Placement Report

BY JAN P.C. HANSON

2016
Student ID: 1339404
Sandwich Year

Executive Summary

This report details the responsibilities and activities I undertook as a Software Engineering intern at Tomo Motor Parts Ltd commencing 28th September 2015; focusing on my professional development and achievements during the course of the placement. The success of the placement is measured by the S.M.A.R.T objectives (*see section 9.1 & 5*) as well as the Testimonials (*see section 7*) provided by my colleagues and managers.

The most prominent aspect of this placement is the design and development of a system semi automatically manage the sale of car parts on eBay using eBay's Java API, this project covered a wide range of technical skills: implementation of a MySQL database, database design, JDBC, concurrency, connection pooling, thread pooling, SOAP API's, XML, design patterns, dynamic HTML using Java and Javascript as well as much more.

I also cover the role I took in the requirements gathering and specification formalisation for the inception of a new warehouse management system, as well as the role I took in testing and feedback.

Figure 1. Tomo Motor Parts Ltd logo

Figure 2. Brunel University Logo

Table of contents

List of figures

List of tables

1 The Company

Tomo Motor Parts Ltd was formed in 2005 by Darren and Dean Perkins with the aim to provide genuine manufacturer car parts at wholesale prices. Tomo buy their stock direct from the manufacturers (in most cases), and have a very close relationship with PSA(Citroen/Peugeot).Because of the quantity of stock they purchase from manufacturers they get quite a substantial discount, which they can then pass on to their customers. The majority of their customers are trade customers: body repair shops, garages and other parts dealers including Euro Car Parts. Although they also sell to consumers through their on-premises counter and eBay shops.

Over the past 16 years they have built up their customer base and business to the point that they made a turnover in the region of £7.5M last year, and a profit in the region of £1M [9] This is soon to increase further after signing a deal they negotiated with PSA to become a dealer hub [8], after which Tomo will distribute all parts direct from PSA within the M25 and out to Oxford and Basingstoke. This will conservatively increase turnover by £4M and bolster expected profits to around the £1.5M mark.

Tomo are classified as a small company and employ just over 30 individuals (to date), including: office staff, warehouse staff and drivers. Accounting and IT are handled by consultants who work on-premises on a more or less full time basis.

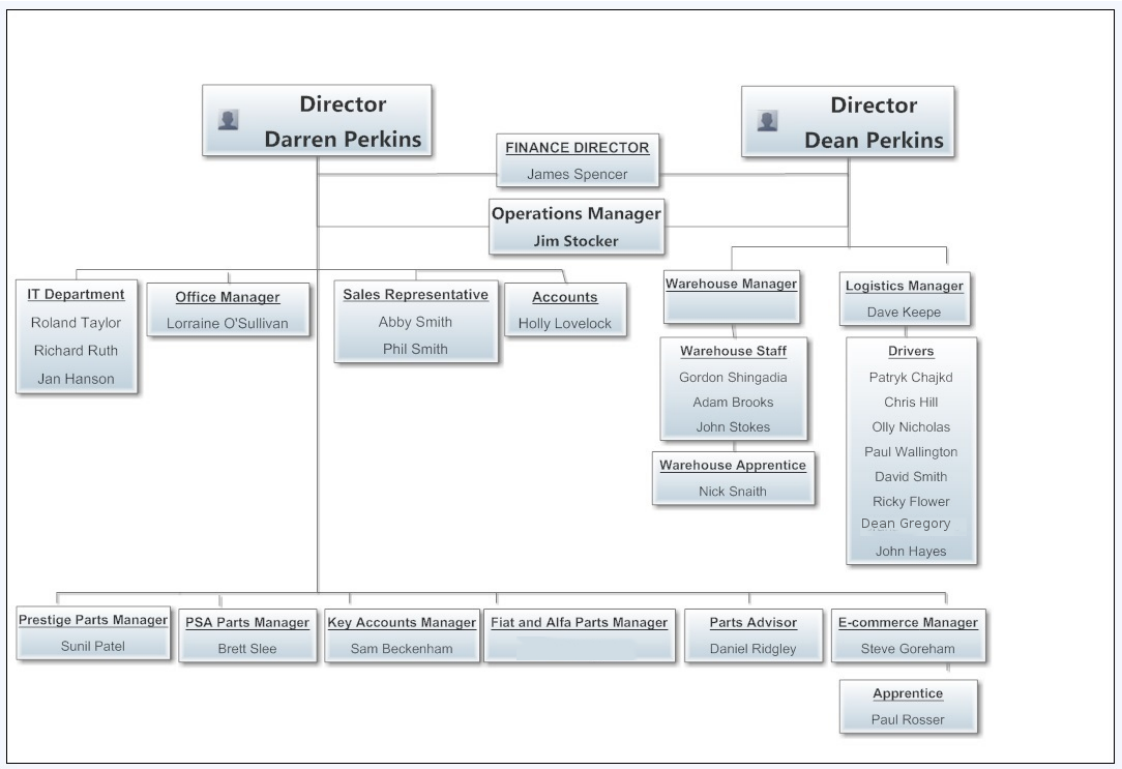

Figure 3. Company Organisational Chart

Tomo have recently acquired a new building and are looking to move their offices there later in 2016, this will help to alleviate the growing lack of space within the warehouse and the office. Accommodating for an increase in business due to the PSA hub deal as well as the accompanying office staff necessary to cope with the increase in business.

2 The Placement

I have taken on a role of software engineer [10] within the IT department. The IT department at Tomo Motor Parts Ltd consists of 3 employees, this is 10% of the overall workforce. The IT department provides infrastructure and support for the company including:

- Email Services
- File Servers
- Web Services
- VPN services
- Network Infrastructure
- Setup of new users
- IT infrastructure purchasing

Tomo is fairly unique with respect to its software usage, in so far as; most companies in this business use proprietary third party software to manage their business needs, Tomo has built its own ERP from the ground up over the last 15 years and this gives them the flexibility to modify their software to fit with their requirements as and when they change. This software (called WINSTOCK) was developed and is maintained by a single programmer, who is at the time of writing this report 65 years of age. Winstock has become a relatively large system and, with the recent expansion of the business, Tomo feels that having an extra programmer on board would help lighten the load.

The original remit of the placement was to aid in the expansion and maintenance of Winstock. Winstock is written in C++ and is heavily integrated into the Microsoft Windows infrastructure. The preliminary discussions revolved around moving new functionality away from this to a more modern language and more generic way of doing things, to make it easier for any programmer hired in the future to take over from the current engineer (should he wish to retire at some point etc), we discussed using Java and the JNI to interface with the current code but it soon became obvious that this would not fulfil the requirements and a full rewrite of the software was necessary.

Obviously this was too much to ask a single programmer to do in the time frame that Tomo wanted (6 - 8 months), So the project was contracted out to an external software company Cevo Studios, to complete. I was heavily involved in initial discussions and requirements gathering; advising Tomo Parts on technologies and feasibility.

In January 2015, Tomo Motor Parts started selling parts on eBay and it soon grew to the point where two full time employees where needed to manage and fulfil all the orders, and within a couple of months (at its current growth rate) this would no longer be enough; which is the point at which I joined the company. So I took on the development of a system to automate, as much as possible, the management of the various facets of the eBay process.

