CloudStaff Technical Report

Matthew Ankerson  
27A Willis Street  
Dunedin  
027 738 6850

sm.ankerson@gmail.com

Dean Huakau   
Prince Albert Road  
Dunedin  
022 094 1257

dahuu11@gmail.com

**Abstract**

This paper reports on the development of CloudStaff - a multi-platform, flexible workplace management tool for AbacusBio Ltd. The nature of AbacusBio's business is such that essential staff are often off-site and unavailable for contact. The purpose of CloudStaff is to facilitate the timely communication of essential location and activity information amongst busy consultants, while remaining unobtrusive in the average working day.

CloudStaff consists of a full featured, touch friendly web application and a cross platform mobile application. The development of this system has enabled staff members at AbacusBio to effectively broadcast their availability, or lack thereof, and easily report when an event in the field is going to extend their absence from the office. CloudStaff reduces the amount of time spent waiting for busy staff to return or respond and allows busy staff to complete their duties out of office with a greater peace of mind.

The development process has embraced Otago Polytechnic’s Agile Development Framework and principles drawn from pure agile methodologies. An iterative development cycle has helped us embrace change and cultivated open communication between team members and Mark Teviotdale at AbacusBio. For the duration of the development term we have placed emphasis on working software and meaningful documentation. Our outputs have earned us a positive working relationship with AbacusBio and yielded a working software solution for the business problem.  
The web application was built with Microsoft ASP.NET MVC, Web API, Entity Framework and SQL Server. Front end technologies are comprised by Bootstrap, JQuery, HTML, and CSS. The cross platform mobile application is built with Xamarin Forms.

**Keywords**

**Status**: The current activity of a person associated with associated with the workplace (i.e. “On Farm. or “Out to lunch”). **Client**: Mark Teviotdale and AbacusBio Ltd. **Us/we**: Dean Huakau and Matthew Ankerson.

# Introduction

## The Problem

Our client, AbacusBio Ltd is a Dunedin based agribusiness consultancy firm who offer scientific and technologic solutions to agribusinesses. They have identified a need in their own workplace which they believe is pervasive in workplaces everywhere. AbacusBio Ltd employ agricultural consultants who often spend much of the working day out of the office. The nature of their work is such that their availability for contact frequently changes. AbacusBio wishes to have the ability to accurately identify the location of their roaming consultants, ie. if they’re off-site, and if so - when they are due back or what they’re doing. The crux of the issue is that it’s difficult to determine where staff are, what they’re doing and if they’re available for contact. Unavailability of this information leads to time wastage when a colleague or a client wishes to contact a particular staff member, only to find they’re unreachable. AbacusBio Ltd has asked us (Matthew Ankerson and Dean Huakau) to help engineer a solution for this problem.

AbacusBio has significant experience in servicing the agricultural sector and New Zealand’s other primary industries. Knowledge of these industries has led AbacusBio to recognise the benefits of applying a similar solution to meet their business needs. A vastly different work environment; for example a sheep or dairy farm, is likely to have similar basic needs regarding employee location and activity awareness. There are however some factors associated with these work environments that introduce complexity to the solution, these factors must be addressed for such a system to be worthwhile. These factors include (but are not limited to): visitor location and activity awareness, health and safety requirements of employees and visitors, and operability in geographically remote locations.

This paper will illustrate the research undertaken and the development workflow employed in delivering a solution appropriate for the needs of AbacusBio, while recognising and accommodating the requirements of other industries. Requirements of other industries are catered for by supplying relevant features now, and providing a platform for future development.

## AbacusBio Ltd

AbacusBio Ltd are a Dunedin based agricultural-business consultancy company with a range of staff roles. From their website:

*“AbacusBio are international leaders in agribusiness consulting, providing innovative tools and technology to improve food system productivity. We bridge science and business, using our technical and commercial expertise to work at all levels of the value chain, from farm to market.”* ([www.abacusbio.com](http://www.abacusbio.com))

The nature of AbacusBio’s work is varied and requires some employees to leave the office regularly for work. With time needlessly spent attempting to contact busy consultants, AbacusBio recognises the value in applying an appropriate software solution to this communication problem.

## Related Research

By reviewing and evaluating existing systems, literature and ideas in this problem space we were able to reach a higher understanding of what the problem required, in order to maximise benefit to both individuals and organisations. Certainly there are numerous potential technologies and techniques appropriate for approaching employee and visitor location and activity awareness, our focus during this review was to hone in on the most appropriate mix of technologies for our particular problem set.

### Currently Used Systems

Previously, AbacusBio was using a table drawn on a conventional whiteboard with permanent marker as a means of making an employee’s location and activity visible to colleagues. Consultants were required to update the appropriate section of the table with erasable marker each time they entered or left the building, indicating their status. It is important to note that using a whiteboard provides a non-invasive means of monitoring staff on and off site - though its information is not necessarily accurate or timely. In the case also of agribusinesses and most other sites where employee location is critical, a comparable whiteboard, paper based system or a card ticketing system is often employed or it is simply expected of managers to know the whereabouts of staff and visitors.

