Student 5	Proposal	Approved
Bo	**Step 1: High-Level Requirement** Use Case Name: Reporting spam Use Case Brief Description: In this use case, citizens are able to report a spam message or a spam user by clicking the report option for any message shown on the public wall, so as to keep important emergency messages visible. If the message is reported by different 3 users, then it will be hidden. If the user is reported by different 5 users, then he can’t send a message on the public wall forever.	Y
	Faculty/TA Comments:	
	Step 2: Clickable UI Mockups Link: https://drive.google.com/open?id=1wv0p6BaorCDuE2pHfSOhJNO89laCxzXt	Y
	Faculty/TA Comments:	
	Step 3: Elaboration - Use Case Specification See below (min 6 steps in basic flows and 6 alternative flows or rules)	Y
	Faculty/TA Comments:	
	Step 4: Use-Case Analysis Model (OOA) See below	Y
	Faculty/TA Comments:	

Use Case Specification

Participating Actors

Citizens

Brief Description

In this use case, citizens are able to report a spam message or a spam user by clicking the report option for any message shown on the public wall, so as to keep important emergency messages visible. If the message is reported by different 3 users, then it will be hidden. If the user is reported by different 5 users, then he can’t send a message on the public wall forever.

Assumption

User is logged in.

Flow of Events

Basic Flow

1. The use case starts when the citizen elects to report a spam through clicking the report option in the message box.

2. The system shows up a spam report form under the message box, and requests that the user selects the spam level, as per the **Spam Level Rule**, a spam type (False, Harassing, AD, others), and specifies a report reason. Message is the DEFAULT spam level.

3. The user selects the message spam level, a spam type, and specifies a report reason. Then click the submit button.

4. The system receives and saves the spam report into the database, and updates the message status on every user's public wall.

5. The user sees the message ‘submit successfully’. Submit button and cancel button disappear, because the user can only report once for the same message. Even if the user refreshes the public wall, the report option of that message is not clickable for him.

6. All users will see the number 1 near report option in that message, which indicates this message has been reported once. The use case ends.

Alternative Flows

- A1. In step 3, if the user didn’t click the submit button but the cancel button, the form under the message box will disappear. The use case ends.
- A2. In step 3, if the user didn’t provide a spam type, or specify a report reason. The user can’t be allowed to click submit, until the info is provided. The use case proceeds to step 4.
- A3. In step 6, if the message has been reported by 3 different users according to **Spam Message Hidden Rule**, then the content of that message will be replaced to “This is a spam message reported by 3 users”. Report option won’t be clickable. The use case ends.
- A4. In step 6, if the message has been reported 2 times, then all users will see the number 2 near report option in that message, which indicates this message has been reported twice. The use case ends.
- A5. In step 3, if the user selects the user spam level, and provides a spam type and a report reason. Then click the submit button. The system receives and saves the spam report into the database, and updates the user status on every user's public wall. The use case proceeds to A6.
- A6. All users will see the number N near that user’s name in every message in the public wall, which indicates the user has been reported N times. The use case ends.
- A7. In step A6, if the user has been reported more than 5 times according to **Spam User Block Rule**, she/he can’t send any message on the public wall. When he/she clicks the send button, one error message will pop up with the content “You are identified as a spam user. ” But, other functions like private chat and changing status are still available to the spam user. The use case ends.

Rules

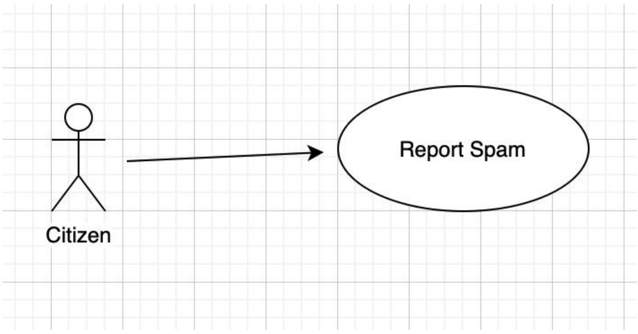
- **Spam Level Rule**
    1. Message level is used when the reporter believes this message is spam.
    2. User level is used when the reporter believes this user sends too many spam messages or has abusive behaviour.
- **Spam Message Hidden Rule**

If one message is reported by 3 different users, it will be judged as a spam message and then be hidden and invisible to all users.

- **Spam User Block Rule**

If one user is reported by no less than 5 different users, it will be judged as a spam user and then be forbidden to send messages on the public wall.

