



INFORMATICS 3A10

Final Year Project



Group Name	The Overclockers
Group Number	25
Project Name	Eternal Vibes
Name of Project Supervisor	Brian Greaves

Name	Surname	Student Number
Ronald	Lai	201433999
David	Innes	201474149
Devandrin	Kuni	201320596
Marc	Zuze	201477488

Table of Contents

Plagiarism Document.....	3
Mobile App Interaction Sequence Diagrams.....	4
Website Interaction Sequence Diagrams	11
Component and Deployment Diagram.....	17
Entity Relationship Diagram	18
Appendix – Use Cases.....	19
Appendix - Class Diagrams.....	31

Plagiarism Document



Academy of Computer Science and Software Engineering

Informatics 3 Group Project


Plagiarism Declaration


Deliverable #	3	Team #	25	Supervisor	Brian Greaves
Team Name	The Overclockers				

1. Plagiarism is to present someone else's ideas as our own.
2. Where material created by other people has been used (either from a printed or digital source) this has been carefully acknowledged and referenced. We have used the appropriate style for citation and referencing. Every contribution to and quotation from the work of other people in this deliverable has been acknowledged through citation and reference.
3. We know that plagiarism is wrong.
 - 3.1. We understand what plagiarism is and am aware of the University's policy in this regard.
 - 3.2. We know that we would plagiarise if we do not give credit to my sources, or if we copy any part of a book, article, or Internet source without proper citation.
 - 3.3. We know that even if we only change the copied work slightly, we still plagiarise when using someone else's work without proper citation.
 - 3.4. We declare that we have created our own work throughout this deliverable and we have credited all ideas we have gained from other people's work.
4. We declare that this deliverable is our own original work.
5. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as his or her own work.

Signature 
Full Name Marc Lerhanzo Zuze
Student # 201477488
Date 18/04/2017
Member 1

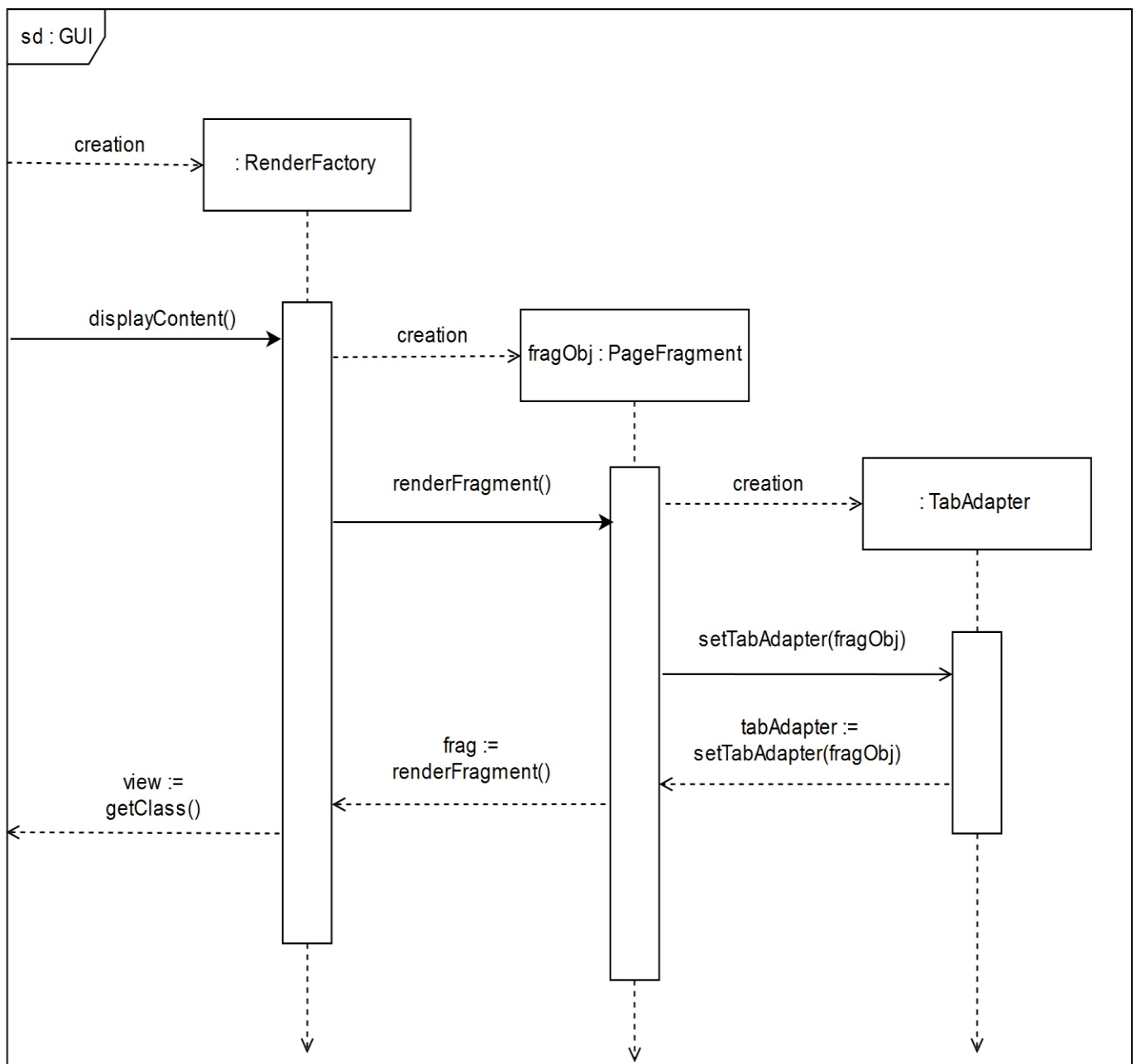
Signature 
Full Name Devandrin Kuni
Student # 201320596
Date 18/04/2017
Member 2

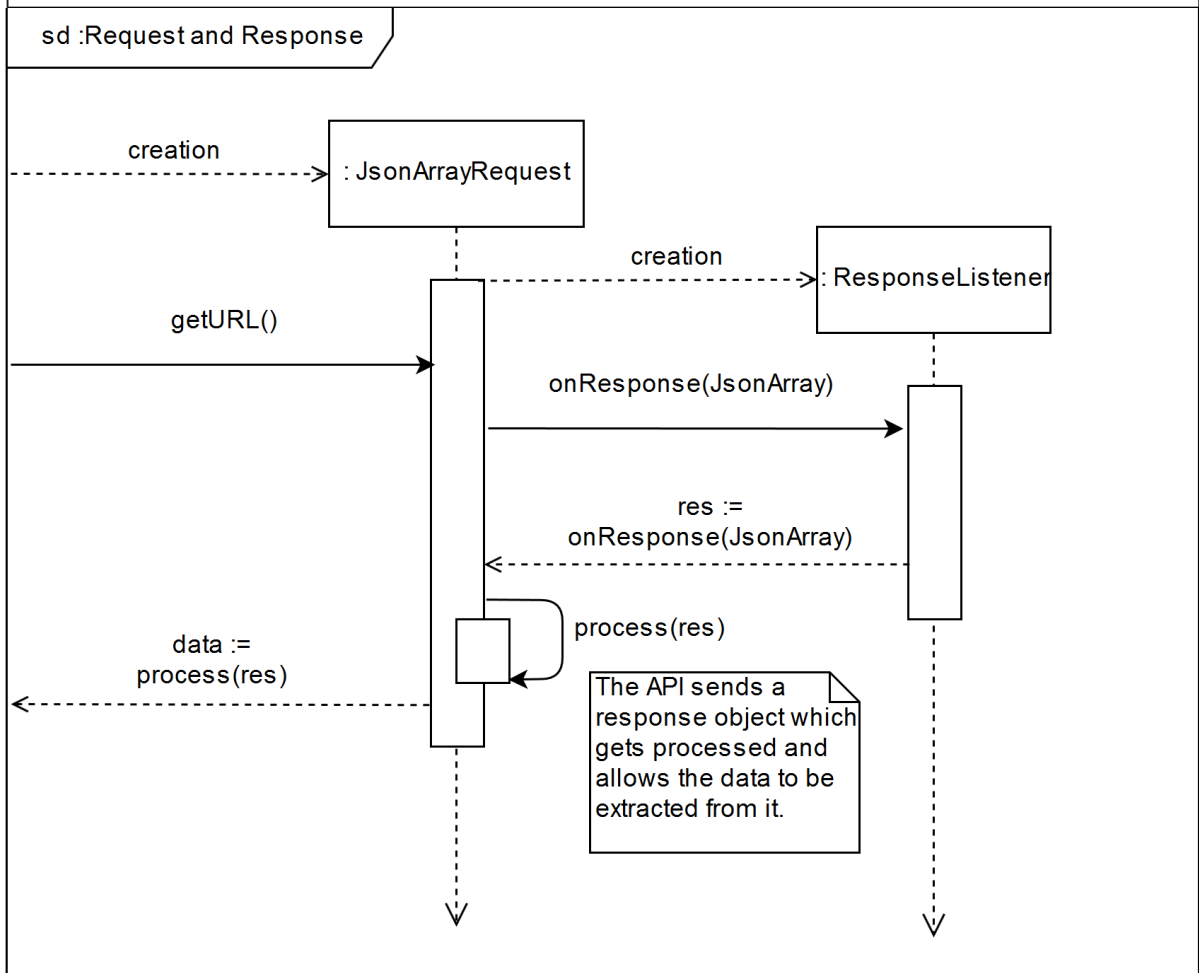
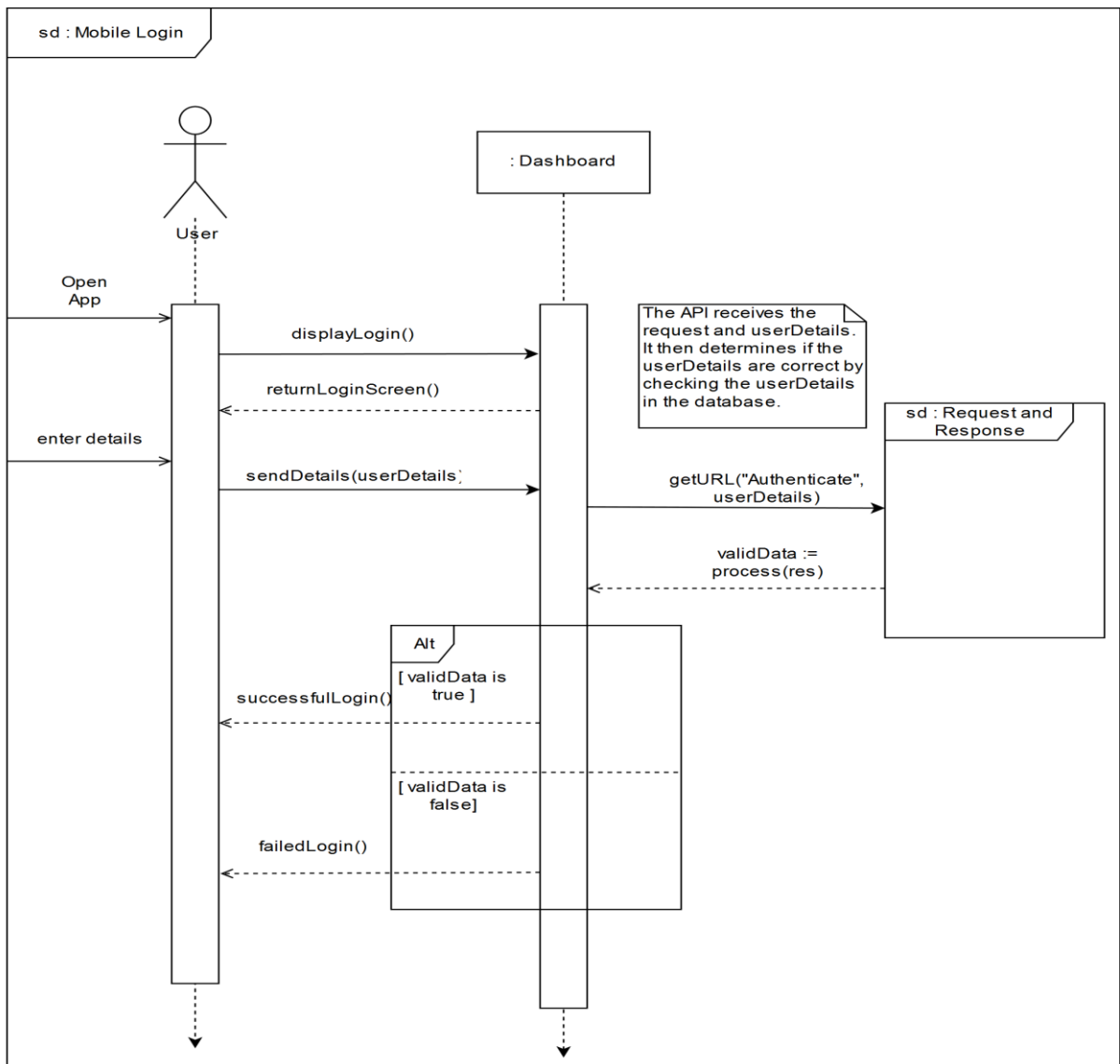
Signature 
Full Name David Innes
Student # 201474149
Date 18/04/2017
Member 3

