The redesign of an existing intranet system and its effect on usability and user experience

Sid Jotsingani

11th August 2014

Abstract

My project is to improve an existing intranet system by adding new and innovative functionalities to provide solutions to existing problems. The goal is to improve overall usability and to create a better user experience in the hope of improving internal business practices by employees of the company. This paper covers my proposed changes to their system which include a new profile management concept, a mapping application to provide a visual overview of users locations, a system to submit applications and convert the data to graphs, a yet to be completed attempt at retrieving academic research papers, and a redesigned theme for the intranet system along with my reasoning behind design decisions. Testing in the future would need to be done to determine whether the changes I propose have any impact on the usability or user experience of the system.

Acknowledgements

I would like to thank the following people for the continuous support throughout this project:

Dr. Sulo Shanmuganathan – Industry Supervisor.
Opus International Consultants Ltd.

Dr. Xinfeng Ye – Academic Supervisor.
The University of Auckland

Dr. Sathiamoorthy Manoharan – Information Technology Coordinator.
The University of Auckland

Introduction

Opus International Consultants is a multi-disciplinary infrastructure consultancy firm with over 3,000 employees in over five countries. They are focused in 8 major sectors such as buildings, transport, water, energy and more. Opus deals directly with clients and most of the projects are completed off-site at various locations depending on the client's requirements.

The current intranet system is used to manage profiles of all the employees, to search and find contact information of other employees and to view internal documents from inside the company. However the usability of the system is quite low, and performing what are meant to be trivial tasks such as editing one's profile can turn out to be a very complex task. Along with the complexity of the system, it is missing some pieces of functionality that the company wants implemented in order to automate and simplify some tasks that are normally performed manually by employee.

An issue that this project addresses is that user's do not keep their profiles maintained, including keeping their current location around the world up to date. This is quite an important aspect which is needed in a firm where users are constantly travelling between different sites and locations. In order to improve this practice, the design of the intranet needs to be improved without being too complex of a change where users have to re-learn the system.

Research needs to be done on how usability can be improved in the current intranet system without increasing overall complexity, and to understand what functional and design decisions need to be made in order to enhance user experience in the current intranet system?

A quick note about the project

My supervisor and peers at Opus have been very supportive of my work so far. One of the issues that arose earlier in the process of this project is that there currently is (or soon to happen) a migration going on from Opus's current intranet system to a SharePoint solution. What this meant for me is that my academic supervisor and I had to discuss whether I am developing for the older system which only has a limited lifespan left or whether I am to wait and develop for the newer SharePoint system. However since then, the IT team has kept any changes that need to be done to the intranet on hold and I no longer have an idea or timeframe as to when the migration is to happen. However due to this being an academic project we had to keep development moving and so any work done so far has had to be done locally on my own webhost and Apache server and not on Opus's actual systems.

This may require changes later in the future (possible significant changes) when migrating – but at least the fundamental aspects of the logic and design will remain the same. I also acknowledge that any of the work I do may not be possible to migrate on the Opus's servers anytime soon (assuming if it even happens this year), but it does give us something to show to the IT team about what we are able to improve on (since they are not in any hurry) and can hopefully convince them to pursue or to allow us to work on their current intranet systems for them. Along with all this – I have still not been able to meet the IT team in Wellington yet and nor have I been able to look at how the backend of the intranet is actually working, what the infrastructure is like and have a confirmation about how the backend of the intranet is structured or what it runs on.

Despite the fact a lot of the stuff I have developed may need to be modified and migrated so it will work on the newer SharePoint system, the project is still and will continue to move forward (at least locally).

Related Work

The goal of this project is to improve the usability in the system *and* to enhance user experience of the intranet system. Usability and user experience are often used interchangeably, however there is a distinction between these two terms. (Bevan, 2009) writes that a survey at Nokia stated that User Experience was perceived as being usability plus anticipation and the pleasure from the usage.

(McNamara & Kirakowski, 2006) state the three areas of concern in Human Computer Interaction are Functionality, Usability and User Experience. They define usability as a *"characteristic of the interaction between the user and the product"* and state usability answers the question "*can I make the product do what I want it to do*"? This differs from their definition of user experience which is "*the wider relationship between the product and the user in order to investigate the individual's* *personal experience of using it*". User experience should answer "*how the person felt about the experience, what it meant to them, whether it was important to them, and whether it sat comfortably with their other values and goals".*

Both Bevan and McNamara & Kwiatkowski agree that user experience links with usability plus the *emotional response* attached after using the product. This is also quite similar to what *Domain7* – a group of designers from around the world, say about usability and user experience. They state usability plus user experience are both essential to the success of a website/application. Usability is about task-based interactions and is allowing something to be done easily and intuitively whereas user experience is how a person feels when they interact with your product and their emotional connection to the task. What this means in the context of the project, is just because the website is easy to use – it doesn't mean it that users are enjoying using the site or are having a good experience. The intranet must be easy and simple to use (usable), but must be designed in such a way that it is delightful and a pleasure to interact with, and that there are no aspects that give the user frustration or inconvenience.

One of the most known authors in the world of Human Computer Interaction is Jakob Nielsen. In his research, (Nielsen, Usability 101: Introduction to Usability, 2003) define usability as "*a quality attribute that assesses how easy user interfaces are to use"*. In the same paper he also states that for intranet systems usability relates to employee productivity. So if an employee has to spend time deciphering their next actions, then that time wasted is money lost by the company without any work or productivity accomplished. (Nielsen & Molich, Heuristic evaluation of user interfaces, 1990) also list out 10 usability heuristics for interface design. These heuristics are meant to solve common problems that people often face in interaction design. They are:

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

The proposed work, completed till date will be using Neilson's heuristics as guidelines for some design decisions.

(Hinchliffe & Mummery, 2008) performed a similar task of attempting to improve a health promotion website – where the main goal was improving usability. They broke down their usability improvements into 6 different categories. The Design which was about changing actual visual elements; the feedback which was the response the system gave; the format such as entering dates in the same format consistently throughout the website; The instructions given to the user; The navigation and how easy it was for a user to get back to where they came from; And the terminology. Upon improving these 6 categories they found an overall improvement in Usability. There are again similarities between the changes performed by Hinchliffe & Mummery and Nielsen's Heuristics. The instructions representing the help and documentation, the terminology relating to the match between the system and real world, the navigation relating to the visibility of system status and so forth. This again indicates that Nielsen's heuristics are in fact a reliable method to asses and increase usability of a system and that usability can be improved on any website by following a set of guidelines.

A recent book by (Lal, 2013) states that the best interfaces are those which have a minimum design, simplicity, accessibility, consistency, feedback, forgiveness and are user driven. This also matches with what Nielsen also states in his heuristics, 10 years later. The core concepts appear to still be the same, what seems to be changing is how designs are expressed and our perception of what consists of good design.

Figure 1, on the next page shows the top most English websites sites on *Alexa.com* (Alexa, 2014). After looking at each of these websites in detail, I aggregated a list of design patterns that were similar between most of the sites. This list shows showed some common design trends that are slowly emerging from my observations:

- The use of white and grey to distinguish foreground and background. Most of the sites are using grey as either the body background/navigation bar colour, and using a white background for the main content of the site.
- “Card styled” layouts using squares and rectangles filled with the main content. These cards generally have sharp corners, thin margins between other content and contain a combination of text and images.
- Shadows to give depth to the main content. Despite moving away from gradients and using single “flat” colours, most of the areas that contained content (cards) had shadows to make them appear as if they are sitting “on-top” of the background.
- Highlights/outlines on input boxes when focused onto them. This is a form of feedback when clicking inside a textbox to show which box you are typing in.
- Larger than the default sized input boxes with large fonts. Most of the input boxes had modified CSS to make them larger than the HTML default size, along with larger font than the size of the text used in the body content and paragraphs.
- Some form of separation with the logo and search, and the rest of the content. The logos mostly sat in the top left corner with the navigation horizontally across or vertically down on the left hand side in some cases. There would either be some break in colour, border lines, or by padding to break where the navigation ends to where the content starts.
- Consistent colour schemes and styling throughout webpages. Each website had their own set of colours they would use throughout on multiple elements. The layout would also be consistent throughout pages. An exception was normally the homepage which may have had a different layout – but which still used the same colours and styles.

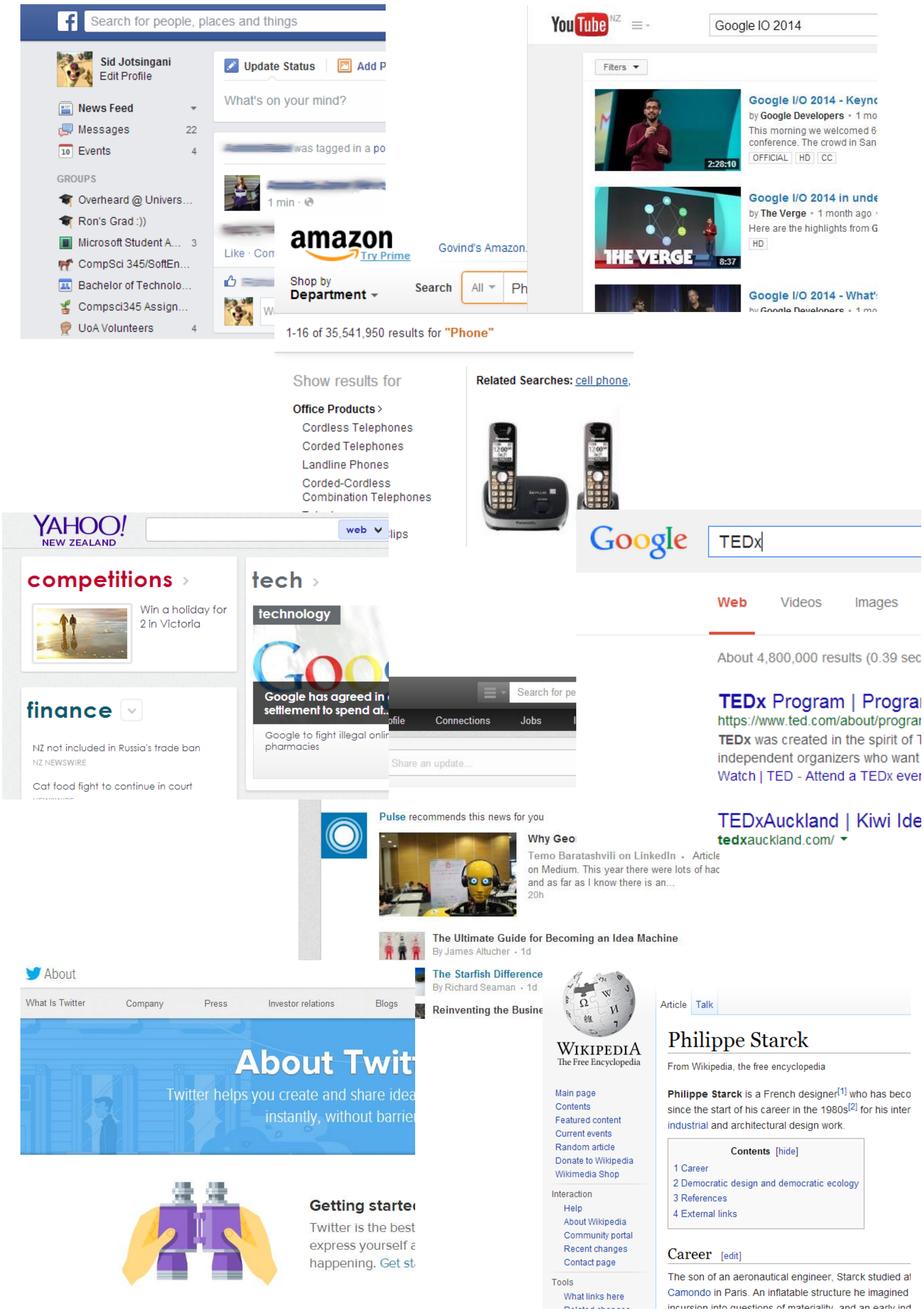


Figure 1, Top Alexa.com websites from left to right: Facebook.com, Youtube.com, Amazon.com, Yahoo.com, Google.com, LinkedIn.com, Twitter.com, Wikipedia.org

On most of the researched websites, an emerging trend is the use of the "card-style" layout where the overall design is flat – yet depth is perceived. This combination of a card layout along with flatness is not only emerging in web technologies, but is also advancing in mobile design too. Apple overhauled their layout in 2013 with the release of iOS7 to a flattened layout – yet added depth using parallax. Windows 8 and Windows Phone 8 use a tiled layout with and have adapted a flatter style of icons too. Whilst Android has been using flat card style layouts for some time now, and recently with the announcement of "*Android L*" has changed the design to "*material design*" which follows the idea of cards with depth and layers (again using shadows).

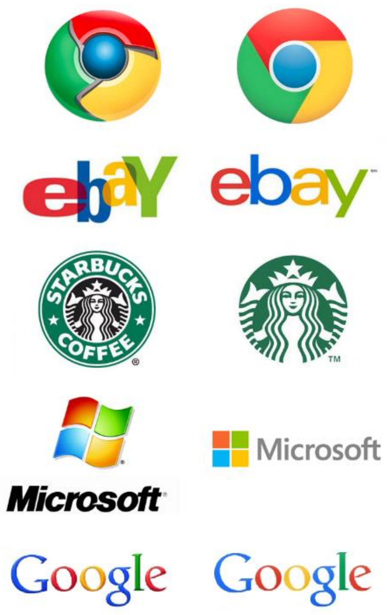


Figure 2, A change in design trends to flat logos

This increase in popularity in flat design contrasts skeuomorphic design which came into play when the term *Web 2.0 (*and understanding what it was about) gained popularity in 2004 (O'Reilly, 2005). Until about 2010 iconography was focused on skeuomorphism, meaning making icons look as close as they can to their real life counterparts. This meant the use of gradients, 3D perspectives and excessive use of shadows to give it real life characteristics.

