Maximum total 100 points. Part 1 is 50 points and part 2 is 50 points.

Part 1 50 points

1A 25 points

There should not be any UML use case diagrams, particularly since the use cases lecture notes (October 8th) states explicitly "if you are asked for a use case and you submit a use case diagram, *you are doing it wrong"*. If a student submits a use case diagram, that may be evidence that the student googled for something like "tic tac toe use cases" and copy/paste. Ask Prof. Kaiser to look at it.

8 points. Is there more than one use case? These should correspond more-or-less to New Game, Start Game, Join Game, Move (2 points each), which are documented in assignment I1. There does not need to be an exact match, e.g., it's ok if New and Start Game are combined (so that's 4 points) or there's an extra case like End Game (no extra points).

3 points. Does every use case have at least title, actor, description and basic flow? (Take off 1 point if any of these parts is missing from any of the use cases, but don't take off more than 3 points total = don't go negative.)

4 points. Is every step in the basic flow labeled? For example, 1, 2, 3, etc. (2 points) Is every alternative flow labeled as to where it diverges from the basic flow? (2 points) For example, 2A, 2B and 2C are three different alternatives for step 2 in the basic flow. If an alternative has multiple steps itself, these should also be labeled, e.g., 2A1, 2A2, 2A3, etc. (but do not consider this in grade).

6 points. Are there alternative flows corresponding to the likely player errors? There are three error situations listed in 5.1 Test Cases of assignment I1 that should be covered. That is, player 1 might try to make a move before player 2 has joined the game (2 points), player 2 might try to make the first move (2 points), one of the players might try to make two moves in a row (2 points). Another alternative flow possibility is timeout, where it's a player's turn and nothing happens for some period of time, but the assignment did not address timeout so it's fine to omit.

4 points. Are there basic and alternative flows corresponding to the possible end games? Assuming the basic flow ends in one player winning the game, there should be an alternative flow that ends in a draw. OR the basic flow could end in a draw, in which case winning would be an alternative flow. Either way is ok,or some other approach, but both cases should be covered somehow (2 points each).

1B 15 points

The purpose of this part of the question is to try to ensure that the use cases and their flows reported for 1A indeed match the student's own code, and are not written from the assignment description or copied from somewhere. You do not need to read each student's entire codebase (as given in the most recent version submitted via courseworks). Instead, make a few spot checks that the method(s) stated as corresponding to the xxx part of the yyy use case indeed

seem to do so. The details do not need to be perfect. Take off the full 15 points if you find significant discrepancies and ask Shirish to look at it.

1C 10 points

This is intended to be "free points". Give the full 10 points for any non-empty answer that says anything plausible, including something like "I would have dropped the course."

Part 2 50 points

2A 25 points

Take a look at the Google Soli information at https://atap.google.com/soli/. This chip is a miniature radar that recognizes motions and movements. You do not need to know the details of what it does or how it works. Treat the Soli chip as magic and accept any functionality that students invent for it, as long as it does not include camera/microphone capabilities and does not include AI to itself answer questions like "is the user of this laptop paying attention?", which is explicitly disallowed. If someone does this anyway, ask Prof. Kaiser to look at it.

Take a look at a few of the endpoints for the Zoom API at https://marketplace.zoom.us/docs/api-reference/zoom-api (expand a couple of the categories and look at a few pages from the menu listed below "ZOOM API" on the left hand side). All of the entries are of the form REST METHOD with a /path/to/somewhere endpoint. The REST METHOD is one of GET, POST, PUT, PATCH, DELETE. There might be embedded parameters like /path/to/{parameter1}/something/{parameter2}/somewhere in the endpoint.

Besides the REST METHOD and endpoint path, each API page has a brief prose description saying what it does, prerequisites, scopes, rate limit label, authorization, request parameters, request body, responses and test request. The students do not need to include prerequisites, scopes, rate limit label, authorization or test request. The students only need to invent at least two API methods with a REST METHOD, endpoint, description, parameters (if any), body (if any), and responses. It's fine if there are no parameters or body for methods that don't seem to need any.

4 points. Are there at least two distinct methods defined for the API? (2 points each)

4 points. Does every API method have a legit REST-METHOD, i.e., one of GET, PUT, POST, PATCH, DELETE? Take off 2 points for every API method that does not include one of these REST methods (but don't take off more than 4 points total). I'm expecting mostly GETs.

4 points. Does every API method have an endpoint path? Take off 2 points for every API method that does not include an endpoint path (but don't take off more than 4 points total).

4 points. Does every API method have a prose description that sounds relevant to getting data from the user's laptop? This should be from the Soli chip, but it's ok if the keyboard, mouse, touchpad, or similar are used, as long as the camera and microphone are not used. Take off 2 points for every API method that does not include an appropriate prose description (but don't take off more than 4 points total). I do not expect students to invent JSON or other formatted data, just prose, but if they do that's fine.

4 points. Does every API method have one or more HTTP responses? These will be something like 200 OK or 404 Not Found. Take off 2 points for every API method that does not include an HTTP response (but don't take off more than 4 points total). I do not expect students to invent JSON or other formatted data, just prose, but if they do that's fine.

5 points. Give full credit (i.e., these 5 points, not necessarily 25 points) for any discussion of the API that seems useful for detecting when the Zoom/Soli laptop user is not paying attention, and zero otherwise.

2B 15 points

4 points. Are there at least two distinct user stories? (2 points each)

8 points. Does every user story have conditions of satisfaction that seem plausible? Take off 4 points for every user story that does not include conditions of satisfaction (but don't take off more than 8 points total).

3 points. Give full credit (i.e., these 3 points, not necessarily 15 points) for any set of user stories that seem useful for checking whether there are any meeting participants who are not paying attention, and zero otherwise.

2C 5 points

Give full credit for any answer that seems to understand what acceptance testing means for a REST API, and zero otherwise. For a REST API, the minimum acceptance criteria are that every API method is tested with the corresponding parameter(s) and body (if any) and returns the appropriate HTTP response.

2D 5 points

Give full credit for any answer that seems to understand what acceptance testing means for a set of user stories, and zero otherwise. For user stories, the minimum acceptance criteria are that every condition of satisfaction is tested and each condition is indeed satisfied.