An Apparel PLM System Based on a Service-oriented Architecture

Paul Malone
BSc Computing (with Industry)
Session 2011/2012

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student) ________________________________
Summary

This project is a self-proposed project working with an external company called ‘PDP Limited’. The idea for the project was developed while on my placement year at PDP Limited whose business is based around consulting on existing PLM solutions in the fashion and apparel industry.

This report investigates the challenges for a PLM vendor in using Web services to develop a PLM system that is based on a SOA. Current solutions and practices were researched in order to gain an understanding of the choices that had to be made when developing Web services and implementing a SOA. Requirements analysis was carried out to obtain requirements for the Service-based PLM system, the requirements were then transformed into a design document where upon completion; the System was implemented. After the solution had been implemented, it was evaluated and further enhancements to the solution were discussed.
Acknowledgments

I would like to thank my supervisor; Roy Ruddle, for challenging me early on in the project to ensure that the project was feasible and for his invaluable support and guidance throughout the project. I would also like to thank my supervisor; Lydia Lau, for her very good feedback in the mid-project report and progress meeting.

Finally, I would like thank my family for their financial and emotional support throughout my life as a student.
# Contents

1  Introduction.................................................................................................................. 1

1.1  Project Aim .................................................................................................................. 1

1.2  Objectives ..................................................................................................................... 1

1.3  Minimum Requirements ............................................................................................... 1

1.4  Possible Extensions ...................................................................................................... 2

1.5  Relevance to Degree ..................................................................................................... 2

2  Background Research...................................................................................................... 3

2.1  Introduction ................................................................................................................... 3

2.2  Product Lifecycle Management ..................................................................................... 3

2.2.1  Summary ................................................................................................................... 6

2.3  Service-oriented Architectures ...................................................................................... 6

2.3.1  Summary .................................................................................................................. 8

2.4  Web Services ................................................................................................................ 8

2.4.1  Simple Object Access Protocol ................................................................................. 8

2.4.2  Representational State Transfer ................................................................................. 9

2.4.3  SOAP and REST ...................................................................................................... 10

2.4.4  Summary ................................................................................................................ 11

2.5  Tools ............................................................................................................................ 11

2.6  Methodology ................................................................................................................ 13

2.7  Evaluation ..................................................................................................................... 14

2.8  Conclusion .................................................................................................................... 14

3  Requirements Analysis .................................................................................................... 15

3.1  Introduction .................................................................................................................. 15

3.2  Business Case ............................................................................................................... 15

3.2.1  Problem Statement .................................................................................................. 15

3.2.2  Project Description ................................................................................................ 17

3.2.3  Solution Description ............................................................................................... 18

3.2.4  Assumptions and Risks .......................................................................................... 19
5.6 Revised Client Application ........................................................................42

6 Evaluation .................................................................................................45

6.1 Introduction ............................................................................................45

6.2 Steering Development of an Apparel PLM System towards a Service-oriented Architecture ........................................................................45

6.3 System Requirements Evaluation ............................................................47

6.3.1 Functional Requirements ...................................................................47

6.3.2 Non-functional Requirements ..............................................................49

6.3.3 Summary ............................................................................................49

6.4 Comparison with Traditional PLM Systems ..........................................50

6.4.1 Introduction .........................................................................................50

6.4.2 Method ...............................................................................................50

6.5 Project Evaluation ....................................................................................53

6.6 Further Work ..........................................................................................54

7 Bibliography ..............................................................................................55

Appendix A: Personal Reflection ....................................................................57

Appendix B: Materials Used In the Solution ................................................59

Appendix C: How Ethical Issues Were Addressed .......................................60

Appendix D: Summary of Functionality .......................................................61

Appendix E: Gantt Charts ..............................................................................63

Appendix F: Participant Instructions .............................................................65
Figures