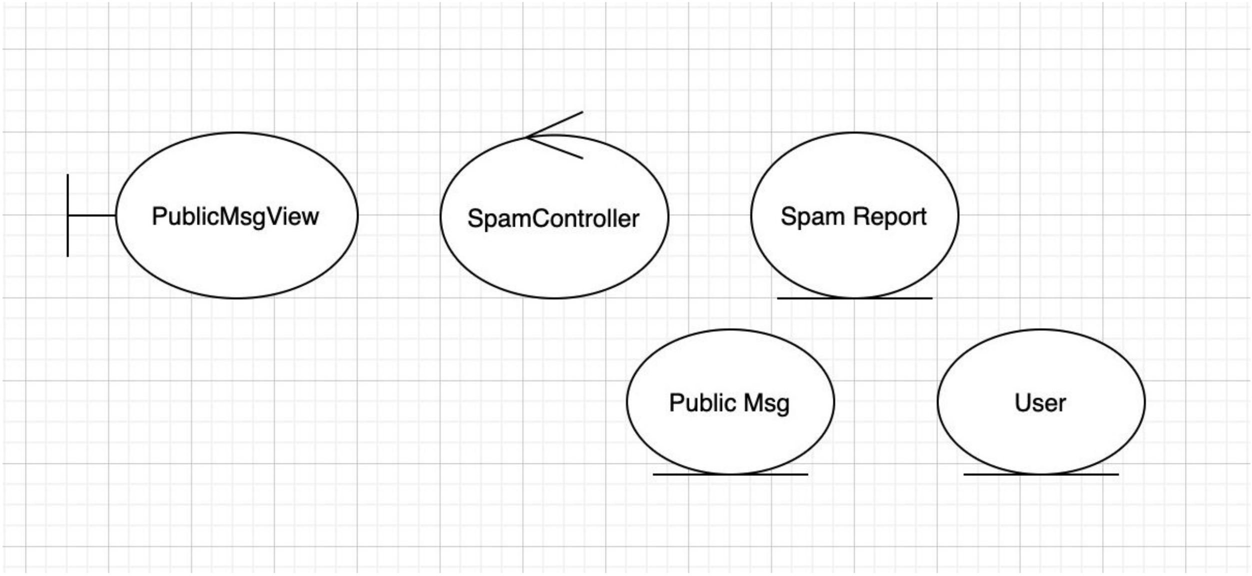
Use Case Analysis Model (OOA)



