CSC 581 [PROJECT REPORT – 3]

I. PARTICIPANTS

Sajeel Mohammed Abdul
Cole Allen
Ananya Kakumanu
Shoaibuddin Mohammed
Malika Hafiza Pasha
Lokesh Purohit
Christian Quintero

II. RELEVANT USE CASES AND FLOWS

1. Request Service
    - Description: This use case describes the process of a ***customer*** requesting a service from a ***handyman***. The ***customer*** must first select a particular ***handyman***, then send a request to the latter; requests are saved to the ***external database***.

    - Basic Flow:
        1. A customer navigates to their dashboard(A1) and clicks the “Create Service Request” button.
        2. The customer inputs a description of the requested service and selects the desired appointment date and time. (A2)
        3. The system displays all the available handymen with their ratings and other information
        4. The customer selects a preferred handyman and writes a service request with a text explanation, optionally accompanied by images.
        5. Upon the customer clicking “Send”, the service request is saved to the external database and sent to the selected handyman. Both the handyman and the customer can view the service request at any time. (A3)

    - Alternative Flow:
        i. <A1 – The user browses through handymen and finds one with the qualifications that they desire. The customer then starts a service request from that handyman’s profile.>

ii. <A2 – If the customer has preferences or does not see any handymen that they would like to choose, they can set filters for the list of displayed handymen to narrow down the results to best fit their criteria.>
iii. <A3 – The customer or handyman may cancel a request if they desire.>

2. Service Response
    - Description: This use case describes the process of the ***handyman*** viewing a service request from a ***customer*** and giving the ***customer*** a quote based on that service request, and the ***customer*** accepting the service quote and setting an appointment time.

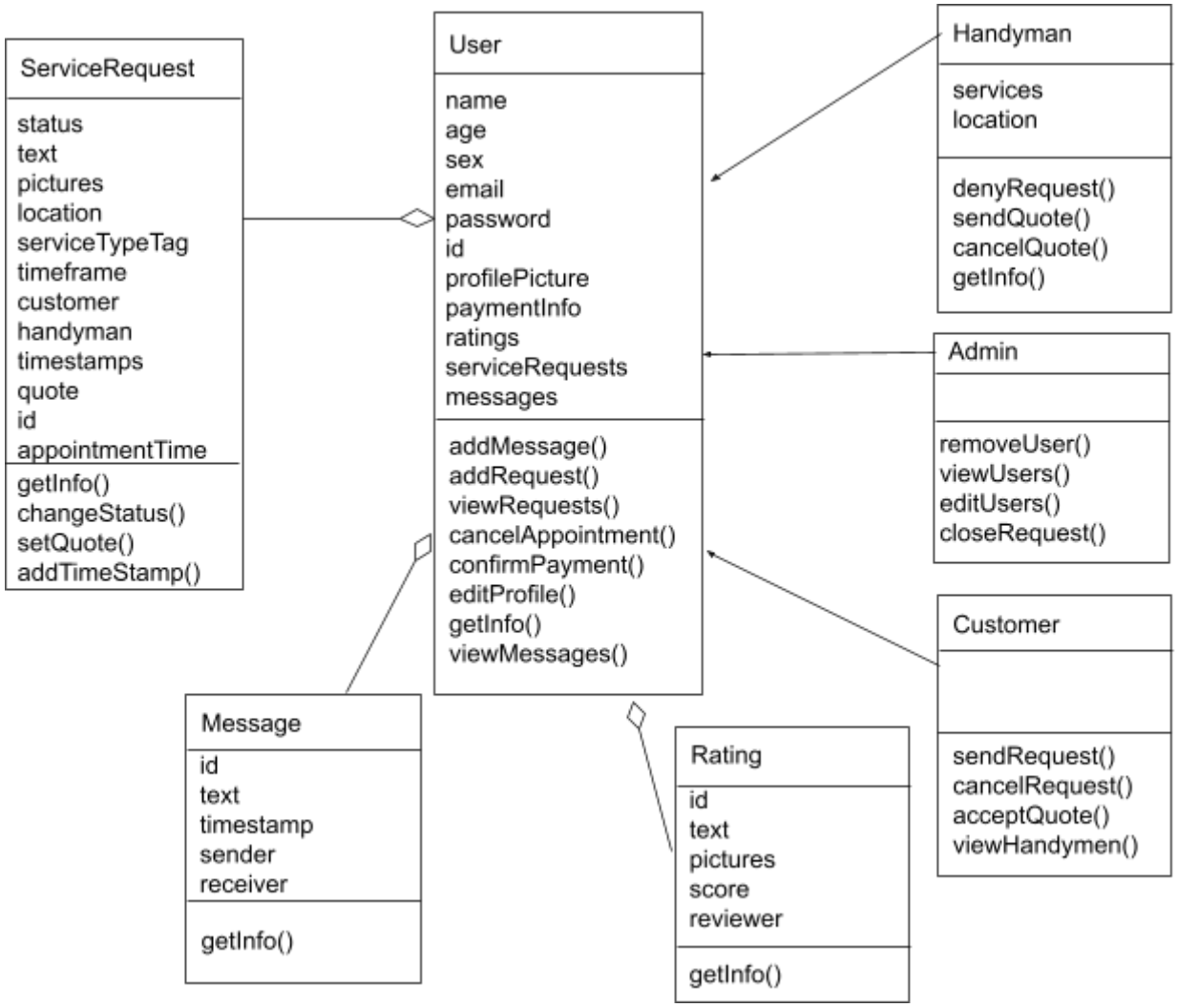
    - Basic Flow
        1. The handyman receives a service request, proceeds to check the description of the service required, appointment date and time requested, and location distance, and sends a service quotation.(A1)
        2. On the customer's acceptance of the quotation(A2), the system asks for payment confirmation.
        3. The customer fills in the payment information to complete the transaction setup(A3), in which the system confirms the service appointment to both the customer and the handyman.(A4)

