

**Team:**

Ryan Rouleau  
Ryan Turner  
Jacob Crawford  
Brent Dagdagan

**Title:** Clean Community

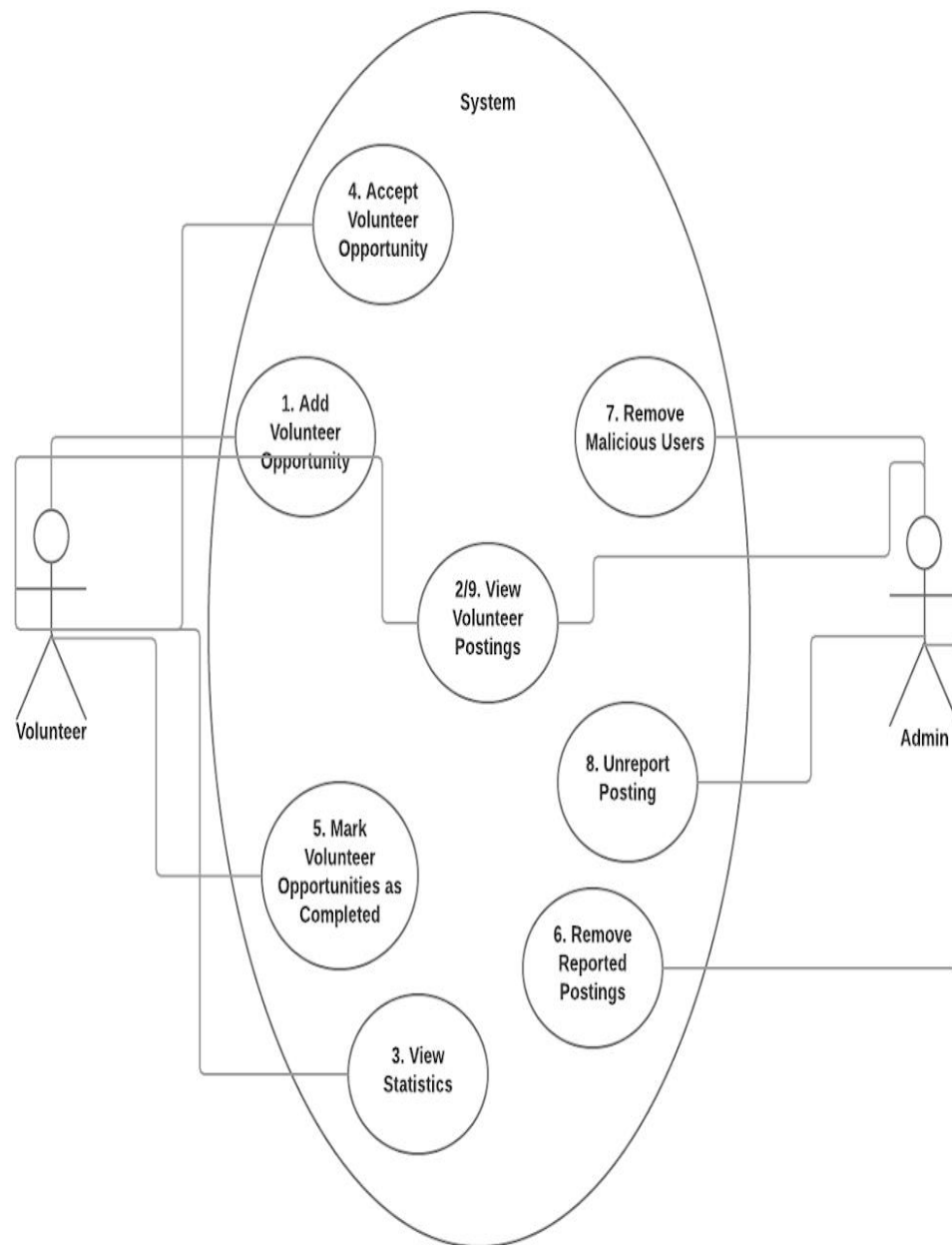
**Project Summary:** A website that allows volunteers to report issues and potential volunteer opportunities around their city that other volunteers can take on and get credit for their good deed.

**Project Requirements:**

1. As a volunteer, I want to be able to create a posting for a volunteering area so that others know where volunteer work can be done.
2. As a volunteer, I want to be able to view volunteer postings so that I can choose opportunities to complete.
3. As a volunteer I want to be able to view statistics on how many postings I and other volunteers have completed.
4. As a volunteer I want to be able to accept the postings by other users so that others know that I am completing the work.
5. As a volunteer, I want to be able to mark other postings as completed so that others know that the work is completed.
6. As an admin, I want to be able to take down reported postings so that volunteers do not see postings that have been reported.
7. As an admin, I want to be able to remove malicious users so that they can no longer post on the site.
8. As an admin, I want to be able to unreport postings that do not violate the rules so that volunteers do not abuse the report system.
9. As an admin, I want to be able to view volunteer postings so that I can monitor the work that the volunteers are posting.
10. Interface: The system shall be interacted with through an API conforming to RESTful standards.
11. Usability: The user interface shall be intuitive and easy to use.

**Data Storage**

MySQL via JDBC



## Use Cases:

<b>Use Case ID:</b>	UC-01
<b>Use Case Name:</b>	Create Postings
<b>Description:</b>	Volunteers can create postings for a volunteer area so that other users know what work can be done.

