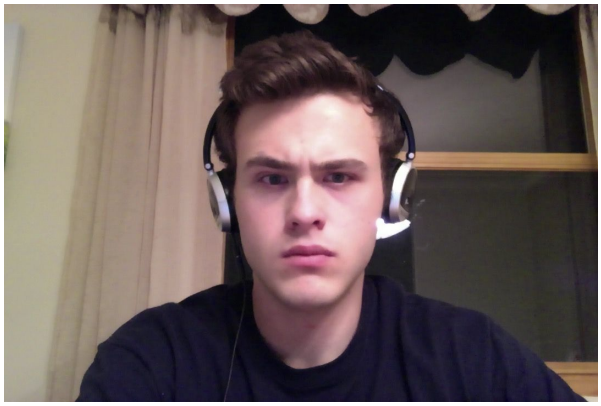
DSGN 1 Portfolio

Tyler Sorensen

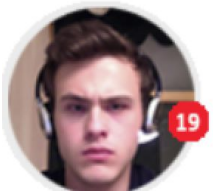
A12434894



(I don’t always look angry)

<u>This class portfolio serves as a collection of my design thinking from September to December (2015).</u>

Abstraction of biometrics for user health profiles



Tyler Sorensen

The objective here is to provide a simple health metric for use in a user-facing health digital profile. People typically care about health for two reasons: quality of life and length of life. Quality of life tends to correlate with length of life, so this particular metric focuses on length of life.

This abstraction is a distilling of key biometrics into a single metric with intent to primarily serve as an indication of general health. The abstraction is a user's *biological age*, calculated from a dynamic algorithm considering numerous biometrics, demographics, and statistics.

Biological age is the abstracted metric for a few reasons. First, it doesn't carry the burden/difficulty of predicting life span, but provides similar meaning. Lifespan is tough to accurately predict. However, an objective look at biomarkers like telomere length, individualized genetics and polymorphisms, systemic inflammation, risk for disease; and other data points such as lifestyle, geography, and actual age (among many other things), gives an idea of how old your body actually is.

The number in the red circle alongside the profile picture is the biological age. Say you are a 33 year old man with a biological age of 50. Seeing your face with a seemingly incorrect age inside of a sickly red circle is highly motivating. It plays into the personal age crises that most of us experience. You see your health represented in a highly emotional metric: age. You are a semantic 33 year old, but a biological 50 year old. You've only got 25 years left. That's motivation to get healthier.

Feedback and the Benchmark of Sports Cars

The Porsche 911 is better than it's ever been. The two door sports coupe is going on its 53rd year and it's always the market *benchmark*. The 911 has enjoyed a legacy of fantastic cars, each model surpassing the last a multitude of ways. And while most car models of similar age can say the same, none evolved as well as the 911. While Corvettes and Camaros and Mustangs of prior eras are beautiful cars, they're miserable to drive. They don't turn. They don't stop. They can barely go straight for long enough that the chassis stays in one piece.

Every air-cooled 911 (built before 1996) is rising in value. Even the neutered and oft-mocked Porsche 912 (yes, a 911) is gaining in value. Why? The design. *Not the aesthetics*, those have hardly changed in 50 years. It's the driving experience that's changed -- for better and for worse.

Each iteration of the 911 is more faster and more capable than the last. From a business perspective, it had to be. And from about the 1970s (oil crisis) on, each iteration has been more efficient than the last. From a legal perspective, it had to be. And from those two standpoints, the 911 is better than it's ever been. But each successive generation has lost a little something, and it shows in the collector market. That *something*, can be explained in design speak.

A key component to *that something* is feedback. Think about driving down a twisty road. It involves the gas, brake, and clutch pedals, the steering wheel, and the shifter. These are the connection points between car and driver. The feedback to the driver communicated through these connection points is very important. The subtle vibration of the steering wheel communicates the texture of the road surface and the limit of tire grip. Which in turn gives the driver confidence and satisfaction. The manual gearbox affords a satisfying maneuver in it's own right. It's a gratifying skill to master beyond normal driving, but it provides a satisfying clunk when put into place. It's position in the gear tree is feedback as to which gear the transmission is in, allowing the driver to keep their eyes on the road. The small oscillations in the floor-mounted brake pedals communicate "brake feel", which hints at how much grip the tire has. All this feedback increases driver confidence. It allows the driver to push the car's limits and get more of the adrenaline rush that so many drivers crave.

