

Correction Criteria for the experiment:

Subtask 1:

1. The first question consists of allocating the class responsible for initiating the use case and allocating the corresponding method. The correct answer is the following:
 - a. MessageCompose is the correct class. 1 point.
 - b. createMessageBuilder () is the correct method. 1 point.

The scores of the first question is calculated as the following:

- a. The participant gets one point for allocating the MessageCompose class
 - b. The participant gets one point for allocating the createMessageBuilder ().
 - c. The participant does not get any points otherwise.
2. In the second question, the participants are asked to create a sequence diagram. For the sequence diagram. We have created a ground truth sequence diagram. The answers to the second question in sub-task 1 are corrected according to our ground truth. More specifically, the scores of the sequence diagram is are calculated as the following:
 - a. The participant gets one point for each correct provided component.
 - b. For each correct message between two objects, the participant gets one point

Finally, we add the score gained in question 1 and score gained in question 2 to calculate the overall score of sub-task 1.

Sub-task 2:

Subtask 2 consists of 6 multiple choice questions. First, we list the questions of sub-task 2 and corresponding right answers:

1. The LocalFolder class is responsible for:
 - a. Holding information about itself (e.g. information about the folder config, databseId, name, visibleLimit...) and offering it to other objects on demand (e.g. MessageCompose calling getters methods).
 - b. Holding information about other objects and offering it to external objects on demand.
 - c. Managing persistence features such as serialization and storing to and fetching from the database (e.g. save(Preferences preferences) method that saves the account preferences).
 - d. Both a and c are correct. **This is the correct answer.**
2. The FolderList class is responsible for:
 - a. Initiating GUI objects and delegating user requests to other objects (e.g. initializeActionBar()). **This is the correct answer.**
 - b. Holding information about users activities and providing it to other objects on demand.
 - c. Collaborating with other objects and implementing interfaces.
3. The MessageCompose class is responsible for:
 - a. Collaborate with other objects to make decisions and control tasks.
 - b. Control tasks (e.g. createMessageBuilder()) and make decisions (e.g. sending a message) depending on information fetched from other objects or invoked events (onMessageBuildSuccess). **This is the correct answer.**
 - c. both a and b.

4. The Message class is responsible for:
 - a. Holding information about other objects and providing it to external objects on demand (e.g. Holding Date mInternalDate, Folder mFolder, Set <Flag> mflag, String mUid).
 - b. Holding information about itself (e.g. String mUid) and providing it to external objects on demand.
 - c. Both a and b are correct. **This is the correct answer.**

5. The RecipientPresenter is responsible for:
 - a. forwarding information between objects e.g. BuilderSetProperties (MessageBuilder builder) which takes in a builder and forwards builder instances to messageBuilder.
 - b. receiving requests from some objects and forward it to other objects. e.g. BuilderSetProperties (MessageBuilder builder) which is called from the MessageCompose object and forwards the request to the MessageBuilder object by calling its methods. **This is the correct answer.**
 - c. Both a and b are correct.

6. The MessageBuider is responsible for:
 - a. Processing incoming requests from other objects on demand. e.g. setters methods being called from the MessageCompose object, which in turn set new values to the variables.
 - b. Setting up configurations necessary for system functionality. e.g. BuildAsync method being called from the MessageCompose objects to build a message asynchronously.
 - c. Both a and b are correct. **This is the correct answer.**

The scores of sub-task 2 is calculated according to the following criteria:

1. If the chosen answer is correct, the participant gets one point.
2. If the correct answer includes two answers (e.g., Both 1 and 2), the participant gets two points for choosing the answer that includes both correct answers. For instance, in question number 1, if the participant chooses d (which is the right answer), he/she gets 2 points. If only a or d is chosen (which are correct but not enough), the participant gets 1 point only. If the participant does not get any point.

Overall Score:

To calculate the overall score for sub-task 1 and sub-task 2, We add the scores gained in both sub-tasks.