### Implications of Introducing Technology

Whiteboard or card ticketing systems are adequate for managing basic workplace activity; but with the evolvement of liability, health and safety, security issues, and the time critical nature of business, the solution needs to evolve as well. A conventional whiteboard falls short when a consultant’s plans change while they’re out of office - there is often no way to modify their status remotely without phone calls or text messages to managers. Additionally, the information on a whiteboard is limited to that which the employee wishes to disclose with their peers. In reviewing literature regarding ethical use of location based services, the idea of balance surfaces, as does the requirement to juggle multiple competing motives and interests. (Abbas, R., Michael, K., Michael M., 2014)

A technological solution to this problem has much to consider in terms of balancing the interests of the organisation and the interests of individual freedoms. The effectiveness of currently used systems is at the mercy of the individual and the extent to which they will embrace the system. A solution that removes some elements of responsibility from the individual and places it with the organisation will no doubt increase the effectiveness of the system and minimise discrepancies. The issue that we faced is implementing a system that doesn’t subtract significantly from an individual’s freedom.

### Value of this Project

As mentioned prior, the nature of AbacusBio's business is such that essential staff are often off-site and unavailable for contact. By developing a system that allows a staff member to communicate their status and easily report when an event in the field is going to extend their absence from the office we are able to reduce the amount of time spent waiting for staff to return or respond.

The value of any systematic workplace management tool is unquestionable. Significant benefits are realised in day to day business. With increased visibility into an individual’s work activities, the individual is more likely to exert more effort serving clients in a better manner. (Akers, 2015).

### Risks

Sturgeon and Christenberry argued that:

*“It is better to spend a few minutes a day tracking time than to spend one’s day looking for a new job due to  outsourcing”, “downsizing”, “redirection” or some other variation of not getting resources because nobody knows what workers spend their time doing.”*

This outlines the value of good worker activity tracking and the dangers of poor or insufficient worker activity tracking. The primary risk is adoption. The organisation has to convince the users that the benefits will outweigh the work required for the system to run. The advantage of a tailored software solution is such that we can mitigate these risks with clever automation and good interaction design, thus providing pertinent information to users for a relatively small investment of time and effort.

A secondary danger to most workplace management tools is a generic problem faced by most I.T technologies. It is the struggle to reach a balance between freedom and security; a particularly ubiquitous tradeoff that must be considered when designing software systems. (Abbas, R., Michael, K., Michael M., 2014) When implementing a new system, often a particular consideration competes with other concerns. In the case of our system, we have identified that the need for accurate, real-time information conflicts with the need for privacy of the individual. Again, the value of our system is realised as our solution satisfies both requirements. Rather than implement GPS location based services, our system stores and modifies employee information at the individual’s discretion. Automation makes it easier for the user, and the organisation gets timely and reliable employee activity information.

### Related Literature

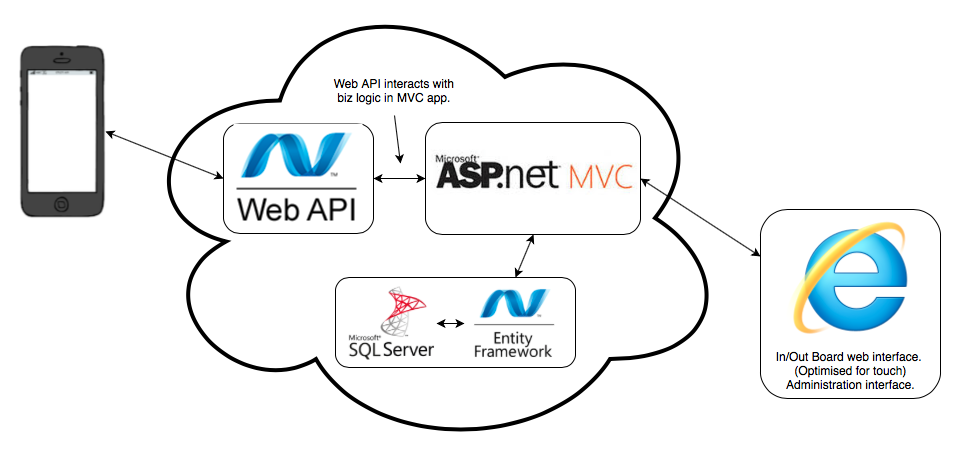
Other institutions have found value in implementing a technological solution to workplace management, even many years ago when the technology was in relative infancy. During the last few decades we have seen accelerated advancements in information systems and indeed there are several ways in which technologies embody the ability to track and locate people.

*“The first benefit to the University, therefore, is knowledge of how it’s valuable human talent is used. With this knowledge, the University can align its resources with its mission and justify allocation of resources, as appropriate.”* (Sturgeon, J., & Christenberry, J.1996)

The findings of several studies lay blame on these technologies for “the serious detriment of civil freedoms.” (Clarke, R. 2001). Implications are encountered by technology-embracing organizations, technology providers, engineering standards developers, and for policy creators. (Clarke, R. 2001). This illustrates the risks and ethical challenges that one will face when dealing with any form of human behaviour monitoring or surveillance, in addition to possible legal issues. Few laws directly address location monitoring, however ethical considerations such as privacy, accuracy, inconsistency, security, and reputation may affect future legislation (Kaupins and Minch, 2005). We (Matthew Ankerson and Dean Huakau) have had the opportunity to design a superior system, in the sense that it provides a solution to the issue wherein the trade-off exists between individual’s privacy and business advantage for the organisation by means of location and status information.

# Solution Overview

AbacusBio’s desire to maintain accurate and timely location and activity information about employees and visitors was addressed by a people management application composed of two distinct components. The core product is web based application that offers users the widest range of features. Users can view and update their status, or statuses of colleagues, add groups and perform multiple signouts in a single operation, add new or returning visitors, and several other operations designed to smooth the user interaction. The web application leverages modern responsive web design and scales to a usable state for most screen sizes. Our design goals for CloudStaff embrace versatility and effortless usability. Administrators can perform database management and general admin tasks by logging in with an approved username and password. The secondary component is a cross-platform mobile application that interacts directly with the core application to offer the most important functionality to users on the go. Figure 1 illustrates a high level representation of how the system works.



