

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAŞI
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DOMENIUL: Calculatoare și tehnologia informației
SPECIALIZAREA: Tehnologia informației

Sistem de gestiune a atracțiilor turistice

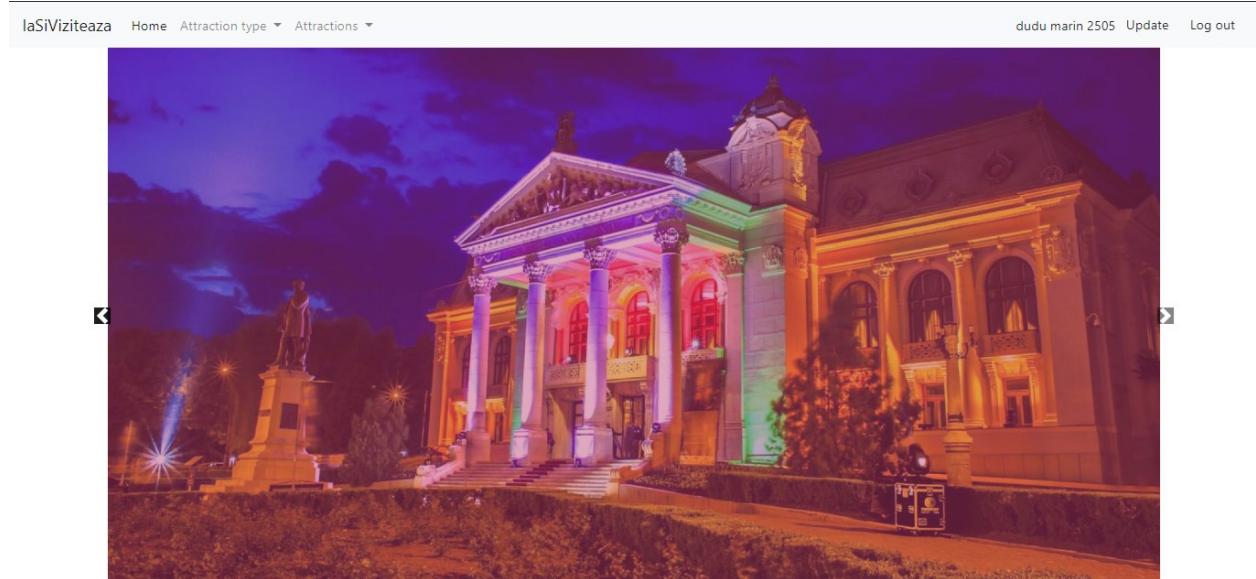
Proiect la disciplina
Proiectarea sistemelor de baze de date - proiect (TI)

Student
Ştefan Ignătescu

Iași, 2019

Introducere

„IaSiViziteaza” este o aplicație destinată turiștilor aflați în căutare de noi evenimente și locuri. Aceștia pot observa tipurile de atracții din oraș, atracțiile în sine, diverse comentarii lăsate de alții utilizatori, cât și rating-ul pentru o atracție anume.



Dezvoltare

Aplicația dezvoltată folosind o arhitectură pe 3 nivele, obținând astfel un nivel de abstractizare înalt. Tehnologiile folosite sunt: .Net (backend), AngularJs (fronted), Oracle (baza de date).

Baza de date – Modelul Logic și Relațional

Structura tablelor din baza de date a fost creata cu scopul de a menține cât mai multe date într-un mod cât mai simplu. Astfel avem următoarele modele:

