

HDDC Schedule Redesign

BIG IDEA

Restructure the High Desert Dance Classic's online schedule for users attending the event **so that they can quickly identify key information** needed to participate in their activities.

The current design of the schedule is **cluttered with text that stresses the cognitive load for users** and causes frustration when they spend an excessive amount of time trying to find specific information for activities they want to participate in.

Most users are **over the age of 35** and have **limited experience with QR codes and mobile websites** which are needed to access the schedule online. With this limitation, we should reorganize the structure of information to simplify the user's journey and make it easier to follow. This will provide an overall better experience for the user and improve brand credibility.

PROCESS STRATEGY

LEAN APPROACH

This organization runs a non-profit event that only happens once a year, so taking the *Lean* approach to this project allows us to meet the end-user's needs while minimizing the amount of resources wasted on unsuccessful designs. We validate what works based on user and stakeholder feedback early on in the design process and iterate frequently in order to produce only the most optimal outcome for our users.

STAKEHOLDERS

PRIMARY USERS

- Millennials/GenX (The Well-Versed), ages 35+
 - Comfortable with technology
 - Wants to be able to locate information quickly and efficiently
 - PAIN POINTS:
 - Used to clearly structured information, may be frustrated with the current layout for a schedule (table)
 - May see the webpage as primitive compared to more modern websites
- Baby Boomers (The Retired), ages 60+
 - Limited experience with modern tech and no motivation to learn how to use it
 - May prefer less customization/filter features and more straightforwardness
 - May use a screen reader or something similar to help them get information from the website
 - PAIN POINTS:
 - Tends to get lost/overwhelmed when there is a large amount of text
 - May have a hard time initially getting to the website if they are unable to work a QR code
 - May be blind/partially blind or just prefers bigger text
- Ballroom and swing dancers
 - Busy schedule, especially for those competing and/or leading workshops
 - Wants to get the most out of their money, having their activities planned out so that they don't waste any time
 - PAIN POINTS:
 - Impatient and unwilling to take their time to go through information that isn't relevant to them

STAKEHOLDERS

- Web developers
 - Needs clear instruction on how to organize the information to make it accessible for primary users
 - Would like to know how complex the restructure will be
 - PAIN POINTS:
 - Concerned that there is no real benefit and does not want to waste time and resources on a restructure that won't stick with users
- UIX Designer
 - Wants to make the schedule more accessible for the mass user
 - Needs further research to determine the best course of action for redesign
 - PAIN POINTS:
 - Needs to figure out how to streamline information without fatiguing the user with too many actions
 - Needs to balance constraints set by other stakeholders with the needs of the user
- Event Organizer
 - Wants to promote the youth ballroom program that also runs at the same time as the main event
 - Would like to keep all information for both events accessible on the same page
 - PAIN POINTS:
 - Needs to be able to look at the entire schedule to ensure the event is running smoothly
 - Needs information of events to be as straightforward as possible so they do not waste time scrolling through
- Sponsors
 - Believes in the cause, wants to see the dance event thrive
 - Wants to know that their donation efforts are being put to good use in helping fund the event
 - PAIN POINTS:
 - May believe that there is poor management from the event organizer if the schedule is unorganized

AI INSIGHTS

By utilizing ChatGPT, I was able to simulate a [kick-off meeting](#) with other stakeholders using the following prompt:

"I have a rough idea for a UIX project. I have created a draft of my problem, and identified stakeholders. Now I want you to pretend you are my design team of 2 other designers, and 2 developers, reviewing my document in a kick-off meeting. Some of the stakeholders that I listed in my problem have also been invited, and are invited to participate. They all read the document, and prepare comments, which are shared in the kick-off meeting.

Replicate responses from these designers and stakeholders after reading my initial design document.

Also output the following:

- *5-10 possible **user pain points***
- *3-5 **unexpected stakeholders***

After sharing my document with the AI, I reviewed the transcript and identified the following:

Which AI suggestions are useful?

- The project goal should be further defined by the USER, and we assume the solution before the research
 - Right now we are limiting the possibility of solutions by defining the goal as "restructuring the schedule" when it should be about improving the findability, scan-ability, and navigation experience for the user
- We have conflict in user needs from different users. We should identify primary users (who we optimize for) and secondary users (who we consider, but not at the cost of the primary)

- With a live event, we need to be prepared to make last minute changes to the schedule if necessary. This means designing for easy editing in addition to user needs
- Refined user pain points
- Suggesting a redesign that is fast and predictable (faster load times AND lighter cognitive load) over something with high customizability (fancy animation and filter systems)
- Suggested “layered access” to schedule to accommodate both primary and secondary users
 - Simplified view for attendees
 - Full view for event organizer/volunteers
- Introduced potential problems if we split the schedule into different views
- Suggested a starting point for the project and next steps

Which are weak or irrelevant?

- The end-user (dancer) simulation did not give any insights to the topics discussed, just had an attitude about wanting specific information :(
- When asked for a push-back on a suggestion, one designer brought up the fact that a single schedule could effectively serve all our users even though it was established earlier that there were conflicts in user needs
- In the same push-back, the second designer brought up the fact that users might get information wrong between the different views even though both views would contain the same information
- Event organizer was worried that separating the schedule into role-based views might cause volunteers to refer to the wrong schedule and users might get sent the wrong link
 - If the information is universal across both schedules why would volunteers get confused?
- When debating on whether split views of the schedule was beneficial to explore, one of the main cons it kept suggesting was mismatching information when it was previously discussed that the views would have consistent data
- Suggested that the effort put into the project was not enough for a non-profit once-a-year event—assuming everyone here is getting paid and not volunteering for it?

