

Application Support

Overview

The right candidate for this role will join the CTO and his team in Application Support. You will be responsible for looking after, and developing our back office systems to provide a service that delights our customers.

Responsibilities

- **Business Intelligence:** You will take high level business questions from senior decision makers, and figure out how to answer these questions using our underlying databases. You will then present the answer in an intuitive graphical manner. Where a report is required regularly, you will schedule the report and deliver as required.
- **Billing:** You will be responsible for executing billing, dealing with queries, analysing and applying new call rates, and reconciling the bills from our suppliers.
- **Fraud:** You will work on and enhance our comprehensive fraud detection and prevention systems.
- **Open Source VoIP services** - as our portfolio expands, we will create services on top of open source VoIP tools.

Personal Qualities

- You must be intellectually bright - you are either a graduate, or graduate calibre. This role will require a high level of abstract thought, and the ability to hold many variables in your head at once.
- You must be adaptable: The joy of this company is that everybody, "chips in".
- Capable: This is not a role where you are micromanaged. You will be given tasks, guiding principles, and support as you require. Provided you do what you say, and give early warnings of things not going to plan - you will enjoy this role and received ample recognition.

Knowledge

- **SQL:** This is the cornerstone of the role, and our requirements are for somebody that can take a business question and craft a logical SQL statement. We value easy to understand SQL over and above cryptic but resource efficient SQL.
- **Scripting:** You will require good scripting skills to 'munge' the data into a form that can be acted upon. We prefer Ruby with Perl as an alternative. We know there are lots other great languages out there, but if we don't draw the line, I will use Lisp everywhere - and we don't want that!
- **Unix tools:** you will use grep, comm, duff, sed, awk and others almost constantly to quickly analyse data.

How To Apply

Before you send a CV, please only apply if you can complete the qualification task. If you supply a CV without attempting the task, your CV will not be reviewed.

- Create an excel spreadsheet that reads directly from a MySQL or Postgresql DB using the appropriate ODBC plugin.
- The contents of the DB are entirely up to you.
- If you are searching for inspiration consider a DB with four tables; Customer, Product, Order and OrderStatus
- The Customer table contains an auto-incrementing ID, a name, and a postcode.
- The Product table contains an auto-incrementing ID, a description and a price in pounds and pence.
- The OrderStatus table contains an ID, and a description: For instance id = 10, desc = 'Ordered'; id = 20, desc = 'Paid'; id = 30, desc = 'Dispatched'. Assume an order has to progress logically from status 10 to 20 and finally 30.
- The Order tables contains an auto-incrementing ID, an order ID, a status ID, a date time, a customer id, a product id, and a quantity.
- If a client orders 2 of product 1 and 4 of product 2 - you can see that this would result in 6 entries in the Order table - tied together by having the same order id, and the same customer id. They would differ in the status ID (as the order progresses), and the product ID.
- You might want to provide three tabs on the spreadsheet.
- The 1st tab will provide the raw, normalised data.
- On the 2nd tab, show a pivot table, for instance to count the number of orders at date = X grouped by status.
- On the 3rd tab, manipulate the data to show a graph of revenue earned against time. Hint: assume when an order is ordered, that is when revenue occurs.
- Once you have achieved this - send us the Excel sheet, describe the ODBC plugin you used, and provide an export of your DB (including the schema) so that we can verify.

If you are very keen - consider creating a web front end (for instance in rails) which delivers a working ordering system, with a controller to administer the DB directly, a controller used by the customer to create and check order status, and a controller used by admin staff to progress the order. tar and zip your files, and e-mail those too: This task is entirely optional - and if you have completed the first task, and find this is taking too long - don't wait. Send in the Excel!