CSE361 Software Design

Team Maverick Applications Corporation:
Ryan Carlson, Stephen Pandorf, Jeremy Wagner, Kevin Rock

October 24, 2013

1.0 Introduction

This document serves as a guide at which developers and the client can better understand more implementation-oriented subjects in the project. It will outline qualities that we believe are important and how we will measure effectiveness in those areas. It will present a diagram showing the MVC (model view control) architectural system of the application. Next, it will show Use Case, ER, Class, and Interaction diagrams to show how objects are modeled in our system, how they interact, and how the user will interact with them at a high level. Finally, it will provide discussions about how each of the stated requirements are fulfilled by the design of the system.

2.0 Software Qualities

-Obviously Correctness, which can be checked by looking at the functional requirements.

-Portability and Interoperability since this is a web-based application. These could be measured by saying that the website will work of x% of operating systems and with y% of browsers.

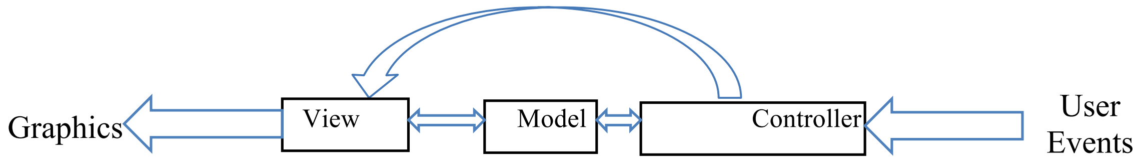
-Maintainability because hopefully this application will be around for a long time. This will be measured by saying any future improvement will only take x amount of time.

3.0 Architecture

Model = Database (Tables including Users, Posts, and Hashtags among others)

Controller = PHP objects/classes (Including Users, Pages, and Posts among others)

View = UI (includes pages and posts)

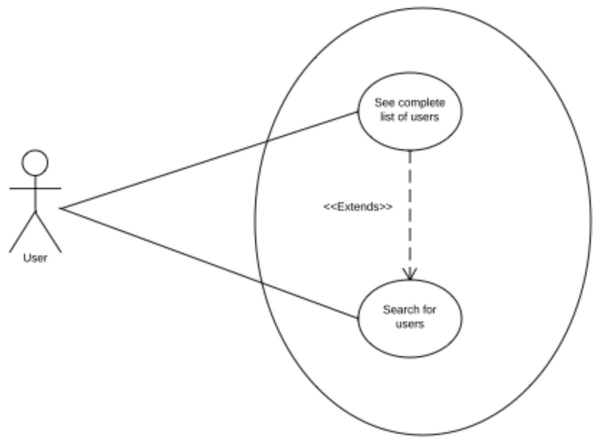


This architecture satisfies the functional requirements because it facilitates users logging in/out, making posts, and including hashtags in their posts.

