

Beyond Jobs: Kauffman Challenge Deliverables

DOCUMENT 1A: Narrative Web interface work

Table of Contents

Scope of the mobile app	1
Mobile screens	2
Issues in development	4
Availability intakes.....	4
Agency controls	6

Scope of the mobile app

uFlexi is hugely sophisticated across its ten levels of users and 4 groups of entities (back office, agency, buyer, worker). We needed to identify the most useful, and achievable, parts of our functionality for an initial web app.

It was decided to focus on an app for workers that delivered all the day-to-day functionality they were likely to need, particularly for an individual with every uncertain availability for work. (These people - typically those with fluctuating medical issues, unpredictable caregiving/parenting needs or the demands of flexible studying or a home business – are our most benchmark users. If we can fully handle their needs, individuals blessed with more predictable lifestyles are easy.

We assumed registration would require a desktop device because:

- uFlexi requires far more detailed information than simpler labor markets with no serious commitment to expanding options for each worker.
- We have to explain new concepts. For instance, “availability” – the hours a person wants to work - is often not understood at first. The concept of having that choice in advance is alien. Many low-skilled workers are simply called and asked “will you do today’s 1PM-9PM shift, Yes or No?”. The flexibility and control we offer can be more intuitively communicated on a larger screen.
- We need careful selections, not “fat-finger” typing on a small screen. For example, there are hundreds of junior nursing qualifications, some with confusingly similar names. It is vital the system knows exactly which one an individual claims to have.
- The intermediary has primacy. The system itself will not authorize an individual into the market, an intermediary has to take responsibility. This means the worker will likely attend for an interview with an intermediary if not already known to them. In this context they are likely to attend a branch of the intermediary where desktop devices will be available even if they don’t have one at home.

Once a worker is set up in uFlexi, attached to an intermediary (agency), we need to cater for these user journeys:

- a) Reviewing availability: “availability” is the hours the individual can work; today, tomorrow or weeks ahead).
- b) Changing availability: this has three variants; (a) entering a new block of available hours (b) amending an existing block (c) deleting an existing block.
- c) Handling bookings: the user needs to receive details of a new booking then be able to confirm or reject it after reviewing details, including seeing a map of how to reach the place of work.
- d) Handling timesheets: we have to push timesheets that need to be approved this week, pursue those that are overdue and enable quick approval. Timesheets drive payroll and must be completed for the buyer to be charged and the worker paid.
- e) Review “Checks”: Checks are the datapoints known about each individual and used to align them with roles (below). An individual needs to be able to look over their checks, particularly those with an expiry date (such as a licence reaching the end of a two year applicability period).
- f) Manage roles: Roles are the types of work the person can do (based on their checks). Each can be set to Active (that worker will do that type of work at present) or inactive (they are qualified but chose not to do that work, perhaps because it is lower paid compared to their other roles). Any worker with a wide range of checks will have “borderline” roles; ones they would rather not do but may have to because there aren’t enough more desirable bookings yet. We therefore see toggling roles as a daily interaction (“I have several childcare bookings next week which help towards a qualification I seek, I want to try turning off my willingness to do secretarial work for a few days”.)

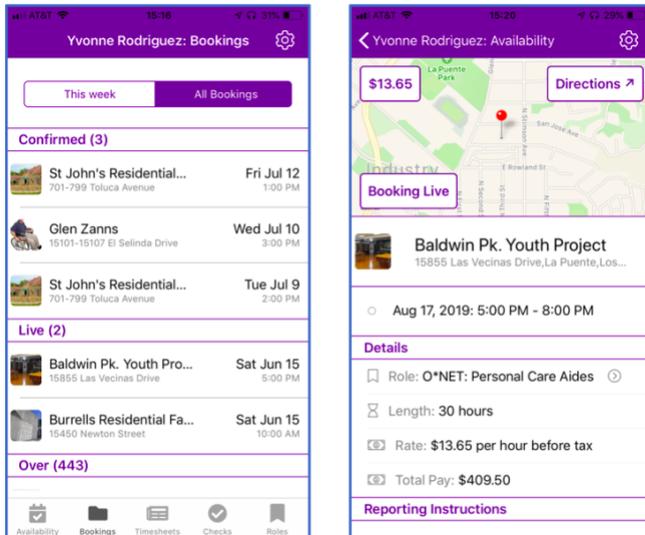
[Xamarin Visual Studio](#) from Microsoft was used as the underlying tool. This limited the range of off-the-shelf elements available but greatly increased productivity.