**Figure 1. High level system architecture.**

## Staff In Out Board

A streamlined tool was required by AbacusBio for maintaining staff and visitor information in a non-cumbersome, effortlessly usable fashion. The In / Out Board is the bread and butter of CloudStaff and is expected to be subject to hundreds of interactions a day. We have developed a web interface for a range of screen sizes, highly optimised for touch screens. A crucial focus was to reduce the number of distinct touches or movements required by the user to perform any operation, keyboard interactions are only required when absolutely necessary. In an effort to increase speed and responsiveness, the In / Out Board performs client side computation where appropriate to reduce the calls necessary to the server. The amount of markup is kept as minimalistic as possible to ensure fast element lookup for modifications, though it will inevitably grow proportional to the number of users represented in the screen space. To reduce page loads a single page of markup was employed, as opposed to a traditional multi page paradigm. This of course introduced a challenging battle for optimal screen real estate utilisation, though as previously mentioned - with the application of modern responsive web design the application renders sensibly for most screen sizes.

Perhaps the most challenging screen space optimisation was representing all pertinent user information in the small ‘modules’. Each user has one module, displaying the following information: full name, status, location, availability, return date, return time, and whether or not the user has exceeded the time they had allotted themselves for a particular status. Verbose textual information was quickly abandoned in favour of colour coding and clever logic for representing information, ie. only applicable information for each user is shown.

## API For Remote Communication

With operability in remote locations a core requirement for CloudStaff, we knew we needed to provide a means of communicating with the core application. Early in the project it was decided that any external mobile application needed to interact with the same datastore (and indeed the very same business logic) as the core application if we were to achieve consistency across all platforms. We have developed a public facing RPC (Remote Procedure Call)  style API with industrial strength token authentication, namespaced within the same core application.

Security was a primary concern for this API, remote applications form a HTTPS request to the server with an existing user’s username and password, the appropriate user id and API token is sent back to the remote app for subsequent requests. When the next request is received, the token is pulled from the HTTP header and hashed, then compared to the hash stored in the database.

## Administration

A toolset was essential for administration staff at AbacusBio to configure the content and options available to general users. From the outset this was deemed to be crucially important to the value of the application as a whole. Having the ability to easily perform this customisation effectively allows administration and human resources staff to fine tune the product to the organisation’s changing needs.

In addition to customisation of content and options, the administration interface also offers a means of performing general CRUD style operations on the database - ie. Reading exisiting information, Creating, Deleting and Updating users, adding or updating authentication information, modifying groups etc.

As some of the tasks performed by the administration module are quite complex and require a lot of user input, we spent a considerable amount of time developing interface features to reduce complexity. Client side logic makes calls to the server in order to auto populate fields according to information already saved. Client side preventers go some way to ensuring valid input, but in the event of an invalid input, the server will return a friendly message detailing the problem encountered.

## Cross Platform Mobile Application

The Mobile App was developed with similar design principles as the core product, though the required functionality was greatly reduced, i.e no administration requirements. Once again, ease and simplicity of use was a major factor. The app allows the user to view and edit their own status, location and return time remotely and allows for updates to the same data store behind the In / Out Board.

Once the user is logged in, their current status is displayed on the home screen along with an edit button. Upon pressing the edit button the user is taken to a screen with all three editable fields. These fields display as drop down lists and simple time and date pickers to negate the need for any typing.

Once these options are submitted the user returns to the main screen where the updated status is displayed for assurance.

Because of the nature of mobile application development, asynchronous frameworks are often employed to implemented efficiencies, especially around any network access which is typically very slow.

*“In the last few years Asynchronous programming has become more popular because it helps to solve issues encountered with Synchronous programming which makes the user experience smoother and feel more responsive. Unfortunately using an Asynchronous approach has its own drawbacks. The development of applications is more complicated and can require more time to implement the needed functionality.”* (Working Synchronously or Asynchronously?, n.d.)

The need to program asynchronously of course arose when we needed the app to interface with our RPC (remote procedure call) API.

In order to maintain consistency, an AbacusBio ‘theme’ was designed with small icons on the buttons and ‘AbacusBio blue’ being  used as separator lines between the input fields and as the text color on the buttons.

# Technology/Platform Justification

Requirements were imposed on the development of CloudStaff to use Microsoft technologies for all server side computation, data storage etc, thus CloudStaff has been developed in the Microsoft .NET paradigm.

## ASP.NET MVC

*“ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns”* (The Official Microsoft ASP.NET Site, 2015)

Microsoft’s ASP.NET Model View Controller framework was employed to facilitate the flow of information between the core product’s user interface and the data model. We selected this technology due to its unquestionable popularity and particular suitability for agile development. A framework like ASP.NET MVC certainly introduces some magicor obscured complexity to the code base, however this drawback is overcome by the ease of development when adhering to *convention rather than configuration*.

## MS SQL Server and Entity Framework

When working with a Microsoft stack there is no relational database that better integrates than MS SQL. When architecting our solution we identified the suitability of a relational database from the outset. Technical advice suggested a document database as an alternate, though AbacusBio was not in favour of a NOSQL implementation.

Although using Transact-SQL is a perfectly reasonable means of interacting with MS SQL Server, development veered towards the use of Microsoft Entity Framework and LINQ (Language Integrated Query). Entity Framework is an object-relational mapper which translates relational entities to plain old CLR objects or vice versa. There are several workflows with which you can use Entity Framework, of which we chose *code first*. This workflow involves building a data model with C# classes which then generates the underlying SQL schema. The model is queried using LINQ.