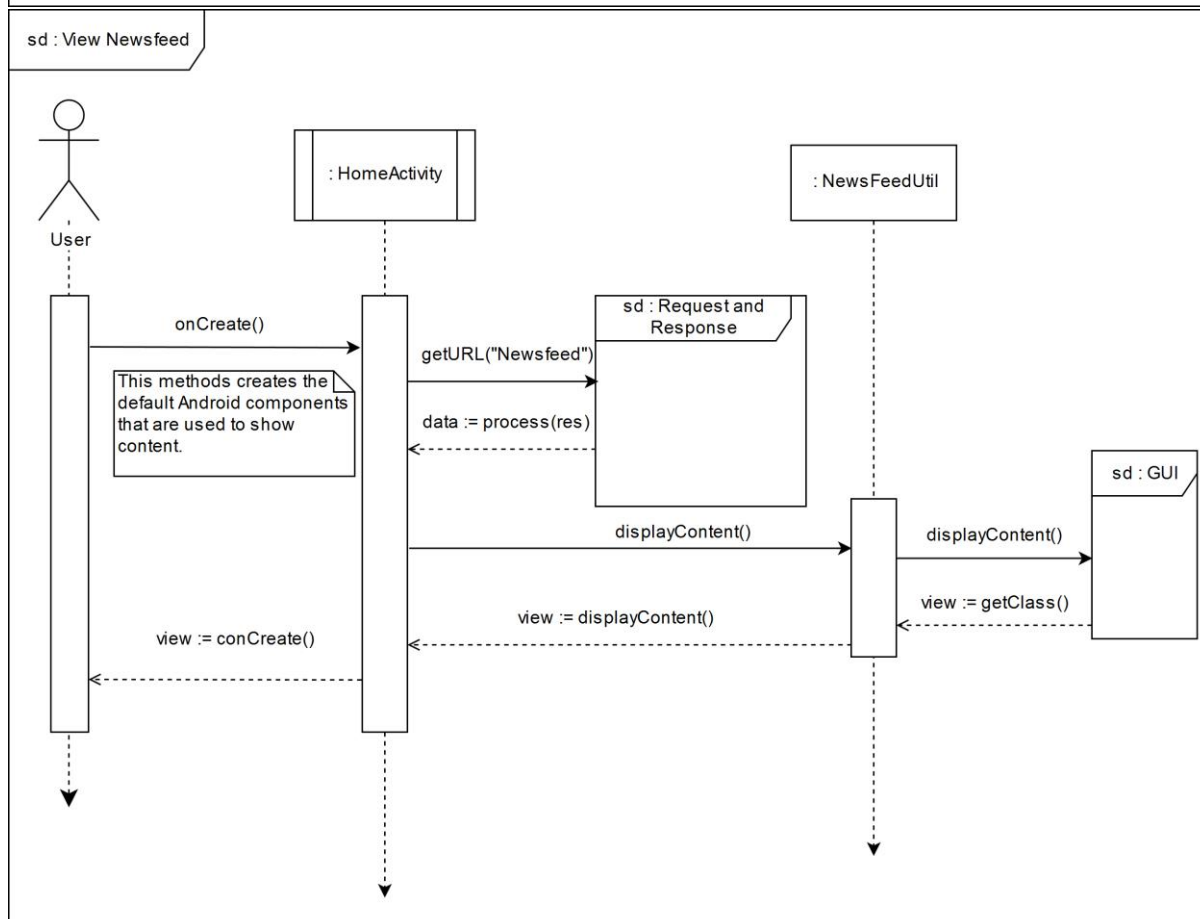
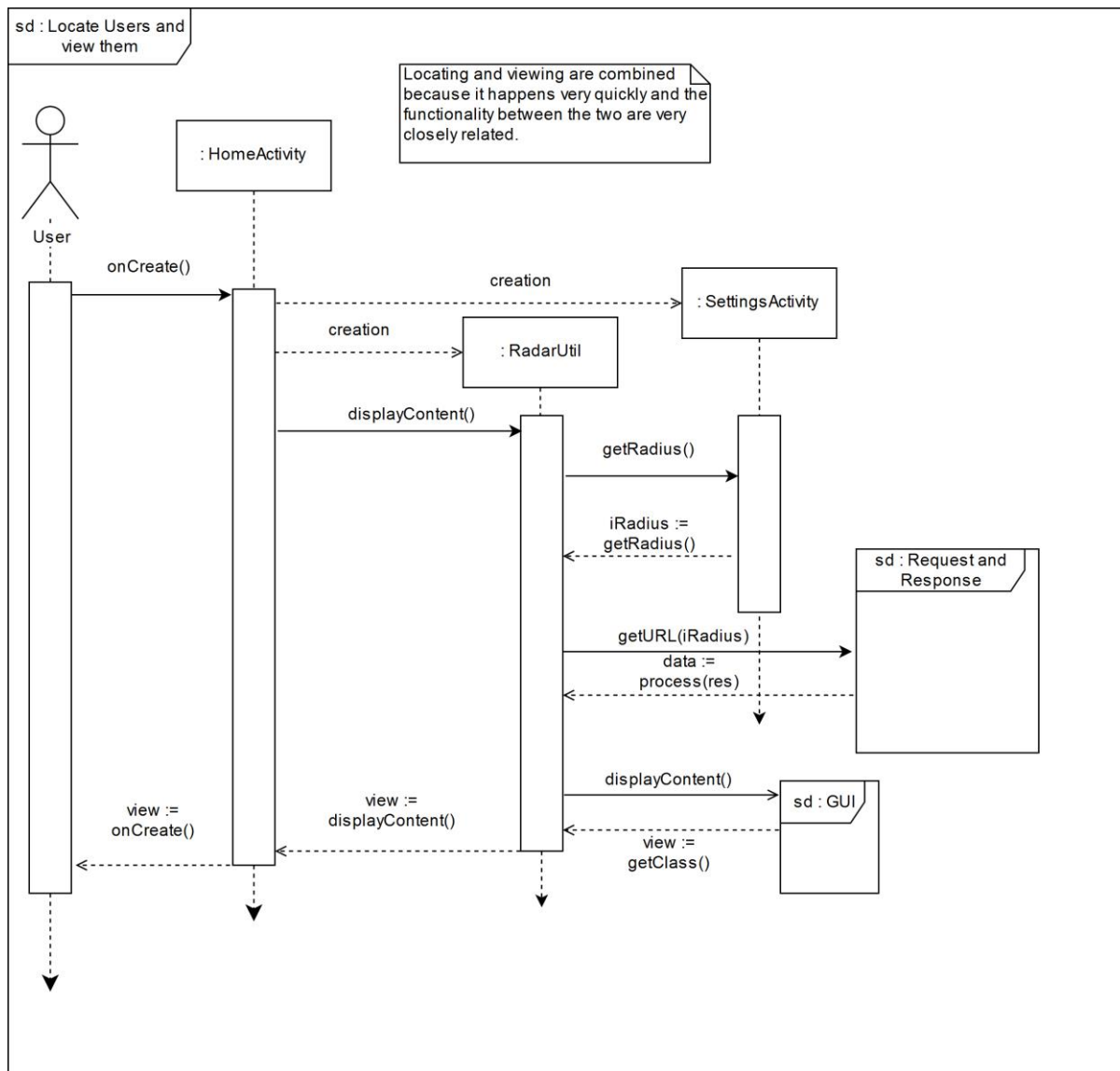
Signature 
Full Name Ronald Lai
Student # 201433999
Date 18/04/2017
Member 4

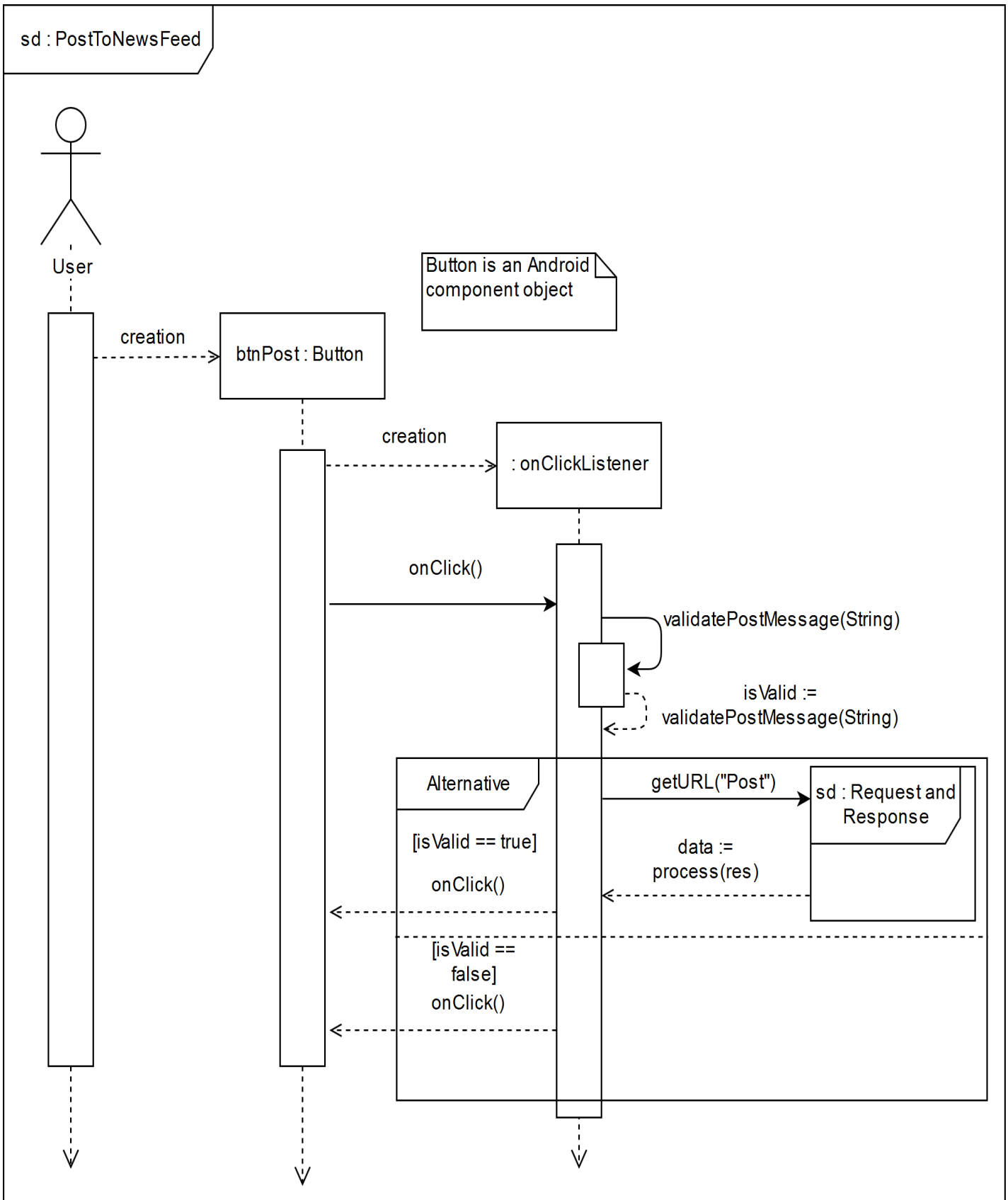
Mobile App Interaction Sequence Diagrams

The GUI and Request and Response are utility sequence diagrams that supports the other diagrams. Request and Response is also used for parts of the website diagrams.

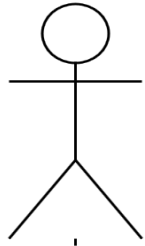








sd : Like Post



User

creation

likePostObj : Post

notifyUser()

getURL("like")

sd : Request and Response

data :=
process(res)

likePost()

isLiked :=
likePost()

displayContent()

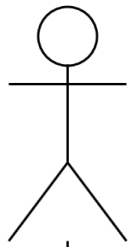
sd : GUI

notify :=
notifyUser()

view :=
getClass()



sd : Flag Post



User

creation

flagPostObj : Post

notifyUser()

getURL("flag")

sd : Request and
Response

data :=
process(res)

flagPost()

isFlagged :=
flagPost()

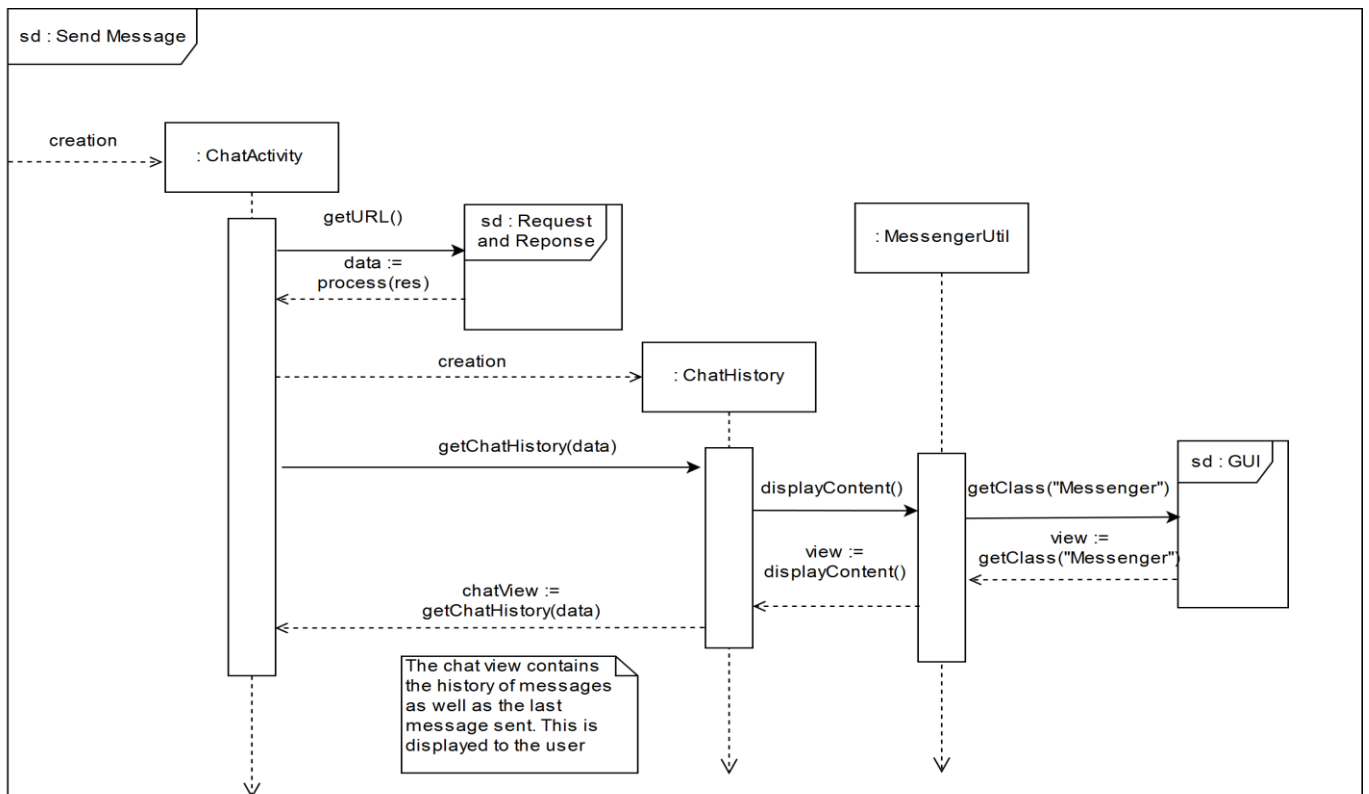
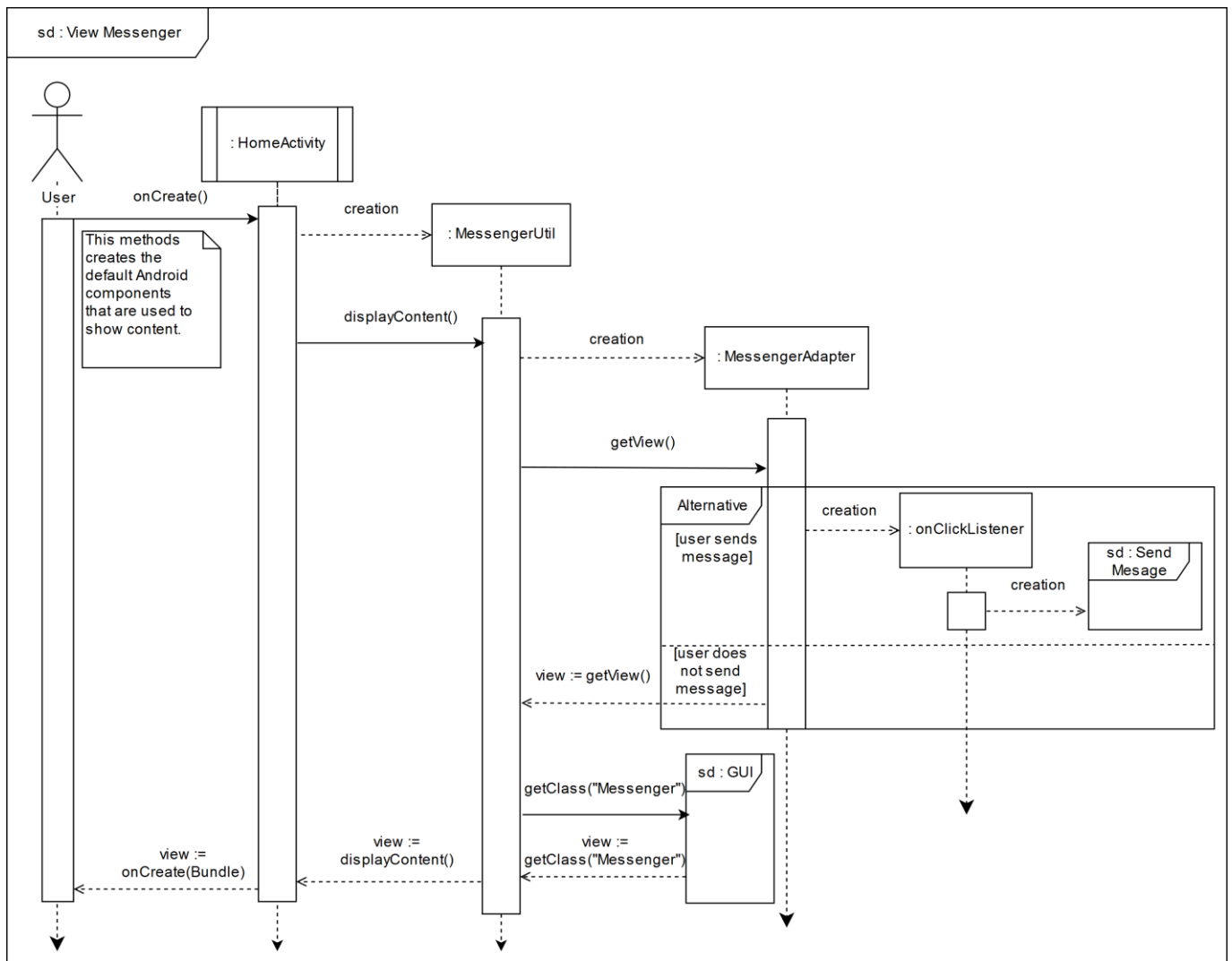
displayContent()

