CS 240: Chess Server Design Phase 2 Transcript

[00:00:00] So we've, we've talked a lot about the chess server design, which you're going to implement in phase three of the chess project.

[00:00:10] Now, I want to talk about phase two of the chess project, which obviously comes before phase three.

Start visual description. The professor demonstrates the initial phase of the chess project, focusing on creating design documentation. End visual description.

[00:00:16] Phase two of the project is to create some design documentation and turn it in.

[00:00:22] So the TAS can verify that you actually understand the chess server design and how it works.

[00:00:28] So we went through in previous discussion about um about the design with uh we went through an architecture diagram, we talked about model classes, data access classes, service classes, handler classes and so on and so forth um and try to describe what all those pieces do and what they're for.

Start visual description. The professor shows an architecture diagram, explaining model classes, data access classes, service classes, and handler classes. End visual description.

[00:00:51] And now it's going to be your Task in phase three to implement all those things.

[00:00:55] So in phase two, we'd like you to create some design diagrams that, that uh show us that you understand what all those pieces are and how they call each other at run time to, to make the chess server work.

[00:01:12] So one of the main reasons we have the phase two assignment is number one to inspire you to dive deeply into the phase three specification.

Start visual description. The professor emphasizes the importance of reading the phase three specification in detail to understand the chess server design. End visual description.

[00:01:22] So in order to do phase two, you're going to need to read phase three specification in detail and probably read it multiple times to extract all the information from it.

[00:01:34] And so that's the first thing you should do uh when working on phase two is just make sure you've read the phase three spec in detail.

[00:01:43] And that's the most important thing you can do.

[00:01:46] And then uh once you think you understand all that stuff, and here's another rendition of the, the architecture diagram that we looked at.

[00:01:55] Once you think you understand those pieces pretty well.

[00:01:59] And here's another uh description of what those pieces are in case you need another uh explanation and you've already read the face three spec, and you understand the web API and you understand all the different parts and pieces.

[00:02:17] The actual assignment for phase three is to create several uh sequence diagrams that um capture your understanding of the phase three architecture.

Start visual description. The professor introduces sequence diagrams, explaining their purpose and showing an example of a sequence diagram for user registration. End visual description.

[00:02:28] Now, what is a sequence diagram? Um It's very likely you, you do not know what that is.

[00:02:33] And so I'm going to um first explain what a sequence diagram is and then I'll explain the diagrams that we are asking you to create.

[00:02:43] So um down here below, there is an example of a sequence diagram and the purpose of a sequence diagram I it's basically a picture as a diagram is and it shows all the different objects that participate in the execution of the server.

[00:03:00] And so along the top here, you have um these boxes and, and icons that represent the different um parts of the of the uh interaction. So, the first one here is what's called an actor.

Start visual description. The professor explains the components of a sequence diagram, including actors, server objects, handler objects, service objects, data access objects, and databases. End visual description.

[00:03:13] This is just a user; this is an actual person.

[00:03:15] They're not a software entity, that's an actual person.

[00:03:18] And then we have various objects that are involved. So, we have a server object, we have a handler object, we have a service object, we have a, a data access object, then we have a database here on the end.

[00:03:32] Now these boxes here, they, they don't represent classes of objects.

[00:03:35] These are individual objects at runtime. So, this is a runtime picture of how your server operates.

[00:03:41] So um this particular diagram, let's zoom in a little bit.

[00:03:46] It demonstrates what happens when a user registers with the chess server.

[00:03:54] And so it it's kind of small here, but you can see it says registrations open the top left corner.

[00:03:58] This is the diagram that shows the interactions of registration.

[00:04:03] And so um underneath each of these objects at the top, you see that there's a dash line that comes out of the object that's a timeline.

Start visual description. The professor demonstrates the flow of information and method calls in a sequence diagram, showing how objects interact at runtime during user registration. End visual description.

[00:04:12] So time zero is at the top of the diagram and then time flows downward.

[00:04:17] So at the beginning of this interaction, we'll assume all these objects already exist.

