

P2 Project Design (25% of project grade)

Design Guidelines

For the purposes of this project, "design" means the visual or graphical design of the site as well as the implementation details that underlie it. Your design document should describe what the finished site will look like and how you will make it work.

There is an alternative view of the design that can prove helpful. This view of the design is that it should be so clear, detailed, and complete that you could, in principle, mail it to a team abroad and they could implement the website without further consulting you or the client. The idea is to leave nothing implicit, no unspoken understandings between the parties.

Mockups

An important part of the design will be a set of mockup images: pictures of what you want each page to look like or perhaps of common elements such as the logo or the navbar.

You can create the mockups in your favorite graphic/image editor tool (Photoshop, Pixlr, Paint.net, Illustrator, Power Point, etc.). Once you've done them, save or export them as web images (formats such as JPEG or PNG) and upload them to a folder named `mockups` in the `public_html/Documents` folder of your shared team account.

If you have not received necessary images and text from your client, such as for a logo or the history of the organization, or you have imaginary client, feel free to use dummy images and text. In this phase, focus on the layout of your pages: sizes, alignment, fonts, colors, borders, and so on.

Design Document

The final product of the design phase is a webpage that contains a structured essay - a combination of text and images of your mockup pages. The document (webpage) should contain these components:

1. **Header:** A title, the author names, and date.
2. **Administrative Details:** The name of your client, and his or her position in the organization.
3. **Purpose, goals, and audience:** A summary paragraph to remind the reader about the purpose, goals, and audience of your website. Make it brief and succinct.
4. **Navigation Structure:** A brief discussion about the site's navigation structure using the terminology learned in class and the notes on navigation structures. This is *not* a sketch of your navbar (or whatever), but instead a map or description of which pages connect to which other pages. (It sometimes better to describe a clique in words than to try to draw it.) The goal is to explain why the choice of a particular structure makes sense for your website.
5. **Page Layout and Appearance:** A description of design decisions that are **universal** to your site. You should specify in writing the fonts (types and sizes), the color scheme (for text, background, links, shadows, borders, etc.), navbar position in the page and its appearance,

banner, footer, the social icons bar, or background images (if any of them is applicable). In addition to the textual description of these elements, you should include a mockup image of this generic page that has only these common elements and no other content. Finally, you should describe how these universal design decisions fulfill certain goals of the website and its audience.

6. **Content:** This is the longest section of your document. For every page, you will need to give a text description of the content of the page as well as an image of the mockup of the page. The text describes what goal(s) this page fulfills, and how design decisions specific to it contribute into fulfilling such goals. The reason why you should describe the content is that you might not have yet received the actual content from the client and in your mockup of the page you will be using fake text, fake images, etc. While the content can be fake, you should try to use meaningful headings and captions to convey its meaning. Finally you should describe the design decisions that are unique for this particular page (if any). Here is an example:

Images of e-board members are stored as thumbnails to reduce the loading time of the page. They appear in a grid format on the left side of the page. Each thumbnail is of size 150x150. When clicked, they will open as larger images (320x480) on the right side of the page, surrounded by text describing the member's duties and reasons for joining the org. The chosen thumbnail will have a surrounding border in gray color, to provide visual feedback to the user for the selected content.

7. **Minimum Requirements:** The website must include four distinct JavaScript applications (for example, rollovers effects on images or menus, user-defined functions, form validation, animations etc.). The four applications must all be different from each other.

Clearly indicate how each of you will be fulfilling these minimum requirements.

Please note that each of these items must be constructed entirely by you and not copied from external websites on the Web (though you can modify code from our class materials). Once you fulfill the minimum requirements with your own code, you are allowed to get code from the web (with source attribution) to incorporate certain advanced components (e.g. image galleries with transitions), but they will not count as part of the minimum requirements.

8. **Plan:** Provide a clear, specific description of your plan for building the site. Outline the tasks to be accomplished, the deadlines for those tasks, and who is responsible for them. Include a list of files and folders, which will help each partner link to files her partner has built. Include a complete list of everything you need to get from your client (text, pictures, sound, etc.) and when you expect to get them. To the greatest extent you can, try to have all of these things by the time your design is done, so that nothing will delay your coding.

Now that you know what should be in your design document, you can read the following sections to get more details on the parts of this document.

Create the Design

Your design should say *what* you will create, and it should be clear, complete and concise.

Design includes the organization and navigation of the site. What pages will there be? How will they be connected? What sequence of steps will a user need to take for a common task? For example, if users most often visit your site to find out what the next game is, describe the path they will follow to get to that information. Design also covers the layout and appearance of each page.

Think a bit about your target audience, and remember that some of them may have certain disabilities, thus, design with accessibility issues in mind (see reading below).

Also, remember that not every user will have precisely the same size screen you have, so think about page layouts that are flexible and will still look okay in a variety of sizes. Document the decisions you make.