What does this tell you about AI's strengths/limits?

- Without being able to show the AI the current product, it is hard to explain what I am looking at when I bring up problems and potential solutions
- Kept bringing up suggestions that might broaden the scope of the project
- Could not generate solid push back that made me completely rethink my decisions/suggestions
- Good at refining ideas
- Needs explicit direction in the prompt to determine an outcome, cannot understand intention behind question—I think when I ask if it “can” it will no matter what

PROJECT APPROACH

GOAL

Improve the scannability, findability, and navigation of the High Desert Dance Classic's online schedule for users attending the event **so that they can quickly identify key information** needed to participate in their activities.

OBJECTIVES

1. Reduce the cognitive load on users when they are looking for schedule information.
2. Improve content scannability and align the information architecture with user needs.

STRATEGY

This organization runs a non-profit event that only happens once a year, so taking the *Lean* approach to this project allows us to meet the end-user's needs while minimizing the amount of resources wasted on unsuccessful designs. We validate what works based on user and

stakeholder feedback early on in the design process and iterate frequently in order to produce only the most optimal outcome for our users.

RESEARCH METHODS

- Observational research
 - To gather initial feedback on where the problem lies in the use of the schedule and sharpen the scope of the project
- Survey
 - To establish what information is most valuable to users when looking at the schedule
- Card Sort
 - To define the information architecture within the schedule in a way that makes sense to the user
- A/B testing
 - After creating initial designs based off of previous research to determine the design path to take, iterate frequently based on continuous feedback
- Usability testing
 - To gather qualitative feedback on redesigns, observe how the redesigns affect the user journey, and identify any usability issues that we may need to iterate on

Kick-Off Meeting Sim.

Initial Prompt:

"I have a rough idea for a UIX project. I have created a draft of my problem, and identified stakeholders. Now I want you to pretend you are my design team of 2 other designers, and 2 developers, reviewing my document in a kick-off meeting. Some of the stakeholders that I listed in my problem have also been invited, and are invited to participate. They all read the document, and prepare comments, which are shared in the kick-off meeting.

Replicate responses from these designers and stakeholders after reading my initial design document.

Also output the following:

- *5-10 possible **user pain points***
- *3-5 **unexpected stakeholders***

INITIAL KICK-OFF RESPONSE

Kickoff Meeting — Team & Stakeholder Reactions

Designer 1 (Product Designer – Information Architecture)

"The core idea is strong: reducing cognitive load in a schedule-heavy experience. That's a real, tangible problem.

But right now, the solution is *implicitly* defined as 'restructure the schedule'—I'd like us to stay problem-focused a bit longer.

A few things I want to challenge:

- Are we solving *findability*, *readability*, or *navigation*? These are different problems.
- 'Cluttered with text' — is that due to formatting, hierarchy, or content volume?

Also, I think we should explore multiple paradigms beyond a table:

- Timeline view
- Personalized agenda
- “Now / Next” view for live usage

Right now, we’re assuming restructuring = improvement, but we haven’t explored alternatives yet.”



Designer 2 (UX Researcher)

“There’s a really important tension here that I love:

- Users 60+ want simplicity and low interaction
- Dancers want speed and efficiency
- Organizers want full visibility

Those needs *conflict*.

I’d strongly recommend we define:

- Primary persona (who are we optimizing for first?)
- Secondary personas (who we support, but not at the cost of the primary)

Also, a big callout:

‘Limited experience with QR codes’

We should validate this. That assumption could heavily shape the solution. If it’s wrong, we might over-design for a constraint that doesn’t exist.

I’d propose quick validation:

- 5–7 user interviews before design direction”



Developer 1 (Frontend Engineer)

“From a build perspective, I see three possible levels of effort:

1. **Low effort:** Improve typography + spacing + hierarchy (same structure)
2. **Medium:** Add filters/search/sorting

3. **High:** Fully dynamic schedule (personalized, live updates)

We need alignment on which direction we're heading.

Also:

- Will users be logged in? Or is everything anonymous?
- Are we designing mobile-first? (Feels like yes, but not explicitly stated)
- Is internet connectivity reliable at the venue?

Because if not, we may need:

- Offline-friendly design
 - Extremely fast load times"
-



Developer 2 (Backend / Systems)

"The organizer requirement is interesting:
'See the entire schedule at once'

That directly conflicts with simplifying for users.

We might need:

- Different *views* of the same data (not one universal UI)

Also:

- Who maintains the schedule?
- How often does it change during the event?

If updates are frequent, we need:

- Real-time or near-real-time updates
 - Clear version control (users hate outdated schedules)"
-



Stakeholder Feedback



Product / Business Stakeholder

“The problem is clear, but I’m missing the *impact*.”

What happens if we solve this well?

- Fewer complaints?
- More workshop attendance?
- Better sponsor perception?

Also, since this is once a year:

→ Failure is expensive

→ We should consider *testing before the event* (maybe with last year’s attendees)”

Marketing / Brand Stakeholder

“You mentioned ‘brand credibility’—that’s important.

Right now, the schedule might be one of the *first touchpoints* users have at the event.

If it feels outdated or confusing:

→ It reflects poorly on the entire event

I’d love to see:

- Visual polish
- Clear branding
- A sense of professionalism

This is not just UX—it’s perception.”