4.0 Diagrams

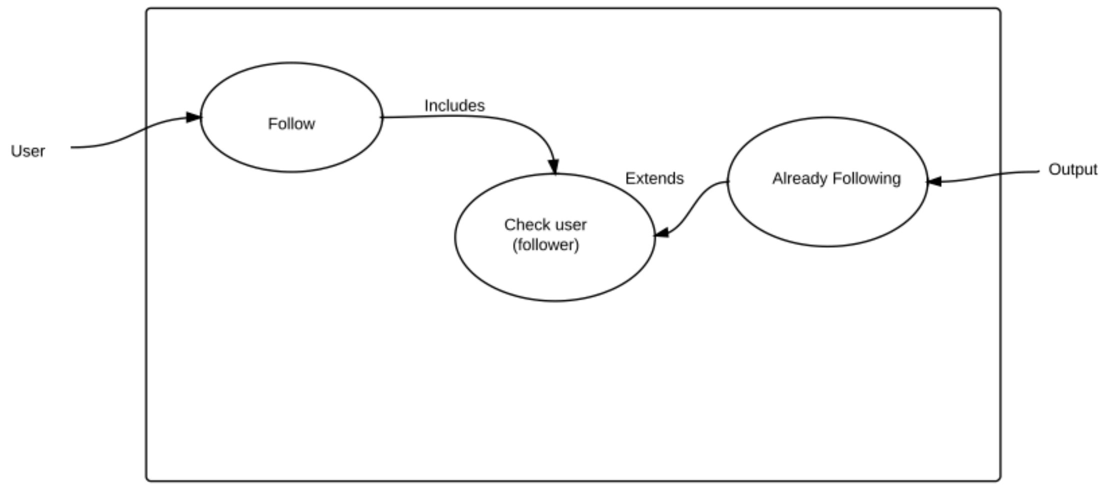
4.1 Use Cases

4.1.1 Follow Management



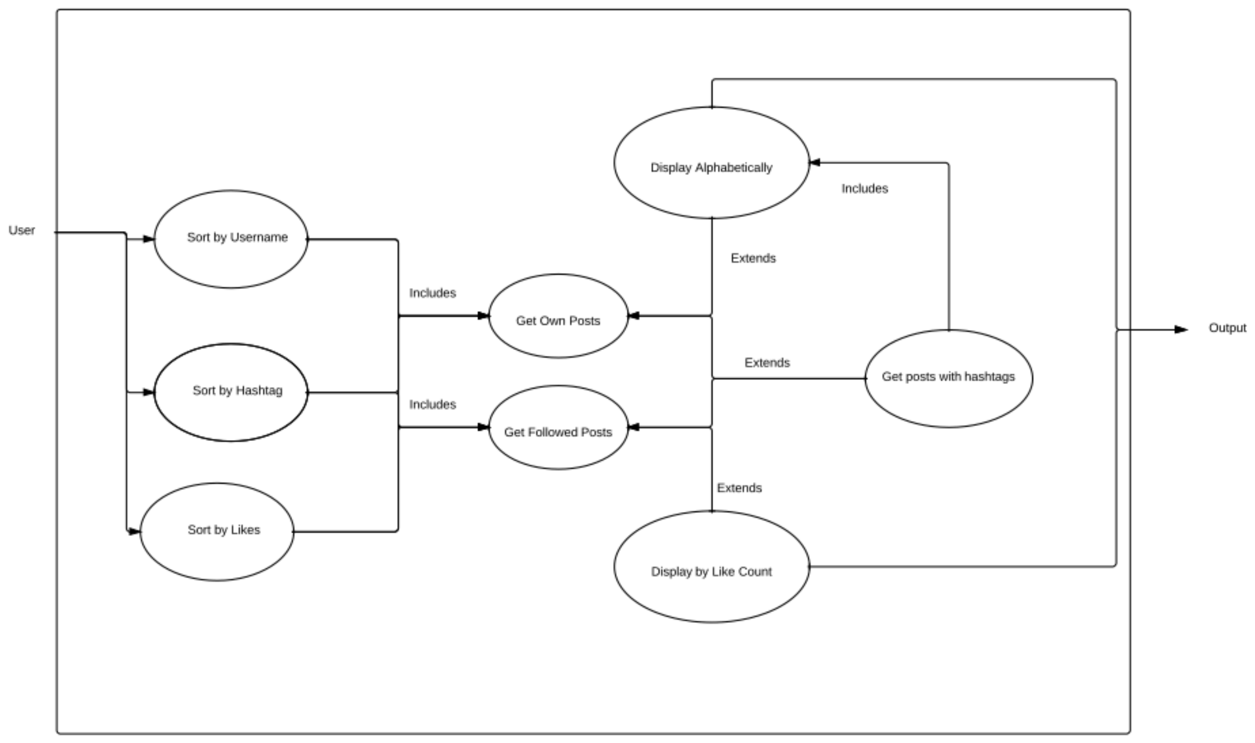
The follow management use case shows that a user can see a complete list of users and also search for specific users.

