



Case Study 2: Instructor Assignment Sorter



Presented by Arjun Tyagi

A screenshot of the DFS IAS software interface showing the 'Unassigned Instructors' section. It displays a grid of instructor profiles with their names, schools, and availability. The interface includes a search bar, filter options, and a sidebar for managing assignments.

A screenshot of the DFS IAS software interface showing the 'Programs' section. It displays a grid of program icons (AppJam, LESTEM, Scratch, WebJam) and a single 'Engineering Inventors' program card. The interface includes a 'View By' dropdown, filter options, and a sidebar for managing sections and instructors.



Instructor Assignment Sorter

Introduction

Client: Dreams for Schools is a non-profit org that lead programs and initiatives contributing to educating youth about STEM/Comp Sci.

- Working with various school districts in SoCal to offer STEM courses
- Hires high school and university students as instructors
- In the process of scaling up their operations to the rest of California



Background: HCI capstone project, 2 quarters long

- Team of 4 designers with a background in development
- First quarter was focused on design exploration and refinement
- Second quarter was for development and updating designs

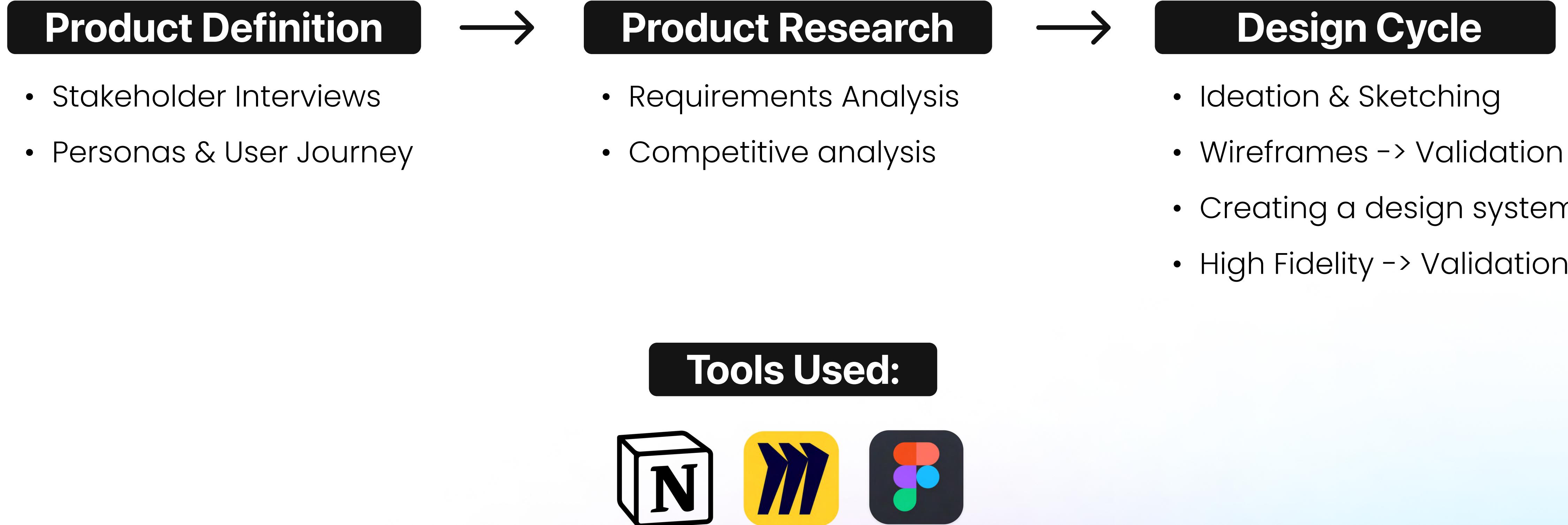


Project Overview

Problem Statement: As DFS works to scale up operations, they needed an integrated system to manage academic seasons, programs, partners, and instructors.

- DFS enrolls instructors from nearby high school and universities.
- Instructors provide location, availability, experience, and course preferences.
- Manually assign instructors based on their preferences, availability, and distance.
- Expecting significant increase in time/complexity as the org scales up across California

Our Design Process



Product Definition: Stakeholder Interviews

Goal: Understand the end to end process for managing instructors, partners, and programs.

- Background
- Overall responsibilities
- Day to day tasks
- What are their core needs?
- Pain points?
- How can we streamline their entire process?



Audrey Auyang



Nithin Jilla

Product Definition: Personas



Audrey Auyang

Programs Director at Dreams For Schools

Age 29 years old

Education B.S. in Business Management

Location Hacienda Heights, California

Tech Literacy High

Busy

Social

Organized

Biography

- UCI alum
- Passionate about social outreach, event planning, and teaching students about tech.
- Currently works as the programs director at Dreams For Schools
- Plans course offerings and assigns instructors to teach programs.

Frustrations

- Time consuming
- Cross referencing spreadsheets is prone to error
- Lots of data redundancy
- Do not have a one-stop program to view all data about programs, partners and instructors.
- Not scalable

Core Needs

- Streamlined process for keeping track of program, partner, and instructor data.
- Auto-assign based on preference, location, and availability.
- Intuitive way to manually assign instructors
- Should integrate well with existing onboarding process and seasons data

Tech Traits

Technical background



Desktop devices



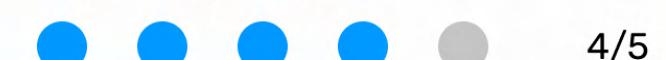
Mobile devices



Social media



App navigation



"

I'm currently using multiple spreadsheets to complete this process, so having something that is fool-proof and fast to use would be a game changer.

"

Motivations

- Eager to migrate to a less tedious process.
- Auto assign functionality will save huge amounts of time.
- Will allow her to focus her efforts towards org expansion/business goals

Product Definition: User Journey



Audrey Auyang

Scenario: Enrollment is open for winter 2020, and Audrey sees that they have received a new instructor application for James.

Expectations

- Add James as a new instructor for Winter 2020
- Assign James based on location, availability, preferences

1. Discovery	2. Consideration	3. Decide	4. Crosscheck	5. Assign
Activities <ol style="list-style-type: none">1. Receives James instructor application through email.2. Acquires information about James and reads about his educational background.	Activities <ol style="list-style-type: none">1. Reviews James application to understand if he is a good match.2. Considers availability, location, educational background and preferences.3. Considers instructor credibility.	Activities <ol style="list-style-type: none">1. Check program spreadsheets to see which classes still need an instructor.2. See if James' teaching preferences match available programs.3. Decide to accept application.	Activities <ol style="list-style-type: none">1. Crosscheck James' availability with programs still needing an instructor.2. Verify James' skills/preferences align with course subject.3. Verify James is close enough to teach the course.	Activities <ol style="list-style-type: none">1. Send out an acceptance email to James and ask him to accept his role.2. Upon acceptance, manually assign James to the program in the Excel spreadsheet.3. Update James availability.
Touchpoints <ul style="list-style-type: none">• Email• Google forms• Instructors spreadsheet (Google)	Touchpoints <ul style="list-style-type: none">• Email• Google forms• Instructors spreadsheet (Google)	Touchpoints <ul style="list-style-type: none">• Google forms• Instructors spreadsheet (Google)• Programs spreadsheet (Excel)	Touchpoints <ul style="list-style-type: none">• Google forms• Instructors spreadsheet (Google)• Programs spreadsheet (Excel)	Touchpoints <ul style="list-style-type: none">• Google forms• Instructors spreadsheet (Google)• Programs spreadsheet (Excel)
KPIs <ul style="list-style-type: none">• N/A	KPIs <ul style="list-style-type: none">• Time on task	KPIs <ul style="list-style-type: none">• Time on task• Instructor approval rate	KPIs <ul style="list-style-type: none">• Time on task• Human errors	KPIs <ul style="list-style-type: none">• Acceptance rate• Overall instructor satisfaction
Pain Points <ul style="list-style-type: none">• N/A	Pain Points <ul style="list-style-type: none">• Google sheets is not a very user friendly way of reviewing data.	Pain Points <ul style="list-style-type: none">• Visually sifting through spreadsheets is tedious.	Pain Points <ul style="list-style-type: none">• Making sure the assignment is optimal based on location, availability, and preference of the instructors	Pain Points <ul style="list-style-type: none">• Manually check schedules to ensure instructor not teaching back-to-back classes• Manual moving of data across spreadsheets.

