Database scheme

The database is organized in collections:

- `contests`, to manage contests
- `submissions`, to store user's submissions
- `users`, to store users and their datas

`contests` collection

This collection contain the contest's data. It's a `Contest` object.

Data types

Testcase

```
{
    "input": "here goes the input",
    "output": "here goes the output",//Optional field if you don't want to always check with the solution
}
```

Subtask

```
{
    "score": 30,//how many points to give if every testcase is completed correctly
    "testcases": [],//Array of Testcase
}
```

Task

```
{
    "name": "task name",
    "full_name": "Friendly name",
    "time": "time limit",
    "memory": "memory limit",
    "has_subtask": true,//or false
    "testcases": [], //array of Testcase, mandatory only if has_subtask=false
    "score": 100,//mandatory only if has_subtask=false, the amount of point is then equally divided between t
    "subtask":[],//array of Subtask, mandatory only if has_subtask=true
    "statement": "base64-encoded PDF",
    "grader": "the source code used to evaluate the answers"
}
```

Contest

```
{
    "name": "Contest's name",
    "date_start": null,//Use a Date object instead of null
    "date_end": null,//Same as date_start
    "task": []//array of Task
}
```

`submissions` collection

This colection contains all submissions. It's made out of `Submission` as below.

Submission

```
{
    "user": "the username of whom submitted the solution",
    "id": "an univoque id of the submission",
    "task": "task's name",
    "source": "the c++ file itself",
    "date": null, //The date of submission (new Date())
    "status": "compiling, evaluating or evaluated",
    "score": 100//null when compiling or evaluating
}
```

`users` collection

This colection contains all users. It's made out of `User` as below.

User

```
{
    "username": "the username",
    "full_name": "user's full name",
    "password": "BCRYPTed password"
}
```