

## Short report

# Bespoke automation of medical workforce rostering using Google's free cloud applications

Peter B. M. Thomas

Moorfields Eye Hospital, London, UK

Cite this article: Thomas PBM. Bespoke automation of medical workforce rostering using Google's free cloud applications. *J Innov Health Inform.* 2017;24(4):334–338.

<http://dx.doi.org/10.14236/jhi.v24i4.885>

Copyright © 2017 The Author(s). Published by BCS, The Chartered Institute for IT under Creative Commons license <http://creativecommons.org/licenses/by/4.0/>

### Author address for correspondence:

Peter B. M. Thomas  
Moorfields Eye Hospital  
62 City Road  
London EC1V 2PD, UK  
Email: [pbmthomas@gmail.com](mailto:pbmthomas@gmail.com)

Accepted November 2017

## ABSTRACT

**Background** Providing safe and consistent care requires optimal deployment of medical staff. Ensuring this happens is a significant administrative burden due to complex working patterns. Electronic rostering platforms can help to reduce this burden, but offer limited customisability to the requirements of individual organisations or become costly at scale.

**Objective** To describe a pilot feasibility study of the bespoke automation of medical duty rostering in a busy tertiary Ophthalmology department.

**Methods** A cloud-based web application was created using Google's free cloud services. Users access the system via a website which hosts live rosters, and use electronic forms to submit requests which are automatically handled by Google App Scripts.

**Results** Over a 2-year period (8/2014–6/2016), the system processed 563 leave requests and 300 on-call swaps automatically. 3300 emails and 1000 forms were automatically generated. User satisfaction was 100% (n = 24).

**Discussion** Many time-consuming aspects of roster management were automated with significant time savings to all parties, allowing increased clinical time for doctors involved in administration. Planning for safe staffing levels was supported. The system was customised to fit in with existing administrative processes.

**Keywords:** automation, electronic rostering, cloud, efficiency, workforce

## INTRODUCTION

In order to provide consistent, safe and high-quality care, it is necessary to ensure that medical staff are optimally deployed at all times.<sup>1</sup> There must not only be a sufficient number of doctors covering each service, but they must also be of the appropriate level of experience. Timetabling doctors to ensure these requirements are met is challenging since each will participate in several duty rosters (e.g. emergency cover, ward cover and on-call) in addition to their usual clinic, ward and theatre timetables.

A doctor's overall timetable is therefore a composite of several timetables which describe specific clinical duties, non-clinical duties and leave. The *status quo* in many departments is to keep track of each of these individual timetables in spreadsheets which are often owned by different members of staff.

Approaching administration in this fashion has two significant drawbacks. First, it is inefficient. Storing rosters in separate spreadsheets owned by the same or distinct members of the department imposes a significant time investment in manual crosschecking. This is particularly costly for a department when a doctor is involved in administration as it will reduce departmental income as well as the doctor's clinical performance.<sup>2</sup> Second, it can give rise to unsafe staffing levels. Miscommunication could lead to a doctor being timetabled to two duties at once, or leave being approved when insufficient cover is available. Certain combinations of duty could also lead to illegal working patterns (e.g. under the European Working Time Convention<sup>3</sup>).

