

First Individual Assessment

✓ Published

👤 Assign To

✎ Edit

⋮

This is an individual assignment. You are NOT permitted to post, discuss or work together on the assessment questions with other students in the class or with anyone else other than the teaching staff. Your answers can be as long or as short as you want, but a long meandering answer that incidentally mentions something relevant will receive less credit than a short high-quality answer. Submit your answers all in a single document in response to this assignment on courseworks. You can submit as many times as you want until the deadline. Submissions via email, ed discussion, or any other means besides courseworks will not be graded and will receive zero credit.

Academic Honesty policy: All submissions will be automatically analyzed by turnitin.com for plagiarism/collusion. Plagiarism or collusion will result in zero credit on the assessment and referral to the relevant disciplinary committee.

Late policy: Submissions posted one day late (which includes everything from 1 minute late to 23 hours 59 minutes late) will receive maximum 75% credit; two days late maximum 50% credit; three days late maximum 25% credit; courseworks will not accept submissions more than three days late.

Maximum total 100 points. There is no credit for part 0, but 0A is required in the sense that we will not grade the graded questions if part 0A is missing or incomplete and you will receive zero credit for the assessment. Part 0B is optional and will not affect your grade.

Ungraded questions: part 0A is required and part 0B is optional.

part 0A (required): You have 100 total points to allocate among the members of your team. **Assign some number of points (possibly 0) to each member of your team based on what you believe was that student's relative participation in the team project thus far, and explain your scores.** Make sure the total is exactly 100. Remember to include yourself! For allocating points, consider level of effort and contributions for anything relevant to the team project. Note equal participation is not the same thing as equal lines of code produced. Please do not predict the future, consider only the contributions to the team project as of the date of writing. It is very important that you explain the scores you give to each student, including yourself: *If you do not provide meaningful explanations, then your response will be considered incomplete and you will receive 0 overall on the assessment.*

For example, in a team where all team members participated more or less equally, the scores would be: Cora 25, Kaedi 25, Kaylah 25, Nina 25, and your explanation would describe who did which part(s) of the team project effort thus far.

In a team with a substantial disparity in contributions, however, the scores might be: Iron Man 35, Captain America 35, Hulk 25, Thor 5, and your explanation might describe Iron Man and Captain America as doing most of the work, and Hulk helped, but Thor rarely showed up for team meetings and did not seem interested in the project. Again, your explanation should describe who did what, or who did not do what, as the case may be.

part 0B (optional): This is your opportunity to provide feedback about the course separate from the SEAS evaluations, which in the instructor's opinion do not ask the right questions. Is there anything that you think should have been covered already but wasn't? (Various topics will be covered later in the semester.) Is there anything that was covered unnecessarily or should have been put off until later? Is there anything that should have been presented in a different way? Is there anything else you would change about the course, including contents, teaching methods, delivery, assignments, anything else you think relevant? Please explain. Suggestions for the remainder of this course will be greatly appreciated.

Graded questions: There are two graded questions, part 1 and part 2, each worth 50 points.

Part 1 (50 points): Consider the service your team has proposed as your team project. Imagine the US federal government has imposed a new rule that all software services operating in the US provide “*special access*” for the NSA to monitor all activities of all client services and applications using the service. Maybe the NSA wants to know whether foreign nationals are using your service to infiltrate art schools or to trade photos of kittens (if information about art schools or photo sharing is supported by your service, respectively). Your service operates in the US and your small company has no means to fight this new rule. Removing your service from the US market is not an option for the purposes of this assessment, and the clients using your service are not able to

“opt out”. The article [here](https://www.reuters.com/article/us-usa-security-congress-insight/spy-agency-ducks-questions-about-back-doors-in-tech-products-idUSKBN27D1CS) (<https://www.reuters.com/article/us-usa-security-congress-insight/spy-agency-ducks-questions-about-back-doors-in-tech-products-idUSKBN27D1CS>) explains that such functionality is not as far-fetched as it sounds.