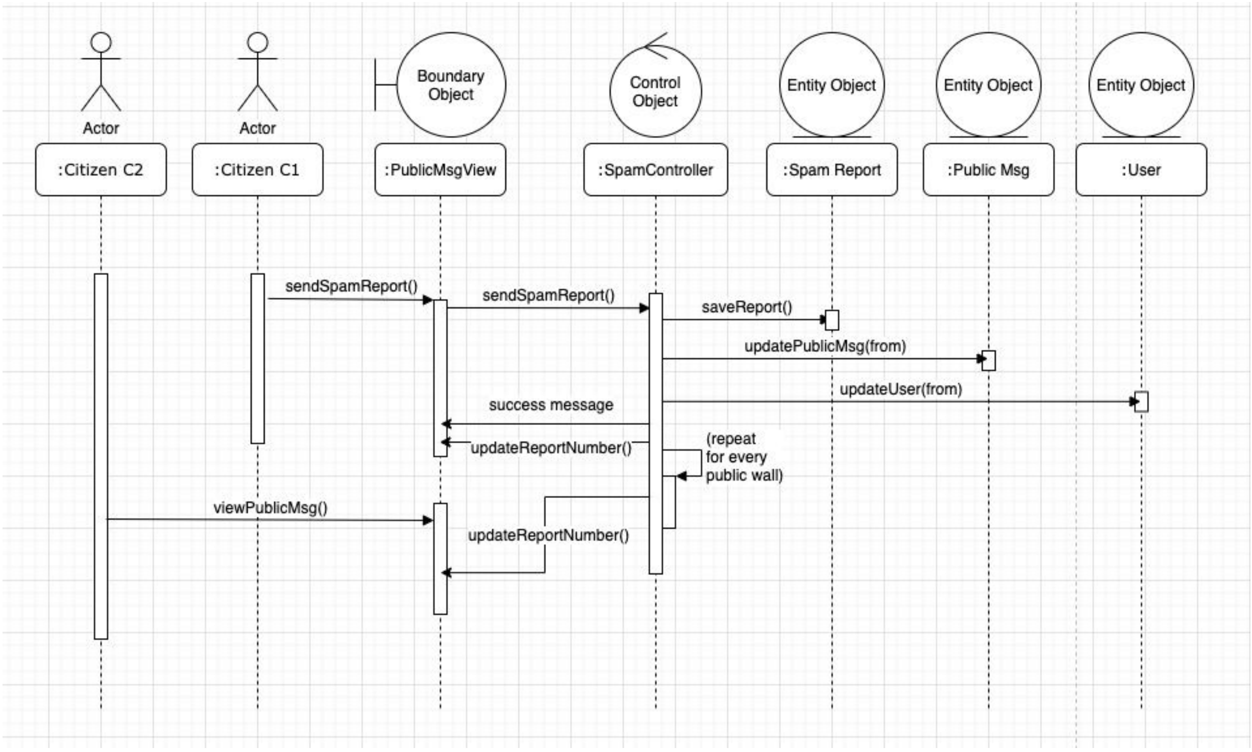
Entity classes: SpamReport

Boundary classes: PublicMsgView

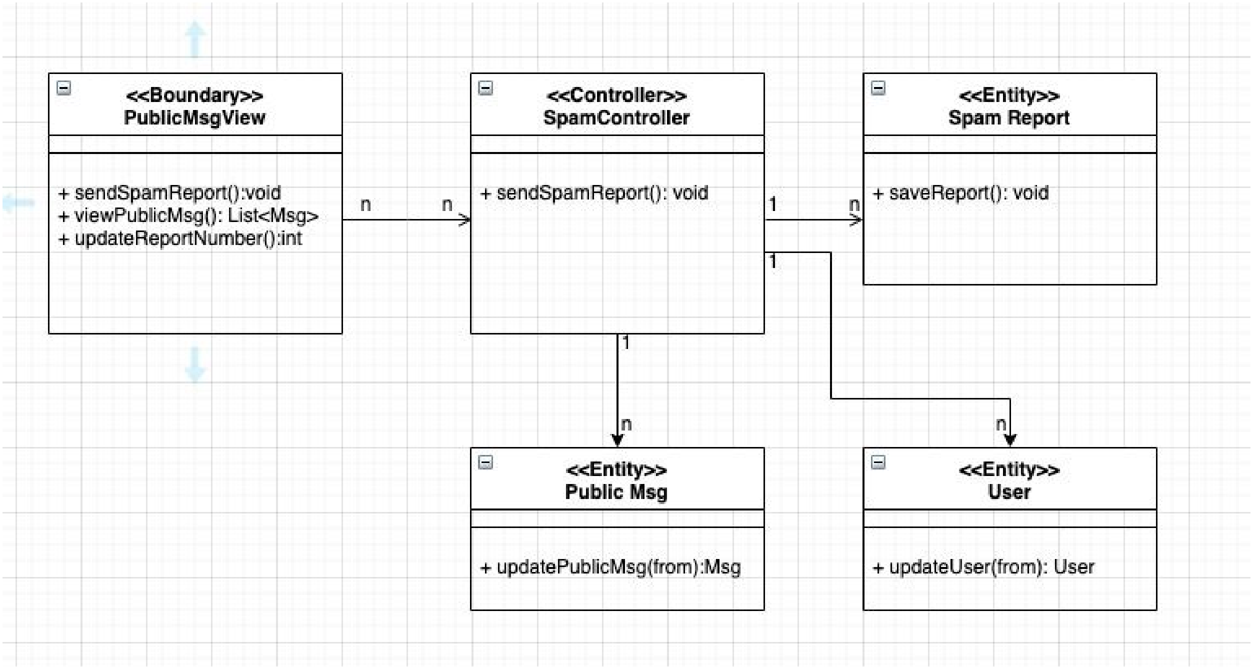
Control classes: SpamController



Sequence diagram modeling the use-case behavior:



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)
PublicMsgView	public/javascripts/chat.js
SpamController	controllers/SpamController.js
Spam Report	model/SpamReport.js
Public Msg	model/chatMessage.js
User	model/user.js

Quality Report (for your individual branch only, not master)

Unit tests you added in Iteration 4

Module/File Tested	**# Tests**	**# Failing Tests**
models/spamReport.test.js	Create a new message spam report	

models/spamReport.test.js	Create a new user spam report	
models/user.test.js	Set user spam in db	
models/chatMsg.test.js	Set message spam in db	

Integration tests you added in Iteration 4

Module/File Tested	# Tests	# Failing Tests
api/spamReport.test.js	User spam	
api/spamReport.test.js	Message spam	

Bugs found through testing in Iteration 4

Module/File	Description	Fixed? (Y/N/Partially)
No bug found through testing		

Coverage (your branch only)

Line Coverage	**94.98%**
Branch coverage	**88.12%**