Being part of the IT team I was also involved in support as well as development, helping users with general IT related issues, installing and setting up new hardware, improving and streamlining existing systems as well as the sourcing and purchase of required hardware. I undertook such tasks as:

- Updating the ageing PPTP VPN service to an OpenVPN implementation on a Linux environment.
- Setting up Windows Server 2012 environment for a Java, MySQL development/production stack.
- Advising the Infrastructure technician with regards to the setup of the new server cabinet.
- Upgrading hardware for users (and building PC's) as the need arose.

3 The Projects

During my time at Tomo Motor Parts Ltd I have been involved in two Major projects. The largest of these projects: the architecting and development of the eBay management system (TomoBay) is where I spent the majority of my time and energy. From the initial requirements gathering and design of the basic system, through to implementation and testing and then extension and maintenance.

I was also responsible for the gathering of requirements for a new warehouse management system that Tomo commissioned, as well as aiding in testing and being a general point of contact.

3.1 Warehouse Management System

See appendix 9.1 and section 5.7 for a complete description.

3.2 Product Photography

Early on in my placement the managing director learned that I have an enthusiasm for photography and asked me if i'd like to upgrade their current product photography setup, as well as train the current member of staff responsible for the product photography in the use of the manual camera settings.

Their initial setup was a lightbox, desk lamps and an ageing Nikon D2H (4 Megapixel) camera, and because the member of staff responsible for the product photographs was not familiar with the use of the manual functions of the camera; he was limited to using the auto features of the camera, which were producing far less than optimum results.

I was given a budget of £500 to improve the setup with, the first thing to do was sort out the equip the new setup consisted of:

- Panasonic G5 Camera body
- Sigma 30mm lens
- extension tubes
- 1250W 5500K softboxes with stands
- Tripod
- A1 Paper

The next step was start training the member of staff in their use. I organised a series of eight training sessions, covering the different aspects of the camera, lights, and how to use them to get better product shots. The difference in the quality of the product photography was noticeable and as a result looked much more professional.

3.3 eBay Management System (TomoBay)

In January 2015 Tomo Motor Parts started selling car parts using eBay as a sales medium. Over the course of the year (up until September, when I joined) eBay business had quickly picked up to the point where the one person assigned to it could not keep up with the volume of orders, at this point another person was added to the eBay team. By the time i joined Tomo Motor Parts, however, demands of manually dealing with every order: picking/packing, invoicing, labelling, shipping, etc had become too much for the two people on the team to handle.

It was decided that I would work on a project to attempt to automate and streamline the paperwork side of eBay as much as possible. The primary goal of this was the management and centralisation of orders from multiple accounts and invoice generation and printing for these orders.

My first port of call was to look at the eBay developer documentation and find out what the eBay API was actually capable of. As it turns out the eBay API allows the user to perform almost any task that can be performed using the website as well as doing tasks and finding information that are not possible using the website.

The downside to the eBay API (as it currently stands) is that, while the documentation covers every class in the API, it is sparse at best; with many methods having cursory descriptions that tell you little more than the method name did. This, coupled with the lack of community documentation, made life mildly frustrating to start with; as getting API calls to work was a process of educated trial and error, which in many cases was not consistent between different API calls. However afterwards I had a more detailed idea of what was feasible.

The next thing to do was to try and identify some kind of specification for the project. This was easier said than done as neither the potential users of the eBay system, Managing Director nor the IT Manager had a concrete idea of what they wanted from the project. The two things that were consistent were that firstly: orders had to be grabbed directly from eBay and the system I was to build would invoice these orders followed by printing the invoiced orders and their associated packing lists; and secondly: the system would be web based (i.e. no desktop application).

So given this minimal set of requirements, I set to work trying to produce a system that would do as they asked, but was easy to extend without becoming a tangled mess of code. The first thing I decided on was to use the Jetty embedded web server as the display engine for frontend. This had multiple benefits: no external web server necessary in order for it to work, complete control of all settings from within the server program and lastly the whole program could be wrapped up into one neat binary and some associated files (configs/logs/keystore) making deployment and server management easy.

The second major decision with regards to the design of the system was to use a modified Model-View-Presenter style architecture (I shall explain the 'modified' qualifier shortly). The Model would hold all the functionality to directly interact with the data sources (eBay API/database/Winstock/etc) as well as external systems such as the Javamail/Gmail interactions; and taking a leaf out of the Service Oriented Architecture way of doing things, i decided to include in the model a package for services.

In the context of this project, I defined services to be one of two possible things:

1. Background tasks that would need to happen periodically in order to maintain the system, such as syncing orders from eBay, checking the database for errors and so on.

2. Functionality the user chooses to invoke that would be substantial enough to cause delays to the system.

The reason for classifying the above as services as opposed to having it fall under some other area was that the background tasks referred to in (1) are (in general) fairly lengthy processes in terms of execution time as are the items of user invoked functionality referenced in (2). If these were carried out in the main program thread could lead to user perceivable slow downs, which should be avoided where possible.

I decided to use Java's built in 'executor' framework to create two distinct types of threadpools in order to cope with the two distinct types of services: the first was a threadpool for 'scheduled services', those services which come under the remit of the first definition; and a threadpool for 'triggered services', those services that fall under the second definition.

4 Links to Academic activity

In this section I would like to take a moment to provide some information about my 1st & 2nd year courses and how the information that I was taught in them relates to activities I performed on my placement

Group Projects	This module acclimated me to working with people of different skill sets and skill levels and aided me in communication between Tomo Motor Parts Ltd and external developers used for a variety of tasks.
Logic and Computation	Provided me with a background in the lower level aspects of how computers and programming languages work as well as formal logic. All this stood me in great stead allowing me to write much better programs and giving me more tools to solve problems with in general.
Data and Information	Gave me a background in the theory behind relational databases as well as statistical analysis. This came in particularly useful when designing the database backend and processing the data that is retrieved from eBay before storing it in the database.
Information Systems and Organisation	The content of this course was particularly useful while gathering the requirements and writing the specifications for the warehouse management system. The content regarding use cases in particular, and whilst 'use cases' where not used as part of the specification (in any formal sense), the information that they would have contained was kept in mind when putting the specification together.
Introductory Programming	Unfortunately this course was of limited use to me as I had covered all the material in this course before starting at Brunel University, and was already comfortable with the concepts and techniques contained in this course.
Software Development and Management	This course provided me with knowledge about a range of tools that are exceedingly useful in a software development context: Doxygen for code documentation, Git for version control and backup. It also made me familiar with the concept of using code metrics as a code analysis tool and as a guideline to analyse the design of a system.
Usability Engineering	The content of this module eased my design of the frontend to my system and encouraged me to be more aware of many HCI concepts as relates to the user interface. It was also directly responsible for my choice of tool used to produce the requirements specification for Cevo Studios (POP - Prototype-On-Paper).
Networks and Operating Systems	This module gave me a background in a variety of technologies that came in useful on my placement: XML, Cryptography and to a lesser extent, networking protocol. XML was used as a transfer format between Winstock (in-house ERP) and external software, and as such knowledge of this was invaluable.
Algorithms	This module gave me a headstart and good background information when it came to creating a custom sorting algorithm for the eBay management system. Thanks to this module I was able to draw inspiration from classical sorting algorithms and customise them for my particular purposes.