4.1.2 Following a User that's Already Followed



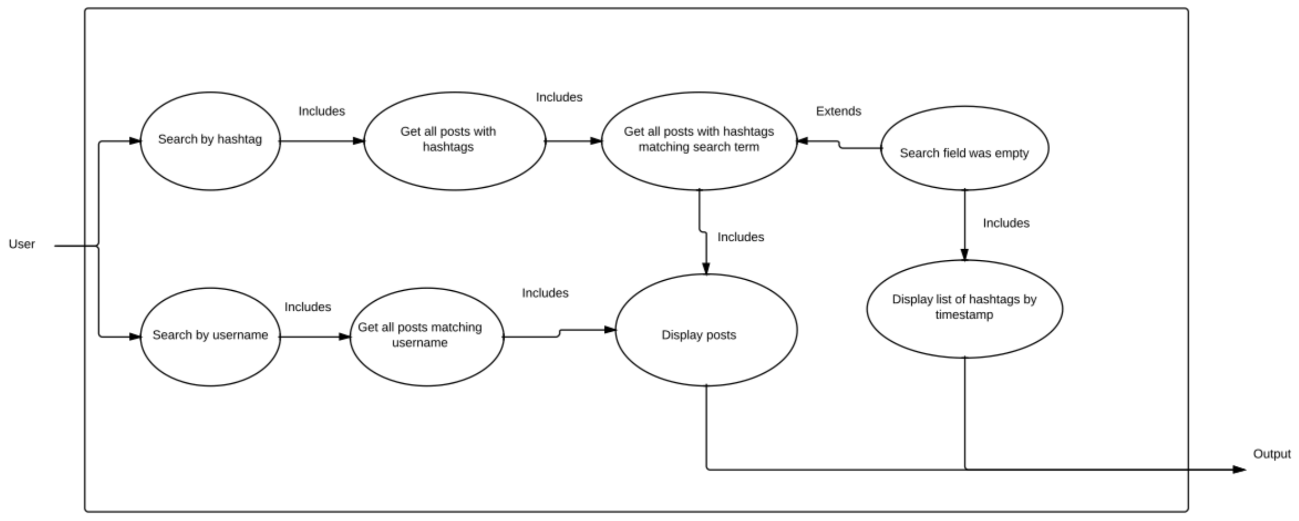
This use case handles R10; it illustrates how the system will check if the user is already following another user when trying to follow them.

4.1.3 Sorting



This use case handles R13, and more specifically R13.2. It illustrates how the contents of the main page can be sorted by username, hashtag, or by likes. It considers the situation where a post may not have a hashtag as well.

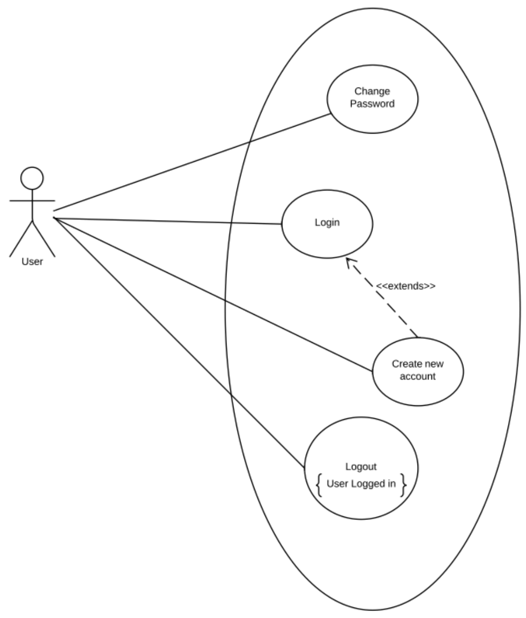
4.1.4 Searching



User
Search by hashtag
Includes
Get all posts with hashtags
Includes
Get all posts with hashtags matching search term
Extends
Search field was empty
Includes
Includes
Display list of hashtags by timestamp
Search by username
Includes
Get all posts matching username
Includes
Display posts
Output