Event Organizer

“I like the idea, but I’m concerned about losing visibility.

I need:

- Quick scanning of the *entire event*
- Confidence nothing overlaps incorrectly

Also:

‘Keep youth program visible’

That's non-negotiable—we need both audiences served.”

Sponsor Representative

“From our perspective, user confusion = poor organization.

If attendees feel frustrated:

- It reflects on the event's professionalism
- Which reflects on us as sponsors

If this redesign improves clarity, it directly supports sponsor confidence.”

Dancer (End User Voice)

“During the event, I don't want to *explore*.

I want to:

- Know where I need to be
- Know when
- Get there fast

If I have to scroll, zoom, or think too much—I'm already annoyed.”

Key Frictions Identified

- One interface trying to serve *too many needs*
 - Assumptions about tech literacy not yet validated
 - “Restructure” may be too narrow as a solution direction
 - Strong need for **real-time, fast, low-friction access**
 - Organizer vs attendee needs are in tension
-

5–10 Refined User Pain Points

1. Difficulty scanning dense schedule tables quickly
 2. Inability to filter only relevant activities (e.g., competitions vs workshops)
 3. Getting lost in long vertical scrolls of information
 4. Struggling to read small or poorly structured text (especially older users)
 5. Friction accessing the schedule initially (QR codes, links, navigation)
 6. No quick “what’s happening now” visibility
 7. Hard to build a personal plan for the day
 8. Cognitive overload from too much simultaneous information
 9. Fear of missing important events due to poor organization
 10. Lack of trust in whether the schedule is up-to-date
-

3–5 Unexpected Stakeholders (Specific to YOUR Project)

1. **Venue Staff**
→ Get asked for directions constantly when schedules are confusing
 2. **Volunteers**
→ Often help attendees navigate—bad UX increases their workload
 3. **Accessibility Users (Screen Reader Users)**
→ Especially relevant given your 60+ demographic
 4. **Photographers / Videographers**
→ Need to track key moments quickly across the schedule
 5. **Judges / Instructors**
→ Have tight schedules and zero tolerance for confusion
-

What Your Team Would Recommend Next

If this were a real kickoff, we’d push you toward:

- Define **primary persona**
- Validate **QR code assumption**
- Map **user journeys (before, during, after event)**
- Explore **multiple schedule formats (not just restructure)**
- Decide **scope level (light fix vs full redesign)**

If you want, I can take this further and:

- Turn your idea into a polished case study
- Create a UX flow or wireframe directions
- Or simulate a design critique of your *next iteration*

Just say the word.

ADDRESSING INITIAL SCOPE OF PROJECT

Designer 1, do you think we would be able to address findability, readability, AND navigation issues by redesigning the layout of the schedule?



Designer 1 (Product Designer – Information Architecture)

“Short answer: **yes—but only partially if we rely on layout alone.**”

Let me break that down a bit more practically.



Where layout *does* help a lot

A strong layout redesign can significantly improve:

1. Readability

Absolutely—this is the most directly impacted.

- Better typography (size, spacing, contrast)
- Clear visual hierarchy (headings, grouping)
- Chunking information into digestible sections



You can reduce cognitive load *immediately* here.

2. Findability (to a degree)

Layout helps by:

- Grouping related items (e.g., workshops vs competitions)
- Using visual cues (color, icons, labels)
- Creating predictable patterns

But here's the catch:

Layout helps users *scan faster*, but doesn't always help them *jump directly* to what they want.

Where layout alone falls short

3. Navigation (the weakest if only layout changes)

If users need to:

- Jump to a specific time
- Find only their events
- See “what’s happening now”

👉 Layout alone won't solve that.

You'll likely need:

- Filters or categories
 - Anchor links or quick navigation
 - Possibly a “Now / Next” or “My Schedule” view
-

The real takeaway

“Layout is the foundation—but not the full solution.”

Think of it like this:

- **Layout = reduces friction**
- **Navigation tools = reduce effort**
- **Content structure = reduces confusion**

You need all three working together.

If I were guiding the team, I'd suggest:

Start with layout improvements **first** (low cost, high impact), then layer in light navigation enhancements:

- Phase 1: Clean, scannable layout
 - Phase 2: Add simple navigation aids (jump links, sections)
 - Phase 3 (if needed): Filters or personalization
-

One pushback for you

Right now, your idea is framed as:

“Restructure the schedule”

I'd encourage you to reframe it slightly to something like:

“Improve how users **scan, find, and navigate** schedule information”

That opens the door to better solutions—not just prettier layouts.

REDUCING QR CODE FRICTION

Designer 2, we validated the assumption that our primary users have limited experience with QR codes, but the QR code is a non-negotiable access point for the schedule. What do you suggest we do to reduce that friction?

Designer 2 (UX Researcher)

“Great—this is exactly the kind of constraint that leads to good design decisions. If the QR code is non-negotiable, then we don't remove it—we *support it*.”

Right now, the friction isn't the QR code itself—it's everything *around* it:

- Understanding what it is
- Knowing how to use it
- Trusting that it's worth doing

🎯 Strategy: Reduce friction *before, during, and after* the scan

🟡 1. Before the Scan — Set Users Up for Success

“Most failures happen *before* users even try.”

We need to remove uncertainty and intimidation.

What I’d recommend:

- Add a **clear instruction line** near the QR code
→ “Open your camera and point it at this code”
- Include a **visual cue** (simple diagram or icon)
- Add a **plain-language fallback**
→ “Or go to: HDdanceclassic.com/schedule”

👉 This is especially important for 60+ users who may hesitate if unsure.