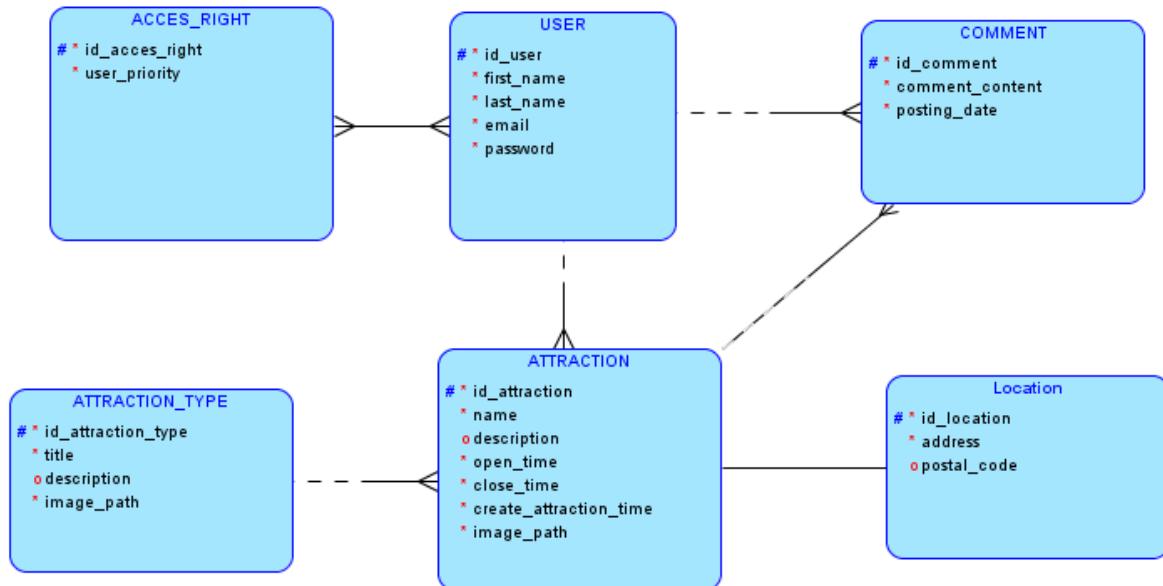


Figura 1: Model Logic

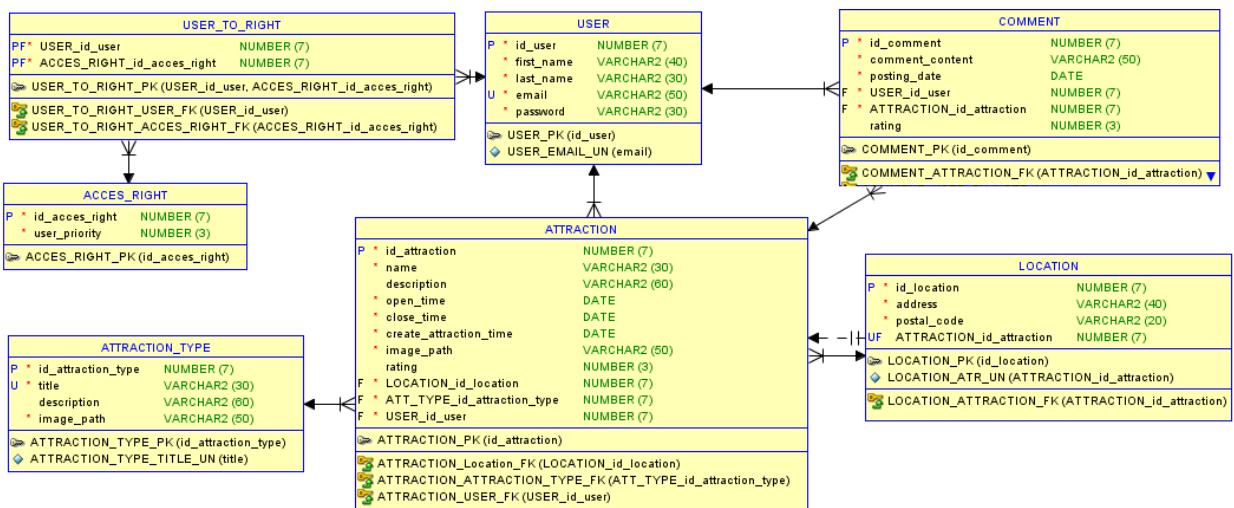


Figura 2: Model Relațional

Structură table și constrângeri

Tabelul „USER”

Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 ID_USER	NUMBER (7, 0)	No	"PROJECT"."ISEQ\$\$_75148".nextval	1	(null)	
2 FIRST_NAME	VARCHAR2 (40 BYTE)	No	(null)	2	(null)	
3 LAST_NAME	VARCHAR2 (30 BYTE)	No	(null)	3	(null)	
4 EMAIL	VARCHAR2 (50 BYTE)	No	(null)	4	(null)	
5 PASSWORD	VARCHAR2 (30 BYTE)	No	(null)	5	(null)	
6 PHONE_NUMBER	VARCHAR2 (15 BYTE)	Yes	(null)	6	(null)	