Motor Trend defines driver confidence as a key indicator of a good sports car. It's considered the single most important attribute in their annual Best Driver's Car competitions. In recent times, the newer incarnation of the 911, the 991 (not a typo) has dominated the competition, partly due to success in driver confidence. But drivers still crave the analog tactility of the older models.

This feedback is increasingly absent in newer models. That's partly why the oldest 911s are typically the most valuable. Each successive 911 has lost a piece of that *something*.

In the engine, the 911 was carbureted until 1972, wherein it was then fuel-injected. It lost a little bit of character in terms of the raspy engine note, but was ultimately a better car because of it. Then, in 1996, the 996 generation switched from air-cooled to water-cooled engines. Again, the engine raspiness and character was diluted, but the car was better on paper. The raspiness gave an emotional and handbuilt character to the engine that is lost in the water-cooled cars. That kind of character doesn't show up on paper, but the personality oozes when driven.

In the steering, the 911 had unassisted steering until the 964 generation in 1989. The switch to hydraulic steering made for easier city driving, but at the expense of steering feel. In 2012, the 991 switched to electric power steering, which was more efficient, but numbed the steering feel even more. On paper, the steering system was improved. But in practice, feedback and tactility was lost.

It's happened to the car in several different ways. On paper, the car is better than ever before. But the feedback is missing. Corvettes, Camaros, and Mustangs have all had similar systems and similar feedback. But their driving dynamics don't hold up. Old 911s do hold up. And that's part of the reason why the 1973 Porsche 911 Carrera 2.7 RS is worth 7 times what it was 10 years ago. Progress has lost the feedback, tactility, and character of all cars, though the 911 has fared better than most.

Week 7: design in the world of business

At what point is innovation radical versus incremental? The first automobile is considered a radical innovation, even though it was essentially an alternatively-powered horse carriage. It wasn't entirely new, but it was also significant enough to not be an incremental innovation. The original iPhone fits in this category as well. The original iPhone was radically different from every other phone at the time, but it still was a phone. It did everything (and more) that other phones did, and more importantly, *it did it well. Excellent execution is what made the original iPhone a radical innovation.* It was both the most feature-packed and most easy to use phone on the market at the time.

In competition with Android phones, the iPhone has suffered from featuritis. The iPhone 6s is hardly recognizable from the original iPhone. The 6s is more complex, difficult to use, and capable than the original iPhone. But I think most people would agree that the iPhone 6s is a superior phone to the first iPhone. It may have some design shortcomings in poor attempts to accommodate new features, but the features themselves are valuable enough to overcome the difficulty of use. Sure, it's difficult and confusing to tinker with individual 3rd party app settings, but i'd much rather have 3rd party apps than none at all. That's not an excuse for the poor design execution, but a justification for featuritis.

The incremental innovative features tend not to cause enormous usability issues in and of themselves. A better camera, screen, and processor tend not to negatively affect the user experience. Some even improve usability. The TouchID sensor, for example, allows an effective solution to securely unlocking the phone. In this respect, the incremental features inspired and motivated by competition from Android phones actually *improved* the user experience. In the case of the iPhone, featuritis is well worth the cost. Especially in the presence of the competition, Android, which suffers from featuritis but doesn't cope nearly as well. Ease of use is one of the primary complaints against Android devices.

Effort 4

Understanding 4

Week 6:

"The day a product development process starts, it is behind schedule and above budget."

I don't have really any experience in product development, but I notice a few corollaries between Don Norman's Law of Product Development and video game development.

Video game development is incredibly difficult and it's becoming increasingly so. As gameplay and graphics progress, so does programming complexity. Game development, therefore, is becoming increasingly time and resource intensive.

In their embryonic state, games are very ambitious. Games like "The Last of Us" planned several features that didn't make it in the original game. "The Last of Us" was one of the highest rated games ever. The developer projected a certain A.I. logic to be in the original game, but couldn't make their deadline. Why not push the release date back? Well, they already did. And Sony, the developer's parent company, was hemorrhaging finances. They needed to release the game, even though it wasn't perfect. In fact, the sole complaint about that game was the A.I. logic. But it didn't matter. It's widely considered to be one of the best games ever made. It wasn't until 9 months later, when the story expansion was released, that the new A.I. logic was implemented.