## ASP.NET Web API

*“ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices.”* (The Official Microsoft ASP.NET Site, 2015)

CloudStaff required a means by which a mobile application could communicate with the main business logic and datastore. Microsoft offers several API building services including Windows Communication Foundation, we opted for Web API for it has become the standard in this paradigm and provides routing configurations for either REST or  RPC style APIs. After seeking technical advice we decided to implement a Remote Procedure Call style API which leads external applications to interface with what’s essentially a set of named functions, very similar to MVC style endpoints.

## JQuery and JQuery UI

*“jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.”* (jquery.org, 2015)

JQuery provides a natural extension to JavaScript and during development was used extensively for event handling, DOM manipulation, Ajax, and of course it’s a dependency for Twitter Bootstrap and most touch enabled controls. The In / Out Board is especially reliant on JQuery for Ajax calls needed to fetch data for various operations at the user’s request. Any autocomplete or pre-populated field is enabled by JQuery Ajax, including querying for appropriate time ranges when users specify a return day for their current status.

## Twitter Bootstrap

Bootstrap is a sleek, intuitive front-end web framework for fast and easier web development for all screen sizes. (Getbootstrap.com, 2015)

Twitter Bootstrap comes bundled with ASP.NET MVC 5 and during early development and we decided this would be an ideal platform on which to build our responsive web application. Bootstrap is fast, facilitates rapid UI development and developers can use as little or as much as required. For CloudStaff, all web interfaces have been built with Bootstrap, and all DOM manipulation injects elements that maintain consistency with Bootstrap classes. Using this framework ensures maintainability of the web interfaces, if a new developer knows Bootstrap then the UI will be easier to reason about and extend because it adheres to convention.

## Xamarin Forms

Xamarin Forms is part of the Xamarin Studio cross platform application development paradigm. Xamarin enables the developers to write in one language, C#, for Android, iOS and Windows Phone. Now that Xamarin has collaborated with Microsoft, developers can now use Visual Studio to develop for all three mobile platforms.

The client suggested the use of Xamarin Forms for this application as they had a paid license for Xamarin Studio. AbacusBio supplied us with ‘seats’ for our development machines. Our development machines are Windows computers with a networked Macintosh for building to iOS. Xamarin Forms was thought to be a good fit for our needs because

*“Xamarin.Forms is best for: Data entry apps, prototypes and proofs-of-concept apps that require little platform-specific functionality and apps where code sharing is more important than custom UI”* (Xamarin, 2015).

Xamarin Forms fitted our requirements well.

Other choices on offer were Xamarin iOS and Xamarin Android. These required separate coding for separate Platforms.

*“Xamarin iOS and Xamarin Android are best for apps that require specialized interactions,apps with highly polished design, apps that use many platform-specific APIs and apps where custom UI is more important than code sharing”* (Xamarin, 2015).

It was decided to first use Xamarin Android to develop a very small android prototype and get a ‘feel’ for the application. This worked out well so the move was made to Xamarin Forms. Xamarin Forms has the ability to run an iOS emulator on a networked Mac from a Windows machine. (Also possible in the reverse).This was extremely beneficial as one could first run an android emulator on Windows and then run the iOS simulator on Mac. So that one could view any unexpected discrepancies that arose between the platforms.

# Development Process

The development of CloudStaff followed Otago Polytechnic’s Agile Development Framework. This development methodology is a mix of pure Agile principles and traditional waterfall development principles. The Agile Development Framework applies iteration to a structured process, designed to help teams embrace change and facilitate more effective client communication. Three iterations are completed during the development term, the end of each iteration sees a working deliverable and each iteration builds upon the last. Throughout each iteration our team maintained constant communication with the client and made use of feedback received from users at AbacusBio.

Dean Huakau and Matthew Ankerson have had an effective working relationship with Mark Teviotdale and AbacusBio, all parties have maintained a good common understanding of CloudStaff’s requirements, direction and ambitions. Staff at AbacusBio have embraced the project and provided valuable feedback for the development and refinement of features.

Scrum meetings are an effective tool from the Agile methodology that enabled us to voice concerns and development roadblocks everyday within the team, with the intention of leading the project in the most productive direction. During development we engaged in countless informal scrum meetings which ensured visibility into team member’s contributions and ultimately drove the project to success.

Project source code was fully versioned with git -

*“a fast, scalable, distributed revision control system”*. (git man pages.)

The project repository features three branches: ‘master’, ‘matt’ and ‘dean’. Both Dean and Matt regularly commit to their respective branch, with builds merged into ‘master’ when deemed to be stable. The repository was frequently pushed to GitHub for collaboration, viewing for reference and enabling peers to easily provide code reviews. An added benefit to hosting on GitHub was having the project and entire commit history backed up for safety.

For the duration of the first iteration we formed a group management plan, focused on articulating the value of the project and decided upon functional requirements. Time and effort were spent building a shared knowledge base amongst the team and the client. With the ambitions and requirements identified, the project faced a panel of industry professionals at Dragons Den. Industry professionals shared a positive opinion of the project, though raised concern regarding the potential availability of existing solutions. Our response was a clear articulation of the specific shortcomings of existing workplace management systems when compared to our proposed system, with which the panel was satisfied.