    - Alternative Flow:
        i. <A1 – The handyman may instead opt to deny the service request.>
        ii. <A2 – The customer may instead opt to deny the quotation or message the handyman to discuss pricing.>
        iii. <A3 – If the customer plans to instead pay by cash or check, they must note this here to continue without putting in online payment information; the handyman then has the option to confirm or deny payment method acceptability, and thus confirm the appointment.>
        iv. <A4 – The customer and the handyman both have the option to review appointment details at any time and cancel the appointment if unforeseen circumstances or changes in plans require the service to be postponed or canceled.>

III. LIST OF CLASSES

- User: This class represents any user of the system, including customers, handymen, and admins.
- Handyman: This class represents a handyman who provides services to customers through the system.
- Admin: This class represents an administrator of the system, who has special privileges such as viewing and managing user accounts.
- Customer: This class represents a customer who requests services from handymen through the system.
- ServiceRequest: This class represents a request from a customer for a service from a handyman.
- Rating: This class represents ratings assigned to the user after every service request is completed.
- Message: This class represents a single message sent from one user to another.

IV. UPDATED CLASS DIAGRAM



Notes:

1. Admin and Customer intentionally have no unique attributes.
2. Database and payment methods are assumed to be handled by library methods and thus no classes are presented for them here.

V. CLASS UPDATES FROM THE SEQUENCE DIAGRAM

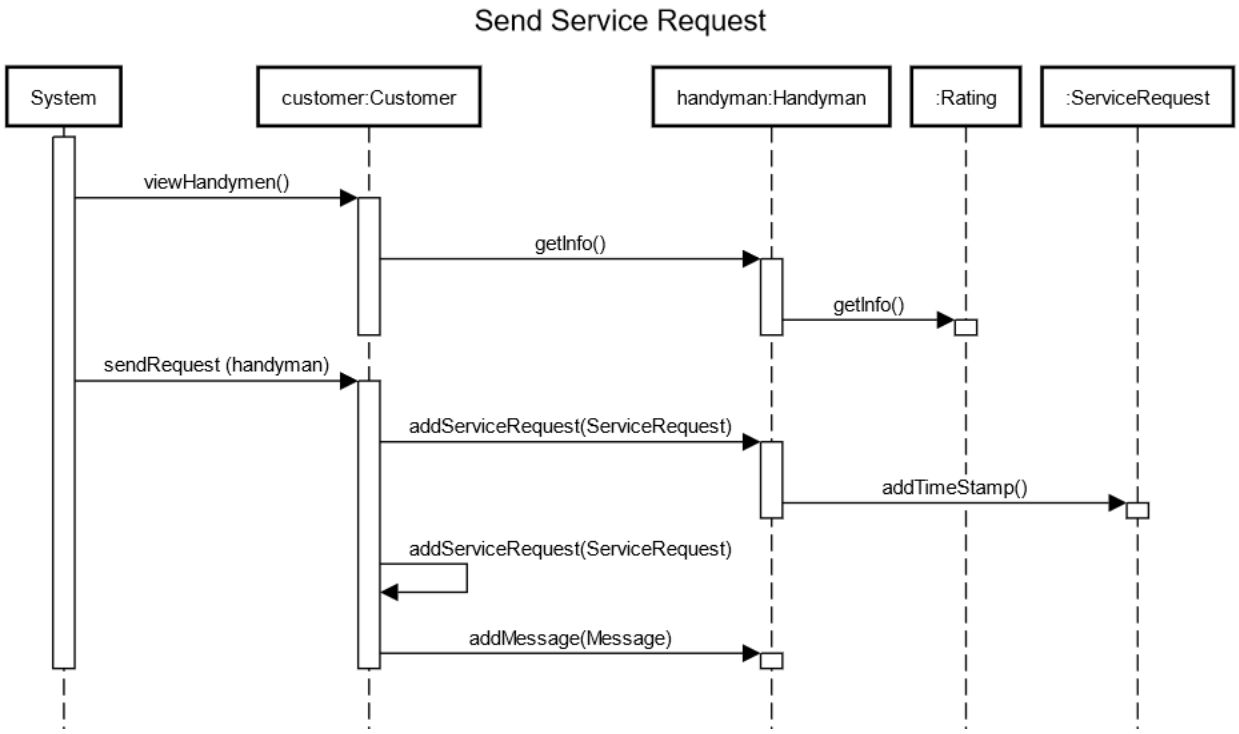
Over the course of working out the sequence diagrams, we realized that we were missing several methods:

1. User needed to be able to add/view Service Requests
2. User needed to be able to add/view Messages
3. Customer needed to be able to view Handymen

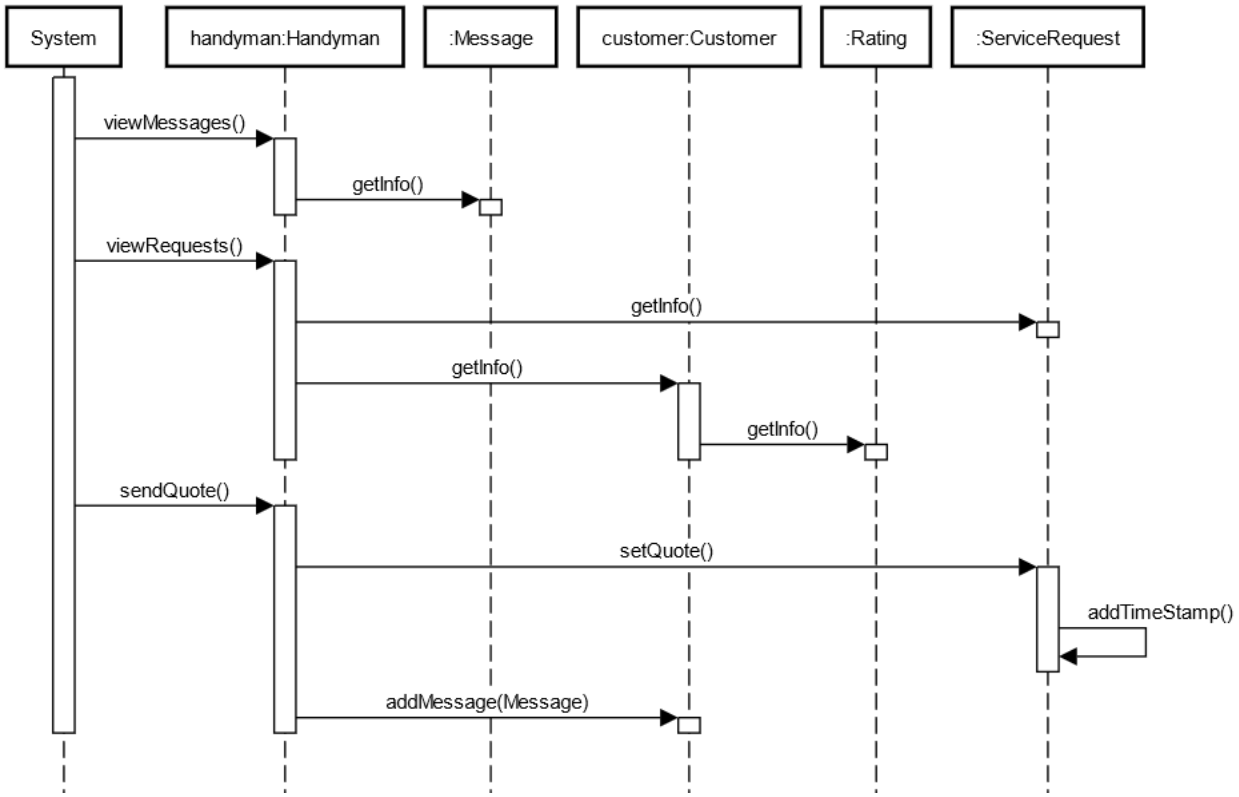
We also realized that we never made a proper UI or App class, so "System" fills in that functionality in the current sequence diagrams.

VI. SEQUENCE DIAGRAMS

Sequence Diagrams



Response to Service Request



Accept Quote