Use Case Specification

Participating Actors

The use case is initiated by a Citizen.

Brief Description

This use case will allow user to create specific groups. The system will have a page showing all available groups created. And people can choose to join the group. When an emergency happens, people in the group can post the details of the specific emergency and all people in the group will be notified of the emergency.

Assumption

...

Flow of Events

Basic Flow

1. The use case starts when the citizen elects to see all the groups created.
2. The system displays all the groups created by users and allows the user to join the group.
3. The Citizen elects to join one of the group.
4. If the user is not already in the group, the system ask the user to confirm that he/she wants to join the group.
5. The Citizen confirms that he/she wants to join the group.
6. The system adds the group to the list of the Citizen's groups and alerts the Citizen that he/she is in the group.
7. The Citizen elects to see all the groups he/she is in.
8. The system displays all the groups the user is in and allows the user to see the messages in the group and allows the user to leave the group. And the system provides a space for the user to create a group.
9. The Citizen elects to post a message in a group he/she is in.
10. The system displays all the messages posted in the group and provides a space to write a new one.
11. The Citizen writes a new message.
12. The system stores and displays the message (together with sender name, timestamp, and status at the time the message was sent) so others can see it.

Alternative Flows

- A1. In step 2, if there are no groups created, none is displayed.
- A2 In step 4, if the user is already in the group, the system alert the user that he/she is already in the group
- A3 In step 5, if the Citizen does not want to join the group. The use case returns to step 2.
- A4 In step 8, if there are no groups that the Citizen is in, none is displayed.
- A5 In step 9, if the Citizen elects to leave one of the group he/she is in.
  - 1.The system ask the user to confirm that he/she wants to leave the group.
  - 2.The Citizen confirms that he/she wants to leave the group.

    - 3.The system removes the group from the list of the Citizen’s groups and alerts the Citizen that he/she has left in the group.
- A6 In A5 step 2, if the Citizen does not want to leave the group. The use case returns to step 8.
- A7 The Citizen elects to create a group.
    - 1.If the group name is not empty, the system ask the user to confirm that he/she wants to create the group.
    - 2.The Citizen confirms that he/she wants to create the group.
    - 3.The system adds the group to the list of the Citizen’s groups and all the groups created and alerts the Citizen that the group is created.
- A8 In A7 step 1, if the group name is empty, the system alerts that the group name can not be empty
- A9 In A7 step 2, if the Citizen does not want to create the group. The use case returns to step 17.
- A10 In step 10, if there is no existing messages in the group, none is displayed. The system continues in step 15.

<u>Rules</u>

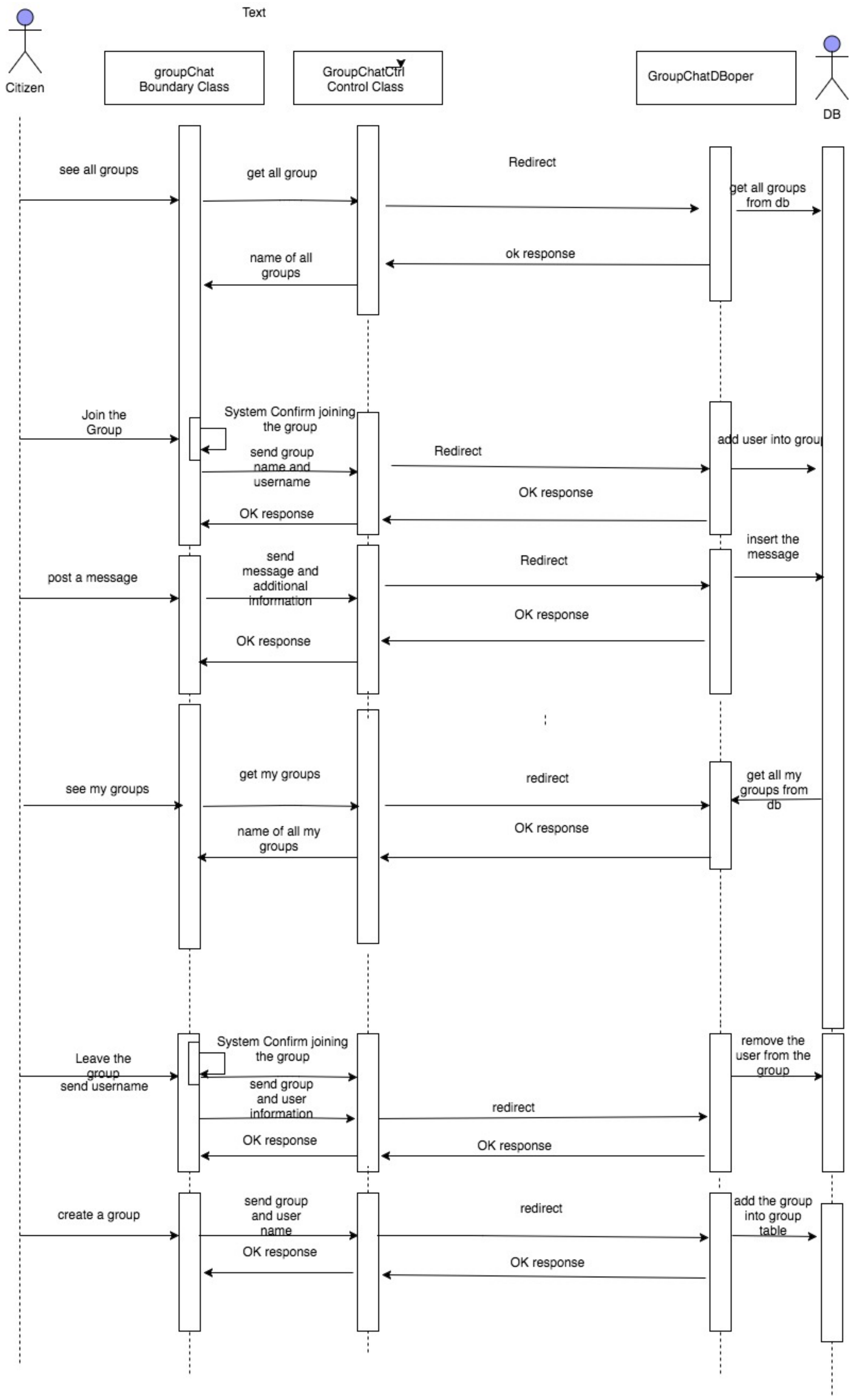
- The user can not join a group he is already in.
- The group name can not be empty

