## 

## 

## 

## RAWNET RAILS DEVELOPER TEST

## There are two parts to this test, each one targeting a different aspect of the work you will be expected to carry out at Rawnet. The two sections aim to test your skills in the following areas:

* Planning and architecture
* Rails development
* More complex algorithms
* Coding and problem solving ability

## 

## Q1. PLANNING AND ARCHITECTURE

This section is to test your ability to plan a small project and architect the system and database. What follows is a basic brief for a simple asset management system.

What I would like to see is the process you would go through when designing and architecting the system. No code is required, although if you want to include a basic database structure or similar that’s OK. I want to know things like how you would implement the search, what methods of image manipulation could be used etc... I also want to know what sort of development methodology you’d use and so on.

Keep in mind that this is a simple system that needs to be completed on a small budget, and will need to be maintained by a number of different people.

The system should be a small file repository to allow shared access to a number of different assets such as images, Microsoft Excel and Word files, and Adobe Acrobat PDFs.

There are a number of users who access the system using their email address and password. Each user can either browse through the assets by category, or search based on title, keywords or description. The user can choose to download assets directly to their computer.

For the case of images, users should have the option to download the image at a size of their choosing. For example, if a high resolution image is uploaded, the user should be able to choose that they want to receive a 640x480 image back. The system should deliver a resized version to the user while retaining the original.

New assets can be uploaded by any user. Once uploaded, the asset can be assigned to one or more categories (the user can create new categories, displayed in a tree-like style). The asset will also have an amount of arbitrary metadata such as title, keywords and description, each of which are user entered.

An additional requirement is that this system will (in the future) have multiple front end clients, for example a desktop website and iPhone application.

## Q2. COMPLEX ALGORITHM WITHIN A RAILS APPLICATION

This section aims to test your coding and problem solving ability. Note that I’m not giving any structured guideline, but rather want to see how you’d tackle the problem left to your own devices. Here is your chance to show us what you can do, with the opportunity to go above and beyond.

The challenge here is to write a Rails application that outputs a paginated list of every number from 1 (one) to 1,000,000 (one million), however the output should be in words. The functionality should allow for the user to choose how many numbers get displayed per page (defaulting to 1000), along with the ability to move forwards and backwards through the numbers.

An example output is as follows:

one

two

...

one hundred and twelve

one hundred and thirteen

...

four hundred and seventeen thousand one hundred and fifty four four hundred and seventeen thousand one hundred and fifty five

...

nine hundred and ninety nine thousand nine hundred and ninety nine

one million

If you can’t complete it: don’t worry! All we want to see is you giving it your best go. Even if you don’t get past one, it gives us a chance to see the thought process behind what you’re trying to do.