Figure 1: Results of the question - "Was your installation on your own servers or via a SaaS model?" ........................................................................................................................................... 4
Figure 2: Results of the question - "What prompted you not to move forward with the other solutions?" ......................................................................................................................................... 5
Figure 3: PLM Process Flowchart ................................................................................................................................................................................................. 21
Figure 4: High Level Architecture of an Apparel PLM System .......................................................................................................................... 26
Figure 5: High Level Architecture of a Service-oriented Architecture Apparel PLM System. 27
Figure 6: PLM Web service design ...................................................................................................................................................................... 28
Figure 7: Example Data .......................................................................................................................................................................................... 29
Figure 8: Database Table .......................................................................................................................................................................................... 29
Figure 9: Add Style Use Case ........................................................................................................................................................................ 29
Figure 10: Search Use Case .................................................................................................................................................................................. 30
Figure 11: Count Styles Requiring Approval Use Case .......................................................................................................................... 31
Figure 12: View All Styles Use Case ............................................................................................................................................................ 31
Figure 13: Remove Style Use Case ............................................................................................................................................................... 32
Figure 14: Browser Layout ............................................................................................................................................................................. 33
Figure 15: Revised Menu Design ................................................................................................................................................................... 34
Figure 16: Hello World Output ..................................................................................................................................................................... 36
Figure 17: HTML Post Form ............................................................................................................................................................................. 39
Figure 18: Insert Style Input .......................................................................................................................................................................... 43
Figure 19: Results Window ............................................................................................................................................................................. 43
1 Introduction

1.1 Project Aim

The aim of the project is:

To investigate the challenges for a Product Lifecycle Management (PLM) vendor in using Web services to develop a proof of concept PLM system based on a Service-oriented Architecture.

1.2 Objectives

The objectives of the project are to:

- Understand the key functionality required for an Apparel PLM System.
- Gain a good understanding of the key principles of Service-oriented Architecture’s.
- Research Web service design models.
- Learn how to implement and consume Web services.
- Design and implement a database to store product data.
- Design and implement a Web service to build towards an Apparel PLM System that is based on a Service-oriented architecture.
- Design and implement a Client application to consume the Web service.
- Evaluate the use of Web services in a Service-oriented apparel PLM System.

1.3 Minimum Requirements

The minimum requirements are:

- A business case justifying a SOA PLM including an analysis of a traditional client/server PLM systems and identification of problem areas.
• A Design of the Web service using the SOA design principles.
• A basic PLM system that uses the PLM Web service to perform common tasks such as creating, searching and editing product data.
• An evaluation on the use of Web services to steer the development of a PLM system towards a Service-orientated architecture.

1.4 Possible Extensions

The possible extensions are:

• The ability for different user types to access the system with certain permissions.
• The ability to use JSON for data exchange.
• The ability to use different data formats in the system, e.g. JPEG’s for thumbnails.
• The ability to use Connection Pooling methods for database connections.

1.5 Relevance to Degree

This project focuses on Web services and Service-oriented architectures; these subjects have strong ties to the Distributed Systems module as technologies and methods related to these domains were studied. The Software Systems Engineering module will provide an understanding of how to plan and execute a project including key processes such as planning and requirements analysis. As my project involves working with the external company with whom I worked for on my placement year; the project can be seen as a natural extension of the previous year of study. It is true to say that the project will also provide an opportunity to demonstrate the skills and knowledge that have been obtained throughout the degree.
2 Background Research

2.1 Introduction

The purpose of undertaking background research is to discover literature that is relevant to the problem and gain a good understanding of all subjects related to the project. Another reason for comprehensive background research is to find information that will assist in any decisions that have to be made throughout the project. Literature has been discovered using several journal databases that were available at the time of writing and by exploring referenced resources, in any papers or books that were relevant to the key issues of this project.

2.2 Product Lifecycle Management

Product Lifecycle Management (PLM) is a multi-industry business methodology for managing the lifecycle of a product or service that a given organisation provides. Stark [1] discusses how the methodology facilitates the management of a product from the initial idea all the way the lifecycle until it is eventually retired; from cradle to grave. A PLM system is an application that an organisation would implement so that the PLM business methodology can be put into practice throughout the organisation. As the methodology is concerned with the full life of a product or service, it is clear that any system developed in this project must endeavour to maintain focus on the product at all times.

A PLM system is vital for implementing the PLM methodology and without such a system; it can be said that the task of managing a product would be close to impossible. For this reason; PLM systems are widely adopted in industries such as aerospace, automotive, electronics and utilities. This project will focus on PLM systems in the apparel industry only as they represent a growing challenge when compared to other industries. Using the design and development of an aeroplane as a comparison with the apparel industry; an aeroplane is made up of parts that do not vary in large numbers. However, in the apparel industry thousands of variations can exist for one product with organisations often producing
hundreds of products per season. This means every year a sizeable apparel organisation needs to manage a vast amount of product variations across the supply chain, which is usually spread over multiple continents. Stark [1] explains that in addition to managing product data; service data and information regarding any processes that need to undertaken at stages throughout the lifecycle needs to be managed by a PLM system. Therefore, any system produced in this project must provide the key features of a PLM system that allow an organisation to manage the people, processes and data that are required for each stage of the lifecycle.

