CSE 460

Programming Assignment Report

Posting ID: 2150-386

18th November 2019

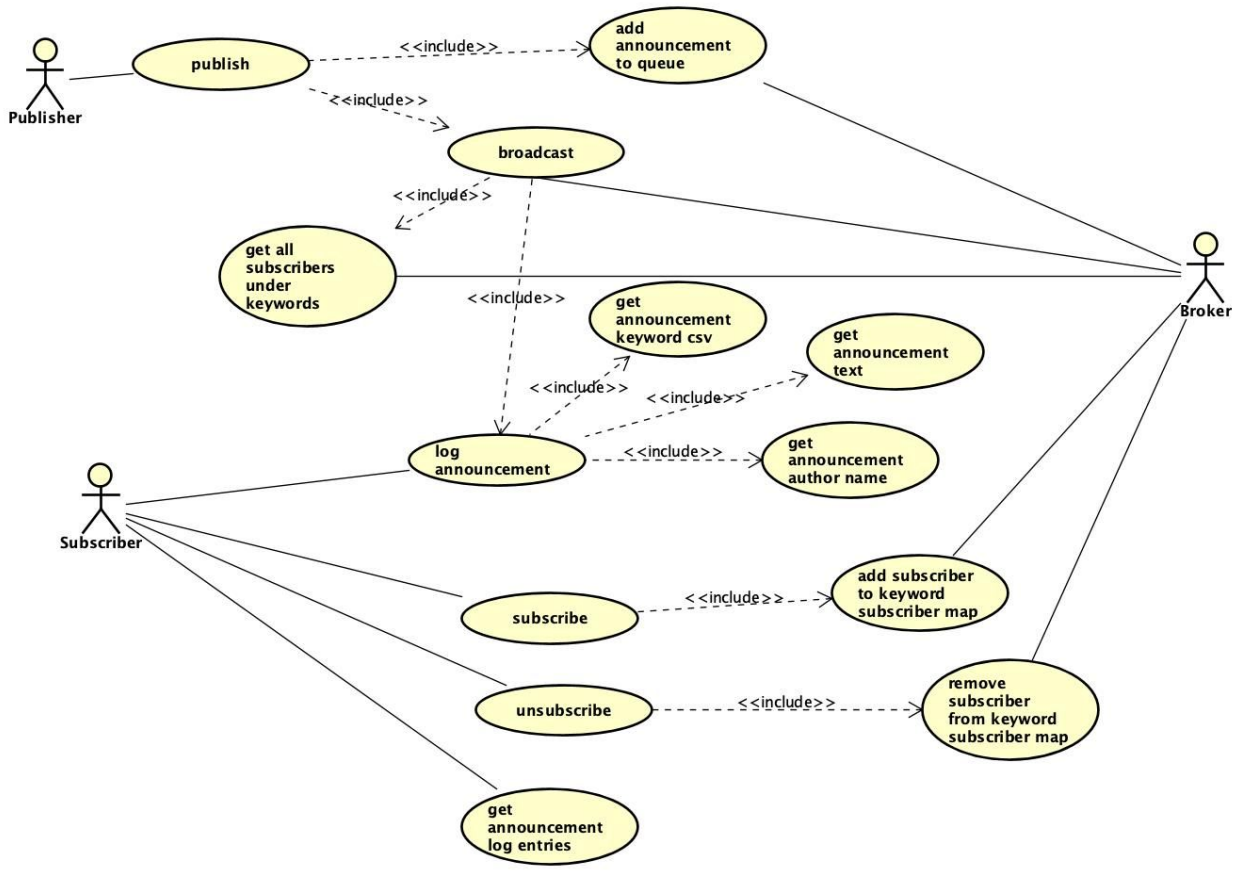
Fall 2019

1. Description and Assumptions

The SocialNetworkingSystem is a system to communicate between publisher and subscribers. It uses a publisher-subscriber design pattern to reach multiple customers with the same content. The system has two interfaces for each type of user, like for publisher its specific functionality are from the IPublisher interface and for subscriber its specific functionality are from the ISubscriber interface.

The current system keeps announcements as an instance of Announcement class. Means when the author publishes an announcement with text and keywords, our broker in system will initialize one announcement object with the text of the announcement, keywords and publishing author. With this it is easy to maintain throughout the system with clear abstraction.