<b>Actors:</b>	Volunteer																
<b>Pre-conditions:</b>	Volunteer has an account and is logged in																
<b>Post-Conditions:</b>	A new post has been created																
<b>Frequency of Use:</b>	Daily by Volunteers																
<b>Flow of Events:</b>	<table><tr><th></th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Click Create Posting Button.</td><td>Displays volunteer form.</td></tr><tr><td>2</td><td>Fills in form with information regarding volunteer opportunity.</td><td></td></tr><tr><td>3</td><td>Click Post Button.</td><td>Validate form, display success dialog.</td></tr><tr><td>4</td><td></td><td>Displays newly created posting on map.</td></tr></table>			Actor Action	System Response	1	Click Create Posting Button.	Displays volunteer form.	2	Fills in form with information regarding volunteer opportunity.		3	Click Post Button.	Validate form, display success dialog.	4		Displays newly created posting on map.
	Actor Action	System Response															
1	Click Create Posting Button.	Displays volunteer form.															
2	Fills in form with information regarding volunteer opportunity.																
3	Click Post Button.	Validate form, display success dialog.															
4		Displays newly created posting on map.															
<b>Variations:</b>																	
<b>Exceptions:</b>	3. Error Message: Form is not valid																
<b>Developer Notes:</b>																	

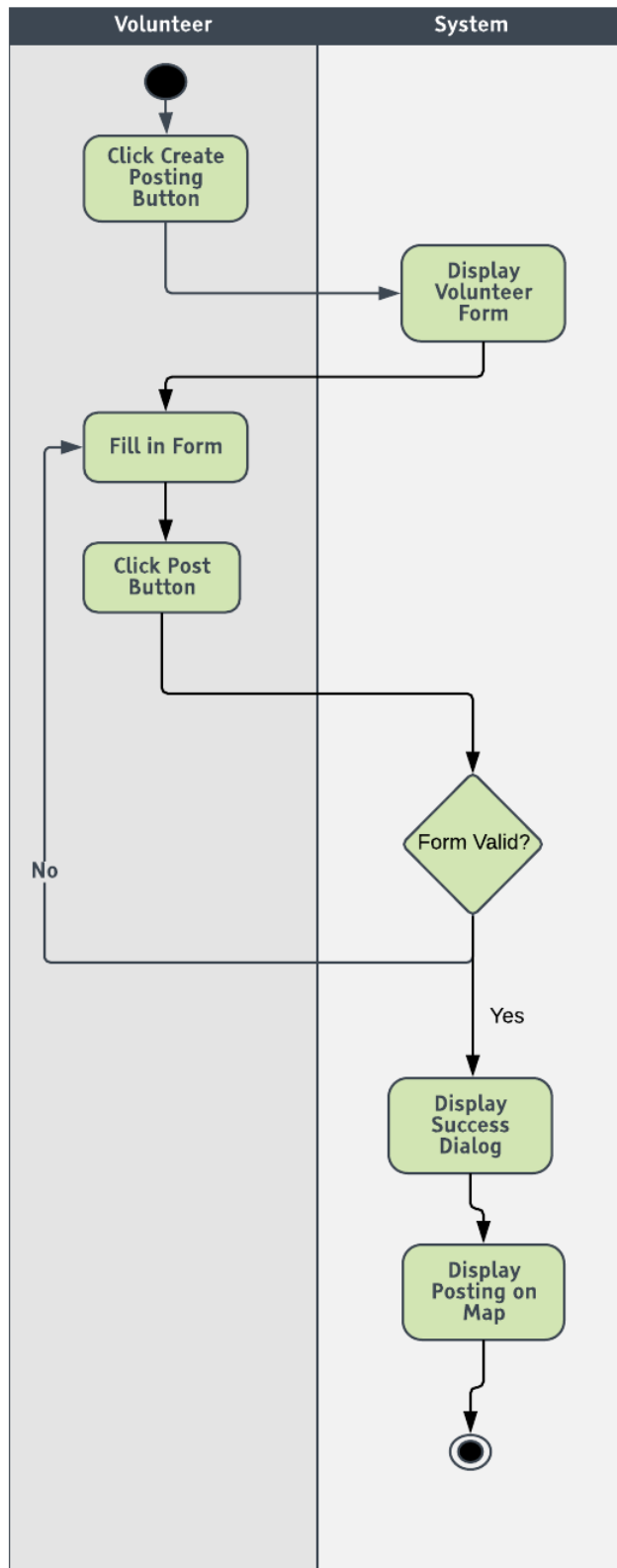
<b>Use Case ID:</b>	UC-02
<b>Use Case Name:</b>	Remove reported postings
<b>Description:</b>	Volunteers can report bad postings and admins have can remove postings that have been reported.

