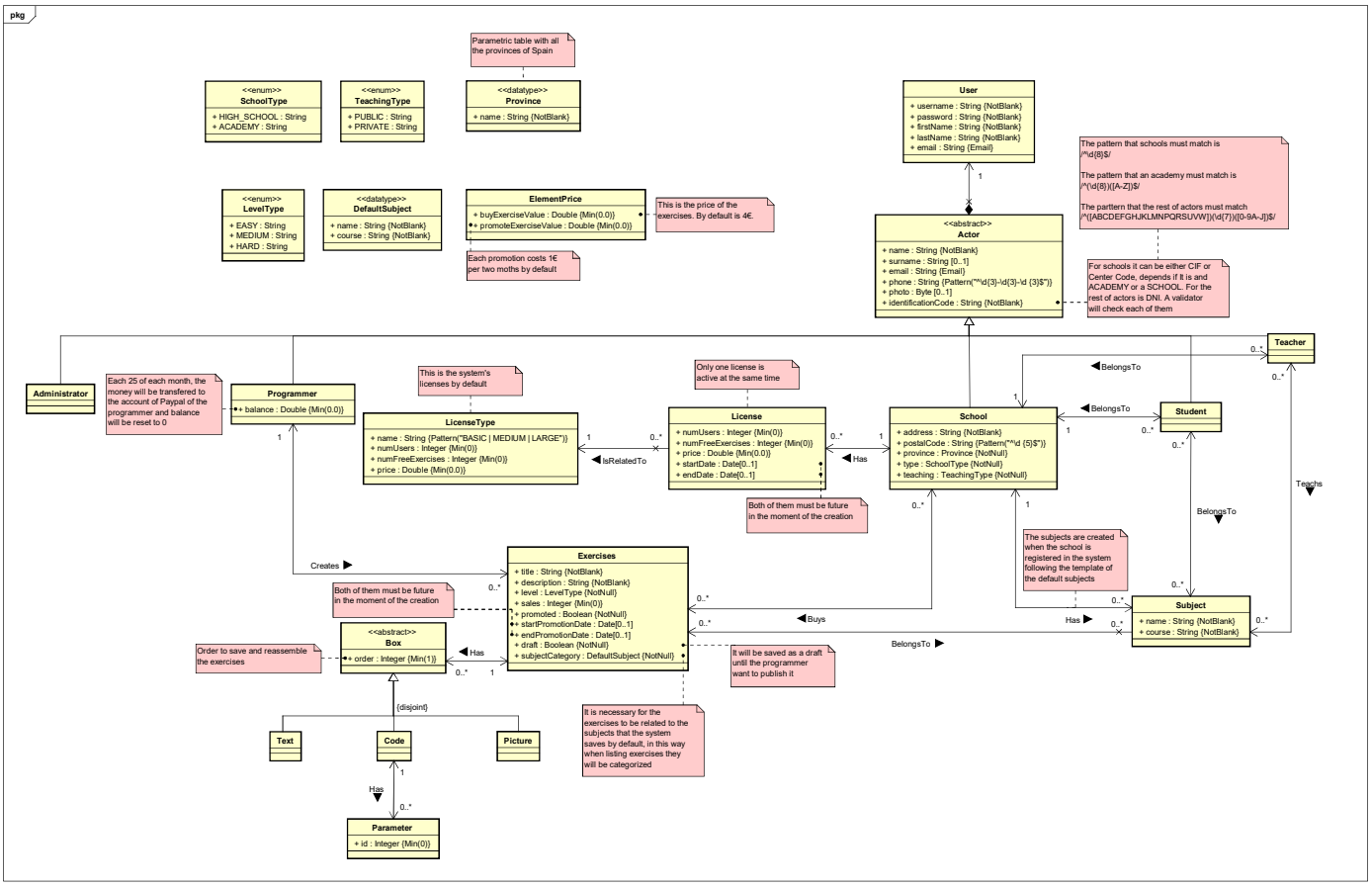
pkg
Parametric table with all the provinces of Spain
<<enum>>
SchoolType
+ HIGH_SCHOOL : String
+ ACADEMY : String
<<enum>>
TeachingType
+ PUBLIC : String
+ PRIVATE : String
<<datatype>>
Province
+ name : String {NotBlank}
User
+ username : String {NotBlank}
+ password : String {NotBlank}
+ firstName : String {NotBlank}
+ lastName : String {NotBlank}
+ email : String {Email}
The pattern that schools must match is
/^\d{8}$/
The pattern that an academy must match is
/^(\d{8})([A-Z])$/
The partern that the rest of actors must match
/^([ABCDEFGHJKLMNPQRSUVW])(\d{7})([0-9A-J])$/
<<enum>>
LevelType
+ EASY : String
+ MEDIUM : String
+ HARD : String
<<datatype>>
DefaultSubject
+ name : String {NotBlank}
+ course : String {NotBlank}
ElementPrice
+ buyExerciseValue : Double {Min(0.0)}
+ promoteExerciseValue : Double {Min(0.0)}
This is the price of the exercises. By default is 4€.
Each promotion costs 1€ per two moths by default
<<abstract>>
Actor
+ name : String {NotBlank}
+ surname : String [0..1]
+ email : String {Email}
+ phone : String {Pattern("^\d{3}-\d{3}-\d {3}$")}
+ photo : Byte [0..1]
+ identificationCode : String {NotBlank}
For schools it can be either CIF or Center Code, depends if it is and ACADEMY or a SCHOOL. For the rest of actors is DNI. A validator will check each of them
Teacher
BelongsTo
Administrator
Each 25 of each month, the money will be transfered to the account of Paypal of the programmer and balance will be reset to 0
Programmer
+ balance : Double {Min(0.0)}
This is the system's licenses by default
LicenseType
+ name : String {Pattern("BASIC | MEDIUM | LARGE")}
+ numUsers : Integer {Min(0)}
+ numFreeExercises : Integer {Min(0)}
+ price : Double {Min(0.0)}
IsRelatedTo
Only one license is active at the same time
License
+ numUsers : Integer {Min(0)}
+ numFreeExercises : Integer {Min(0)}
+ price : Double {Min(0.0)}
+ startDate : Date[0..1]
+ endDate : Date[0..1]
Has
Both of them must be future in the moment of the creation
School
+ address : String {NotBlank}
+ postalCode : String {Pattern("^\d {5}$")}
+ province : Province {NotNull}
+ type : SchoolType {NotNull}
+ teaching : TeachingType {NotNull}
Student
Teachs
The subjects are created when the school is registered in the system following the template of the default subjects
Creates
Exercises
+ title : String {NotBlank}
+ description : String {NotBlank}
+ level : LeveltType {NotNull}
+ sales : Integer {Min(0)}
+ promoted : Boolean {NotNull}
+ startPromotionDate : Date[0..1]
+ endPromotionDate : Date[0..1]
+ draft : Boolean {NotNull}
+ subjectCategory : DefaultSubject {NotNull}
Both of them must be future in the moment of the creation
Buys
Subject
+ name : String {NotBlank}
+ course : String {NotBlank}
Order to save and reassemble the exercises
<<abstract>>
Box
+ order : Integer {Min(1)}
It will be saved as a draft until the programmer want to publish it
{disjoint}
Text
Code
Picture
It is necessary for the exercises to be related to the subjects that the system saves by default, in this way when listing exercises they will be categorized
Parameter
+ id : Integer {Min(0)}