Currently PLM systems for apparel are a new industry with few current adopters. The industry magazine ‘WhichPLM’ [2] identify that the term ‘PLM’ has only recently started to be used in the fashion industry. As PLM systems are not yet an established component of typical apparel organisations, it is clear that functionality provided by any system produced in this project needs to enhance the business process of organisations in order to demonstrate that the adoption of a PLM system is beneficial to them.

Some examples of major PLM vendors are; ‘Lectra’, ‘Dassault Systèmes’ and ‘Infor’. The solutions they all provide are based on a traditional client-server model and while some features of their systems may be termed as being ‘cloud-based’; the underlying architecture of apparel PLM systems is usually client-server. Figure 1 shows the results obtained from a question taken from the WhichPLM survey [1].

![Figure 1: Results of the question - "Was your installation on your own servers or via a SaaS model?"
](image)

A number of organisations were asked if their PLM system was installed on their own servers or whether it was provided via a ‘Software-as-a-Service’ (SaaS) model. The results show that an overwhelming majority of apparel PLM systems fit the traditional client-server model despite the claims by many PLM vendors that their solutions are available via a SaaS
delivery model. Based on this research it is clear that any solution in this project that is intended to be based on a service model should be a true SaaS implementation and should not implement SaaS or ‘cloud’ functionality as an afterthought.

As PLM systems are relatively new to the apparel industry and require large implementation procedures; problems exist with the current traditional solution. Some of these are typical of other types of systems such as; the system is too complicated to use, the system is slow and the support sold with the system is limited. Figure 2 shows the results of another question taken from the WhichPLM survey [2].

![Figure 2: Results of the question - "What prompted you not to move forward with the other solutions?"](image)

The results show that the most popular replies were that the solution was too expensive and that there was not enough ‘out-of-the-box’ (OOTB) functionality. From this information; it is clear that any solution produced in this project should endeavour to be cheaper for organisations to use than existing solutions and should not require large amounts of customisation before functionality can be used.
2.2.1 Summary

Based on information obtained from this research the decision has been made to produce an apparel PLM system that has the ability to be delivered via a SaaS model. A PLM system based on services will cost less to the organisation or user than existing systems and will require less in-house customisation which in itself incurs more cost. The system must provide functionality to manage the people, processes and data that are required throughout the lifecycle of a product. To enable the PLM system to provide its functions through services a Service-oriented Architecture (SOA) will be designed and implemented.

2.3 Service-oriented Architectures

Silcher [3] and Coulouris [4] agree that a system based on a SOA must adhere to a set of design principles which state the services in the system must be loosely coupled, platform independent and make use of existing services. As the PLM system will be based on a SOA it is clear that any services developed should not be tied to one application. Also, processing in any applications that are developed should be carried out by the services and the client application should make use of these services. It also clear that research must now be carried out to investigate the key principles of a SOA.

One of these key principles is the principle of loose coupling. Erl [5] describes coupling as a measure of how much one service depends on another and how much of a relationship the two services have. Therefore, any services implemented in the apparel PLM system should be independent of one another; a given service in the PLM 'A' and a given service 'B' should not need to obtain any information from each other to operate and they should both be able to function without the other.

Although the services should be loosely coupled, the consumer of the service may require details such as the parameters that need to be sent. The principle here described by Erl [5] is to find the correct balance between the content and quantity of information that is made available to the consumer with a view to prevent the user accessing details about the service that is not relevant or useful to them. This principle makes it clear that any services in the PLM system should provide only the required information to the consumer of the service and that the system should hide irrelevant information on how the PLM service works.
In addition to exposing a suitable quantity of information to the consumer, the system should also expose services that have a coarse granularity. Huhns and Singh [6] discuss the need to capture high level functionality in the services provided as opposed to providing services that offer low level functionality. This principle enforces the previously mentioned requirement that appropriate functionality should be provided by the PLM system to demonstrate the benefits of adopting a service-based PLM system.