Product Research: Requirements & Userflows

Meeting Notes + Add a view

- [Meeting #1](#)
- [Meeting #2](#)
- [Meeting #3](#)
- [Meeting #4](#)
- [Meeting #5](#)
- [Meeting #6](#)
- [Meeting #7](#)
- + New

Sprint Reports

- [Sprint: #1](#)
- [Sprint: #2](#)
- [Sprint: #3](#)
- [Sprint: #4](#)
- [Sprint: #5](#)
- [Copy of Sprint: #5](#)
- + New

Design Documentations

- [Requirement Document](#)
- [Technical Debt](#)
- [Heuristics Evaluation](#)
- [API Documentation](#) 1
- [SQL Relational Schema / ER Diagram](#)
- [Old Setup Documentation](#)
- [Super Sequence Diagram](#)

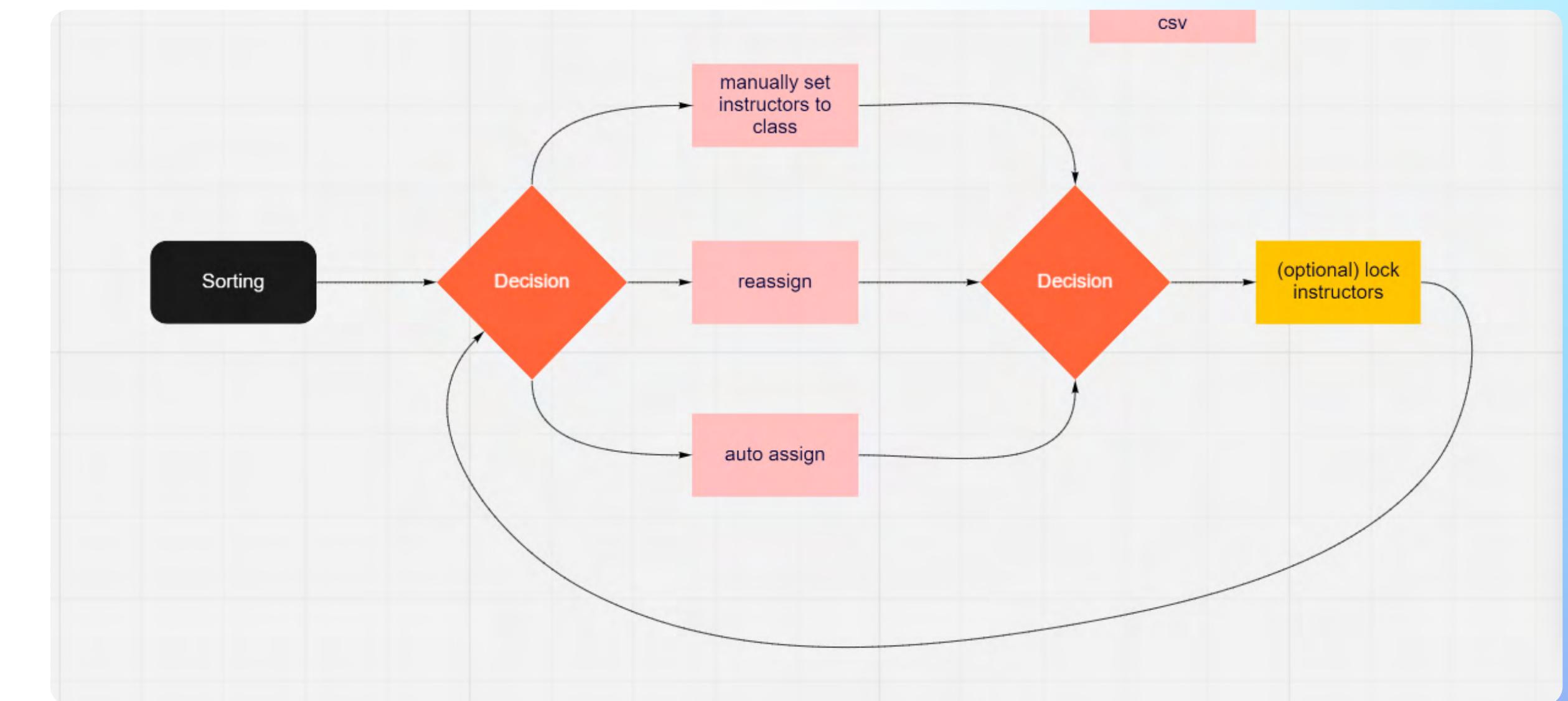
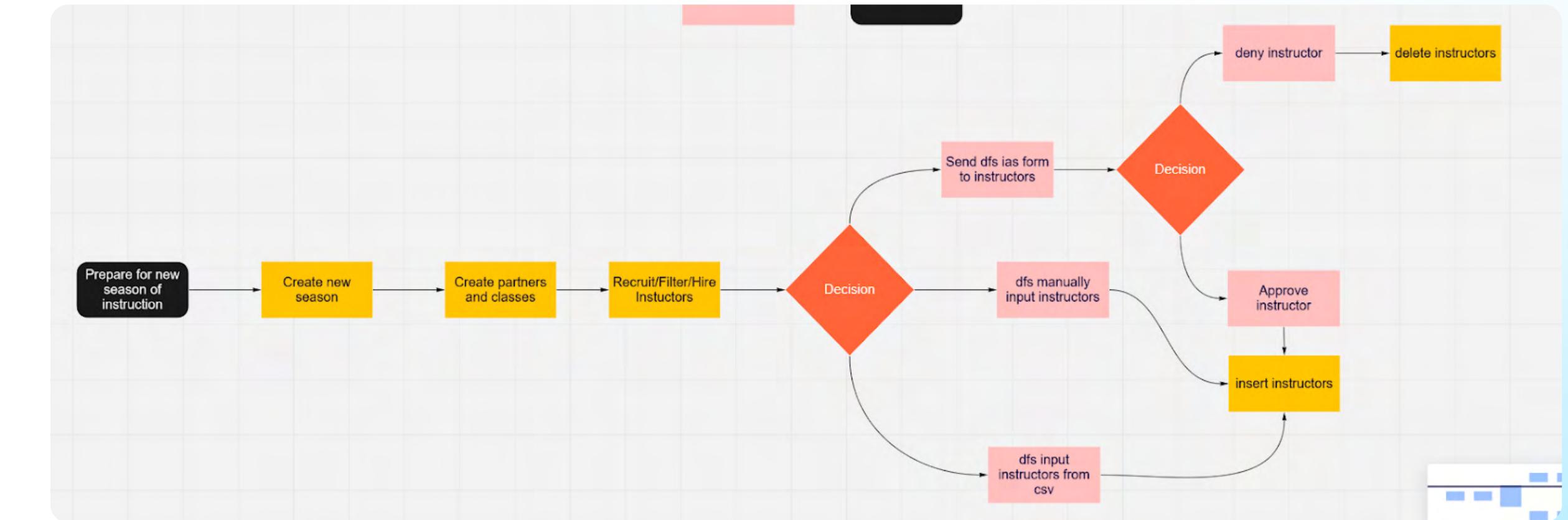
DFS Instructor Assignment Sorter (IAS) Requirements

Instructor Dashboard Tab:

- Should have list of all instructors
- Be able to edit instructor info
- Be able to add instructors and delete instructors
- Ability to sort (using columns)

Instructor Inputs:

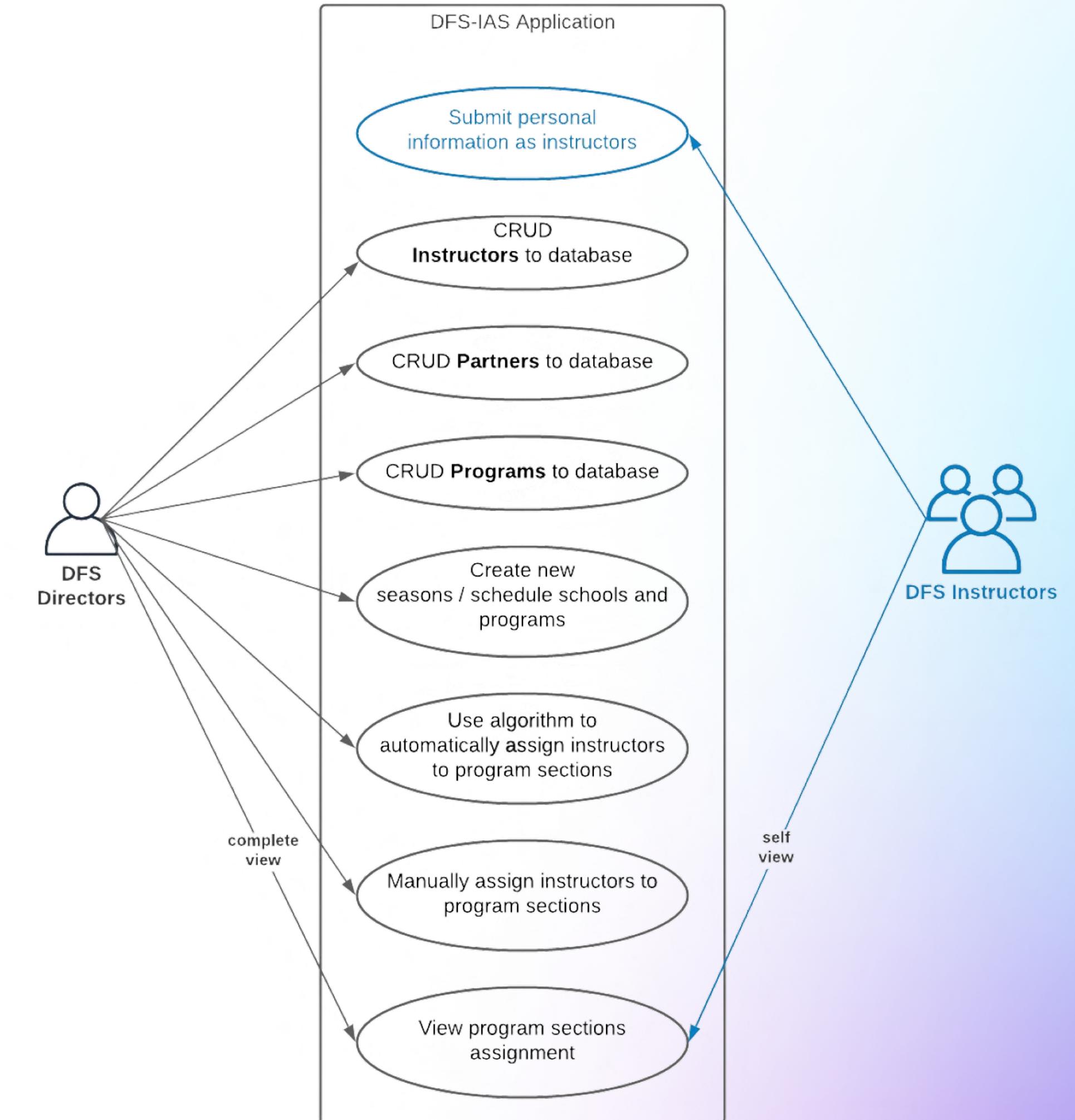
- Name
- Gender
- Year of the instructor in school (if applicable)
- Major
- University
- Region
 - Orange County
 - Riverside
 - Los Angeles
- Starting location (where they live)
- Schedule of days available
 - windows of time available
 - Timezone
- Car - Yes or no
 - minimum 1 for each group
 - if possible, more evenly spread out with 2
- Returning Instructor - Yes or no
 - being able to distinguish between new vs returner
- Shirt Size
- Program they're teaching
 - Let's Explore STEM
 - Coding Games with Scratch
 - Engineering for Kids
 - Web Development - WebJam
 - Mobile App Development - AppJam+
- Preferred Ranking of Programs
- Languages Spoken (check boxes for all fluent)
 - Spanish
 - Vietnamese
 - Mandarin