🟢 2. During the Scan — Reduce Technical Friction

“We assume scanning is instant—but for many users, it’s not.”

Opportunities:

- Ensure the QR code is:
 - Large enough
 - High contrast
 - Placed at eye level (not awkward angles)
- Place it in **multiple locations**:
 - Entrance
 - Registration desk
 - Near event rooms

👉 Redundancy reduces stress and crowding.

🟢 3. After the Scan — Make It Feel Worth It (Critical)

“If the page they land on is confusing, they’ll blame the QR code.”

This is where your redesign becomes *essential*.

We should ensure:

- Fast load time (no lag = no doubt)
- Immediate clarity:
 - “You’re viewing the event schedule”
- Large, readable content (especially for older users)

👉 The first 5 seconds after scanning determines whether they trust the system.

🟣 4. Add a Human Safety Net

“For this audience, human support is part of the UX.”

We should *expect* some users to struggle.

Add:

- Volunteers or signage saying:
 - “Need help accessing the schedule? Ask us!”
- Possibly a **printed mini-schedule** for edge cases

👉 This reduces anxiety and prevents total failure.

🔴 5. Optional but High Impact: Reduce Repeat Friction

“Scanning once is okay. Scanning 5 times is frustrating.”

After users access the schedule:

- Prompt them to:
 - Bookmark the page
 - Add to home screen (simple instruction)

👉 This turns a painful action into a one-time cost.

 **Big Insight**

“We’re not designing a QR code experience—we’re designing a *confidence journey*.”

Users are asking themselves:

- “Will this work?”
- “Do I know how to do this?”
- “Is this worth the effort?”

Your job is to make all three answers:

“Yes.”

One Challenge Back to You

Right now, your problem statement focuses heavily on the *schedule design*.

But based on this constraint, I’d expand it slightly:

You’re also designing the **entry experience into the schedule**

That includes:

- Signage
- Instructions
- First impression after scanning

ASKING ABOUT CONTENT MANAGEMENT

Developer 1, we have confirmed that users will navigate anonymously, not logged into an account. How do you suggest we manage the content on the schedule to make it load faster for users?

Developer 1 (Frontend Engineer)

“Good call confirming anonymous access—that simplifies a lot. Now the goal is:
make the schedule feel instant, even on slow or spotty connections.”

Since we *can’t* rely on user accounts or saved preferences, performance has to come from how we structure and deliver the content.

1. Keep the Page Lightweight (Biggest Win)

“Most slow experiences come from shipping too much stuff upfront.”

What I'd push for:

- Avoid heavy frameworks if possible (or keep them minimal)
- Use **simple HTML + CSS-first structure** where we can
- Limit large images, fonts, and unnecessary scripts

👉 For a schedule, content > visuals. Speed matters more than flash.

2. Structure Data for Fast Rendering

“Don't make the browser *work hard to understand* the schedule.”

Instead of one giant block of content:

- Break schedule into **logical chunks**:
 - By day
 - By room
 - By event type

Then:

- Render only what's needed first (e.g., current day)

👉 This reduces initial load time and improves perceived speed.

3. Prioritize “Above-the-Fold” Content

“Users should see something useful in under a second.”

Load first:

- “Happening Now”
- Next upcoming events
- Key navigation options

Delay loading:

- Full schedule
- Less relevant sections

👉 This creates the feeling of instant usability.

4. Design for Bad Connectivity (Very Real at Events)

“Assume the Wi-Fi will be bad. Because it will be.”

We should:

- Minimize API calls (ideally **1 request** for schedule data)
- Cache the schedule in the browser after first load
- Avoid constant refreshes unless necessary

👉 Once it loads once, it should feel *snappy forever*.

5. Use Caching Aggressively

“Anonymous users don’t mean stateless performance.”

We can still:

- Cache static assets (CSS, JS)
- Cache schedule data locally (browser storage)

So when users come back:

→ It loads almost instantly

6. Handle Updates Smartly

“Schedules change—but we don’t want to reload everything.”

Options:

- Lightweight “Last updated” timestamp
- Manual refresh button instead of auto-refresh

- Optional background refresh (non-blocking)

👉 Avoid anything that interrupts the user mid-task.

7. Avoid Overbuilding Features

“Every feature has a performance cost.”

For example:

- Heavy filtering systems
- Complex animations
- Real-time updates everywhere

👉 If it's not critical to the user goal (find info fast), it's a liability.

Tradeoff I Want You to Be Aware Of

“There's a tension between *flexibility* and *speed*.”

- Highly dynamic, interactive schedule → slower, more complex
- Simple, structured, fast schedule → less flexible, but reliable

Given your users (35+, 60+, event context), I'd strongly lean toward:

Simple, fast, predictable over clever

One Idea I'd Love to Explore With You

Since users are anonymous:

👉 We could use **device-level persistence** (not accounts):

- “Recently viewed events”
- “Quick jump back to last section”

No login required, but still feels personalized.

My Pushback

Right now your doc frames this as a *design problem*, but:

“Performance *is part of the UX here*, not just engineering.”

If the page takes 5 seconds to load:

→ It doesn’t matter how good your layout is

PROPOSING SOLUTION TO EVENT ORGANIZER

Event organizer, we are trying to improve the findability of information on the schedule. While it may be beneficial for you to have everything listed on one page, it is difficult for our users to navigate. We are thinking about breaking the schedule down into categories so that our users can jump to specific activities that they are interested in. What do you think of this?