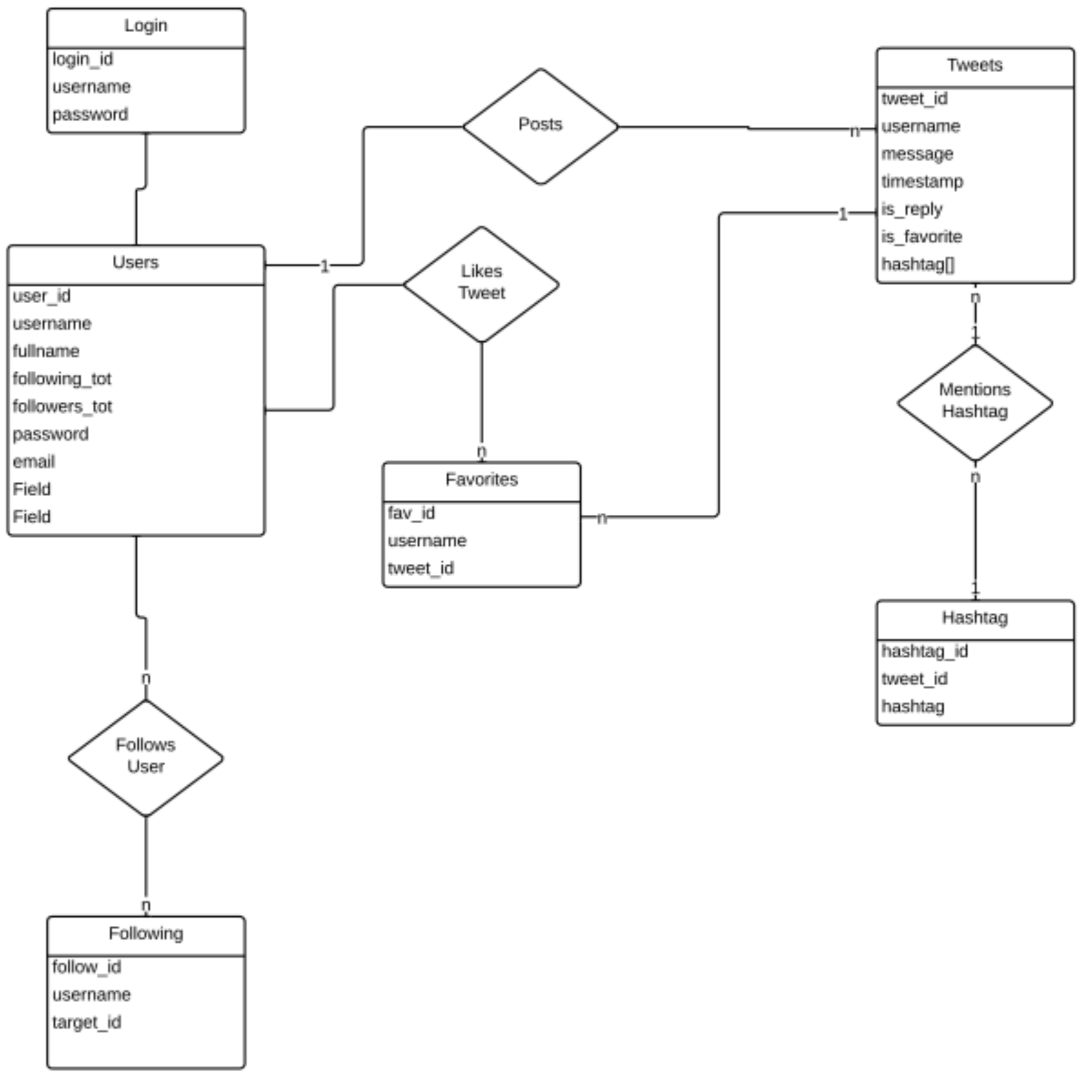
This use case handles R17. It illustrates how searching is accomplished by getting the posts, then ones matching the field, then displaying the relevant information. It considers the case where the search field is empty, where the system will display a list of all hashtags ordered by their time of creation.