Think about what features you would need to add to your service to support the special access rule. Assume your development team will implement these features, and the NSA will not modify your codebase or its dependencies in any way. **Important note:** Do not write or modify any code for this assessment, just think about what you would need to do, do not actually do it! Include in your answer a link to your team project github repository but do not change anything in your team’s github repository in response to this question. Make sure your repo is public or, if private, the instructor and all IAs have collaborator access

To make this question more complicated as well as more realistic, a large company named Orange is fighting the special access rule in US courts. A US court judge might grant Orange a temporary suspension of the new rule. Such a suspension would also apply to all other companies affected by the rule, including yours. Thus the new version of your service should include a “*kill switch*” that can be instantly **activated** by administrators to immediately **turn off** all the special access features, even when their use is in progress. But since the suspension would be temporary, possibly ending at a specific time/date or possibly ending at an arbitrary time at the whim of a US court judge, administrators also need to be able to instantly **deactivate** the kill switch to immediately **turn on** all the special access features again. Another US court judge might later resume the suspension and yet another might end the suspension again. The US has a complicated court system, so the kill-switch may be activated and deactivated many times, turning special access off and on and off and on over and over.

part 1A (15 points): Discuss how your service’s *interface* for regular clients would need to change to support the special access and kill-switch features. Do any new API functions need to be added? Do any existing API functions need to be removed? Does the parameter list of any API function signature need to add parameters, remove parameters or change the type of parameters? Does the set of possible return values of any API function need to change (including status or error codes and raised exceptions)? Or if your service’s interface does not need to change to support the special access and kill-switch features, why not? For this question, you do not need to discuss any new API functions specifically for NSA access and/or kill-switch functionality, only those entry points intended for regular clients.

part 1B (15 points): Discuss how the *implementation* of your service would need to change to support the new special access and kill-switch features. Make sure you address the implementation changes needed to fulfill any interface changes discussed above as well as the new interfaces you envision for 1. NSA and 2. Controlling the kill-switch.

part 1C (20 points): Invent and describe what API testing your development team would conduct to check whether the special access and kill-switch features are working as intended. You should discuss specific tests that include calls to the API functions from part 1A to simulate activities by regular clients that NSA monitors when special access is turned on and loses access to when special access is turned off. Assume you have some easy way to check what NSA can “see” when special access is 1. turned on and 2. turned off, e.g., make up greybox tools like WhatCanNSASeeNow and CanNSASeeThisNow.

Part 2 (50 points): Imagine that your team project was required to develop a *library* instead of a *service*, with the intent that the library could be used by other developers as part of their own application or service and run inside the same operating system process rather than being accessed over the network.

Important note: Do not write or modify any code for this assessment, just think about what you would need to do, do not actually do it! Include in your answer a link to your team project github repository but do not change anything in your team’s github repository in response to this question. Make sure your repo is public or, if private, the instructor and all IAs have collaborator access

part 2a (15 points): What specific changes would be needed in your API interface and why? Make sure to consider all of your service’s entry points.

part 2b (15 points): What specific changes would be needed in the implementation of the persistent storage and multi-client aspects of your service and why?

part 2c (20 points): What are the specific changes that would be needed in the test suite for your service, considering both system and unit tests? Discuss every test case that would need to change in some way and explain how and why it would need to change. Be sure to consider all test cases in your codebase and all manual tests cases run by you and your teammates

This is the end of the assessment.

Points 100

Submitting a file upload

File Types pdf, doc, docx, and txt

| Due | For | Available from | Until |
|-------------|----------|------------------------|------------------------|
| Nov 4, 2022 | Everyone | Nov 1, 2022 at 12:01am | Nov 7, 2022 at 11:59pm |

