Q1		20
Q2		20
Q3		20
Q4		20
Q5		20
TOT		100

Name: Jacob Chlebowski

CS509 Final Exam F24

Q1 [20 pts.] Write the corresponding Use Case for the following proposed change to the Auction House System.

The Seller wants to make sure they set the initial price for an item so it is properly valued, based on other similar items for auction:

Before activating an item, a Seller can change the price of that item to be the same as the highest bid for any current active item with the same name. (Republished)

use case: Change Price of Item

Participating Actor: Initiated by Seller

Entry Condition: Item not activated and other current active item exists with a bid (with same name)

Exit criteria: Price on unactivated item changed to be same as highest bid from activated item with same name, OR nothing happens

Flow of Events:

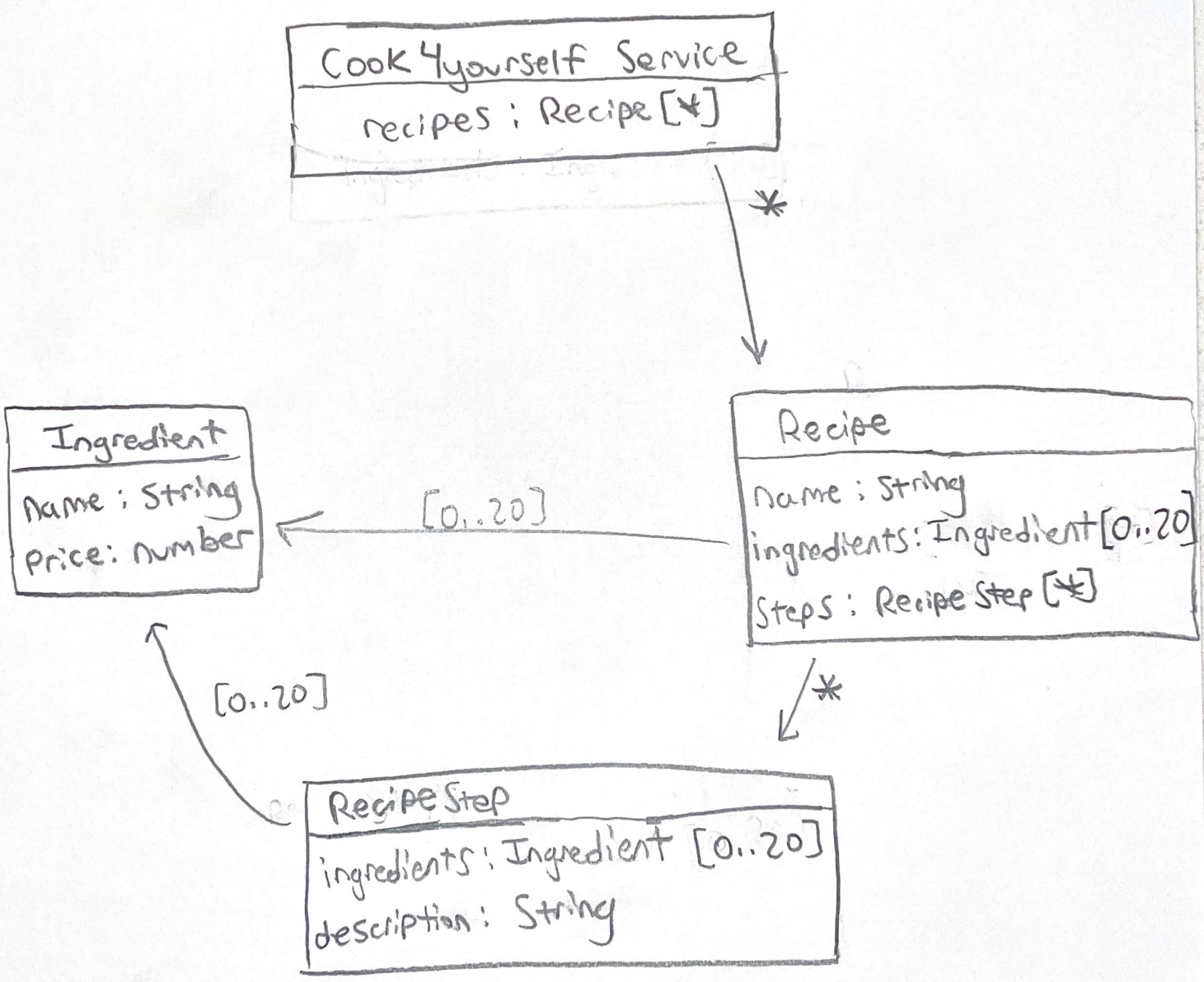
1) Seller requests to change price of an inactive item to be same as highest bid from an active item with same name

2) AuctionHouseApp updates price of inactive item and refreshes display

Q2 [20 pts.] Given the following application domain, construct a UML class analysis diagram (including attributes and relationships between classes) for the following context:

You run **Cook4yourself**, *a service for people to cook their own food at home.* **Cook4yourself** *has a list of recipes that you can make from a list of 20 common ingredients. Each recipe has a name and lists the ingredients that it uses. Each ingredient has a name and a price. A recipe contains a numbered sequence of steps to perform. Each step records the ingredients it uses as well as a description of the action to perform in that step.*