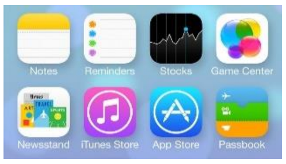


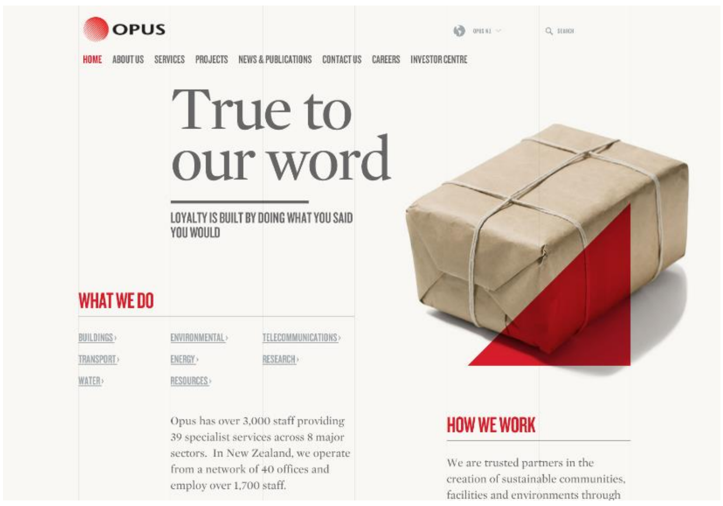
Figure 3, iOS7 has a flat design, yet uses shadows behind flat icons in order to perceive depth

As shown in *Figure 2* design has changed since then to now become flatter. And ever since more and more people and firms have been adapting to this new flat design style which even today is still gaining in popularity. However as (Turner, 2014) points out, one of the biggest drawback of flat design has to do with icons losing meaning, and that flat icons often do not offer feedback when clicked or hovered upon. From a usability perspective, this lack of feedback is a step backwards. Neilson states there should be visibility of system status – which includes providing feedback on a performed action in a reasonable amount of time. It also violates one of Shneiderman's8 golden rules of principle design which states an interface should offer informative feedback (Shneiderman & Plaisant, 2010). So despite having a flat design – some type of feedback should be given to a user, even when using flat icons in order to maintain a certain level of usability. This is a balance that needs to be achieved throughout the entire design process.

(Pozin, 2014) from Forbes.com mentions that the fact that responsive web design is the new emerging trend of web design. He says we need to embrace higher resolutions, so from a web design perspective we need to make our websites more responsive and fluid. Long gone are the days of fixed size. But along with responsive web design he mentions about how the other biggest change is that design is becoming flatter and is something we need to adapt to.

The Design & Theme

When deciding what design elements should be changed for the current intranet system, I decided to have a look at what Opus's current public facing website looks like. One of the first things which stood out was the colours used. The content background was a light shade of grey, with text a darker grey colour sitting on top. The text for any important information (main headings and the current navigation item) was all uppercase, and in a bold red font. Other headings (sub-headings, and other navigation items) used the same font but in a darker grey colour. The use of red was what was standing amongst the shades of grey, and that was being used for any important information.



True to our word

LOYALTY IS BUILT BY DOING WHAT YOU SAID YOU WOULD

WHAT WE DO

BUILDINGS › ENVIRONMENTAL › TELECOMMUNICATIONS ›
TRANSPORT › ENERGY › RESEARCH ›
WATER › RESOURCES ›

Opus has over 3,000 staff providing 39 specialist services across 8 major sectors. In New Zealand, we operate from a network of 40 offices and employ over 1,700 staff.

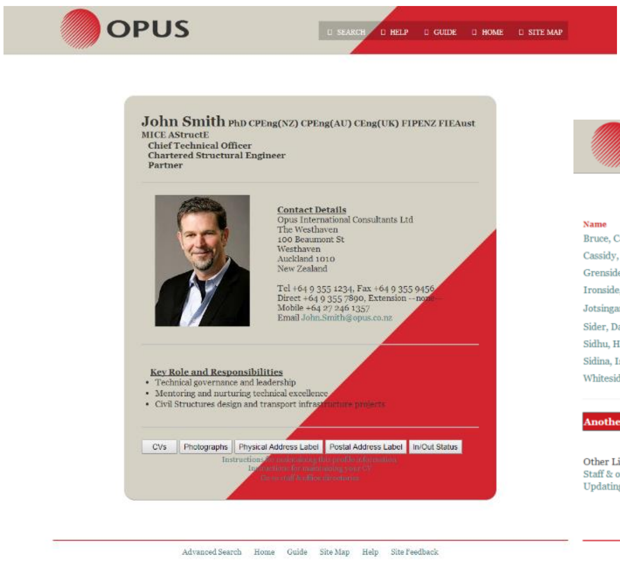
HOW WE WORK

We are trusted partners in the creation of sustainable communities, facilities and environments through

Figure 4, A screenshot of the homepage from Opus's public website

Apart from the colour scheme, the logo was placed in the top right corner with a horizontal navigation bar below it (which fits in with other common design trends). The text used throughout was a large serif font, and the design was overall very flat. The background colour was not a standard white as most websites use, but instead a lighter shade of grey – (which I believe still works quite well since the text still has a large contrast with the background colour). Overall the website was very pleasing to look at. When this is compared to a page from the current intranet system, the differences immediately become visible.

The two screenshots below are from the firms current intranet system. The first is a screenshot from the profile page of another user, the second being the results of a search for an employee in the company. The first noticeable difference is the lack of consistency between the two pages itself – despite being a part of the same intranet. The logo and header has remained the same, however the navigation bar has changed completely upon searching for a new user. Secondly the layout of the main content has changed quite drastically as well in terms of width. Where in the first screenshot the content is centered inside a beige box that has a fixed size, in the second screenshot the content takes up almost the full size of my screen (despite still being a fixed size), and is placed inside the background itself – rather than inside a content box like on the profile page. Both these pages had layouts of completely different widths – one looked portrait, whereas the order looked landscape. One of Nielsen's principles was about consistency and standards. This is something that will need to be addressed in my implementation ensuring that there is consistency between the pages. Secondly Nielsen also mentions about aesthetics and a minimalist design. Compared to the website counterpart, the intranet interface looks less aesthetically pleasing and there is a lot of crammed content especially on the profile page where there is a lot of information crammed by the name.



John Smith PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE
Chief Technical Officer
Chartered Structural Engineer
Partner

Contact Details
Opus International Consultants Ltd
The Westhaven
100 Beaumont St
Westhaven
Auckland 1010
New Zealand

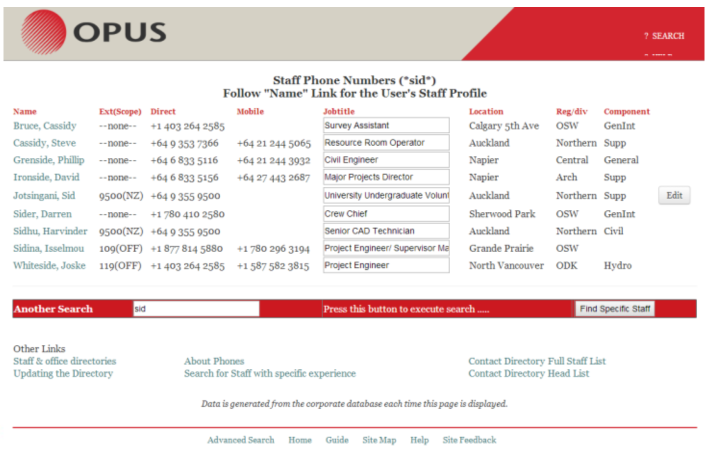
Tel +64 9 355 1234, Fax +64 9 355 9456
Direct +64 9 355 7890, Extension --none--
Mobile +64 27 246 1357
Email John.Smith@opus.co.nz

Key Role and Responsibilities
- Technical governance and leadership
- Mentoring and nurturing technical excellence
- Civil Structures design and transport infrastructure projects

Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Advanced Search Home Guide Site Map Help Site Feedback

Figure 5, A screenshot from the profile page from Opus's current Intranet system



Staff Phone Numbers (*sid*)
Follow "Name" Link for the User's Staff Profile

Name	Ext(Scope)	Direct	Mobile	Jobtitle	Location	Reg/div	Component	
Bruce, Cassidy	--none--	+1 403 264 2585		Survey Assistant	Calgary 5th Ave	OSW	GenInt	
Cassidy, Steve	--none--	+64 9 353 7366	+64 21 244 5065	Resource Room Operator	Auckland	Northern	Supp	
Grenside, Phillip	--none--	+64 6 833 5116	+64 21 244 3932	Civil Engineer	Napier	Central	General	
Ironside, David	--none--	+64 6 833 5156	+64 27 443 2687	Major Projects Director	Napier	Arch	Supp	
Jotsingani, Sid	9500(NZ)	+64 9 355 9500		University Undergraduate Volunt	Auckland	Northern	Supp	Edit
Sider, Darren	--none--	+1 780 410 2580		Crew Chief	Sherwood Park	OSW	GenInt	
Sidhu, Harvinder	9500(NZ)	+64 9 355 9500		Senior CAD Technician	Auckland	Northern	Civil	
Sidina, Isselmou	109(OFF)	+1 877 814 5880	+1 780 296 3194	Project Engineer/ Supervisor Ma	Grande Prairie	OSW		
Whiteside, Joske	119(OFF)	+1 403 264 2585	+1 587 582 3815	Project Engineer	North Vancouver	ODK	Hydro	

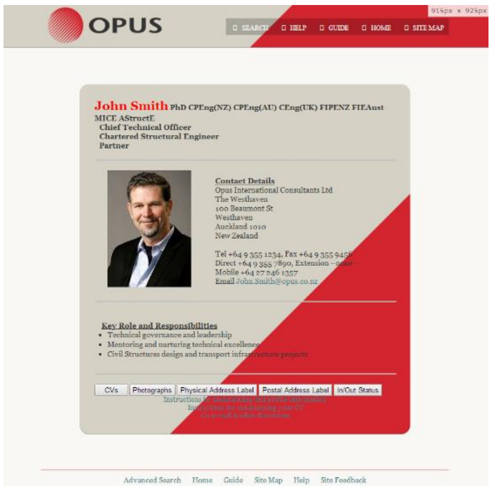
Another Search sid Press this button to execute search ..... Find Specific Staff

Other Links
Staff & office directories About Phones Contact Directory Full Staff List
Updating the Directory Search for Staff with specific experience Contact Directory Head List

Data is generated from the corporate database each time this page is displayed.

Advanced Search Home Guide Site Map Help Site Feedback

Figure 6, A screenshot of search results from Opus's Intranet



John Smith PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE
Chief Technical Officer
Chartered Structural Engineer
Partner

Contact Details
Opus International Consultants Ltd
The Westhaven
100 Beaumont St
Westhaven
Auckland 1010
New Zealand

Tel +64 9 355 1234, Fax +64 9 355 9456
Direct +64 9 355 7890, Extension --none--
Mobile +64 27 246 1357
Email John.Smith@opus.co.nz

Key Role and Responsibilities
- Technical governance and leadership
- Mentoring and nurturing technical excellence
- Civil Structures design and transport infrastructure projects

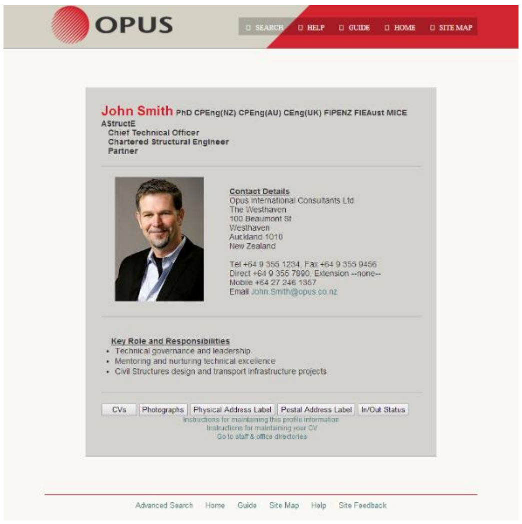
Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Figure 5, The first modification to the existing intranet including a grey background and bold red headings

One of the design choices made during the design process of the intranet, was to most importantly bring consistency between pages. Not only that, I also wanted to bring consistency between the intranet and Opus's website so that employees can recognize the system easier and possibly have a better understanding of it. This meant bringing certain elements over to the intranet system such as bold red headings, a light grey background and a minimalist navigation system.

In order to do this the colours used on the website were sampled and then used on the intranet page. The heading colour was now a darker red rather than black, and the background was also a lighter shade of grey. As you see on the right how the design changes – it is barely noticeable, but it was a start. These changes can be seen on the left.

The next biggest change came when redesigning the content box. First of all the background colour was changed from beige to a darker shade of grey to better match the website background colour. The edges of the boxes and input buttons were squared off rather than being rounded. Sharp edges were more common than rounded corners in recent design patterns (such as Google card layouts, Yahoo news categories, Facebook news feed items, and LinkedIn updates all using square rather than rounded corners). The square corners also seemed to be a better match fit due to the fact the Opus triangle (which appears on most pages) has sharp corners and not round ones. The background image in the content box of the red triangle was removed (temporarily). The font was also changed to a much more legible and common, sans-serif font, Arial rather the previous serif font, Georgia. The use of Arial does not follow what's on the Opus website, but was chosen because of its easy readability at small sizes, simple design and it being acknowledged as a CSS safe web font to use for web design (W3Schools). The screenshot on the right shows the changes after this.