Table 1. !st & 2nd Year courses and their relevance to this placement

5 Evaluation

Here follows an evaluation of the S.M.A.R.T. [2] learning objectives that I set myself (in agreement with my tutor, manager and the study guide [1]). Please refer to appendix 9.1 for the concrete definitions of these objectives.

5.1 Objective 1 - Design Patterns

I already had some experience using and implementing design patterns before the start of my placement; However I have never been responsible for a project of this size and scale before. This gave me the perfect chance to stretch my knowledge and experience of using design patterns. I learned a great deal whilst working on the TomoBay project, and feel that my skills are a good deal more advanced than they were when I started. The greenfield nature of the project as well as the fact that I had sole responsibility for the project gave me a lot of leeway when choosing the approach to take for any given problem, this is something that I probably would not have been afforded to the same extent working on an existing project with other developers.

One problem that I ran into constantly, and the one i learned the most from, was: many areas of related functionality Lent themselves to pattern or another however, were impeded in their implementations by the fact that the signatures or return types for these pieces of functionality varied. For instance: database queries, it was relatively simple to identify functionality common to all database queries and to generate an inheritance hierarchy for the different types of database query (Select, Insert, Update & Delete) and my first thought was to implement a command pattern and factory methods [4], to execute and hold (respectively) these queries as well as having the benefit of loose coupling between classes and a unified public API. However many of these queries require different types of information in order to be executed and many of them return very different types of information. So there was a trade off to be made, if I wanted to use the patterns mentioned I would have to make the method signatures and return types uniform. This type of problem cropped up repeatedly, the majority of the time I seem to have chosen the correct trade off as it made my life easier in the long run, but there were occasions where it is quite obvious that I have not made the right choice as it made life much harder, and the software less robust.

There is a particular combination of Command and Factory method [4] that i have discovered to be particularly useful and have used extensively throughout my code; the Command pattern is used to select and call a particular piece of functionality, on its own, this is a fairly standard usage for the pattern and useful in its own right. However in combination with a object that acts as factory, that is the user provides some kind of key and the factory creates and provides the user with the instantiated object they requested (usually using a Map implementation whose keys are defined by an Enum in order to clearly define and limit the possible input space, and whose values are factory objects). The combination of these two acts almost as a form of dependency injection container [3], which is rather useful for centralising access to related functionality whilst keeping each specific piece of functionality decoupled from the user, and the lazy instantiation of the end object (thanks to the Map providing factory objects) ensures that the end objects are not kept on the heap unless they are actually requested.

Another area that provided a great deal of benefit to the project was Enums and Enum type patterns such as the precursor to java 1.5's introduction of the enum keyword 'typesafe enum pattern'. In general the keyword version is preferred however in some cases the older pattern provides some flexibility that is lacking in the newer style enums(such as being able to form an inheritance hierarchy). However in both cases I found that the utility of these constructs was incomparable, as it allows for a way to clearly define: inputs to methods, object aliases, etc and to limit the possible inputs to objects an collections to what is defined within a given Enum.

Overall I have improved my knowledge of the use of design patterns a good deal although in hindsight the focus of my design pattern usage has focused mostly on fairly common patterns, going forward I should put more active effort into getting to grips with some of the less common patterns as well as those put forward by Martin Fowler.

5.2 Objective 2 - Concurrency

Going into this placement, my knowledge of concurrency was effectively limited to knowing the definition of threads, and that multi-threaded programming is almost universally considered tricky (based on anecdotal evidence). I had never actively created and used them in any project up to that point (with the exception of the main program thread..... obviously). As such I have gained, in my opinion, an intermediate level of knowledge regarding threads and concurrency.

Given that concurrent programming has a reputation for being somewhat thorny, my first port of call in the search for knowledge on the subject was to look for pitfalls and how to make code threadsafe; and it seems that there are two cases: the first is the relatively simple case where threads are, directly or indirectly, non interacting, that is to say that no information is transferred between threads and they remain effectively isolated from one another. The second case is where thread interactions can and do occur. These two cases can be treated almost completely separately from each other.

In the first case, non-interacting threads, it seems that the only real way to run into trouble is when threads use common, non final, static functionality, i.e. multiple threads use different instances of a class that contain static variables. This can cause trouble due to multiple threads attempting to act on the same variable with the possible outcome of causing unintended behaviour, or possibly deadlocks and race conditions. Effectively these previously non-interacting threads have become interacting. If no, non-final, static variables exist then no such issues can arise. In my placement I attempted to use only non-interacting threads as I came across no need for them to interact, and to the best of my knowledge I have followed the guidelines i have already explained, and as a result my code should be threadsafe for threads that do access shared object instances.

In the second case, it is necessary to use some form of locking, either in the form of synchronized blocks/methods/variables or through the implementation of mutexes. These operations are expensive though so it is preferable if at all possible to use only non-interacting threads.

Onto the business of actually creating threads, best practices [5](and others) as well as logic dictate that thread pools are the preferred way of managing thread lifecycles. This makes sense as creating threads is itself an expensive operation and should be limited to only what is needed. Java has a built in thread pool framework (ThreadPoolExecutor) which provides this functionality, and I used this to create two thread pools: the first was a scheduled thread pool, which I used for performing tasks that needed to run in the background at set intervals (such as grabbing data from ebay, or checking errors in data), the second was a standard thread pool for what i called 'triggered services', that is to say, functionality that needed to run in the background so as not to interrupt the program flow, but would be 'triggered' at a users request.

Something that I did not realise at the beginning of this project but which, in hindsight, I should have, was that: every request made of the web server embedded into the application (Jetty9) is handled by a separate thread. With this knowledge I could have reduced the workload on the thread pools resulting in a smaller number of total threads necessary to perform the same functionality. If I get time during the remainder of my placement I will attempt to remedy this.

In conclusion, I feel that I have been largely successful in this objective as I have learned a lot and become fairly comfortable implementing thread based division of labour as well as becoming mindful of the pitfalls inherent in multi threaded programming, It would have been nice to have had to use interacting threads and gain experience with them but as it was not really necessary it was probably for the best that i didn't. Although I do look forward to working with them in the future.

5.3 Objective 3 - MySQL

While I had been taught about the basics of Databases, in particular MySQL including how to write queries, the various normal forms, public & foreign keys etc. Transactions were not really covered in any detail during the course of my degree, either in modules focused on Databases, or programming modules.

The two requirements for this objective where that any database tables i produce be in third normal form, and that all database interactions be carried out through transactions. The first requirement is important in order to limit the amount of redundant information and to insure referential integrity of the tables. This makes sure that the database is more space efficient as well as making it harder for modification anomalies to occur, thus increasing the robustness of the database and as a result, the application that relies upon it.

The second requirement is equally important as the application that this database supports is a potentially very multi threaded piece of software, and as such many of the threads may be interacting with the database at any one point in time. Should the database fall over at any point then there is the potential for multiple queries to be mid execution when this happens and should any of these queries have dependencies on other queries then this could cause problems for referential integrity. Transactions allow the user of the database to define their own atomic operations and should something happen during the transaction (before it is committed) then none of the operations will go through, which is a much more desirable.