<b>Actors:</b>	Volunteer, Admin													
<b>Pre-conditions:</b>	User has an account and is logged in, a posting exists													
<b>Post-Conditions:</b>	A reported post is either removed or allowed													
<b>Frequency of Use:</b>	Daily by volunteers and admins.													
<b>Flow of Events:</b>	<table border="1"> <thead> <tr> <th></th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>A Volunteer clicks the Report Button on a post.</td><td>System flags post and adds to Admin report queue.</td></tr> <tr> <td>2</td><td>Admin clicks the View Reported Postings Button.</td><td>System shows reported Postings</td></tr> <tr> <td>3</td><td>Admin clicks the Remove Posting Button next to the desired post.</td><td>System removes post.</td></tr> </tbody> </table>			Actor Action	System Response	1	A Volunteer clicks the Report Button on a post.	System flags post and adds to Admin report queue.	2	Admin clicks the View Reported Postings Button.	System shows reported Postings	3	Admin clicks the Remove Posting Button next to the desired post.	System removes post.
	Actor Action	System Response												
1	A Volunteer clicks the Report Button on a post.	System flags post and adds to Admin report queue.												
2	Admin clicks the View Reported Postings Button.	System shows reported Postings												
3	Admin clicks the Remove Posting Button next to the desired post.	System removes post.												
<b>Variations:</b>	3. Admin clicks the Posting OK Button, System removes post from Admin report queue.													
<b>Exceptions:</b>														
<b>Developer Notes:</b>														