John Smith PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE
Chief Technical Officer
Chartered Structural Engineer
Partner

Contact Details
Opus International Consultants Ltd
The Westhaven
100 Beaumont St
Westhaven
Auckland 1010
New Zealand

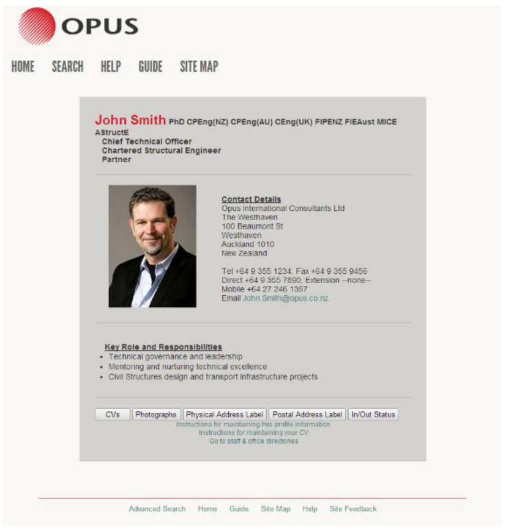
Tel +64 9 355 1234, Fax +64 9 355 9456
Direct +64 9 355 7890, Extension --none--
Mobile +64 27 246 1357
Email John.Smith@opus.co.nz

Key Role and Responsibilities
- Technical governance and leadership
- Mentoring and nurturing technical excellence
- Civil Structures design and transport infrastructure projects

Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Figure 6, The next modification including sharper corners, new colours and a sans-serif font

As you can see on the left, the navigation bar was then changed to better match that of the Opus website. A similar horizontal layout was used. The reason for this was to enable users to recognise the navigation layout that they may be familiar with from the Opus website, to bring consistency between the intranet and website, and to better suit the common design pattern of keeping the navigation aligned to the left rather to the right. The font



John Smith PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE
Chief Technical Officer
Chartered Structural Engineer
Partner

Contact Details
Opus International Consultants Ltd
The Westhaven
100 Beaumont St
Westhaven
Auckland 1010
New Zealand

Tel +64 9 355 1234, Fax +64 9 355 9456
Direct +64 9 355 7890, Extension --none--
Mobile +64 27 246 1357
Email John.Smith@opus.co.nz

Key Role and Responsibilities
- Technical governance and leadership
- Mentoring and nurturing technical excellence
- Civil Structures design and transport infrastructure projects

Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Figure 7, The third modification cycle consisting of a changed navigation bar used for the navigation is not the same as the Opus website, since the font used there was a premium font called "Knockout". Instead a free font called "League Gothic Regular" was used. In the future, this font can be replaced with "Knockout" if permission is given by Opus to use it. Instantly after these changes, the layout seemed to better match the Opus website. However even at this point it felt like some things were lacking and could still be improved upon.

The next changes that were made were made based upon looking at the other websites above, and the original intranet page rather than the website. This was to bring more design elements that were seen as contemporary whilst at the same time try and retain the existing look and feel of the original intranet page so it did not seem like a completely overhauled design change in the system that may frighten some users.

The design was now flatter and was better suited to modern design patterns. However with most of the top sites, some form of shadow was still being used to distinguish foreground from background. A box shadow was added around the content box, to give it depth as if it were sitting on top of the background. The width of the content box and some elements on the page were also modified to be more fluid and to change according to how a website scales for different sized screens. The logo and navigation was aligned with the left side of the content. Also the layout of elements. The background of the content-box was also changed to be transparent, the red Opus triangle was also added back which was made using pure CSS rather than an image with it's z-index set behind the content box. This also helped give the effect of depth using a flat design. The image on the right shows the template minus any content (that will vary per page) for my proposed solution. To bring consistency this will be used on every page filled with content inside the grey area. The style also follows the card-style layout used on most of the top sites of the web. The figure on the left shows the layout of the profile page using the new theme (Note: the layout of the content for content box has changed compared to the original web page).

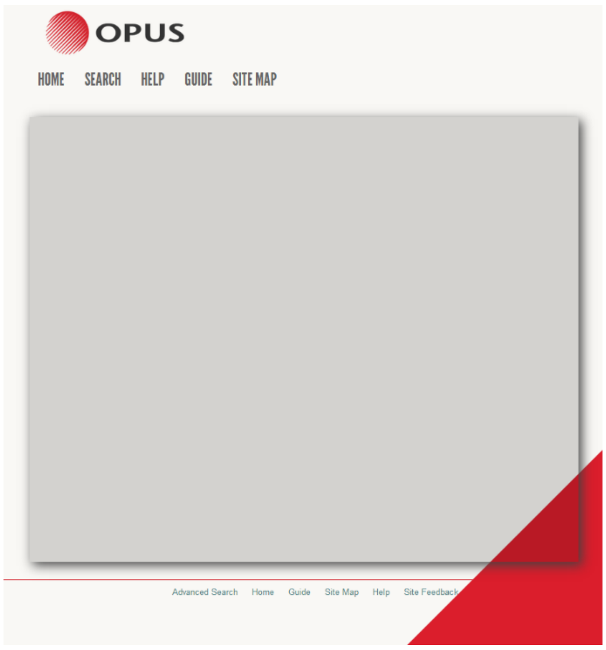
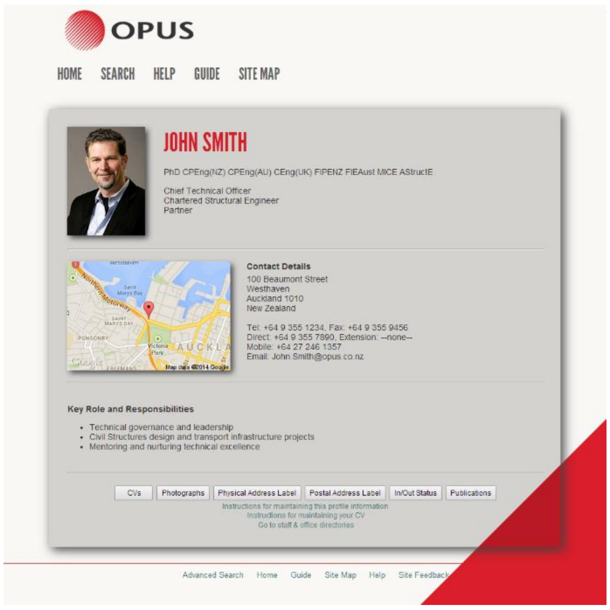


Figure 9, The proposed theme with no content

The design here also compromises some features which are not directly visible in a static screenshot that have been embedded within the theme such as a lightbox (pop up window inside the current page) when needed, and the use of some JQuery animations as an attempt to enhance user experience by making the overall feeling of using the site "fun" and making the experience a joyful one.



JOHN SMITH

PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE

Chief Technical Officer
Chartered Structural Engineer
Partner

Contact Details
100 Beaumont Street
Westhaven
Auckland 1010
New Zealand

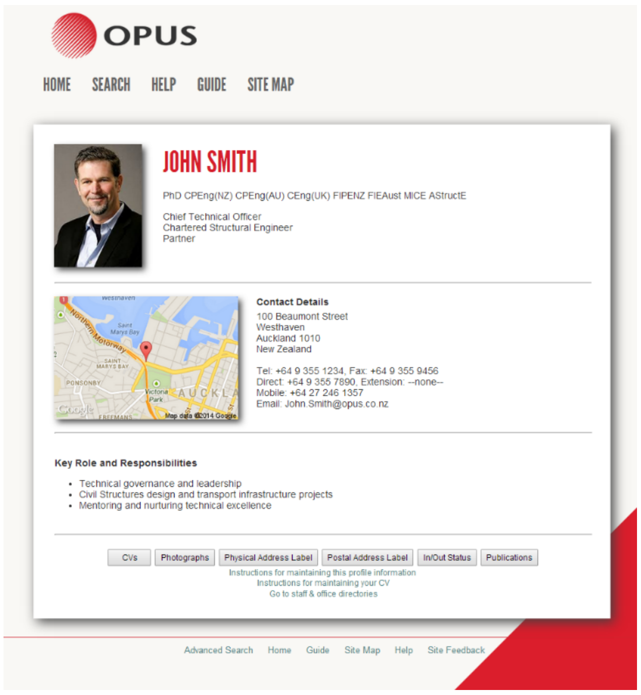
Tel: +64 9 355 1234, Fax: +64 9 355 9456
Direct: +64 9 355 7890, Extension: --none--
Mobile: +64 27 246 1357
Email: John.Smith@opus.co.nz

Key Role and Responsibilities

- Technical governance and leadership
- Civil Structures design and transport infrastructure projects
- Mentoring and nurturing technical excellence

Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Figure 8, The profile page with the new proposed theme



JOHN SMITH

PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE

Chief Technical Officer
Chartered Structural Engineer
Partner

Contact Details
100 Beaumont Street
Westhaven
Auckland 1010
New Zealand

Tel: +64 9 355 1234, Fax: +64 9 355 9456
Direct: +64 9 355 7890, Extension: --none--
Mobile: +64 27 246 1357
Email: John.Smith@opus.co.nz

Key Role and Responsibilities

- Technical governance and leadership
- Civil Structures design and transport infrastructure projects
- Mentoring and nurturing technical excellence

Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Figure 10, An alternative to the current theme where the background colour of the content box has changed to white

One of the major differences between my proposed design above and common design patterns, is that the background I am using for the main content is not white. Most of the top websites use a white background for the content followed by a grey colour for the rest of the background. The left figure shows an alternate version of the theme using a white background for the content as opposed to a dark grey one. The reason for the dark grey was used, was to better fit the Opus website color-scheme that did not contain any white at all for any element. Secondly having the darker background for content and lighter background for the body was a closer match to that of the original opus intranet page, so it did not look like a significant difference. I do not know which of the designs will be perceived as more usable or which may provide a better user experience, so I may have to conduct some further research or testing on users in order to deduce whether the white or grey background looks better and whether the new design has improved usability and user experience overall.

I have attempted to cover as many of Neilsens Heuristics as I could in my proposed design:

- Showing the system status at any given point by giving feedback on actions such as hover on links, adding loading bars for certain events and the use of modal messages upon errors or successful actions *(See Implementation section on Technical Funding Applications)*.
- Matching between the system and the real world was something I could only do to a certain extent. However this seemed to be quite fine to begin with. I am using the same text that was on the original intranet pages and so I had no control over the jargon.– However when it came to me using natural language rather than jargon, any messages I was showing/display to the user are using simple non-technical terms. In the future though, I may be looking at adding icons to each of the navigation bar items and address this issue in further detail.
- User control and freedom in the form of the ability to reverse an action such as close a lightbox when not needed, and the ability to modify content on the page itself. *(See Implementation section on Profile Management)*
- Consistency and standards is being addressed by creating a single theme that is to be used amongst different pages, along with keeping some design aspects of the intranet similar to the website. Modern standards have also been used throughout such as the use of shadows.
- Error prevention by red modal boxes containing the error in a simplified language upon an incorrect action such as uploading an image file as opposed to a PDF *(See Implementation section on Technical Funding Applications).*
- Recognition rather than recall – most of the common actions have been kept in the same location between pages (navigation and footer). However in future iterations of the design, a more prominent and improved help system could be implemented to help new users understand the system better.

- Flexibility and efficiency of use - there are no shortcuts implemented for experienced users. Both novice and experienced users go through the same interface to accomplish tasks. In the future some functionality could be added in order to assist more advanced users.
- Aesthetic and minimalist design – the proposed design appears to have been simplified down from the original, however there is always room for improvement. There are very little frills (such as the Opus triangle), apart from the core elements on the page.
- Help users recognize, diagnose, and recover from errors – all error messages give a clear indication of what the user did wrong and how to fix the error. Also any errors in inputting data are immediately highlighted in a red colour around the border of the input box.
- Help and documentation – currently only the help section which is accessed in the footer. In future iterations of the design a more prominent and improved help system could be implemented on a page-by-page basis to help new users understand the system better.

When I started looking into usability and design, what I kept reading about responsive web design and the enormous advantages of creating websites for responsive design. All the articles and journals contained information about the responsive web. However after reading some papers – I soon learnt that fluid web design and adaptive web design are also have some similarities to responsive web design. Although there is a fine difference between them – I wasn't really sure which one I was designing for.

As (Carlos, 2013) states in his book *Responsive Web Design with jQuery*, the difference between responsive layouts and. Fluid Design is adjusting the dimensions of elements but doesn't vary the layout structure – you generally use percentages for all the elements. Adaptive design changes the layout for different resolutions – but its more like having a bunch of static layouts and is not fluid (Davison). Finally responsive web is having fluid layouts while also changing the layout structure for devices with different resolutions. An example for responsive is where all the content moves to a more vertical layout on mobile phones (since phones are larger vertically), and horizontally on traditional monitors (since displays are larger horizontally). (Campbell & Shelly, 2014) raise a good point about static layouts – that static layouts are always consistent no matter what the resolution is. However the drawback here is that for screens with extremely small or large resolutions it may appear to small or too big.

Now the question was what was I designing for? The current Opus layout was a static layout that did not respond to any changes in resolution. The layout style I had begun working on and will continue in the future is a fluid layout. My justification behind this choice is that adaptive layouts appear to be less responsive to changes compared to fluid and responsive layouts since they do not respond to changes as well as those, whilst static layouts do not respond at all. Although a responsive layout consists of a form of fluidity layout plus changes in structure (and is therefore generally seen as being the better choice) – in the scenario of this project – the Intranet (at least currently) will only ever be accessed on desktop/laptop devices with larger screen resolutions. This is because connectivity to the intranet is only possible through a wired Ethernet connection – that is only available on desktop/laptops or larger tablets. Responsive layouts are ideal for scaling a website down for mobile browsing, which isn't needed in this case in this scenario therefore a good fluid layout seems to be the best choice. Nevertheless in the future if a responsive layout is needed – one can always adapt the fluid layout into a responsive over time.

The updated styling I am using throughout all the webpages has a larger size for content than the original did. The main content box width has increased from 625px to 880px – however for devices with not as much screen width, the content box size decreases down to a minimum of 625px. This feature can be emulated by re-sizing the window down to a smaller content size. This fluidness was achieved using the CSS2 property of max-width.

This same principle was applied to many different elements, such as the profile picture on the page in order to achieve a low level of fluidness and to also ensure these elements have an upper limit so they do not scale too largely.

The Implementation and Interactive Design

Throughout the process from ideas to prototypes has all been done using an agile methodology. I would go to meet my industry supervisor almost every week to find out what was needed. Upon gather that information I would search for ways to accomplish what my supervisor wants, and try and find solutions to those problems myself. I did not create many mockups or sketches. Instead I would make something that I felt best fitted what my supervisor wanted and would create a quick semi/fully-functional prototype of it. During my next week meeting I would then show this prototype to my supervisor and gain feedback on how to improve it. This process repeated over many iterations. I would also not be working on a single piece of functionality at one time, generally I was working on two-three different parts of the project at a single time focusing on one the most, while doing minor revisions on the others.