Product Research: Key Requirements

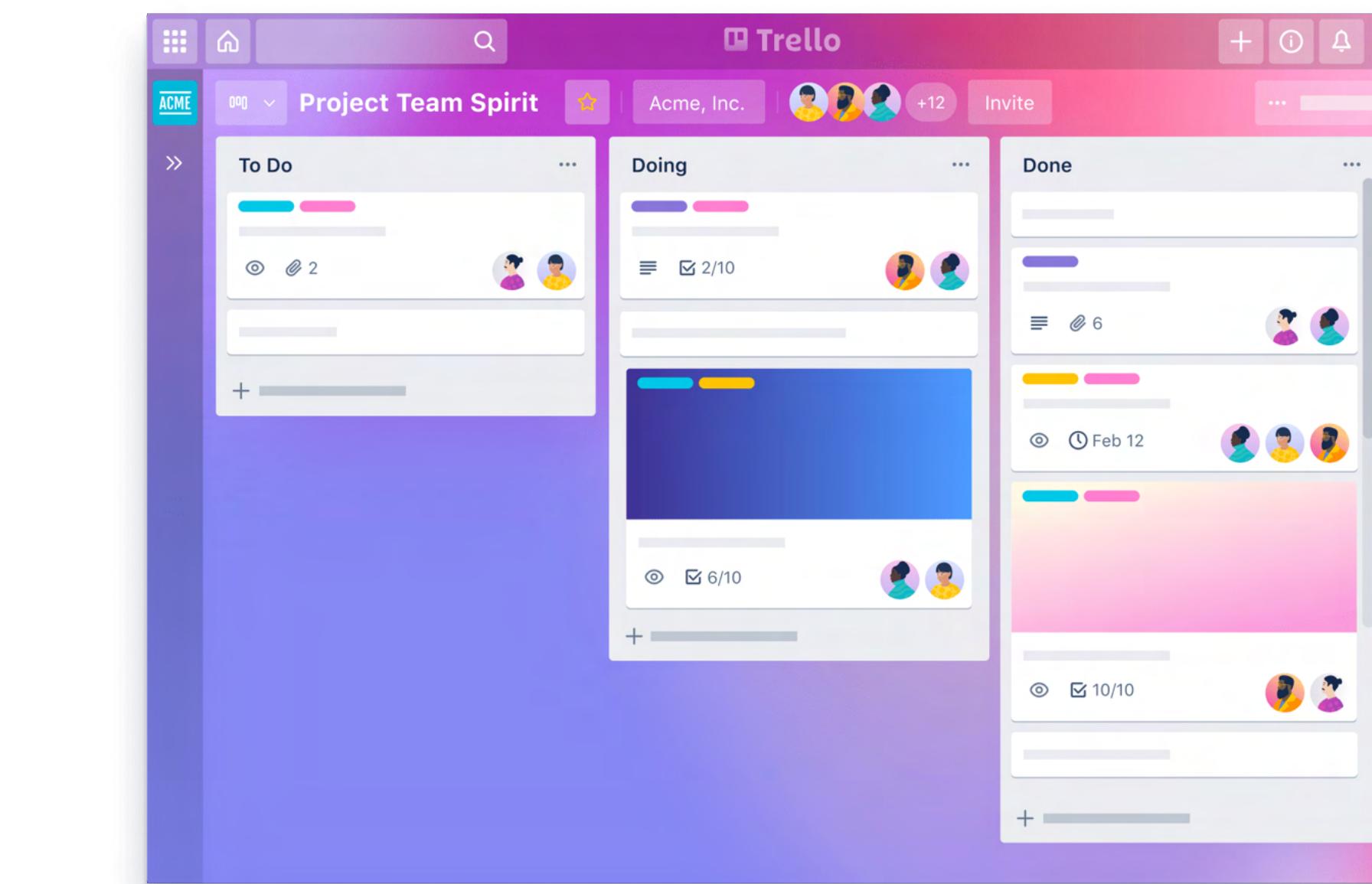
A web based system for managing programs, partners, and instructors.

- Automated instructor assignments
- Manual assignments via drag and drop.
- Ability to lock assignments in place and auto assign.
- Support CSV upload, manual entry, and self-onboarding links.
- Align with design language from their existing web portal.