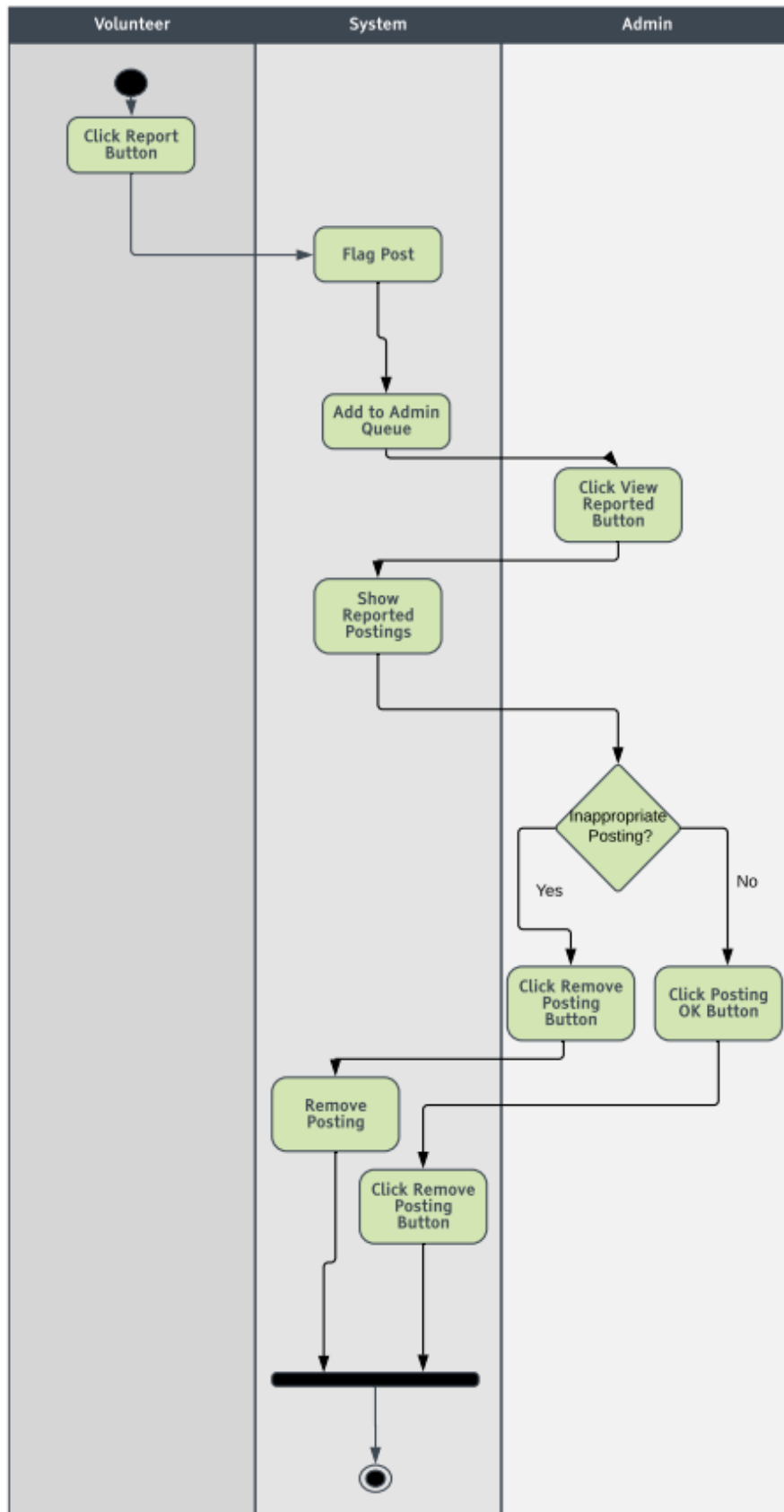
## Activity Diagrams

Requirement ID #1

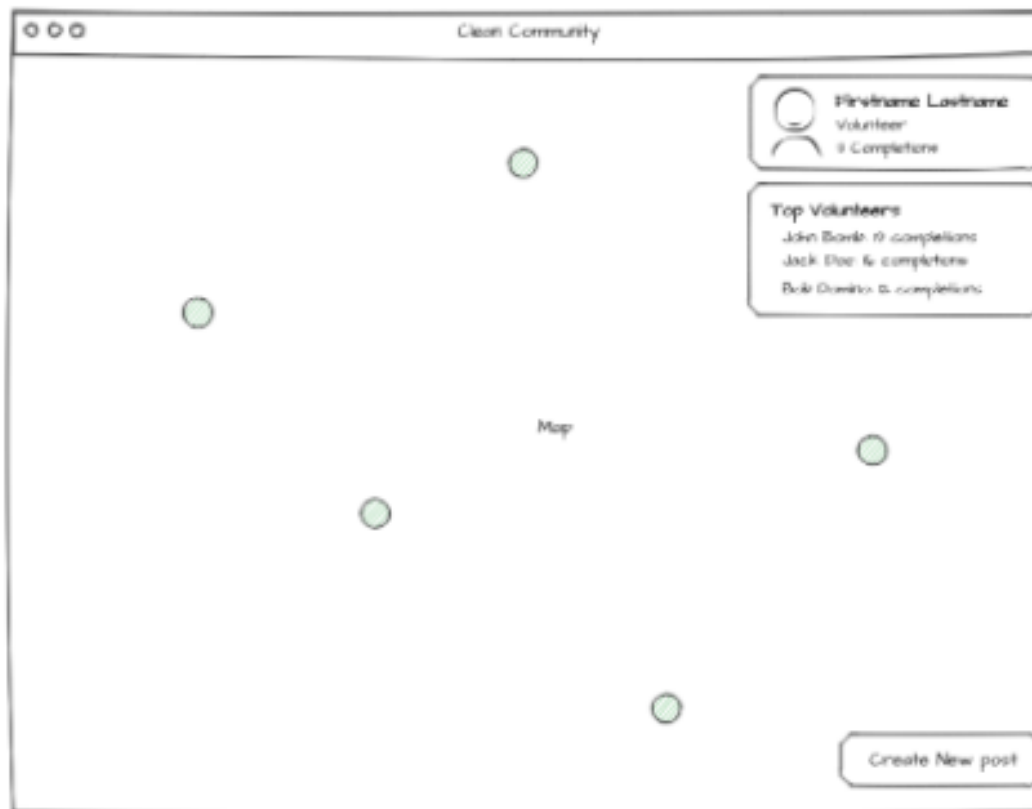
### UC-01 Create Postings



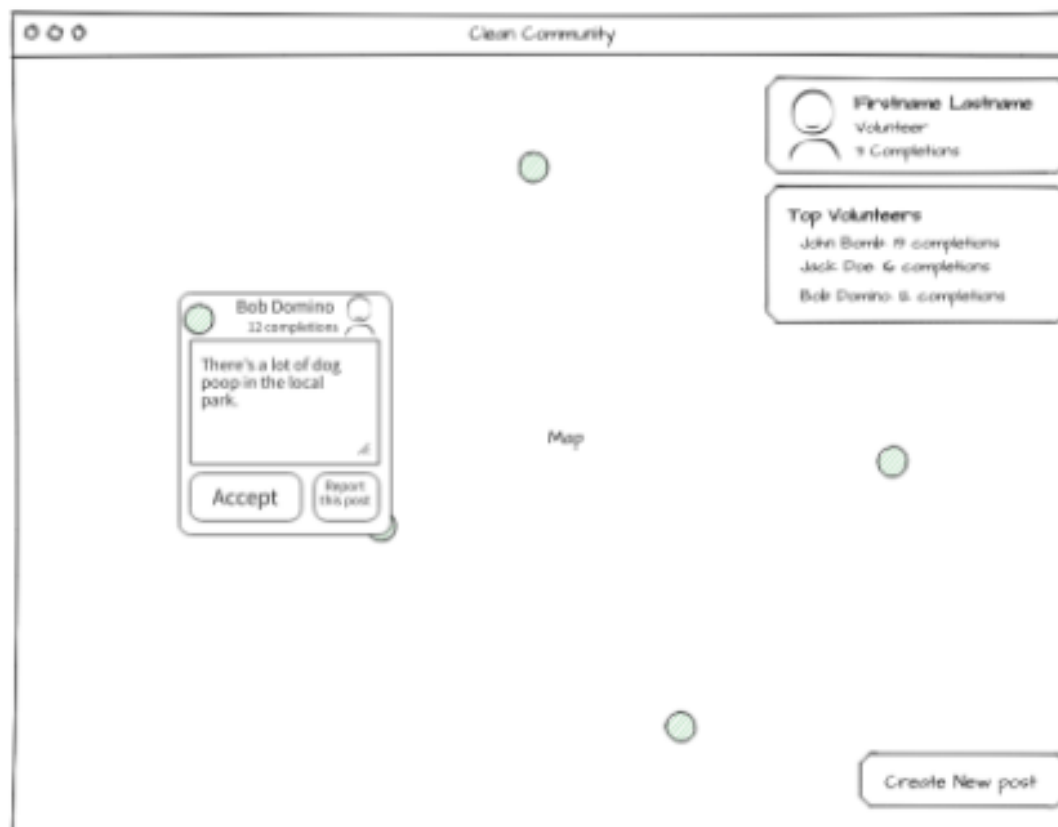