Are ca și cheie primara câmpul id_user, care se generează automat la înregistrarea unui nou utilizator, fiind de tipul autoincrement. Coloanele first_name, last_name și phone_number sunt folosite pentru a stoca diverse date despre un utilizator. Celelalte coloane sunt folosite pentru a loga un utilizator în aplicație, câmpul email fiind UNIQUE și având constrângerea de email iar câmpul password nu poate avea o lungime mai mică de 6 caractere.

Tabelul „ACCES_RIGHT”

Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 ID_ACCESES_RIGHT	NUMBER (7, 0)	No	"PROJECT"."ISEQ\$\$_75131".nextval	1	(null)	
2 USER_PRIORITY	NUMBER (3, 0)	No	(null)	2	(null)	

Este folosit pentru a gestiona drepturile de acces ale unui utilizator. Câmpul id_acces_right este cheie primară.

Tabelul „USER_TO_RIGHT”

Actions...						
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS	
1 USER_ID_USER	NUMBER (7, 0)	No	(null)	1	(null)	
2 ACCESES_RIGHT_ID_ACCESES_RIGHT	NUMBER (7, 0)	No	(null)	2	(null)	

Acest tabel este folosit pentru a obține o relație many-to-many dintre tabele USER și ACCES_RIGTH, astfel în acest tabel avem două chei străine user_id_user și acces_right_id_acces_right.

Tabelul „ATTRACTION”

Actions...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_ATTRACTION	NUMBER(7, 0)	No	"PROIECT"."ISEQ\$_75134".nextval	1	(null)
2 NAME	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3 DESCRIPTION	VARCHAR2(60 BYTE)	Yes	(null)	3	(null)
4 OPEN_TIME	DATE	No	(null)	4	(null)
5 CLOSE_TIME	DATE	No	(null)	5	(null)
6 CREATE_ATTRACTION_TIME	DATE	No	(null)	6	(null)
7 IMAGE_PATH	VARCHAR2(50 BYTE)	No	(null)	7	(null)
8 LOCATION_ID_LOCATION	NUMBER(7, 0)	No	(null)	8	(null)
9 ATT_TYPE_ID_ATTRACTION_TYPE	NUMBER(7, 0)	No	(null)	9	(null)
10 USER_ID_USER	NUMBER(7, 0)	No	(null)	10	(null)
11 RATING	NUMBER(3, 0)	Yes	0	11	(null)
12 PHONE_NUMBER	VARCHAR2(15 BYTE)	Yes	(null)	12	(null)

În acest tabel avem diverse coloane pentru a descrie o atracție, dar și coloane care ajuta la crearea relațiilor cu celelalte tabele. Câmpurile user_id_user,att_type_id_attraction_type sunt chei străine care formează o relație one-to-many cu tabelele „USER”, respectiv „ATTRACTION_TYPE”. Location_id_location este o cheie strană care realizeaza o relație one-to-one cu tabela „LOCATION”.

Tabelul „LOCATION”

Actions...					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_LOCATION	NUMBER(7, 0)	No	"PROIECT"."ISEQ\$_75144".nextval	1	(null)
2 ADDRESS	VARCHAR2(40 BYTE)	No	(null)	2	(null)
3 POSTAL_CODE	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4 ATTRACTION_ID_ATTRACTION	NUMBER(7, 0)	Yes	(null)	4	(null)

Acest tabel este într-o relație one-to-one cu tabelul „ATTRACTION”. O atracție poate avea o sigură locație, și într-o locație poate fi o sigură atracție, astfel îmagine de a insera o locație verificăm dacă codul poștal introdus există înregistrat.

```

CREATE OR REPLACE NONEDITIONABLE TRIGGER "PROIECT"."CHECK_INSERT_LOCATION" BEFORE
    INSERT ON location
    FOR EACH ROW
DECLARE
    v_check_postal_code NUMBER;
BEGIN
    SELECT
        COUNT(*)
    INTO v_check_postal_code
    FROM
        location
    WHERE
        postal_code = :new.postal_code;

    IF v_check_postal_code > 0 THEN
        raise_application_error(-20000, 'already exists a location at this postal code: ' || :new.postal_code);
    END IF;
END;