"A delayed game is eventually good, a bad game is bad forever" - Shigeru Miyamoto, creator of Mario and Zelda

Unfortunately, gaming's business environment is dire. "Core" console games are cheaper than they've ever been. Games on cartridge-based systems from the 80's and 90's cost upwards of $90 in 1980's money. Games today are $60 in today's money. Why not raise the prices? Gamer's hold grudges better than most consumers. The first company to raise game prices will lose market share to rivals. What does that mean? Profits are dwindling, and most companies can't afford to delay a game until it's perfect. I assume this is common in design.

effort: 3

understanding: 4

Designing for problems?

Week 5:

Slips, Errors, and Resilience Engineering

As designers, we want to minimize slips and errors. That objective is simple enough, yet users still suffer from such things. Luckily, most users are intelligent, and can negotiate their slips and errors. But artificial intelligence can't. Well, not yet anyway. One of the problems with development of A.I. is in negotiating slips and errors. We can get a system to recognize a

slip/error/mistake, but so far that requires person to adjust the algorithm. Any mistake in A.I.'s interactions (at this point in time) is rule based. That's a difficult problem to solve.

Antifragility is a concept that I think is integral to explaining why we can negotiate slips/errors by ourselves and why A.I. can't. More significantly, I think it's what is holding true artificial intelligence back.

Antifragility: benefitting from uncertainty, randomness, chaos. (Antifragile, Nassim Nicholas Taleb, 2012)

Antifragility is *not* akin to resilience. A resilient system is a system that holds up to chaos, but does not benefit from it. An antifragile system thrives under stress, randomness, and uncertainty. Antifragility is most common in biological systems.

Antifragility is rather intuitive in hormesis. Hormesis is the process of subjecting small doses of stressor to produce a favorable biological response. The most common example is physical exercise. A minor/unpredictable stress produces a biological overreaction, wherein the organism actually *benefits* from the stress.

Evolutionary genetics is my favorite example of an antifragile system. Whereas hormesis is only effective until a point, evolution and genetics is the antifragility of information. An individual organism may die, but it passes on information in the form of genetic code. The gene pool enhances itself with stressors and uncertainty. Gene fitness improves with randomness and unpredictability. Hence, we don't have a singular 'uber organism'. It's difficult to prepare for all possible death sources, but this distributed form of information/genetic survival allows species to survive death.

Let's contrast these examples with today's artificial intelligence. Unpredictability is the bane of A.I.'s existence. If you throw an unknown input at current artificial intelligence systems, they don't quite know how to respond. Today's A.I. can't effectively respond to circumstances not already foreseen by a programmer. They lack antifragility in that an unpredictable input stalls the system, rather than benefiting it. The antifragile, true A.I. would learn from unforeseen and unexpected circumstances, much like we do.

A fragile system depends on prediction.

Week 4:

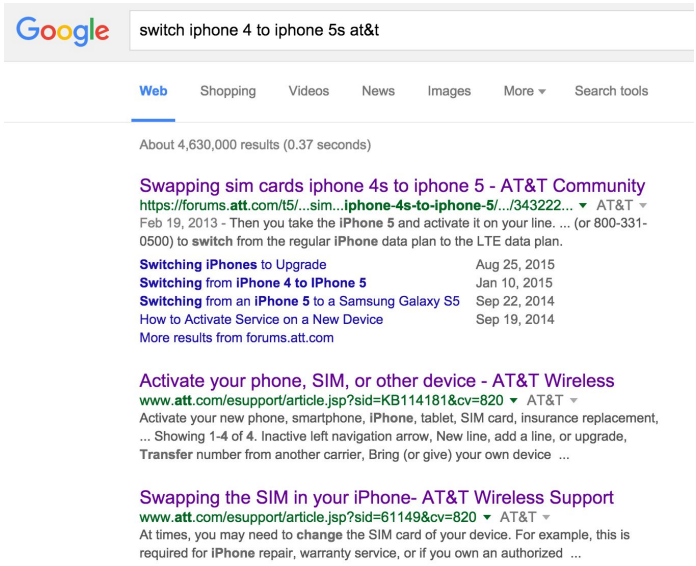
Yesterday, after several weeks of dread, I upgraded from an iPhone 4 to an iPhone 5s. Don't get me wrong, the technological upgrade is fantastic. But I still dreaded the process. This is not the

kind of dread to keep me up at night, but it's enough for me to procrastinate. I'm going to tie in my upgrading experience at AT&T with concepts touched on in chapter 4. This is more of a discussion of these concepts in the context of experience design than anything else.