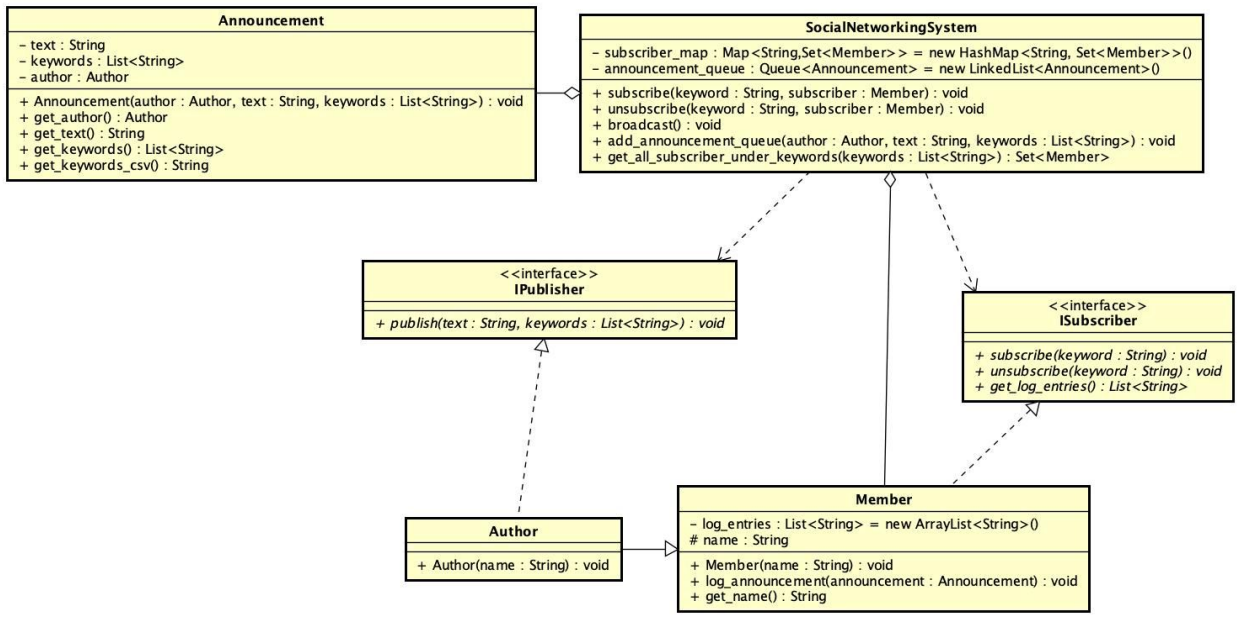
2. Use Case Diagrams



Actors:

1. **Publisher:** They are the author of the announcements. Publishers can publish announcements but to publish they use functions of broker which is adding announcements to queue and after broadcast the announcement.
2. **Subscriber:** They are the consumers of the announcements. Subscribers can subscribe or unsubscribe to a keyword and in order to achieve that subscribe/unsubscribe functionality call broker to keep record of its subscription. Another functionality is to keep a log of all the announcement subscribers receive and also provide that log whenever requested.
3. **Broker:** It is a mediator between publisher and subscriber. Broker broadcast announcements which are published by the publisher to the specific subscribers who have subscribed to the keywords. Broker also keep track of subscriber subscription lists under keywords.

