Homework 1: User Interface Design

Due date: Thursday, 08 September 2016, 11:59 p.m.

This homework introduces you to some basic elements of user interface design, and is the first in a series of homeworks in which you will build an increasingly sophisticated nano-blogging site, **grumblr**. **grumblr** will eventually be a featureful, interactive web application including user registration and authentication, email integration for user verification, photo upload, and quasi-real-time updates. However, for this homework your task is just to design and implement user interfaces that you will reuse in your development of web applications in subsequent assignments.

The goals of this assignment are for you to:

- Become familiar with the course infrastructure needed to access and turn in course work.
- Become familiar with basic HTML and CSS.
- Gain experience with a modern version control system (Git) and practice documented incremental development with Git commit (and commit messages).
- Learn a formal user interface design process.
- Optionally, gain familiarity with Twitter Bootstrap.

Part 1: Designing a user interface

For this part of the assignment you will formally design your **grumblr** site, documenting your design process by turning in wireframe diagrams and feedback for iterations of your design.

The UI design process

To design your site, you must embark on a process of iteration, feedback, and revision. You should start by interpreting our **grumblr** specification (below) and outlining which site features will appear on each **grumblr** page. You should then draft your design by sketching wireframe diagrams for the site and obtain feedback from TAs or your peers. You must record who provided feedback and briefly describe their feedback; include this information in a file named feedback.txt or feedback.md. After receiving feedback, revise your wireframe diagrams and include both your original and revised drafts in your homework submission.

Your wireframe diagrams may be hand-drawn (turn in a scan or photo of your diagrams) or computer generated (using a sketching or wireframing tool).

The grumblr specification

This section describes the main features for **grumblr**, a nano-blogging site that you will implement over several assignments.

- Non-logged-in, non-registered users may register for the site. New users must provide a username, first name, last name, and password. Registering for the site leaves the user logged in as the newly-registered user.
- Registered users may log in using their username and password.
- Logged-in users are able to post a short (42 characters or less) message. Posts, when displayed, show the following information:
    - the contents of the post,
    - the user who posted it (linking to the user's profile), and

    - the date and time the post was written.
- Logged-in users may view a 'global stream', displaying all posts that have been posted in reverse-chronological order (most recent first).
- Logged-in users may view profiles of other users (or their own profiles) when clicking on links provided with posted messages in the global stream. Profile pages contain information about a user (e.g. first name and last name) as well as all of the posts that user has made, in reverse-chronological order.
- Logged-in users are able to log out.

Requirements for your grumblr design

You must create wireframe diagrams for at least four pages:

1. A **login** page; displays a form to the user requesting username and password, linking to the registration page if the user wants to register instead.
2. A **registration** page; displays a form requesting a username, first name, last name, and password (with a password confirmation field).
3. A **global stream** page; lists posts from all users.
4. A **profile** page for a user; shows information about a user as well as all posts from that user.

To avoid over-specifying the requirements of this assignment, we will not describe strict requirements for the design of these pages. We expect that you have personal experience with social networking, blogging, and/or microblogging sites and that you can make reasonable UI design choices based on your personal experience.

Part 2: Creating static pages for **grumblr**

After designing your user interface, receiving feedback, and revising your design, you should create static HTML pages modeled on your wireframes. When you are done, you should have a simple, static user interface for your site, but the site's features will not work because you have not yet written a web application to run the social network.

Requirements for your grumblr implementation

- Implement static HTML pages based on your wireframes.
- All pages on your site should share a common look and feel.
- Even though non-static functionality will not work because you do not write a web application for this assignment, you should include reasonable static sample data for features your web application will generate later. For example, your home stream interface should include some sample posts and some associated images. Similarly, your view-profile page should show sample profile data. You can use any appropriate text or images in your sample interface.
- You should correctly implement static-seeming features to allow us to browse your site. For example, if your home stream interface includes a link to see the profile for the logged-in user, that link should correctly link to the sample profile page. All static sample data (such as a link to a static image for a post) should be correct in its static form.
- You must not use a UI generator tool. You may base your solution on templates and examples from http://getbootstrap.com/getting-started/#examples, but you may not use examples or templates from other sources. Cite any resources (such as Bootstrap templates) you use.