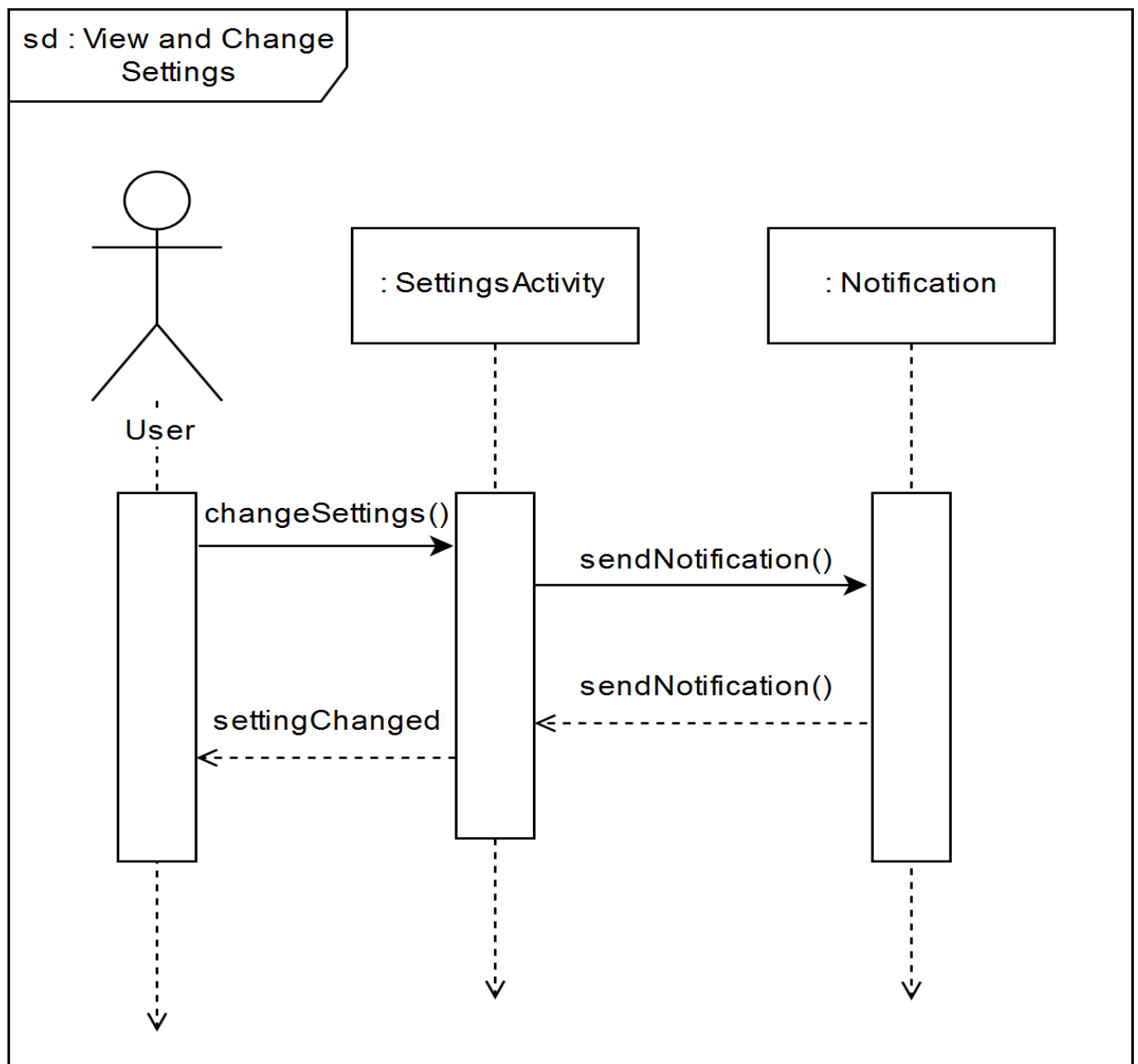
sd : GUI

notify :=
notifyUser()

view :=
getClass()

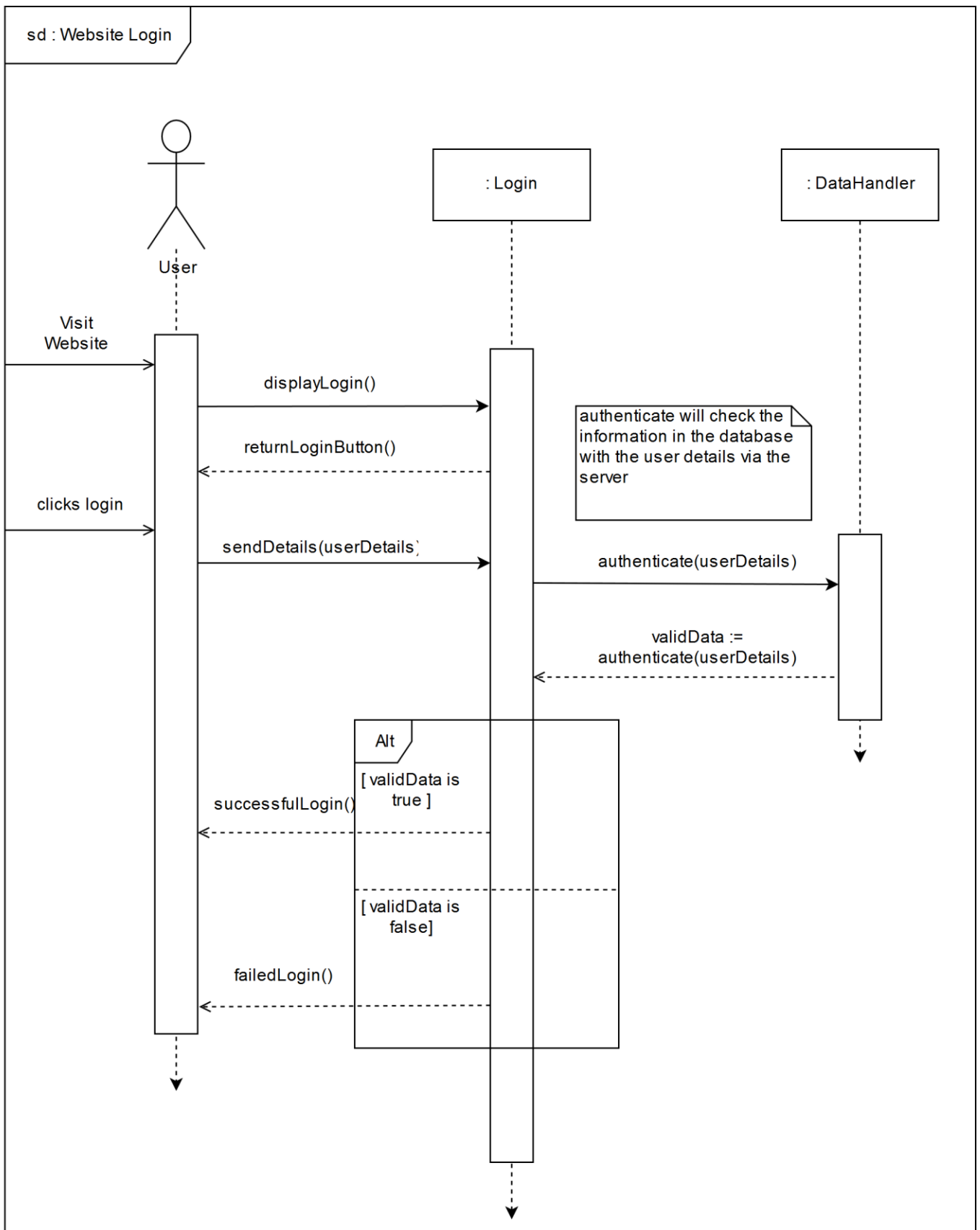




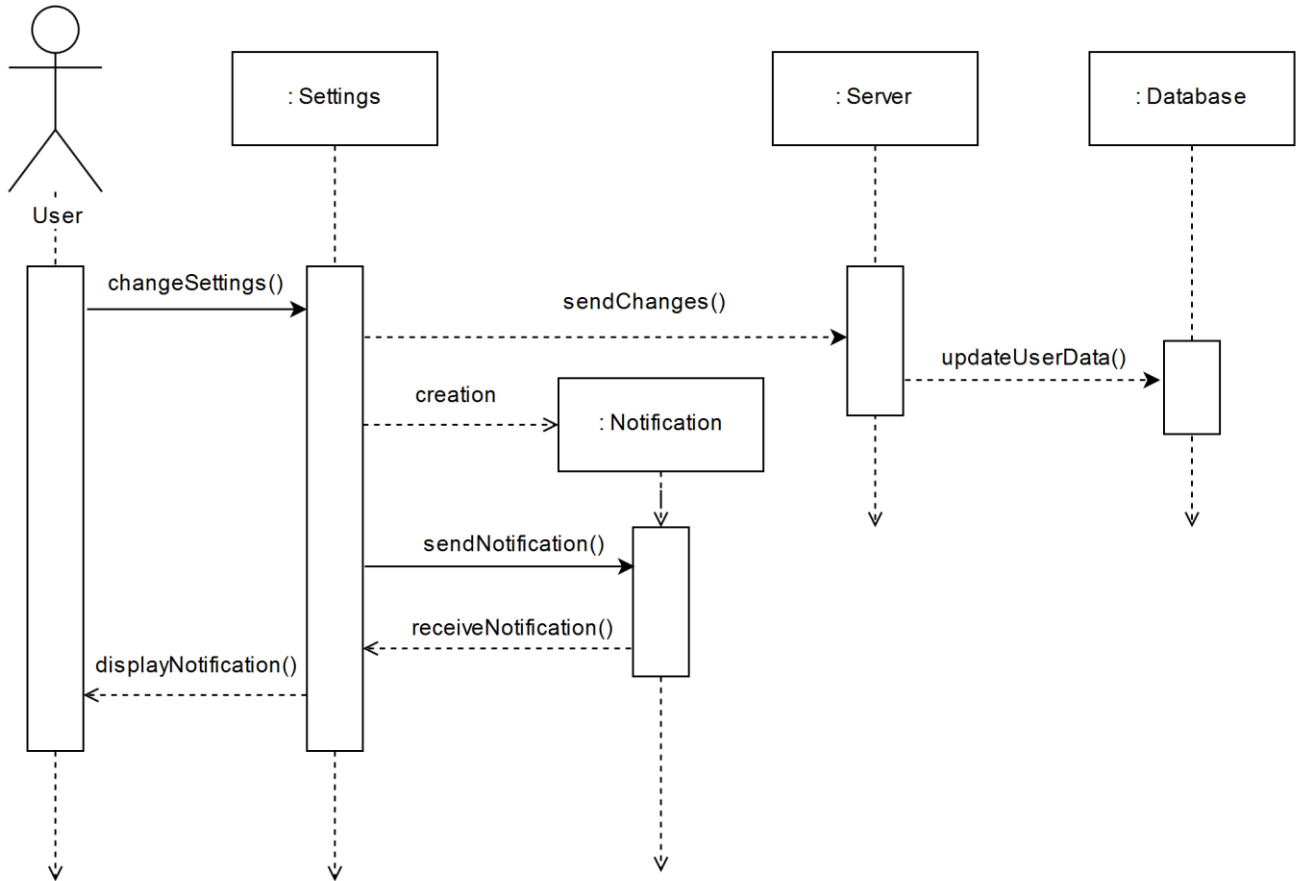


Website Interaction Sequence Diagrams

The like post and flag post diagrams have the same functionality as in the mobile app except they are displayed differently.