There were two issues I came across implementing the first of the requirements was in defining many to many relationships after the database was already in use. This caused issues as many of my foreign keys were VARCHAR based, if i had used INTEGER values the mapping table necessary for modelling a many to many relationship then the database could have been more space efficient.

The second area to cause me some mild vexation was that I could not make drastic changes in the tables already implemented, changing names and adding columns was not too much of an issue, but a complete restructure would not have been possible without causing unacceptable downtime. In hindsight however I could have cloned the database and restructured it without causing any downtime, and then after everything had been completed i could have made sure any data missing in the development database was added and swapped the two over again. This caused one of my database table to remain in second normal form rather than third, and as such I have not met this objective to the standard that I stated in appendix 9.1.

The second requirement for this objective, wrapping all queries in transactions, however; has been 100% met, and in fact required very little effort to implement. I chose the InnoDB engine to run the MySQL database, after which it was a matter of using the functionality built into JDBC in order to run all queries as transactions or part of transactions.

As an addendum to this objective I would like to mention that due to the same logic that caused me to implement thread pools [5] I also decided to implement connection pooling for the purposes of efficiency [7]. In order to achieve this I used a third party open source library called C3P0.

5.4 Objective 4 - Doxygen

One of the major headaches that seem to have plagued my colleagues, both on my degree course as well as at this placement and others is documentation; the need to write coherently commented code as well as easily accessible documentation at various levels of granularity to make a project accessible to others can sometimes feel like a lot of duplicate work. I have long been aware of the Doxygen tool and have used it in the past to provide class level documentation auto generated from class/method/field comments, in combination with the GraphViz dot tool this provides a minimum level of documentation required for a fellow developer to be able to, with *relative* ease jump in and become familiar with the code. However it still doesn't provide any big picture documentation.

One of the things that I learned about Doxygen whilst using it on this project is that external markdown files can be included in the build process to provide extra documentation for the areas that were lacking. I have used this feature to document the overall design of modules of the code and to show how particular classes within modules/packages fit together. It is my belief that this will cut down on the learning curve inherent in joining a project with an existing code base. As well as standard markdown these external files support most of the doxygen special annotations that allow the: formatting of particular blocks of text, hyperlinking between files/code and the building of a document hierarchy.

I have learned to use a variety of these special annotations in the code comments that I write. They have come in particularly useful when writing unit tests as the: @test, @pre, @post annotations allow me to respectively create a definition and description of the unit test in question, the pre-conditions necessary for the test to be successful and the post-conditions or guarantees that this test ensures. Another annotation that I have found rather useful is the @see, which creates a 'See also' block letting me create lists of references for certain materials, or to link to a related class.

Lastly I found out that by Doxygen will generate package level documentation if a package-info.java file is included within the package to be documented. While this is technically a java source file, its only purpose is to provide documentation for that package, this allows me to almost entirely separate the code documentation from userspace documentation; with package level docs covering the design and use of that particular package, and the external markdown files providing a top-down general view of the system as well as usage docs and tutorials.

There have been no real problems in using Doxygen and as far as I can see there are very few downsides to this rather wonderful system, the end result being; that for relatively little effort I have created extensive, beautiful and easy to implement documentation to go with my project.

5.5 Objective 5 - Git

Before starting my placement i received permission from Tomo Motor Parts Ltd that all code I created could be open-source, and in order to host it i chose to use the GitHub platform. This afforded me a place to publish my code (ensuring that all private information was not uploaded) as well as providing me with a version control system and some project management tools.

While I had used git in the past, I learned a lot from using it for such an extended project. One of the key things that became obvious trying to maintain a relatively large project was the amount of effort needed to keep the documentation branch and master in sync with each other. I ended up writing a script to build the project and documentation together and then commit the project source to the master branch and the documentation to the gh-pages branch.

The gh-pages branch is a branch that GitHub uses as the project website for that repository, many projects use it for that exact purpose, however I have discovered another use for it; publishing the documentation created using Doxygen.

Another feature of GitHub that is useful for documentation is the project wiki. This feature allows the user to use markdown to create a wiki. I started out with the intention of using the wiki to document design decisions and other areas not covered in the auto-generated doxygen documentation, however it proved to be much easier to add markdown documents to doxygen, and have all the docs in the same location, The wiki does point to the gh-pages documentation however.

The last feature that has provided a great deal of benefit is the GitHub issue system. This feature allows the developer to raise 'issues', which can be anything from bugs to feature requests or developers notes to themselves. Initially I used this system as an interactive ToDo list, but later on i discovered a browser plugin called Zenhub (https://www.zenhub.io) which integrates with Google Chrome and Firefox. This plugin transforms the GitHub issue system into a scrum/kanban style setup, adding boards that issues can be moved across as well as providing burndown charts for a Scrum style sprint. I discovered this about halfway through my placement and have been using it ever since to implement my one man scrum. Issues have become story cards in a sprint backlog, I then chose the storycards that need to be implemented next and move them to the sprint backlog, and as they move through various phases of development (Design to Testing to Implementation to Completion) I move them across the boards until all the functionality for that sprint has been implemented. The Burndown chart gave me a deadline to work to, which helped both motivate me, and improve my time estimation.

I have learned much from using Git and GitHub during this project and I very much hope to continue using it while progressing in my career.

5.6 Objective 6 - User Support

My final S.M.A.R.T. objective regards User Support; due to the small size of the IT team at Tomo Motor Parts Ltd, it was necessary to lend a hand occasionally with the user support, this included tasks such as:

- Helping users with email
- Setting up and diagnosing printer problems
- Installing and upgrading operating systems
- The installation, setup and configuration of an electronic cash register.
- Diagnosing and fixing problems with routers and the internet
- Migrating users from the existing email system to Office 365

Initially, when starting my placement I had a limited but working knowledge of the Microsoft Windows operating system. Having used Linux almost exclusively for the last 10 years it was relatively challenging and a little bit awkward having to perform administrative tasks in a relatively unfamiliar operating system.

Over the course of the placement I have become more familiar with Windows, and have learned how to perform a variety of tasks quite well, and feel comfortable with helping users with the majority of problems that they encounter (with the exception of some complex application specific problems).

One of the tasks that I am most proud of is installing, setting up and configuring an electronic cash register for the shop counter. This involved writing a batch file to interact with a command line application that communicates over the com port to open the till. I then also discovered a 'hack' to place an icon for it on the taskbar, making it easy for users to open the electronic cash register from the counter PC.

5.7 Stretch Objective - Warehouse Management System

This project was conceived during negotiations with PSA (Citroen/Peugeot) [8], to create a system that could handle the following tasks electronically:

- Picking
- Packing
- Deliveries (to Tomo)
- Stock checks
- Internal Stock audits
- Stock management
- Location management

This system would, ideally, be able to handle these common tasks performed in the warehouse using an Android handheld device with attached barcode scanner.

The first stage of this project involved sitting down with various stakeholders, primarily the Managing Director and Warehouse Manager to find out, in general terms, what they required from this system, how the system would be used and to what degree manual intervention was wanted/required.