Event Organizer

“I understand why you want to do that—and honestly, I agree with the *problem*. People *do* get lost in the current schedule.”

But I want to be very clear about my concern:

“If you split things too much, I lose my ability to **see the whole event at once**—and that’s critical for me.”

My Initial Reaction

I’m **open to categorization**, but not if it replaces the full schedule.

Here’s why:

- I need to quickly check for **time conflicts across categories**
- I need to ensure **rooms and instructors aren’t double-booked**
- During the event, I’m scanning constantly—not searching

👉 If everything is separated, my job actually gets harder.

Where I Think We Can Align

“I don’t think this is an either/or—I think we need both.”

What I’d support:

For Attendees:

- Categories like:
 - Workshops
 - Competitions
 - Social dances
- Quick jump links or filters
- A simplified view that removes irrelevant info

👉 This makes total sense for users trying to *find their thing quickly*.

For Me (and staff):

- A **full master schedule view**
- Everything visible in one place
- Easy to scan across time + rooms

👉 This is essential for operations.

Suggestion I Like

“Instead of breaking the schedule apart, can we *layer access* to it?”

For example:

- Default view: **Simplified / categorized (user-friendly)**
- Secondary option: **“View Full Schedule”**

That way:

- Attendees aren’t overwhelmed
- Staff still has full visibility

One Risk I Want to Avoid

“Users missing important events because they stayed inside one category.”

Example:

- Someone only looks at “Workshops”
- Misses a competition or announcement they care about

👉 We should make sure:

- Key events are still visible globally
- Or there’s a “Highlights” / “Don’t miss this” section

What Success Looks Like to Me

- Attendees stop asking basic schedule questions
- I can still monitor the entire event without friction
- Nothing gets “hidden” in the system

My Bottom Line

“I support categorization—as long as it doesn’t replace the full picture.”

If you can give:

- **Users** → **simplicity and speed**
- **Me** → **control and visibility**

Then I think this is a strong direction.

ALIGNMENT SUGGESTION

I suggest that we separate the attendee schedule and the master schedule for people running the event. Both will have the same content, but the attendee schedule will be what is on the event website (what we will focus on in the project), and the master schedule will be on a separate platform (maybe emailed to the organizer and volunteers before the event). This way, we can focus on the primary users while still providing support for our secondary users.

Event Organizer

“I actually think this is a strong direction—with a couple of important conditions.”

Separating the **attendee schedule** and the **master schedule** could solve a lot of the tension we’ve been talking about. It lets each group get what they need without compromising the other.

What I Like About This

1. Clear Focus on Attendees

“If the website is optimized just for attendees, that’s a big win.”

- Simpler layout
- Easier navigation
- Less overwhelming

 This directly addresses the complaints we’ve been getting.

2. Operational Flexibility for Staff

“A separate master schedule gives us breathing room.”

- We can keep full detail
- We don’t have to worry about “design constraints”
- It can include extra info (notes, buffers, internal details)

👉 That's actually *better* than forcing everything into one system.

⚠️ My Concerns (Important)

1. Risk of Version Mismatch

"We cannot have two schedules that say different things."

This is my biggest concern.

If:

- The website says one time
- The master schedule says another

👉 That creates chaos instantly.

What I'd want:

- One **source of truth**
 - Both versions generated from the same data
-

2. Accessibility During the Event

"Email alone might not be enough."

During the event:

- I'm moving around constantly
- Volunteers are too

If the master schedule is:

- Buried in email
- Hard to open quickly

👉 That becomes a problem.

I'd suggest instead:

- A simple internal link (mobile-friendly)
- Or a downloadable PDF that's easy to access quickly

3. Volunteers Need Speed Too

“Volunteers are closer to attendees than to me.”

They often:

- Answer questions
- Help people navigate

So they might actually benefit from:

- **Both views**, depending on the situation
-



Adjustment I'd Recommend

“Don't think of it as two separate schedules—think of it as two *views* of the same system.”

- **Attendee View (Website)**
 - Clean, filtered, simplified
 - Designed for quick lookup
- **Staff View (Internal)**
 - Full schedule
 - More detail
 - Possibly different layout (table/grid)

👉 Same data underneath, different presentation.



Why This Works

- Reduces cognitive load for attendees ✓
 - Preserves control for organizers ✓
 - Avoids design compromises ✓
-



Final Take

“I support this direction—as long as we maintain a single source of truth and make the staff version easy to access during the event.”

If you can guarantee that, I think this is one of the cleanest solutions we’ve discussed so far.

👁️ Quick Nudge Back to You

You’re making a subtle but important shift here:

From “one interface for everyone” → to “role-based experiences”

That’s a strong product thinking move.

If you want, the team can now:

- Pressure-test this idea (designer + dev perspectives)
- Or help you define what exactly goes into each version (attendee vs master)

TEAM RESPONSE TO ROLE-BASED VIEW

What does everyone else think about this suggestion?

Team Reactions to Your Proposal

(Separate attendee schedule + master schedule)

Designer 1 (Product Designer – IA)

“I’m strongly in favor of this—this resolves a core tension we identified earlier.”

From a design perspective, this is a **big unlock**:

- We’re no longer trying to force one interface to serve conflicting needs
- We can fully optimize for **clarity and hierarchy** in the attendee view
- We avoid overloading the UI with edge-case requirements

👉 This lets us design with *confidence*, not compromise.

