	1	100	Rex	Câine	Ciobănesc German
101	Maria	Ionescu	0722-345-678	maria.ionescu@example.com	2	101	Luna	Câine	Labrador
100	Mihai	Popescu	0711-234-567	ion.popescu@example.com	3	100	Miau	Pisică	Europeană Comună
102	Andrei	Popa	0733-456-789	andrei.popa@example.com	4	102	Pufi	Câine	Bichon Maltez
103	Elena	Dumitrescu	0744-567-890	elena.dumitrescu@example.com	5	103	Whiskers	Pisică	Persană



```
-- Query folosind mai multe functii.
select a.FirstName,a.LastName,a.Email,b.PetName, b.Species, b.Breed
from owners a
join pets b on a.OwnerID = b.OwnerID
where b.PetName like 'M%' and b.Species = 'Pisica'
order by PetName asc;
```

FirstName	LastName	Email	PetName	Species	Breed
Mihai	Popescu	ion.popescu@example.com	Miau	Pisică	Europeană Comună
Simona	Tudor	simona.tudor@example.com	Mimi	Pisică	Birmaneză

Vă mulțumesc pentru atenție!

Github Link: <https://github.com/odette2490/DB-Testing>

