# Introducere

Aplicația își propune să ofere un sistem modern de colectare și gestionare a datelor în cadrul unei grădinițe, utilizând arhitectura de tip MVC Model-View-Controller, sub forma unei aplicații Web ASP.NET. Aplicația va salva informațiile într-o bază de date MySQL, care va fi creată în interiorul aplicației și va apela această bază de date pentru generarea statisticilor și a rapoartelor. Pentru o mai bună accesibilitate, informațiile vor fi introduse prin intermediul unei interfețe, în funcție de rolul stabilit pentru fiecare utilizator iar gestionarea datelor se va implementa prin funcțiile CRUD - Create, Read, Update, Delete.

Pentru atingerea obiectivului, aplicația trebuie să îndeplinească următoarele cerințe:

* Înregistrarea și gestionarea datelor legate de copii, educatori și grupele grădiniței.
* Înregistrarea copiilor în grupe și gestionarea fișei zilnice de prezență.
* Gestionarea taxelor aplicate și a plăților înregistrate.
* Asigurarea accesului în aplicație, pe baza rolurilor, atât a părinților cât și a cadrelor didactice.
* Generarea rapoartelor.

# Posibilități de rezolvare a problemei

Datorită multiplelor metode de implementare a acestei soluții dar și a caracterului general al aplicației, am să prezint în continuare câteva alternative pentru realizarea aplicației.

Aplicație Desktop ”Stand-Alone”, realizată în C# și conexiune la o bază de date locală de tip SQLite. exemplu 1, exemplu 2, etc.

Aplicație Web realizată în PHP și conexiune la baza de date.

Aplicație Desktop JAVA și conexiune la o bază de date locală.

(De completat aici cu ceva….)

# Justificarea metodei alese

Am optat pentru realizarea aplicației de tip Web deoarece oferă foarte multă flexibilitate în dezvoltare și îmbunătățirea ei. Un alt avantaj al acestei soluții este accesibilitatea. Fiind o aplicație Web, aceasta este găzduită în Internet și poate fi accesată chiar și de pe un dispozitiv mobil. Nu necesită instalare și nu necesită un calculator pe care să ruleze. De asemenea, o aplicație Web facilitează accesul tuturor utilizatorilor (atât a cadrelor didactice cât și a părinților care doresc să acceseze diverse rapoarte de încasări sau fișa de prezență a copilului lor).

# Prezentarea detaliată a cerințelor aplicației

Pentru atingerea obiectivului, aplicația trebuie să îndeplinească următoarele cerințe:

1. Să permită înregistrarea și gestionarea datelor de bază legate de copii, educatori și grupele grădiniței.
2. Să permită înregistrarea copiilor în grupe și gestionarea fișei zilnice de prezență.
3. Să gestioneze diversele tipuri de plăți și înregistrarea lor.
4. Să asigure accesului în aplicație, pe baza rolurilor, atât a părinților cât și a cadrelor didactice.
5. Să existe posibilitatea de a genera rapoarte.

În mod evident, înregistrarea și gestionarea datelor implică și conectarea aplicației la o bază de date. Tipul bazei de date alese este MySQL pentru că această componentă software este distribuită în mod gratuit. Clasele implementate în proiect: ”Children”, ”Instructor”, ”Group”, ”Enrollment”, ”DailyAttendance”, ”Payments” și ”FeeTypes” vor permite afișarea, editarea și ștergerea oricărei înregistrări. Pentru o mai bună accesibilitate în afișarea datelor se va implementa un modul care să permită funcțiile specifice pentru o bună paginare a rezultatelor din listele generate. Instrumentul ales este Widgetul Kendo UI Grid care oferă ”un control puternic pentru afișarea datelor într-un format tabelar. Acesta oferă numeroase opțiuni, cum ar fi paginarea, sortarea, filtrarea, gruparea și editarea, care determină modul în care datele sunt prezentate și manipulate. Gridul poate fi legat de date locale sau de la distanță utilizând componenta Kendo UI DataSource.”[[1]](#footnote-1)

## Înregistrarea și gestionarea datelor de bază

Pentru înregistrarea și gestionarea datelor de bază legate de copii, educatori și grupe, se vor inițializa tabelele necesare în baza de date MySQL, proces derulat automat de aplicația noastră.