One caution:

“We need to be very intentional about what gets *removed* from the attendee view.”

If we oversimplify:

- Users might lose important context (e.g., room info, timing overlaps)

So:

- This isn't just *less information*
 - It's **better structured information**
-



Designer 2 (UX Researcher)

“Conceptually, this is the right move—but I want to validate one thing before we commit fully.”

I agree with:

- Role-based views
- Prioritizing primary users

But I'd want to test:

“Do attendees ever need ‘full schedule awareness’?”

For example:

- Planning their day holistically
- Comparing overlapping events

👉 If yes, the attendee version still needs:

- A “browse everything” option
- Or a lightweight full view

Also:

This gives us a great research opportunity:

- Test **simplified vs full schedule** usability
- Measure time-to-find information



Developer 1 (Frontend Engineer)

“From an implementation standpoint, this is actually cleaner than a single overloaded interface.”

Why this works well technically:

- One data source ✓
- Two presentation layers ✓
- Less conditional logic inside one UI ✗ (which is good)

👉 That means:

- Faster development
- Easier maintenance

My recommendation:

“Build one structured dataset, then render two different UIs from it.”

Also:

- Attendee version → optimized for performance
- Staff version → can be slightly heavier, that’s fine



Developer 2 (Backend / Systems)

“I like this—as long as we enforce a strict single source of truth.”

Echoing the organizer:

“Two schedules = high risk *unless* they’re generated from the same data.”

What I’d propose:

- Central schedule database (even something simple like a structured JSON or CMS)
- Both views pull from it

Bonus opportunity:

- We can include **internal-only fields**:
 - Notes

- Setup times
- Staff instructions

👉 These show up only in the master view, not the attendee version.



Product / Business Stakeholder

“This is a smart prioritization decision.”

You’re effectively saying:

- Optimize for the majority (attendees)
- Still support operational needs

That’s good product thinking.

What I’d want next:

- Clear definition of success metrics:
 - ↓ Time to find an event
 - ↓ User confusion/questions
 - ↑ Workshop attendance
-



Marketing / Brand Stakeholder

“I’m very supportive—this helps us control the attendee experience.”

Right now:

- The schedule *is* part of the brand

A simplified attendee view means:

- More polished experience
- Less perceived chaos

👉 That directly improves credibility.



Dancer (End User Voice)

“As long as I get what I need faster, I don’t care how you structure it behind the scenes.”

What matters to me:

- I open it → I find my event quickly
- I don’t have to think

One request:

“Don’t hide things too much. Sometimes I want to explore.”



Overall Team Consensus



Strong alignment: This is the right direction

But with 3 critical conditions:



Non-Negotiables

1. Single Source of Truth

- No duplicate data
 - No manual syncing
 - Both views always match
-

2. Attendee View Still Needs Flexibility

- Not *too* simplified
 - Include:
 - Browse all option
 - Clear navigation
-

3. Staff Access Must Be Fast

- Not buried in email
 - Mobile-friendly
 - Easy to open during the event
-



What You Just Did (Important)

You shifted the project from:

“Fixing a layout problem”

To:

“Designing role-based experiences with shared data”

That’s a **major UX maturity step**—this is how real-world systems are designed.



Suggested Next Step (As a Team)

We’d now want to define:

1. Attendee View

- What’s included?
- What’s removed?
- Default state?

2. Master View

- What extra info exists?
- What layout works best (table likely)?

3. Shared Data Structure

- Fields (time, location, category, etc.)
- Internal vs public fields

PUSH BACK ON SUGGESTION

What are some opposing opinions on separating role-based schedules?

Great question—this is exactly the kind of thinking that strengthens your case study. Here's how people in the room would *push back* on your idea.

Opposing Opinions on Separating Role-Based Schedules

Designer 1 (Product Designer – IA)

“I see the benefits, but we may be introducing **system complexity to solve a UI problem.**”

Right now, the core issue is:

- Poor hierarchy
- High cognitive load

Not necessarily:

- Wrong audience segmentation

Concern:

“Are we splitting the experience too early instead of fixing the structure?”

It's possible that:

- A well-designed single schedule (with progressive disclosure)
- Could serve both groups effectively

👉 We might be solving the symptom, not the root problem.



Designer 2 (UX Researcher)

"I'm cautious about creating **inconsistent mental models**."

If:

- Attendees see one version
- Staff sees another

Then:

- Conversations get harder
- Instructions get confusing

Example:

"Go to the 2PM workshop in Ballroom B"
→ "I don't see that on my screen"

👉 Even small differences can create friction.



Developer 1 (Frontend Engineer)

"This doubles the surface area we have to maintain."

Even with shared data:

- Two UIs = two code paths
- More testing
- More edge cases

Concern:

"For a once-a-year event, is that overhead justified?"

If something breaks:

- We now have to debug *two* experiences

👉 There's a real cost to this decision.



Developer 2 (Backend / Systems)

“Data consistency becomes a critical risk.”

Even if we *intend* one source of truth:

- Different views can introduce:
 - Formatting differences
 - Missing fields
 - Timing discrepancies

Worst-case scenario:

Staff and attendees are looking at *slightly different realities*

👉 That’s operationally dangerous during a live event.



Event Organizer (Skeptical Mode)

“I’m worried about **fragmentation**.”

Right now:

- There’s one schedule
- Everyone refers to the same thing

With two:

- Volunteers might look at the wrong version
- Attendees might get sent the “wrong” link

👉 Simplicity has value, even if the UX isn’t perfect.



