ClimbCode

ClimbCode is the result of a fusion between teaching and programming. This fusion is, in fact, a platform used to support the teaching of different subjects at high schools and academies. This support is given as interactive exercises made by programmers which use programming in order to teach other concepts from other different subjects.

Information requirements

- **Actors:** The actors of the system can be either administrators, schools, teachers, students and programmers.
    - For schools, the system must store: name, address, postal address, province, city, email, *the type of school (academia, publica, privada…), datos fiscales… (TO-DO).*
    - For the other actors, the system must store his or her name, surname, email, nick, phone, identification card and optionally a photo *(TO-DO: algo más?).*
    - Specifically, programmers need to register a Paypal account if they want to cash the profits they obtain by selling exercises.

- **Licenses:** schools select a basic license when it's registering in the system; this basic license determines its capabilities on the platform: a number of users (teachers or students) and some free exercises. For every license, the system must store the type (or name), the number of users (teachers or students), a free number of exercises and a fixed price. Also the system let schools to add extra users to the basic licenses (and the system will update the price).
    - There are three types of licenses initially: "BASIC", "MEDIUM", "ADVANCED":
        - BASIC: 60 users and 20 free exercises for 200€/year.
        - MEDIUM: 200 users and 50 free exercises for 550€/year.
        - ADVANCED: 500 users and 75 exercises for 700€/year.

- **Exercises:** the system must store the following information: title, description, details, code, subject, level of difficulty and number of sales.
    - There are three difficulty levels: "LOW", "INTERMEDIUM", "DIFFICULT".

Exercises are created by programmers, so they can earn profits for every school which buys that. Every exercise is priced at 4 € (1 € for the system (25%), 3 € for the programmer) and the money (of every sale) is stored in the programmer deposit which is cashed automatically by the system each 25 of every month using the programmer's Paypal account.

The exercises can be saved initially as a draft until the programmer decides to publish them. When an exercise is published, it can be bought by schools.

- **Promotions:** a programmer can promote an exercise which he or she had created in order to advertise them in privilege positions into the system. So they appear first? *(Yo lo veo como que aparezca por encima del listado algo así como "Ejercicios*

patrocinados") or in another tabla (tabla de recomendados antes de la tabla general de ejercicios).

Programmers must register a valid PayPal account before they can advertise their exercises. Every promotion is priced at 1€ per two months (60 days). After that period, the exercise will lost its promotion and it will show as normal exercises.

- **Subject Types (for programmers)**: administrators can create one or more subjects in order to categorize exercises. The system must store for every subject: a code, name and a course.

This subject type will be used for programmers in order to categorize the exercises they created.

- **Subjects (for schools)**: schools can manage their own subjects in order to save the exercises they bought. For every school subject, the school must inform the following fields: code, name, course and academic year.

Functional requirements

*A user who is **not authenticated** must be able to:*
- See the welcome and contact pages.
- Register as programmer (with a free account). Note that a programmer must register a valid Paypal account before completing the registering.
- Register into the system as school through one of the licenses offered by the system. In addition, the school can add an extra number of users if the license does not adapt to its needs. *(Esto es, selecciona una licencia y puede pagarla tal cual o añadir usuarios (calcula precio de usuario dividiendo) y se recalcula el precio).* The school must pay before complete the registration.

*An actor who is **authenticated** must be able to:*
- Do the same as an actor who is not authenticated except registering into the system.
- Edit his/her profile.

*An actor who is authenticated as a **programme**r must be able to:*
- List all the exercises in the system. It must appear only the name and details. The list can be filtered by keywords, level or subject.
- List the exercises that he or she has created. He or she must see the sales of them.
- Create, modify or delete his/her exercises. The exercises will be saved initially as a draft, until the programmer decides to publish them.
- Apply for a promotion of his or her exercises.
- Display his/her profits deposit: that is the amount of money he/she has earn by selling in the current month.

*An actor who is authenticated as a **school** must be able to:*
- List all the exercises in the system. It must appear only the name and details. The list can be filtered by keywords, level or subject.
- Select the free exercises that are included in their license.
- Buy exercises.

- Manage its teachers: list, create, modify and delete. A teacher must be related with the different subjects he/she teachs.
- Manage its students: list, create, modify and delete. A student must be related with the subjects in which he/she is enrolled.
- Manage its own subjects: list, create, modify and delete.

*An actor who is authenticated as a **teacher** must be able to:*
- List all the exercises in the platform. They must show only the name and details. The list can be filtered by keywords, level or subject.
- List all the subjects of his or her school.
- List only the subjects he or she teaches.
- See and execute the exercises his/her school owns and see their results.

*An actor who is authenticated as a **student** must be able to:*
- List only the subjects he or she is enrolled.
- See and do the exercises of his/her subjects.

*An actor who is authenticated as an **administrator** must be able to:*
- Delete exercises if the content is not appropriate.
- Manage schools: list, create, modify and delete.
- Manage general subjects (to categorize exercises): list, create, modify and delete.
- Manage licenses: create or update templates.

Non-Functional requirements

- The system must track the actual profits of the programmers, so they can check it anytime.
- *Teachers can see how many students do the exercises.*
- All the listings in the system can be ordered by its different fields.
- The system will be able to pay programmers each 25 of every month using Paypal. Until that time, the profits are stored in the programmer deposit.