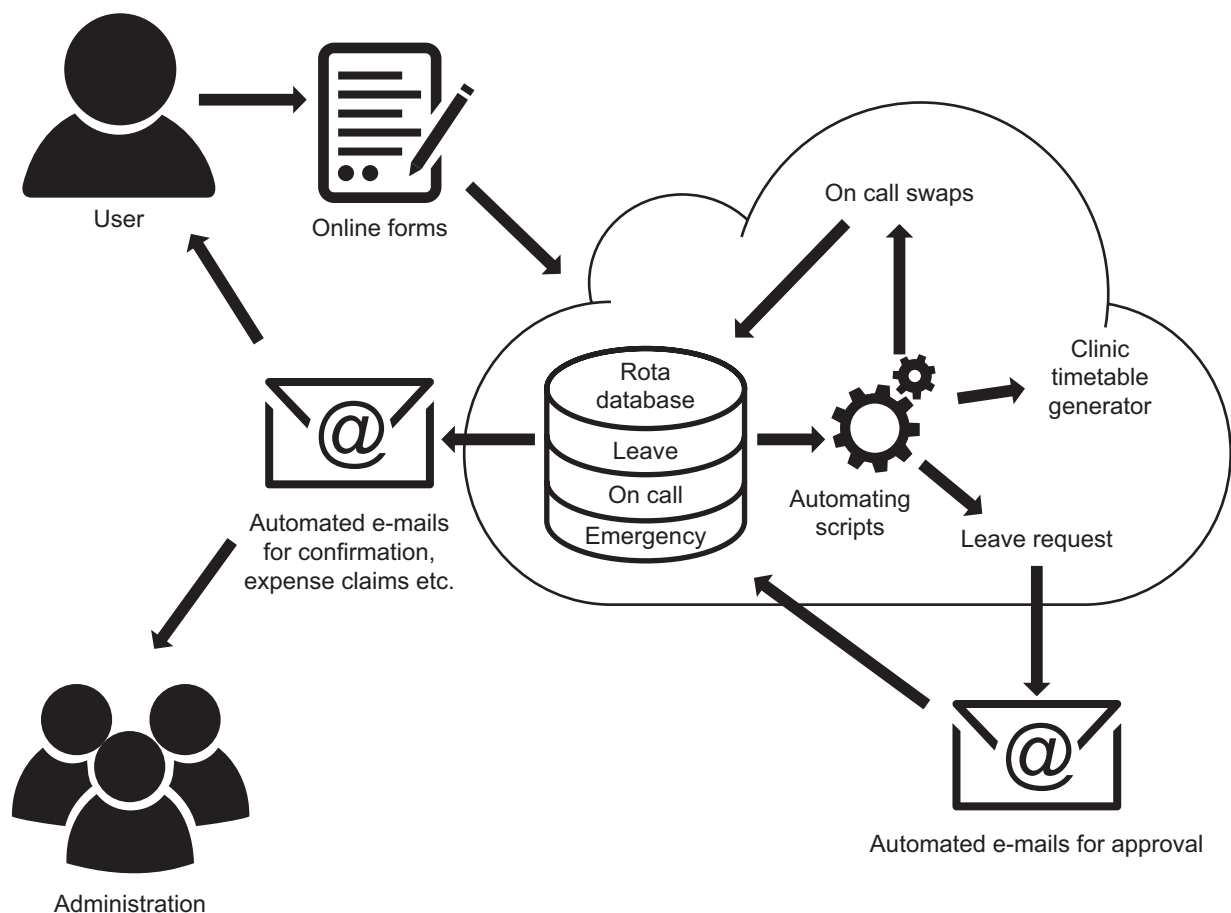
There exist many off-the-shelf electronic rostering systems which aim to streamline staff rostering. Some of these offer

free versions which are often limited to a small number of employees or a reduced feature set. These are excellent options to track employee leave, sickness, pay and similar. However, the needs of a unit within a hospital are often starkly different to other units within the same hospital, so use of a generic system will often fail to alleviate many of the manual administrative tasks needed to run a specialised clinical service. For instance, a valid outpatient clinic timetable (i.e. allocation of staff to particular clinic rooms or services) might need to take into account leave rotas, on-call rotas, theatre rotas and the distribution of staff competency.

More complete automation of medical workforce management requires a more flexible solution that allows rule-based interaction of multiple clinical timetables and rosters, as well as the addition of custom functionality to meet local needs. This paper describes a pilot study to assess the feasibility of a bespoke cloud-hosted administration system that stores all rosters describing a doctor's clinical duties, and automates many of the processes needed to deploy staff optimally. The performance and user satisfaction of this system in a busy tertiary Ophthalmology department over a 2-year period is described. The system described is fully customised to the needs of the department and runs on Google's zero cost cloud architecture.