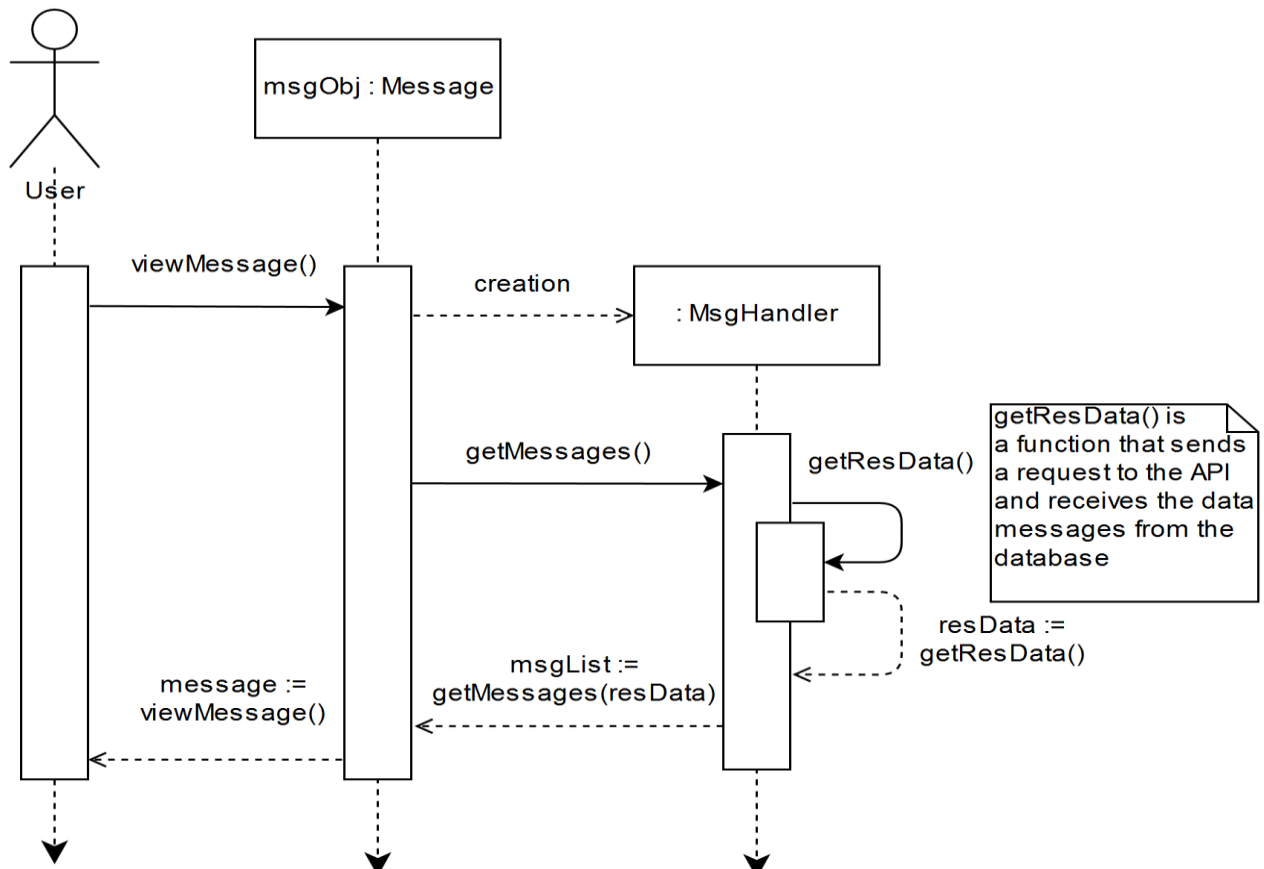


sd : View and Change Settings



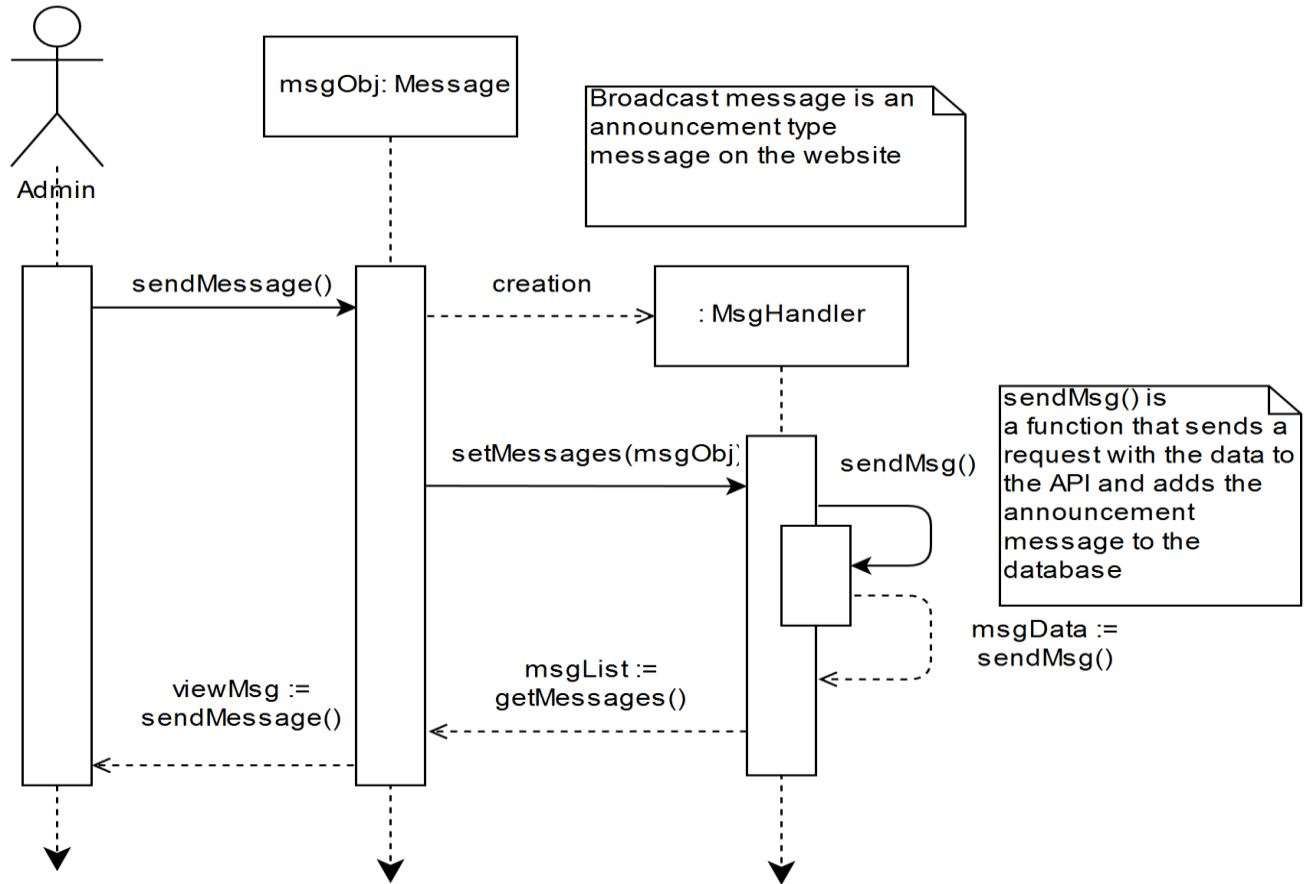
sd : View Broadcast Message

A broadcast message is an announcement that users of the website can see and admins can post these.

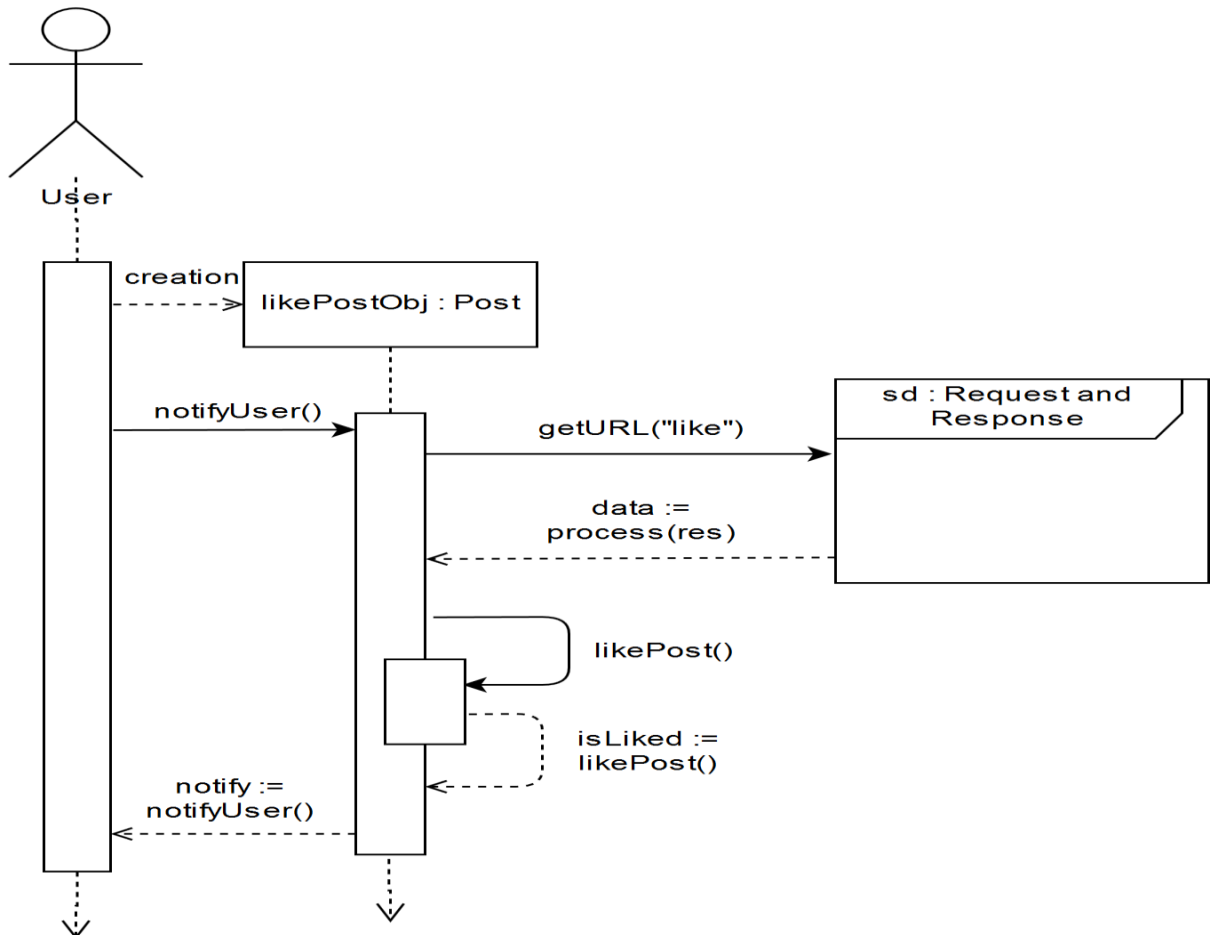


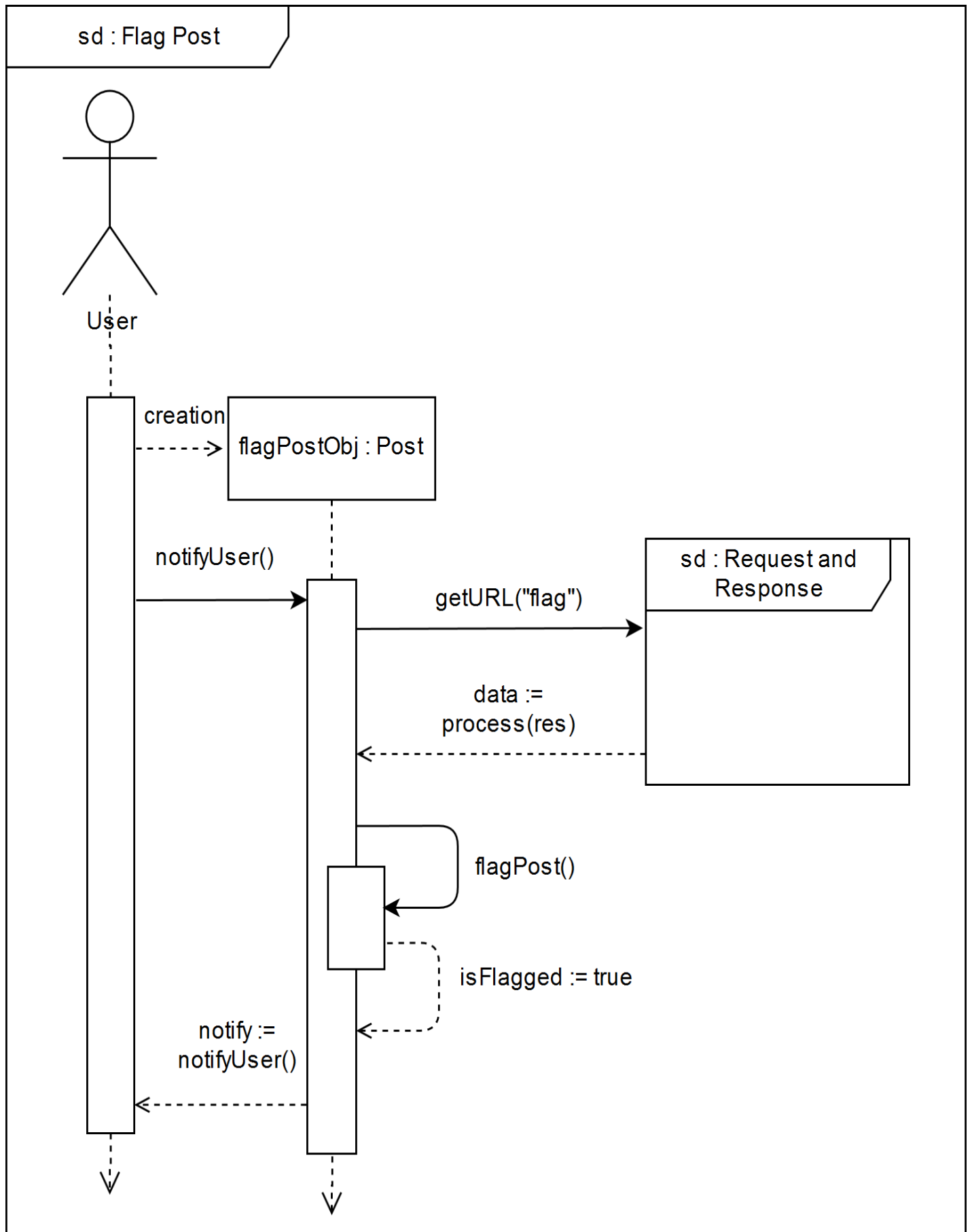
`getResData()` is a function that sends a request to the API and receives the data messages from the database