## UC-02 Remove Reported Postings



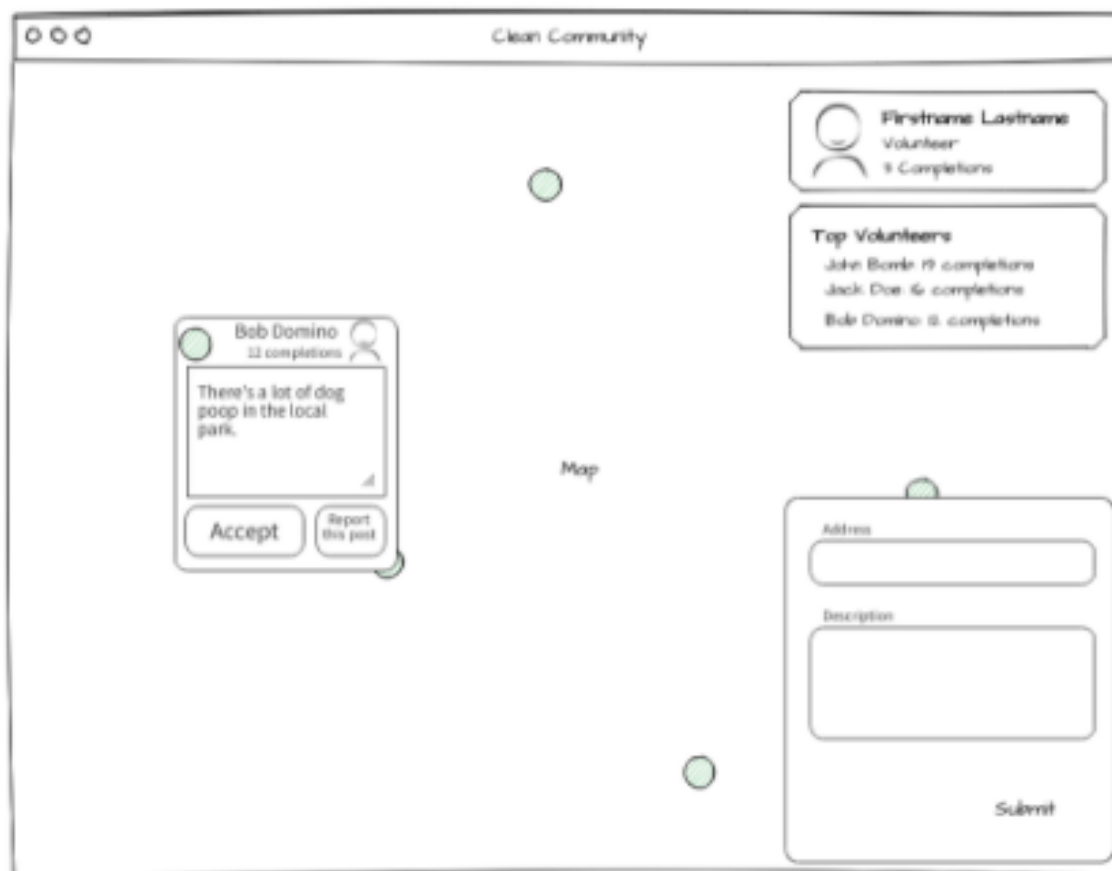
## UI Mockups



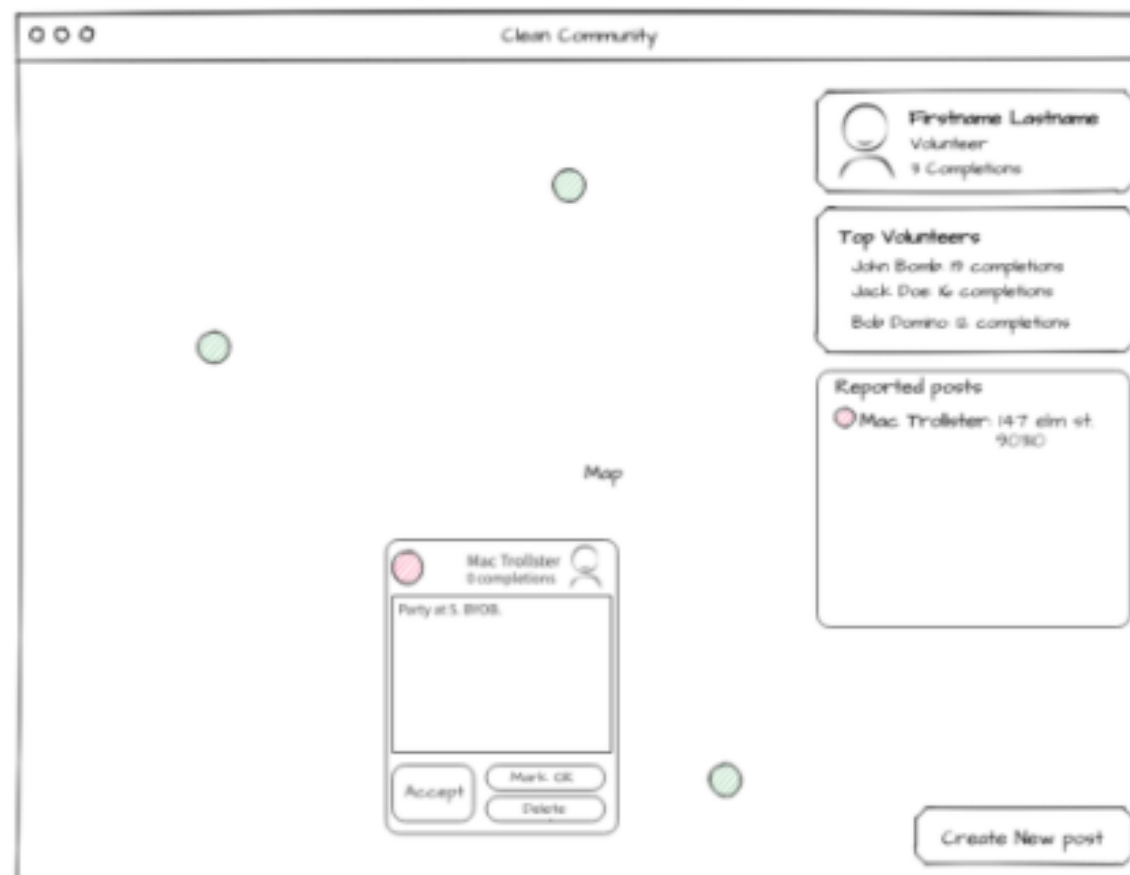
The main view for a volunteer. The green circles are clickable, depicting the location of posts in the system.