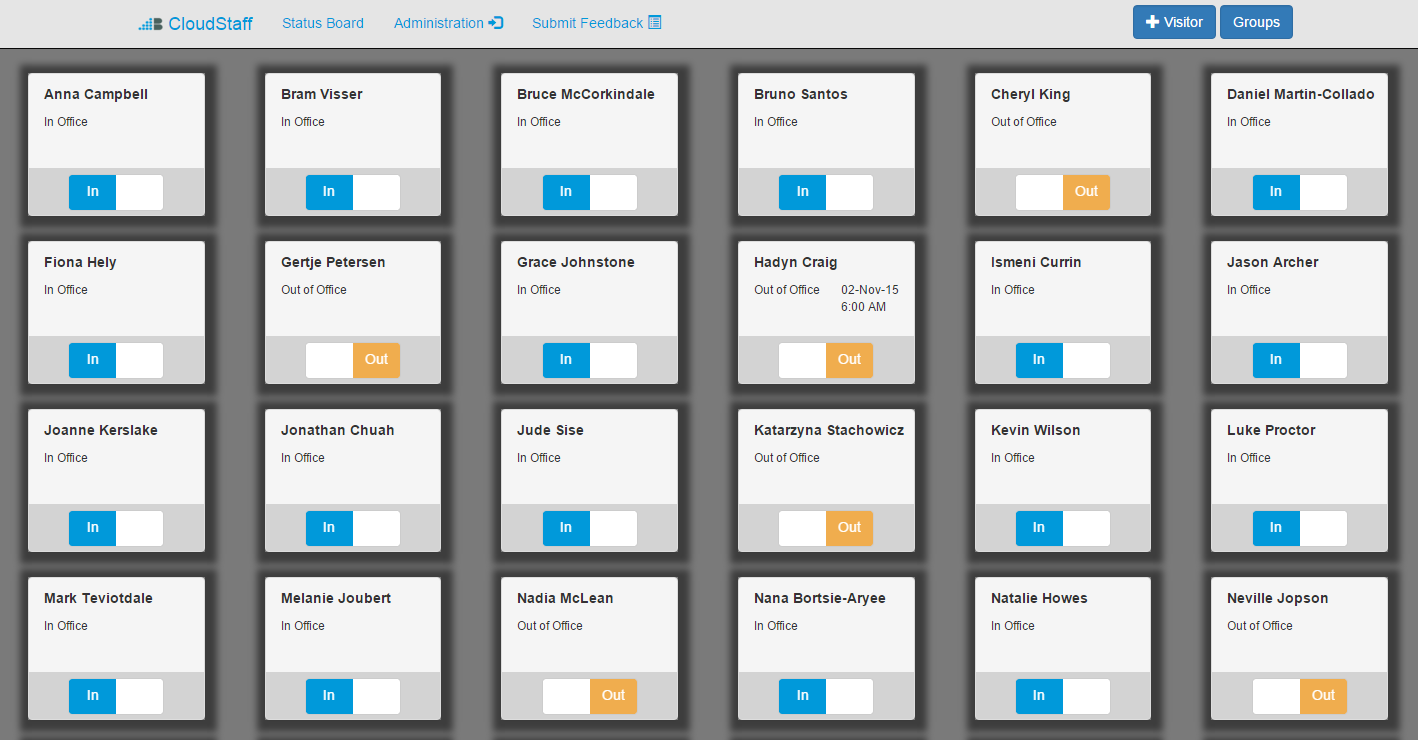
The second iteration involved lengthy research and thought about what to build to solve the business problem, without introducing any cumbersome processes to the staff’s working day. At this point in the development process there was little else to do but start coding. It was essential that we start building the core features in order to get user feedback as early as possible. The result was a robust minimum viable product. The software performed only a few tasks, but it performed them well and without error. We deployed the first version to AbacusBio in early August. Although the minimum viable product was positively received; the real win came with exposure to staff at AbacusBio. The minimum viable product was motivation and a platform for discussion about what features, improvements and ambitions should be encompassed by future releases. Mark Teviotdale and the software development team at AbacusBio were impressed and indicated that they’d like development to continue.

The third iteration ran for the entire second half of the year and involved a considerable amount more tangible development. During this iteration we took exceptional advantage of Agile principles, completing many ‘mini’ iterations, involving a development sprint, reflection and planning for the next sprint. Sprints were a good way to squeeze development time into weekends and stretches during weeks when course load was more minimal. By the end of the third iteration we had addressed all initial functional requirements, plus a handful more. The system now supported return dates and times, visitor sign in and sign out, groups of users and a myriad of useful administration tasks. The API was fully implemented and secure, with the cross platform Xamarin Forms application successfully making calls to update the database via the API. The In / Out Board also received a major look-and-feel refresh, in response to feedback from users and the recommendation from Mark to use AbacusBio’s brand colours. Throughout this process we were able to perform several deployments to AbacusBio, each proving to be valuable for user testing and feedback while inching closer to the initial functional requirements and ambitions. By the end of September CloudStaff was in full use by the staff at AbacusBio with development continuing in an iterative fashion.