Product Research: Competitive Analysis

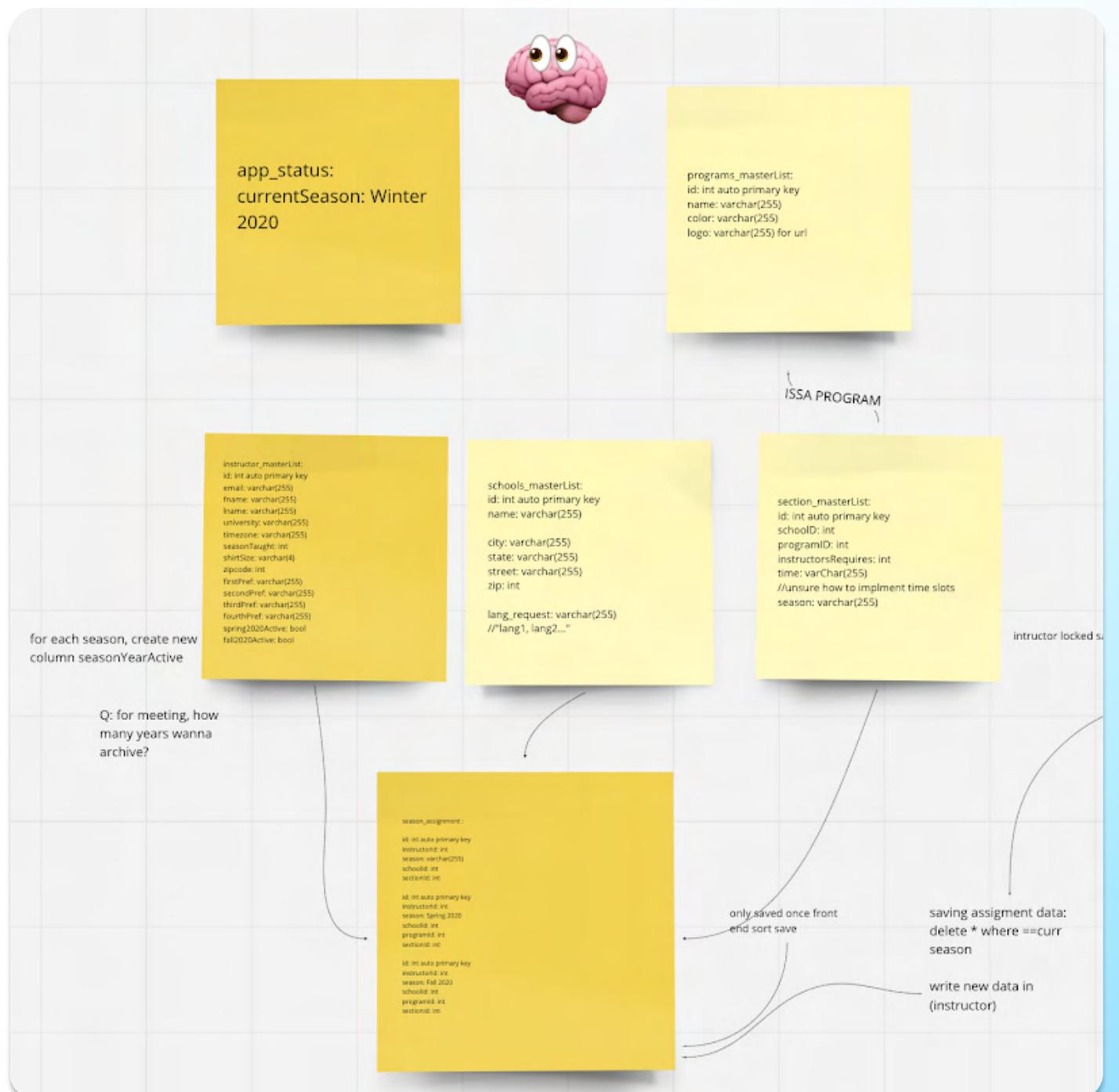
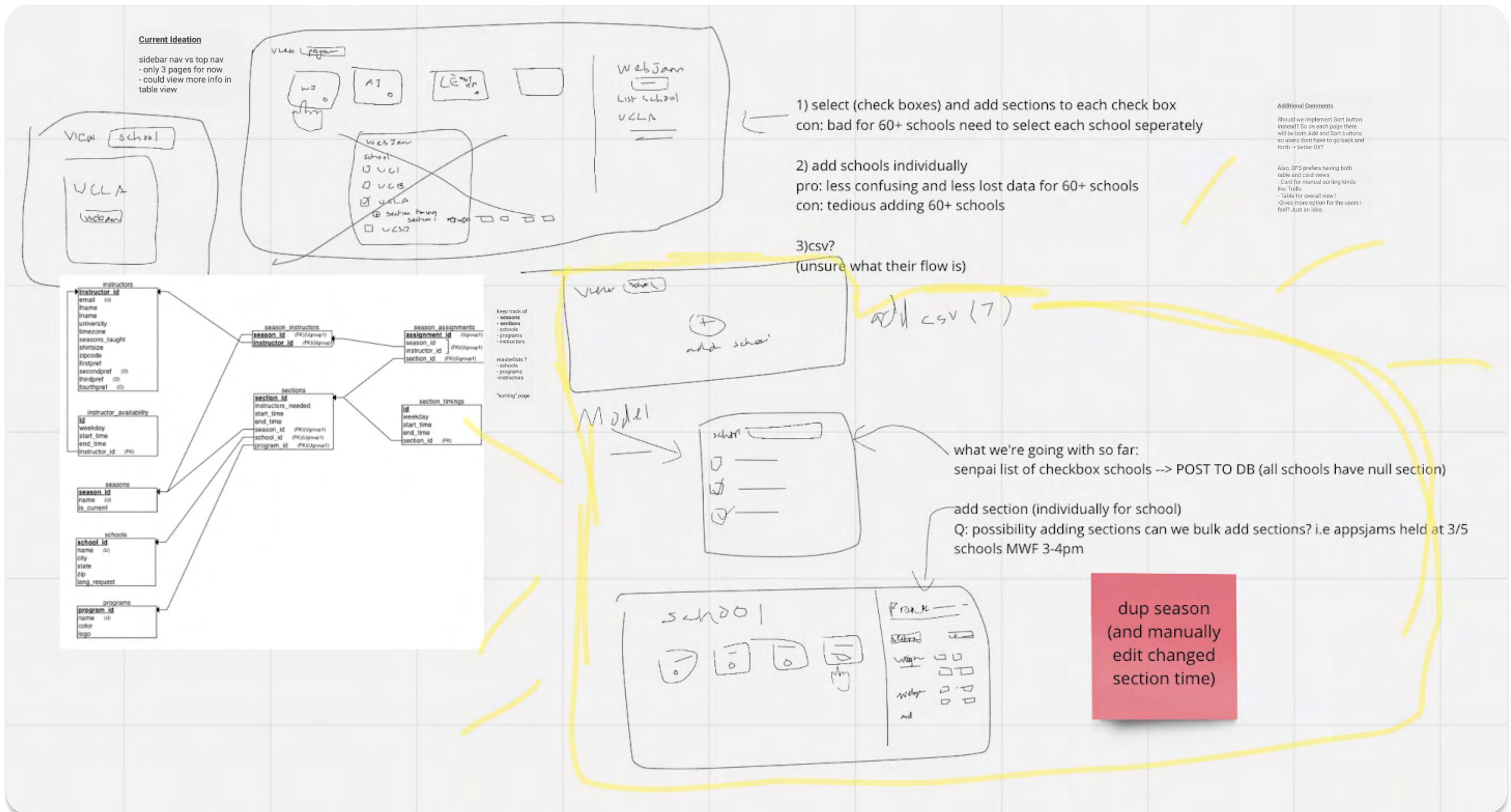
- Course management systems
- Learning dashboards like Duolingo
- Trello drag & drop cards



 DFS IAS

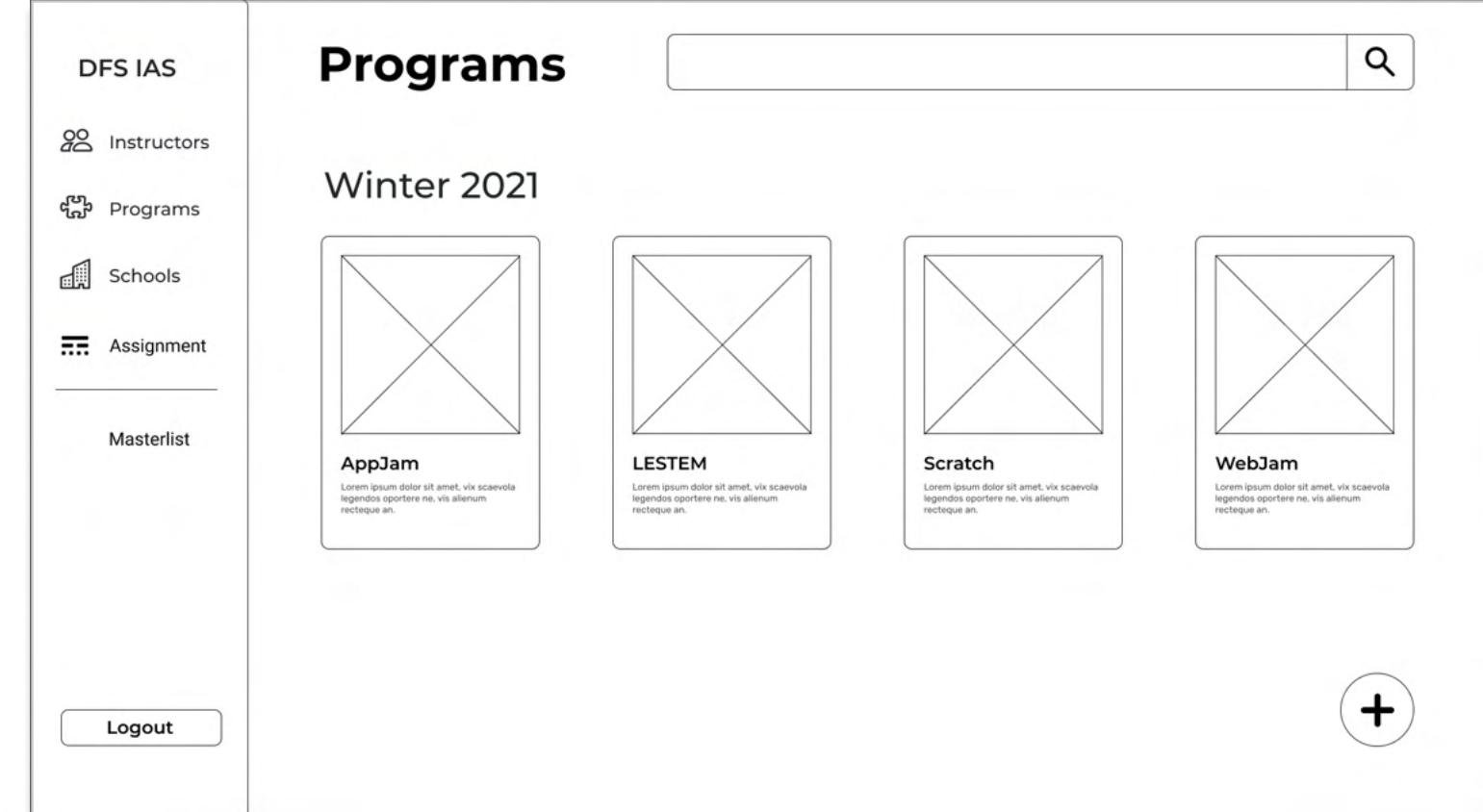
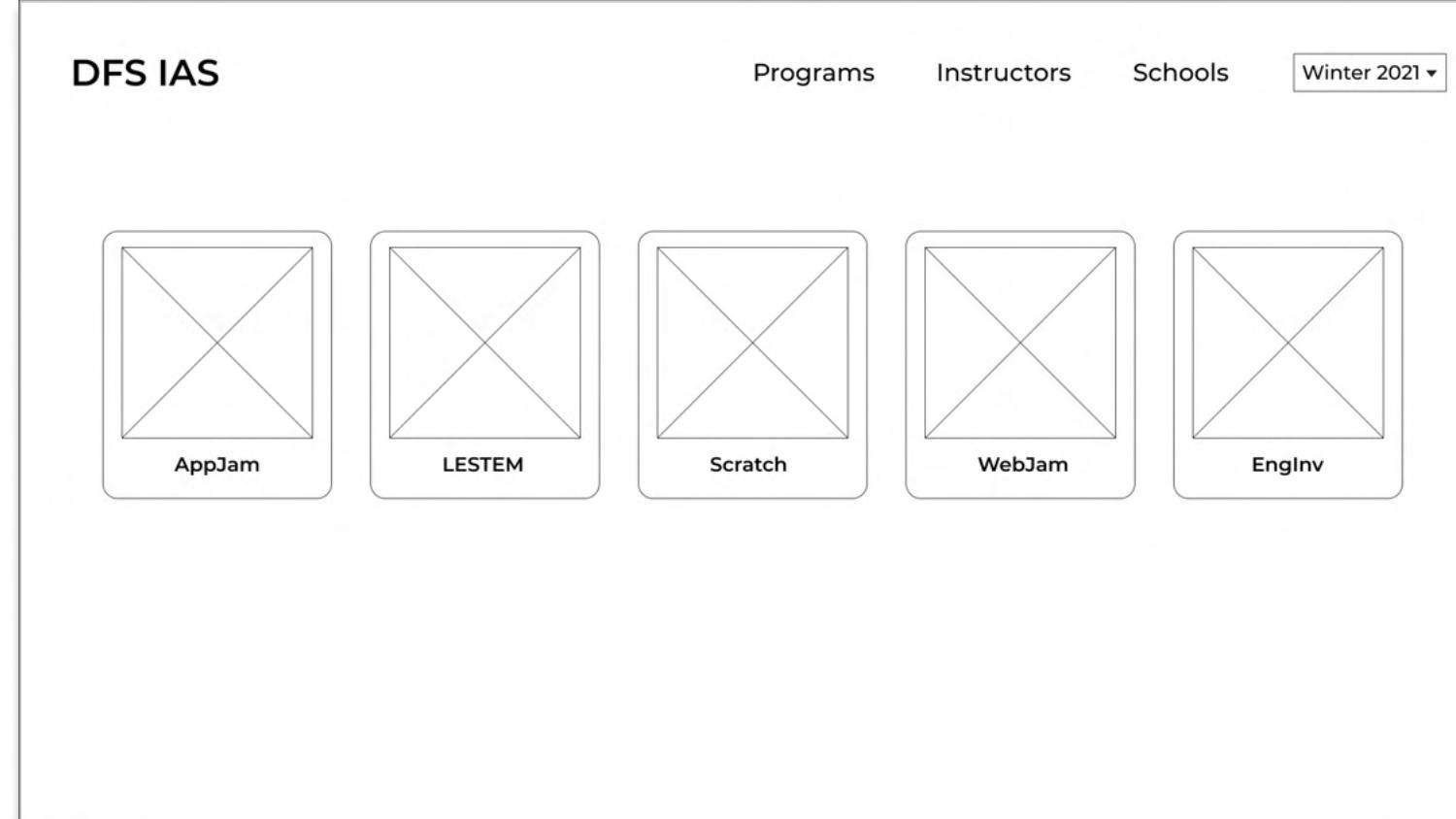
A composite image containing three screenshots of educational software interfaces. Top right: Duolingo's learning dashboard, showing progress for "French 2 Period 4" across 12 students. Middle right: Blackboard's course management system, displaying a list of courses like "Astronomy 212", "Chemistry 101", and "English Composition". Bottom left: A Trello board titled "Project Team Spirit" with cards in "To Do", "Doing", and "Done" columns.