This experience isn't as polished as it could be. This process is typically operates on a 'once per two years' basis, hence little incentive for AT&T to improve the process.

I received the replacement phone about a month ago. I've swapped phones before, so I had some idea of what needed to happen, but I wasn't entirely sure of the actual process. Next step, a Google search.

If there is ever a 'standard' for discoverability, it'd be Google. It's the basis for how most people discover how to do anything. For most products, discoverability shouldn't require a 'secondary' Google search. What I mean: a 'primary' Google search is fine so long as it functions to take users where they want to go. A 'secondary' Google search typically functions to answer some questions that the target destination couldn't answer itself. If a 'secondary' Google search is required, the primary destination site fails to properly design discoverability.



Above is my relatively standard search. The first results are forum posts -- not good. This same upgrade experience has happened literally millions of times, why isn't there a centralized hub for this specific experience? It took me (an experienced Googler) about 20 minutes to find out exactly what I had to do to upgrade my phone, simply because the discoverability of this process was so poor. If this was a process to be replicated by a handful of people, a forum post is fine. But this process has happened millions of times and the best avenue for discoverability is a forum post with contradicting answers? Poor discoverability.

Eventually, I 'discovered' that I needed a new sim card for the iPhone 5s. In fact, I had already known that the 5s required a different sim card (constraints: a later discussion). But the phone I

was given had a functioning AT&T sim card in it already. Did I need another sim card? Another 10 minutes of ‘discovering’ later, and a 15 minute phone call later, I ‘discover’ that I need to head into the AT&T store to get a new sim card.

Two weeks later, I finally muster the time and frustration (with the aging iPhone) to drag myself to the AT&T store. The service representative pops both phones’ sim cards out, retrieves a new one, configures the card, and pops it in the 5s.

Now he has to check if it works. The phone takes 5 minutes to ping the towers. The phone takes another 5 minutes to appear to work. 5 minutes later, it sends a phone call. The feedback at this stage is just really poor. Neither I, nor the representative knew if any actions were successful in 5 minute increments. The feedback was not only slow, but cryptic.

This experience wasn’t really a big deal. It was annoying, but it’s infrequent enough for me (and everyone else) to put up with it. That often seems to be the case in instances of poor design. The design could be poor, but it won’t change because there isn’t enough incentive to do so.

Understanding: 4/5

Effort: 3/5

Week 3:

Week 3’s concepts are about knowledge in the head and in the world. I couldn’t help but draw similarities to video game design. Video games are all about world immersion -- combining knowledge in the head and the virtual world. Video game designers face the same problems that industrial designers face, but with extra challenges. The environments, objects, mechanics, and

interactions that they design must carefully balance usability and complexity to develop an appropriately fun and satisfying gameplay experience. Fun has a formula.

There is one game franchise that really sticks out in this regard, *Uncharted*. Revered game developer Naughty Dog pioneered modern techniques in covertly guiding players through an environment without breaking immersion. They use concepts found in chapter 3 to do so.

It's important for the player to know where to go in the environment, if the player can't figure it out, he/she gets frustrated. Conversely the player is told exactly where to go, she/he gets bored. Uncharted: Drake's Fortune (2007) was one of the first games to solve this problem.

Here's an environment from Uncharted 3 (not player's perspective)

The player controls a 3rd-person character facing the 'camera' at the start arrow. The player needs to make her way towards to 'end' box, utilizing the physical capabilities of a parkour-expert-action-hero. How would you get there without going in the water?

Below is how this section is typically negotiated. How close were you?