I then sat down, taking what had been said before, and to the best of my ability translating this into a skeletal wireframe structure showing the process flow from one screen to the next with the aim of encapsulating all the requirements withing this.

The next stage was taking these preliminary wireframes back to the stakeholders; including, now, the users of the system. This was important as they were able to point out various improvements and flaws that the Management level stakeholders had missed. After a great deal of discussion and some modification to the preliminary wireframes, everyone was agreed upon what the system should look like, what information should be presented and the security measures that should protect the system from unauthorised and accidental use.

After the requirements where gathered and finalised (as much as they ever can be), I started the process of formalising these specifications so that they could be passed on to Cevo Studios to implement. Taking the advice of Wilson Fletcher [11] I decided to use a web based application called POP (Prototype On Paper) to create an interactive prototype to convey these specifications. This allowed me a compact prototypical representation of the specifications that was self documenting in so far as by using the interactive prototype the program flow is obvious. It also allowed me to add comments to the slides that make up the prototype, and so, to provide any clarification necessary (should there be behaviour that the prototype can not deal with using POP). The comments also allowed me to make (pseudo) user stories, that go along with the prototype.

Another useful feature of the POP web app is that it allows you to share the prototype with others by providing them with a link. This link allows the person that you have shared it with to view and use the prototype that you have shared, as well as (in some cases) write notes in reply. In our case i decided to have communication go through email rather than the POP comments (this could lead to a very cluttered specification).

This specification turned out to be succinct and effective as there was very little clarification needed by Cevo and they could get started almost immediately. From this point onwards I took on the role of first point of communication whenever Cevo needed clarification on a point about the functionality, in addition to this I provided support in the form of testing and feedback as well as providing advice/sounding board, for Cevo when they were having some issues with the barcode scanner functionality (and interfacing it with the android framework).

The specification was not provided to Cevo all at once, but was drawn up module by module, and then passed to Cevo as soon as a particular module was finalised. This allowed the development of the warehouse management system to start sooner and as a result be finished sooner. It also allowed us to go through a testing cycle on the module while the next one was being developed, passing feedback to Cevo regarding anything that needed changing/fixing. This methodology worked well and we quickly gained a very useful piece of software.

In hindsight, however, there are certain areas where having a couple more face to face meetings would have been useful. Particularly when integrating the parts of the warehouse management system that require data from Winstock (the in-house ERP). This is because it runs a custom protocol for communications over a socket, and while the specification for this protocol was sent to Cevo, it would have made for quicker development having them on premises while they were dealing with this; rather than an exchange of copious amounts of emails over a period of weeks.

5.8 Conclusion

In conclusion, with the exception of objective 3 (section 5.3) I believe I have completed all my objectives to a level that I am satisfied with, and in the process have developed my technical (objectives 1-5) and personal (objectives 6 and stretch) skills to a level greater than they were at the start of my placement. I feel comfortable with a number of situations both technical and personal that I was not comfortable with tackling before my placement. I would highly recommend doing a placement to anyone that asks.

6 Case Study

6.1 What motivated you towards applying to the company?

One of the big draws of Tomo Motor Parts Ltd, for me, was that it is a small company. There is a difference in the atmosphere in small companies vs large companies, it is more intimate, the chain of command is shorter and you get the feeling that what you are doing is affecting the company as a whole, in a real monetary way. I am not disparaging against large companies, they have their plus points too, but I felt that working for a small company my placement would be more focused on one project as opposed ti moving from department to department doing little bits here and there. I like a project that I can get my teeth into.

Along with this, the company had an IT team of 2, one of whom was a developer. Now this has both positive and negative points, but for me it indicated that this would be a challenge and a chance for me to stand on my own two feet in a commercial setting. As well as being personally responsible for all my achievements.

6.2 What did your role entail?

The role I took on essentially boiled down to full lifecycle Software Engineering, from gathering the requirements for software to be developed, then taking those requirements and designing software around it. Then Developing that software into a viable, working product; testing it and deploying it. Then making improvements, introducing new features, working out bugs (unwanted features) and integrating into the existing deployment. It really did cover pretty much every stage of the lifecycle of a commercial software project.

In addition to the responsibilities mentioned above, my role occasionally entailed liaising with developers from an external software house helping them with their testing, providing them with requirements specifications, and being a point of contact for them in case of issues.

6.3 Do you feel your work experience was valuable?

I feel my work experience was incredibly valuable, it allowed me to gain experience of a large scale software development project, engage in full lifecycle development. It also benefited me by allowing me to explore technologies and techniques that there are not covered at university such as: sockets, concurrency and dependency injection. The last area I feel my work experience has been valuable is in allowing me to regularly communicate with the stakeholders involved in the projects, forcing me to get used to explaining technical concepts and problems who, by their nature, are not technically minded.

6.4 How has your work benefited the company?

My work has benefited the company in a number of ways. The most obvious of these is the amount of man hours saved due to the introduction of my eBay management system. This is made obvious by the fact that when I joined there were two members of staff permanently employed and busy with dealing with eBay orders. Since then the amount of business going through the eBay channel has increased quite substantially with another two accounts being managed and yet the amount of staff has remained constant; that is to say, the number of staff dedicated to eBay went down to one, but in the last couple of months an apprentice has been hired in order to help with the picking/packing side of things. In terms of concrete savings to the company, the system saves them about 35%-40% in time, which at the time of writing equates to between 18.2 and 20.8 man weeks per year or a concrete saving (assuming minimum wage) of between £5792.15 and £6619.60.

Other concrete savings to the company were made by myself with technology related purchases, including: finding a good deal on android handheld units, as well as other hardware purchases. This lead to a total savings of around £5400 and lastly I also installed two high efficiency printers which are currently saving the company over £2000 per year.

Then there are the less tangible benefits: time saved helping external developers test code, writing formal requirements specifications and liaising with them with respect to prospective solutions, as well as general user support.

6.5 What did you learn that was not covered on the course?

Thanks to the open nature of the TomoBay project (eBay management system) I was able to learn about a good number of technologies and techniques not covered on the Brunel course, including:

- Concurrency and multi-threaded programming
- Communication using Sockets
- Database connection pooling
- Thread pooling
- Inversion of control & Dependency injection
- Java Generics
- Java JVM tweaking and optimisation

- Windows system administration

6.6 What general advice would you give to students who are yet to apply for a placement?

Applying for a placement (at least in my case) can be a tedious, long winded and often disheartening affair. Employers often don't reply for weeks, and when they do often its to say "were sorry to inform you .....", it is important to keep applying. Do a little bit every day, even if its just one application, over time you will get responses and some of them will be positive. Don't give up, you WILL find a placement.

Having an up to date CV is important, but I would say spinning your CV to the job/sector you are applying for is as important if not more so. I found the most useful way to do this was to tailor my CV to my ideal job, and my strengths. This has the benefit of getting favourable replies from companies that are more likely to be a good fit with what you want to do.

My last piece of advice is reply as soon as you get a response, the more communication you can coax out of an employer the less likely you are to stick in their minds and as a result; the more likely you are to progress further with your application.