```

Tabelul „ATTRACTION_TYPE”

Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ATTRACTION_TYPE	NUMBER (7, 0)	No	"PROIECT"."ISEQ\$\$_75137".nextval	1	(null)
2	TITLE	VARCHAR2 (30 BYTE)	No	(null)	2	(null)
3	DESCRIPTION	VARCHAR2 (60 BYTE)	Yes	(null)	3	(null)
4	IMAGE_PATH	VARCHAR2 (50 BYTE)	No	(null)	4	(null)

Tabelul este folosit pentru a stoca tipuri generice de atracții. Conține o cheie primară id_attraction_type generată cu autoincrement. Câmpul title are o constrângere de tipul UNIQUE care nu permite înregistrarea mai multor tipuri de atracții cu același titlu.

Câmpul image_path conține calea către o imagine reprezentativă tipului respectiv. Imaginea este preluată din fronted, stocată în backend și calea către imagine este stocată în baza de date.

Tabelul „COMMENT”

Actions...						
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_COMMENT	NUMBER (7, 0)	No	"PROIECT"."ISEQ\$\$_75141".nextval	1	(null)
2	COMMENT_CONTENT	VARCHAR2 (50 BYTE)	No	(null)	2	(null)
3	POSTING_DATE	DATE	No	(null)	3	(null)
4	USER_ID_USER	NUMBER (7, 0)	No	(null)	4	(null)
5	ATTRACTION_ID_ATTRACTION	NUMBER (7, 0)	No	(null)	5	(null)
6	RATING	NUMBER (3, 0)	Yes	0	6	(null)

În acest tabel reține comentariile poste de utilizatori la diverse atracții. Se află într-o relație one-to-many cu tablele „USER” și „ATTRACTION”. Conține de asemenea un câmp pentru a reține rating-ul unui comentariu.

```

CREATE OR REPLACE NONEDITABLE PROCEDURE
"PROIECT"."UPDATE_COMMENT_RATING" (
    v_comment_id    IN          NUMBER,
    v_increase      IN          NUMBER
) AS
    v_current_rating NUMBER(3);
BEGIN
    SELECT
        rating
    INTO v_current_rating
    FROM
        "COMMENT"
    WHERE
        id_comment = v_comment_id;

    SAVEPOINT before_update_data;
    IF v_increase < 1 THEN
        v_current_rating := v_current_rating - 1;
    ELSE
        v_current_rating := v_current_rating + 1;
    END IF;
END;

```

```

END IF;

UPDATE "COMMENT"
SET
    rating = v_current_rating
WHERE
    id_comment = v_comment_id;

SELECT
    rating
INTO v_current_rating
FROM
    "COMMENT"
WHERE
    id_comment = v_comment_id;

IF v_current_rating < 0 THEN
    ROLLBACK TO before_update_data;
    raise_application_error(-20210, 'rating negative');
END IF;

END update_comment_rating;

```

Conectarea la baza de date

Aplicația se conectează la baza de date prin intermediul pachetului Oracle.ManagedDataAccess:

```

public static void CreateConnectionString()
{
    try
    {
        System.Diagnostics.Debug.WriteLine("Reading configurations.....");
        String serverName="localhost:1521",
               username= "project",
               password="parola1234";

        constr = "User Id=" + username +
                "; Password=" + password +
                "; Data Source=" + serverName +
                "; Pooling =false;";

        System.Diagnostics.Debug.WriteLine("ConstrVariable created ");
    }
    catch (Exception ConstrError)
    {
        System.Diagnostics.Debug.WriteLine(ConstrError.Message);
    }
}

public OracleConnection GetDBConnection()
{
    try
    {
        CreateConnectionString();
        conn = new OracleConnection(constr);
        conn.Open();
        System.Diagnostics.Debug.WriteLine("Connected");
    }
    catch (OracleException e)
    {
        System.Diagnostics.Debug.WriteLine("Not connected : " + e.ToString());
    }

    return conn;
}