Research publications:

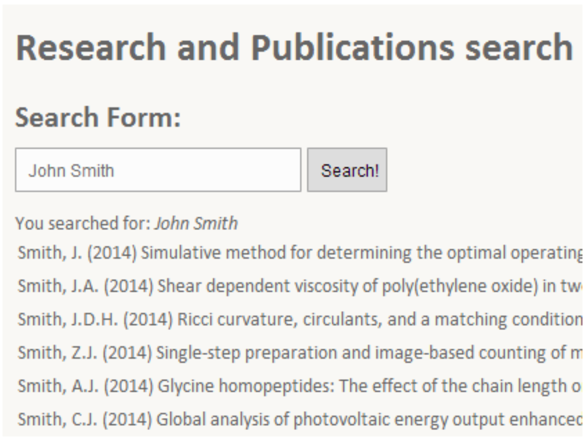
One of the functionalities that my academic supervisor wanted to add to the existing profile pages, was the ability to view a list of publications and research papers that have been written by the user you search for. My supervisor wanted to implement this by adding a button that links to these publications at the bottom of the profile page. Hosting the publications may not be an option due to copyright and licensing issues, so to solve this problem the page would simply contain links to the articles which are hosted on the scholarly publication website.

The bigger problem though, is currently users are currently not in the habit of updating their profile pages. So users would not have any incentive to fill up this information and keep it maintained, even if we were to add these additional fields of information. So to overcome this, the plan was to gather this information dynamically off internet and automatically attach this information to the profile of a user (without requiring them to enter any details themselves). This would be done as a scheduled task in the background to ensure information that is retained is up to date. The source to collect this information is to gather it from a scholarly publication site.

(Falagas, Eleni, Malietzis, & Pappas, 2008) conducted a comparison of scholarly websites. Their results stated that Scopus missed out on older articles whereas Web of Science didn't. However Scopus had a larger coverage of articles than Web of Science did and more than double the number when compared to PubMed. PubMed was excellent for producing biomedical articles. Google Scholar was good at finding many results (but that also included publications which may be of a lower standard) – but covered all range of topics.

Initially the plan was to start off retrieving results from one publication source by iterating through every user in the Opus Database and figuring out if they had written any publications. Additional parameters such as the country they are staying in would be used to identify the correct user on the publication site. Then in the future we would expand to later collect data from multiple sources, aggregate them together and remove duplicate results. This would have ensured we collected most of the publications available without requiring to do anything themselves. However many limitations soon kicked in upon further research into the solution.

PubMed and Google Scholar both did not offer any API's to access resources. *Web of Knowledge* requires a large fee to access their resources. Only *Scopus* (which includes *Science Direct*) had a free API, but that too was limited. Their paid API version again was a significant subscription fee. Upon further investigation into other academic databases I discovered smaller ones such as *CiteSeer* which supports OAI (Open Archives Initiative) but is limited in size and has no public API that can be used for search. *JournalSeek* and *BASE* have no API at all. After taking all this into account I decided I would still try to see if I could gather resources using Scopus's free API access – since it still was a significantly sized library. I had to register and had to submit the IP address that I would be using the API from. In the future this would have been something that would need to be addressed, since it would be a tedious process adding every single IP address in the company to the list of allowed API's on Scopus's website. A better solution would be to make all API requests route through a single proxy that handles all communication with Scopus's servers. This would have been needed to be implemented in the future.


Research and Publications search

Search Form:

John Smith Search!

You searched for: *John Smith*

Smith, J. (2014) Simulative method for determining the optimal operating
Smith, J.A. (2014) Shear dependent viscosity of poly(ethylene oxide) in tw
Smith, J.D.H. (2014) Ricci curvature, circulants, and a matching condition
Smith, Z.J. (2014) Single-step preparation and image-based counting of m
Smith, A.J. (2014) Glycine homopeptides: The effect of the chain length o
Smith, C.J. (2014) Global analysis of photovoltaic energy output enhanced

Figure 11, The inital publications search page

The initial prototype used a search box to search for an author and using *REST* retrieve a list of publications written by the entered name in a *JSON* format. Due to some search limitations with the free API, a lot of instances when I was searching an author I was getting an empty result set. So to overcome this issue, if the first and last name combination could not be found then a second search is automatically repeated using the initials of the authors first name and their full last name. Although the set received back would be substantially bigger, it at least was a quick and simple fix to what potentially was looking at being quite a large problem.

The received information is parsed using *PHP as shown below*

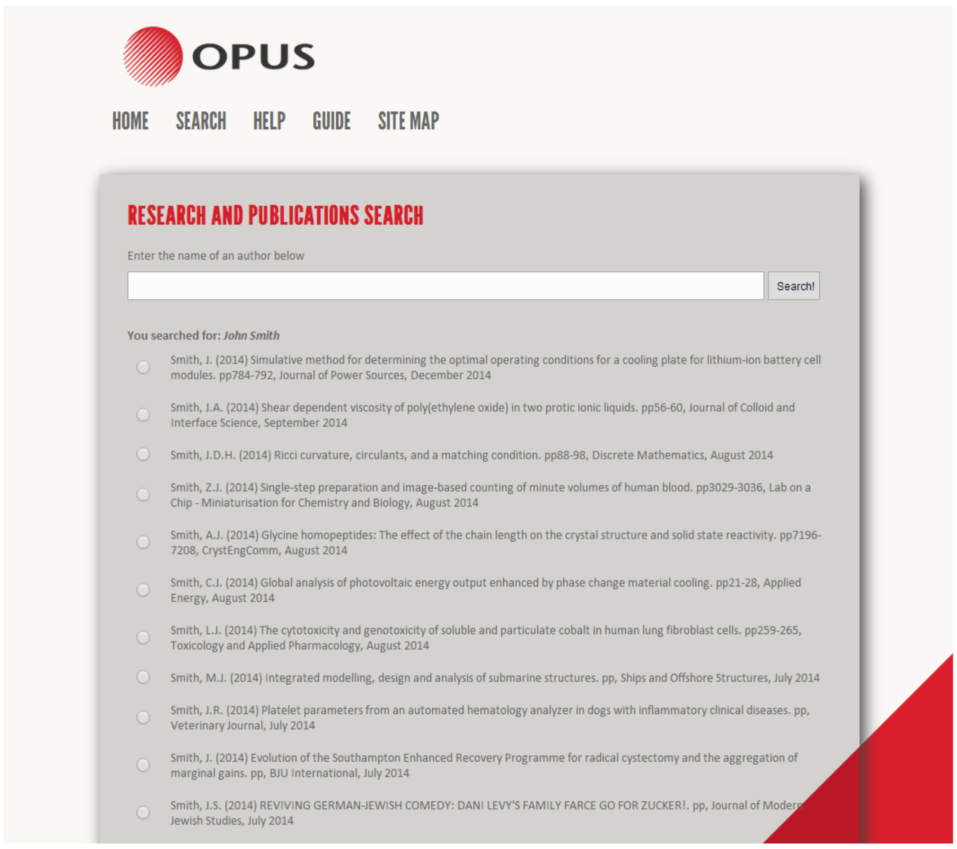
```
$results = $url."&query=firstauth(".$firstName.",".$lastName.")";
$file = file_get_contents($results);
$json = json_decode($file, true);
```

Depending on whether it's a Journal or Article the appropriate formatting is applied. The problem I realized upon creating this is that people have the same name – so we need a way to identify whether the name searched maps to the correct person in real life – e.g. Is *John Smith* the correct John Smith who we searched for in real life. A proposed solution to this that would require minimum input from the user, is to use filters (e.g. entering a country to narrow down results) plus producing *all* the results from the search to the user. From these results let the user select anyone article that they have written themselves. With the article they selected, we would get the Authors ID that we could then use to search for more publications from that author.

I added checkboxes next to each search that were to be used to select which entry was written by the user themselves. I then animated the process of display more publications using *JQuery*.

However what I realized is despite the documentation for Scopus claiming that you retrieve an *Author ID* with each result – in the free version of the API you do not – so even if a user selects a publication they have written – we cannot get their ID to find more publications by them. Secondly searching by *Author ID* is also not allowed in the free version of the API.

The above limitation became a real issue and discussions were made about workarounds to the solution, but nothing seemed to be a viable option. This was the last bit of development that was done for the publications search till date. Another option was looked into which was scraping webpages to gain additional information or results. *Scholar.py* is a python based scraper for Google Scholar that when running locally, will scrape any results from *Google Scholar,* and acts like an API to your client side application. This however was not a viable solution again as web scraping was a part of Google's Terms of Violation, and for a commercial organization is not an ethical option to do. Secondly if the website/structure of Google Scholar results change – this may break functionality in the web scraper that will need to be patched before it can be used again. Therefore this piece of functionality was put on hold and still remains like that till date.



RESEARCH AND PUBLICATIONS SEARCH

Enter the name of an author below

You searched for: ***John Smith***

- Smith, J. (2014) Simulative method for determining the optimal operating conditions for a cooling plate for lithium-ion battery cell modules. pp784-792, Journal of Power Sources, December 2014
- Smith, J.A. (2014) Shear dependent viscosity of poly(ethylene oxide) in two protic ionic liquids. pp56-60, Journal of Colloid and Interface Science, September 2014
- Smith, J.D.H. (2014) Ricci curvature, circulants, and a matching condition. pp88-98, Discrete Mathematics, August 2014
- Smith, Z.J. (2014) Single-step preparation and image-based counting of minute volumes of human blood. pp3029-3036, Lab on a Chip - Miniaturisation for Chemistry and Biology, August 2014
- Smith, A.J. (2014) Glycine homopeptides: The effect of the chain length on the crystal structure and solid state reactivity. pp7196-7208, CrystEngComm, August 2014
- Smith, C.J. (2014) Global analysis of photovoltaic energy output enhanced by phase change material cooling. pp21-28, Applied Energy, August 2014
- Smith, L.J. (2014) The cytotoxicity and genotoxicity of soluble and particulate cobalt in human lung fibroblast cells. pp259-265, Toxicology and Applied Pharmacology, August 2014
- Smith, M.J. (2014) Integrated modelling, design and analysis of submarine structures. pp, Ships and Offshore Structures, July 2014
- Smith, J.R. (2014) Platelet parameters from an automated hematology analyzer in dogs with inflammatory clinical diseases. pp, Veterinary Journal, July 2014
- Smith, J. (2014) Evolution of the Southampton Enhanced Recovery Programme for radical cystectomy and the aggregation of marginal gains. pp, BJU International, July 2014
- Smith, J.S. (2014) REVIVING GERMAN-JEWISH COMEDY: DANI LEVY'S FAMILY FARCE GO FOR ZUCKER!. pp, Journal of Modern Jewish Studies, July 2014

Figure 12, The final research and publications Search page

Maps:

As mentioned earlier employees do not maintain their profile, and since they may be travelling a lot this is a bad practice if physical locations are not updated. One of the solutions that was looked into was the creation of a mobile application where one's location is automatically updated and saved on the intranet, or a mobile application to allow users to check in to the system. This was rules out very early due to the constraints of the intranet only being accessible internally and the fact that a mobile application is not part of the scope or requirements for Opus. Instead a solution was to implement a visual map where the location can be viewed rather than just plain text, and a method to simplify the process of saving one's location, in order to encourage location mantanainance.

Along with the above the map also required additional features that needed to be implemented was the ability to display multiple sets of map markers on the map as an overlay for different groups of users, to be able to geocode a location into a latitude and longitude, to be able to view ones current location on the map, and possibly to get directions between two points in the future. I also raised the point that the ability to cluster points together on the map would be needed at such a scale if many employees are in the same location, the map would look very busy if there were 100 points all in roughly the same area.

There are many mapping services available to use from the internet such as OpenStreet maps, Bing maps and Google maps. When it came to a comparison between such services, (Cipeluch, Jacob, Winstanley, & Mooney, 2011) found no clear winner between either of the services and stated it was down to one's personal requirements. After researching online I created a table showing the features required for the project along with what each Vendor provides.

	Google Maps	**Bing Maps**	**OpenStreet Maps**
Clustering of Map Markers	Yes with plugin	Yes with plugin	Yes with plugin
Gecoding of Information	Yes	Yes	Yes
Group markers by category	Yes	Yes	Yes
Free API	Yes	Yes	Yes
Routing between two points	Yes	Yes	Yes with plugin

Table 1, A comparison of desktop mapping applications

The table above summarizes the functionality available between different mapping platforms. All the services offered all the features and functionality that were required for this project. After using all the three different mapping applications, I also found them to be quite similar in terms of aesthetics *and* functionality. The final decision was made to go with Google Maps due to the fact that it was the most popular option in terms of mapping software used online. Google maps is the most popular standalone mapping application and also the most popular embedded map on used online with 68% of mapping application users (BuiltWith, 2014). Therefore with such a majority it made sense to go with the mapping software which users would be most familiar with which they most likely have possibly used either on the Google Maps website or embedded in another website. This past-experience (if any) would allow users to recognize the tools and controls used to operate the map . This would hopefully make the map more familiar and easier to use (compared to any other competitor mapping application). Secondly the API has a free limit of 25,000 map loads per day which should be sufficient for a company the size of Opus, (Google, 2014).

The maps page embeds the Google Maps HTML5 canvas which uses JavaScript to manipulate the canvas. After successfully embedding the application the first thing I implemented was the ability to show your own location on the map. Most mobile devices have a GPS built in to do this – however for an intranet system that is being used entirely on laptops, this is not a possibility since laptops do not contain a GPS. Instead I am using one of HTML5's new features for geo-location. This native feature can approximate your location by using information data from a variety of sources such as Wi-Fi triangulation and your IP address to calculate a rough approximation.

The code below shows how to use HTML5's geo-location feature is implemented and how to extract the latitude and longitude to display as a map marker for Google Maps.

```
navigator.geolocation.getCurrentPosition(function(position) {
      var pos = new google.maps.LatLng(position.coords.latitude,
      position.coords.longitude);
      var marker = new google.maps.Marker({
            position: pos,
            map: map
      });
});
```

A large number of map markers were then also added to the map in several randomly generated locations to represent the positions of employees in the company (dummy data). One of the requirements was to enable grouping of markers by role/skills. In order to do this, the same map marker image was used on different locations on the map but the colour was changed to represent different groupings. Each of these groupings were placed inside their own array, which represented a different group of users. In the future the locations will be retrieved from the Opus database, which will then need to be geocoded into a latitude and longitude to display on the map. Secondly as opposed to having hardcoded arrays for each group of users – a better solution would need to be looked at such as giving each marker its own class or by iterating through a single unbounded set of points so there is no limit on the number of group of users that can be placed on the map (as currently the limit depends on the number of arrays created).

I then added checkboxes which had JavaScript events attached to them to toggle the visibility of the map markers to show and hide these different groups on the map. One of the issues that was then realized was something that I initially anticipated. With multiple markers in one area, it was hard to see where the points actually were, so we would need some form of clustering.