4.1.5 User Login



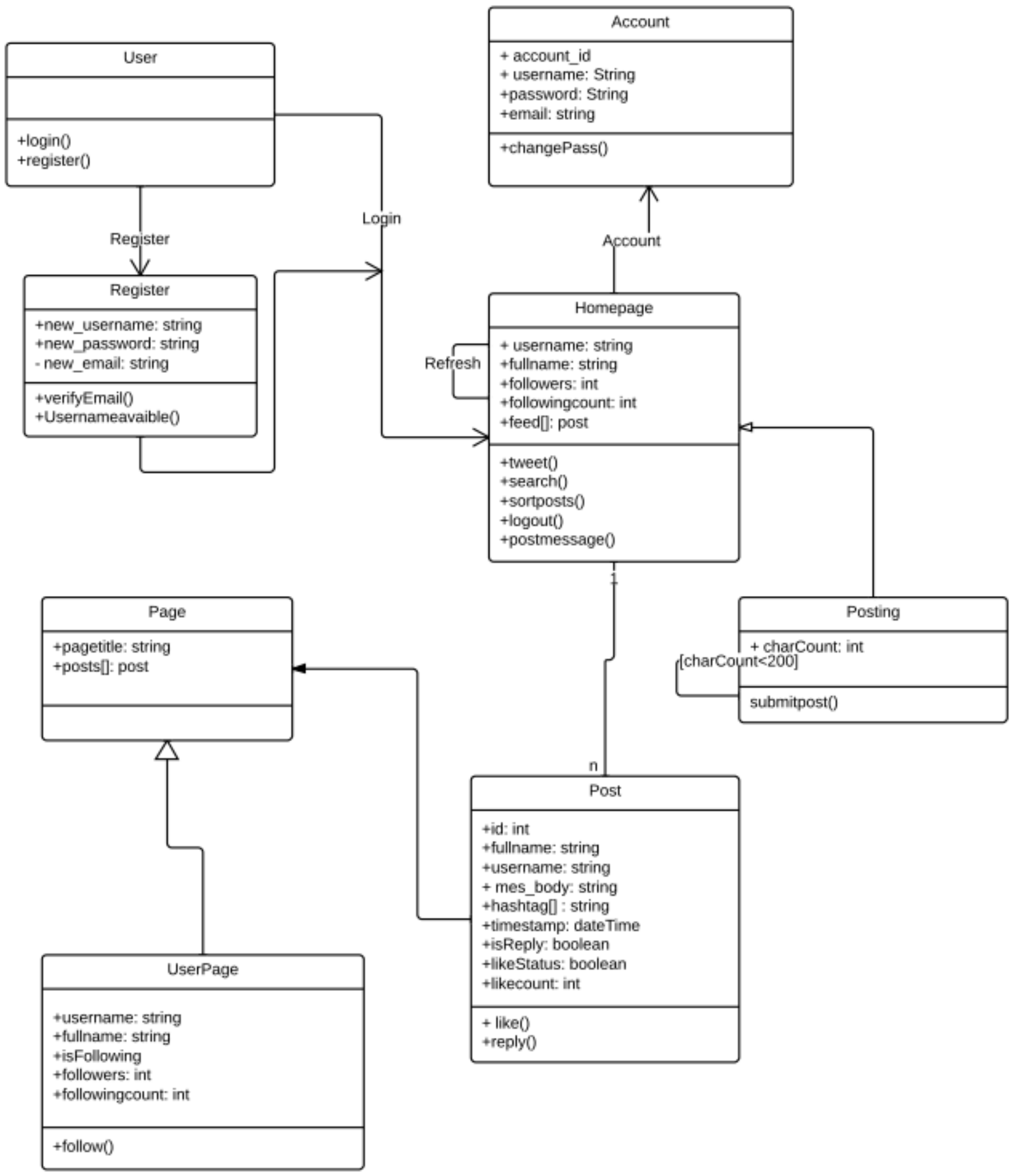
The User Login use case shows various ways the user interacts with the site framework by logging in, logging out, or changing their password.

4.2 ER Diagram



The ER diagram for our application consists of 5 tables within our database. The login table contains user passwords for logging in to the website. This connects with the user table which contains information for profile information. The user table connects with flollowing, tweets, and favorites. Following contains user associations. Favorites conatins information of saved tweets. Tweets may contain one or more hashtags which is merged with the hashtag table.