Design Cycle: Ideation & Sketching

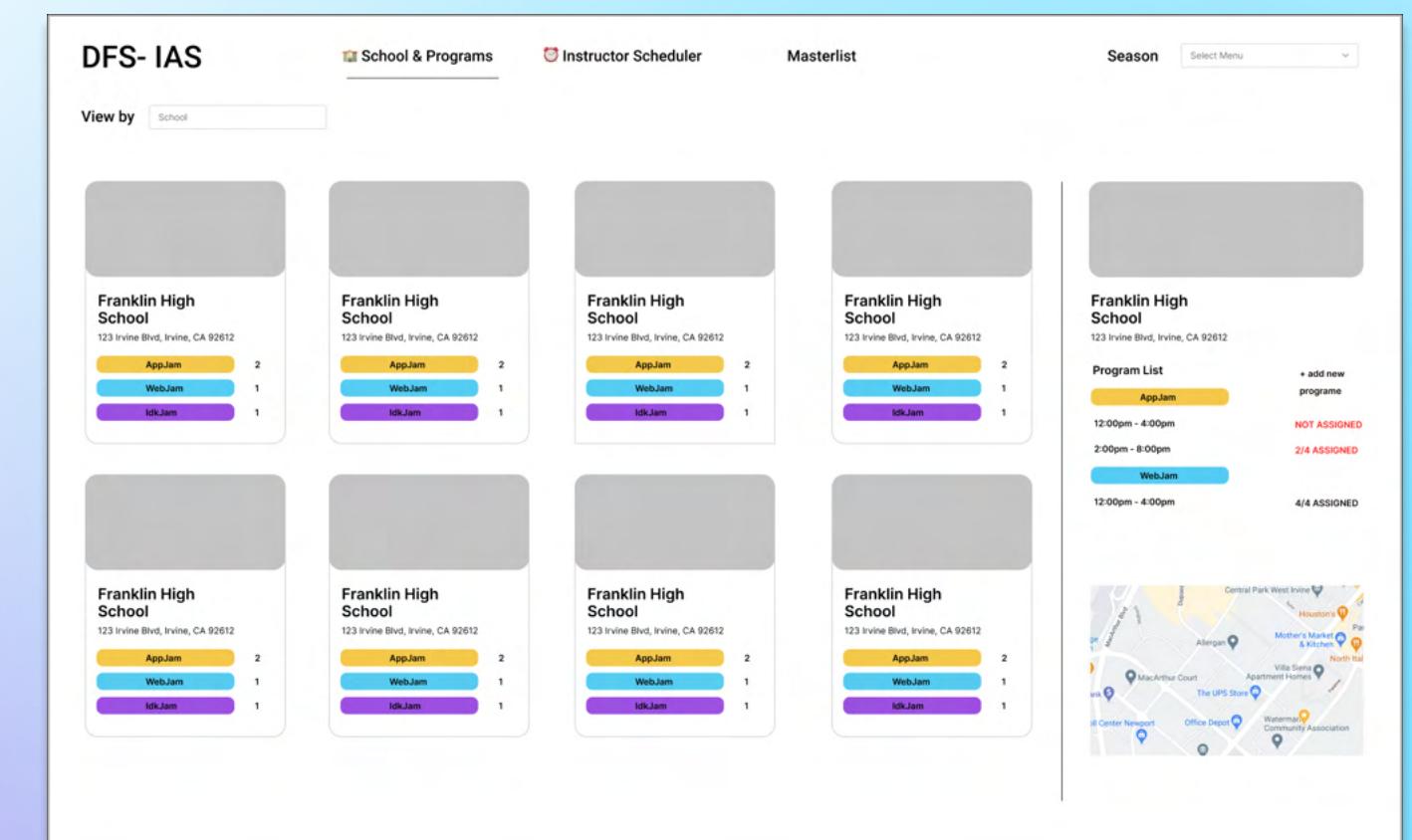
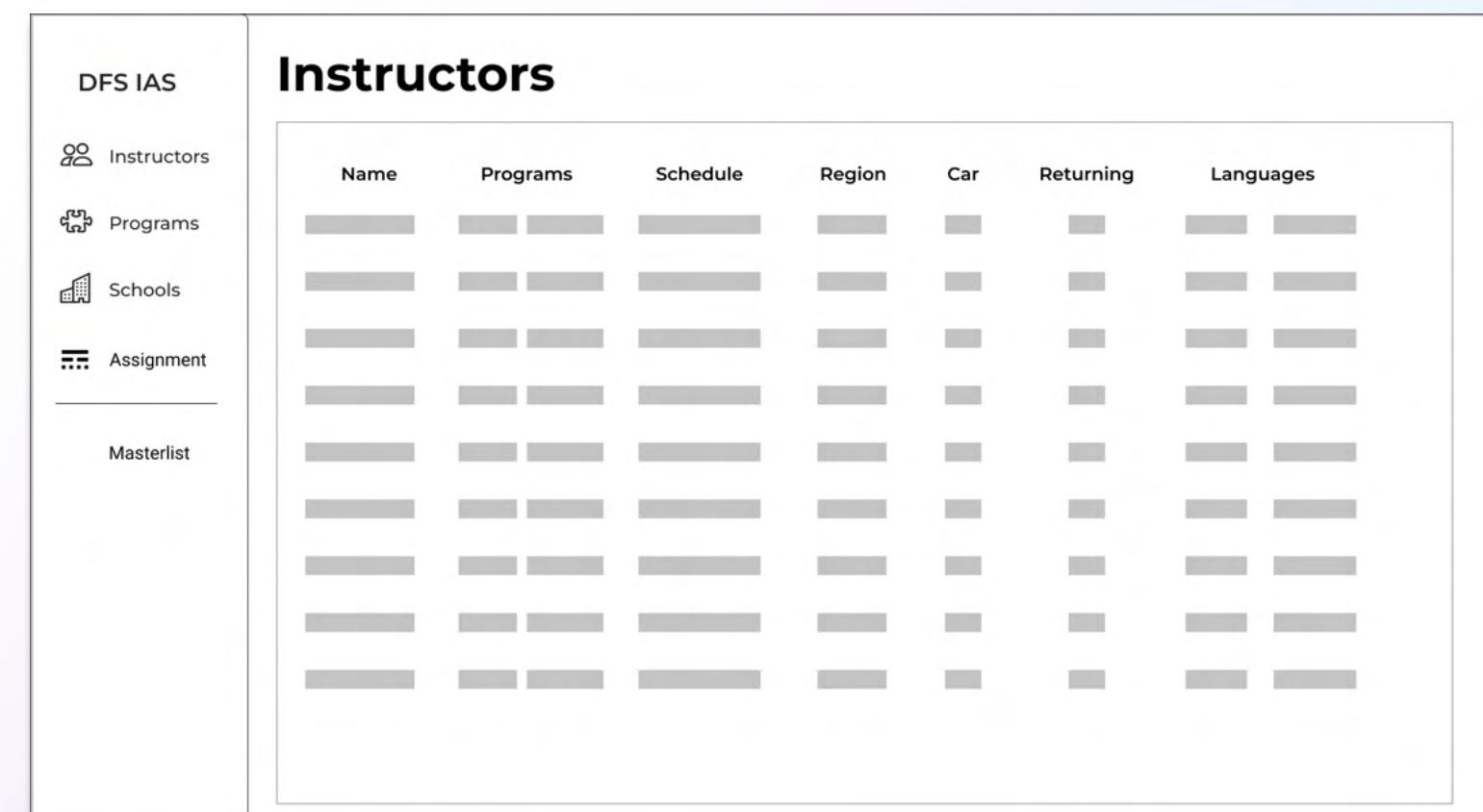
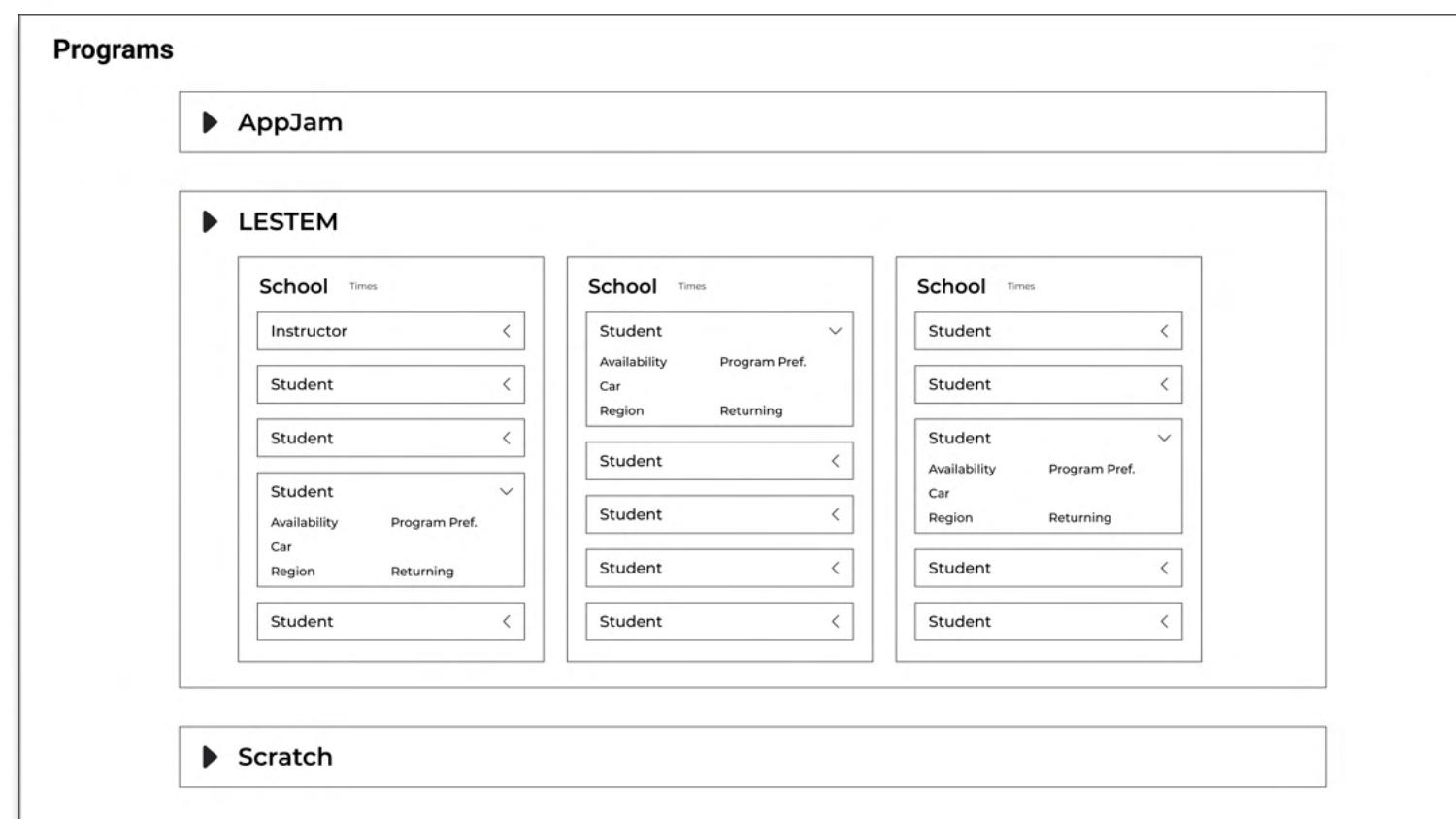
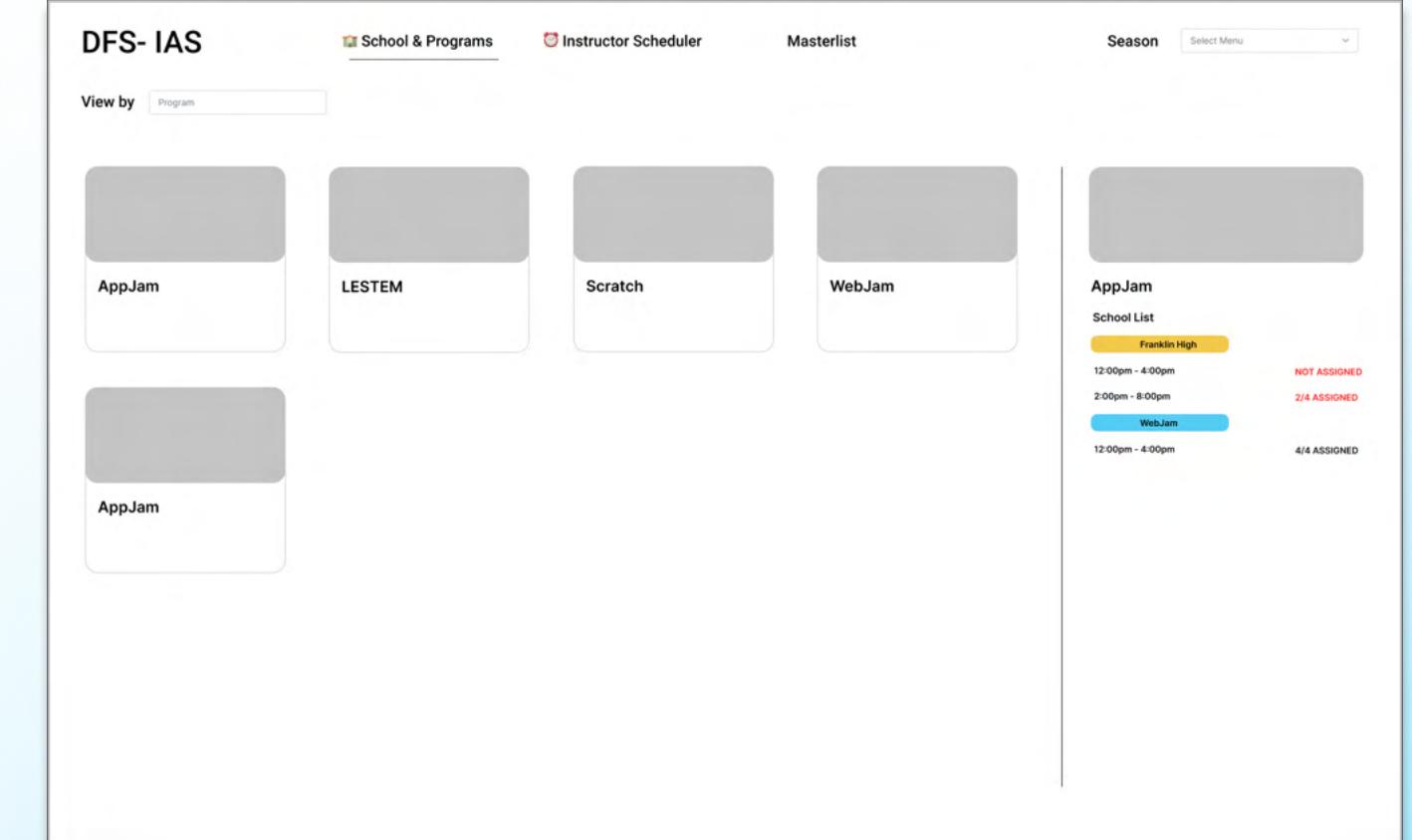