Product / Business Stakeholder

“This might be over-engineering for the scale of the problem.”

Key questions:

- Is the current issue severe enough to justify a system split?
- Could we get 80% of the benefit with a simpler fix?

👉 If this is a lean project, we should be careful about scope creep.

Marketing / Brand Stakeholder

“We risk inconsistency in brand experience.”

If:

- Attendee version = polished
- Staff version = utilitarian

That’s fine internally—but if:

- Staff accidentally shares their version
→ It could feel off-brand or confusing
-

Dancer (End User Voice – Unexpected Take)

“I don’t want to feel like I’m seeing a ‘limited version.’”

Even if simplified:

- Users might feel:
 - Something is hidden
 - They don’t have full access

👉 Transparency matters more than we think.

Core Risks of Splitting the Experience

1. Increased Complexity

- Two interfaces to design, build, and maintain

2. Risk of Mismatch

- Even small inconsistencies can cause confusion

3. Fragmented Communication

- People referencing different views

4. Over-Scoping a Lean Project

- More effort than necessary for a once-a-year event

5. Loss of Shared “Single Source” Experience

- Everyone no longer sees the same thing
-



The Underlying Debate

This really comes down to:

Option A: One Unified Experience

- Simpler system
- Harder design challenge

Option B: Role-Based Experiences

- Easier to optimize per user
 - More complex system
-



Balanced Team Perspective

“Separating roles is powerful—but it’s not free.”

The team would likely say:

👉 **Do it if:**

- User needs are clearly conflicting (they are)
- You can guarantee a single source of truth

- You keep both versions tightly aligned

👉 **Avoid it if:**

- It significantly increases build/maintenance cost
 - You can solve most issues with better hierarchy alone
-

What This Means for Your Project

If this were a real design review, we'd ask you:

“Can you prove that a single, well-designed schedule *cannot* solve this problem?”

If yes → your direction is justified

If no → you may be overcomplicating

JUSTIFYING RISKS

Let's acknowledge these risks. We want to address the needs of both our primary users (attendees) and secondary users (event organizer and volunteers), but we do not want to broaden the scope of our project. Can we justify spending extra resources on creating two separate views of the schedule?

Justification for Two Role-Based Views

Primary Users (Attendees) – High Impact

Problem we're solving:

- Attendees struggle with **cognitive overload** on the current schedule
- Many are **over 35**, less tech-savvy, and easily frustrated

- They need to **find events quickly** without scrolling endlessly or interpreting dense tables

Value of a separate attendee view:

- Clean, simplified interface optimized for scanning and orientation
- Reduces time-to-find by **significant measurable margins**
- Improves perception of the event (trust, professionalism, brand credibility)

This is the *highest-impact improvement*—directly tied to the user experience.

2 Secondary Users (Event Organizer & Volunteers) – Operational Necessity

Problem we're solving:

- Organizers need a **full overview** for planning and coordination
- Volunteers may need internal notes or logistical details not relevant to attendees

Value of a separate master view:

- Maintains operational control without cluttering the attendee experience
- Ensures volunteers and staff can **work efficiently during the event**
- Mitigates the risk of mistakes or confusion in real time

Without this, staff workflows could be hampered, indirectly affecting the attendee experience.

3 Risk Mitigation – Avoiding Negative Outcomes

Risk	Mitigation via Two Views
Users overwhelmed by too much info	Attendee view is simplified
Staff can't coordinate events	Master view has full detail
Mismatched info causing chaos	Single source of truth (same data powering both views)

Investing resources here prevents **both UX failures and operational failures**, making the extra cost **high leverage**.

4 Scope Management – Why It's Still Lean

- **Single data source** → reduces duplication
- **Separate views only** → no extra features, just different presentation
- **Focus on attendee view in this project** → master view can be generated automatically from the same system
- Limits scope creep because we are **not adding personalization, filtering, or accounts** yet

Extra resources are spent on *presentation*, not new features—this keeps it lean.

5 Measurable ROI

- Attendees: faster access, lower frustration, better event perception
- Organizers: fewer questions from users, smoother operations
- Volunteers: more confidence navigating logistics
- Event credibility and brand perception are preserved

These are **concrete, measurable outcomes** that justify the investment.

✓ Team Consensus

- **Designer 1:** Optimizes attendee UX without compromise → supports separate views
- **Designer 2:** Validates primary scenario → confirms it's testable and measurable
- **Developers:** Single source of truth keeps costs manageable; no redundant logic
- **Organizer:** Staff needs preserved; risk of operational failure minimized
- **Marketing / Brand:** Attendee experience polished → brand reinforced

All stakeholders agree: **two views are justified if done efficiently and the data source remains unified**.



Framing It in the Project

You can position this in your case study like this:

“While creating two views adds complexity, the decision is justified because it directly addresses the conflicting needs of attendees (primary users) and event staff (secondary users) without expanding the feature set beyond our lean approach. Both views are generated from the same source of truth, keeping maintenance minimal while maximizing user and operational impact.”

STARTING POINT FOR PROJECT

We need to narrow down our scope and focus on one part of this project. We have now established that it would be beneficial to have two different role-based views of the schedule. I want to focus on how we tackle the attendee view. With everything we discussed here, where should we start?

Designer 1 (Product Designer – IA)

“We should start with the **core user moment**, not the full system.”

Right now, it’s tempting to design the whole schedule—but most attendees are trying to answer a very simple question:

“What do I need right now?”