Mobile screens

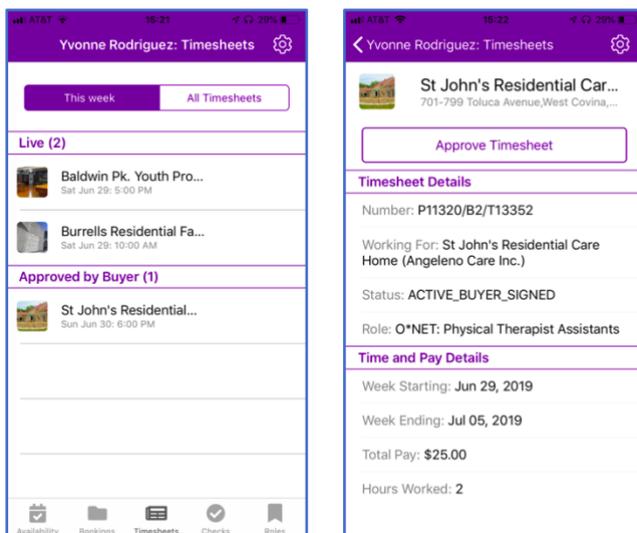
At time of writing, the app’s screens looked like this:

- Screens for handling availability became a separate project because of the unique challenges. They are covered in the section below.

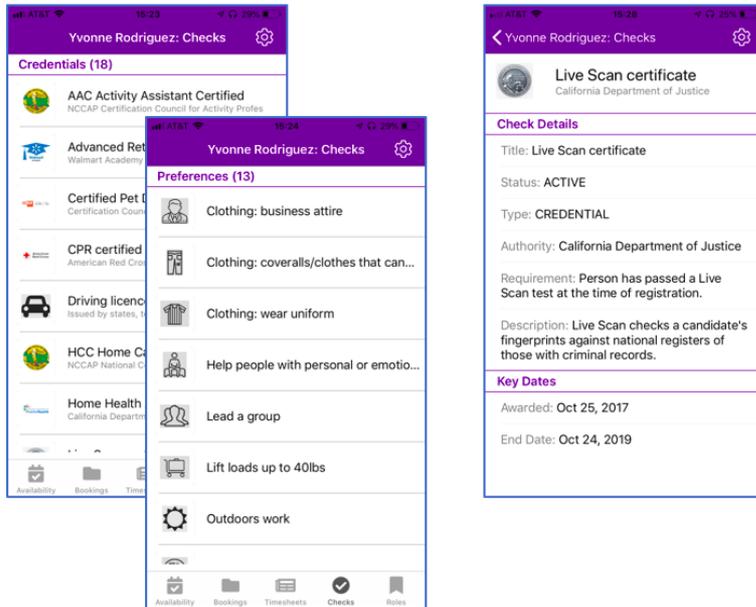
a) List of bookings and view of an individual booking once the worker has confirmed acceptance in time:



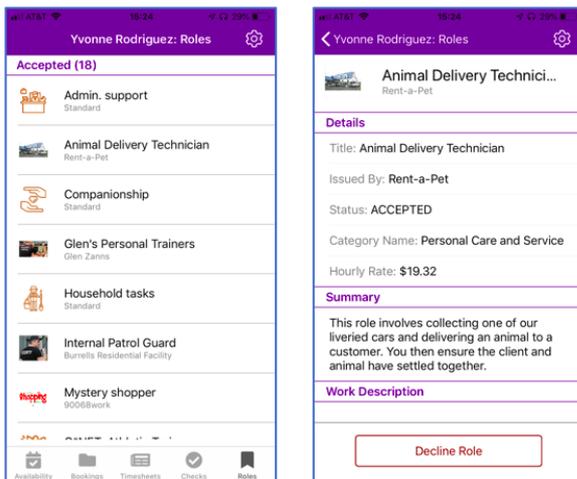
b) List of timesheets for this week separated by the status of the timesheet at the present time (a fuller list is available by tapping the "All" tab at the top) and a view of an individual timesheet that has not been approved by the worker yet.



c) Checks page shown in Credentials mode, then spooled down to user's list of Preferences. Plus an individual Check's screen (note expiry date at bottom, this pertains to the individual not the Check overall).



d) List of the user's Roles with those that are currently active at the top. On the right, the view of an individual – buyer-specific – Role that is currently set at active.



- All data is fictitious.

