Assignment T1: Preliminary Project Proposal

Published | Edit | ⋮

This is a team assignment. See submission instructions at the end.

Invent an idea for a useful service that 1. maintains some relevant data persistently during its operation, 2. supports multiple clients, and 3. has no GUI.

Maintaining data persistently means that when your service terminates and starts up again some arbitrary time later, the data still exists and can be used and modified by later processing. The datastore should be read/write, not append-only. Your service is not required to support crash recovery, in the sense of the machine suddenly losing power or rebooting, only intentional termination (shut-down) of either the service itself or some supporting infrastructure.

Multiple clients means some of the data may be associated with one client and some with another. There may also be some aggregate or shared data. Your service needs to be able to tell the clients apart and it should provide basic protection for the integrity and confidentiality of client-specific data, protecting it from other clients. You do not need to consider carefully crafted hacker exploits or protecting client data from the service-provider itself. It is strongly recommended that you use some "free" widely-used authentication package, e.g., **Auth0** (https://auth0.com/docs/), not develop the security code yourself.

Instead of a GUI (graphical user interface), your service should provide an API (Application Programming Interface). The API does not need to be a REST API. The API inputs might be intended to ultimately come from some application that does have a GUI (or multiple different GUI apps), and you can optionally built a sample GUI client as part of the second iteration, but for the first iteration you need to develop a service with no GUI (and no pseudo-GUI like **ascii art** (https://www.asciiart.eu/)) and devise some non-GUI way to demonstrate that your service "works", e.g., using **Postman** (https://www.postman.com/) or a CLI (command line interface). You can use **this link** (https://docs.google.com/forms/d/e/1FAIpQLSdWrdayMvsD1PssUA3Za6l5dLpbwprW7UpAQwTdXqkQdDW usp=pp_url&entry.1767743879=kaiser%40cs.columbia.edu&entry.1788256813=Columbia%20University&er to sign up for a free Postman account (please inform the instructor asap if the link does not work).

Web servers, application servers, content servers, database servers, etc. are not acceptable; your service must "do something" other than host other applications and services. You can look at **https://github.com/public-apis/public-apis** (https://github.com/public-apis/public-apis) and/or **https://github.com/n0shake/Public-APIs** (https://github.com/n0shake/Public-APIs) for inspiration but you need to do something at least margically different, not reimplement verbatim one of the APIs listed on those pages. **Google cloud accounts** (https://cloud.google.com/billing/docs/how-

to/edu-grants) with very limited credits can be requested from the instructor by sending a note on piazza; please do not request unless you really plan to use for your team project.

Part 1:

If your team name, team membership, planned programming language, platform and/or team github repository has changed since your team formation assignment, please explain. If not, simply state that nothing has changed. You need permission from the instructor *in advance* (before you submit this assignment) to change to any language other than C++, Java, Python and to change to any platform other than Linux, Mac, Windows.

Part 2:

Write a few paragraphs that provide an overview of the service that your team would like to develop and answers these three sets of questions: 1. What will your service do? What kind of functionality or features will it provide? 2. Who or what will be its users? What might they use the functionality for? 3. What kind of data will your service create or accumulate? What will the data be used for? For this assignment, you do not need to present your features as user stories or use cases.

Part 3:

Write a few paragraphs that describe, at a high level, how you plan to test the functionality of your service without any clients, GUI or otherwise. (It's ok to use testing tools that have GUIs.) You will expand testing in the second iteration to include sample clients, but need to test stand-alone during the first iteration. Think in terms of testing your service as a whole, in addition to unit testing of individual subroutines, and answer these three questions: 1. How will you test that your service does what it is supposed to do and provides the intended functionality? 2. How will you check that your service does not behave badly if its clients use it in unintended ways or provide invalid inputs? 3. How will you test that your service handles its data the way its supposed to?

Submission instructions: One member of your team should submit a single file (preferably pdf) presenting your preliminary proposal. Your file must contain your team name and the names/unis of all team members. You may submit repeatedly until the deadline (note that if multiple team members submit, the most recent submission will override all previous submissions by the same or different team members).

Points	20
Submitting	a file upload
File Types	pdf, doc, docx, and txt

Due	For	Available from	Until
Oct 13, 2021	Everyone	Sep 1, 2021 at 12am	Oct 20, 2021 at 11:59pm

Team Assignment 1

You've already rated students with this rubric. Any major changes could affect their assessment results.

Criteria	Ratings		Pts
Part 1 1 point each for 1. team name, 2. names of all team members, 3. unis of all team members, 3. programming language 4. platform 5.Github repo	**5 to >0.0 pts Full Marks**	**0 pts No Marks**	5 pts
Part 2 2 points each for answering the 3 questions: 1. What will your service do? 2. Who or what will be its users? 3. What kind of data will your service create or accumulate? What will the data be used for?	**6 to >0.0 pts Full Marks**	**0 pts No Marks**	6 pts
Part 3 2 points each for answering 3 questions: 1. How will you test that your service does what it is supposed to do and provides the intended functionality? 2. How will you check that your service does not behave badly if its clients use it in unintended ways or provide invalid inputs? 3. How will you test that your service handles its data the way its supposed to?	**6 to >0.0 pts Full Marks**	**0 pts No Marks**	6 pts
Overview Project ideas meet the 3 requirements: 1. Maintain relevant data persistently 2. Supports multiple clients 3. No GUI	**3 to >0.0 pts Full Marks**	**0 pts No Marks**	3 pts
			Total Points: 20