În continuare, voi prezenta structura claselor care vor alcătui componenta de bază a aplicației:

* Clasa denumită ”Children”, având următoarele proprietăți:
  + Cheia primară ”CID” de tip int.
  + Coloana ”FirstName” de tip string care va conține numele.
  + Coloana ”LastName” de tip string care va conține prenumele.
  + Coloana ”CNP” de tip string care va conține CNP-ul.
  + Coloana ”Address” de tip string care va conține adresa.
  + Coloana ”City” de tip string care va conține orașul.
  + Coloana ”MothersName” de tip string care va conține numele mamei.
  + Coloana ”FathersName” de tip string care va conține numele tatălui.
  + Coloana ”ContactEmail” de tip string care va conține adresa de EMail a unuia dintre părinți. Această adresă va fi folosită ca și o cheie pentru corelarea conturilor părinților de informațiile legate de copil.

Pe langă datele care sunt introduse de către utilizatorul aplicației, clasa va conține și proprietăți derivate sau calculate din alte proprietăți ale aceleiași clase.

De exemplu:

* + Coloana ”FullName” de tip string care va concatena numele și prenumele.
  + Coloana ”Gender” de tip string care va conține genul copilului, informație calculată din CNP-ul introdus.
  + Coloana ”DOB” – Date of Birth, de tipul DateTime care va genera data de naștere a copilului în funcție de CNP-ul introdus.
* Clasa denumită ”Instructor”, având următoarele proprietăți:
  + Cheia primară ”InstructorID” de tip int.
  + Coloana ”FirstName” de tip string care va conține numele.
  + Coloana ”LastName” de tip string care va conține prenumele.
  + Coloana ”StartDate” de tip DateTime care va conține data la care instructorul a început activitatea.
  + Coloana ”EndDate” de tip DateTime care va conține data la care instructorul a încetat activitatea.
  + Coloana ”PayRate” de tip decimal care va conține salariul lunar al instructorului.
  + Coloana ”Email” de tip string care va conține adresa de email.
  + Coloana ”FullName” de tip string care va concatena numele și prenumele.
* Clasa denumită ”Group” care va conține următoarele proprietăți:
  + Cheia primară ”GroupID” de tip int.
  + Coloana ”GroupName” de tip string care va conține numele.

# Înregistrarea copiilor în grupe și gestionarea fișei zilnice de prezență

Pentru acest proces vom avea nevoie de informații suplimentare din alte tabele iar din acest motiv, vom folosi ”Foreign Key”-urile atât în tabelele din baza de date cât și în clasele aplicației.

* Clasa denumită ”Enrollments” având următoarele proprietăți:
  + Cheia primară ”EnrollmentID” de tip int.
  + Coloana ”CID” de tip int care va face referire la ID-ul din tabela ”Children” (Foreign Key).
  + Coloana ”InstructorID” de tip int care va face referire la ID-ul din tabela ”Instructor” (Foreign Key).
  + Coloana ”GroupID” de tip int care va face referire la ID-ul din tabela ”Group” (Foreign Key).
  + Coloana ”Date” de tip DateTime care va conține data înregistrării în grupă.

Pentru a accesa datele suplimentare ale tabelelor accesibile prin ”Foreign Key”, se vor include și proprietățile virtuale ale tabelelor menționate:



* Clasa denumită ”DailyAttendance” cu următoarele proprietăți:
  + Cheia primară ”AttendanceID” de tip int.
  + Coloana ”Att\_Date” de tip DateTime care va conține data înregistrării.
  + Coloana ”isPresent” de tip bool care va conține valoarea ”true” sau ”false”.
  + Coloana ”Notes” de tip string care va conține note adiționale.

