

Communication Establishment Peer Review

- Done by Jesus Cerda

General Feedback

- I believe there are some major issues with the current design of the IChat interface object
 - The way the IChat is designed, each IChat object appears to be a single object associated with one single message, linked together to other messages by their `sessionID`
 - This design may lead to confusion or be harder to handle. For example, in CE-2 where you describe sending an IChat object, you are sending all the values of that individual message, without verifying that the message being sent is going to the right conversation.
 - It could be possible that the user manually modifies the `sessionID` value to send a message to someone they are not supposed to, or maybe to a session that does not exist at all, and that could lead to a whole new set of errors
 - If two chats are created simultaneously, there is a chance they can receive the same `sessionID`
 - There is no way to check what `sessionID` values are valid, there is no constraints to what chats can have what `sessionID`
 - A better approach may be to potentially create an `IChat` object, which represents one session, and an `IMessage` object, which represents the entire chat session
 - If we had a list of chat sessions, each of which contained a list of messages, it would also make it easier to figure out how many chats there are, rather than trying to figure it out by checking every `sessionID` value in each of the messages.
 - Because chats can exist without any messages being sent yet, this also gives the option of opening a chat without sending a message yet.
 - It also makes it possible to use serial values for the `sessionID`, so we can automatically assign the next available `sessionID` value
 - It also allows for input validation, as we can check if the `sessionID` exists before adding a chat to that value
- I had some confusion initially, as at the top (when showing the function you are calling), you state that the entry point of the browser returns a Response, but at the bottom, it shows you are either returning just the Status code or the Status code with the error from the response

CE-1 (Authenticated users can open a chat session with the seller or renter of the car that they are interested in)

Pass

- For all your "browser to entry point" functions, I think it would be a better idea to write a "Fetch" function, with the url to the specific endpoint that you are using and potentially the body of the function as well, rather than a generic function, which I presume is the javascript that is going to be done in the front end
 - As for the function you wrote, it might be better to clarify that as "View to Script" rather than "Browser to entry point", for the sake of clarity

- I don't think passing in `searchParameters` is necessary, as to find the vehicle owner we would just need the vin number. Adding Search parameters feels like extra overhead that may not be needed
- There should probably be a check to see if the user is allowed to talk to the user they are trying to contact
 - This can possibly be done by making a claim such as `hasBlockedUser: BuyerUID` and associating that with the Buyer
 - This is to say, The buyer can have a list of users they have blocked, so that they cannot contact them prior to messaging them
 - It can also be implemented the other way (the user can have a list of people they are not allowed to message) depending on which would be easier to implement or more secure
- As mentioned earlier in my "General Feedback" section, you don't have a proper way to determine the `sessionId`
 - Because it is not unique (same `sessionId` across multiple chat objects), you cannot use a serial to auto-increment it, or to automatically assign one to a new chat
 - Because it cannot be unique, I cannot come up with a way to make sure the input is valid, so there is no way to implement input validation

FailDatabase

- Small issues, such as typo on `CE-1 Fail Browser to Entry Point` for the word `Response`
- There is potential to further clarify what type of error has occurred, depending on what type of error the database throws
 - The `503 Service Unavailable` could be used if the database is offline, as this may contain details such as how soon it can be retried
 - `503` could be if the database is potentially offline, or if the backend server is not responding
 - The `400 Bad Request` could be used if the error was caused by something along the lines of the user requesting to talk to someone that doesn't exist, or some other bad input that is the fault of the user
 - The `500 Internal Server Error` error code could be used as a catch all, as is shown in your design

CE-2 (As an authenticated user, I can see all the chat sessions that I have opened or I have responded to)

Pass

- To some degree, I question the need to continue passing down the entire `User Account` from the manager layer down, as
- I have similar critiques here about the usage of the `SearchParameters` type here, as from the last user story
- I would be curious to see or hear any details about the front end design, such as how the chats would be displayed, or how they would be organized
 - Other things, such as the front end function are missing as well. I am assuming there will be a list of chats that you can select, and I am assuming that the `IChat` object will contain all the messages present in the chat"
- Although moreso a critique about the organization of the design document and not the design itself, I believe that sending a message would likely fall under the scope of User Story CE-1, not user story CE-

2, as CE-1 seems to pertain to creating and sending "Chat sessions" while user story 2 seems to be pertaining to the retrieval of chats that have already been in use

- It would be a good idea to Create Claim that checks if they are allowed to send a message into a specific chat. This could be added to their set of permissions upon the moment of chat creation (ie CE-1), and be used to validate that the values we are receiving from the front end are actually genuine, and they are not trying to send messages into chats they are not part of.

Fail Database

- No serious issues, as long as the front end handles this gracefully, and maybe shows the user what exception occurred

CE-3 (As an authenticated user, I can confirm deals with any buyers, this will change the status of my vehicle profile on the marketplace from "Available" to "Sold" and provide the buyers with my location as well as guiding them. I can also cancel any request or deals that are ongoing and change the status of my vehicle profile post to "Available".)

Pass Confirm Deal

- For the confirm deal, I think that the first part where it sets the status is fine, but I believe that the location message as shown here does not look well implemented
 - There is no details onto what an `IMessage` is
 - If we are just sending a message separate from the chat, it might be better to just use our emailing service, which is already implemented

Pass Cancel Deal

- Although I'm sure this is not how it will be implemented, as designed this skips authentication entirely

Database Fail

- No major issues, as long as it lets the client know in the front end that the value was not successfully changed