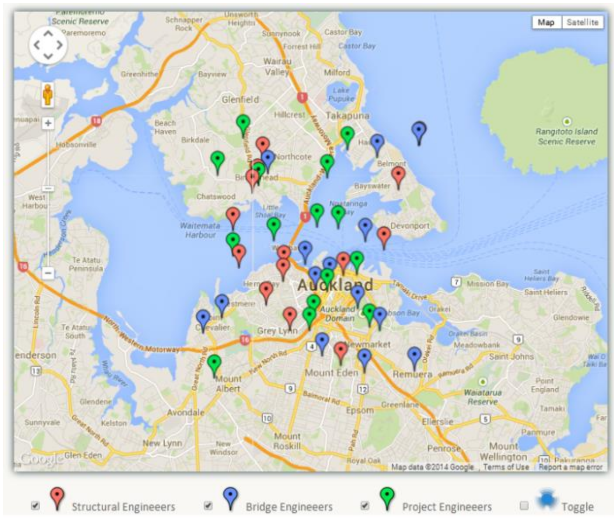


Figure 14, The standard map without using clustering

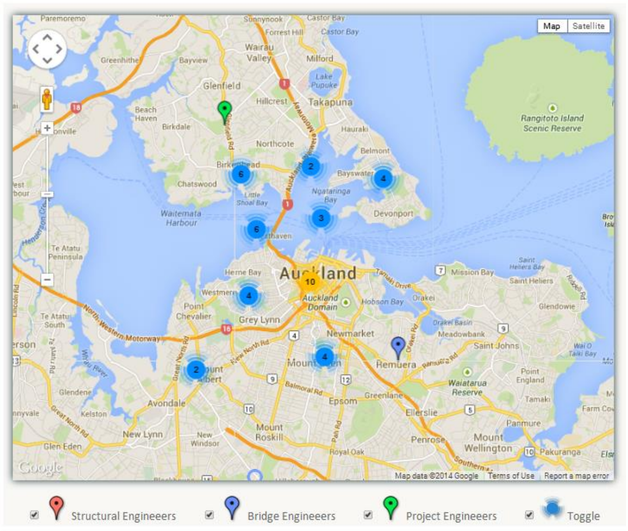


Figure 13, The map after implementing clustering.

Google Maps did not come with a standard method to cluster markers together. However there is an open source utility library for Google Maps, that I added to the project that added the ability to cluster objects. The image above to the left shows the map without any clustering, and the right image shows the map once all clustering was enabled on all three sets of data. The numbers inside each icon represent the number of markers in that cluster. A yellow cluster indicates 10+ people.

The last map functionality added the search with the geocoding capabilities was added. This allowed a user to search for a location, which would be converted into a longitude and latitude that would then be displayed on the map. The textbox followed the style of previous input boxes being a big size with large font. Implementation on how to use Google's geocoding service was used is below.

```
geocoder.geocode({
    'address': address
}, function(results, status) {
    if (status == google.maps.GeocoderStatus.OK) {
        //do action
    }
});
```

Along with those core features mentioned above, other little frills were added such as animations on pin drops, popup information boxes on clicking a map marker and basic error handling. The final maps page design is below. Note – the reason the map is not themed with the proposed theme, is because the map isn't accessed directly (for now at least). Instead it's viewed by accessing it from the profile page which opens in a lightbox over the page.

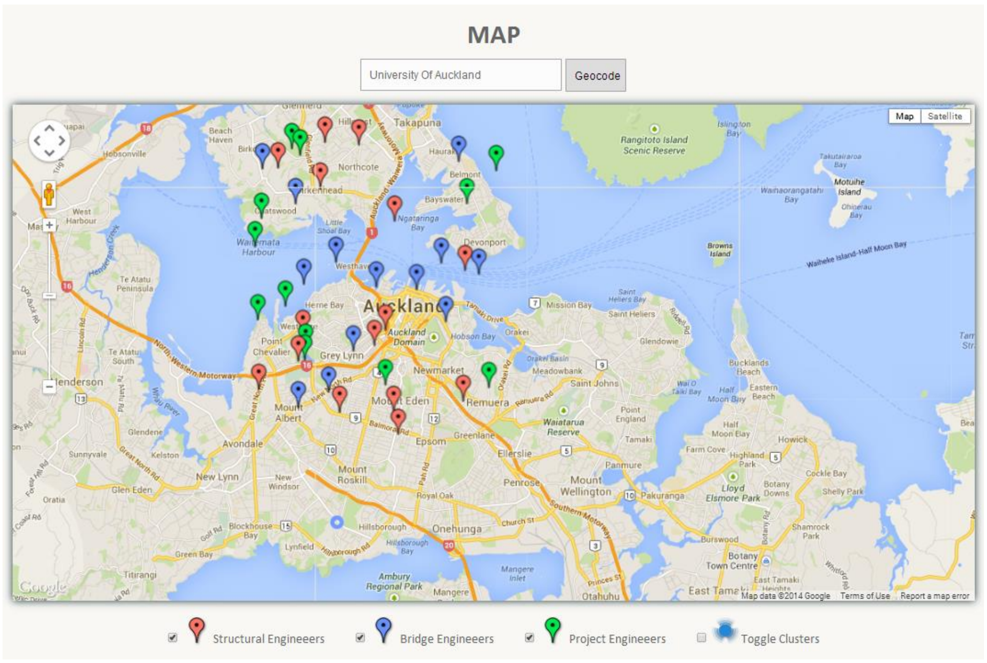
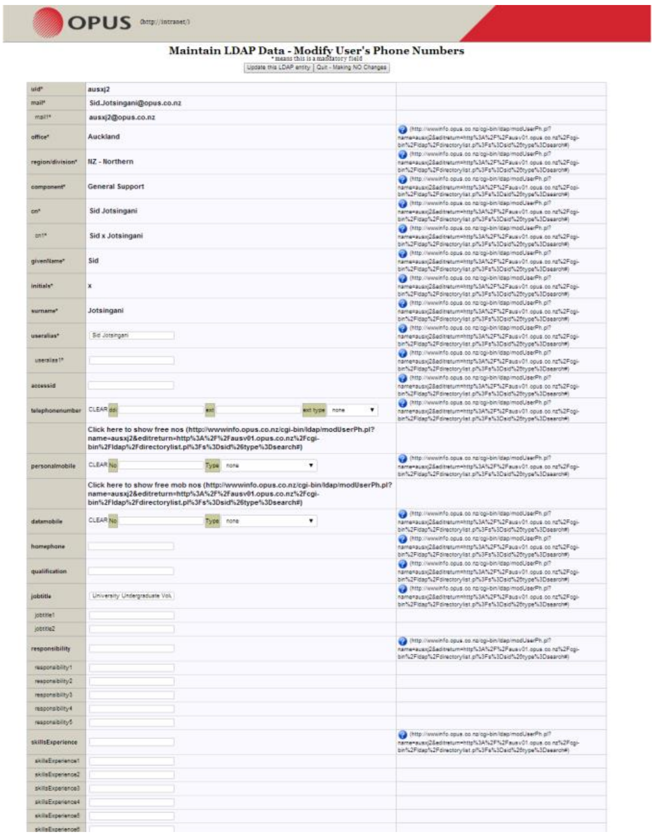


Figure 15, Screenshot of the final maps application page

Profile & Profile Management:

The profile page is one of the most deceptive pages in the proposed redesign. Despite looking like not much has been changed, beneath the surface there have been a lot of functional changes completed in the hope of enhancing usability and user experience. The profile page has to display the following information: Users name, qualifications, job titles, address, phone numbers, email and roles and responsibilities.


OPUS (http://intranet/)

Maintain LDAP Data - Modify User's Phone Numbers

* means this is a mandatory field

Update this LDAP entity | Quit - Making NO Changes

uid*	ausxj2
mail*	Sid.Jotsingani@opus.co.nz
mail1*	ausxj2@opus.co.nz
office*	Auckland
region/division*	NZ - Northern
component*	General Support
cn*	Sid Jotsingani
cn1*	Sid x Jotsingani
givenName*	Sid
initials*	x
surname*	Jotsingani
useralias*	Sid Jotsingani
useralias1*	
accessid	
telephonenumber	CLEAR ddi ext ext type none
	Click here to show free nos (http://wwwinfo.opus.co.nz/cgi-bin/ldap/modUserPh.pl?name=ausxj2&editreturn=http%3A%2F%2Fausv01.opus.co.nz%2Fcgi-bin%2Fldap%2Fdirectorylist.pl%3Fs%3Dsid%26type%3Dsearch#)
personalmobile	CLEAR No Type none
	Click here to show free mob nos (http://wwwinfo.opus.co.nz/cgi-bin/ldap/modUserPh.pl?name=ausxj2&editreturn=http%3A%2F%2Fausv01.opus.co.nz%2Fcgi-bin%2Fldap%2Fdirectorylist.pl%3Fs%3Dsid%26type%3Dsearch#)
datamobile	CLEAR No Type none
homephone	
qualification	
jobtitle	University Undergraduate Vol.
jobtitle1	
jobtitle2	
responsibility	
responsibility1	
responsibility2	
responsibility3	
responsibility4	
responsibility5	
skillsExperience	
skillsExperience1	
skillsExperience2	
skillsExperience3	
skillsExperience4	
skillsExperience5	
skillsExperience6	

Figure 16, A screenshot from the profile edit page

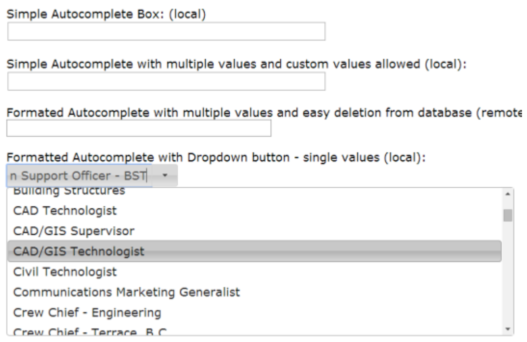
One of the biggest challenges with Opus's current profile system, is managing ones profile. The page that is used to edit a profile is overly complicated. *Figure 16* on the left shows a screenshot from the editing page. There are an excessive number of fields and the overall layout is very crammed and displeasing. Also all the input fields for text were only textboxes, dropdown menus were not being used anywhere, which for data validation and data aggregation is quite bad. One user may call themselves a *Structural Engineer* whilst another may say *Engineer – Structural.* Although they both have the same meaning, they will be seen as two different things by the database. So some form of validation needs to be implemented. An alternate to a dropdown, is an autocomplete box. It could be used in order to allow users to select on option from a predetermined set of data for better validation.

I started to look up different options of input boxes that could help with data validation. To achieve autocomplete and responsive input boxes, the choice to go with JQuery since it would be a good choice to use for something like an autocomplete that needs quick, dynamic and responsive results. What I also saw was that there were a lot of autocomplete/dropdown plugins that were using JQuery, so I decided to try out some to see which works best.

There were four types of autocomplete/dropdown menus (*Figure 17*) that I put together on one page. They all had the basic functionality required, but were slightly different such as having different styles or the ability to select multiple items vs. single items. I showed all these four to my academic supervisor, and there seemed to be an obvious choice between the four. The one selected, named *token-input*, can hold multiple values and could easily delete any one of the multiple values.

I had to setup a MySQL database and create a table containing the different types of roles at Opus. The connection was done via PHP. Since I do not have access to actual data, I gathered this list through the list of open jobs on the careers page of the Opus website. Once the table was set up the values were being fetched from the database and fed into the autocomplete box. To do this I had to set up a persistent SQL connection with the database.

```
mysql_pconnect($host, $user, $password);
mysql_select_db($database);
$query = sprintf("SELECT id, name from
roles WHERE name LIKE '%%%s%%' ORDER BY
name DESC",
mysql_real_escape_string($_GET["q"]));
$results = mysql_query($query);
```


Simple Autocomplete Box: (local)

Simple Autocomplete with multiple values and custom values allowed (local):

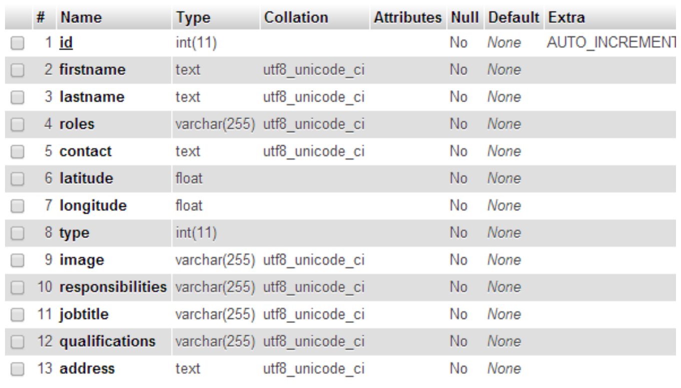
Formated Autocomplete with multiple values and easy deletion from database (remote

Formatted Autocomplete with Dropdown button - single values (local):

n Support Officer - BST

Building Structures
CAD Technologist
CAD/GIS Supervisor
CAD/GIS Technologist
Civil Technologist
Communications Marketing Generalist
Crew Chief - Engineering
Crew Chief - Terrace, B.C.

Figure 17, Four of the different auto-complete box plugins that were contemplated between



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	firstname	text	utf8_unicode_ci		No	None	
3	lastname	text	utf8_unicode_ci		No	None	
4	roles	varchar(255)	utf8_unicode_ci		No	None	
5	contact	text	utf8_unicode_ci		No	None	
6	latitude	float			No	None	
7	longitude	float			No	None	
8	type	int(11)			No	None	
9	image	varchar(255)	utf8_unicode_ci		No	None	
10	responsibilities	varchar(255)	utf8_unicode_ci		No	None	
11	jobtitle	varchar(255)	utf8_unicode_ci		No	None	
12	qualifications	varchar(255)	utf8_unicode_ci		No	None	
13	address	text	utf8_unicode_ci		No	None	

Figure 18, The structure of the "Users" table in the MySQL database

After the visual redesign of the profile page, the next implemented thing was adding a table for users in the database. I wanted the data shown on the profile page to be data that was dynamically retrieved from the database when the page loads. Again since I have no access to Opus's actual database or servers, this was all done locally including the connection using dummy data. The structure of the table was created, based on the information I could view from the old profile page, and from the edit my profile page to figure out some of the fields that would exist in the database. *Figure 18* shows the structure used for the Users table of the database.