De asemenea, clasa va conține și referința la tabela ”Children”:



## Gestionarea diverselor tipuri de plăți și înregistrarea lor

Pentru gestionarea plăților, aplicația va folosi două tabele:

* Clasa ”Payments” având următoarele proprietăți:
  + Cheia primară ”PaymentID” de tip int.
  + Coloana ”CID” de tip int care va face referire la ID-ul din tabela ”Children” (ForeignKey).
  + Coloana ”Amount” de tip decimal care va conține suma plătită.
  + Coloana ”FeeID” de tip int care va face referire la ID-ul din tabela ”FeeTypes” (ForeignKey).
  + Coloana Date de tip DateTime care va conține data plății.

De asemenea, vom enumera și tabelele pe care le invocăm:



* Clasa ”FeeTypes” care va conține următoarele proprietăți:
  + Cheia primară ”FeeID” de tip int.
  + Coloana ”FeeType” de tip string care va conține numele.

## Asigurarea accesului în aplicație, pe baza rolurilor, atât a părinților cât și a cadrelor didactice

Se va implementa modulul de autorizare și autentificare specific arhitecturii software de tip ”Model-View-Controller”.

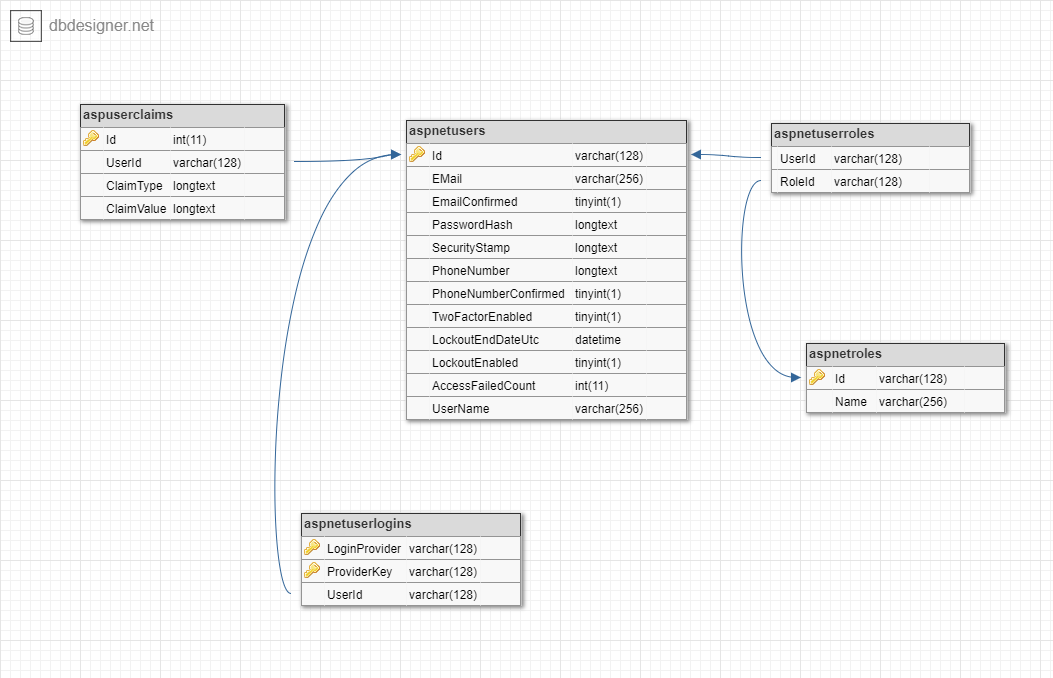


Fig. 1.1 – Reprezentarea grafică a tabelelor de autorizare specifice aplicației ASP.NET MVC

1. https://docs.telerik.com/kendo-ui/controls/data-management/grid/overview [↑](#footnote-ref-1)