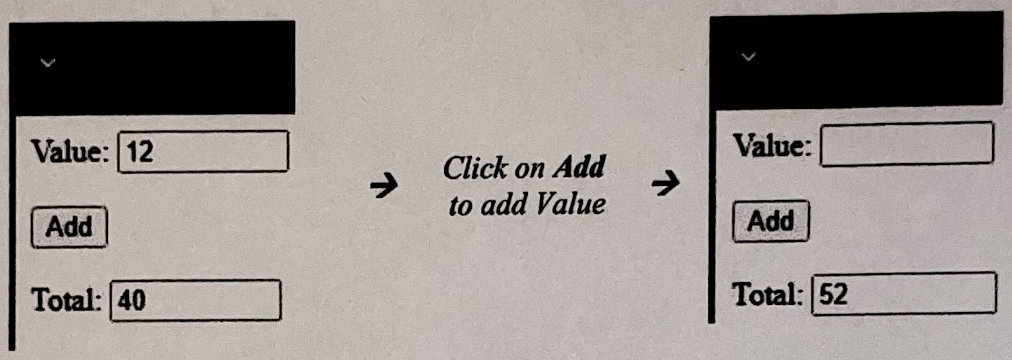
You must show attributes and relationships between classes. Do not show methods or constructors.



Q3 [20 pts.] Given the following problem, show the UML class diagram for the appropriate React **Entity, Boundary** and **Controller** classes that you would use. This application has one use case. The user can add the numeric value in the "Value" text field, and the accumulated total appears in the "Total" text field:

- Add Value

There is no need to show relationships (i.e., arrows) between the classes.



Use proper UML class diagrams to capture the attributes and methods.

Entity

Model

totalSum: number

Controller

AddValueController

addValue(~model, Model, value: number): void

Boundary

AddValueApp

value : Input
total : label
add : Button
~model : Model

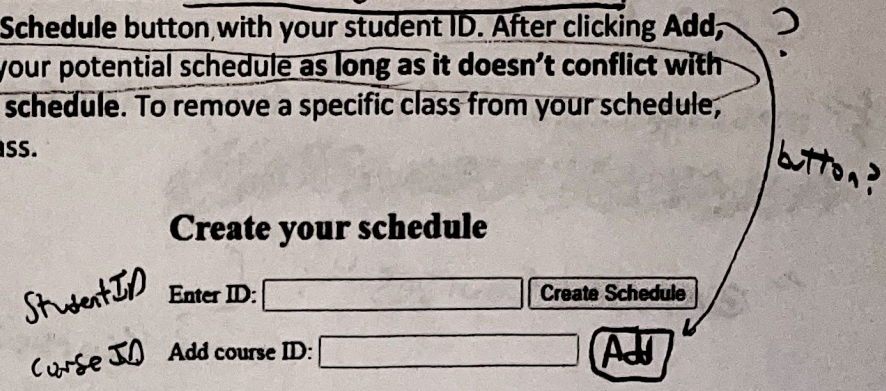
addValue(~model, value)

Q4 [20 pts.] You want to create a schedule of classes whose meeting times do not overlap. To create a new schedule, click the **Create Schedule** button with your student ID. After clicking **Add**, the class ID you've entered is added to your potential schedule **as long as it doesn't conflict with the meeting times for any class in your schedule**. To remove a specific class from your schedule, click the **Remove** button next to that class.

Use Cases

Here are the use cases:

- Create Schedule
- Add Class
- Remove Class



/create Schedule

POST { "student ID" : "student-id" }

Response

200 success

{ "message Success" : "Successfully created schedule!" }

400 failure

{ "message Error" : "failed to create Schedule" }

/add Class

POST { "course ID" : "course-id" , "student ID" : "student-id" }

over

/addClass Continued

Response

200 Success

{"messageSuccess": "successfully added course"}

"courseID": "course-id"

"enrolledCourses": [

{"courseID": "course-id",

"courseName": "course-name",

"courseTime": "course-time"

},

{...}

]

}

400 failure

{"messageError": "failed to add course, conflicts with other class time"}

/removeClass

POST {"courseID": "course-id", "StudentID": "student-id"}

Response

200 success

Same as above

{"messageSuccess": "successfully removed course",

"enrolledCourses": [{...}]

}

400 failure

{"messageError": "failed to remove course from your course Schedule"}

Q5 [20 pts.] These questions are concerned with AWS. There are SIX possible questions, and you only need to answer **FOUR**. Note that I will only grade the first FOUR of your answers, so do not answer more than FOUR. **Identify the FOUR questions that you have chosen to answer.**

(a) Describe the steps you performed to host your React application in your S3 bucket.
(b) After you created your database in RDS, describe the steps you performed so you could create your first schema in the database.
(c) Describe the steps you performed to enable a lambda function to access information from the RDS.
(d) Explain the purpose of a stage in the API gateway.
(e) Explain how to pass parameters to a lambda function using an HTTP GET request.
(f) Explain how a lambda function is able to store image files in your S3 bucket.

(a) First locally, I "npm run build" to gather an /out file with my html pages and /next modules. Then, on AWS S3, I make sure I upload all of the contents of the /out folder so that the html is updated while also ensuring all pages are publicly accessible

(c) First add lambda function. Ensure config.json has pool query host names & password to access database. Then, go to API gateway and add the resource method of lambda function and enable CORS. Then, re-deploy API.

(d) An API gateway stage contains a list of resource methods (and lambda functions) so far for the current snapshot of the site.

over

(f) Lambda functions can store image files in the S3 bucket by properly encoding image in base64 format as a string. Then, when displaying the image on the front end, decode the base64 to image format