[00:04:23] So we're not trying to show how those objects got created in this case.

[00:04:26] But when somebody registers, what happens? Well, these objects are going to call methods on each other.

[00:04:33] And that's what these arrows represent between the dashed lines is each arrow represents a method call or a function call.

[00:04:39] And so what this means is the first thing that happens when a user register is the client. It's actually the client software in this case.

[00:04:47] But the client software is going to send a request to the server.

[00:04:52] So they're going to call the um register end point.

[00:04:57] They're going to pass in a JSON object that has the username, the password and the email address.

[00:05:02] That's what that, that arrow represents. It's really just a method call this this object over here is calling this object over here.

[00:05:10] And here's the function that's being called and here's the inputs.

[00:05:15] And then the next thing that happens and you'll see that this is a little bit lower down in time.

[00:05:19] That's why we know this is first and this is second.

[00:05:23] And the second thing that happens is the server is going to delegate that register request to a handler.

[00:05:29] Now, in this case, it would be the register handler.

[00:05:34] OK? So, the, the server delegates the register request to the register handler.

[00:05:40] And then what does the handler do? It's going to then call the register method on the service, the user service probably in this case.

[00:05:49] And then what does the user service do? Well, it's going to going to call the database to check to see if that username is already taken.

[00:05:59] But if uh that query comes back on that means that this username is available.

[00:06:04] And so then the next thing the user service is going to do is it's going to ask the database to create um the user object.

[00:06:12] And the, and the data access class will actually insert that, that new user object into the database.

[00:06:19] And then um the user service is going to ask the off token dow to, to go ahead and create an off token for the user which uh gets inserted into the database and then gets returned.

[00:06:34] Now, you can see here that these arrows going in the other direction, the dashed arrows represent return values from an operation.

[00:06:40] So the off token comes back, the user service is going to create a registered result which then gets passed back to the handler um which is then going to encode that register result as JSON and then it will pass that back to the client.

[00:06:58] So it's kind of just shows the flow of information, it shows the flow of method calls, it shows the interactions between the objects at run time.

[00:07:06] But what this really does is um it, it forces you to understand, OK, not only what are these objects but how do they interact to actually make the system go? And so that's, that's an example of a sequence diagram.

[00:07:23] And so we actually provide that um as an example and the assignment, we call this the starter diagram.

[00:07:32] And so if you, if you click on the starter diagram, it'll take you to a website that's really cool and that it's a, it provides a nice tool for creating sequence diagrams.

Start visual description. The professor explains how to use the sequence diagram tool, showing how to create and edit diagrams using textual descriptions. End visual description.

[00:07:44] And so what you'll see here is on the right-hand side is the, the diagram that we created.

[00:07:50] But the way we created this diagram was not using a drawing tool of some sort.

[00:07:54] What you actually do is you write some text; you write a textual description of the diagram over here on the left side and then the tool will automatically create the picture for you.

[00:08:05] Like at the very top here, it says actor client.

[00:08:09] Well, that tells them to put a person icon up there that's named client, participant server.

[00:08:15] Well, participant is just an object. So, it's a box and it has the name server.

[00:08:20] And so we add the participants and then at the end, we have a database named DB. And so that's how those objects across the top got um created.

[00:08:29] Now, I could create another um participant.

[00:08:32] If I wanted to add somebody else, I could say participant, Bob.

[00:08:39] And so you see how Bob just shows up there.

[00:08:42] Now, we don't want to keep Bob, but um we'll delete him, but you can see how editing the text over here updates the picture as well and then having created all the objects uh across the top, then you can um describe all the method calls between the different objects.

[00:09:00] And so that's how these different arrows got created.

[00:09:02] And you can look at the syntax for that.

[00:09:04] It's, it's quite straightforward and obvious.

[00:09:07] And so that's all it took to create that, that diagram was just that little bit of, of text right there.

[00:09:12] And so we're going to ask you to use this tool. But what we're going to ask you to do is to fill in and create a sequence diagram for all of the other six web API functions that a client could call.