The view after clicking a post. The volunteer can accept the posting to complete it, or they can report it if there's something wrong with the post.



The view after the volunteer clicks the "Creating New Posting" button.

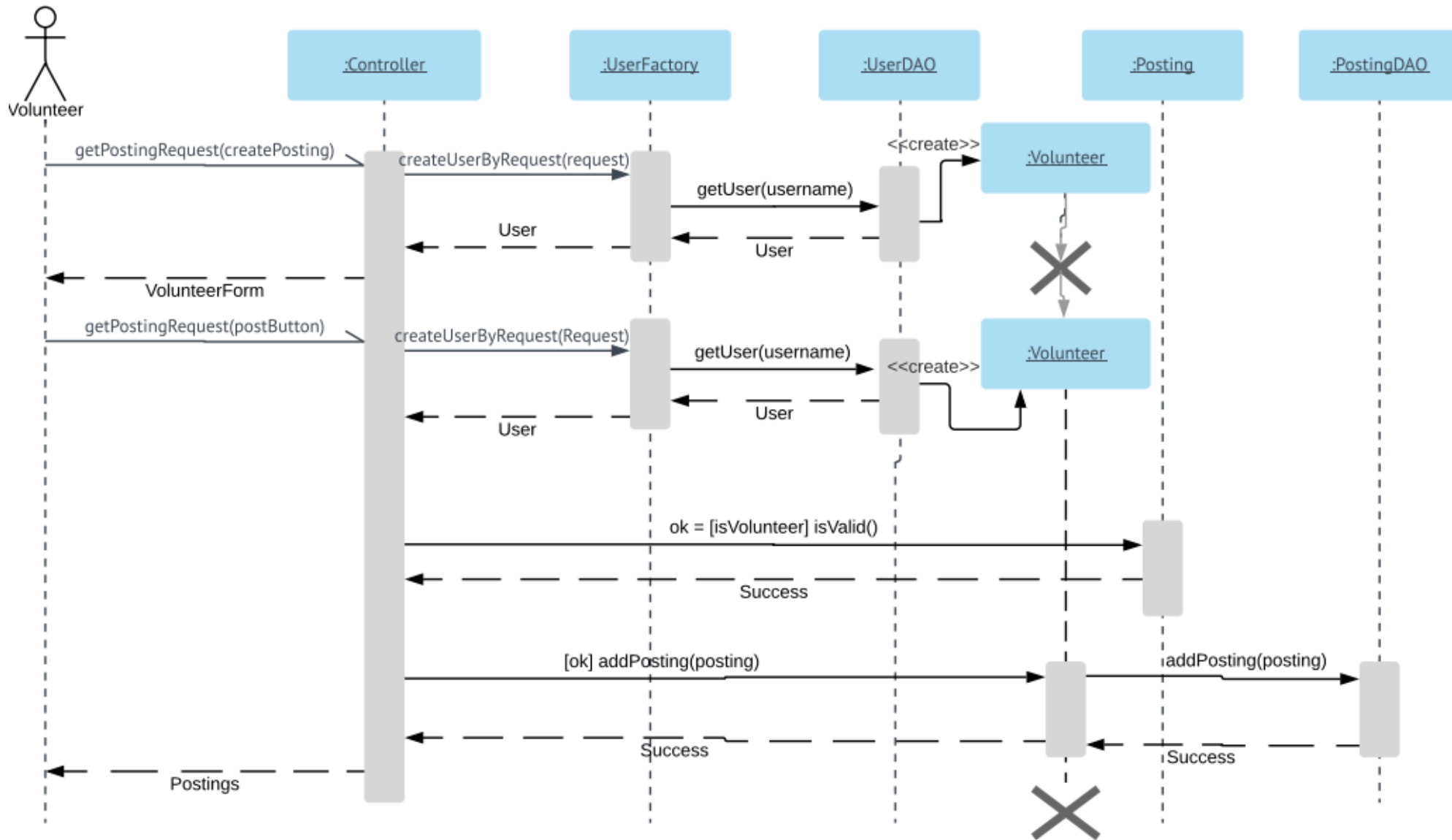


The main admin view. The posts that have been reported are showing a list format on the right while the corresponding post on the map is red.