(The character hangs off the yellow siding -- like a climber)

In the Uncharted games, brighter/lighter hues tend to the intended path through the environment. This knowledge starts off in the world. It's not really even conscious. The eye is drawn to the brightly-colored features in the environment. The player perceives what is colored, and analyzes its potential affordances. The eye is drawn to, say, the yellow ladder. The player recognizes the ladder. The player declaratively knows it affords climbing, they know it will take them towards the 'end', and they make their way over towards it. 'Artificially natural' constraints on the environment confirm the ladder is likely the only path towards the 'end' -- there's no other path.

The user's knowledge of how to negotiate the environment is procedural -- deduced from deliberately placed environmental clues. As the player plays through Uncharted games, their knowledge of how to navigate certain environments is deliberate: yellow/bright coloring means path. The deliberate coloring isn't jarring, it looks like a normal environment. But it functions by signifying environmental affordances. Ultimately, the designer's guidance is provided without breaking immersion, a difficult task.

A side note on natural mapping: video game controls are a combination of natural and arbitrary mapping. For example, to shoot a gun in Uncharted, you pull the controller's right trigger, like you would a gun, an example of natural mapping. Movement is mapped to the analog sticks, a somewhat natural mapping, but with no real world corollary. Other functions are arbitrarily mapped. They need to be learned by the player.

- Understanding: 5/5

- Effort: 4/5

Week 2:

I'm somewhat enamored with web apps. 'Enamored' may be a strong word, but I find their position in the tech-sphere fascinating. I've been thinking about the concepts from week 2 in relation to web apps, so I thought I'd share my thoughts here.

People fought for an open web with unified conventions and succeeded. But it didn't really matter as much as everyone thought it would. Before iOS, many people expected web apps to be dominant because of its platform agnosticism. One product would exist everywhere. But iOS and Android threw a wrench in that idea, and web apps were sort of left out to dry on mobile. I think part of the reason web apps were replaced with native apps on mobile comes down to the gulfs of evaluation and execution.

Web apps tend to be less responsive than native apps, and struggle with the gulf of evaluation because of it. The key to successful navigation of the gulf of evaluation is fast and responsive feedback, something that web apps struggle with, especially on mobile. Oftentimes, I'll execute a function in a web app, and I'll wait a second for affirmative feedback of function execution. If I string a few function-executions together, I start to wonder if I'm even connected to the internet anymore. If I do this on my phone, I wait so long that I actually look up and talk to people. The mobile web app experience is typically more frustrating than the equivalent native mobile app experience.

The gulf of execution on a mobile web app and a native web app are usually pretty similar. The two tend to use similar interfaces to execute similar functions. Furthermore, unlike web apps, native apps allow for multi-touch (...force-touch, vibration, gyro, etc...) interactions with interfaces. This allows Native mobile apps don't have the constraints that mobile web apps do, hence the rise of native mobile apps.

Why is the native app experience so much better on mobile? It comes down to the gulfs of execution and evaluation. But why? Let's take a look at the three stages of processing.

1. Visceral: quick, subconscious judgments
    - the delayed feedback of web apps leads to anticipation anxiety (gulf of evaluation)
    - the crisp fluidity of native apps (speed, responsiveness, swiping, etc.) avoids anxiety and increases satisfaction (gulf of execution/evaluation)
2. Behavioral: learned skills
    - the 'clustered loading' of web apps (that makes interface rearrange to fit screen size after page is seemingly loaded -- leads to errors) doesn't allow learning and subconscious operation of device because the loading operation of web apps is inconsistent. (gulf of execution)
    - native mobile apps can follow mobile interface conventions better than mobile web apps can. Native apps take advantage of device affordances. Can swipe, vibrate, etc. (gulf of execution)

3. Reflective: conscious cognition
    - native mobile apps perform as expected more consistently. This increases reflective satisfaction. (gulf of execution)
    - the emotions at the reflective level stem from emotions generated from visceral and behavioral levels (anxiety, frustration, etc.)