First Assessment Rubric

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

| Criteria | Ratings | Pts |
|--|--|--------|
| <p>1a Discuss how your service's interface for regular clients would need to change to support the special access and kill-switch features.riterion (max 15 points)</p> <p>Addresses whether any new API functions need to be added. (Give full credit if not applicable.) (3 pts)</p> <ul style="list-style-type: none"> - Specify signature. (1 pt) - Specify return value (including status or error codes and raised exceptions). (1 pt) - Discuss all necessary new API functions that need to be added. (1 pt) <p>Addresses whether any API functions need to be removed. (Give full credit if not applicable.) (2 pts)</p> <p>Addresses whether any existing API functions need to be changed. (Give full credit if not applicable.) (10 pts)</p> <ul style="list-style-type: none"> - Discuss if the signature needs to change. (e.g. add, remove, change parameters.) If so, what API functions should be changed, and how? If not, why not? (5 pts) - Discuss if the return value needs to change (including status or error codes and raised exceptions.) If so, what API functions should be changed, and how? If not, why not? (5 pts) | <p>This area will be used by the assessor to leave comments related to this criterion.</p> | 15 pts |
| <p>1b: Discuss how the implementation of your service would need to change to support the new special access and kill-switch features. (max 15 points)</p> <p>Discuss implementation changes to activate the kill switch that is able to immediately turn off all the special access features. (5 pts)</p> <ul style="list-style-type: none"> - Discuss implementation for activating the switch. (3 pts) - The ability to activate immediately. (2 pts) <p>Discuss implementation changes to deactivate the kill switch that is able to immediately turn off all the special access features. (5 pts)</p> <ul style="list-style-type: none"> - Discuss implementation for deactivating the switch. (3 pts) - The ability to deactivate immediately. (2 pts) <p>API functions for regular clients, NSA, and kill switch controlling are all considered. (5 pts)</p> <ul style="list-style-type: none"> - Consider API functions for regular clients. (The API entry points in GitHub repository are referenced, and interface changes discussed in 1a are considered.) (2 pt) - Consider API functions specific to NSA. (1.5 pt) - Consider API functions specific to the kill switch. (1.5 pt) | <p>This area will be used by the assessor to leave comments related to this criterion.</p> | 15 pts |
| <p>1c: Invent and describe what API testing your development team would conduct to check whether the special access and kill-switch features are working as intended. (max 20 points)</p> <p>Discusses tests that include calls to the API functions from Part 1A to simulate activities by regular clients that NSA monitors when special access is turned on (10 pts)</p> <ul style="list-style-type: none"> - Tests what activities NSA should see for each regular API call. (4 pts) - Tests actions NSA should not be able to do for each regular API call. (e.g., call some regular clients' API functions that may change their data) (2 pts) - Tests regular API functions don't break or have behavior changes that are irrelevant | <p>This area will be used by the assessor to leave comments related to this criterion.</p> | 20 pts |

| Criteria | Ratings | Pts |
|---|--|--------|
| <p>to the kill switch and NSA access. (1 pts)</p> <ul style="list-style-type: none"> - All regular API function calls are covered. (2 pts) - Edge case tests are included and justified (1 pts) <p>Specific tests that include calls to the API functions from part 1A to simulate activities by regular clients that NSA monitors when special access is turned off (10 pts)</p> <ul style="list-style-type: none"> - Tests what activities NSA should not see for each regular API call. (4 pts) - Tests actions NSA should not be able to do for each regular API call. (e.g., call some regular clients API functions that may change regular user data) (2 pts) - Tests that regular API functions don't break or have behavior changes that are irrelevant to the kill function. (1 pt) - All regular API function calls are covered. (2 pt) - Edge case tests are included and justified (e.g., when special access user query with an invalid value) (1 pts) | | |
| <p>2a: Discuss what specific changes would be needed in your API interface for a library. (max 15 points)</p> <p>Discusses non-HTTP API layer. What interfaces are constructed, and what do requests and responses look like now? Are the API interface changes aligned with how libraries are typically used? (6 pts)</p> <p>Discusses moving away from any web-framework related dependencies for auth, session management, routing, error logging, etc. (5 pts)</p> <p>Discusses the shift from serving multiple direct clients to serving one direct client and possibly multiple indirect clients. (2 pts)</p> <p>Are all entry points considered? (2 pts)</p> | <p>This area will be used by the assessor to leave comments related to this criterion.</p> | 15 pts |
| <p>2b: Discuss what specific changes would be needed in the implementation of the persistent storage & multi-client aspects for a library (max 15 points)</p> <p>Discusses specific implementation changes to support multi-client (i.e. one direct client and multiple indirect clients.) (7 pts)</p> <p>Discusses specific implementation changes to support persistent storage: (8 pts)</p> <ul style="list-style-type: none"> - Discuss details of how clients would configure their database now that they own it. (e.g. the ability for a library user to select their own supported drivers/connections, etc) (4 pts) - Discusses details of how clients would set up their database. (e.g., if there are any pre-seeded migrations or something to be run) (4 pts) | <p>This area will be used by the assessor to leave comments related to this criterion.</p> | 15 pts |
| <p>2c: Discuss specific changes for system and unit tests for a library. (max 20 points)</p> <p>Considers system test changes (10 pts):</p> <ul style="list-style-type: none"> - Considers test setup, tear down, and/or fixture changes if applicable. (2 pts) - Explain how and why it would need to change. (6 pts) - Are all test cases considered? (2 pts) <p>Considers unit test cases (10 pts):</p> | <p>This area will be used by the assessor to leave comments related to this criterion.</p> | 20 pts |

| Criteria | Ratings | Pts |
|--|---------|-----|
| <ul style="list-style-type: none">- Considers test setup, tear down, and/or fixture changes if applicable. (2 pts)- Explain how and why it would need to change. (6 pts)- Are all test cases considered? (2 pts) | | |
| Total Points: 100 | | |