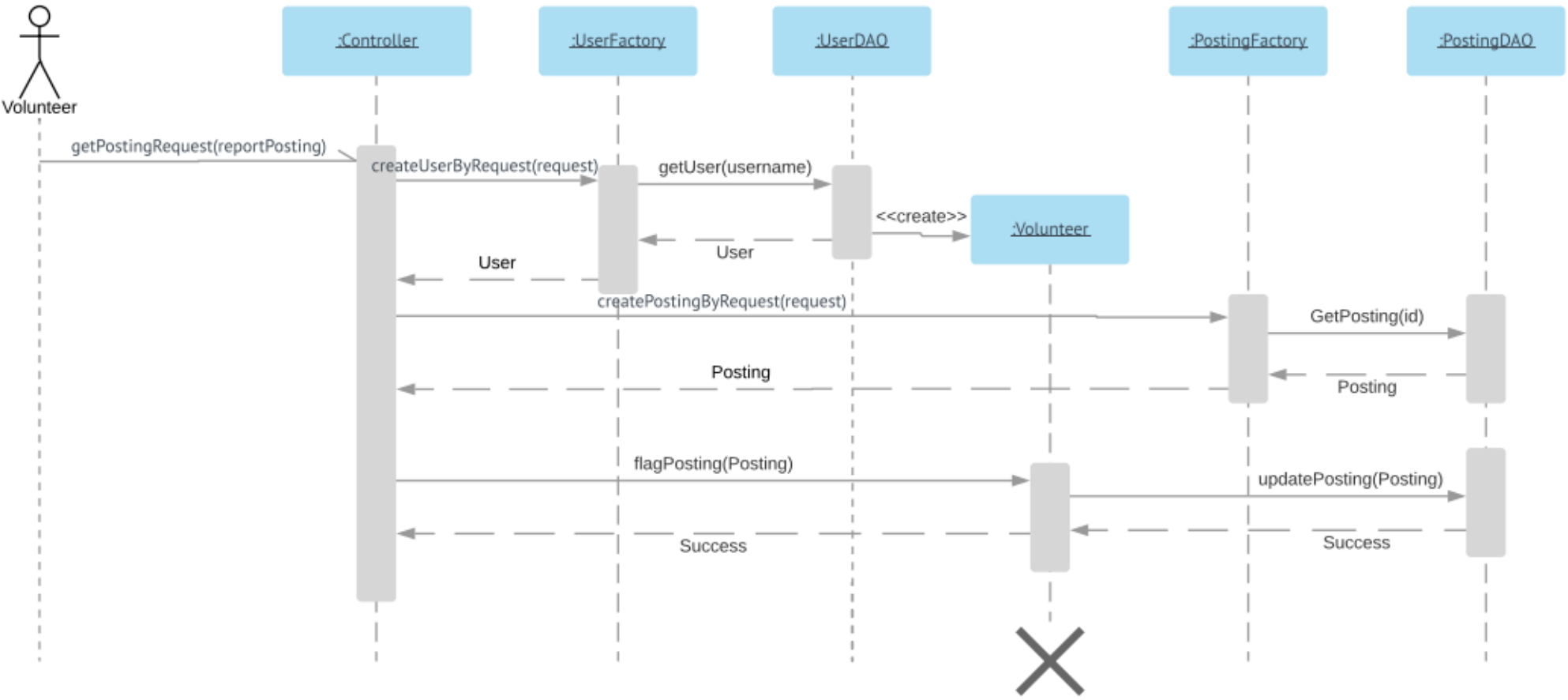


## Sequence Diagrams

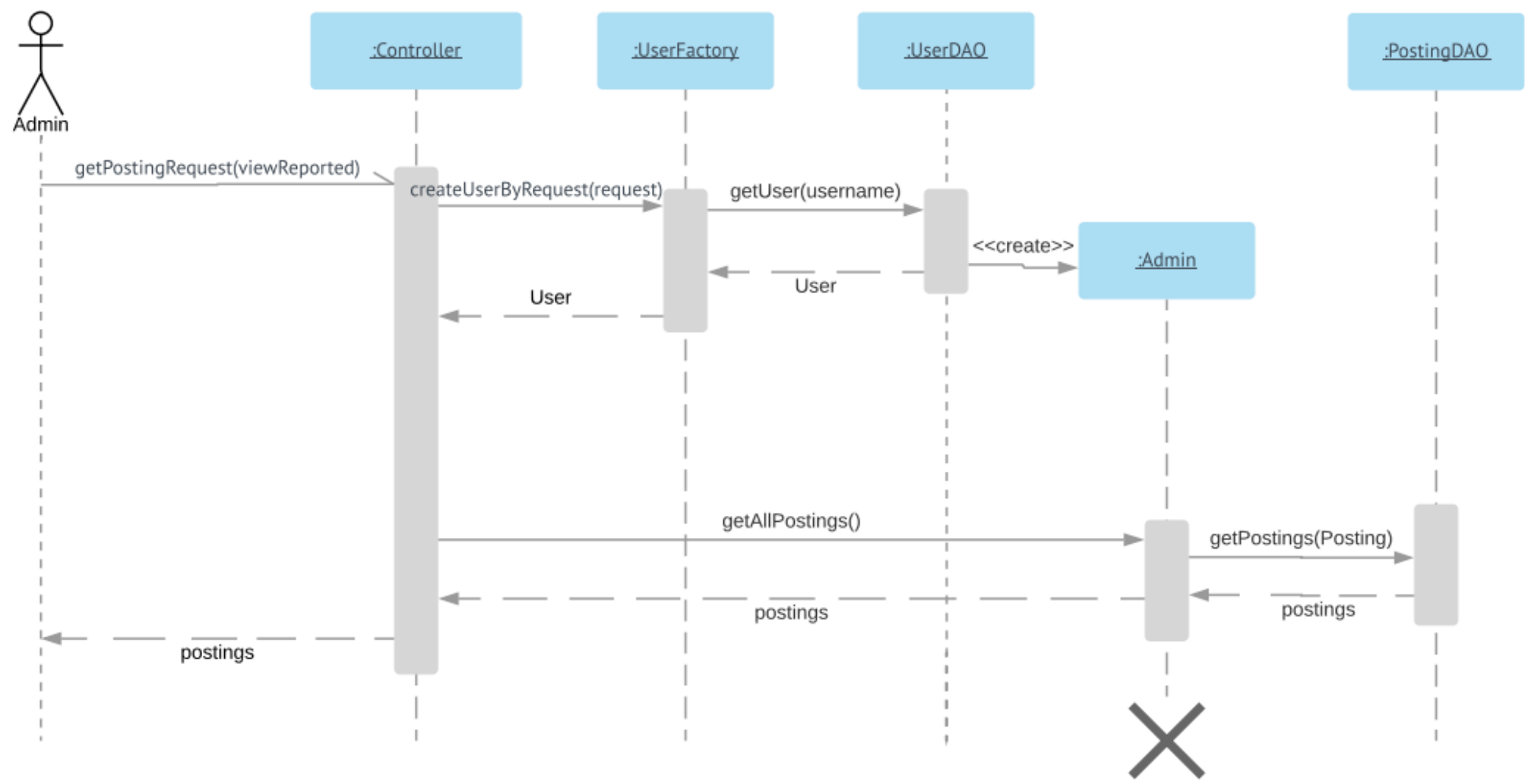
