COMPSCI 345

Human-Computer Interaction

Assignment 3: Realizing a Design

- Worth 12% of your final grade
- This assignment is due Monday 21 May 04:00 pm
- This assignment must be done **individually**

Aims

The aim of this project is to give you experience in designing and implementing a prototype graphical user interface.

Background and Scope

This assignment is based on the same problem as Assignment 2. Now your task is to further develop the design of the interface as an HTML-based prototype.

1. Note that you are *not* bound to your Assignment 2 work as a specification; that is:
    a. You do *not* have to match the lo-fi designs your group did, but you may if you wish
    b. You do *not* have to implement any or all or the features in your Assignment 2 design, but you may implement some of them if you wish
2. Exclude from your prototype any steps involving system access and authentication. Assume the user has their browser open to the first screen of the problem-based learning environment that they would see after selecting this specific assignment.
3. Do not include a splash screen.
4. Out-of-scope links (if any) on your prototype should not cause errors if clicked, but do not need to be appropriately tool-tipped.

Details

Create and submit deliverables for the following tasks.

1: Design Documentation

(a) Context: Set the scene for your prototype in terms of:

1. What is the domain/sub-domain?
2. Who are the users? Not a full persona background, just an outline of where they live and what their study situation is – should answer questions like whether peer group members are in the same school, or maybe just same region or country, or spread internationally. Name the 'viewpoint user' (the one we're logged in as when the prototype launches) and any other users we need to know about (i.e. ones that have posted visible feedback or who's solutions are viewed and commented on by the viewpoint user).
3. What is the 'moment' represented by the prototype (where are the viewpoint user and their peers up to on the assignment)?

Tabular or bullet point format recommended for this.

(b) Walkthrough: Using three to five illustrative screen shots and surrounding narrative text (150-600 words [more isn't always better!]), step the reader through how the prototype works showing key features that you implemented. Finish with a statement of what is 'out of bounds' for the implementation; i.e. things that might be expected that were not implemented (don't be too exhaustive – just some items that might help the marker to set their expectations properly); and things where the UI is implemented but the functionality is not necessarily as per a full implementation (e.g. the user can enter and send peer feedback but the result isn't added to a list of peer feedback previously sent).

(c) Design decisions. Include a brief entry outlining each of the following:

1. Colour scheme: Describe the basic type of your site's colour scheme (e.g. monochromatic, or complementary). List ALL the colours used in your prototype and their role, showing a block of colour and the RGB value. E.g.

RGB: 53, 94, 145 Navigation pane background

2. Borders scheme: Provide a description of your approach regarding choices of lines and borders, backgrounds (including images), use of white space and any other methods for grouping or segmenting content on your site. Best formatted as a table or bullet point list – 4 to 10 entries. You don't need to describe normal text alignment practices unless you've done something you especially want to highlight for its effect on grouping.
3. Fonts scheme: As a table provide one row for each font used in the interface. Provide the font name (including important style aspects, like bold or italic), some example type and a description of its role in the interface.
4. Animation effects (optional): List (table or bullets) any dynamic effects implemented in your prototype that are not instantaneous.

(d) Resources used: List all external resources used for the HTML Prototype as a bulleted list or table including the source and a description of the role it plays in your design. This should include any image files that you did not create yourself. You also need to list Bootstrap components used in the design. You do *not* need to list the base JQuery library, or any code from tutorials/labs or lecture slides of this course. You do need to list any **code snippets from other external sources (e.g. Stack Overflow) that are greater than 3 lines of code or more than 300 characters in length**. Such code snippets must also be indicated by comments in the source code (see below).

Include all this as a single document called Design.* with * equal to .doc, .docx, .rtf or .pdf as appropriate to the format of your document file.

2: HTML Prototype

Implement the design described in your Design Documentation using standardised Web browser technologies of HTML5, CSS and JavaScript only. You may use any of the following resources as a starting point or to implement useful features:

- Example code as developed in the labs/tutorials, as well as any code from the lecture slides
- JQuery (necessary for Bootstrap)
- The Bootstrap 'bare' template. See https://startbootstrap.com/template-overviews/bare/ for description; preview at https://blackrockdigital.github.io/startbootstrap-bare/ . You can download a .zip of the complete package from https://codeload.github.com/BlackrockDigital/startbootstrap-bare/zip/gh-pages . No other Bootstrap template is allowed
- Bootstrap components – see https://www.cs.auckland.ac.nz/courses/compsci345s1c/tutorials/Prototyping/#user-content-4-bootstrap and https://getbootstrap.com/docs/3.3/components/ . List each component you use in part 1(d) of your Design Document
- Code snippets you find on the Web (list in part 1(d) and provide a source code comment if over 3 lines in length or greater than 300 characters)
- Image files (for backgrounds, icons, etc.) from the Web where re-used within their terms of use (list in part 1(d) of your Design Document)

You cannot include complete code files other than those listed above (i.e. no other CSS or JavaScript libraries, although you may use specific snippets or functions if you document the resources used clearly). You cannot use an IDE that automatically generates code for you. Your solution cannot use a DBMS – where you wish to persist data use DOM and Web Storage methods as outlined in lecture.

