

# Homeflow Development Test

You can always find the most up to date version of this document at:

[https://github.com/homeflow/homeflow\\_test](https://github.com/homeflow/homeflow_test)

Homeflow is a fast paced, exciting, small team. We need our developers to be self starting, to understand specs intuitively, and be well rounded. We also like people who take their own initiative.

We're all about property search, in a number of different applications, across over 200 different websites. Some are estate agent websites, some are property portals / search engines, and some are mobile sites or are embedded within Facebook applications. Most of our systems are built around internal APIs.

If you are the kind of developer who can thrive in that kind of environment, you're likely to enjoy the Homeflow Technical Test... So here we go...

## About the test

The test is designed to help us assess your application development skills with a fun, quick project - which you can cook up on top of our API.

It will test your ability to:

- Understand a loose spec
- Work with a good clean - but only lightly documented API
- Be well rounded
- Work quickly - under your own steam
- Show any front end skills you've got, not Photoshop (unless you can) - really we aren't judging this based on look and feel, but certainly any HTML / CSS / Javascript prowess you've got
- Demonstrate any natural talents for UX

Don't be scared. We are NOT expecting candidates to finish the application - or have it particularly polished - that would be outstanding. You may only get as far as a form calling back results. Or you may get the whole app working, but without any styling, or in a different layout. Frankly, you might create something different, simpler, more interesting, or better.

Please take around 4 hours to make the test - certainly no more - we're not looking for a polished item. Just ways you approach code.

## The Brief

Take a moment to have a look at some examples of traditional property portals. They each feature:

1. A way to perform a search
2. A listing of search results
3. A 'property details' page

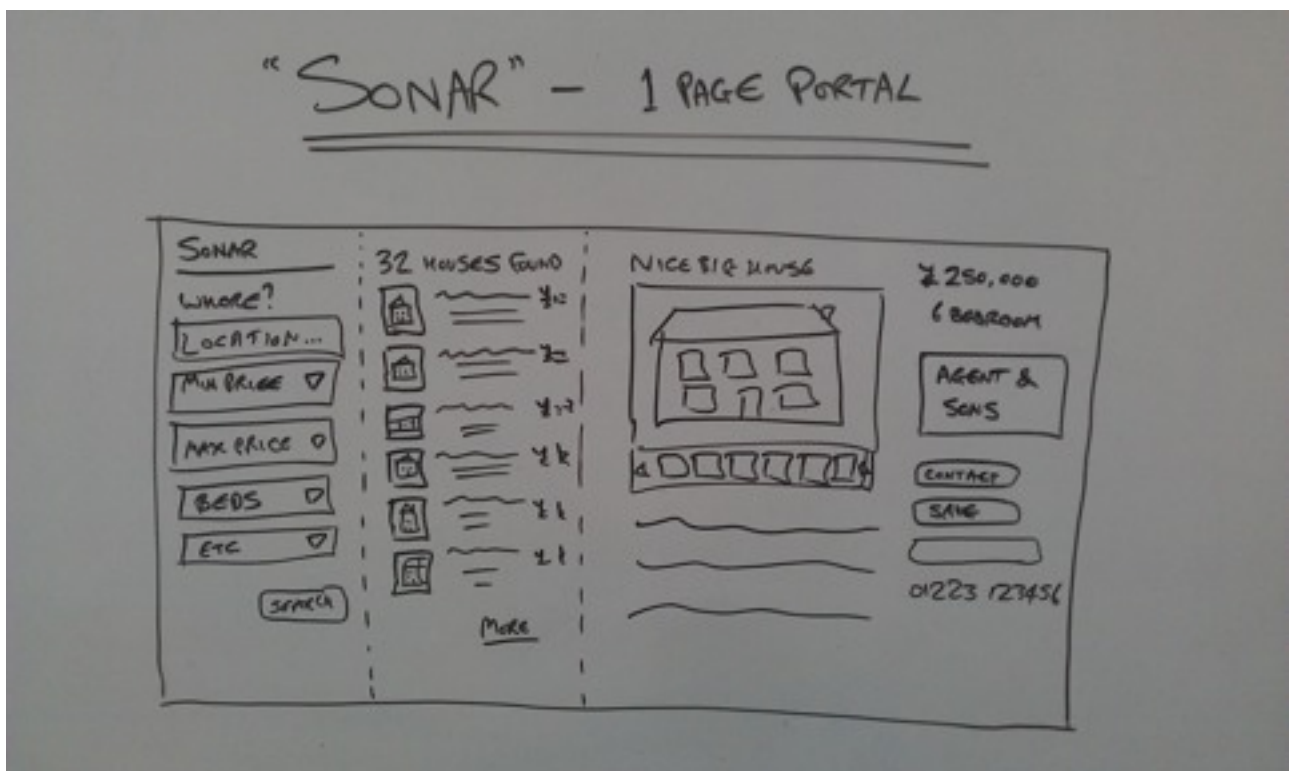
<http://www.needaproproperty.com>  
<http://www.countrylife.co.uk/property/uk>  
<http://www.zoomf.com>

We'd like a simple, one page, portal prototype called Sonar. Sonar is intended to be a proof of concept portal designed for ease of use on both desktop and tablets.

You can build sonar with whatever technology you feel comfortable with (although we would prefer if you chose Ruby if you decide to have a server side component). We're going to be deliberately vague with the spec as we'd like to see what you come up with.

Here's a sketch of the sort of thing we have in mind. A user should be able to select a bunch of search criteria and should then be presented with a list of houses. When the user then selects a house they should be presented with the property details page.

Again, this is just a guide - if you can come up with something to wow us that's different to this then you should feel free to experiment.



The data for this portal will be provided by the homeflow api.

## Data

We'll provide you with an API key. Our API documentation for all the calls you'll need can be found at: [https://github.com/homeflow/homeflow\\_api/wiki/The-API](https://github.com/homeflow/homeflow_api/wiki/The-API)

If you choose to use ruby for the data parsing then you'll find a gem pre-packaged at [https://github.com/homeflow/homeflow\\_api](https://github.com/homeflow/homeflow_api)

Our API is new, and so is our documentation. If you find any errors or anything you're unsure about feel free to get in contact.

Jeremy Tapp - 07779 293 454  
Daniel Cooper - 078 251 66908

### **Submitting & hosting**

When you've completed your project, upload it to github and send us a link. Try and keep your API key secret (ie, don't put it version control - .gitignore should be helpful here).

You should be able to get your project hosted in the time you've got. Heroku is often a good choice for free hassle free hosting (and it integrates well with git)