Now all the data displayed on the profile page was being grabbed from the database. I now needed to implement the profile editing page. As *Figure 16* shows, the current editing page had a very complex look and feel to it. So the goal was to create a way to simplify down the editing process to the lowest level possible, to make it significantly easier to edit ones profile. The options were:

- To improve on the existing GUI and use the standard form based layout to edit data. This would mean a user has to navigate to a new page, and they have to find the fields they wish to edit, change the values in that field and hit save. In the user's mind there would be some form of mapping going on such as which field relates to what part of the profile page
- The second option was to implement some form of *WYSIWYG* editor. This however would not bring any more simplicity than the standard method of editing forms, and a WYSIWYG editor leaves the design decisions up to the end user. For example in a rich text box a user may decide to use a bold green font throughout. This would look unappealing and would lack consistency between profile pages. Plus the layout of information may be in different places since the end user now has all this additional flexibility.
- The third option was the one I implemented. This was a feature I created from scratch which I like to call "double click to edit".

The double click to edit function, means that a user could double click any text on their profile page, and the text would then be replaced by a textbox/text-area according to what the context of the content was. The user could then enter the updated value and click a save button that would appear next to the textbox. This would require no mapping between text-fields and actual text on the page (since it is not requiring the user to navigate to another page). It would also act similar to a limited WYSIWYG editor, except only for changing content and not overall styles.

The feature was initially implemented as being hardcoded, but later this was changed to make it work more like a library/plugin. I created new HTML tags called `<editable></editable>` that wrap around a normal HTML element. The editable tags understand what the context is, by going through and reading its child nodes (the wrapped content). Depending on what the wrapped element is, the double clicked content will be replaced with an editable version of the element. Such as a list will be converted to a token input box (the jQuery library list box), whilst paragraph text will be replaced with a plain text-area. An example of how to use the editable tags is shown:

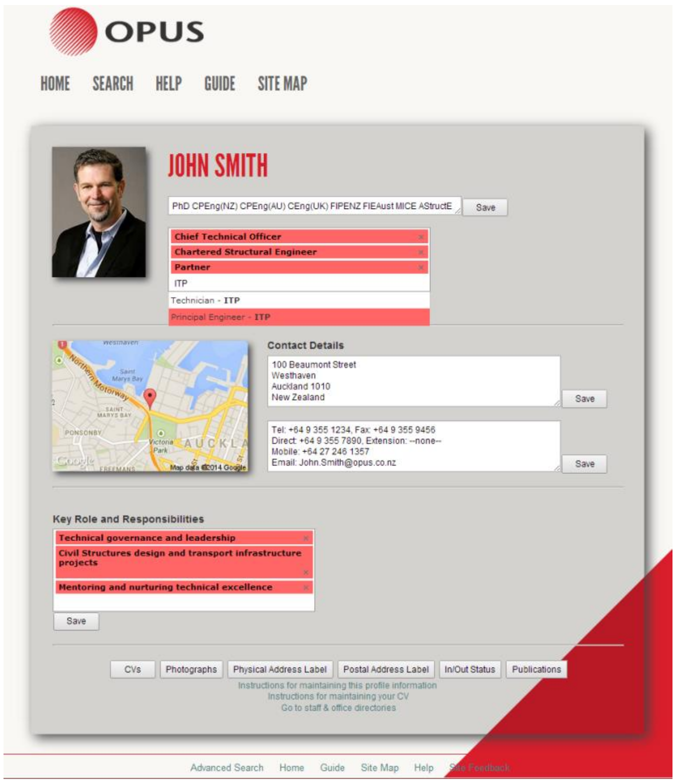
- A single line text area when the content only contains one line
  `<EDITABLE id="role">graduate</EDITABLE>`

- A multiple line text area when the content contains more than one line
  `<EDITABLE id="items">line 1<br/>line 2</EDITABLE>`

- An inputbox containing list items with when the content contains <li></li> list items and the attribute autocomplete is set to off.
  `<EDITABLE id="fruit"><ul><li>apple</li><li>pear</li></ul></EDITABLE>`

- An inputbox containing list items with autocomplete functionality when the content contains <li></li> list items
  ```
  <EDITABLE id="drinks” autocomplete=”off”><ul>
  <li>Coke</li><li>Pepsi</li><li>Sprite</li></ul></EDITABLE>
  ```

There are currently some limitations in terms of functionality, since the double click to edit feature has still not been completed, and there is still room for improvements such as:



JOHN SMITH

PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE

Chief Technical Officer
Chartered Structural Engineer
Partner
ITP
Technician - ITP
Principal Engineer - ITP

Contact Details

100 Beaumont Street
Westhaven
Auckland 1010
New Zealand

Tel: +64 9 355 1234, Fax: +64 9 355 9456
Direct: +64 9 355 7890, Extension: --none--
Mobile: +64 27 246 1357
Email: John.Smith@opus.co.nz

Key Role and Responsibilities

Technical governance and leadership
Civil Structures design and transport infrastructure projects
Mentoring and nurturing technical excellence

Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories

Figure 19, The double click to edit features showing how textboxes and autocomplete-list boxes are replaced with the text the user wishes to edit

- Every editable tag needs a unique ID which currently has to be the name of the row in the database that the information will be saved in. This is a current limitation, but however in the future can be improved on by using a Map or by saving the information about which editable field ID’s relate to which column in the database.
- The second limitation is that currently, for example the contact details are all saved in one row. The telephone, mobile, and fax number would normally be separated as different columns when saved in the database. In the future this can be improved upon by Example of how this works is below.
- Currently there is no way to define where the source of the autocomplete is coming from. All input boxes currently share the same source for auto-complete data. This in future iterations will need to be solved so that different input boxes can get their data from different sources. A possible solution may be to make every editable input box that requires autocomplete have additional parameters in either jQuery or as HTML attributes where you can define the source of the data.
- Fourthly Editable tags will not work when being wrapped around unordered lists which have any attribute attached to them. If an unordered list has for example the following attribute

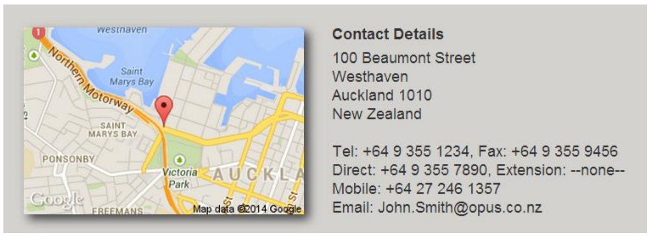
`<ul border="1">` will currently not work. This is due to the way the child nodes are being parsed when trying to resolve the type of the inner element.

There are most likely other small things which need to be improved on, along with extending functionality to other HTML elements such as tables, and the ability to replace an image on double click (such as to change the profile picture), however this prototype shows the possibility of creating a simplified profile editing system that does not require one to navigate to a new page.

The very simplified jQuery code that allows the editing to work is as follows

```
//Attach a double click handler to all items which have the editable tag.
    $("editable").dblclick(function(event) {

//If the item is in editable mode already then break here
    if($(this).attr('clicked') == 'true'){ return; }

//Get the content of the double clicked element
    var content = ($(this).html());

//Work out the element type
    if ($(this).children()[0].tagName == "UL") {

//If a list then split and extract all the list items from the HTML and
//add them to an array else just save the contents
    for (var i = 0; i < listArray.length; i++) {
        items.push(item);

//Replace content with a textarea/listbox and save button and map it to
//appropriate ID/field in the SQL database
    $(this).html("<textarea>" + content + "</textarea>");
    $(this).append("<input type='save'>");

//Enable the autocomplete Ajax Callback requests to the database
    $('#' + col).tokenInput(<?php echo $json_response;?>, {

//Get the current focused input box and change the size of the input boxes
depending on user input
    $(this).animate({ height:new+"px" },200);

//On save, submit the updated content via Ajax and replace the steps
    $.ajax({
        type: "POST",
         url: "update.php",

//Replace the input boxes back by doing the above steps backwards to
//convert the listbox/texbox objects to HTML code to be displayed.
```

The next step that was completed was adding the image of a map next to the contact details. The picture of the map needed to be a visual representation of the address to the right, so that map needed to grab that contact address. This map would show a visual representation of the address on the right. The feature I felt would be best for compactness and simplicity is to simply have an image of the users location on the map that would be displayed, and upon clicking that map, the full interactive HTML5 map opens in a new window/lightbox (rather than embedding a very tiny interactive map on the page). The reason for this was to ensure speed and responsiveness when loading the profile page, and to hide all the extra bits of functionality (such as the geocoding search) that exist on the map page already. So rather than trying to add all the functions of the maps page and place it on to the profile page (where it would just take up space and may not even be needed) instead I am just showing a simple image which can be clicked to access the full functionality of the mapping application.



Contact Details
100 Beaumont Street
Westhaven
Auckland 1010
New Zealand

Tel: +64 9 355 1234, Fax: +64 9 355 9456
Direct: +64 9 355 7890, Extension: --none--
Mobile: +64 27 246 1357
Email: John.Smith@opus.co.nz

Figure 20, The map image from the profile page. The map is a dynamically generated image which is created by reading the contact details from the database and sent via an API to Google to generate a PNG image that is used on the webpage Clicking this will open up .

The way the static image is created is by using the Google Maps Static API. The API works by taking in an address and geocoding it to a latitude and longitude which it displays on the map. The result is an image that can be embedded by using normal HTML `<img>` tags.

The code below shows us the call made to Google to retrieve a static image for the address: *100 Beaumont Street, Westhaven, Auckland 1010*. Note: to get the address in the right format some whitespace, newlines and special characters need to be removed. Also the URL will need to be encoded before making the request.

```
https://maps.googleapis.com/maps/api/staticmap?
zoom=14&size=300x200&maptype=roadmap&markers=color:red|
address:100 Beaumont Street Westhaven Auckland 1010 New Zealand
```

Upon clicking this image, the full maps page that was created earlier was to be opened in a new window. As opposed to opening the maps in a new window, I decided to open it in a lightbox. This makes it easy for a user to return back to the profile page and to exit from the map quickly without having to reload the page, or close a new tab or window. Different solutions were looked into for a lightbox library, and there were many to choose from.

The most popular options were all implemented in jQuery. jQuery LightBox, ColorBox, FancyBox, ThickBox, OrangeBox and SlimBox. The functionality though differed slightly between all the options. There were three pieces of functionality that I felt were the main required.

- The ability to embed an iFrame inside the lightbox. This iFrame would be required in order to open the maps page inside the lightbox.
- Also for the technical funding application (*see next section),* one of the features that would be needed planning ahead, is the ability to embed a PDF inside a lightbox

- Again planning for ahead, the lightbox should be able to handle images (which is what a lightbox is designed for)

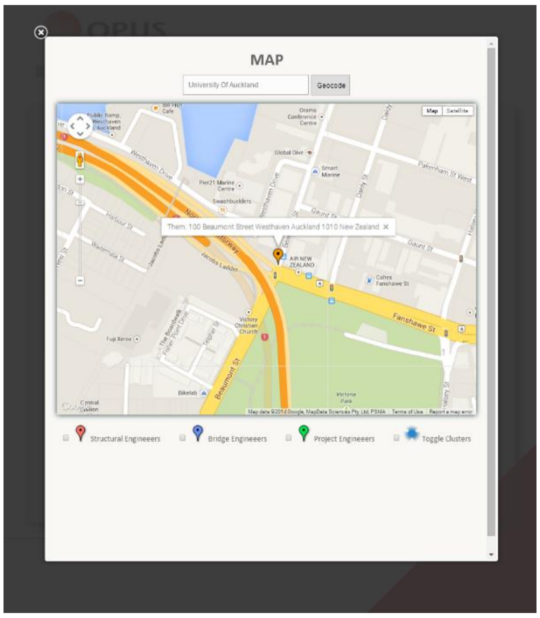


Figure 21, The lighbox library - OrangeBox as implemented on the profile page to display a popup modal window of the interactive map.

(Deering, 2011) had a quick comparison between different libraries in which he states the differences between the lightbox libraries are minimal, and your selection should be based upon the optional extras and customizability you require. Between all the available libraries, *Thickbox* can view PDF's in iFrames, but you will still need to implement some PDF viewer yourself for those browsers that do not support it. *OrangeBox* and *FancyBox* both could support PDF's natively. The rest were ruled out since they could not display inline PDF's. Both could also support iFrames and images. The deciding factor between the two options was that FancyBox was not free for commercial use, whereas OrangeBox was – therefore the lightbox library chosen was OrangeBox.

To get the lightbox to show the correct location upon opening it, data had to be sent from the profile page to the inline iFrame. This was done by using HTTP *GET* method implemented in PHP. The address was copied from the "*Contact Details"* paragraph using jQuery, and was sent over the URL to the maps page. The maps page received the information, and geocoded the address to a latitude and longitude. Then a new map marker was created at that latitude and longitude with an information box above showing the address text and the map was centered around it. The OrangeBox plugin also needed to be tweaked. Although I wanted to keep the plugin native, I could not find a way at the time to allow the lightbox iFrame to update without having to modify the plugin files.

Combining both the functionality to double click to edit and the static map together required a little bit more to be done in order to get them working together. In the scenario a user double clicks to change their location, we would also like the static map to be updated to the entered location. In order to do this we need to get the new value entered, and change the Image ref to contain the new address. So two steps needed to be added to the double click handling process on *Figure 21* above

```
//If the address has changed, then update the static Google map image.
    $map.href = $url + encodeURIComponent($new_address);

//Execute the OrangeBox script again to update link references.
    head.appendChild("orangebox.js");
```

Now the last change to the profile page was to fix an inconvenience. When a user double clicked a paragraph of text, the text was replaced by a text-area. The text-area created had a height that fit perfectly to the paragraph it replaced. The problem was however when a user entered a new line, the text area size would stay the same, so a vertical scrollbar would appear. Although being a small issue, upon using the double click to edit feature many times myself – I realized how annoying this could be. In order to solve this problem I looked at a solution of making the textbox grow or shrink depending on whether you add a new line or delete a new line.

There were already existing plugins available for this, however for some reason they were all very large in size. *autogrow.js* takes up over 120 lines, *Autogrow-TextArea* was almost 100 lines without any animations, and the most popular, Jack Moore's AutoGrow over 270 lines. I felt I could implement this myself in a shorter way including animations. I implemented my own solution with 12 lines with one statement per line including the JQuery event handling. It accomplishes the same job in a much simpler way by using some math and matching the special new line characters, to calculate what the height should be based on the font size and the line-height after every key stroke. Upon adding a new line, the textbox animates and "grows", while it "shrinks" on deleting one.