UC – 01



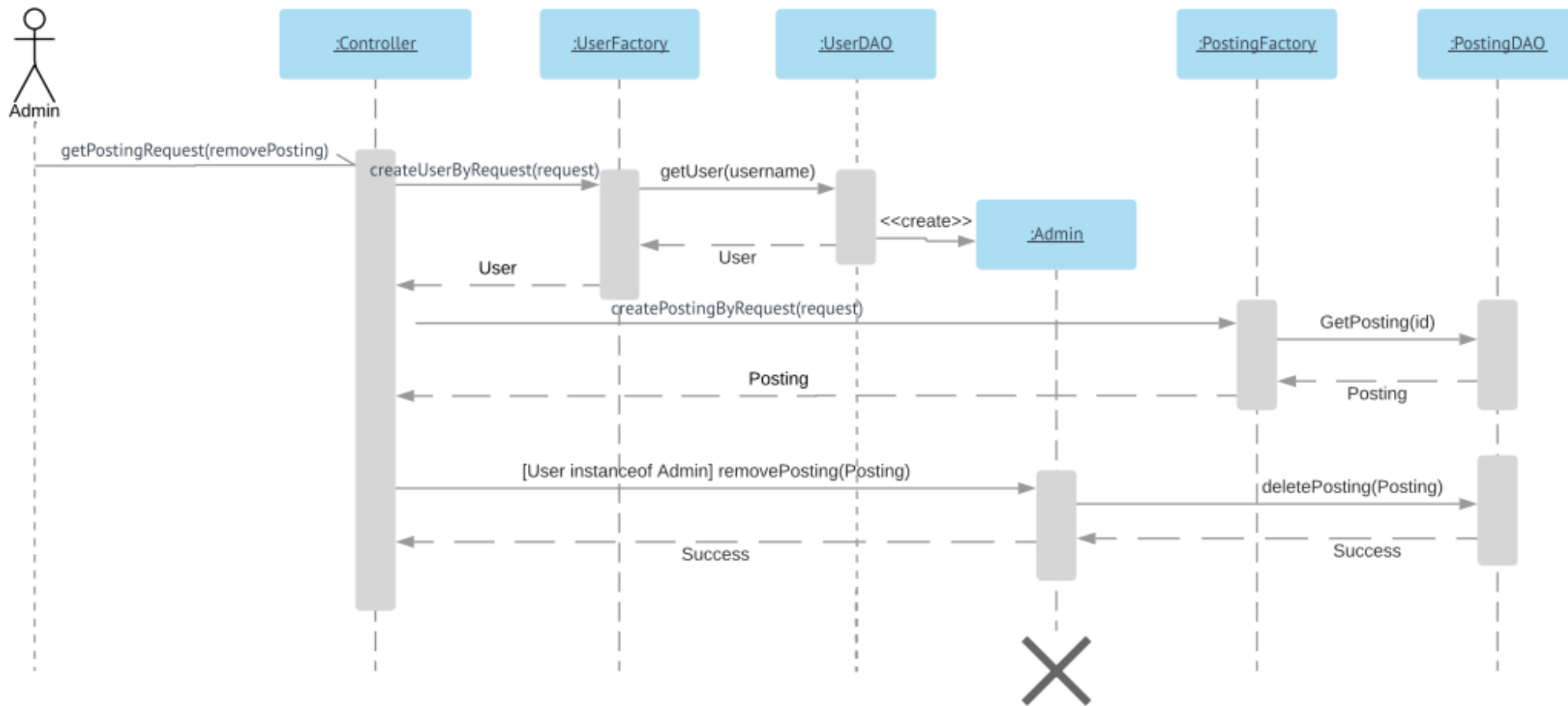
UC – 02A



UC – 02B



# UC – 02C



UC – 02D

