# Setup

A new customer is setting up the system on premises.

May require some hardware and server management skills

Should be doable with desktop system and minimal experience for small projects

Considering a 2-sign system.

1. User installs server/manager package on a server/computer
2. User images media for use with player hardware
3. User configures player media with network information
4. User installs display units (e.g. TV) at each player site
5. User attaches player media to TV
6. Upon boot user can see a status screen with network info, welcome message on each player
7. User accesses the manager interface, enters configuration for each player, and receives verification of correct configuration
8. User may continue to deploy content

# Authorized user can deploy media to players

(Assuming we have some authorization somewhere)

1. User has validatable credentials (or logs in with something else?)
2. User has viable media
3. User can drag+drop file(s) into the interface, or browse for files
4. User can assign a name to each file uploaded
5. User can add media to templates/playlists
6. User can give the media a global schedule (specifying union or intersection of times)
7. User can deploy new media set to specified players

# Marketing person uploading a campaign

MD (Marketing Director) Jekyll is running a campaign for a weekly event, and has created two images for use in advertising. One is to be used the day of the event, and the other is to be used other days. In particular, he only wants the images displayed in the mornings, as the event occurs “at lunchtime”. D’tor Jekyll is authorized to use the system.

1. Jekyll logs in to the system
2. Jekyll navigates to media upload, uploads both files
3. Jekyll names the former “Lunch Campaign - Day Of”, the latter “Lunch Campaign - Other Days”
4. Jekyll assigns the “Day Of” image a schedule of UMTRFS by 0000-1130
5. Jekyll assigns the “Other Days” image a schedule of W by 0000-1130
6. Jekyll adds both images to a template with a media slot fitting the TV, instructing the player to distort the media to fill the space (D’tor Jekyll doesn’t like wasted space)
7. Jekyll deploys the content to whichever signs contain/refer to the altered template

# Hypothetical: non-authorized user upload

1. Joe student wants an ad up for his student group
2. Joe accesses server (maybe authenticates?), submits file
3. Joe specifies what schedule he wants
4. Authorized user Sue comes by later (or is prompted by email) to confirm that this is OK
5. Sue makes any necessary modifications
6. Sue assigns to a template, deploys as normal

# Layout - V → H

So you want to have more than one media item on screen at a time…

1. Sam wants to have a few things on the screen at once. They have a bunch of portrait images but the display is landscape, so they’re thinking put two side-by-side. They also want [something] at the top (logo? time?)
2. Sam creates a new template for this endeavor
3. Sam drags in a vertical tiling container (PP), to contain two segments: the title and the media area
4. Sam drags a horizontal container into the second area of the first container, to contain two areas for media
5. Sam creates the appropriate playlists and drags them into each of the new areas
6. Sam adds an item to the top area of the first container
7. Sam sets the first container to have a fixed height, and for the second to “grow automatically”
8. Sam assigns the template and deploys

# External System Submission

1. “Tech team” sets up external system to submit data to signage server for user content
2. User [does whatever] in external system to submit an ad
3. “Signage manager(s)” accept sign parameters, sign is added to schedule and is deployed as usual