7 Testimonials

Here follow some excerpts taken from letters of recommendation written by a number of stakeholders at Tomo Motor Parts Ltd(copies of the full letters are available upon request):

> "*Jan's work ethic has been exemplary and he has shown great flexibility in adapting his system to changes in business needs promptly and effectively. Jan is knowledgeable in his field and always willing to give support and advice when it is needed. He has been a dependable employee and an asset to the IT team.*"
>
> — Darren Perkins, Managing Director

> "*Jan works well in a team, as he has shown time and again when coordinating efforts with our external consultants; but is also more than capable of working unsupervised, and left to his own devices will push on with work that needs doing.*"
>
> —- Richard Ruth, IT Infrastructure Manager

> "*Jan is very approachable, friendly and very happy to help out with any IT issues I had, he would deal with them quickly with no fuss at all. If the system got disconnected from the servers and I couldn't invoice anything temporarily, he was quick to fix the problem and it was business as usual after that. His hard work and knowledge has made my job on Ebay so much quicker and easier.*"
>
> —- Steve Goreham, eCommerce Manager

For more professional information please feel free to check my LinkedIn Profile and GitHub pages:

https://uk.linkedin.com/in/jpchanson

https://github.com/jpchanson

8 Bibliography/References

[1] Coughlan J., (2015) CS2555 *Work Placement Study Guide* [ONLINE] Available from: *<WEB ADDRESS>*. [Accessed 20 August 2015]

[2] Doran, G. T., (1981). *"There's a S.M.A.R.T. way to write management's goals and objectives".* *Management Review* (AMA FORUM) **70** (11): 35–36.

[3] Fowler M., (2004). *Inversion of Control Containers and the Dependency Injection pattern.* Available: http://www.martinfowler.com/articles/injection.html. Last accessed 27th February 2016.

[4] Gamma E., Helm R., Johnson R., Vlissides J., (1994). *Design Patterns*. California: Addison-Wesley.

[5] Goetz B., (2002) *Java theory and practive: Thread pools and work queues* [ONLINE] Armonk, New York Available from http://www.ibm.com/developerworks/java/library/j-jtp0730/index.html [Last Accessed 23rd November 2015]

[6] Martin, Robert C., (2003). *Agile Software Development, Principles, Patterns and Practices.* Prentice Hall. p. 127-131

[7] MySQL AB, (2016) *MySQL Developer Guide Version 6.0 Chapter 8 Connection Pooling with Connector/J* [ONLINE] Uppsala, Sweden & Cupertino, California Available from: http://dev.mysql.com/doc/connector-j/6.0/en/connector-j-usagenotes-j2ee-concepts-connection-pooling.html [Last Accessed 15 April 2006]

[8] Perkins D., *personal communication regarding PSA hub deal*. 18th May 2016

[9] Spencer J., *personal communication regarding accounts*. 20th April 2016.

[10] Tomo Prestige Ltd, (2015) *Contract Of Employment*, available upon request, contact *jpchansondev@gmail.com*

[11] Wilson Fletcher, (2014), 'Prototyping - The Tools: What We Use and When To Use Them', CS2003: Usability Engineering , Available at: https://blackboard.brunel.ac.uk/webapps/portal/frameset.jsp?tab_tab_group_id=_2_1&url=%2Fwebapps%2Fblackboard%2Fexecute%2Flauncher%3Ftype%3DCourse%26id%3D_17219_1%26url%3D, [Accessed 16 March 15]

9 Appendices

9.1 S.M.A.R.T Objectives

Objective 1	**Skills**: *Design Patterns*
Description:	I aim to develop a good understanding of Design Patterns and utilise them extensively (where appropriate) throughout my code.
Barriers:	Due to the urgent need for a working system (so that Tomo does not lose custom or gain negative feedback on eBay), there is only a limited amount of time to weigh up the pro's and cons of various approaches. As a result, during the initial push towards a working core of code I will need to balance learning and implementing patterns I have previously not used against expediency.
Opportunities:	Due to the nature of the project, I have a blank slate to work with, which affords me the freedom to tackle the whole project from start to finish, as I think is best. I am not limited by having to conform to a prior code base.
Achievements:	• Many modules within my code conform to the inversion of control principal, primarily due to a map of factory method objects to enum constants, allowing objects for a particular module to be created and distributed from a central location. This central location can only accept values defined as constants of a particular Enum, avoiding the need for sanitising input to the class. Overall this approach leads to high cohesion within the module but loose coupling within the module w.r.t behaviour, as well as loose coupling between the module itself and users of it. • The creation of a general purpose SQL framework. • The creation of a framework to replace complex conditionals using double dispatch. • Implementation of an Email framework using Java mail and the builder pattern.

Table 2. Objective 1: Design Patterns

Objective 2	**Skills**: *Concurrency*
Description:	I aim to develop a working knowledge of Concurrency, its pitfalls and strategies for coping with them. I aim to produce typesafe code throughout any projects I work on.
Barriers:	My choice of frontend system (Jetty embedded web-server) means that I have to deal with the realities of concurrency from the get-go as opposed to building up to it. This is very much a sink or swim scenario.
Opportunities:	Jetty, Services
Achievements:	• No member variables in shared objects • No non-final static variables or objects within objects that could have more than one instance in existance at any instance during the program lifecycle. • Typesafe singletons. • Thread pooling • Use of Java's Executor system for the creation and management of threads

Table 3. Objective 2: Concurrency

Objective 3	**Skills**: *MySQL*
Description:	I aim to develop my knowledge of SQL and transactions, utilising transactions in all database queries and attempting to keep any databases I develop in third normal form.
Barriers:	I will be setting up the MySQL implementation of Windows Server 2012, an environment that I have previously only had limited experience with, In fact Microsoft Windows in general is something that I do not have much experience with; as for the last 10 years I have used Linux almost exclusively.
Opportunities:	This is a chance to familiarise myself with Windows Server from both an OS standpoint as well as a Server standpoint (maintainance, installation, security, etc)
Achievements:	• The design, creation and management of an Oracle MySQL database of live information taken from the eBay API. • All database tables in third normal form.

Table 4. Objective 3: MySQL

Objective 4	**Skills**: *Doxygen*
Description:	I aim to use the Doxygen tool(including some advanced features and annotations) to document ALL code that I produce during my placement, with AT LEAST documentation for all public methods and variables as well as class and package level documentation, for all classes and packages.
Barriers:	This is not a small project and the amount of documentation needed will be rather substantial, as a result, I will need to keep up with the documentation needs as I go; or else run the risk of having to play catch-up after a given period of time.
Opportunities:	I see this as an opportunity to get into/reinforce good habits when it comes to code documentation. I have always been quite rigorous with the documentation of code, but I have never had the chance to do this on a substantial commercial project.
Achievements:	• Built-in manual created from external Markdown documents. • Extensive package level documentation for code that is framework-like, including code snippets and usage examples. • In-Page cross-linking of classes to other relevant classes to convey code intent. • UML style inheritance and collaboration diagrams using dynamic SVG's as well as caller and call graphs.