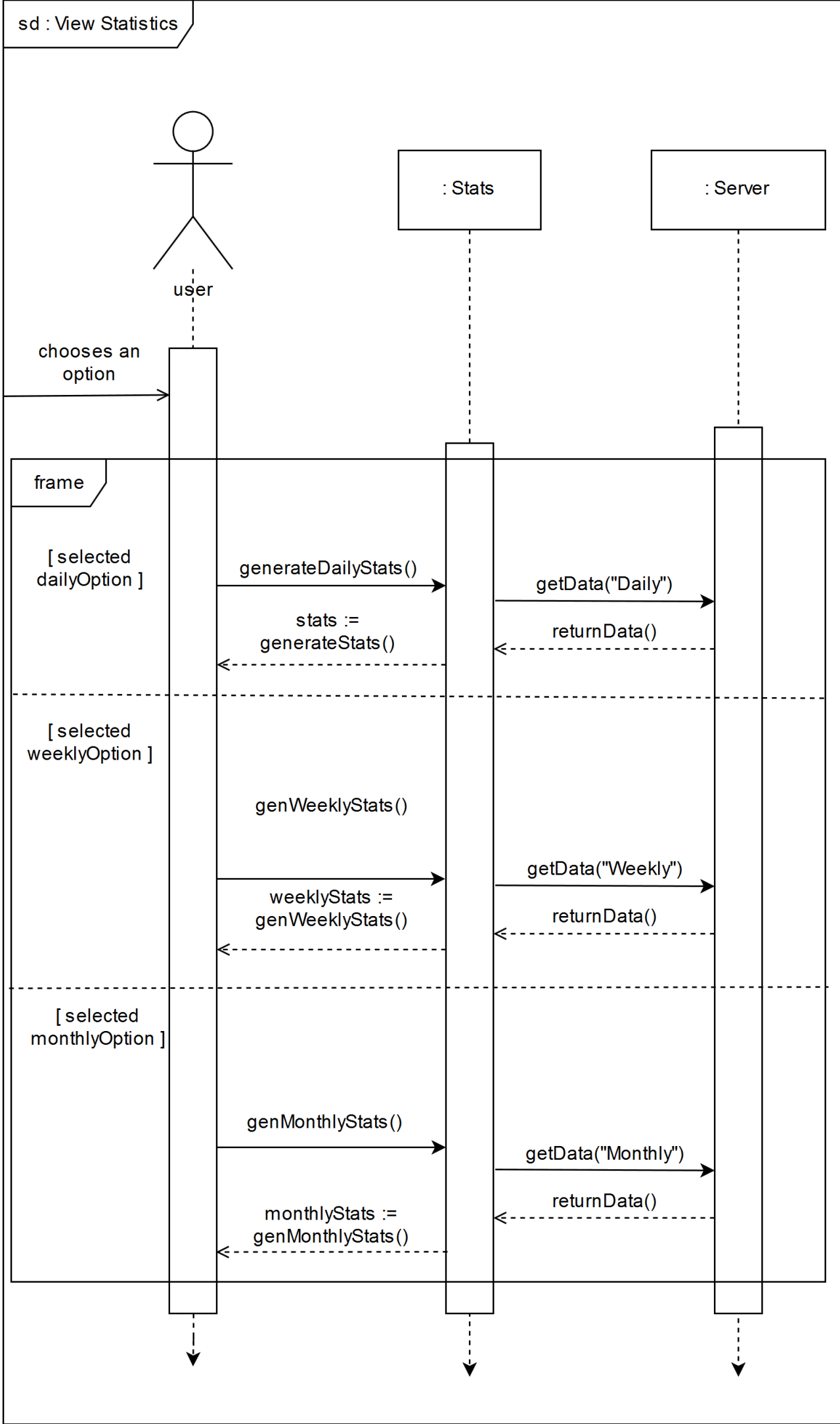
sd : Broadcast Message



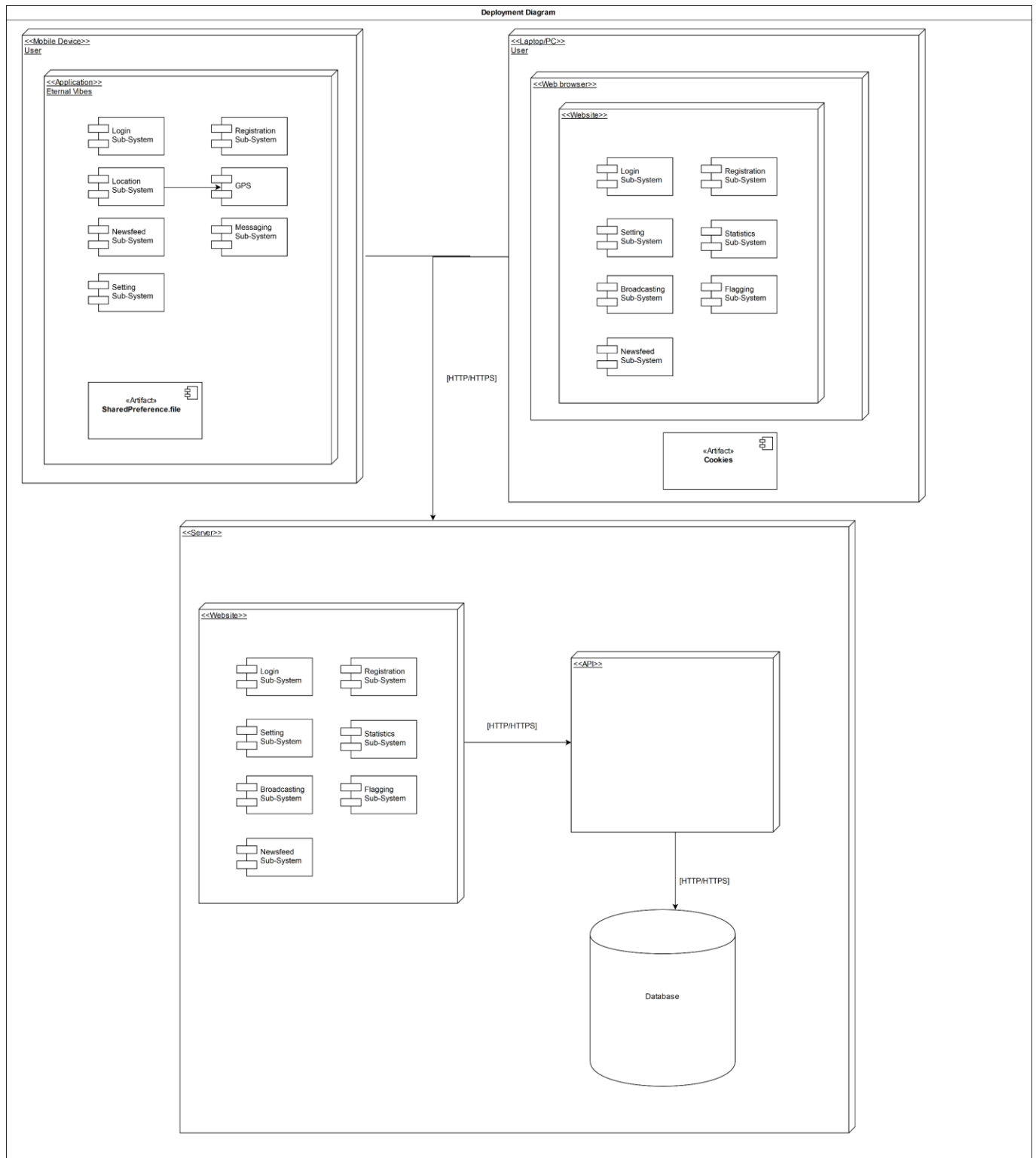
sd : Like Post



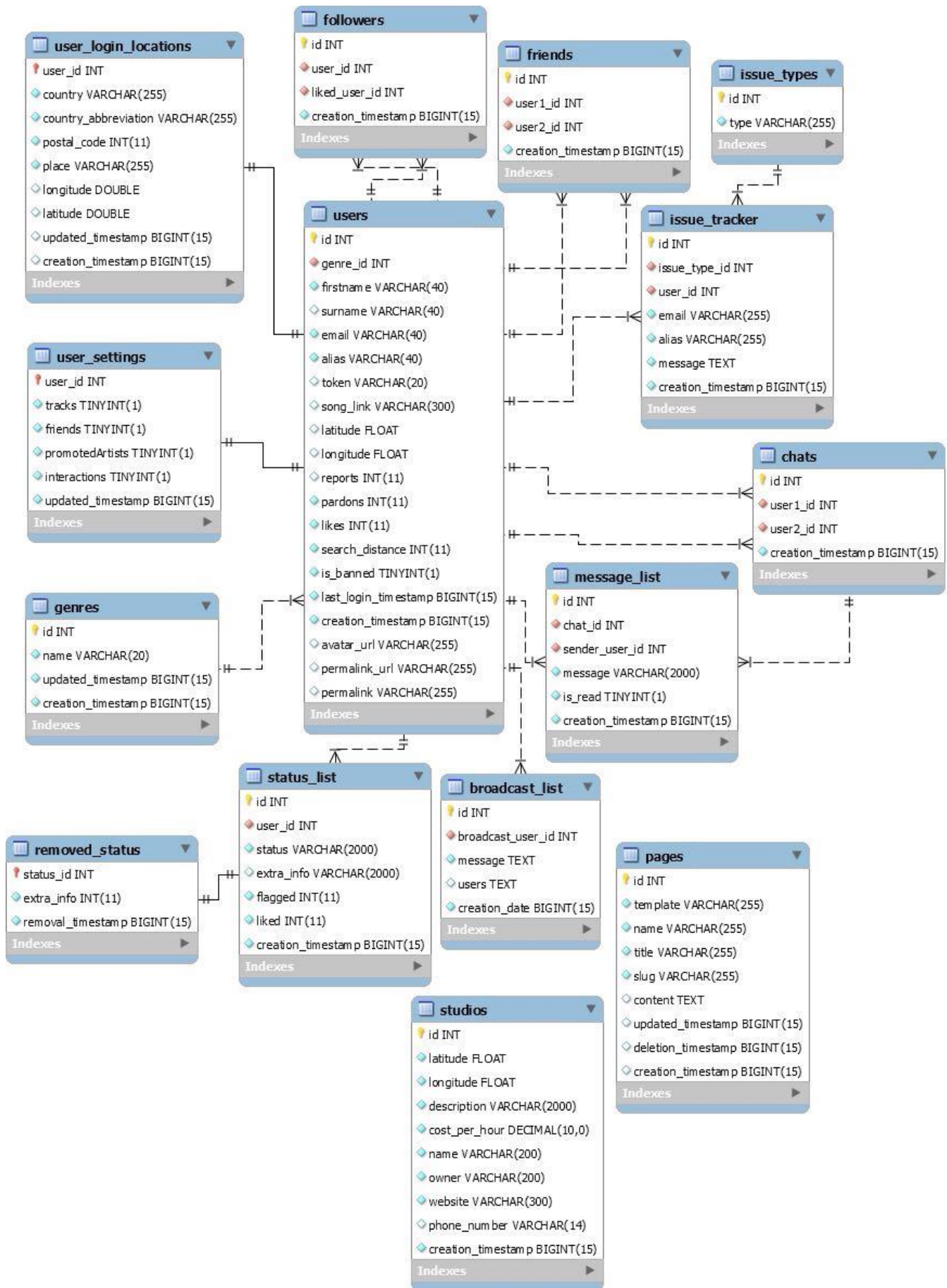


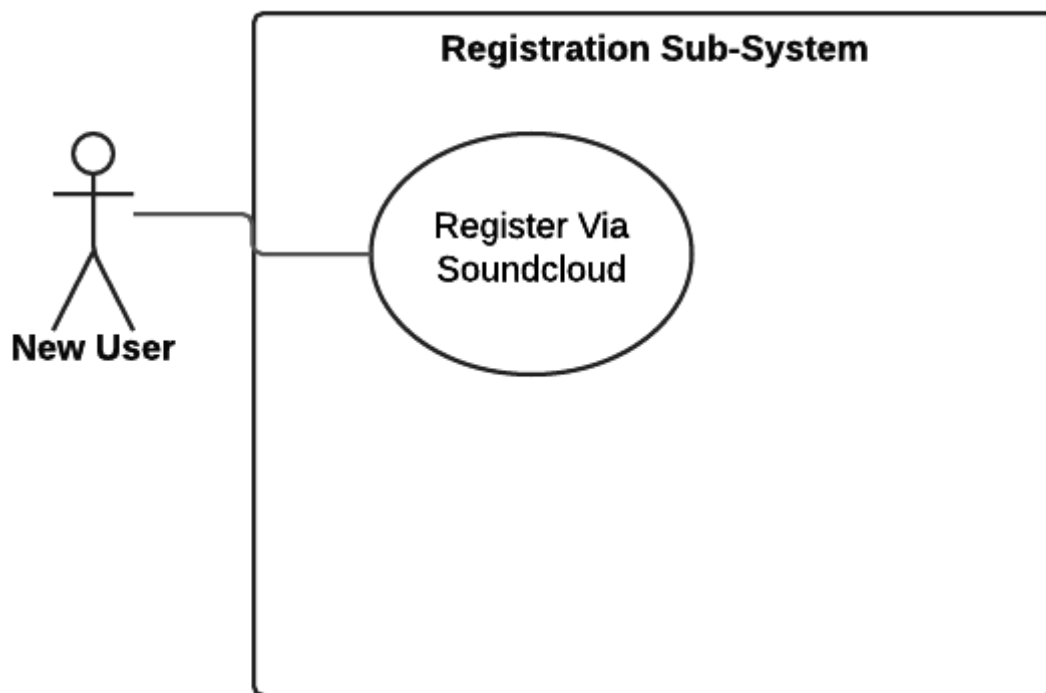


Component and Deployment Diagram



Entity Relationship Diagram





Use case: Register User Via SoundCloud

- 1) User registers using existing SoundCloud account
- 2) System requests account authorization using SoundCloud
- 3) System authorizes SoundCloud account
- 4) System Links account to User
- 5) User Fills in additional registration information
- 6) System creates account

Extensions

1a) User data does not have existing SoundCloud account

1. Redirect user to SoundCloud registration page

1b) User already exists on the system

1. System notifies user
2. User goes to login page

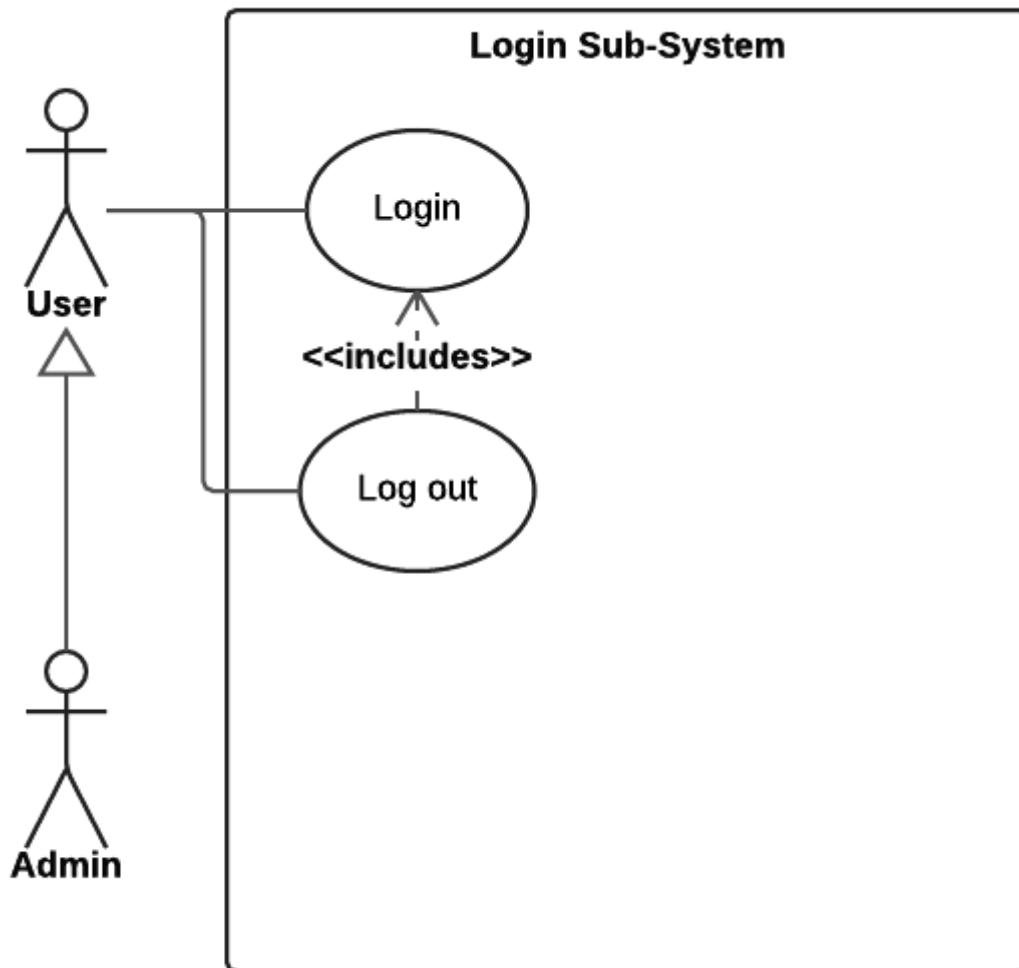
2a) System cannot request authorization User's SoundCloud account