Technical Funding:

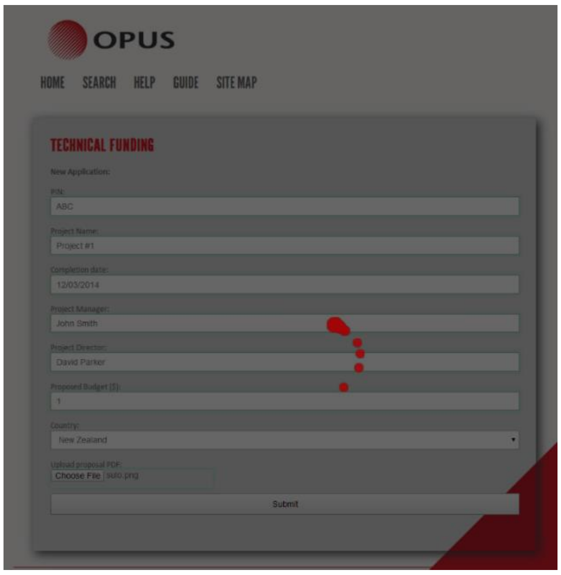
Every year, employees of the company apply for funding for the projects they are working on. These can be small applications with low budget requirements, to projects that depend heavily on funding. Every application has to be filled out using an online form. This form is not easy to use and one user even pointed out without even me asking, that they did not fill up the form themselves since the application process didn't work for them.

All applications have to include a project director, manager, a budget, the country the project is being worked on, the completion date of the project and a PDF with further details about the project itself. In the back end of the system the data is aggregated by hand. Someone manually creates an excel spreadsheet after accumulating all the information entered from all the applications at the end of every year. There is also a time delay between when all the applications are submitted to when the excel spreadsheet is created. My academic supervisor who then sees these applications has to wait and rely on someone else to create the spreadsheet and send it to them.

The first part of my solution involved redesigning the application form online. To do this I created basic fields in HTML all designed with usability in mind. I took advantage of new HTML5 form input types and functions to provide better control and validation (W3Schools). The reason for this is to try and improve on the usability, since users will receive appropriate feedback upon performing actions. Some of the feedback shown is e.g. by entering letters in a *"number"* field will highlight the textbox red telling the user something is incorrect. Or by being shown a date picker modal box when entering a date, and receiving messages about required input boxes when they have been left blank. Standard conventions have also been used along with these such as red highlights for incorrectly/blank fields and green highlights for correctly entered fields.

For the design aspect I am using large textboxes with large padding, width and a bigger font-size. The goal is to make the application process as easy as possible and to retain focus on the content entered on the input fields. The bigger font and input boxes are an attempt to stand out, and ensure focus is retained on those elements and also follows this pattern used by many larger websites which also follows modern design conventions as seen in the top websites.

I have used a combination of Ajax and PHP to process the forms contents. Although HTML already handles some errors such as blank fields - some additional error handling had to also be implemented. Upon submitting the form, the contents are grabbed and are sent via POST to a PHP page using Ajax. The PHP page processes the data by first looking at the attachment. It checks the file size to see if it's less than 20MB, and then the extension to ensure it is only a PDF file and nothing else. Whilst this is happening I created a custom animated loading screen using purely CSS and an animated GIF (red - to suit the Opus colours), that fades in to look like a lightbox (overlay on the webpage where the webpage is dimmed out) as seen on the next page in *Figure 22.*



TECHNICAL FUNDING

New Application:

PIN: ABC

Project Name: Project #1

Completion date: 12/03/2014

Project Manager: John Smith

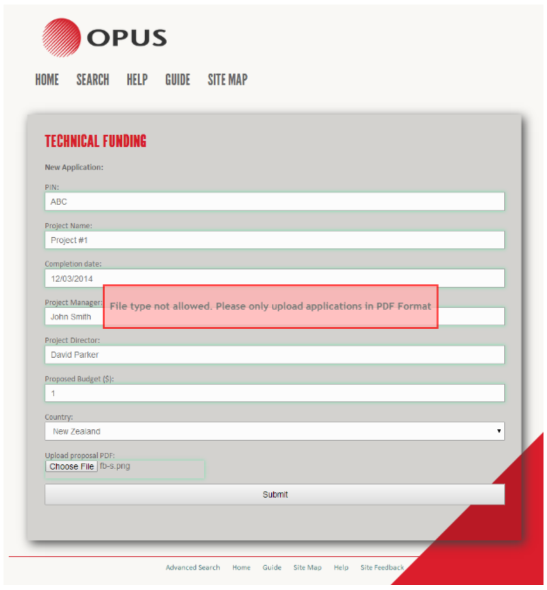
Project Director: David Parker

Proposed Budget ($): 1

Country: New Zealand

Upload proposal PDF: sufo.png

Figure 24, Custom animated loading screen created using purely CSS and images


OPUS
HOME SEARCH HELP GUIDE SITE MAP
Choose File
Submit
Advanced Search Home Guide Site Map Help Site Feedback

TECHNICAL FUNDING

New Application:

PIN: ABC

Project Name: Project #1

Completion date: 12/03/2014

Project Manager: John Smith

File type not allowed. Please only upload applications in PDF Format

Project Director: David Parker

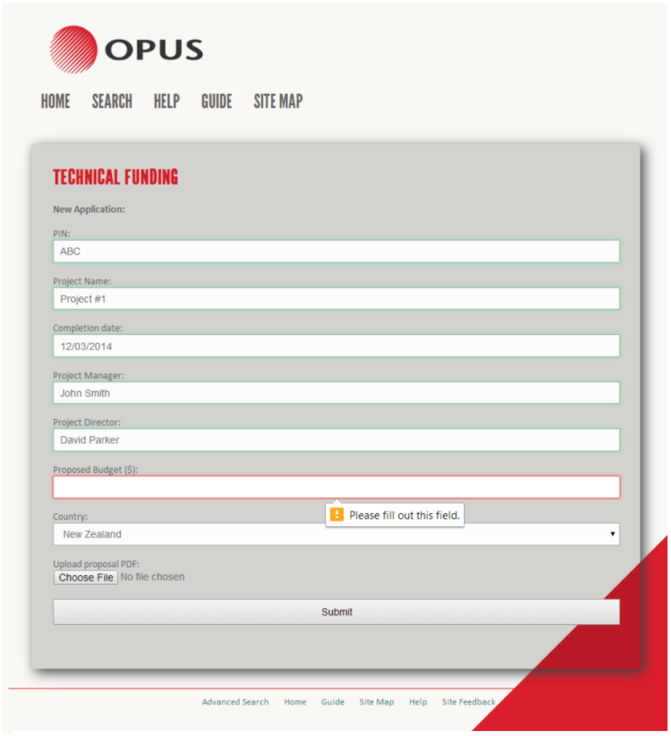
Proposed Budget ($): 1

Country: New Zealand

Upload proposal PDF: fb-s.png

Figure 24, Example of a CSS modal box appearing on invalid file type input

After the PHP page has processed the form content it will send a call back to the Ajax function telling it whether it could successfully save the form in the SQL database or whether an error occurred. An error could occur due to the file size being too big, the extension not being a PDF or other reasons such as it could not simply connect to the database (*Figure 23*). Whatever the reason, a custom modal box which I created using purely CSS is transitioned on screen in either a red colour with the appropriate error message, or in a green colour saying the application was successfully submitted. This whole process happens on the one page for the user – and if an error message is sent back then none of the fields of the form are cleared. This allows the user to simply edit the incorrect data rather than having to await a page refresh and possibly having to enter all the information again.


OPUS
HOME SEARCH HELP GUIDE SITE MAP
Choose File
Submit
Advanced Search Home Guide Site Map Help Site Feedback

TECHNICAL FUNDING

New Application:

PIN: ABC

Project Name: Project #1

Completion date: 12/03/2014

Project Manager: John Smith

Project Director: David Parker

Proposed Budget ($):

Please fill out this field.

Country: New Zealand

Upload proposal PDF: No file chosen

Figure 22, The final application form for technical funding showing validation and an example message and highlighted textbox when leaving a field empty

All of the above functionality was my implementation/possible solution for the front-end process on submitting a technical funding application. The second problem which was faced by my supervisor was the fact that all this data needed to be aggregated somehow. Until now someone had to manually do this aggregation and had to manually create a spread-sheet from this information and send it along with all the individual project PDFs to my supervisor.

The front-end system saves all the information in a MySQL database. The data is saved in its own applications table which does not need to be directly accessed. Instead I have created a basic management panel to view all the applications (which in the future will only be accessible by my supervisor or anyone with permissions).

The management (administration) panel uses primarily JQuery and HTML5 to display the application information. I am also using to libraries for JQuery which I have customised to work with a SQL database. The first is called *Flot* which is a plotting library for creating different types of graphs. And the second is *DataTables* which is used for enhancing standard HTML5 tables.

Upon visiting the management panel, one can view graphs about all the applications that have been submitted till date. To do this I had to first make a connection to retrieve the applications table from the SQL database, then to convert each of the rows into an array of objects into a format that could be understood by the Flot Library. With some additional parameters and information Flot then outputs a HTML5 canvas which contains a graphed version of the inputted data. I am using bar graphs and pie graphs because for this application they are useful at displaying a large amount of data in a relatively simple manner.

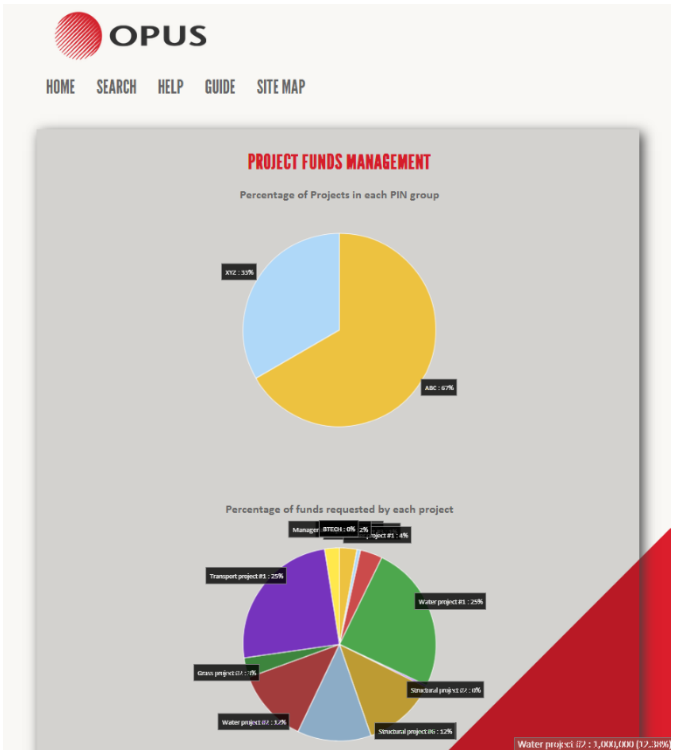
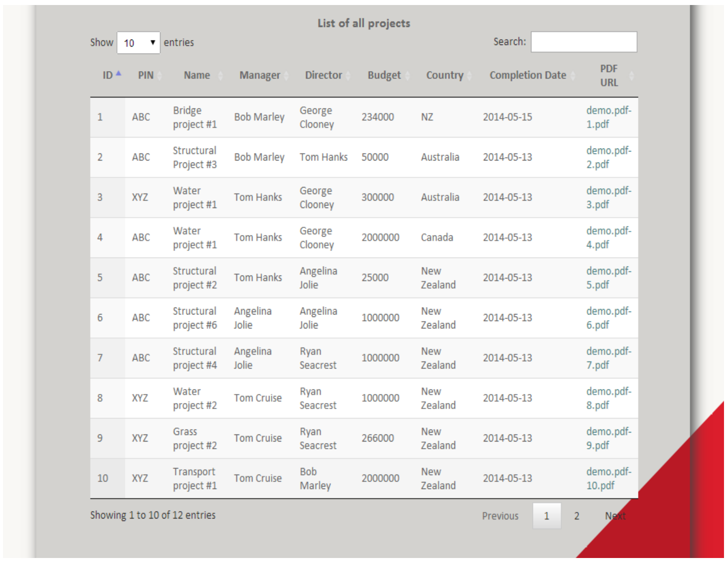

OPUS
HOME
SEARCH
HELP
GUIDE
SITE MAP
PROJECT FUNDS MANAGEMENT
Percentage of Projects in each PIN group
XYZ : 33%
ABC : 67%
Percentage of funds requested by each project
Manager
BTECH : 0%
2%
project #1 : 4%
Transport project #1 : 25%
Water project #1 : 25%
Grass project #2 : 3%
Structural project #2 : 0%
Water project #2 : 12%
Structural project #6 : 12%
Water project #2 : 1,000,000 (12.38%)

Figure 25, The management panel for technical funding applications, showing pie graphs created using the Flot library

There were many libraries available for plotting over many different languages. JavaScript/JQuery again seemed like a good choice as it is a web language that will fit well with all the other pages I have created so far which also are strongly linked with JavaScript and JQuery. Also JavaScript is highly responsive and interactive, which is ideal for viewing graphs online as users can interact with it. A comparison of JavaScript graphing engines was done by (Whelan, 2010) and by (Social Compare, 2014). Despite all libraries having their differences most of the graphing software will cover the basic functions needed to generate graphs. The best libraries seemed to be *HighCharts*, *D3* and *Flot* because of their customisability and resultant output – that for all three were much more aesthetically pleasing than the others. HighCharts seemed to be the best choice – except it costs from \$90 – \$3600 to use for commercial organisations whilst D3 and Flot were the free options. D3's output charts seemed to be more for complex data analysis and seemed too complex and contained too much information compared to what was required for this application – where simplicity and usability triumphs over the amount of functions the program contains. Therefore I chose Flot as the library to use.

The resultant design of the graphs is mostly standard to what comes default with the program. Labels for each slice of the pie chart were added along with the mouse over function to view the percentages in the bottom right corner to view information that may not fit or be obscured due to being very small slices in the pie chart. Secondly multiple graphs were added so the same information can be viewed and grouped by different criteria. Along with the pie graphs a simple bar graph was also created to view the budget required by each project.