Design Cycle: Low Fidelity Wireframes

Iteration 1



Iteration 2



Design Cycle: Mid Fidelity Wireframes

Iteration 3



DFS IAS

View by: Programs ▾ Filter by: All ▾ Add Programs

AppJam LESTEM Scratch WebJam

Engineering Inventors

Winter 2021 ▾

AppJam

LESTEM

Scratch

WebJam

Engineering Inventors

Villa Fundamental
MW 5:00pm-6:00pm

Carr Intermediate
5:00pm-6:00pm MW

Fremont Fundamental
5:00pm-6:00pm MW

McFadden Intermediate
5:00pm-6:00pm MW

DFS IAS				Programs	Instructors	Sorter	Winter 2021 ▾
Filter	ALL ▾			Programs: ● ● ● ●			
NAME	YEAR	MAJOR	UNIVERSITY	PREFERENCES			
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater	2nd	Computer Science	UCI	●	●	●	●
● Peter Anteater 2	2nd	Computer Science	UCI	●	●	●	●



Programs Instructors Sorter Winter 2021 ▾

View by Partners ▾ Filter by: All ▾ Add School



Raymond A. Villa Fundamental



McFadden Intermediate



Gerald P. Carr Intermediate



Mendez Fundamental



Fremont Fundamental



Madison Elementary School



Muir Fundamental



Edison Elementary School



Raymond A. Villa Fundamental



Edit

AppJam
5:00pm-6:00pm MW

WebJam
5:00pm-6:00pm MW

LESTEM
5:00pm-6:00pm MW

Scratch
5:00pm-6:00pm MW

 DFS IAS

Programs Instructors Sorter Winter 2021 ▾

- ▶ Program 1 
- ▶ Program 2 
- ▶ Program 3 
- ▶ Program 4 

Unassigned Instructors

Search 

Filter by: Availability ▾

Peter Anteater 

MW, 2:00 pm - 3:00 pm
Th, 2:00 pm - 3:00 pm 

Peter Anteater 

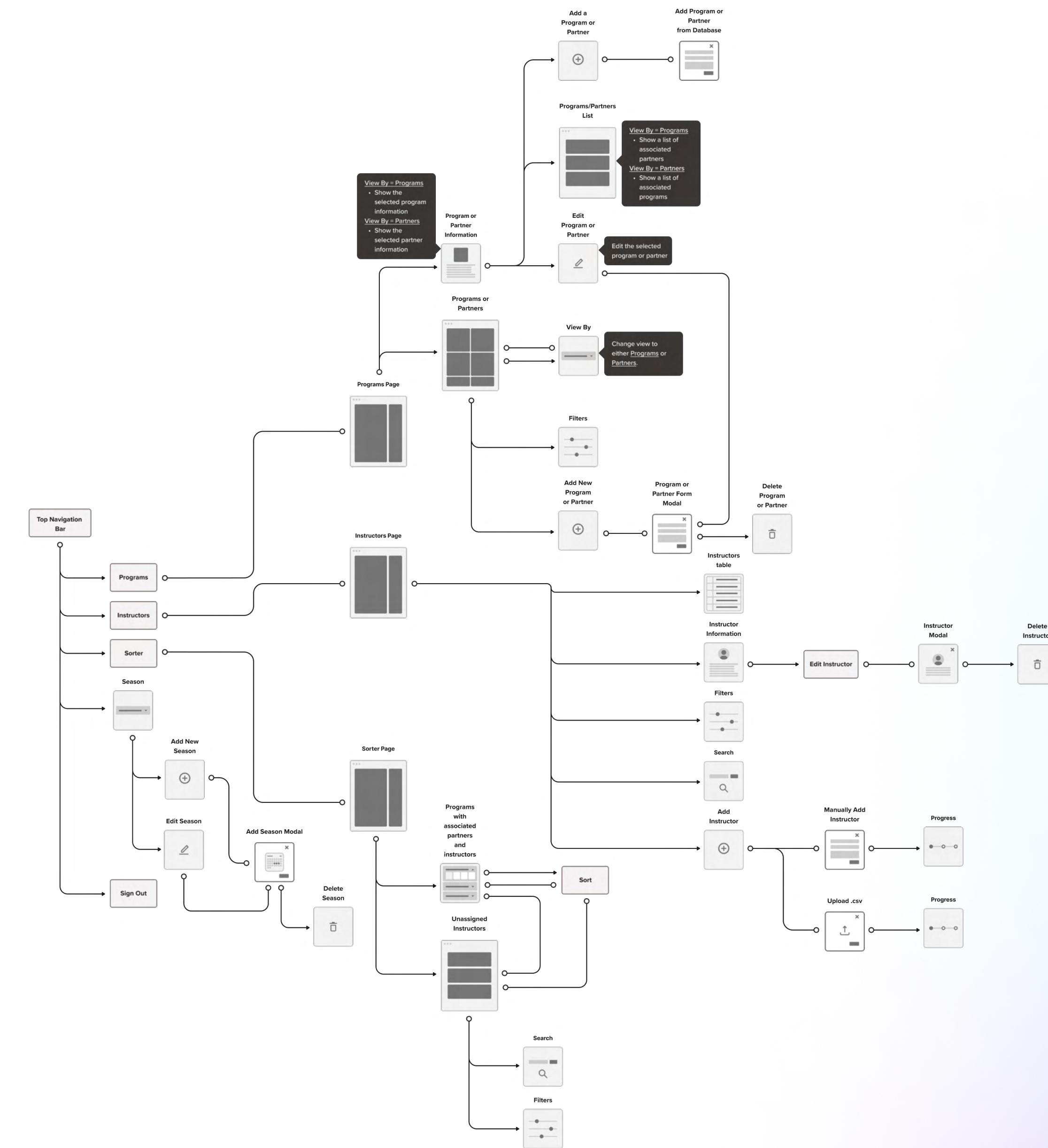

Peter Anteater 


Design Cycle: Creating a Design Systems

The image displays a collection of wireframe and design system documentation pages from a platform called DFS IAS. The pages are organized into several sections:

- Atoms:**
 - Typography:** Shows a grid of font samples for Display 01 (Medium, Bold), Display 02 (Medium, Bold), Heading 01 (Medium, S(S) Medium, Bold), Heading 02 (Medium, S(S) Medium, Bold), Heading 03 (Medium, S(S) Medium, Bold), Heading 04 (Medium, S(S) Medium, Bold), and Heading 05 (Medium, S(S) Medium, Bold). It also includes a section for Subheadings (Subheading 01, Subheading 02, Subheading 03) with a note: "It is a long established fact that a reader will be distracted by the readable content."
 - Icons:** A collection of Bootstrap framework icons including magnifying glass, file, upload, and cloud.
- Components:**
 - Cards:** Includes Side Bar Cards (e.g., Raymond A. Villa Fundamental, AppJam, Stella Adriana, LESTEM, Scratch, WebJam, Engineering Inventors) and Program and Partner Cards (e.g., Savanna High School, Dublin High School, Anaheim High School, John F. Kennedy High School).
 - Buttons:** A grid of buttons categorized by size (Small, Regular, Large) and type (Primary CTA, Secondary CTA, Ghost CTA).
 - Modals:** A grid of modals for various actions like "Add New Session", "Add An Existing Program/Partner", "New Program", "New Partner", "Add Instructor", "Upload Instructor File", "Instructor Lookup", "Add Instructor Manually", "Add Instructor Confirmation", etc.
 - Input Form:** A grid of input form components including labels, inputs, and dropdowns.
 - Sorter Board:** A grid of cards for "WebJam" and "Raymond A. Villa Fundamental" showing lists of items like "Barbara Bates", "Margaret Boyd", "Doris Wally", and "Shirley Agular".