1. System notifies User
2. User may re-enter SoundCloud credentials

3a) Authorization fails (authorization timeout)

1. System notifies User

2. User tries again
- 4a) User inputs incorrect information
1. System notifies user
 2. User re-enters information correctly



Use Case: Login

1. Users/Admin Enters Login Credentials
2. User i/Admin Logged into System
3. System is redirected to User is/Admin dashboard

Extensions

1a) User's/Admin's Login credentials are incorrect

1. System notifies User/Admin

2. User/admin re-enters credentials

1b) User/Admin account does not exist

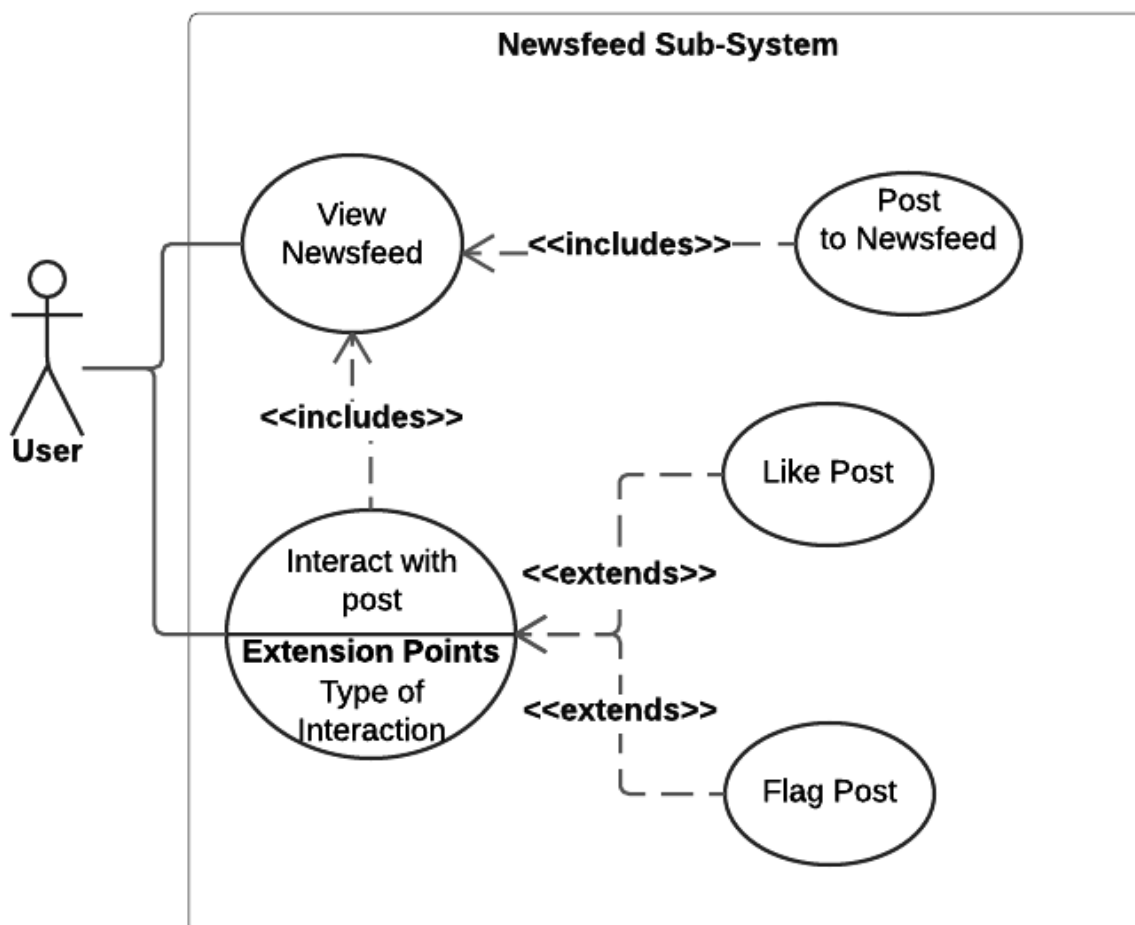
1. Notify User/Admin
2. User/Admin goes to registration page

1c) User/Admin forgets login credentials

1. User/Admin enters email address linked to account
2. System sends an email to User/Admin with their Login credentials
3. User/Admin tries again

Use Case: Logout

1. User/Admin is finished using the system
2. User Logs out of System
3. System redirects User/Admin to login/registration page
4. User closes system



Use Case: View Newsfeed

1. User goes to Newsfeed
2. System loads Newsfeed
3. User browses Newsfeed

Extension

2a) No internet connection

1. System waits for internet connection
2. System loads newsfeed

Use case: Post to Newsfeed

1. User goes to Newsfeed
2. System displays Newsfeed
3. User browses Newsfeed
4. User adds a post
5. System uploads post
6. User's Newsfeed is updated with new posts

Extension

1a) User discards post

1. System discards post

5a) No internet connection

1. Post not uploaded
2. System waits for internet connection
3. System uploads post

Use Case; Interact with post

1. User browses Newsfeed
2. User views an Artist's post
3. User Interacts with post

Use Case: Like Post

1. User browses Newsfeed
2. User views an Artist's post
3. User Interacts with post
4. User Likes/Flags post

Extension

4a) User Likes Post

1. System notifies Artist that their post has been liked

4b) User Flags Post

1. System notifies Admin on flagged post

1. User chooses Artist to message
2. System displays chat between the Users
3. User types a message
4. User sends message
5. System notifies Artist that new message received

Extensions

3a) Message discarded

1. User leaves chat
2. System discards message

4a) No internet connection

1. System notifies user
2. System waits for internet connection
3. System sends message when connection is established

Use case : Receive a message

1. Server sends a notification to User
2. User opens chat with Artist
3. User reads received message
4. User responds to message

Extension

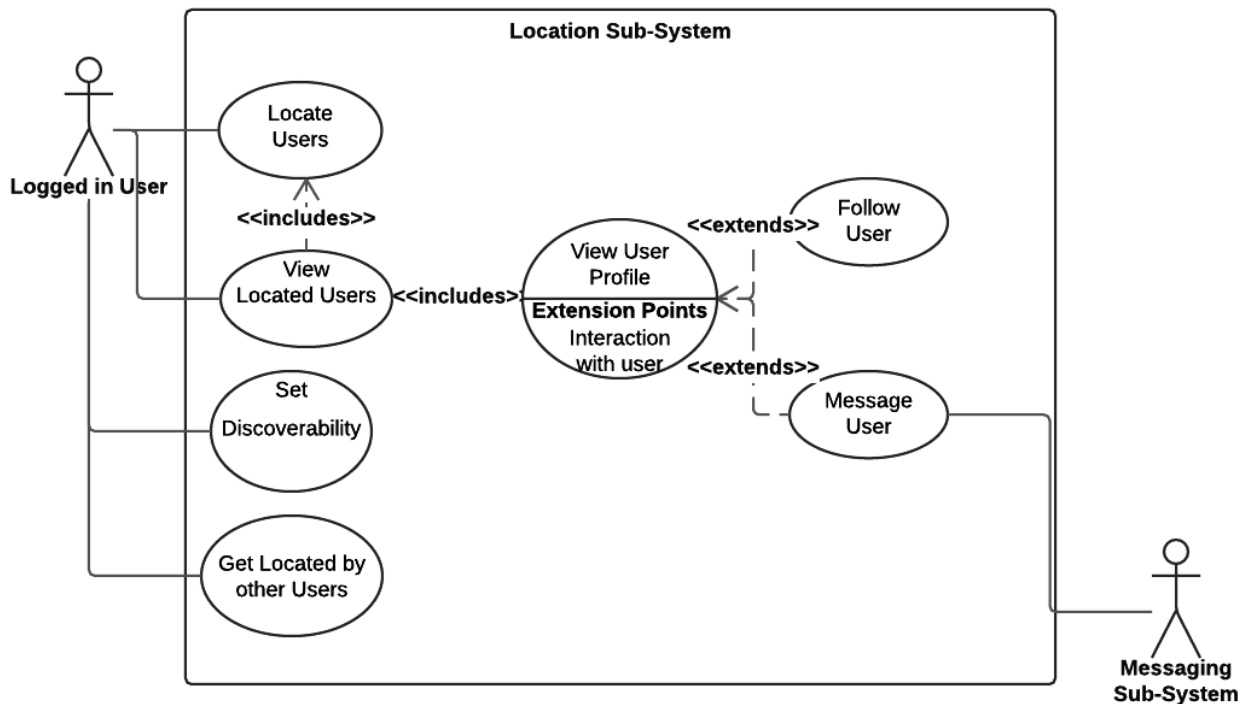
1a) No Internet connection

1. System waits for a connection
2. System sends notification once connection is established

4) User does not respond message

Use Case: Notify User

1. System notifies User that a message has been received
2. System notifies user based on internet connection



Use Case: Set discoverability

1. User sets discoverability on or off
2. System updates User's discoverability status
3. System stores User's location

Extension

2a) No internet connection established

1. System waits for internet connection
2. System updates User's discoverability status once internet connection established

Use Case: Locate User

1. User sets discoverability on
2. System updates User's discoverability status
3. System stores User's location
4. System locates nearby Artists
5. System displays located Artists
6. User views located Artists

Use Case: View User Profile

1. System locates nearby Artists
2. System displays located Artists
3. User views located Artists
4. User views a Artist's profile

Use Case: Follow User

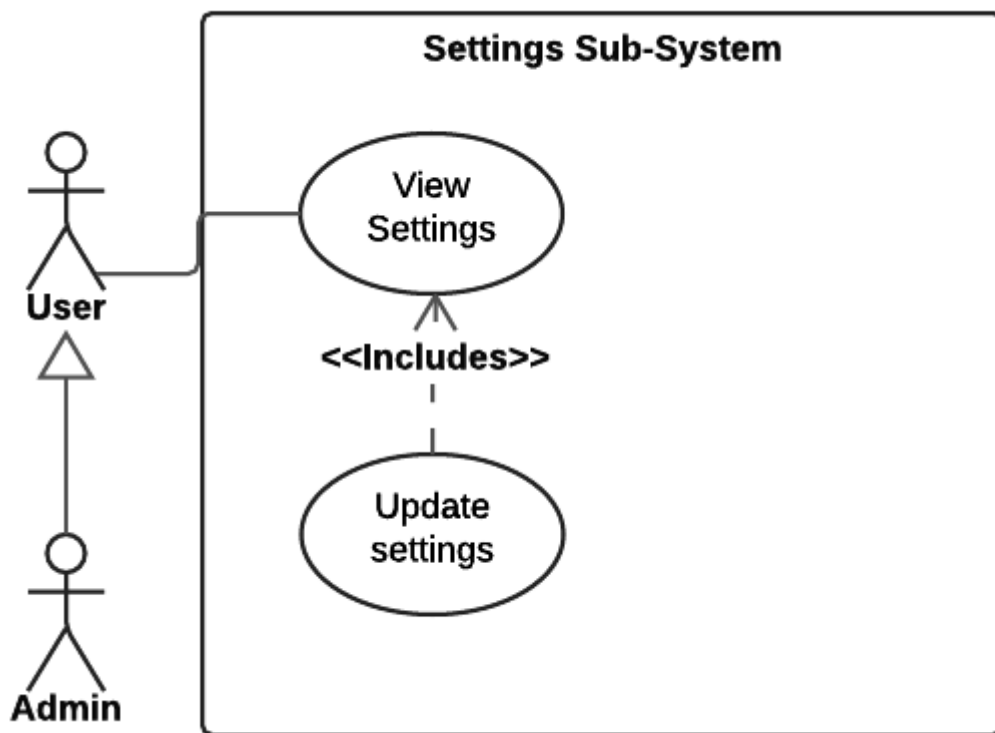
1. System locates nearby Artists
2. System displays located Artists
3. User views located Artists
4. User views a Artist's profile
5. User Follows Artist
6. System updates User's following
7. System updates Artist's Followers
8. System updates User's newsfeed
9. System notifies Artist

Use Case: Message User

1. System locates nearby Artists
2. System locates nearby Artists
3. System displays located Artists
4. User views located Artists
5. User views a Artist's profile
6. User sends message to Artist
7. System notifies Artist
8. System stores conversation

Use Case; Get Located by other Users

1. User's discovery is set on
2. System updates User's discoverability status
3. System stores User's location
4. Artists locate User
5. Artists interact with user
6. System notifies User