Graphs are great to get an overview of information in a small amount of time. However in this scenario it would also be handy to view information about each individual project so it can be dealt with accordingly. A table with all the information would be a solution to be able to view all the information about each application. Unlike with the graphing applications where there the usage of each library was strongly spread, with tables there were two main plugins that the majority of users seemed to use (Kumar, 2011). *DataTables* and *JQGrid*.



List of all projects

Show 10 entries | Search:

ID	PIN	Name	Manager	Director	Budget	Country	Completion Date	PDF URL
1	ABC	Bridge project #1	Bob Marley	George Clooney	234000	NZ	2014-05-15	demo.pdf-1.pdf
2	ABC	Structural Project #3	Bob Marley	Tom Hanks	50000	Australia	2014-05-13	demo.pdf-2.pdf
3	XYZ	Water project #1	Tom Hanks	George Clooney	300000	Australia	2014-05-13	demo.pdf-3.pdf
4	ABC	Water project #1	Tom Hanks	George Clooney	2000000	Canada	2014-05-13	demo.pdf-4.pdf
5	ABC	Structural project #2	Tom Hanks	Angelina Jolie	25000	New Zealand	2014-05-13	demo.pdf-5.pdf
6	ABC	Structural project #6	Angelina Jolie	Angelina Jolie	1000000	New Zealand	2014-05-13	demo.pdf-6.pdf
7	ABC	Structural project #4	Angelina Jolie	Ryan Seacrest	1000000	New Zealand	2014-05-13	demo.pdf-7.pdf
8	XYZ	Water project #2	Tom Cruise	Ryan Seacrest	1000000	New Zealand	2014-05-13	demo.pdf-8.pdf
9	XYZ	Grass project #2	Tom Cruise	Ryan Seacrest	266000	New Zealand	2014-05-13	demo.pdf-9.pdf
10	XYZ	Transport project #1	Tom Cruise	Bob Marley	2000000	New Zealand	2014-05-13	demo.pdf-10.pdf

Showing 1 to 10 of 12 entries

Previous 1 2 Next

Figure 26, The overview of technical funding applications as shown using the DataTables library

Despite doing the same job – the libraries differed significantly on how they work. DataTables work simply by creating a normal Table in HTML and the library then uses JavaScript and CSS to modify the table to give it the additional functionality and styling automatically such as the odd-even row colouring and the ability to limit the number of rows per page. JQGrid on the other hand required passing in data through a source such as JSON/XML without you having to do the HTML structure yourself. Both options had extensive support and documentation, but for commercial usage, JQGrid required a licence which costs $299 – thus DataTables was the option I used. *Figure 26* shows the implementation of DataTables in the management section. The OrangeBox plugin was also used to be able to view any PDFs that were attached with a funding application in a lightbox on the same page itself. This meant a user did not have to navigate away to a new tab in order to view a PDF attachment, and to go back they could simply close the lightbox allowing easy reversal or actions.

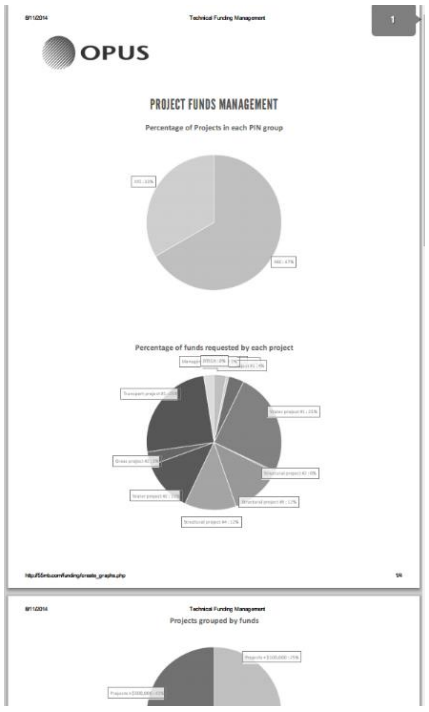

Technical Funding Management
OPUS
PROJECT FUNDS MANAGEMENT
Percentage of Projects in each PIN group
Percentage of funds requested by each project
Technical Funding Management
Projects grouped by funds

Figure 27, A print preview of the technical funding management panel

Since graphs would also be printed out, an adjustment that was done was to ensure that all the content will fit and look good when the page is printed. CSS allows you to specify the layout of your pages when printing to ensure a printer-friendly format using the @media rule. The left figure shows the how the graphic page appears when printing. Note that the navigation, background images and shadows are all removed in the printing process and only the graphs and tables are being printed out.

Finally upon completing the design and functionality, the last step was ensuring cross-browser compatibility. This mean adding extra CSS rules to ensure that the website appears to be the same when viewed on all browsers. The changes were mostly to make sure the design worked in Internet Explorer.

The screenshots below show the same webpage viewed in multiple browsers along with the editing mode enabled of the roles of the user.

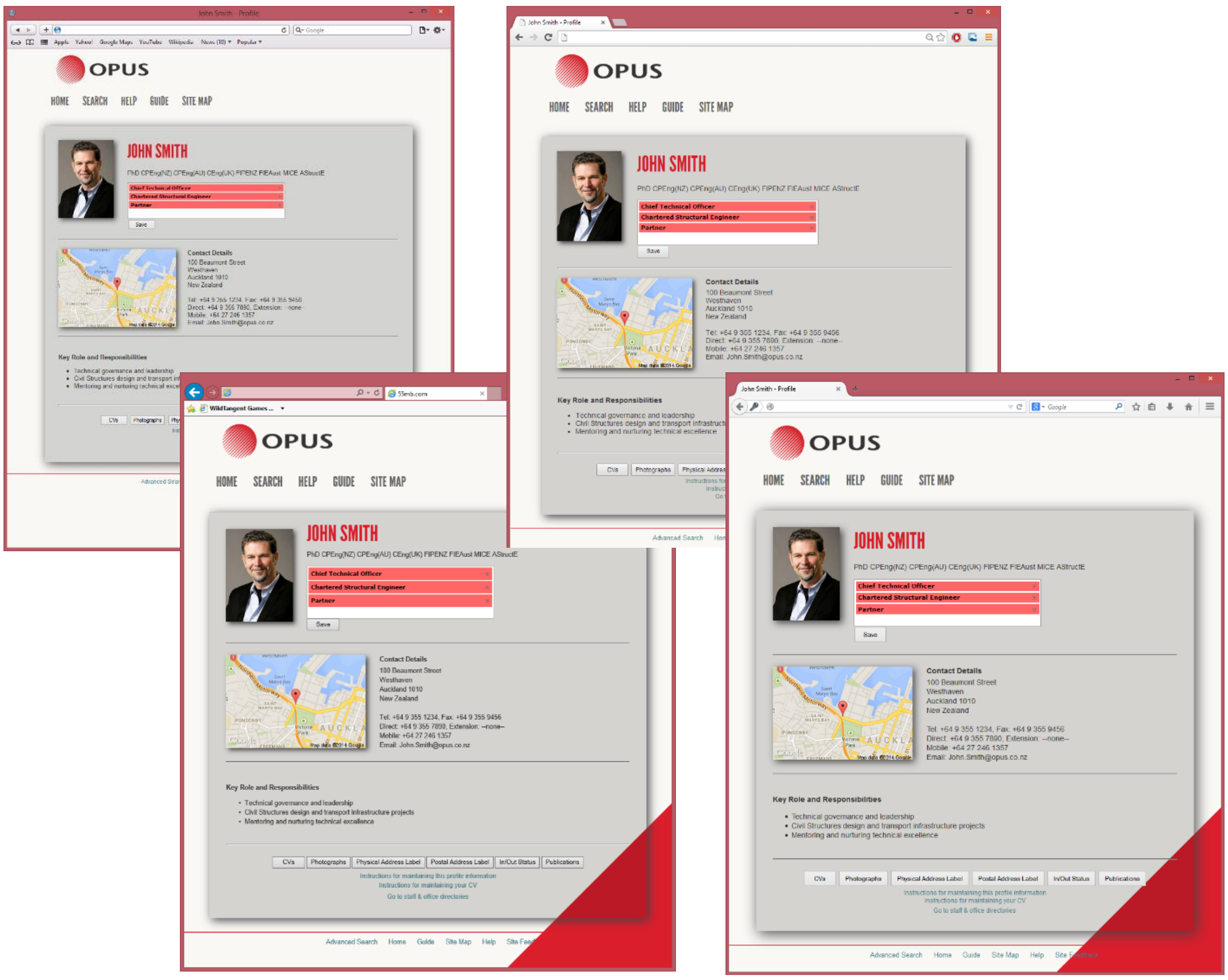

OPUS
HOME SEARCH HELP GUIDE SITE MAP
JOHN SMITH
PhD CPEng(NZ) CPEng(AU) CEng(UK) FIPENZ FIEAust MICE AStructE
Chief Technical Officer
Chartered Structural Engineer
Partner
Save
Contact Details
100 Beaumont Street
Westhaven
Auckland 1010
New Zealand
Tel: +64 9 355 1234, Fax: +64 9 355 9456
Direct: +64 9 355 7890, Extension: --none--
Mobile: +64 27 246 1357
Email: John.Smith@opus.co.nz
Key Role and Responsibilities
Technical governance and leadership
Civil Structures design and transport infrastructure projects
Mentoring and nurturing technical excellence
CVs
Photographs
Physical Address Label
Postal Address Label
In/Out Status
Publications
Instructions for maintaining this profile information
Instructions for maintaining your CV
Go to staff & office directories
Advanced Search
Home
Guide
Site Map
Help
Site Feedback

Figure 28, The proposed design of the profile page as viewed in (from left to right) Safari 5.1, Internet Explorer 9, Mozilla Firefox 30, Google Chrome 36

Conclusion

The above project is just the progress on my goal of enhancing Opus's intranet system's usability and user experience. The changes I have made so far have been based upon the research I have conducted so far, and what I feel will be a design that enhances user experience. Since design is very subjective, I cannot be sure whether my proposed solution will improve, worsen or affect the usability and user experience at all. In the future I will need to conduct some tests to see whether my proposed solution, does in fact improve upon the existing system. Along with the changes, the biggest issue that I will face (if it happens) will be the migration of my changes on to the live intranet system. However this is all dependant on the IT team of Opus. This is just the start to my proposed solution and I will be continuing to work on adding functionality and altering the design to hopefully improve usability and create a better user experience.

Bibliography

Alexa. (2014). *The top 500 sites on the web*. Retrieved from Alexa: http://www.alexa.com/topsites

Bevan, N. (2009). *Wha is the difference between the purpose of usablity and user experience evaluation methods?* Uppalsa: INTERACT.

BuiltWith. (2014, August 4). *Mapping Usage Statistics*. Retrieved from BuiltWith: http://trends.builtwith.com/mapping#

Campbell, J., & Shelly, G. B. (2014). *Web Design: Introductory.* Course Technology.

Carlos, G. (2013). *Responsive Web Design with jQuery.* Packt Publishing.

Cipeluch, B., Jacob, R., Winstanley, A., & Mooney, P. (2011). *Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps.* Dublin: Environmental Research Center Environmental Protection Agency.

Davison, N. (n.d.). *Liquidapsive (Liqui-dap-sive)*. Retrieved from Liquidapsive (Liqui-dap-sive): http://liquidapsive.com/

Deering, S. (2011, November 17). *JQuery LightBox vs. ColorBox vs. FancyBox vs. ThickBox - What are the key differences?* Retrieved from SitePoint: http://www.sitepoint.com/jquery-lightbox-colorbox-fancybox-thickbox/

Falagas, M. E., Eleni, P. I., Malietzis, G. A., & Pappas, G. (2008). Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses. *The FASEB Journal*, 338-342.

Google. (2014, Auguest 1). *FAQ*. Retrieved from Google Maps API: https://developers.google.com/maps/faq#usagelimits

Hinchliffe, A., & Mummery, W. K. (2008). Applying usability testing techniques to improve a health promotion website. *Journal of Austria 19*, 29-35.

Kumar, A. (2011, October 5). *Two Best Display Components :JqGrid And Datatables.Net*. Retrieved from MSGuy: http://www.msguy.com/2011/10/two-best-display-components-jqgrid-and.html

Lal, R. (2013). *Digital Design Essentials: 100 Ways to Design Better Desktop, Web, and Mobile Interfaces.* Beverly: Rockport Publishers.

McNamara, N., & Kirakowski, J. (2006). Functionality, usability, and user experience: three areas of concern. *Magazine Interactions - Waits & Measures, Volume 13 Issue 6*, 26-28.

Nielsen, J. (2003). Usability 101: Introduction to Usability. *Jakob Nielsen on Usability and Web Design*.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *ACM CHI'90 Conference.* , (pp. 249-256). Seattle.

O'Reilly, T. (2005, September 30). *What is Web 2.0*. Retrieved from O'Reily: http://oreilly.com/web2/archive/what-is-web-20.html

Pozin, I. (2014, May 15). *Let It Go: Say Farewell To These 5 Web Design Trends.* Retrieved from Forbes: http://www.forbes.com/sites/ilyapozin/2014/05/15/let-it-go-say-farewell-to-these-5-web-design-trends/

Shneiderman, B., & Plaisant, C. (2010). *Designing the User Interface: Strategies for Effective Human-Computer Interaction:.* Reading: Addison-Wesley Publ. Co.

Social Compare. (2014, July 22). *Javascript Graphs and Chart Libraries*. Retrieved from Social Compare: http://socialcompare.com/en/comparison/javascript-graphs-and-charts-libraries

Turner, A. L. (2014, March 19). *The history of flat design: How efficiency and minimalism turned the digital world flat*. Retrieved from The Next Web: http://thenextweb.com/dd/2014/03/19/history-flat-design-efficiency-minimalism-made-digital-world-flat/4/

W3Schools. (n.d.). *CSS Safe Web Fonts*. Retrieved from W3Schools: http://www.w3schools.com/cssref/css_websafe_fonts.asp

W3Schools. (n.d.). *HTML5 Input Types*. Retrieved from W3 Schools: http://www.w3schools.com/html/html5_form_input_types.asp

Whelan, P. (2010, December 3). *Highchart Vs Flot.js - Comparing Javascript Graphing Engines*. Retrieved from Big Fast Blog: http://www.bigfastblog.com/highchart-vs-flot-js-comparing-javascript-graphing-engines