Table 5. Objective 4: Doxygen

Objective 5	**Skills**: *GitHub*
Description:	My employers have given me permission to open-source the code that I write for them so I will use the Github platform throughout my Placement to record and document my code as well as for project management. As a result, at the end of my placement I will have a fully documented open-source project.
Barriers:	If i create a git repository in the development directory, by default it will upload everything to the repository, including possiblly sensitive information like passwords for accounts and such. For the sake of security it is necessary to make sure that all sensistive information is included in the relevant .gitignore file. Git also has quite a steep learning curve and many of the functions can take experience to use properly.
Opportunities:	This is a chance to launch a self contained open-source project from scratch that could concievably benefit a wider audience than just Tomo Motor Parts Ltd. It is also an opportunity to provide maintainance and design documentation in a central off-site place using GitHubs gh-pages functionality and wiki. This would allow anyone who succeeds me to find it easily and, hopefully, pick up where I leave off without too much hassle.
Achievements:	• Use of the GitHub gh-pages branch to post code documentation(generated by doxygen) online • Use of the GitHub issue tracker to keep track of bugs/new features/improvements/etc. • Regular commits.

Table 6. Objective 5: GIT

Objective 6	**Skills**: *User Support*
Description:	During the length of my placement I will provide training and support to the end users of my system as well as providing technical support to the existing IT infrastructure, I have agreed with the IT Manager that he will provide an informal written assessment of my performance in this regard at the end of my placement.
Barriers:	The infrastructure of the company is based around Microsoft products, and all the workstations are loaded with Windows 7. Being a Linux user, this is an OS that i am not so familiar with, and as such, in order to be helpful in a user support role I will need to learn how to administer the windows platform as well as how to deal with common windows problems.
Opportunities:	The positive side of using a system that I am unfamiliar with is that it provides me the opportunity to learn more about it, which can only be a good thing, and given that the majority of business users around the world use windows on their desktops, this is a skill that will stand me in good stead in the future.
Achievements:	• Script to remotely query users on a particular server and kick them off if necessary (Only a certain number of RDP session licences available, and would often get locked out of server if someone forgets to log out). • Installation of printers that come with somewhat temperamental drivers • Installation and setup of computerised cash register, as well as authoring a script that unlocks and opens the cash drawer. • Frequently changeing users internet gateway as one or the other goes down. Until I came up with the idea of using the RIP listener to automatically select the route to the internet that is most appropriate.

Table 7. Objective 6: User Support

Stretch Objective	
Description:	Tomo is employing an external software house to develop and build a Warehouse Management System for Android, Through discussion and meetings with the IT Manager, General Manager and Warehouse staff I will be responsible for creating the specification for this system. This specification will take the form of a documented interactive prototype. I will also be responsible for communicating this specification to the software house responsible for building it as well as clarifying any questions they have regarding the specification.
Barriers:	Cevo Studios (The external software house) are not familiar with the auto parts trade and as a result I need to make sure that certain jargon terms that are par-for-the-course at Tomo are accurately and correctly conveyed to Cevo so as to minimise (hopefully eliminate) confusion.
Opportunities:	This is a chance to improve my communication accross different groups of stakeholders. It is also an opportunity to practice requirements gathering and formalisation processes.
Achievements:	• Creation of an expansive interactive prototype, indicating the application flow and intent. • Succesful communication of the requirements to Cevo Studios.

Table 8. Objective 6: User Support

9.2 Skills analysis

Skills Analysis

Student Name:	Jan Hanson	**Date last updated:**	18/03/2016
Manager Name:	Richard Ruth	**Date approved by manager:**	18/03/2016
Tutor Name:	Tracy Hall	**Date approved by tutor:**	21/03/2016

Please rate current level of skills currently agreed as relevant to the job using the scale below:

1. Unacceptable.
2. Performance not fully up to requirements. Some improvement necessary.
3. Performance fully meets normal requirements of a placement student.
4. Performance significantly above requirements.
5. Outstanding.

Skill	Rating (1-5)	Action for improvement
Ability to take initiative	4	If a problem is spotted talk to the appropriate member of staff, even if they may not be initially receptive to the idea.
Time Management	3	Work on improving the self assessment of deadline timescales
Prioritising work	4	Take time occaisionally to assess whether the work currently being done is providing the most benefit at this time
Teamwork	3	Work on closer communication with team members
Interviewing technique	4	Interviewing stance could be a little on the passive side, take a slightly more aggressive/assertive role in the interview
Negotiation skills	4	Work on not backing down so easily
General communication skills	4	Learn to communicate technical concepts to non technical people.
Ability to benefit from constructive criticism	3	Act more decisively regarding the constructive criticism offered to rectify the problem at hand.
Ability to understand instructions	5	No action needs to be taken
Research skills	5	No action needs to be taken
Computer literacy	5	No action needs to be taken
Ability to extract core problem	5	No action needs to be taken
Problem solving skills	5	No action needs to be taken
Ability to deal with changing requirements	4	It could be beneficial to treat changes in requirements as a new challenge as opposed to an inconvenience that needs to be dealt with.

Figure 4. Skills Analysis

9.3 Log Book

Activities	Outcomes
Introduction to the company	The first week of my placement was spent familiarising myself with the company and the way that it works, as well as setting up my environment. During this time I introduced myself to my colleagues and became aquainted with the IT setup at Tomo Motor Parts Ltd.
Building knowledge of ebay workflow	During the first couple of weeks, I started to identify how the eBay workflow progressed from an order coming in, to it being picked packed and shipped. This was vital for me to build up some kind of specification for the system that I am required to build.
Familiarising myself with eBay API	The next stage in the process of designing the TomoBay eBay application wsa to sign up for and familiarise myself with the eBay API. This started off well but the more i delve into the API the more sparse the documentation became.
Familiarising myself with connection pooling	I read around the topic of connection pooling using a variety of online resources and found a third party library that was small efficient and did exactly what I needed, C3P0.
Creating a code infrastructure	I constructed a basic infrastructure to work from. This was based around an MVP (model-view-presenter) type architecture.
Refactoring Views	After writing some rather elegant code for the interaction between the presenters and the views, i realised that it was not the most sensible way to implement the views and was forced to restructure my code so that it was more consistant with the way the application was going to work in the long run.
Creating Database Transaction mechanism	I created a framework for the creation and execution of database queries using JDBC and its built in transactional funcionality. I also built in the connection pooling library mentioned earlier.
First look at concurrency	I read around the topic of concurrency looking for common mistakes and pitfalls to avoid, as well as looking into thread pooling and how to make code thread-safe.

Table 9. September 2015 Log

Activities	Outcomes
Database design	I created an initial database design, taking into account the data that would be pulled from eBay and making sure that all tables were in third normal form.
Regular Expressions	I familiarised myself with regular expressions in order to solve some validation problems, testing myself using an online regular expression parser.
Photography	I implemented a new product photography setup for Tomo, buying a new camera as well as a lens more suited to close up shots, and some soft boxes to provide good flexible lighting.
Start of Warehouse management system specifications	I sat down with the Managing Director and warehouse staff to start building a specification for the Warehouse management system that was being outsourced to Cevo Studios. I also started drawing up specifications using the POP application.
Sorting algorithms	I implemented a catagory sorting algorithm to sort the incoming orders according to a priority determined by talks with the eBay department and managing Director
Integration with existing systems	I integrated my system with the existing DMS using a RESTful API provided by the programmer responsible for the development of the DMS.

