Project1c

Casey McGuire and Joshua Nance

March 13, 2015

1 Introduction

2 Use Cases

1. **Case 1:** As a user I want to be able to log in to the website using my unique credentials so that I can have persistent data associated with my profile.
   Description: To differentiate a user from someone who simply comes across a blog post, reads it once, and then never visits the blogging platform again, we would like to implement a feature that allows those visitors who want to become consistent members of the community to have their own profiles. Once they have a profile, they would be able to post their own blogs, and network with the other authors of the platform.
   Actors: User
   Preconditions: The visitor has decided they wish to be made an official member of the community, and they have a valid email address and a password they would like to use to log in to their new account.
   Flow of events:

   (a) The user navigates to the home page of the blogging platform.
   (b) The user clicks on an option to register an account with the platform.
   (c) The user will be taken to a new page, on which he or she can input whichever credentials they would like to use to log in.
   (d) They will be redirected to a page that confirms they now have an account.

   Alternate Flow of Events:

   (a) The user already has a profile, and so may use the ”Log-In” option instead of the ”Register” option.
   (b) The user may navigate to the ”Register” page, where he or she will be prompted to input his or her new credentials. If, however, the potential user decides to not finish making the new account, there will be an option to take that individual back to the home page.

   Post-conditions: The individual now has a new account, and can log in to the platform using the credentials he or she provided during registration.

2. **Case 2:** As a logged-in user I want to be able to see my own blog posts, and have the ability to create, edit, and/or delete them, so that I can control my own content.
   Description: One of the purposes for registering an account with the platform is to have the ability to manage any content you generate using the platform. This content might include posts or comments you have written. Your posts should be easily accessible to you through your own personalized dashboard, for you to have the ability to edit or delete them from your personal blog.
   Actors: User
   Preconditions: The user has successfully logged in to his or her account. In order to edit a blog post, there of course must be some blog post that has already been created, which can then be edited.
   Flow of Events:

(a) The user logs into his or her account.

(b) The user will be directed automatically to his or her dashboard.

(c) The user will see links associated with the creation, editing, or deletion of blog posts.

(d) The user will click on the link that suits their intended purpose. If a user wishes to create a post, the user will click the corresponding link and be taken to a text editor; if the user's intention is to edit an existing post, their will be an "Edit" link next to the title of all previously posted articles; finally, if the user wishes to delete an existing post, there will be a "Delete" link next to every "Edit" link.

Alternate Flow of Events:

(a) The user is logged in and can see his or her dashboard, but does not choose to create or edit any posts.

Post-conditions: The user has successfully created a new post, edited an existing one, or deleted an existing one, or any combination thereof.

3. **Case 3:** As the creator and owner of my own blog posts I want to be able to view and filter comments from other users, so that I can police my blog.
Description: Almost any blogger or content-hosting individual that allows users to comment on the content created by him or her will invariably receive some spam comments or just hateful and/or inappropriate remarks from certain other users. If a comment is posted that does not positively or constructively contribute to the conversation, the author of the post that is being commented on should have the ability to remove such unnecessary comments from their comments section.
Actors: User (multiple)
Preconditions: The user must have created a blog post on which other users have provided comments.
Flow of Events:

(a) The content creator has posted an article.

(b) Other users have commented on the article.

(c) The content creator determines that one of the comments is inappropriate and should be removed.

(d) The content creator removes that comment from the article.

Alternate Flow of Events:

(a) The content creator has posted an article.

(b) Other users have commented on the article.

(c) The content creator determines that all comments are pertinent or constructive in some way, and will not be removing any of them.

Post-conditions: The content creator has successfully removed the inappropriate comments.

4. **Case 4:** As a user I want to be able to comment on other blog posts from other users, so that I can actively participate as a member of this blogging community.
Description: Another way to contribute to the community besides writing your own blog posts is to comment on other users' blog posts. Every user–i.e. everyone who has an account and can sign into the platform–should be able to visit the blogs of other users and comment on any article they have written.
Actors: User
Preconditions: The user must be logged into his or her account. The user must have navigated to an article that another user posted.
Flow of Events:

(a) The user navigates to an article written by another user.

(b) The user scrolls down to the bottom of the article.

(c) The user clicks on "Add a Comment."

(d) The user types his or her comment and submits it.

Alternate Flow of Events:

(a) The user navigates to an article written by another user.

(b) The user may read the article.

(c) The user decides not to comment on the article.

Post-conditions: The user's comment is submitted and stored in the database. The next time that user or another visitor reads that article, the user's comment will be included in the comment section.

5. **Case 5:** As a user I would like to be able to save/bookmark my favorite blog posts from other authors, so that I can come back to them later.
Description: In visiting other users' blogs and reading many different articles hosted on the platform, the user might come across articles that the user would like to come back to at a later date, maybe to re-read or to show other people. It would be very convenient if that user had the ability to "save" articles for later viewing. This feature would allow the user to do just that, and to have a spot on his or her dashboard dedicated to "saved" articles.
Actors: User
Preconditions: The user must be logged into his or her account.
Flow of Events:

(a) The user navigates to an article written by another user.

(b) The user reads the article.

(c) The user reaches the bottom of the article and clicks the "Save this Article" link.

(d) The article can now be found under that user's "saved" section.

Alternate Flow of Events:

(a) The user navigates to an article written by another user.

(b) The user reads the article.

(c) The user reads it once and chooses not to save it.

Post-conditions: The user's saved section will now include the new article.