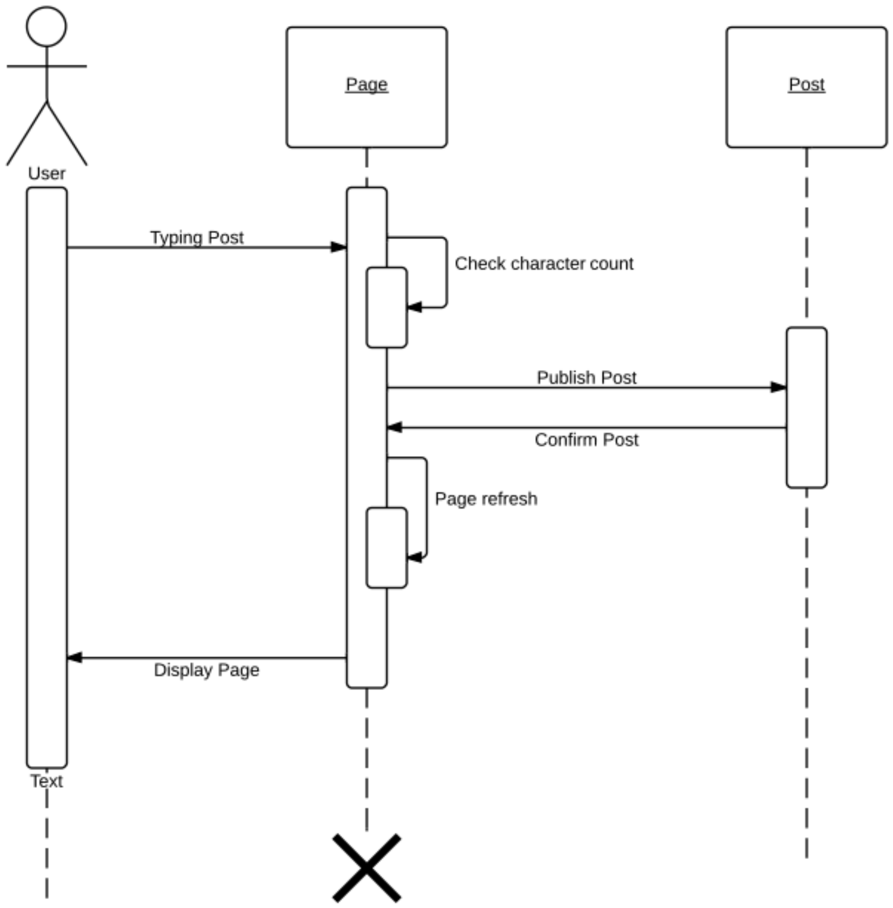
4.3 Class Diagram



The class diagram breaks the application into parts and describes their associated functions. Users start on a login page and are directed either to register or to the homepage. From the homepage they can post, view account settings, search, and view who they are following and who is following them.

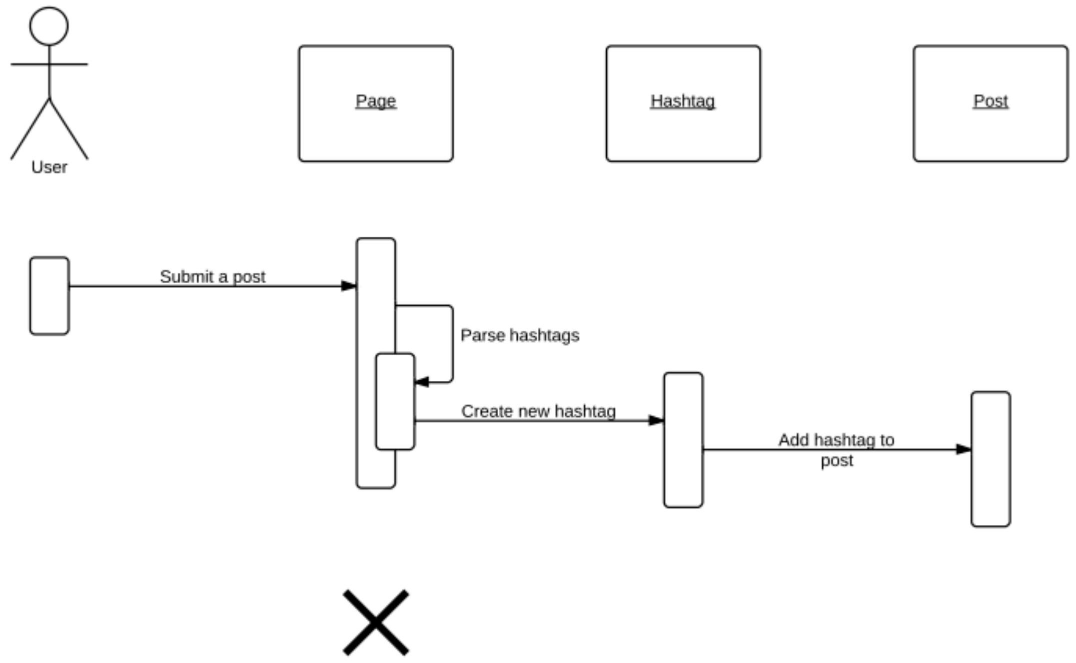
4.4 Interaction Diagrams

4.4.1 Making a Post



The making a post sequence diagram shows how a user can actually make a post.

4.4.2 Hashtagging a Post



This sequence diagram shows the interactions of the system while hashtagging a post. The user submits a post with hastags, the page parses the hashtags out and sends them to the hastag class to create a hashtag, which is then added to the post.