# Artifact Description

## Interaction Design

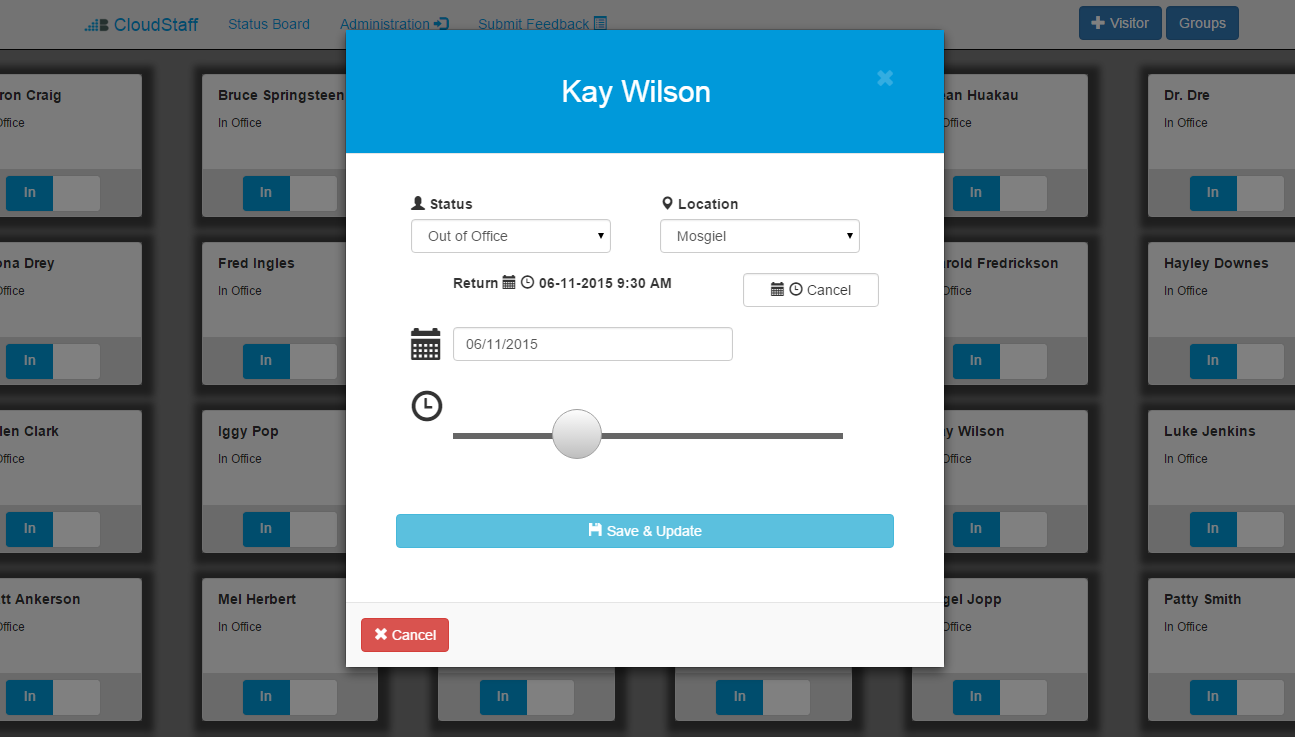
As previously mentioned, one of the primary non-functional requirements of CloudStaff was for it to be *effortlessly usable*. With the In / Out Board being subject to hundreds of interactions every day in even a small organisation, it was imperative that we engineer interactions to be unnoticeable or *invisible* to users going about the average working day. Certainly there are some operations that will always require a keyboard, ie. adding a new group or visitor, however these are unavoidable and less frequent than other operations. To support these operations in the most streamlined manner, the application should be loaded on a touch enabled device, where a soft keyboard is available. AbacusBio have purchased an all in one touch screen PC for the foyer of their office which enables these features without the cumbersome intrusion of a hard keyboard.



**Figure 2. In Out Board.**

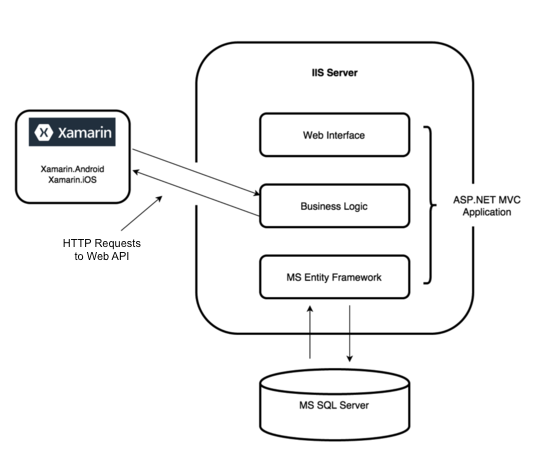
Users can perform a basic sign in / sign out operation with one touch, or if they want to specify more information about their status, they can touch their name for more verbose options.

One of the fundamental goals of the user experience was to allow users to specify as little or as much information as they like, the In / Out Board hides verbosity until it is asked for.



**Figure 3. Selecting status, location and return time.**

## Software/Systems Architecture



**Figure 4. Software Architecture.**

CloudStaff utilises the Model View Controller design so many architectural decisions had already been made for us. Perhaps the most interesting section of our architecture is our business logic. Early in the second iteration we identified the need to separate our logic code from the controllers. We built a suite of static classes composed of static functions which simply accept some input if any (ie. user IDs etc), and return some output from the database (if any). One of the design principles of MVC is separation of concerns, thus we felt it was appropriate to remove business logic code from the controllers and encapsulate it in utility classes, each utility class mapped to a model class. External applications call API endpoints using HTTP, which in turn calls functions of the static utility classes - the external application to interface with the core application with true consistency.

## Functional Requirements

*1. The system shall allow staff to communicate their status with their colleagues. (e.g. “out of office” or  “on farm” or “on leave”)*

The In / Out Board and the mobile application both allow staff to communicate this information. If a staff member is seeking this information they can view via the In / Out Board or the SharePoint view.

*2. The system shall allow staff to view the status of their colleagues.*

Staff members can view this information via the In Out Board or the SharePoint view.

*3. The system shall allow staff to modify their status remotely.*

Staff can modify their status remotely using the mobile app, or by visiting the web application (In / Out Board) using any web enabled device.