```

```

public void CloseDBConnection()
{
    try
    {
        conn.Close();
        System.Diagnostics.Debug.WriteLine("Connection closed");
    }
    catch (OracleException e)
    {
        System.Diagnostics.Debug.WriteLine("Connection closed failed : " + e.ToString());
    }
    finally
    {
        System.Diagnostics.Debug.WriteLine("End..");
    }
}

```

Apelările unor proceduri/funcții în aplicație:

```

public void AddComment(Comment comment)
{
    OracleCommand cmd = new
    OracleCommand("insert_procedure.insert_comment", connection)
    {
        CommandType = CommandType.StoredProcedure
    };

    cmd.Parameters.Add("v_content", OracleDbType.Varchar2).Value =
comment.CommentContent;
    cmd.Parameters.Add("v_user_id", OracleDbType.Int32).Value =
comment.User.Id;
    cmd.Parameters.Add("v_attraction_id",
    OracleDbType.Int32).Value = comment.Attraction.Id;

    try
    {
        if(cmd.Connection.State!=ConnectionState.Open){
            cmd.Connection.Open();
        }
        cmd.ExecuteNonQuery();
    }
    catch (Exception e)
    {

        System.Diagnostics.Debug.WriteLine(e.ToString());
    }
    finally
    {
        DataBaseConnection.GetDbInstance().CloseDBConnection();
    }
}

```

```

    }

public IList<AttractionType> GetAttractionTypes()
{
    OracleCommand cmd = new
OracleCommand("get_function.get_attractiontype", connection)
    {
        CommandType = System.Data.CommandType.StoredProcedure
    };
    cmd.Parameters.Add(new OracleParameter("l_rc",
OracleDbType.RefCursor, ParameterDirection.Output));
    cmd.Parameters["l_rc"].Direction =
ParameterDirection.ReturnValue;

    OracleDataAdapter oracleDataAdapter = new
OracleDataAdapter(cmd);

    try
    {
        if(cmd.Connection.State!=ConnectionState.Open){
            cmd.Connection.Open();
        }
        cmd.ExecuteNonQuery();

        OracleRefCursor t =
(OracleRefCursor)cmd.Parameters[0].Value;

        OracleDataReader rdr = t.GetDataReader();

        var list = new List<AttractionType>();
        while (rdr.Read())
        {
            System.Diagnostics.Debug.WriteLine(rdr.GetInt32(0) +
rdr.GetString(1) + rdr.GetString(2) + rdr.GetString(3));
            list.Add(new AttractionType()
            {
                Id = rdr.GetInt32(0),
                Title = rdr.GetString(1),
                Description = rdr.GetString(2),
                ImagePath = rdr.GetString(3)
            });
        }
        return list;
    }
    catch (Exception e)
    {
        System.Diagnostics.Debug.WriteLine(e.ToString());
        return null;
    }
    finally
    {
        DataBaseConnection.GetDbInstance().CloseDBConnection();
    }
}

```

Exemplu de procedura/funcție stocată:

```
FUNCTION user_login (
    v_email VARCHAR2,
    v_password VARCHAR2
) RETURN SYS_REFCURSOR AS
    v_user SYS_REFCURSOR;
BEGIN
    OPEN v_user FOR SELECT
        *
        FROM
            "PROIECT"."USER"
        WHERE
            email = v_email
            AND password = v_password;

    RETURN v_user;
END user_login;

PROCEDURE update_user_info (
    v_old_email          IN          VARCHAR2,
    v_new_phone_number   IN          VARCHAR2,
    v_new_email          IN          VARCHAR2,
    v_new_password       IN          VARCHAR2
) AS
    v_check_new_email VARCHAR2(32);
BEGIN
    SELECT
        COUNT(*)
    INTO v_check_new_email
    FROM
        "USER"
    WHERE
        email = v_new_email;

    IF v_check_new_email > 0 THEN
        raise_application_error(-20100, 'already exists a user with
this email');
    ELSE
        UPDATE "USER"
        SET
            email = v_new_email,
            password = v_new_password,
            phone_number = v_new_phone_number
        WHERE
            email = v_old_email;
    END IF;
END update_user_info;
```