So I’d start with:

- The **first screen after scanning the QR code**
- And specifically: *what helps them orient instantly*

👉 That’s your highest-impact surface.

Designer 2 (UX Researcher)

“I’d narrow the scope to a single, testable use case.”

Let’s define a **primary scenario**:

“A 50-year-old attendee scans the QR code and wants to find their next workshop in under 10 seconds.”

If we can design for that moment, we can validate quickly.

What I'd do first:

- Sketch 2–3 variations of that *first interaction*
- Test:
 - Time to find info
 - Confusion points

👉 Keep it small, testable, and grounded in behavior.

 Developer 1 (Frontend Engineer)

“Start with the simplest version that could possibly work.”

Don't build:

- Filters
- Personalization
- Complex navigation

Start with:

- A fast-loading page
- Clear sections:
 - “Happening Now”
 - “Up Next”
 - “Full Schedule (by time)”

👉 If this works, we can layer complexity later.

 Developer 2 (Backend / Systems)

“Before UI, define the data structure for the attendee view.”

Even for a prototype, we need clarity on:

- Event name
- Time
- Location
- Category (workshop, competition, etc.)

Optional later:

- Instructor
- Skill level

👉 If the data is messy, the UI will be messy.

Event Organizer

“Make sure attendees can answer these 3 questions instantly:”

1. What’s happening right now?
2. What’s happening next?
3. Where do I need to go?

“If your design solves those, you’ve already improved the experience.”

Dancer (End User Voice)

“Don’t make me dig.”

When I open it, I want:

- Something obvious
- Something fast
- Something relevant

“If I have to think, it’s already too slow.”

Team Alignment — Where You Should Start

Start Here (Tight Scope)

Design the “First 10 Seconds” Experience

Focus only on:

- After QR scan
 - First screen
 - Immediate interaction
-

Define This Clearly

1. Primary User Goal

“Find my next relevant event quickly”

2. Core Features (ONLY these for now)

- **Happening Now**
- **Up Next**
- **Simple Full Schedule (time-based)**

👉 No filters yet

👉 No advanced features yet

3. Key Constraints

- Large, readable text
 - Minimal cognitive load
 - Fast load time
 - Works for low-tech users
-

Why This Scope Works

- Solves the **highest-friction moment**
 - Easy to prototype and test
 - Avoids overdesigning too early
 - Builds a strong foundation
-



What Not to Do Yet

Avoid jumping into:

- Complex categorization systems
- Personal schedules
- Multi-page navigation
- Advanced filtering

👉 Those are *phase 2* decisions.



If We Were in the Room, We'd Assign This

Your next deliverable:

- 2–3 **low-fidelity wireframes** of:
 - The first screen after QR scan

Each should explore a different approach:

- Option A: “Now / Next focus”
 - Option B: “Timeline preview”
 - Option C: “Category entry points”
-

And then:

- Quick usability test (even 3–5 people)
 - Measure:
 - Time to find an event
 - Where they hesitate
-



Final Take From the Team

“Don’t design the whole schedule. Design the moment that matters most.”

If you get that moment right, everything else becomes much easier to layer on.

HDDC 5 Layers

User Problem

HDDC participants have a hard time quickly finding specific schedule information

Strategy Layer

What research or planning defines the product direction?

Actions:

- Identify what information users need from the schedule
 - How familiar are our users with technology?
 - What information can be omitted or moved down in hierarchy?
 - What information needs to be prioritized?
 - How should information be grouped?
 - Mobile first!!!
-

Scope Layer

What features and content will the product include?

Actions:

- Most users attending HDDC are above the age of 35 and have limited experience with technology, keep features limited
 - More features = more room for confusion
 - Streamlined specific information for users instead of having all of the information on their screen at once
 - Clear titles, instructors, time, location, rank/skill level
 - Designing for WEB not PRINT, and must be accessible for older users using mobile website
 - Avoid large blocks of text
-

Structure Layer

How will the system behave and how is information organized?

Actions:

- Schedule organized by Activity TYPE (Comp, Workshop, etc.) > Day > Time > Specifics based on activity (Comp: Activity name, Rank, Prelims or Finals, Classes: Activity name, Skill level, Instructors)
 - Activity TYPE separated into tabs?
 - Accessible immediately from the QR code link, no need to navigate/tap through the website
 - Information clearly organized into categories to lessen the mental load for users
-

Skeleton Layer

How will the interface be arranged?

Actions:

- “HDDC Schedule 2026” header
 - No tables, use cards instead
 - Sub headers listed: COMPETITIONS, WORKSHOPS, DANCE FORCE
 - Per Event organizer: wants Dance Force program schedule in the same area of website as the main HDDC event
 - Short description of the type of activity that is listed
 - Potential ways users can navigate:
 - Tabs listed horizontally at the top, able to side swipe to view other categories OR tap on tabs
 - Headers listed vertically with the ability to tap and bring a drop down schedule per category
 - Category buttons that can be tapped and auto scroll down to whatever part of the schedule user needs
 - Information further separated by DAY
 - TIME listed first, then activity name, then sub information if applicable
 - For COMPETITIONS tab, registration opening and closing times are listed
-

Surface Layer

How will the final interface look and feel?

Actions:

- Clear visual hierarchy of information with most important information very prominent
 - Make use of icons + tags for visual cues
- Big simple font, avoid clutter
- Information is streamlined so users spend less time looking and more time dancing! Also fluid for screen readers