*4. The system shall hold basic information about each staff member.*

CloudStaff maintains basic information on each user pertinent to the operation and value of the system. User profiles can be customised to allow specific users have unique statuses and locations, or be included as members of groups. There is no limit to how the user profile could be extended. If an organisation’s requirements encompassed the need to store health and safety information or qualification information for each user, CloudStaff could cater for this with a trivial extension of the user profile.

*5. The system shall allow visitors to sign in and record information about their visit.*  
Using the In Out board visitors can record themselves as new visitors, and see themselves represented with a user module in the main interface until they decide to leave. Returning visitors can retrieve their existing profiles and sign in with fewer keyboard interactions.

*6. The system shall allow users to form groups and perform sign in / sign out in a single operation.*

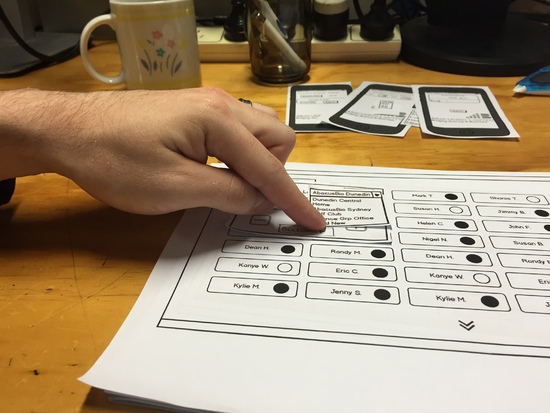
Group functionality is provided with the In / Out Board and maintained in the Administration module. Any user can add a group and perform a sign in or sign out. When signing in or out, a user can refine the members who comprise the group at that time. Administrators can edit group names, priorities (for display order) and members from the Administration module.

# Technical Highlights

* Built a functional multi-platform mobile application utilising native libraries on each platform and a single C# codebase.
* Implemented industry standard authentication and encryption for password and API token hashing. Built upon this is a secure login system for the administration interface and a registration process for new users of the API.
* Designed and built a data model to optimise query speeds for commonly performed operations.
* Designed user interfaces which new users almost instantly understand and engage with, while incuding optimsations for experienced users to perform operations quickly.

# Testing

Testing began early in the second iteration with paper prototyping. We used this testing framework to build an understanding of how people perceive and interact with our proposed user interfaces. There was confusion about the terms used in the user interface at that time, testers were unclear about what “absence” meant in that context. It was discovered that our initial controls for date and time selection were completely beyond comprehension for nearly all testers, this ensued major change for our status selection UI. Comments were also made about the cumbersome nature of adding new statuses with the In / Out Board. This was a feature that Mark expressed interest in at the initial project briefing, though following these comments it was decided to separate the functionality and add it to the administration module.



**Figure 5. Paper Prototyping.**

With development underway this was the time at which some of CloudStaff’s most critical functions were written. Unit tests were written for functions pertaining to the core model class, user profiles and most importantly - authentication and encryption. After a few weeks more, the In / Out Board received a major look-and-feel refresh, in the theme of AbacusBio - and redeployed to their office. By this time we’d configured our staging deployment platform at Polytechnic, this allowed us to show the product to our colleagues, friends and family. We prepared an open-ended feedback survey form using the online tool Google Forms, we embedded a link to the feedback form in the navigation pane of the In / Out Board and shared a publicly accessible link to our project on Facebook. Surprisingly positive and thorough feedback was collected during this week. Our first users didn’t get much functionality, but they could easily determine what CloudStaff was trying to achieve which served as a good motivator for direct and useful feedback. The feedback identified bugs, clumsiness with page refreshes, blandness of colour scheme, legibility issues, button slider and time slider affordances. Users at this time also identified the need for date selection, which was not yet implemented.



**Figure 6. An early version of CloudStaff displayed on a touch screen computer at AbacusBio’s Office.**

Now that CloudStaff has been in actual use at AbacusBio for over a month we’ve been receiving valuable *real use* feedback, straight from the staff actively using it. Many small features and refinements are a result of the iterative process which comprises active use, feedback from users, implementation of suggested features, and reflection. Some of the features and improvements that have resulted from this process include better screen space utilisation, refinement of touch controls, date formatting and disambiguation of returning visitors.

# Deployment

The CloudStaff web application has been deployed to a Windows server at AbacusBio where it is currently fulfilling its intended business requirements. Our deployments consist of two levels, a staging environment at Polytechnic and a release environment at AbacusBio. The web application is hosted in IIS 8 (Internet Information Services) and requires an instance of SQL Server 2014 for the database. Deployments leverage Visual Studio’s multiple configuration functionality. Development, staging and release configurations all use differing database connection strings and slightly different setup. In our development environment we have a collection of functions that run on start up which drop, create and populate the database for fresh testing every time. Although in staging or release, the database is only created and populated if it does not already exist. Our staging environment also seeds different data, as we didn’t think it was appropriate to use production data outside of AbacusBio.

The cross-platform mobile app has not yet been distributed due to difficulties in identifying an appropriate deployment strategy. The app can be trivially distributed and installed on Android devices, though iOS devices require further investigation. Xamarin documentation suggests a deployment strategy called the “Ad Hoc Distribution” (developer.xamarin.com). This allows an organisation to distribute the application to at most 100 iOS devices, without the need for iTunes. Although this may be sufficient for AbacusBio’s requirements, it is a deployment strategy intended for in-house user testing and would not be suitable for the extended life of the project. Should the development of CloudStaff continue, a more realistic means of deployment from Xamarin studio to iOS devices needs to be researched and implemented.