Recommended approach

We recommend the following approach to this assignment:

1. Clone a copy of your GitHub repository to your local computer.
2. Read and/or skim the Bootstrap "Getting started," "CSS," and "Components" pages on http://getbootstrap.com/. We do not require that you use Bootstrap for this homework, but it cleanly provides many UI features you might want for **grumblr** and you should be aware of what Bootstrap Components are available before you start designing your solution.
3. Create an outline of **grumblr** features and draw rough wireframe diagrams for its user interface. Include enough detail for you to understand which features (including both core features and navigational aids) are available on each page. Commit these wireframe diagrams to your homework/1 directory.
4. Solicit feedback from your peers or a TA about your wireframe diagrams and record notes about their feedback in feedback.txt or feedback.md. Commit this feedback.
5. Revise your wireframe diagrams according to the feedback you received. Commit the revised wireframe diagrams.
6. If you choose to use Bootstrap for **grumblr**, get the compiled and minified version of Bootstrap 3.3 (any sub-version) from http://getbootstrap.com/getting-started/. Unzip the files into your repository and select templates from http://getbootstrap.com/getting-started/#examples on which to base your solution. If you choose not to use Bootstrap we recommend that you start from a simple HTML5 page, such as the simple example provided in lecture 02c: HTML.
7. Customize the templates or base HTML5 to include **grumblr**'s features, with all your pages sharing a common look and feel. To achieve a common look and feel for this assignment you will probably need to duplicate some features (such as a navigation bar) for each of your pages.

You can complete the **grumblr** component of this assignment successfully by writing just HTML. We think it's easiest to design a nice looking UI by just writing HTML and using some of the Bootstrap CSS and Components, based on the Bootstrap examples. To do this you would not need to know any JavaScript or LESS or any of the details of how Bootstrap works; you would just need to learn what Bootstrap provides as a UI library and how to use some of its components. You would not even need to know how to write CSS rules yourself; you could just use some of the CSS rules that Bootstrap provides.

Turning in your work

Your submission should be turned in via Git and should consist of all files for your **grumblr** interface and design process. You might want to organize your files logically, such as the following:

```
[YOUR-ANDREW-ID]/homework/1/
   wireframes/
   src/
     [your static html files]
     css/
     images/
   README.md
```

When you are done, you should have turned in the following:

- **grumblr** wireframes, original and revised, in any reasonable file format (.pdf, .png, .jpg, etc.).
- A file (feedback.txt or feedback.md) containing your UI feedback and who provided it.
- Your static web pages (and any supporting files) for the **grumblr** user interface.

You can check that you have turned in everything by either (1) cloning a new copy of your GitHub repository and checking that it contains all of the files you want to turn in, or (2) viewing your repository online at https://github.com/CMU-Web-Application-Development/**YOUR-ANDREW-ID** and checking the files there.

Basic instructions for git were provided in an online course video, and more extensive documentation is available at http://git-scm.com/book.

Committing your work

Your use of Git and your full commit history is an evaluated aspect of all your work in this course. As you work you should demonstrate good use of Git, our version control system. By "good" we mean that you should commit small, incremental changes and use meaningful commit messages.

Incremental changes mean that each commit should focus on a single issue, feature, or addition to your solution. For example, if you implement five new features then it is poor to commit all your changes in a single large commit. Instead you should commit five times, with each each commit containing all the changes for a single feature.

Commit messages should consist of a brief one-line summary and an optional, more detailed explanation separated by an empty line. For example, one commit message might be:

```
Add persistent navigation bar to grumblr UI

As per Marvin's paranoid feedback regarding easy navigation, the
sidebar links have been moved to a persistent top bar.
```

The first line of the commit message is similar to an email subject, and the optional explanation is like the body of an email message. For more guidance on how to write good commit messages, see http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html).

Grading your work

For substantial credit *your solution must clearly demonstrate the learning goals for this assignment*. Points in this assignment will be allocated approximately as follows:

- Become familiar with the course infrastructure needed to access and turn in course work: 5 points
- Become familiar with basic HTML and CSS: 40 points
- Gain experience with a modern version control system (Git) and practice documented incremental development with Git commit (and commit messages): 5 points
- Learn a formal user interface design process: 20 points
- Optionally, gain familiarity with Twitter Bootstrap: 0 points