3. Class Diagrams



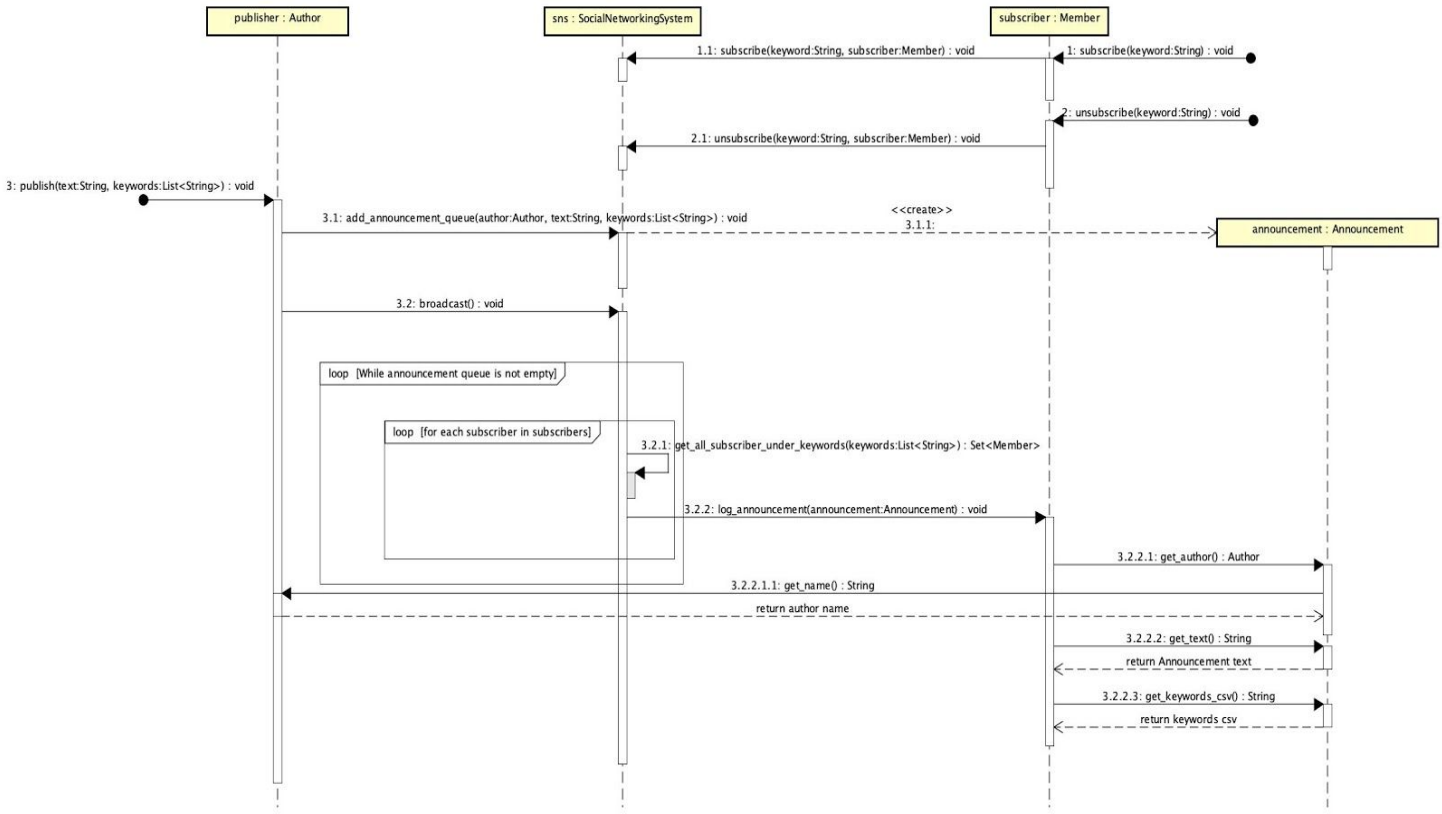
The class diagram shows the whole SocialNetworkingSystem with pub-sub design. The system includes 4 classes named Announcement, SocialNetworkingSystem, Author and Member. And there are two interfaces named, IPublisher and ISubscriber.

Description for class diagram relations:

1. **SocialNetworkingSystem depends on ISubscriber:** It uses ISubscriber interface to create an object of subscriber which is a Member of the system.
2. **SocialNetworkingSystem depends on IPublisher:** It uses IPublisher interface to create an object of publisher which is a Member of the system.
3. **Author realizes IPublisher:** To give Author a set of method which publisher can do is implemented through IPublisher interface.
4. **Author inherits Member:** To give Author a specialized Member who can publish an announcement Author class inherits Member class and enhance its publishing functionality through IPublisher interface.

5. **Member aggregates to SocialNetworkingSystem:** All Members are part of SocialNetworkingSystem.
6. **Announcement aggregates to SocialNetworkingSystem:** The system generates and keeps track of Announcement objects for all the published content from the Author and they are broadcasted to subscribers.