Don't be afraid that by being specific in your design you won't be allowed to change your mind later. You may change your mind during coding; you just have to document it in a [changes.html](#) document that you'll submit along with your code. See the P2 Project Coding.

Design Resources

- The first chapter from the book [Designing Interfaces: Patterns for Effective Interaction Design, 2nd Edition](#)
- [Introduction to Web Accessibility](#)
- [One Hundred Things Every Designer Needs to Know about People](#)

Modularity

Rather than duplicating code for elements of your site that appear on multiple pages, you should handle repetitive items by using shared files of JavaScript code.

Repetitive elements include things like:

- banners, navigation bars, and footers,
- navbar button rollovers in navbar code,
- style sheets (there should not be separate style file for each page, but a single "main" CSS style sheet).
- JavaScript functions - when the same JavaScript function is used on multiple pages, it should be defined in an external JavaScript file (rather than being included and re-defined in each file through the use of SSI).

Depending on the project, achieving modularity may be a large task. Therefore, implementation of modularity should be shared by the partners, since assigning it to one partner only may result in an uneven work load.

Planning your work

In a professional setting, many people may be involved with implementing the website after it's designed. To avoid them stepping all over one another, the implementation must be planned out, determining in what order the parts will be built and by what time. Some parts may be built early to

allow for more testing or because subsequent parts depend on them. We also need deadlines on various parts because management needs to know how far the team is behind schedule (software projects are almost never on time). Finally, we need to determine who will build each part, so that everything gets built and nothing gets assigned to two people.

In this project, even though there are small teams, all these issues arise. Ideally, a good plan will allow the teammates to work independently, without having to meet constantly to discuss "who's doing this page?" and "what did you call your page that I have to link to?" and the like. Thus, your design document must also include a *plan*.

The plan must cover all of the following:

1. A complete list of all files you need to create, including HTML files, CSS stylesheets, part-html and JavaScript files used for modularity, GIFs for buttons, JPEGs from a digital camera, etc. This makes up a complete "to do" list.
2. An organizational scheme for the files. Note that this need not be, and in many cases **should** not be, the same as your organizational scheme for the website! For example, often all the navigational buttons will be put in a folder called "buttons," but that scheme obviously has nothing to do with how the content of the website is organized. Note that a common technique for listing files and folders is the indented list, like this:

```
3.      index.html
4.      about.html
5.      members.html
6.      ...
7.      buttons/
8.          home.gif
9.          up.gif
10.         down.gif
11.        ...
12.        images/
13.            member1.jpg
14.            member2.jpg
15.        ...
16.        main.css
17.        scripts.js
18.        header_nav.part
19.        footer.part
20.        ...
```

3. A person who is assigned to be responsible for each file in the complete list. That way there will be no confusion about who is doing what, and everything will get done. An exception to this may be the external style sheet - defining CSS for the project may be a lot of work, and it is more equitable to share this task between the partners.
4. A deadline for each file to be completed. This is not the same as the coding deadline! It's true that everything must be completed by the coding deadline, but (as we described earlier) you'll want to be able to keep track of whether you're on schedule or not. Resolve to check on each other's progress. Build in some time for testing.

Our intention is for this plan to be helpful to you; make sure it is.

A good way to organize this information is to create a table with the columns: file name, file description (one line description), deadline, and assigned person (items 1, 3, and 4), see also the example.

Submission

Your document, titled `P2_design.html` should be uploaded in its learning. Use a subfolder `images` to store all your mockup files. Make sure that the images are saved in a Web format (not Photoshop or other proprietary format), so that their size is not big. Zip everything together and submit to its learning, provide also a web link with the `P2_design.html` page under the main project page. See the course schedule for the due date. Late assignments will receive no credit.

No changes after the submission

Please do not make any changes to this document after the submission deadline. While you will need to make changes to your design plans, they will go to a separate document `P3_changes.html` to be submitted at the end of the coding phase. Thus, keep track of the changes, but not in the `P2_design.html` document itself.

Grading

Your document should include all components described in the Design Document section (few pages above). Your mockups should reflect the text description of the pages and be as close as possible to how the coded pages will eventually look like. We will look at how well the mockups convey the purpose and goals of the website as well as whether they are appropriate for the intended audience. For example, if one of the target audiences is children, how well do the colors, fonts, images, and the general layout speak to such an audience (it's cheerful, bright, not overcrowded, etc.). Your writing should be formal, clear, and concise.

Example

These examples were drawn from earlier semesters when the course was somewhat different, but may be helpful. Here is an example of a P2 design document:

http://www.idi.ntnu.no/~michaig/IT2805/exampleproject/p2_designexample.html

Notice how the design document is close to the final website, but there are changes which were described in the `P3_changes` document, when the code was submitted