## METHODS

Figure 1 shows an overview of the system's architecture. Users interact with the system via a website hosted on Google Sites. Their individual rosters (on-call, emergency



**Figure 1** A simplified overview of the web application's structure. The database is simulated via Google Sheet tables

cover and leave) are stored in Google Sheets. Google Forms are used to submit requests into the Google Sheets. Scripts, programmed in Google Apps Script, are triggered by the submission of a new request, and at certain pre-defined times.

### The website

Google Sites host the administration system. Users are covered by individual logins (via Google's own user identification system), and are able to view all rosters and timetables as embedded Google Sheets. Request submission is handled by embedded Google Forms. There are no hosting costs associated with Google Sites. Website generation is achieved via a graphical user interface, and requires no coding.

### The forms

Users submit requests (e.g. leave requests and on-call swaps) via Google Forms which are embedded into the website. Forms are made using a graphical user interface and require no coding. All forms require input of the name of the user or the other party involved in the swap via dropdowns, as well as other information needed to process the request or generate hospital paperwork. When submitted, the request is input as a row into the relevant roster (e.g. the on-call roster for on-call swap requests).

### The rosters

Rosters are held in Google Sheets, one workbook each for the on-call rota, the leave rota, the emergency clinic rota and the clinic timetable generator. The first worksheet in the leave and on-call workbook holds the unprocessed requests from the Google Forms. The other worksheets within a single workbook hold the live timetables and perform some other administrative functions. Scripts are used to ensure that, for example, the up-to-date version of the on-call rota has been copied across from the on-call workbook to the leave workbook as soon as an on-call swap has been performed. The live, processed rota from each workbook is embedded in the website.

### The scripts

All non-static functions of the website are performed by Google App Scripts. These fall into several distinct categories.

- 1) *Request processing.* When a form is submitted, it becomes a row in the relevant workbook. Upon creation of a row, a script is triggered that processes the new request. For example, when a leave request is made the script checks the request against the other activities of the requestor, and either rejects the request (via email), or contacts that person's supervisor for approval. More details are given below in the pathways section.
- 2) *Internal housekeeping.* It is a limitation of Google Sheets that a workbook can only take input from a single form. Therefore, the on-call rota and the leave rota need to be handled by different workbooks. It is therefore important that one workbook can access the up-to-date rota from another. Scripts automatically

push a copy of a workbook's rota (e.g. on-call) to all other workbooks (e.g. emergency clinic and leave) as soon as a change is made to the rota. As a failsafe, all rotas are synchronised by an hourly time-based trigger.

- 3) *E-mail and hospital paperwork generation.* All communication is undertaken by email. The Google App Script allows dynamic generation of emails from an associated Gmail account. When a successful request has been made, all involved parties are informed. Where required, PDF paperwork is also generated to satisfy the broader administrative requirements of the hospital. Google App Scripts allows dynamic PDF generation.

### The pathways automated

- 1) *The on-call swap pathway.* The user requests their desired swap via a form on the site. The form collects the names of the swapping parties, and dates involved. Scripts check both parties are not on leave on the relevant dates by interrogating the leave rota, update the roster if indicated, and email confirmation to both parties of the swap and to the hospital switchboard to maintain their register.
- 2) *The leave requesting pathway.* The user requests leave via a form on the site. The form collects the requestor's name, dates of leave, type of leave (study, annual, professional or other) and any arrangements that have been struck for cover of clinical duties. If study leave is requested, the form cascades to allow entry of details of the activity being attended and options to claims expenses. Scripts check that there are no conflicting duties which must be swapped (on-call or emergency cover), and that leave rules are satisfied (e.g. minimum notice period and staffing levels). The script can generate an email to reject the request if it is problematic, or email the relevant manager for approval if not. The approval email contains a link to approve or reject the leave. If approved, another script modifies the roster, generates the hospital's required paperwork in PDF form, and emails it to the requestor and any relevant administrative staff (e.g. for expense claims).
- 3) *Clinic timetable generation* is handled by a script which runs at certain times to generate a weekly clinic timetable. This assists nursing staff in allocating clinic rooms, and in providing an 'at a glance' guide to where all doctors are at a given time. A master timetable is modified to remove those doctors on leave, and re-allocate those who are on emergency cover from their regular duties.

