Feature: Donate Your Car

Feature Developer: Giovanni Contreras

Date Review Submitted: 03/09/2024

Peer Reviewer: Vi Nguyen

Date Review Completed: 03/30/2024

Major Positives

1. ***Reusability***

   Good reusability with the design by making objects like ICharity which can be used with any different charity organizations object from the database. Also, the use of IResponse object as return value makes your services easy to integrate to the application.

2. ***Clear distinction for the flow of design***

   The small descriptions under each function call help the developer understand what each function call is supposed to do and who should call it. Also most of the design follow SOLID principles and meet most of requirements in the BRD.

Major Negatives

1. ***The fail scenarios are incomplete or missing***

   Missing fail scenarios can slow down development because developers would need to refer to other documents to understand how a function should behave in case of failure. Additionally, testing could be hindered as developers would have to cross-reference the BRD to verify if failures are genuine or false positives.

2. ***Ambiguity***

   At the moment, there is no table in the ER Diagram to illustrate the properties of "Charity" table that the DYC-1 Sequence diagram references. Also, without the class diagram to show me the relationship between classes and attributes/properties of objects, hence, at the moment, any reference to any new object type in the sequence diagram is a

little bit ambiguous regarding what they contain, or what their parameters are or what they can return.

3. ***Sequence Diagram***

   Some of the lifelines in the sequence diagram are incorrect, for example, in DYC-2, there is no lifeline at all for the Manager and Service layer, hence, the design does not reflect when the layer is active and when it is not. Moreover, The use of dash lines instead of solid lines to activate the methods calling CreateAsyncLog in DYC-2 is a minor mistake since the dash lines are for return messages to the message caller. The user should also has its own lifeline.

Unmet Requirements

1. ***BRD Requirement***

   - For DYC-1 did not check for preconditions whether user have a car to start listing for charities.

Design Recommendations

1. ***Scalability and Reusability***

   Even though it is stated in the BRD about a fixed number of charity organizations that we can support, for scalability purposes, I can recommend making a function to add and delete charity organizations in case we need to add more organizations or remove some.

2. ***Fail scenarios***

   I recommend adding more fail scenarios to the sequence diagrams according to the BRD so you know how to handle errors when they occur, also it's helpful for teammates to understand how you handle those errors and if needed they can help with the debug process.

3. ***Pagination***

   Take into account pagination, what if the list of charity organizations grows to more than 50, then you might need to retrieve and display only 10 at a time.

4. ***Supporting Document***

   I recommend making class diagrams and updating the ER diagram to provide a clearer understanding of the objects, methods, and table that you are going to use according to your design. Also, a site map would be helpful in helping me understand how the feature will behave in the front end.

Test Recommendations

1. ***Functional and Non-Functional Requirements***

   Stress test your feature to see if the information is correctly retrieved from the database and whether the operations are completed under the stated time period in BRD of 3 seconds. I also recommend end-to-end testing since your feature needs to be able to fill out information in the front end fast and accurate enough so that it does not affect user experience.

2. Front-end Testing

Due to the objective of filling out information in the front end, I recommend front-end testing to see if the provided information by the DYC2 is correct as well as whether the information of the 4 charity organizations we have is correctly displayed to user in DYC view. Also please provide more information about the “Script” mentioned in the sequence diagram as to how the feature will fill out the pages for the user.