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 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 every actor the system must store his or her name, email, phone and an identification code and optimally a photo and a surname in case if it is student, teacher or programmer. Additionally:

  - For schools, the system must store: address, postal code, province, the type of school (Academy, High School) and the type of teaching (Public, Private). If the school is an academy, it must have a CIF; if the school is a High School, it must have a center code.
  - Teachers and students must be related to a school.

- **Licenses:** schools have a license when they're 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, the number of users (teachers or students), a free number of exercises, the starting and expiration date 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 default licenses: "BASIC", "MEDIUM", "ADVANCED":
    - § BASIC: 60 users and 20 free exercises for 200€/year.
    - § MEDIUM: 200 users and 50 free exercises for 550€/year.
    - § LARGE: 500 users and 75 exercises for 700€/year.
  - The default types of licenses are stored in a specific table

- **Exercises:** the system must store the following information: title, description, code, subject, level of difficulty and number of sales.

  - There are three difficulty levels: "EASY", "MEDIUM", "HARD".

  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 balance 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. They appear in the main page in a list made for it. While promoted they appear in the usual list too.

  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, needing to promote it again.

- **Subjects**: 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.

  Each subject must be related to a system default's one, so it will can used for programmers in order to categorize the exercises they create. Also, when a school is registered into the system, it will be associated with the default subjects.

Functional requirements

*A user who is **not authenticated** must be able to:*

- See the welcome and contact pages.
- Log-in into the system using his/her credentials.
- Register as programmer (with a free account).
- 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. The school must pay before completing 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 earned 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, details and number of sales. The list can be filtered by keywords, level or subject.
- Select the free exercises that are included in their license.
- Buy exercises. Note that a school must register a valid PayPal account before completing the registering to do so. Note that when a school buys an exercise, the system automatically check if the school has a subject for it. If it exists, the exercise is added to

the school subject; if not, the system creates a subject (for the school in exclusivity) with the same parameters and added the exercise to that.

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

*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, details and number of sales. 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.
- 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.