### User satisfaction

User satisfaction was gathered using an online survey which was sent to all doctors in the department as well as to nursing and administrative staff who used the system. Three questions were asked: 'Are you satisfied with the system?', 'does

it outperform the previous system?' and 'is it better than systems you have encountered in other hospitals?'

### Technical skills involved

The forms and website were generated using graphical user interfaces which need no particular technical skills. Generating rotas and automating the processes required a good understanding of spreadsheets and learning some basic Google Apps Script. Development of the application took 5 days. The programmer (the author) had not previously used Google Apps Script, though had extensively used another scripting language (MATLAB) in the past. Maintenance time was limited to updating staff rosters as new employees joined or old ones left. In a typical month, the maintenance process would take an hour.

## RESULTS

Over the period August 2014 to June 2016, the system automatically handled 563 leave requests from doctors in the Ophthalmology Department at Addenbrooke's Hospital (Cambridge, U.K.). As the validity of leave requests was checked automatically against departmental rules, the rejection rate of leave requests by managers fell to 0.3%. As a result of the low rejection rate, clerical staff could be informed of medical staffing levels at the time of initial request rather than waiting for following leave approval: a mean of 87 days' notice (median 71 days).

There were no instances of staff being given approval for leave when they were on vital clinical duties. Three hundred on-call swaps were automatically processed with no unexpected gaps in the roster. Around 3300 emails were automatically generated and 1000 PDF forms (for central hospital administration) automatically completed.

The online user satisfaction form was completed by 24 members of staff (20 doctors, 2 nurses and 2 administrators who interfaced directly with the system):

- 1) 100% (n = 24) were satisfied with the system.
- 2) 100% (n = 12) preferred it to the previous system (the remaining 12 had joined after the change in system).
- 3) 100% (n = 20) preferred it to any systems they had encountered in other hospitals (the remaining 4 had never worked elsewhere).

## DISCUSSION

This pilot study assessed the feasibility of introducing a bespoke, unified, cloud-based system to manage the clinical and leave commitments of doctors in a large Ophthalmology department. The user satisfaction survey confirms that automation of many aspects of clinical and personal timetabling was well received.

By building a bespoke administrative solution using Google's cloud services, it was possible to tailor the application precisely to the requirements of the individual unit. This offers some obvious advantages over off-the-shelf rostering solutions, as well as allowing for additional functionality. For example, although our unit had no strict requirements for the specific appearance