Understanding:5/5

Effort: 4/5

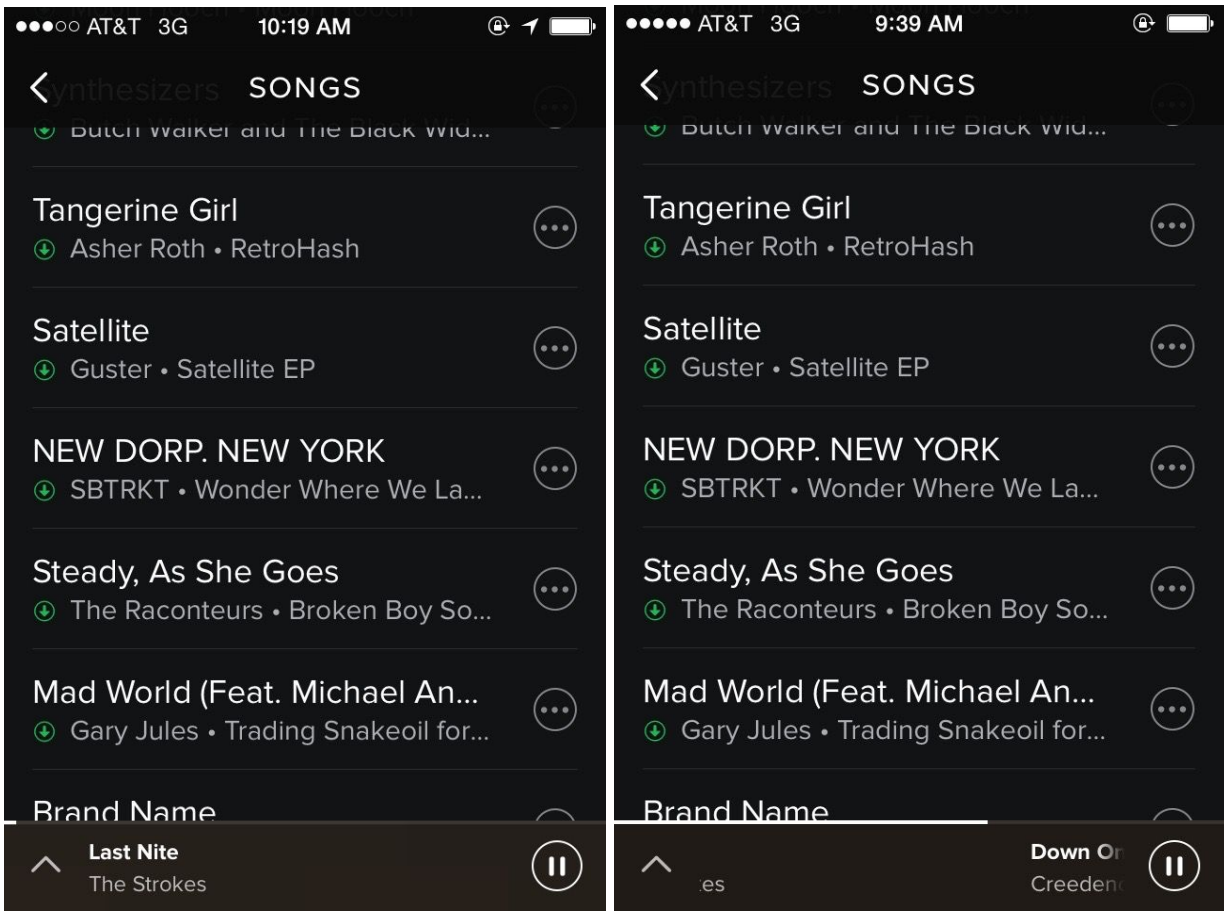
Week 1:

Week 1 focused on Psychopathology of Everyday Things. We were introduced to several concepts: affordances, signifiers, conceptual model, feedback, and system image, amongst