Table 10. October 2015 Log

Activities	Outcomes
Migrating to Office 365	I took the first steps in the migration from the existing hosted exchange server to Office 365. I also produced a list of users who will need accounts when the migration is complete.
Creating a CSV export system	I implemented a system to generate CSV files that can be imported into the Royal Mail dispatch manager. Thus allowing for the semi automatic generation of packing labels saving the eBay department from doing this task manually. This is a stopgap solution while I work on integrating the Royal Mail API into my system.
Familiarising myself with Royal Mail API	I started familiarising myself with the Royal Mail API using documentation found on the Royal Mail developer portal. The API is based on the SOAP protocol and has no language specific wrappers, meaning that it will need to be implemented from scratch.
Writing Placement report	I wrote my placement report, which is a requirement for passing the placement module as the thick sandwhich part of my degree course.
Getting to grips with SOAP and SAAJ	I familiarised myself with the SOAP protocol and started to write some tests using the SAAJ (SOAP with Attachements API for Java).
Implementing Royal Mail APi	I started work on implementing the Royal Mail API into my system for the automatic generation of shipping labels and manifests.

Table 11. May 2016 Log

9.4 Placement Visit Reports

Placement Visit 1 Report

Placement Tutor: Tracy Hall

Date of Visit : 13/11/2015

Student Interview

Student Workplace Experience

- [x] The Student feels that they fit in well in the work environment and has experienced no significant difficulties.
- [ ] Despite initial difficulties, the student feels they have adapted well to the work environment.
- [ ] The student feels that there are unresolved issues in their work environment.

Work Undertaken

- [x] The student feels that the work is entirely appropriate for the requirements of the work placement scheme.
- [ ] Some issues have been identifiedby the student, but have been or can be resolved.
- [ ] The student feels that the work is not appropriate for the requirements of the work placement scheme.

Objectives Agreed with the manager

- [x] The student believes that they are making good progress against the objectives.
- [ ] The student feels they have made reasonable progress but has encountered some problems.
- [ ] The student feels that they are making little progress against the objectives.

Learning Outcomes

- [x] The student understands the learning outcomes and is confident of achieving them.
- [ ] The student understands the learning outcomes and has identified problems/potential problems in achieving them.
- [ ] The student appears not to understand the learning outcomes.

Support for the student

Yes	No	
[x]	[ ]	The student feels that they have good support from the employer
[x]	[ ]	The student feels that they have good support from the university.

Outlook

- [x] The student is satisfied with the plans for the remainder of the placement.
- [ ] The student has some reservations about the plan for the placement, but through discussion and objectiv setting, the student feels that these can and will be resolved.
- [ ] The student has some concerns or uncertainties around the plan for the rest of the placement.

Manager Interview

Name of Manager Interviewed: Darren Perkins

Workload

- [x] The manager feels that a varied and stimulating workload is provided.
- [ ] The manager feels that the workload could be more varied and stimulating.

Objectives and Learning outcomes

- [x] The manager has discussed and agreed the objectives and learning outcomes with the student.
- [ ] The manager has not discussed and agreed the objectives and learning outcomes with the student.

Progress

- [x] The manager feels that the student has made good progress against the agreed objectives and learning objectives.
- [ ] The manager feels that the student has made reasonable progress against the agreed objectibes and learning outcomes.
- [ ] The manager feeks that the student has made little progress against the agreed objectives and learning objectives.

Figure 5. Placement Visit 1: Report(page 1)

Support from Brunel University

☒ The manager is aware of the support provided by the University.
☐ The manager is not aware of the support provided by the university.

Collaboration

☒ Collaborative opportunities were not discussed
☐ No interest was shown.
☐ Some interest was shown, but this needs to be followed up at a subsequent visit.
☐ Interest was shown and is being followed up regarding:

Tutors Comments

"*The student is performing some very challenging development work which he seems to be able to cope with. I did discuss the level of challenge with the manager with a view to monitoring that the student continues to cope with this*."

Tutors assessment of the visit outcome

☐ There were some minor issues, but these have been or are in the process of being resolved.
☒ The placement is fit for purpose. No further action necessary.
☐ There are serious issues which will be brought to the attention of the placement and careers centre and/or the department/division.

Students statement of Acknowledgement

"*I agree with what has been reported*"

Figure 6. Placement Visit 1: Report (page 2)

Placement Visit 2 Report

Placement Tutor: Tracy Hall

Date of Visit : 20/04/2016

Student Interview

Student Workplace Experience

- [x] The Student feels that they fit in well in the work environment and has experienced no significant difficulties.
- [ ] Despite initial difficulties, the student feels they have adapted well to the work environment.
- [ ] The student feels that there are unresolved issues in their work environment.

Work Undertaken

- [x] The student feels that the work is entirely appropriate for the requirements of the work placement scheme.
- [ ] Some issues have been identifiedby the student, but have been or can be resolved.
- [ ] The student feels that the work is not appropriate for the requirements of the work placement scheme.

Objectives Agreed with the manager

- [x] The student believes that they are making good progress against the objectives.
- [ ] The student feels they have made reasonable progress but has encountered some problems.
- [ ] The student feels that they are making little progress against the objectives.

Learning Outcomes

- [x] The student understands the learning outcomes and is confident of achieving them.
- [ ] The student understands the learning outcomes and has identified problems/potential problems in achieving them.
- [ ] The student appears not to understand the learning outcomes.

Support for the student

Yes	No	
[x]	[ ]	The student feels that they have good support from the employer
[x]	[ ]	The student feels that they have good support from the university.

Outlook

- [x] The student is satisfied with the plans for the remainder of the placement.
- [ ] The student has some reservations about the plan for the placement, but through discussion and objectiv setting, the student feels that these can and will be resolved.
- [ ] The student has some concerns or uncertainties around the plan for the rest of the placement.

Manager Interview

Name of Manager Interviewed: Darren Perkins

Workload

- [x] The manager feels that a varied and stimulating workload is provided.
- [ ] The manager feels that the workload could be more varied and stimulating.

Objectives and Learning outcomes

- [x] The manager has discussed and agreed the objectives and learning outcomes with the student.
- [ ] The manager has not discussed and agreed the objectives and learning outcomes with the student.

Progress

- [x] The manager feels that the student has made good progress against the agreed objectives and learning objectives.
- [ ] The manager feels that the student has made reasonable progress against the agreed objectibes and learning outcomes.
- [ ] The manager feeks that the student has made little progress against the agreed objectives and learning objectives.

Figure 7. Placement Visit 2: Report (page 1)

Support from Brunel University

- [x] The manager is aware of the support provided by the University.
- [ ] The manager is not aware of the support provided by the university.

Collaboration

- [x] Collaborative opportunities were not discussed
- [ ] No interest was shown.
- [ ] Some interest was shown, but this needs to be followed up at a subsequent visit.
- [ ] Interest was shown and is being followed up regarding:

Tutors Comments

"*The student seems to have done an excellent job in a very challenging environment*"

Tutors assessment of the visit outcome

- [ ] There were some minor issues, but these have been or are in the process of being resolved.
- [x] The placement is fit for purpose. No further action necessary.
- [ ] There are serious issues which will be brought to the attention of the placement and careers centre and/or the department/division.

Students statement of Acknowledgement

"*I agree with the content of this report*"

Figure 8. Placement Visit 2: Report (Page 2)