

School of Engineering and Built Environment

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

## Task 2 – Define a proposed application architecture

### Background – agile methods and application architecture

It is suggested that this application should be developed using **agile** methods. When developing software using an agile approach, you don’t do detailed design of the whole system up-front. Instead, we build the system by incrementally implementing features defined through capturing the requirements as, for example, use cases or user stories. During each increment, a set of features are implemented, and there should be a product at the end of each increment in the form of working software which, though not the finished product, could be delivered and used. The features selected for implementation in each increment will depend on the progress made in the previous one, and the process can adapt to changes and issues which arise during development.

Even when following this approach, however, it is important to discuss and define an overall architecture for the system, which will guide the way in which the features are implemented. Essentially, you need to decide what the main **components** of the system will be, where will these components be located, what platforms they will be built on and how will they interact? Each feature will probably make use of all these main components and require them to interact to provide that single piece of functionality.

Examples of main components which may become part of your application include:

* **Remote services** - this application will definitely make use of the GitHub APi which is available as a REST web service. Are there any other remote services required? How will the application communicate with the remote API(s)?
* **Client** – this is the component which will present information to the user and respond to user actions. How will this client be made available to the user? Will it be a desktop application, a web application, a mobile application, or perhaps more than one of these? Or, will it be an application which is hosted inside some other application, for example a Chrome extension or a plug-in for a virtual learning environment such as Moodle? How will the client be made available to new users, for example by accessing a URL or installing an application? What programming languages/frameworks are available for building the client type you propose?
* **Domain** – this component defines the entities which define the information held in the system and the behaviour associated with this information. These entities interact to provide the underlying knowledge, rules and logic. Here, this will determine how the raw data retrieved from the GitHub will be given meaning and transformed into the information to be presented to the user. In this application, these rules are likely to be refined over time based on research findings, but the prototype needs to have an initial implementation based on a “first guess” at these rules. Where will the domain model code be located – will it be in the same process/platform as the client, or will it be a separate component, perhaps even on a physically separate tier? In either case, what programming languages/frameworks are available for building the component type you propose? How will the client communicate with the domain – for example by working directly with domain objects, or through another service component which provides a convenient gateway, or façade, for the domain?
* **Storage** – this component provides persistent storage for the information in the system, and will typically be implemented with a database (relational or other). Does this application actually need to store any information, or will it simply present information immediately as it is derived from GitHub? If you need to store data, where will storage be located? How will the other components interact with storage?

### Task – proposed application architecture

Investigate, consider and discuss the questions associated with the components listed above and make choices in each case – choices may be based on your investigations, your own expertise and experience and on discussions if possible with potential users (the latter particularly when considering the form of the client component)

**Deliverable:**

Produce a document which briefly summarises your proposed architecture. A diagram (or diagrams) may be useful in communicating the nature of the components and their relationships.

**Feedback:**

The project supervisors will review the proposed architecture and provide feedback to help refine the proposal towards an architecture which will be adopted for the application.