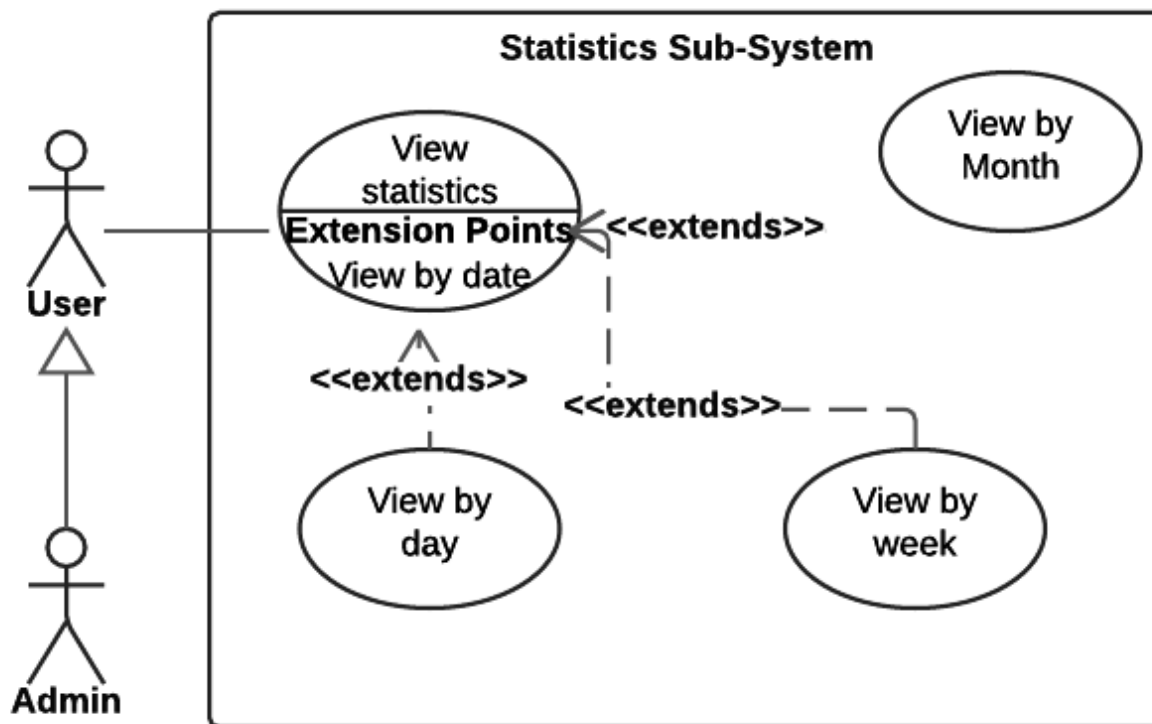


Use Case: View settings

1. User opens settings
2. System displays settings

Use Case: Update settings

1. User opens settings
2. System displays settings
3. System validates settings
4. System saves settings



Use Case: View statistics

1. System generates statistics
2. User/Admin views statistics

Use case: View by day

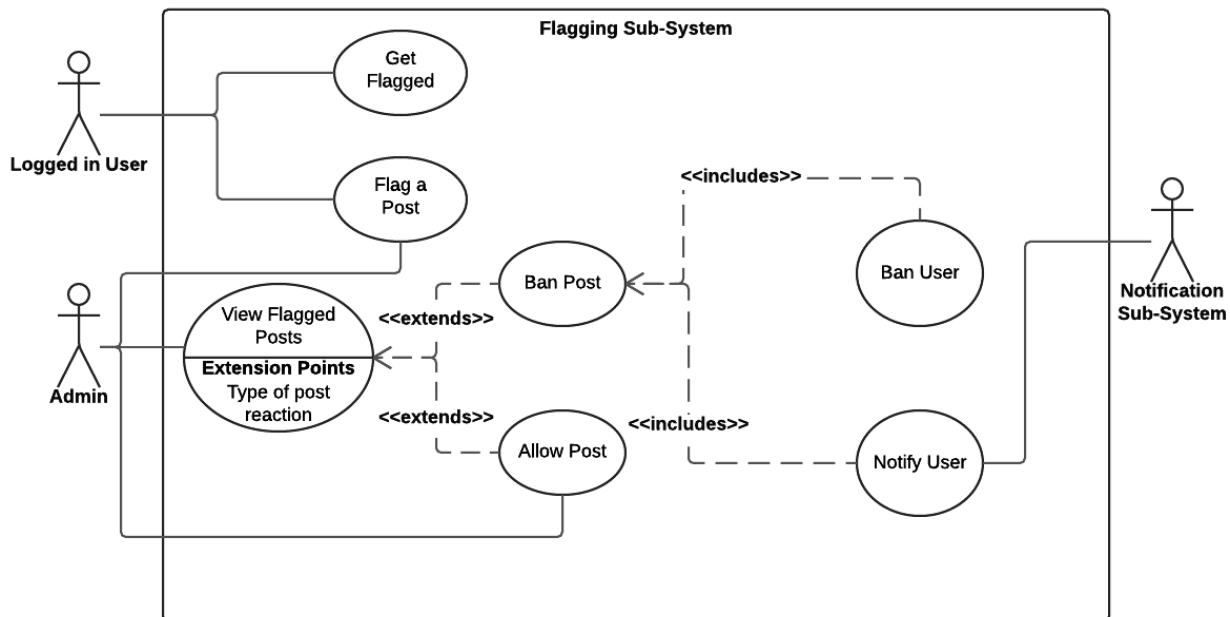
1. System generates statistics
2. User/Admin views statistics
3. User /Admin requests statistics view by day
4. System displays statistics by day

Use case: View by week

1. System generates statistics
2. User/Admin views statistics
3. User /Admin requests statistics view by week
4. System displays statistics by week

Use case: View by month

1. System generates statistics
2. User/Admin views statistics
3. User /Admin requests statistics view by month
4. System displays statistics by month



Use Case: Flag a post

1. User/Admin view an Artist post
2. User/Admin Flags post
3. System updates flagged posts
4. System updates how many times post has been flagged
5. System notifies Admin

Extension

2a) Cancel post flag

1. System updates flagged posts
2. System updates how many times post has been flagged

Use Case: Get Flagged

1. System flags User's post based on other Artist's opinion
2. System stores post as flagged
3. System notifies Admin

Use case: View flagged Posts

1. Admin selects flagged posts
2. System displays all flagged posts

Extensions

2. No flagged posts available

Use Case: Ban Post

1. Admin selects flagged posts
2. System displays all flagged posts
3. Admin Bans Post
4. System Removes Post from User's newsfeed and Artist's who follow User
5. System notifies user
6. System updates flagged posts

Extension

3a) Admin cancels ban

1. System keeps post as flagged

Use Case: Ban User

1. Admin selects flagged posts
2. System displays all flagged posts
3. Admin Bans Post
4. System Removes Post from User's newsfeed and Artists who follow User
5. System notifies user
6. System updates flagged posts
7. Admin Bans User
8. System notifies user on Ban

Extensions

6a) Admin cancels ban

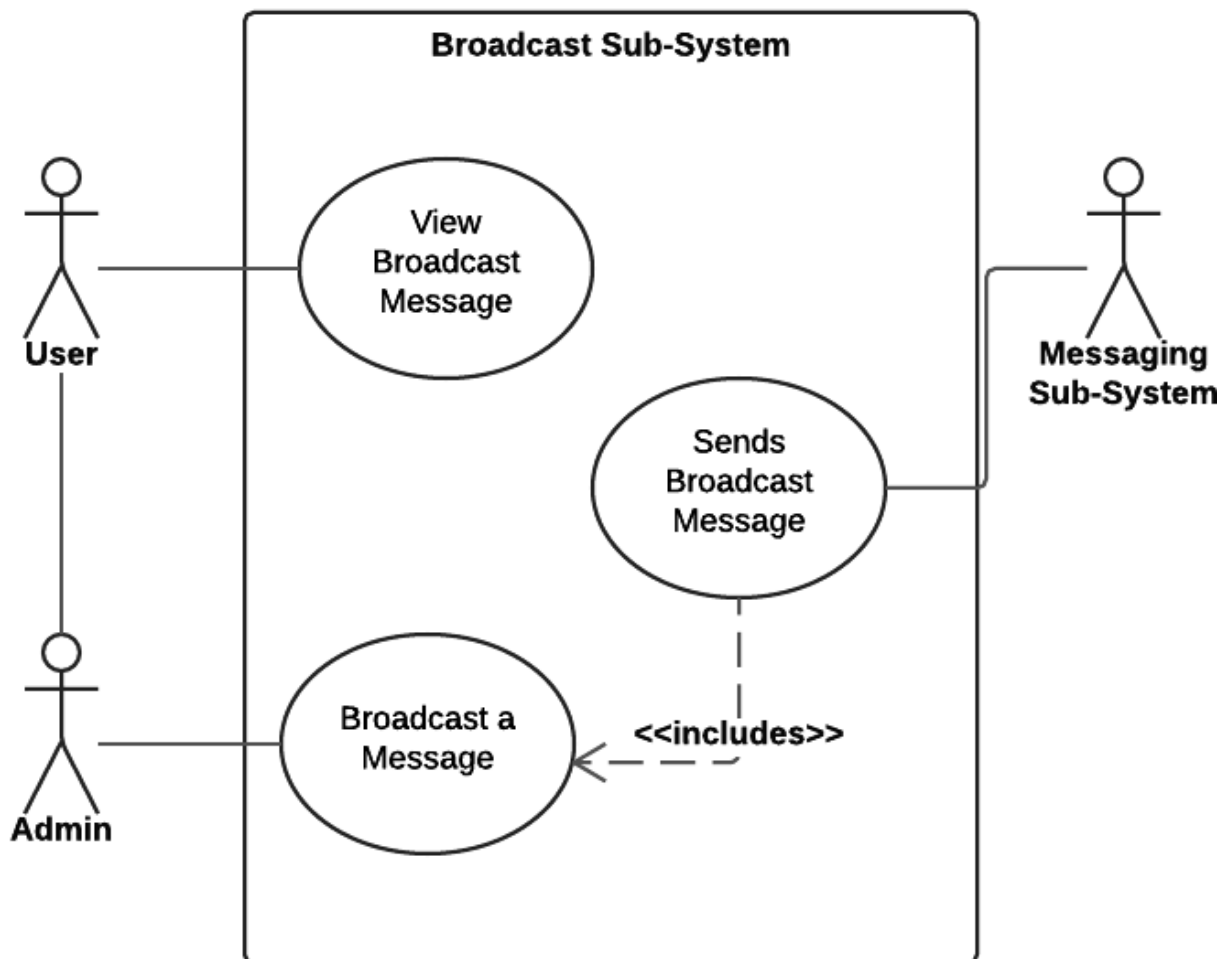
Use Case: Allow Post

1. Admin selects flagged posts
2. System displays all flagged posts
3. Admin approves post and removes flag
4. System keeps post on User's newsfeed and Artists who follow User
5. System updates flagged posts

