

In this project you will design an interactive user interface for a smart object. You can choose from one of the smart objects we have explored in class:

- Smart shower
- Smart shopping cart
- Smart water bottle
- Smart toilet
- Smart flower pot / garden

You can stay with the smart object you selected for in-class activities, or you can shift to another.

We can't fully create these smart objects, but we CAN think about the design of the user interface that will connect the user to the various features of these objects to address the needs we identified during our interviews in week 2.

## Project setup:

You will complete this project in several phases, which we have started on in class.

### 1. Design (20%):

- a. Gathering design requirements, Sketching design alternatives,
- b. Sketching design prototypes, Feedback on designs for each 'level' you complete (C, B, A).

### 2. Implementation (50%)

### 3. Final Documentation (20%)

### 4: Presentation (10%)

In a real UI design scenario, we would do more iterations of designing and evaluation, before coding, but the goal here is to get some experience at multiple stages of the process and to get some experience implementing an interactive user interface.

## Project Deadlines:

### Sept 26th at 11:59pm: Design

Submit 1a (exploring designs) and the 'C-level' UI design prototype with a write up of the feedback.

\*\*\* Edited to add: reflect on how the project is going, and write down what you have accomplished and what you intend to do to say on track to finish.

### ~~Oct 9th at 11:59pm: Code~~

~~Submit the code for your implementation~~

### ~~Oct 11th at 11:59pm: Documentation~~

~~Submit the documentation for your project, presented on your public webpage~~

### ~~Oct 12 and 14:~~

~~Presentations, sign up sheet to follow....~~

**Friday October 14th at 11:59pm- Code submission**

**Sunday October 16th at 11:59pm- Documentation submission**

**Presentations via Zoom Monday 10/17, Wednesday 10/19 (and Monday 10/24 if necessary)**

Reminder: every day late, is a letter grade (10%) off. Turn it in- it'll be ok!

## **Edited to add: Suggested timeline**

If you want to be on target for completing the project, here is a suggested timeline:

~~Design 1a + C design prototype: Aim to complete by 9/19~~

~~C-goals: Aim to complete by 9/25~~

~~B-goals: Aim to complete by 10/2~~

~~A-goals: Aim to complete by 10/9~~

## **Collaboration policy:**

This is an **individual project**, which will give everyone experience developing a user interface with html, css and javascript. The material presented in the tutorials will give you a starting point. The code you submit must be your own.

However, you are encouraged to ask questions on the Discord channel. You can include code snippets, and ask for help debugging. If you are answering questions, you can also post code snippets or links to useful information. You can also show your code to a classmate to ask for their help. Your peers are a resource, and I am strongly in favor of helping each other, as long as everyone submits their own work.

## **Project requirements**

You will design and implement an interactive UI mock-up of the interface on your smart object. The UI for your smart object in its final form would *not* be on a computer screen or controlled through a mouse and keyboard. But, we are going to create the visual interface elements, controls (buttons, sliders, input forms...) and visual indicators that will address design requirements for your smart object.

### **Everyone (those completing C-goals, B-goals and A-goals):**

- Design requirements: We started to develop our design requirements in class- in your teams and through interviews. You will elaborate on these and document it on your webpage. Some of the design requirements you captured were concerned with physical aspects of your smart object (Eg. wheels on shopping cart get stuck). We are going to focus on a visual user interface that enables users to control the object or view data it is capturing through its sensors (Eg. visual indicator of how well the wheels are turning. button to reset a locked wheel....) or suggestions provided by smart functions.
  - Write down the assumptions you are making about your smart object (eg. it measures the water level and temperature, it tracks how much water someone drinks through a sensor in the opening....)
  - Create a clearly written list of design requirements for the user interface. Use the design goals your team generated in class based on in class interviews as a starting point. These can be design requirements you address in your implementation, or ones that could only be realized through a longer design process (which you can label as 'future goals').

