**MANAGEMENT APP FOR A FOREIGN LANGUAGES SCHOOL**

This is a platform, intended for the management of the school courses, and classes, and for the management of the users who have access to these courses.

The main services of this platform are:

* Registration Activity
* School Activity
* Teacher Activity
* Classes Activity
* Students Activity

ADMIN USER can do for each entity:

* Course: search, view description of selected course, add, delete, update Course
* Class: search, view description of selected class, add, delete, update class, view Class Timetable and class messages
* Student: search, view description of selected student, add, delete, update, manage Student payment (search, add, delete, update) and student school history(search, add, update), view student presence, view student class evaluation, view class messages
* Contact Person: search, add, delete, update contact person for a student
* Teacher: search, view description of selected teacher, add, delete, update

TEACHER USER can do for each entity:

* Course: search, view description of selected course
* Class: search, view description of selected class,
* Manage Class Timetable (search, add, delete, update)
* Manage Class Messages (search, add, delete, update)
* Student:
  + search, view description of selected student, view student school history, view ContactPerson
  + manage student presence, student class evaluation (search, add, delete, update)

STUDENT USER can do for each entity:

* Course: search, view short description of selected course
* Class: view description of his classes
* View his Class Timetable
* View his Class Messages
* View his Presence and his Evaluation

**DBO TABLES:**

1. User: Id(primary key, identity), UserName, LastName, FirstName, Password, DateBirth, CreateDate, City, Adress, Phone, Email
2. Admin: Id(primary key, identity), UserId(foreign key to User by Id), Role
3. Student: Id(primary key, identity), UserId(foreign key to User by Id), Status active, ContactId (foreign key to ContactPerson by Id)
4. ContactPerson: Id(primary key, identity), Name, Adress, Phone, Email
5. Teacher: Id(primary key, identity), UserId(foreign key to User by Id), Status active
6. Course: Id(primary key, identity), NumberOfLessons, Description, Category, Language, Level, StatusActive
7. Class: Id(primary key, identity), CourseId(foreign key to Course by Id), TeacherId(foreign key to Teacher by Id), Name, Descriptin, StartDate, EndDate
8. StudentClass: Id(primary key, identity), ClassId(foreign key to Class by Id), StudentId(foreign key to Student by Id) -manage m-n relationship between Class and Student
9. ClassTimeTable: Id(primary key, identity), ClassId(foreign key to Class by Id), LessonNumber, Date
10. ClassMessage: Id(primary key, identity), ClassId(foreign key to Class by Id), Date, Message
11. StudentPresence: Id(primary key, identity), TimeTableId(foreign key to TimeTable by Id), Presence
12. StudentClassEvaluation: Id(primary key, identity), ClassId(foreign key to Class by Id), StudentId(foreign key to Student by Id), Description, Grade
13. StudentPayment: Id(primary key, identity), StudentId(foreign key to Student by Id), ClassId(foreign key to Class by Id), PaymentDate, Ammount

**DESIGN** **CLASSES IN C#:**

EntityBase: base class for all classes with propertie Id-

User: inherit EntityBase, with properties: UserName, LastName, FirstName, Password, DateBirth, CreateDate, City, Adress, Phone, Email, list UsersList

Admin: inherit User, with properties: UserId, Role, list AllAdmins

Student: inherit User, has ContactPerson, with properties: UserId, StatusActive, ContactPerson, list AllAdmins

Teacher: inherit User, with properties: UserId, Status, TeacherClasses, list TeachersList

ContactPerson: inherit EntityBase, with properties: Name, Adress, Phone, Email

StudentClassEvaluation: inherit EntityBase, has Class and Student, with properties: Class, Student, Description, Grade

StudentPayment: inherit EntityBase, has Class and Student, with properties: Student, Class, PaymentDate, Ammount, list StudentPayments

StudentPresence: inherit EntityBase.

Course: inherit EntityBase, with properties NumberOfLessons, Description, Category, Language, Level, StatusActive

Class: inherit Course, has Teacher, with properties: CourseId, Teacher, Name, ClassDescription, StartDate, EndDate, liste: ClassTimetables, StudentPayments, StudentClassEvaluations, AllClasses

ClassTimeTable: inherit EntityBase, has Class, with properties: Class, LessonNumber, ClassDate

Message: inherit EntityBase, has Class, with properties: Date, MessageBody, list MessagesPerClass

StudentClass: inherit EntityBase, ClassId(foreign key to Class by Id), StudentId(foreign key to Student by Id) -manage m-n relationship between Class and Student

Ce am facut:

* SQL: Am creat baza de date cu tabelele, am populat doar tabelul Course, am facut o procedura stocata generica dar nu am reusit sa o apelez din program
* C#:
  + DataAccessConnection cu 2 clase: Connectionmanager si SqlHelper
  + Entities in SchoolDBModel (nu stiu daca trebuie sa suprascriu ToString in entities si daca pot sa adaug in entities def pt colectii de obiecte)
  + LanguageSchool.Repositories: IBaseRepository, BaseRepository(generic class), CoursRepository
  + BaseRepository: generic Delete, generic Delete list of objects, GetAll (returneaza SqlReader), GetById (returneaza SqlReader), generic Count
  + CourseRepository: Add, Save, Update, GetAll return to List, GetById return to list, ReadReader (return SqlReader to List), ReadRow(return SqlReader for a row to an object), ToString methord for Course and for Course List