4. Sequence Diagrams

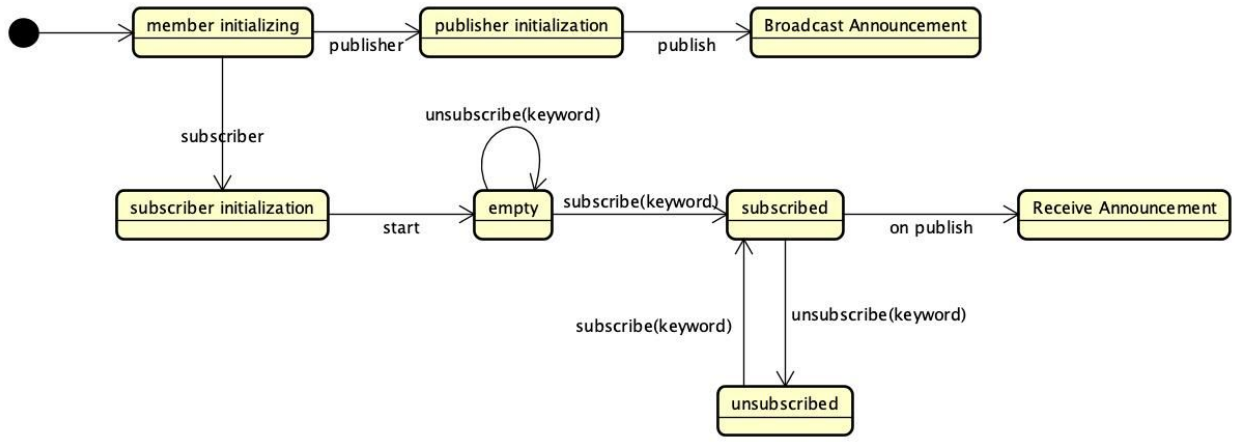


This is a combined sequence diagram of multiple scenarios. All the sequences with their number and short description are state below.

1. Subscriber subscribes to a keyword: subscriber call function of **sns** object to include its record into broker's system.
2. Subscriber unsubscribe from a keyword: subscriber call function of **sns** object to opt out of receiving announcement for the specific keywords.
3. Publisher publish an announcement:
   - 3.1. Add announcement to queue: using **sns** object author can provide its announcement to keep it in a queue before broadcasting it.
   - 3.2. Broadcast: send the announcement to all the subscribed members.
     - 3.2.1. Fetches all the subscribers under keywords of announcement.
     - 3.2.2. For all subscribers, the **sns** object makes a call to log an announcement to the subscriber object.

3.2.2.1. Fetches name of an Author
3.2.2.2. Fetches announcement text.
3.2.2.3. Fetches announcement keywords in csv form.

5. State Machine Diagrams



The above state machine diagram shows different states of a member. First state is member initialization which is the same as the **new Member("name")** in java. Now the member can be of two types: first is subscriber and second is publisher.

Subscriber's initial state is empty means it has not subscribed to any keywords. from the empty state if subscriber subscribe to the keyword it goes to subscribed state. If a member unsubscribes from an empty state then it stays empty. From subscribed state subscribers can receive an announcement or it can unsubscribe to the keyword.

Publisher after initialization gets ready to publish announcements. The published announcement is received by subscribers only.

6. Readme

- In the System Broker class is SocialNetworkingSystem
- Brokers handles communication between publisher and subscriber.
- To create a new publisher object.
    - **IPublisher author = new Author("Name");**
- To create a new subscriber object.
    - **ISubscriber member = new Member("Name");**
- To publish an announcement first create a list of keywords.
    - **List<String> keywords = new ArrayList<String>();**
    - To add new keyword: **keywords.add("ASU");**
- To publish an announcement.
    - **author.publish("Text message", keywords);**
- To subscribe a member to a keyword.

    - **member.subscribe(“keyword”);**
- To unsubscribe a member from a keyword.
    - **member.unsubscribe(“keyword”);**
- To fetch all the announcements received by subscribers.
    - **member.get_log_entries();**

7. Appendix & Credits

- Object-Oriented Analysis and Design with Applications, 3ndEdition, G. Booch, Benjamin Cummings
- Astah Sequence Diagram Documentation: http://astah.net/manual/434-sequence-diagram