

Assignment #2: designing a front-end to an AI/ML infused system

Learning objectives

- Describe specific user needs where AI/ML can be useful, and user constraints for such applications
- Recommend and justify solving for one particular user need/scenario
- Apply heuristics for the design of ML interaction to designing interfaces that match that need, and methods to mitigate the costs of failure in AI/ML infused systems

Deliverables

(see below for description)

- Two storyboards
- A list of technical requirements for the AI/ML part of your app
- Mockups for app
- Two tables for the design decisions you made in this app

What you'll do

In this assignment, you will create a mockup of the front-end to your loan application from Assignment 1. Assignment 1 gave you a chance to see what's happening in the technical back-end; here you will make design decisions for what users see in the front end.

The mockups you create will *not* be functional. While designing these mockups, keep in mind the heuristics, worksheets, and other tools from class for designing AI/ML infused interfaces.

We recommend that you read this document in its entirety before you start on your assignment. This assignment requires you to do a lot of thinking that can't be rushed -- the kind that you can do while you're in the shower or on a run. So, get started early to give yourself time to think!

Application

At its core, your application is a front-end to the classifier you built in Assignment 1: given information about an applicant, the classifier predicts whether or not they will qualify for a

mortgage. You can ask for the same information as was available in your training set for Assignment 1.

But how should users interact with this app? That's what you'll design in this assignment.

Persona for this assignment

In HCI, "persona" is used to mean a stylized description of a user who you are designing for. For this assignment you are designing a mortgage related app for a couple (Pat and Jan) who are applying for a home loan. They are in their early 30s, and you can assume that they are familiar with using both desktop computers and smartphones.

They've both found a beautiful house they would like to buy in Bellevue, WA. This is the first time in their life they are looking to apply for a home mortgage. Because Pat and Jan have both worked for several years, and always paid their bills on time, Jan is confident that they would qualify for a home mortgage. Pat isn't so sure because she has heard there are many factors that go into mortgage decisions.

Design your application to be used by Jan and Pat, either together or individually, on either a desktop or a phone.

Step 1: Draw two storyboards

A storyboard is a visual depiction of how someone might use your app in real-life. The focus of a storyboard is people, and how their life is better after using your solution. The focus is not your app and its screens. To focus the story on the human and not the computer, do not show any details of the UI. Draw one storyboard each for a different way in which Jan and Pat could use your loan application app.

For high scores on this deliverable, your storyboards must:

- Each depict a realistic scenario that your app can help with.
- Be very different from each other
- Not show your UI in detail

Deliverable 1: 2 storyboards, at least three panels each.

Scan/photograph handwritten storyboards. **(10 points each = total 20 points)**

Note: scans/photos must be decent quality, clearly legible images. If we can't read your storyboards because of poor image quality, we can't give you the points.

Step 2: Choose one storyboard, and list out AI/ML requirements for your classifiers

Choose either storyboard and determine what needs to happen technically with your classifiers to realise this scenario.

You will deliver a list of technical requirements for your classifiers. For example, if your app has a “basic” mode where users are only asked for limited information (e.g. loan amount and income), and an “advanced” mode where more information is required (e.g. loan amount, income, location of property, co-applicant information, etc.) then you’ll list that you need two different classifiers: such as a basic one from Assignment 1 part 4.B, and a more elaborate one from Assignment 1, part 4.C. Similarly, if your application needs to show how confident it is in prediction, you need that information (you can calculate a rough estimate from your confusion matrix).

For high scores on this deliverable, your requirements must be:

- Realistic (i.e. you can create those classifiers given your dataset)
- Minimal (ie. not require any more technical work than is minimally required)
- Complete (i.e. if you showed them to a classmate who has aced Assignment 1, they would be able to implement them)

Deliverable 2: List of requirements

Max 1 page, recommended: 2 paragraphs or a list format. Be sure to describe the requirement briefly, but in enough detail that we can understand what the requirement is and why you think it’s needed. **(total 15 points)**

Step 3: Mockup UI

For your chosen storyboard, mock up the user interface. When you do so, follow the heuristics from Microsoft’s ML interfaces paper, and use worksheets from class for matching mental models, designing for different stakes, and maybe even co-adaptation.

Show the main screens of your app that implement the functionality required for your storyboard (max 5 screens). Please do NOT mock up screens that are not central to the storyboard (e.g. login screens etc.) If it helps you, you can submit screens as gifs (with animation). Your mockup does NOT need to be a functioning prototype.

Use whatever tool you are most comfortable with (We cannot pay for licenses, sorry!) Lots of students like [Invision](#) or [Sketch](#), or [Figma](#) (it’s also free). You can also use a low-fidelity tool like Balsamiq, but we don’t recommend it, as it may not let you show some interactions.

Whatever tool you use, please export the screens as png or PDF. Do NOT submit links to a live prototype. We can’t timestamp it.

For high scores on this deliverable, your mockup must be:

- a) Five screens or less (3 points)
- b) Non-functional (2 points)
- c) Show realistic data and interactions (10 points)
- d) Use heuristics for AI/ML infused interfaces (25 points)

Deliverable 3: max five screens of your prototype as PDF, PNG, or GIF.
(total 40 points)

Step 4: Write up

Your write up has two parts.

Part A

In part A, talk about the design decisions your mockups make based on heuristics, mental models etc. Also, assume the technical requirements from Step 2 above are unchanged, and include design decisions your mockups could make, if you had more time for design work.

We recommend you use a table like below:

Technique	User need	Design solution
Mental model	Users believe loan decisions are made by ...	<Your solution>
Heuristic G1: Make clear what the system can do.	Need to clarify that the app does not guarantee a mortgage...	<How you might mockup the solution, if time permitted>
...

Part B

In part B, we ask for heuristics or other requirements your mockup will not be able to match, even if time permitted. For example, maybe your classifiers are unable to say why they predict a particular outcome.

We recommend you use a table like below.

Technique	User need	Why unable to solve
-----------	-----------	---------------------

Heuristic G11: Make clear why the system did what it did	Would be better to say why the system predicts loan would be approved or denied	Using an SVM classifier, and so it's hard to tell which input variable mattered most.

For high scores on this deliverable, your tables must:

- a) Describe at least ten heuristics in Part A (ie. ones that you were able to mock up + ones you could if you had more time)
 - i) At least five heuristics that you were able to use in your mockup
- b) Describe at least ten heuristics total that you were either able to use in your mockup, or are impossible to meet in your design
 - i) This is a confusing way to put it, but here is what we mean: *the number of heuristics you were able to implement + number of impossible to implement heuristics > 10*

Deliverable: two tables for Part A and Part B

Grading

10 points if you have at least 10 heuristics in Part A, of which at least five are implemented

- + **15 points if** the number of heuristics you were able to implement + number of impossible to implement heuristics > 10