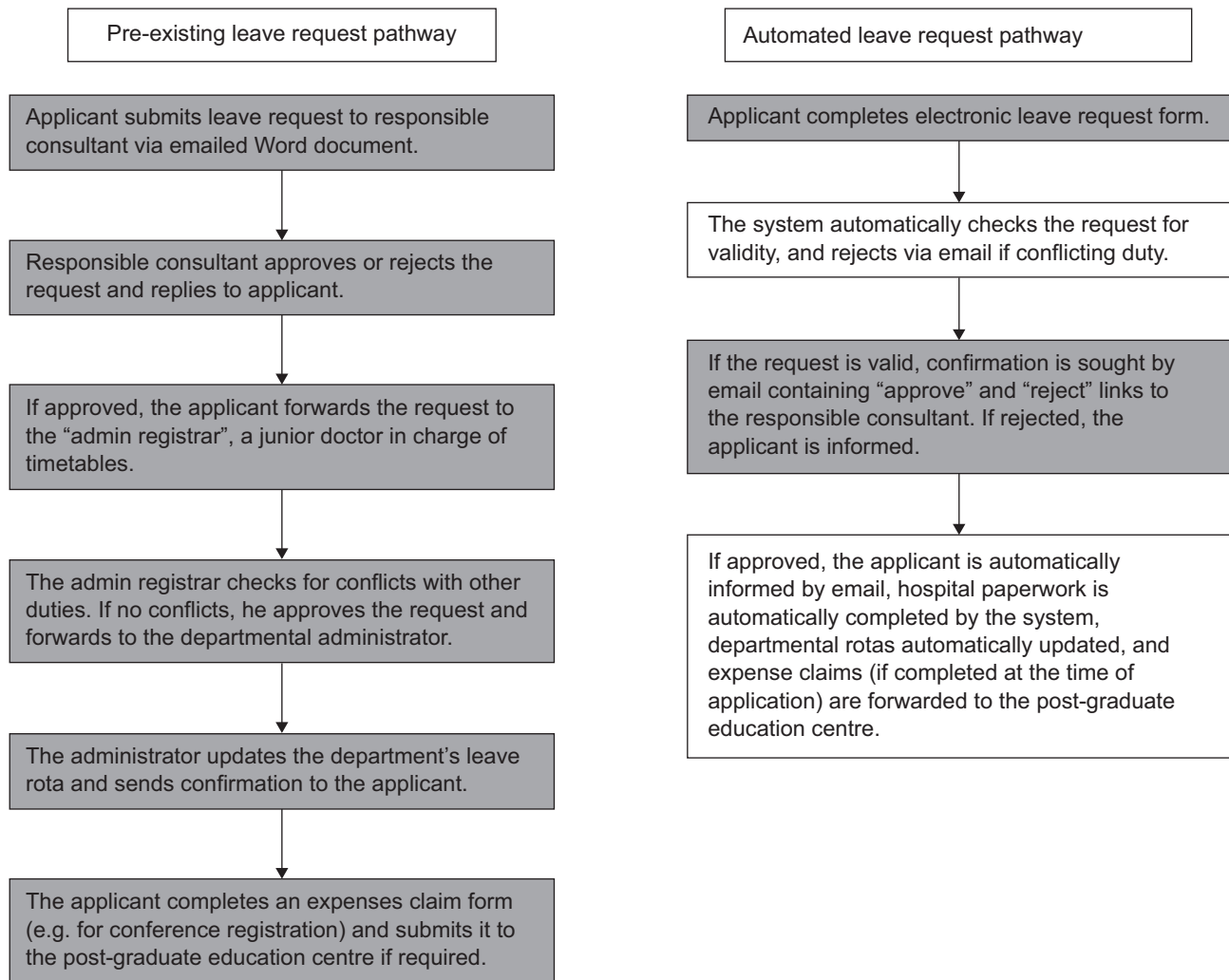
of PDFs generated, the central hospital administration requires leave and expenses requests to be documented on a particular form. Using the PDF editing functions of Google App Script, it was possible for the system to output electronically signed PDFs on the hospitals preferred paperwork. It was also possible to output the clinic timetables (which allocate doctors to particular clinics each day) in the format that the department's administrators and nurses were used to. By using a scripting language to perform validity check on incoming requests (e.g. leave requests), there is no limit to the complexity of rules that can be applied in processing requests or allocating personnel.