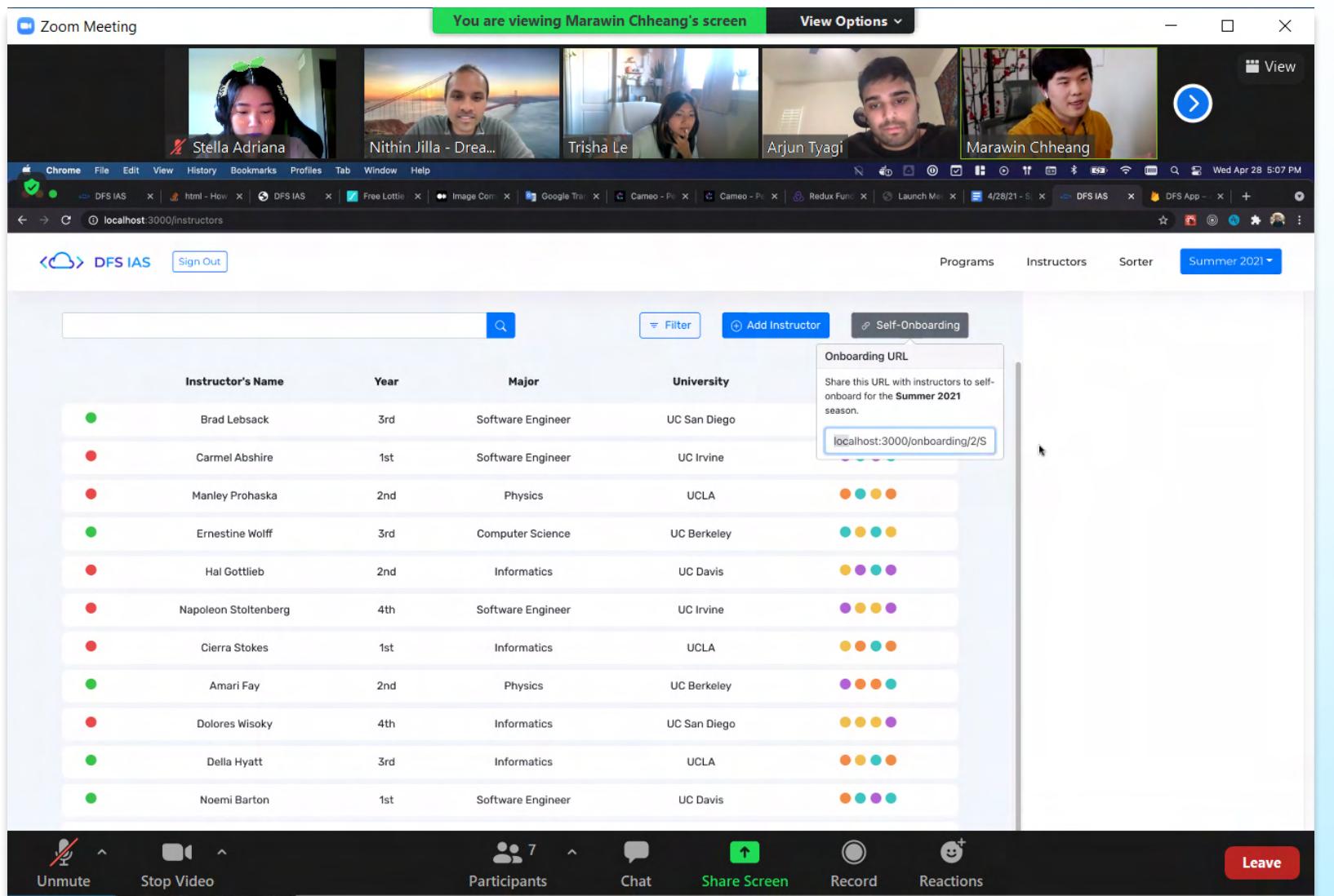
Design Cycle: Sitemap



Design Cycle: Validation

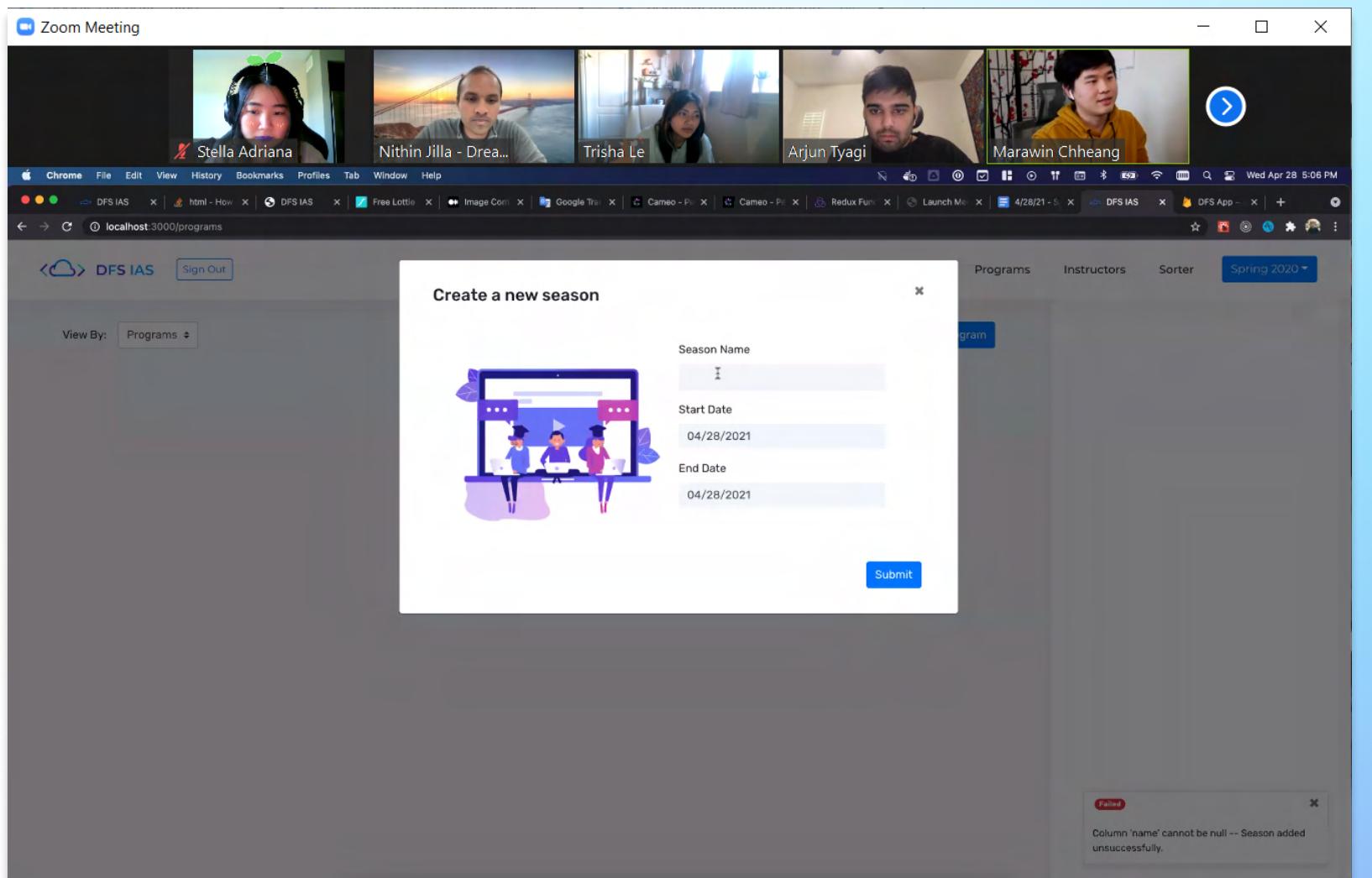
Moderated remote usability testing:

- Participants: Stakeholders/ Program Directors
- Completed once per design iteration
- Cognitive walkthrough for setup and assignment flows.
- Collected insights and feedback about interaction pain points



Insights:

- Required sidebar info for unassigned instructor counts
- Needed improved filtration by partner type and time slots
- Feature request: Instructor look up from previous seasons.
- Information overload on instructor cards. Move to modals.



The Results: High Fidelity Designs

Iterations 4-8

The image displays six screenshots of the DFS IAS application interface, illustrating the design evolution from Iteration 4 to Iteration 8.

- Iteration 4 (Left):** Shows the 'Programs' section with a grid of icons for AppJam, LESTEM, Scratch, and WebJam. A sidebar lists programs like Raymond A. Villa Fundamental and John F. Kennedy High School.
- Iteration 5 (Middle Left):** Shows the 'Partners' section with icons for Savanna High School, John F. Kennedy High School, Anaheim High School, Cypress High School, and Dublin High School. It includes a sidebar for AppJam and LESTEM.
- Iteration 6 (Middle Right):** Shows the 'Unassigned Instructors' section with a grid of instructors (Raymond A. Villa Fundamental, John F. Kennedy High School, Dublin High School) and a sidebar for AppJam, LESTEM, and Savanna High School.
- Iteration 7 (Bottom Left):** Shows the 'Instructors' section with a detailed list of instructors and their profiles. A modal for 'Add Instructor' is open, showing three methods: Instructor Lookup, Upload CSV, and Manual Input.
- Iteration 8 (Bottom Middle):** Shows the 'Add Instructor' modal with a more complex form for manual input, including fields for basic information, educational background, availability, and preferences.
- Iteration 9 (Bottom Right):** Shows the 'Add Instructor Manually' modal with a large search bar and placeholder text: "I-oh, looks like you have not selected an instructor yet. Please select an instructor you would like to access information."

Prototype / Video Demo