# Analysis

CloudStaff addresses AbacusBio Ltd's problem with maintaining accurate, timely information about the location and activity of their roaming consultants. The nature of AbacusBio's business is such that essential staff are often off-site and/or unavailable for contact. CloudStaff allows staff members to communicate their location and activity with colleagues, staff can effortlessly share information about changes in their working day.

In addition to facilitating communication of employee activity, the system also tracks visitors to the business. This allows for visitors to be easily identified and accounted for. In the event of an emergency the system knows precisely who is on-site.

Our solution has been in use by AbacusBio since September 2015. During this time AbacusBio have reported value in terms of knowing the current location and activity of their personnel and the ease of information distribution. In addition to our deployments at AbacusBio we have had the opportunity to explore additional value in offering a similar or extended system to workplaces of other industries, particularly farms, fishing boats and building sites. It is AbacusBio’s intention to use CloudStaff as a basis for building a more capable system, more suitable for these diverse work environments.

# Future Work

AbacusBio plans to continue development on CloudStaff to work on features critical for other work environments, ie. farms, forestry, fishing etc. It is conceivable that we could also fork the project and work in other areas that we see fit. The following future development possibilities highlight the features that we believe would be of maximum benefit to the system.

## In / Out Board

1. Replace keyboard controls when adding a visitor with controls that are both touch and keyboard enabled, despite touch being a somewhat cumbersome method in the context of adding visitors. This feature is more important in some environments than others as some operating systems are better optimised for touch. (Operating systems like iOS and Windows 8.1 offer a soft keyboard for input fields.)
2. Add searching functionality, so a user can refine the grid of names in search of their own. Perhaps this could be achieved by offering users a list of alphabetic ranges (similar to a supermarket self checkout) which would intuitively facilitate fast name lookup.
3. Reduce reliance on page refreshes, particularly when submitting forms on modal dialogue windows.

## Administration

1. Provide functionality for searching and filtering the change logs for statuses and locations.
2. Build an interface for viewing, searching and querying the visitor logs. CloudStaff stores quite verbose visitor records in the database, though these records are never fetched, except for autocompletion for a returning visitor. It was ruled out as a requirement for CloudStaff though it would be a worthwhile addition nonetheless.

## Cross Platform Mobile Application

1. Enable support for low coverage areas. (Cache requests until service becomes available)
2. Persistent storage on devices. This proved to be a problem with Xamarin Forms, finding a cross platform solution was beyond our reach. For the mobile app to be truly complete it requires persistent storage for the user ID, api token, and all current settings for the user. It would not be unreasonable to build the mobile app natively in each device’s respective language, given that the implementation would be fairly straightforward. There are certainly some gotchas to Xamarin Forms, though that does not discredit the tool for the job it was intended for.

# Conclusion

Matthew Ankerson and Dean Huakau have developed the software necessary to address AbacusBio Ltd’s need for accurate activity and location information for all members of staff. This need has been addressed by designing, implementing and testing: a responsive, touch friendly web application, a secure web facing API, and building a cross platform mobile application to offer the basic services of the core product while operating in remote areas.

The development process followed Otago Polytechnic’s Agile Development Framework which facilitated iterative development and strong communication with the client and users. Additionally, this development framework yielded rich documentation and a working product for AbacusBio to use and build upon in future.

# References

1. Akers, Helen. 'The Advantages & Disadvantages Of Employee Monitoring | Ehow'. *eHow*. N.p., 2015. Web. 17 Mar. 2015.
2. Abbas, R., Michael, K., & Michael, M. (2014). The regulatory considerations and ethical dilemmas of location-based services (LBS): A literature review. Retrieved March 15, 2015.
3. Getbootstrap.com,. (2015). *Bootstrap · The world's most popular mobile-first and responsive front-end framework.*. Retrieved 31 October 2015, from http://getbootstrap.com/
4. Clarke, R. (2001). Person location and person tracking: Technologies, risks and policy implications. *Information Technology & People, 14*(2), 206-231. Retrieved from<http://search.proquest.com/docview/222415271?accountid=39660>
5. jquery.org, j. (2015). *jQuery*. *Jquery.com*. Retrieved 31 October 2015, from https://jquery.com
6. Kaupins, G., and R. Minch. 'Legal And Ethical Implications Of Employee Location Monitoring'. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (2005): n. pag. Web. 17 Mar. 2015.
7. The Official Microsoft ASP.NET Site,. (2015). *MVC*. Retrieved 30 October 2015, from <http://www.asp.net/mvc>
8. The Official Microsoft ASP.NET Site,. (2015). *Web API*. Retrieved 31 October 2015, from http://www.asp.net/web-api
9. Microsoft. (2015). What Is RPC?: Remote Procedure Call (RPC). Retrieved from <https://technet.microsoft.com/en-us/library/cc787851(v=ws.10).aspx>
10. Sturgeon, J., & Christenberry, J. (1996, November 14). Project And Employee Time Tracking For Information Systems ... Retrieved March 15, 2015, from http://the-forexjournal.rhcloud.com/link/project-and-employee-time-tracking-for-information-systems/
11. Orfano, Finn. 'The Pros And Cons Of Employee Monitoring'. *Bright Hub*. N.p., 2015. Web. 17 Mar. 2015.
12. Working Synchronously or Asynchronously?. (2015). Retrieved from <http://docs.dataabstract.com/Introduction/WorkingSynchronouslyOrAsynchronously/>
13. Xamarin. (2015). Xamarin.Forms - Xamarin. Retrieved from <https://developer.xamarin.com/guides/cross-platform/xamarin-forms/>