- Add to the list of design requirements. Find 3 people (friends, family) outside of class to interview about your smart object. List the interview questions you posed. Summarize the responses.
- Sketching design alternatives
  - Choose 3 interesting design challenges to explore through the 10-plus-10 method.
    - For instance, if I were developing a microwave+toaster oven combo device, my design challenges might be- user wants to figure out how long to cook something for and in what mode, user wants to be able to quickly heat up foods they repeatedly heat up, user wants to avoid over or under cooking their food.
    - \*\*\*\*\* **UPDATE:** See my announcement about this design alternatives sketching activity. Instead of sketching 10-plus-10 sketches, do 10-plus-5 MINUTES. Link to announcement here: [Design alternatives sketching: aim for 15 minutes per design challenge](#)

## C-goals:

For the C-goals you will create a basic UI for your smart object, within a single view and using simple controls and visual elements. The focus here is on addressing a subset of the design requirements through basic features to input simple settings for the smart object, and to view simple status data- which may be static.

- To enable basic use of your smart object, create a list of the controls you are including in this UI and the indicators you want display, and how these connect to the design goals you captured.
- Create prototype sketches of this interface BEFORE you implement it.
  - Include these sketches in your documentation
- Show your prototype sketches to 3 people (friends, family members, classmates). Record the feedback.
  - You don't need to follow an evaluation methodology here, as we have not yet covered this in class, but get some feedback.
- Implement this UI using html, css and javascript, and serve your webpage using localhost (details to follow). Test your page in the Chrome browser, which is what I will use when I run and test your code.
  - Elements should be clearly laid out, features should be discoverable, and you should provide feedback on user actions.
- Your interface for the smart object should have fixed dimensions (non-resizing, non-responsive layout) with size and aspect ratio that is appropriate for your object.
  - For instance, the smart water bottle could have a front and back UI, in two sections on your page, with an appropriate aspect ratio for a water bottle (eg. tall and narrow, or wide if it wraps continuously around the display). Clearly indicate where the boundaries on UI for the smart object- in one or several regions.
- Outside of the region for your smart object UI, include a title for the project, your name and a link to your project write up.
- Outside of the region for your smart object UI, include a small graphic to indicate where this UI would reside on the physical smart object.
- Within the fixed area for your smart object UI, implement the targeted goals, based on your design prototype. Remember- the goal is to enable users to control the smart object with basic settings and view a snapshot of the information your smart object may collect. Elements should effective use of visual design principles, and appropriate feedback for interactions.
  - For example, if this were a microwave/toaster oven project (as described above), my UI for a C-level implement may include :
    - Controls: ability to set a timer, ability to select heating mode (e.g. bake / broil / toast / bagel / pizza / convection bake), ability to start the oven
    - Display: display the temperature, time and heating mode the user has set; display the current temperature inside the oven (non-updating); display the number of times the user has run the oven that day- increments by one when they start the oven
    - Design choices: Layout of elements allows the user to see the food inside

## B-goals: all of the C goals, plus:

Pick one of the following design directions. Choose one that interests you, or that is a good fit for your smart object.

For the one you choose: UPDATE: use sketching appropriately to think through your designs. Either sketching novel user interface views you are adding, or sketching how your UI will update on events or simulations, or sketching the mobile interface, or sketching the alternate user profiles. Use sketches to think before you code.

- ~~Sketch design alternatives with the 10-plus-10 method~~
  - ~~Can use the 10-plus-5 minutes method, detailed above.~~
- ~~If Sketch your design either the prototype UI or your design plan~~
- ~~Get feedback from 3 people, and record it.~~

### **1. Enable the user to input a complex set of selections:**

The basic interface you implemented for the C-goals may not allow for more complex inputs or user selections. For instance, perhaps they need to input a sequence of shower temperatures and nozzle settings, with timings, something that is more complex than simple selection within a single menu. Design and implement the user interface that will enable your user to input these selections. Ensure that they can see their selections clearly, have quick visual or auditory feedback on their selections.

### **2. Connecting to a mock secondary device.**

If one of your design goals involved integrating a secondary device- like a mobile phone or a watch- include a region on your page which will display an interactive UI of the mobile device. For instance, perhaps you have a mock-UI for shopping list creation on your phone, which you can load onto your smart shopping cart UI. Interactive inputs within the mock mobile device UI, should update the smart object, and vice versa. Clearly define in your write up what the role of the secondary device is, with respect to your smart object. Why have a secondary device?

