# IAESTE Student Project - GitHub Student Team Assessment Tool (GhSTAT)

## Remote Services:

So far only the GitHub API will be needed. In case we decide that integration with Trello is necessary then we can use Trello API. My suggestion is that at the current phase we avoid involving Trello. A similar website to Trello is Piazza which was used in different courses and helped students interacting with each other and with the instructor. Although the assessment in these courses did not cover students' contributions to piazza, however some instructors may think it should.

As a conclusion the following Remote services must be examined:

1. GitHup API (A must)
2. Trello API (Subject to discussion)

While we work new API's may be needed, but for now this seems acceptable

## Client:

We strongly suggest using web application as a first step. These of course has two major disadvantages

1. Further Development for various browser types is needed (Internet Explorer vs. Mozzila Firefox/Google Chrome browsers)
2. You need to host the application somewhere centrally preferably perhaps in the same server with the web services that are described on Domain

On the other hand nowadays internet can be reached from anywhere even from mobile phones making this choice much more flexible than the other candidate (desktop application). Regarding the other choices are very specific and specialized and do not fit well in the broad selection of devices that exist today. Our suggestion is to go through HTML/CSS/JavaScript with ajax remote calls where necessary.

## Domain:

We will also use REST web services in order to communicate with the API that is exposed by Git Hub. Furthermore these web services will reside in the same server as the server that will host the client part. We can use ASP.NET either (VB.Net or C#) for the implementation of the Web Services. Java EE approach could also be considered.

**Databases:**

Dueto the fact thatwe are accessing the information through the Github API at the present phase the use of a Database is considered unnecessary. At a later phase we can use a DB (SQL Express or MySQL) in order to store various data for statistical reasons.

A draft of proposed architecture is appeared below:

**Domain**

Web services

**Remote Services**

**CLIENTS**

Mobile Devices

Web Browsers

DB

GitHub API