

ClimbCode

ClimbCode is a platform oriented to help teaching subjects through programming.

Its principal use is to store all the documents of the subjects and offer support to them by means of interactive exercises which show the concepts of the subjects through programming.

Information requirements

- 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, surname, email, nick, phone, identification card, address, and optionally a photo and additional information.
- Administrators are people who manage all the system. As so, they must be able to manage the licenses which they sell to schools. Every license has a code, numbers of students, number of subjects and a price.
- Programmers are the ones that can create exercises for schools. For every exercise, the system must store the following information: title, description, details, code, subject, level of difficulty and sales. Level of difficulty can be "BASIC", "MEDIUM", "ADVANCED".
- A programmer can advertise an exercise which he or she had created. Programmers must register a valid PayPal account before they can advertise an exercise of theirs.
- Administrators can create one or more subjects. The system must store for every subject a code, name, course, academic year, type, documents and a list of syllabuses. Type can be either "ESO" or "A LEVELS". Each document is described by means of a name, description, content, download link and a syllabus.
- Exercises are created by programmers, so they can earn profits as deposits. For every deposit, the system must store a date and a quantity. Deposits can be retired each 25 of every month.

Functional requirements

- A user who is not authenticated must be able to:
 - o See the welcome page.
 - o See the contact page.
- An actor who is authenticated must be able to:
 - o Do the same as an actor who is not authenticated.
- An actor who is authenticated as a teacher or a student must be able to:
 - o Change the information in his or her profile.
 - o Execute exercises and see their results.
- An actor who is authenticated as a school, teacher or programmer must be able to:
 - o List all the exercises in the platform. It must appear only the name and details.
- An actor who is authenticated as a programmer must be able to:
 - o List all the exercises that he or she has created. He or she must see the sales of them.
 - o Create, modify or delete exercises.
 - o Advertise his or her exercises.
- An actor who is authenticated as a school must be able to:

- Buy a cancel licenses.
- Buy exercises.
- Associate a student with a subject.
- List, create, modify and delete teachers.
- List, create, modify and delete students.
- Manage the content of already created subjects.
- An actor who is authenticated as a teacher must be able to:
 - Execute exercises and see their results.
 - Previsualize exercises.
 - List all the subjects of his or her school.
 - Upload and delete documentation to his or her subjects.
 - Contact school to buy an exercise.
- An actor who is authenticated as a student must be able to:
 - See and download the documentation of the subject to which he or she is registered.
- An actor who is authenticated as an administrator must be able to:
 - List and delete exercises if the content is not appropriate.
 - List, create, modify and delete schools.
 - Approve an advertising request of a programmer.
 - List, create, modify and delete subjects.
 - Add subjects to schools so they can add their own content.
 - Administrate school's licenses.

Non-functional requeriments

- The system must track the actual profits of the programmers, so they can check it anytime.
- Professors can see how many students do the exercises.