Source Code Comment requirement: For code snippets from external sources (e.g. Stack Overflow) where you have copied more than 3 lines of code or more than 300 characters you must provide a clearly marked comment including the word "SOURCE" followed by the source URL. Indicate the end of the copied block with a comment including the words "END SOURCE". You must include this source code commenting even if you have modified the code from the original form – feel free to indicate as further comments after the heading where you cite the source what modifications you have made. In each case also create an entry in part 1(d) as an aid to the marker. This requirement applies to HTML, JavaScript or CSS code.

Hints

Less may be more. Note the mark scheme at the end of this handout. Points are awarded for a prototype that has a good visual design and illustrates the idea of the full implementation. Concentrate on making good design choices rather than implementing maximum functionality. Corresponding to the 3-5 screenshots in your Design Document, just 2-4 'clicks' (probably with some text entry in between in some cases) should be central to your implementation. While 2-4 clicks may be central, you'll probably implement several more click events – e.g. to close a dialog as well as open it; or to Post a comment as well as to start editing one. With use of DHTML techniques you may have only one HTML file – three HTML files is a recommended maximum.

Dummy data. Given the role of peer feedback in the problem-based learning methods, you'll need to pre-populate your prototype with a user solution or draft solution, some peer feedback and/or peer work for feedback (much of which can use *Lorem ipsum* filler text liberally especially for the student solutions, but not so liberally that we lose a sense of what's going on with the interface). Also, while it's not ideal, you may also want to hard-code some UI feedback rather than implementing functionality to do DHTML manipulations with user input data (these short-cuts should be documented under the 'out of scope' part of the Walkthrough in the Design Document).

Iterative design and development. It's recommended that you start your Design Document before you begin any serious development on your prototype. See what decisions you can make and write them down. Then, after you've done some implementation you'll want to revisit your documentation and check, and possibly revise, some of your design decisions. Iterate. You might've had an idea (I'd hope!) about colours, fonts and borders before you started writing code, but this may've changed and will probably become more refined during implementation. Once you've finished the prototype you'll want to take screen shots and probably finalize other details in the design document.

Resizing. The markers won't be trying to stress test the performance of your prototype under window resizing. They will view the prototype with an approximately A4 size (landscape) browser window. Of course exact performance can still vary for many reasons, so it's wise to use sizing and positioning that is reasonably resilient (e.g. minimize absolute widths and positioning – and don't make the last few millimeters of A4 be critical). The marker will aim to align the look to the screen shots from your Design Document if possible.

Submission

Before you submit CHECK THAT ALL YOUR LINKS ARE RELATIVE so that the markers can unzip your folder and everything will work.

Submit a single zip file (and use ZIP format, not RAR, 7z or other) that contains the following via the online Dropbox by the deadline:

- Your Design Document
- Your website prototype, including home page and any other pages that the user navigates to depending on your design, with any supporting files that are needed for the site to be run by the marker. Include all HTML files, style sheet files (unless you did all styles internal or inline), JavaScript files (again, unless internal to the HTML files) and image files. The root page (where the marker is expected to start) should be named `index.html` or `home.html`.
- Please name your submission file using your UPI (e.g., bpli001.zip).

You can make as many submissions as you like, but only your last submission will be marked.
You should plan to spend no more than 24 hours on this assignment.

Marking Guide

Markers will assess your deliverables out of **45 marks** as follows:

Criterion	Features	Marks
Design Document		
Context and Walkthrough	Understandable – clear and concise; complete	5
Design decisions		5
Design Document presentation	Overall document conveys a professional impression both in content and style; is easy to understand and navigate. This includes using correct grammar and spelling.	5
Prototype		
Getting it right*	Deductions will be made for specific errors in visual design (no identifiable errors equals full marks) [also will deduct for implementation errors if explicit – e.g. 404 Not Found]	10
Making it great*	Overall judgement of the visual design effectiveness and aesthetics: e.g. impressive, workmanlike (i.e. competent but not impressive), too sketchy; mark corresponds to UoA grade scheme – e.g. 8/10 for an A-	10
Scope	Takes on substantial and relevant interaction tasks; e.g. navigating between own and peer work, and relevant dynamics such as adding to a comment thread	10
Total		45

* In each case the marker's review will include use of colour, fonts and borders/grouping, and animation effects if applicable, and overall adherence to design and aesthetic principles. For 'Getting it right' the mark moves down from 10 as errors are identified. For 'Making it great' an OK design might get a 7 – a higher mark needs to be earned by exceptional application of techniques. The 'Getting it right' and 'Making it great' scores will be correlated (it's probably not great if there are lots of design errors), but they have some independence, too – e.g. a design that makes few explicit errors could still fail to have made excellent use of visual design elements like colour, fonts, borders and grouping or pleasing but subtle animations.