### **3. Display information for different mock users or situations:**

Go deeper than you did for the C-goals, to display captured information from sensors in the smart device to the user. You do not need to include complex visualizations here, but if one of your design goals is to allow users to see how much water they are drinking, how would you display this to them? Sketch design alternatives. If your smart toilet is tracking health-related data, how would this be visually presented to the user? If the user has added 10 items to their cart, how would you help them see what categories their products are in and how much they cost?

Within your UI, display basic indicators of the captured data for at least 4 different model users or example situations. Create these profiles as json files or as objects. You can include buttons outside of the UI to switch between these data profiles.

### **4. Simulate the smart object in use over a time frame:**

You can't push your UI around a real grocery store. But, if one of your design goals is to highlight where you are in the store, or show coupons for the products near the shopping cart, can you simulate this behavior, in a very simplified store? Outside of the boundaries of your smart object UI, you can include a 'simulate' button, and your UI should act as if it were being used. Or, if your smart object has a program- like the smart shower programmable shower program- you can include interactions or controls to activate this program, and then show a real time (or accelerated) version, of how the UI changes as it runs the program. All visual indicators or visual elements should update dynamically as you run your simulation. For something like the smart water bottle, this might be a simulation of someone drinking water occasionally throughout a simulated period of time, updating

the display of how much water they are drinking. You can use a timer and can pre-script the program, to control how the object should change.

## 5. Propose your own:

It has to be interesting, sufficiently challenging, and it has to involve developing of an interactive UI using html, css and javascript. Run it by me first.

## A-goals: all of the C-goals and the B-goals plus:

Complete a second design direction from the above list for the B-goals.

## Documentation:

For documentation: assume that someone is encountering your project for the first time. This documentation must be publicly available on your portfolio page.

- Present your sketches- both the design exploration and the prototype interface design
- Write up your requirements and your interview findings
- Describe your interface in detail:
  - Explain the features and controls
  - Include screenshots to illustrate
- Include a 2-3 minute demo video, showing your interface in action. The easiest way to record this is with a screen capture tool, which also captures audio- such as Quicktime. Use a voiceover to explain your application. Demo videos should be sufficient on their own, but can reference your documentation. Include the name of the project, your name, the project components, and how your application works. You can present it on your webpage or on youtube, but it must be linked on your webpage.
- Include a link to your source code on github.

## Presentation:

Give a 5 3 minute presentation of your designs and interface.

-----

## Clarifications:

To recap: the C-level interface should allow the user to:

- provide a small set of the highest priority inputs for the smart object
- provide a basic snapshot of the status of the smart object, or simple data collected by the smart object.

To take the smart fridge as an example:

A C-level interface would be an implemented UI (html, css, javascript), in a single view, served using localhost.

As a basic interface, it could (for one moment in time):

- display the temperature of different compartment of the fridge,
- display how full different compartments of the fridge are,
- display how often the fridge has been opened and an average length of time left open,
- display the date and time,
- allow the user to input a desired fridge temperature

- allow the user to input how long to allow the fridge to be opened without an alarm

The implemented UI would be informed by graphic design principles (week 6)- simplicity, layout, effective use of color.

This implementation would be supported by 'pre-design work'- interviews, sketches, lists of requirements (20% of the project grade).

B-level and A-level examples for the smart fridge (choose 1 for each level):

1. Enable the user to input a complex set of selections: Within an 'advanced settings' sub-page, the user would be able to input preferences for how much of different food groups they want to have in their fridge, and which foods they want to have tracked, so they can be notified if they run out.
2. Connecting to a mock secondary device. Create a mock phone interface that displays fridge status and allows the user to input preferences. The mock phone would have additional features appropriate to a phone UI, vs the fridge UI. These inputs should update the main UI view, and vice versa.
3. Display information for different mock users or situations: For 4 different users, display a range tracked data points and varied input preferences, that would help these users understand usage of the fridge. Go deeper than the C-interface.
4. Simulate the usage of the fridge over the course of a mock day. Pre-programmed data, would run on a timer, with the fridge updating as the mock-day plays out.

As before-

The implemented UI would be informed by graphic design principles (week 6)- simplicity, layout, effective use of color.

This implementation would be supported by 'pre-design work'- interviews, sketches, lists of requirements (20% of the project grade).