Formally, expressing efficiency gains brought about by the system is not possible, since no formal pre-intervention assessment was made. However, it is possible informally to estimate the gains. Figure 2 shows the leave requesting pathway in the department before and after automation. Prior to automation, a valid leave request required six steps to fully process, all of which were carried out by one of four humans (applicant, responsible consultant, departmental administrator and 'admin registrar'). After automation, the pathway was reduced to four steps, only 2 of which required human input (applicant and responsible consultant). Over the 20-month study period, 563 leave requests were submitted. Automation therefore averted 2252 human interventions in leave processing alone, while the remaining human interventions were substantially streamlined. There were also no incidences of inappropriate leave approval where critical clinical duty was already allocated during the study period. Although not formally measured, these incidences did occur prior to automation.

It is possible to identify reasons for the satisfaction among both doctors and administrators. For doctors, the system obviates the need to complete physical forms and chase managers for leave approval, and allows access to live rosters for all clinical duties. Administrative staff are able to plan further ahead due to increased notice of intended leave, and no longer have to perform manual comparison across multiple spreadsheets to work out staffing levels. Examining Figure 2, the 'admin registrar' (whose duties were felt to be excessively burdensome in an open letter from the Royal College of Ophthalmologists<sup>4</sup>) and the departmental administrator particularly benefit by being removed almost entirely from duties in leave approval and on-call swapping, freeing them up for other duties.

A number of challenges were anticipated in transitioning to this system. Technically, a lack of funding meant that it was not possible to use the preferred infrastructure of a cloud hosted website with an SQL database to store the rosters. Using Google Sheets and scripts allowed simulation of a relational database by treating each sheet as a table, and coding queries into scripts. This is an inefficient but workable solution. Google Apps Script has excellent functionality within Google Sheets to perform all other automated tasks. On the human front, there was initial concern that doctors would tolerate the level of automation proposed. Leave rules, for example, are often treated flexibly, but the system will issue an automatic rejection if they are breached. Surprisingly, this rigidity led to little discontent.

Although an in-house solution like this offers extensive tailoring of functionality, the solution is not zero cost overall. In



**Figure 2** The leave request pathway before (left) and after (right) automation. Steps requiring human input are shown with grey shading

this pilot, we used Google's cloud-based applications (Forms, Sheets, Sites and Google App Script) which are available at no cost. However, development time of approximately 5 days was required initially, with a further hour or two per month needed to perform updates such as new staff joining. The technical skills required to set up and program the application are well within those of many members of an IT department. Using free cloud services (in this case, from Google) is a viable and financially attractive approach. However, more complex tasks could easily be automated were the system powered by a true relational database. The current commercial alternative

is an off-the-shelf online rostering solution, though these allow for far less customisation of function.

Overall, this pilot study suggests that a bespoke, cloud hosted, centralised and automated repository of timetables of clinical duties is popular, reduces workload and supports planning for safe staffing levels across multiple clinical services. While the solution described here might not be suitable for all organisations, the study illustrates that automation of administrative functions (significantly beyond those supported by off-the-shelf offerings) can be achieved by bespoke solutions commissioned at the departmental level.

## REFERENCES

1. dos Santos MAFRN and Eriksson HKO. Insights into physician scheduling: a case study of public hospital departments in Sweden. *International Journal of Health Care Quality Assurance* 2014;27(2):76–90. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/24745134>. Accessed 23 June 2016.
2. Weigl M, Müller A, Zupanc A and Angerer P. Participant observation of time allocation, direct patient contact and simultaneous activities in hospital physicians. *BMC Health Service Research* 2009;9:110. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/19563625>. Accessed 23 June 2016.
3. Directive 2003/88/EC of the European Parliament and of the Council of 4 November 2003 concerning certain aspects of the organisation of working time. *Official Journal L European Commission* 2003:0009–19. Available from: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32003L0088:EN:HTML>. Accessed 23 June 2016.
4. Royal College of Ophthalmologists. Re: 'Admin Reg' or equivalent Roles [letter]. 2014. Available from: <https://www.rcophth.ac.uk/wp-content/uploads/2014/07/140722-Letter-to-Clinical-Leads-regarding-Admin-StR-role.pdf>. Accessed 23 June 2016.