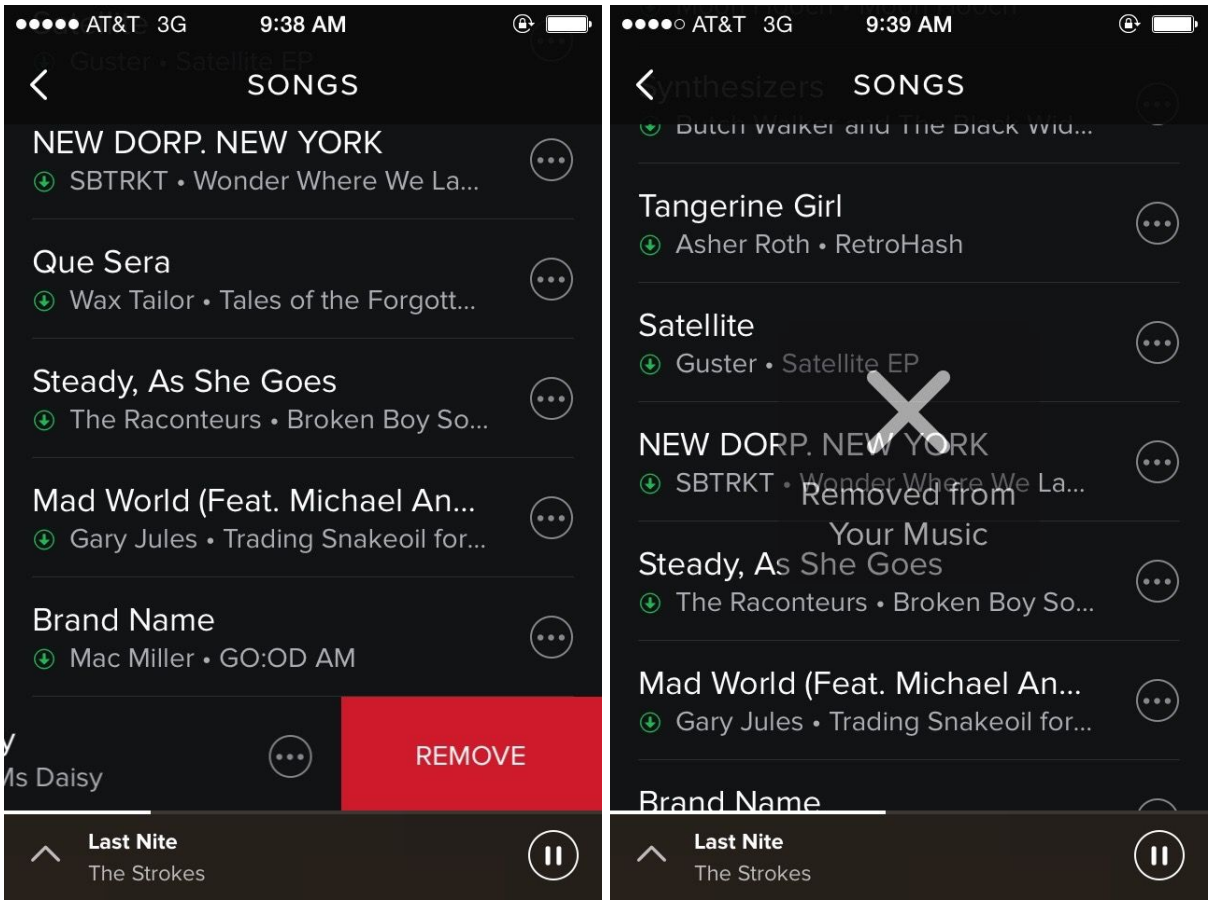
others. I've excerpted part of the first homework assignment, because I think it was the most significant thing we did that week.

Example of a bad design: Spotify Mobile App - 'Now Playing Banner'

> Proper Function: Swipe the 'Now Playing' banner on the bottom of the screen to the left to play the next song. The banner affords swiping to new/previous songs whilst perusing your music library. It's function is not well signified, though. There's no obvious indicator that it even exists. It doesn't have an obvious place in the app's conceptual model, either, because there are several ways to switch songs in the app.



Did you miss the swipe?



A missed swipe only a few millimeters above the tiny 'Now Playing' banner will delete a song from the playlist or library, without any indication of which song was deleted. The 'Now Playing' banner fails to account for user error, a core tenant of human-centered design. The design also lacks feedback as to which song was deleted. The app assumes deletion was intentional, and thus provides no feedback as to which song was removed.

This exercise was the first time I put design thoughts into language. Initially, I was not terribly precise with my language. I found it difficult to take my ideas from casual language to a more deliberate and argumentative form. I chose not to edit the original assignment, because I want to be able to look back and see how I improve in my expression of these ideas.

- Understanding: 5/5
- Effort: 2/5