Use Case Analysis Model (OOA)

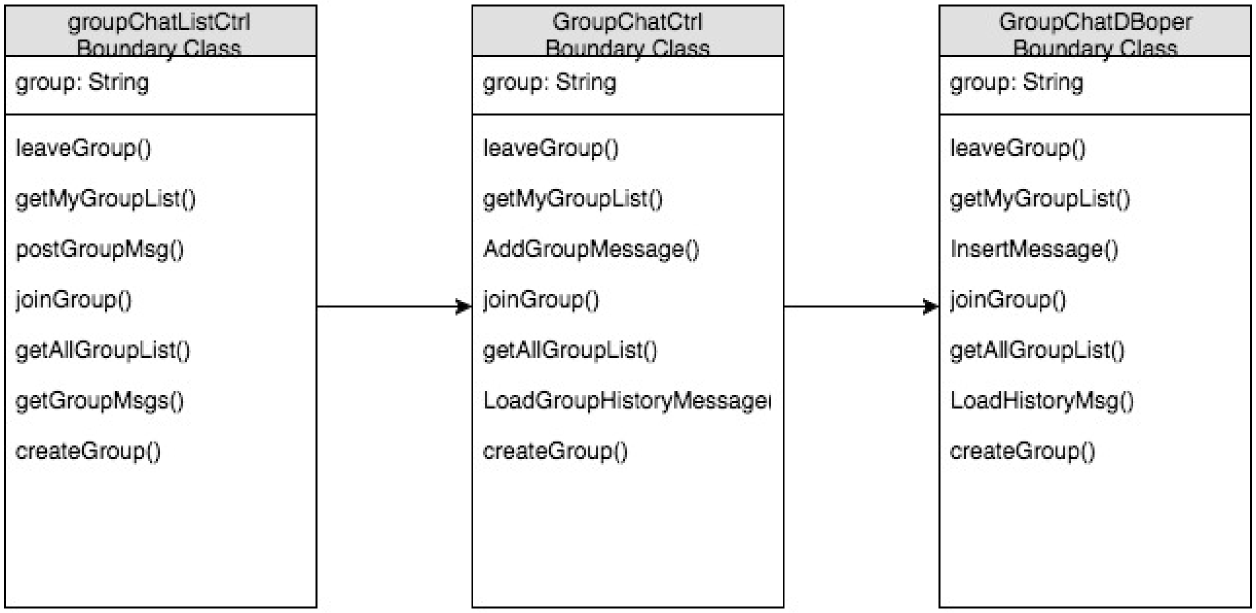
Entity classes: Message
Boundary classes: groupChatListCtrl, GroupChatDBoper
Control classes: GroupChatCtrl

Sequence diagram modeling the use-case behavior:

Text


Citizen
groupChat Boundary Class
GroupChatCtrl Control Class
GroupChatDBoper
DB
see all groups
get all group
Redirect
get all groups from db
name of all groups
ok response
Join the Group
System Confirm joining the group
send group name and username
Redirect
add user into group
OK response
OK response
post a message
send message and additional information
Redirect
insert the message
OK response
OK response
see my groups
get my groups
redirect
get all my groups from db
name of all my groups
OK response
Leave the group send username
System Confirm joining the group
send group and user information
redirect
remove the user from the group
OK response
OK response
create a group
send group and user name
redirect
add the group into group table
OK response
OK response

Class diagram (based on the sequence diagram) modeling the use-case structure:



Mapping Between Analysis Classes and Code:

Analysis Classes	**Implementation Elements** (e.g. modules, files, components, databases)
groupChatListCtrl	/public/javascripts/groupChatListCtrl.js
GroupChatDBOper	/models/GroupChatDBOper.js
GroupChatCtrl	/controllers/GroupChatCtrl.js