Issues in development

Availability intakes

How to take in, and output, an individual's times of availability for work becomes more challenging the more a marketplace aims to empower its workers and accommodate those with uncertain lives. Typical commercial "gig work" marketplaces handle it with an "are you available now: Yes/No" tickbox. But this keeps each individual in a state of permanent standby, impossibly demanding for those who must make arrangements for childcare or the care of an adult dependent before leaving the home to work.

Some commercial staffing agencies will allow inputs of actual date/time availability in four hour blocks. We do it in half hour increments. The intuitive way to handle this is a grid condensing a week's worth of half hour blocks into one grid with 336 cells (which has to offset a given day when clocks go backwards or forwards). This example shows such a grid at 8.30AM on July 3.

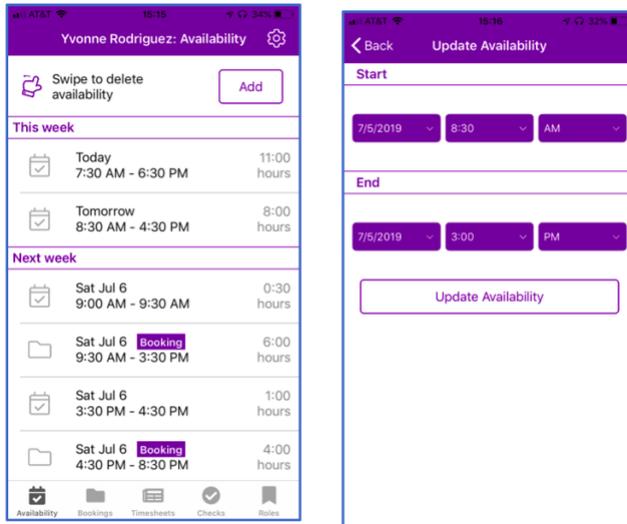


We have found this interface to quickly become intuitive once a person experiences the basics. But the input process – selecting a square then dragging the cursor to change the status of the encompassed cells (from available to not available, or vice versa) cannot work on mobile. To navigate around the grid, a user must touch at least one square thereby changing its state.

Tests with users using a version of this grid generated in Responsive Design for their phone proved this point. It's possible to learn to move around by magnifying the grid and touching the intersection of gridlines. But it is a sophisticated art requiring a high-performance device and contrary to our commitment to simplicity for users.

We therefore alighted on a list of blocks of availability any of which could be amended with the capacity to add a new block. These blocks had to be compliant with working time controls that might be applied to the worker by an intermediary, for example a college agency finding students work but only permitting them to do 12 hours a week with at least two days with no availability as mandatory. Users strongly urged us to include the worker's bookings in that flow of availability. This was done with the capacity to tap a booking to open its details page.

Natural language "Today" and "Tomorrow" rather than the dates for example, was then added. This delivered this list view of a person's availability (left) and a screen to work on a particular block (right):



In the list view: A user swipes right on a block of availability to reveal a Delete button. Availability can be amended, or entered from scratch, using a screen to manage a particular block. That is accessed by tapping the NEW button for an empty block, or the existing block to open it for amendment.

The individual block screen takes inputs through standard spinner wheels. It bounces the cursor between boxes as each one is completed. A user comfortable with their phone can enter a new block within 5 seconds.

We find the intakes on the right hand screen slightly fiddly and know many users would prefer a more “chunky” feel. However, that was tried and found to distort on older, smaller, phone screens. The current size of inputs is a compromise.

Another issue around availability intakes is the likelihood of a user trying to change availability while their device is set to a different timezone. (For example entering availability while on holiday in Hawaii for next week when they are back in Los Angeles.) This is more of a risk on phones because they can reset timezone without a user being cognizant. We believe the dangers of a mistake in trying to mentally offset time was too high. If the app senses a user is not in the right timezone it displays a warning on the left hand page.

Agency controls

The CEDAH (Central Database of Available Hours) displays on the web in each Agency’s colors, with their logo’s using their terminology and following their rules. We wanted to carry this over to mobile. The primary app color (for example the purple in screens above) is agency specific, picked up from their web settings. We attempted to put the agency logo on the screen icon used on a user’s phone homescreen but Apple will not permit multiple iterations of the one app so a generic “CEDAH” logo and wording had to be used. Agency permissions and rules are picked up from the user’s log-in details.

We experimented with Agency logo’s after log-in but users were concerned about cluttered screens and the only feasible size would reduce many logo’s to visual incoherence. Agency’s wording is however used throughout. We are also able suppress payrates within the app if that is the web setting for an agency.