Extensions

3a) Admin cancels approval

1. System keeps post as flagged



Use Case: View Broadcast Message

1. System sends broadcast to User
2. System notifies User
3. User views broadcast

Use Case: Broadcast a message

1. Admin sends broadcasts a message to list of Users
2. System sends message to list of Users

Extension

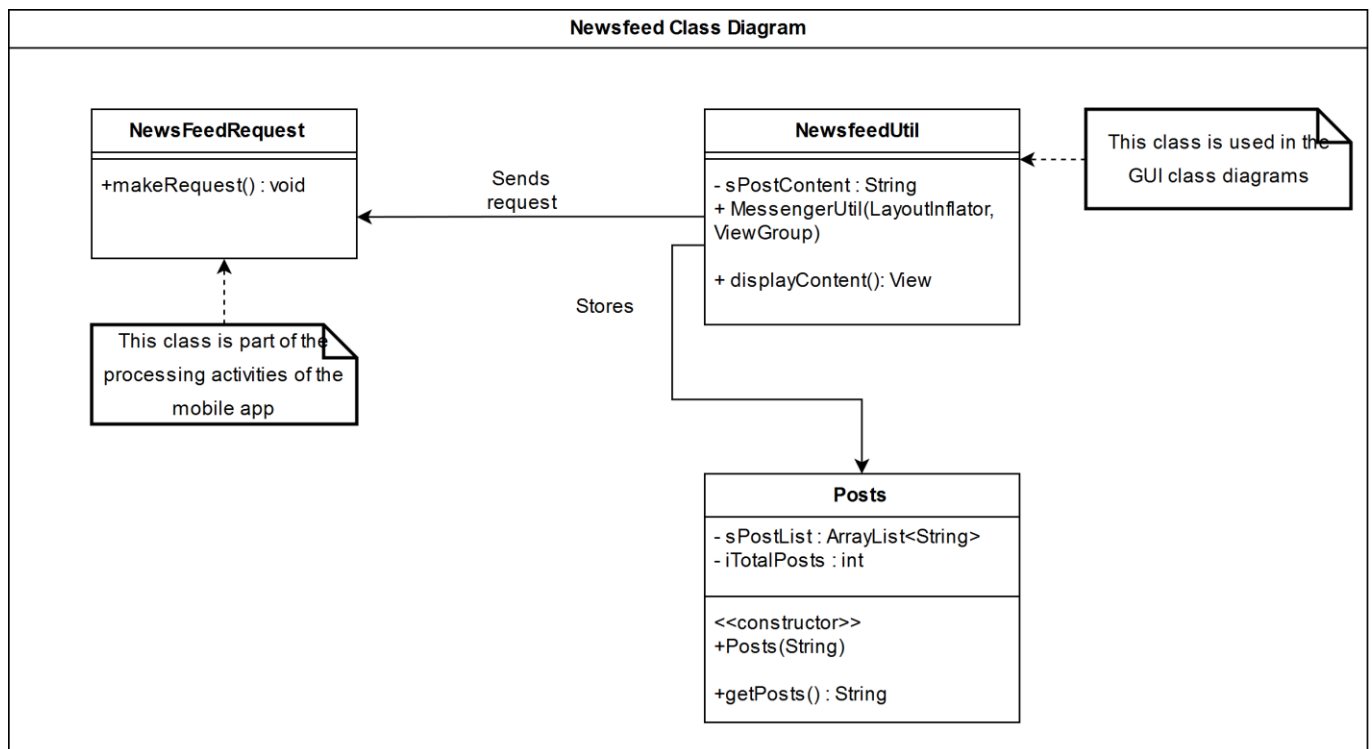
1a) Cancel broadcast

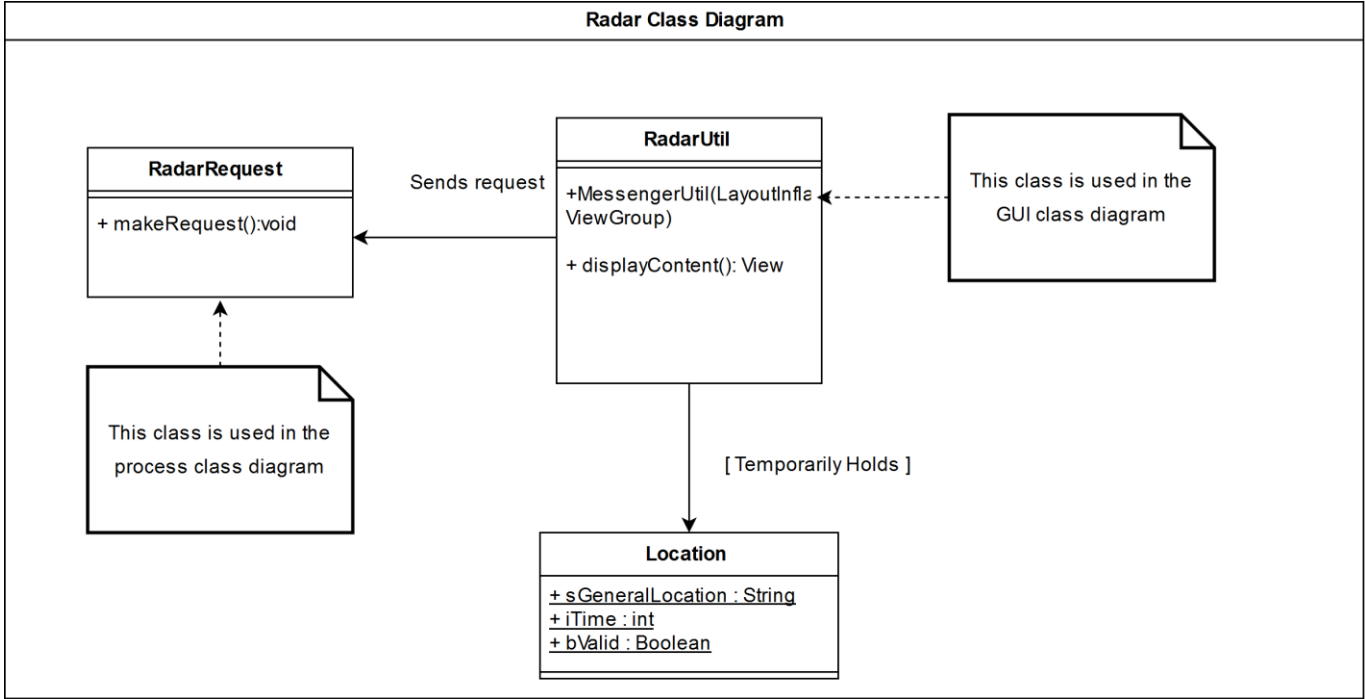
1. System discards broadcast

Use Case: Send Broadcast Message

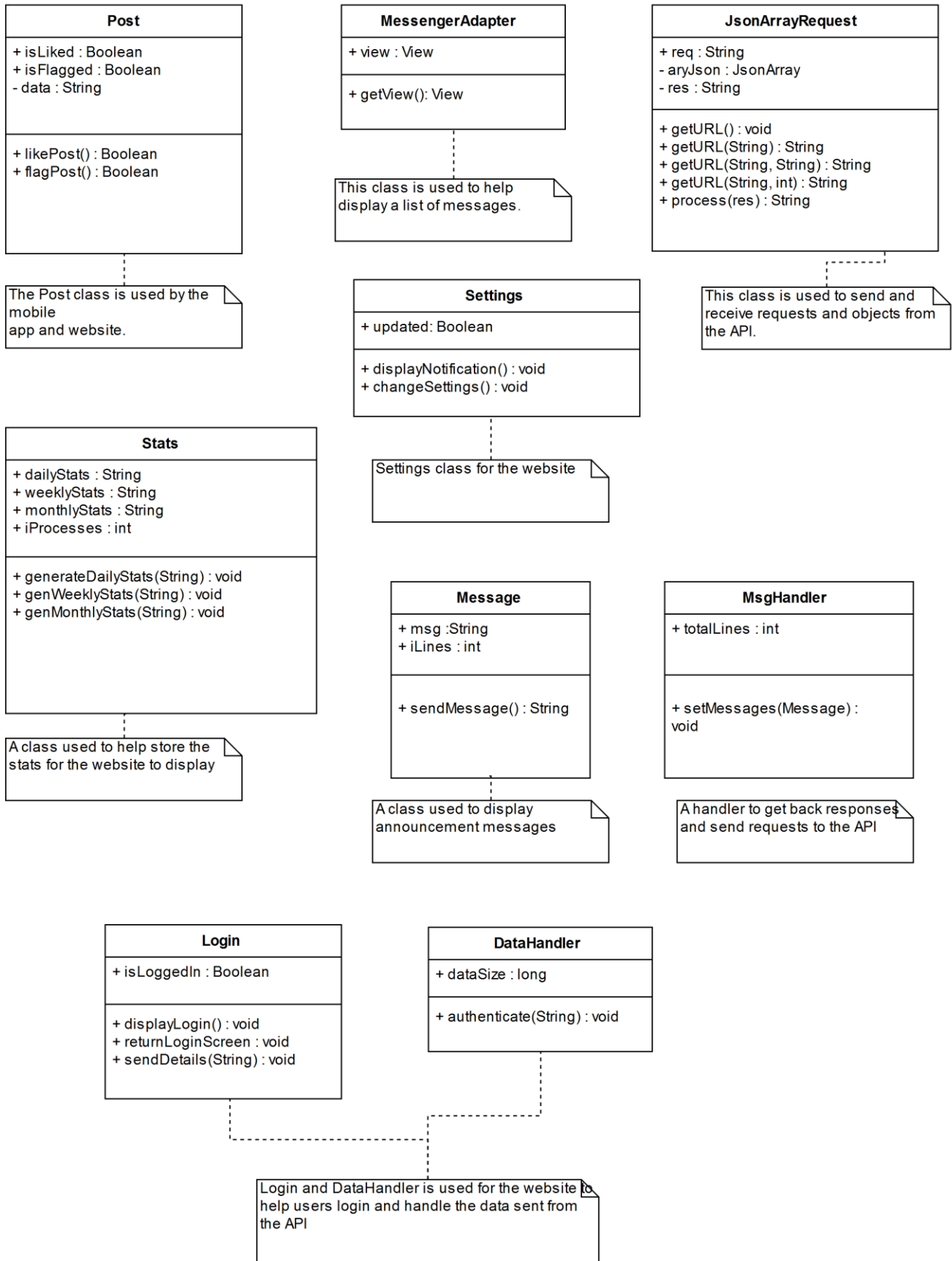
1. Admin sends broadcasts a message to list of Users
2. System sends message to list of Users
3. System notifies Users

Appendix - Class Diagrams

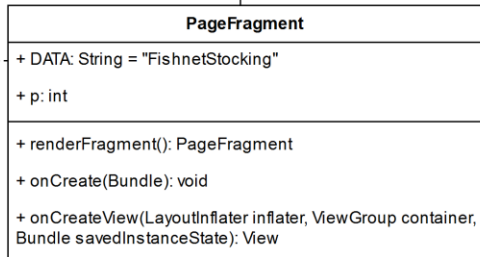
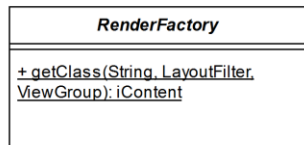
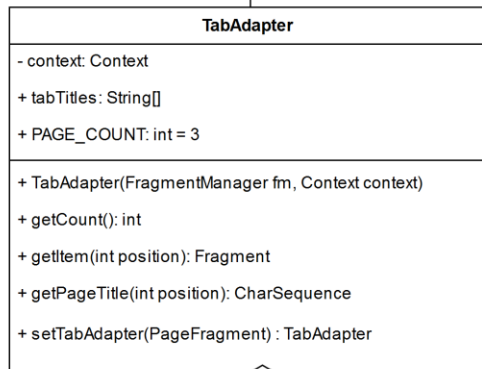
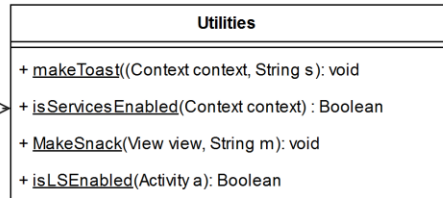
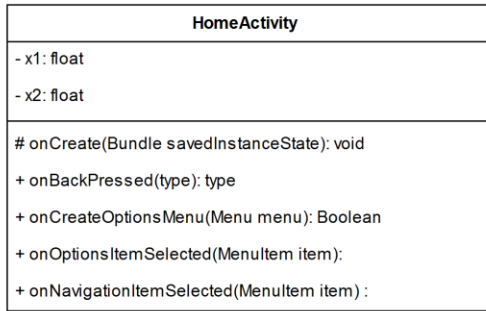
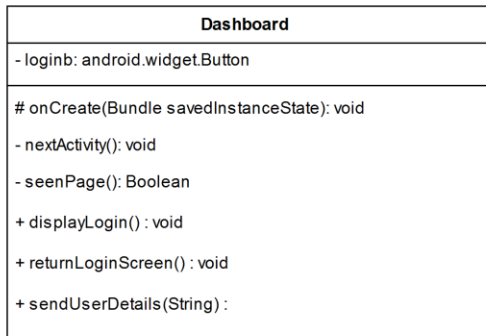




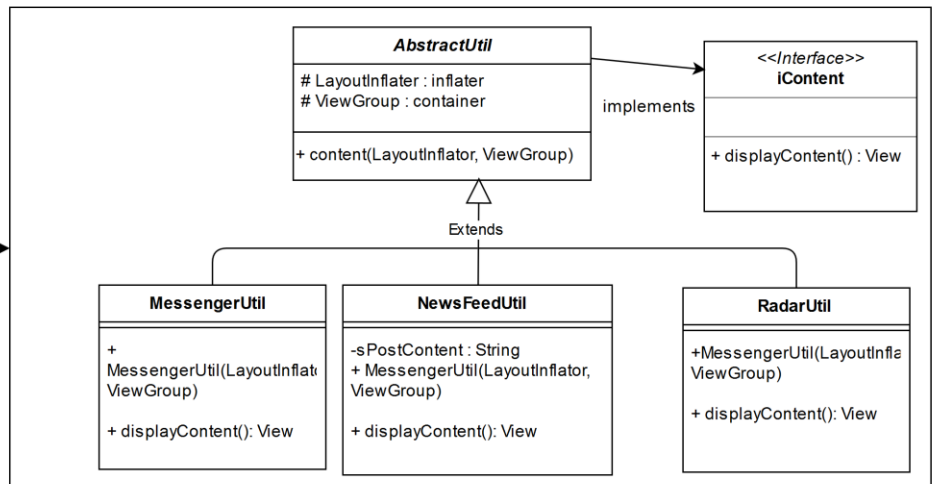
Appendix Classes for Mobile App and Website

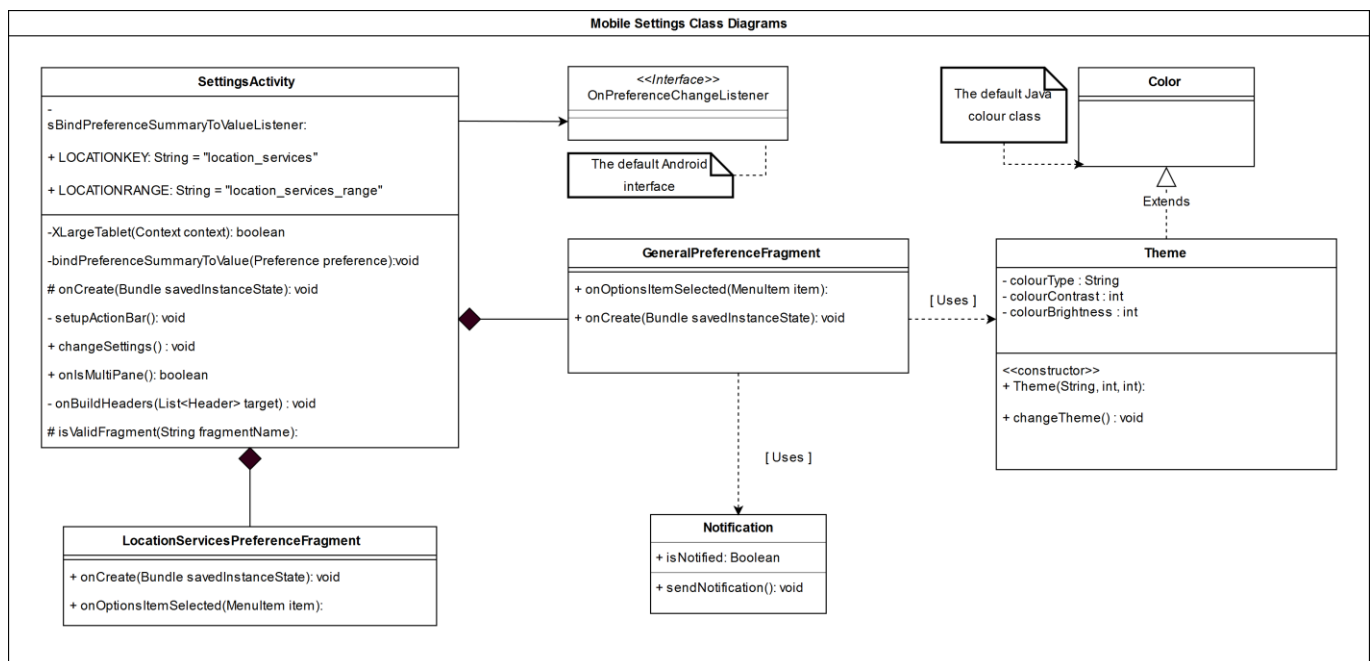
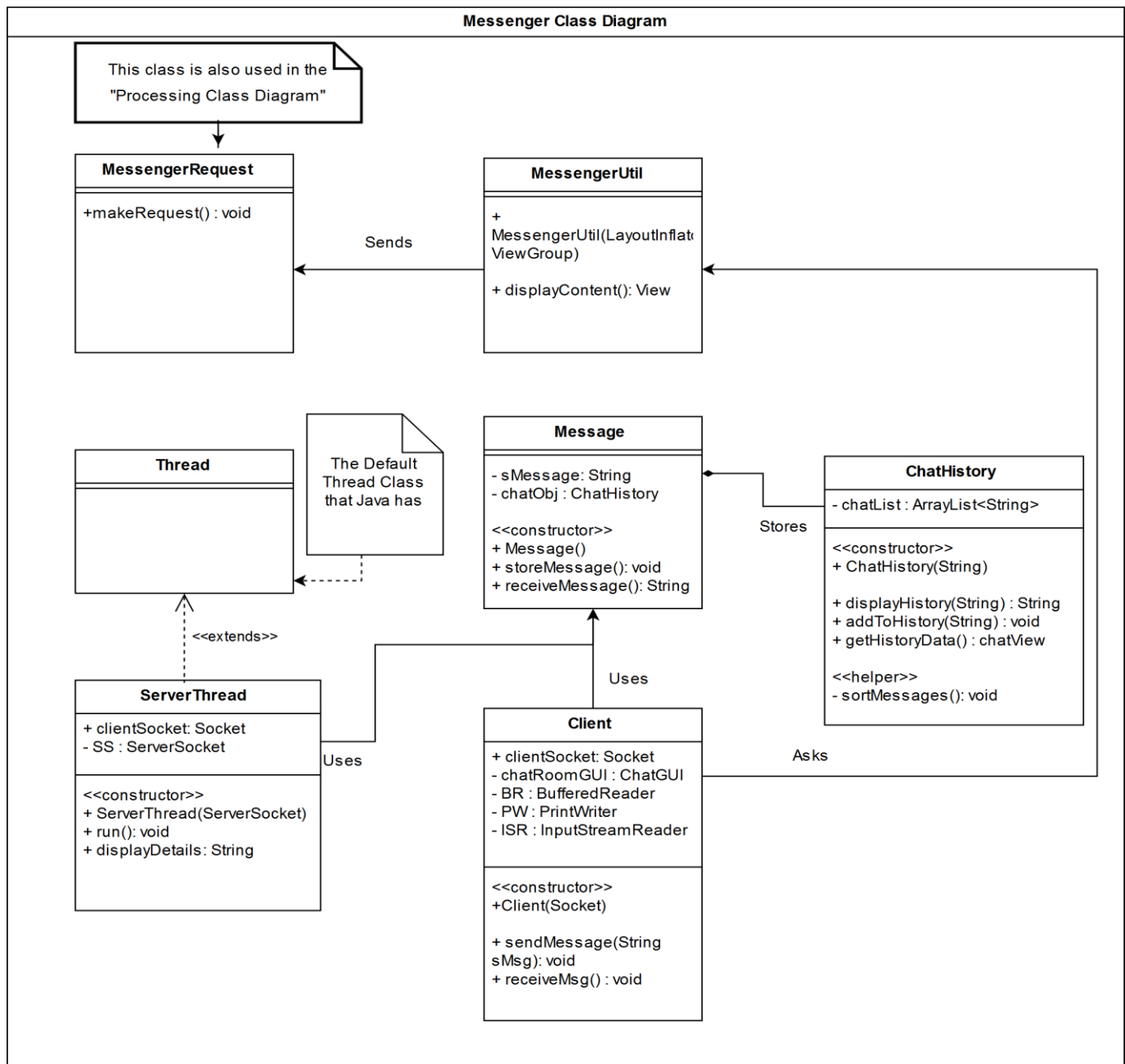


Mobile App GUI Class Diagram



Creates





End of Document