[00:09:28] So we're going to ask you to fill in log in, log out uh list games, create game, join, game and, and clear.

Start visual description. The professor outlines the assignment, asking students to create sequence diagrams for various web API functions, including login, logout, list games, create game, join game, and clear. End visual description.

[00:09:40] And so we, we've already created uh like placeholders for those other six diagrams.

[00:09:45] So you can see here they're called groups.

[00:09:48] So each, each diagram is a group.

[00:09:53] And so we've created empty groups for all the other six functions.

[00:09:58] And we're going to ask you to fill in the details for all of those and explain how do these objects interact with each other at run time to actually make those functions uh work.

[00:10:11] OK.

[00:10:13] So that's the assignment.

[00:10:14] And the example diagram we provide is going to be like gold, it's going to really help your um gets this done.

[00:10:25] Now, the next thing we look at is I had to turn this in.

[00:10:34] So if we go back to the phase two specification at the bottom here, it talks about um the deliverable for this.

[00:10:47] So one thing you can do in the sequence diagram.org website where you're creating the diagram, that's the name of the tool sequence diagram.org, they have an export feature.

[00:10:58] So having created your diagrams, then you can click the export icon on the left-hand side.

[00:11:04] And what you can do is you can save your diagrams to um a link that's, that's hosted on the sequence diagram.org server.

[00:11:14] So when you export your diagram, they'll basically save your diagram in a file and then they'll give you a URL for it or a link URL that you can then turn into the T A.

[00:11:23] So the thing you're actually going to turn in um on canvas is um the URL for your diagram on sequence diagram.org.

[00:11:32] So just uh create that link, submit it on the campus and then you're good. That's all you got to do to turn it in.

[00:11:42] Now, lastly, I want to talk about how they grade they will grade your design.

[00:11:47] The whole purpose of this uh phase two exercise is to just help you understand what you're building in phase three.

[00:11:53] So the way this is graded is based on how well you seem to understand what you're building in phase three. How well do you understand the server design? So, um there's three different scores that you can uh get when the TAS initially grade your design, there's three buckets that they'll divide the designs into.

[00:12:13] So the assignment is worth 50 points.

[00:12:17] So what the TAS will do is they'll look at your design and that if they think that you understand pretty well, what, what's going on, there might be some minor errors in your diagrams but nothing too major.

[00:12:28] Um You'll get full credit and uh the TAS may reply to you on canvas and explain, you know, some problems that you may have in your design diagrams, try to help clarify your understanding of the assignment.

[00:12:41] But if it's mostly correct, they'll just give you full credit and move on.

[00:12:45] Um If your design has some major difficulties, there's some major misunderstandings reflected in your diagrams.

[00:12:52] Um What the TAS will do is they will give you a uh a 25 which is uh half credit basically.

[00:13:00] And they will invite you to come talk to the T A that graded your design um so that they can explain to you the problems that they see in your diagrams.

[00:13:13] And this is a service to you.

[00:13:14] This is a benefit because uh what they're trying to do is help you get on a, get off to a good start on the project.

[00:13:20] So if you'll come talk to the T A and, and work through the issues with them, then we'll give you a week to resubmit um or maybe half a week in a, in a term, but we'll give you some period of time to resubmit your design.

[00:13:35] And if you fix the problems that were identified, then you can get all the points back and you'll get 100% or 50 points as well.

[00:13:44] The third grade you can get, um, is if, if we feel like your uh submission was not a serious attempt at the assignment, but maybe just an attempt at getting points, hopefully, um, what we'll do is we'll give you a zero on that.

[00:14:00] So if it's not a serious effort, then we'll just give you no credit and we'll still give you a week or a half a week depending on whether you're in a semester or term to resubmit.

[00:14:10] So in that case, if you do the design and make a good faith effort to do it and resubmit it, you can get up to 25 points for half credit.

[00:14:18] And so that's how it's created.

[00:14:20] So good luck and if you need help, talk to the TAS or your instructor.