Although the services need to provide high level functionality; they must do so while remaining stateless. He [7] explains how consumers of the services in the system are responsible for providing all the information required for the processing that is performed within a service. This principle describes how each service in PLM system should be able to accept all the parameters that are necessary for the operation it is providing and that any client applications should be responsible for sending this information to the service. As only authorised users within a PLM system are entitled to perform some operations; state management is often required by the system to check if a user is logged in. Erl [5] describes how this can still be achieved by deferring the management of state information by passing it to another component in the architecture. Therefore, if state data is required by the apparel PLM system; a method to manage this data that does not use the PLM service should be developed.

In addition to considering where state management should be differed, places of overlap in the system should be considered. If possible services should have the ability to be used by more than consumer in the architecture. Erl [5] explains how services in a system can be treated as a resource in the organisation if they placed in a position where they can be used by multiple consumers. This principle can be applied to the Service-based PLM system as there will be operations that are performed by individuals in several phase of the lifecycle that are the same and can therefore be implemented as a reusable service.

Finally, any services that are used in a SOA must be discoverable by a consumer. Erl [5] describes how Meta data can be used to enable services to be discovered. This principle demonstrates how the services in a SOA are designed to be used by consumers who may not be aware of how they technically work. Therefore any services designed for this project should take into consideration all the potential users of the service and not be specific to one client application. Although the services in this project may only be used by a client application implemented in this project, they should be made discoverable once the full system is built.
2.3.1 Summary

Based on this research into SOA’s the decision has been made to go ahead with implementing the PLM system based on a SOA. The architecture should be made up of services that operate independently of one another and operate without considering application state. Irrelevant information about the operational details of the service should be hidden from the consumer and the functionality provided should be at a coarse granularity. To implement the services in the SOA based PLM; Web services will be used. However research is first required to learn about the different implementation choices of Web services.

2.4 Web Services

A Web service is a process or piece of business logic that is exposed by an organisation as a service on the internet. Curbera [8] describes how they use Extensible Mark-up Language (XML) for data transport over existing Web protocols to enable communication between applications. As Web services will be used to develop a SOA PLM system it is clear that any services must use XML for data transport.

2.4.1 Simple Object Access Protocol

The communication protocol; ‘Simple Object Access Protocol’ (SOAP), which also uses ‘Hypertext Transfer Protocol’ (HTTP) and XML enables Web services to transfer messages. Curbera [8] explains how SOAP can also be used via other transport protocols such as ‘Simple Mail Transport Protocol’ (SMTP). This is one method that could be adopted to implement Web services in the PLM system; however more research is required before making a decision.

A distinctive feature of SOAP is that the XML message to be sent is wrapped in a SOAP ‘body’ and placed within a SOAP ‘envelope’. A description of the service which specifies information such as the location of the service and a description of how to interface with the service is usually defined using the Web Services Description Language (WSDL). Coulouris [4] explains that the WSDL document represents the types and bindings of a service in an
XML schema. If a SOAP implementation is chosen; the WSDL document can be used by third-party developers to build clients that can consume the PLM Web services.

Once a WSDL document has been created, a registry called Universal Description, Discovery and Integration (UDDI) can be used to find it. Curbera [8] discusses how the centralised registry that is provided by UDDI that enables users to find service providers. If a SOAP implementation of Web services is used in this project, the PLM services could be found by consumers by querying the registry.

2.4.2 Representational State Transfer

An alternative method for the implementation of the PLM Web service is Representational State Transfer (REST). Pautasso [9] and Vinoski [10] agree that it is an architectural style that promotes the transfer of resource state representations in large-scale systems between clients and resources. This is another implementation style that could be adopted to implement Web services in the PLM system; however more research is required before a decision can be made. As REST is an architectural style and not a protocol; key principles exist that the service must adhere to for it to be considered a ‘RESTful’ Web service.

Pautasso [9] explains how one principle of REST is that each resource must be identified by a ‘Uniform Resource Identifier’ (URI). If the REST style was used for the implementation of the PLM Web services then each service must have one of these unique identifiers.

A RESTful Web service must adhere to the Uniform Interface principle. Pautasso [9] describes how the only operations that can be used to manipulate resources are the four Create, Read, Update and Delete (CRUD) HTTP verbs. If the Web services in the PLM system are RESTful then the only methods a client application will be able to perform on a PLM resource are; ‘GET’, ‘DELETE’, ‘PUT’ and ‘POST’.

In agreement with the SOA principle of stateless services that was discovered previously, a RESTful Web service must be self-descriptive. Vinoski [11] believes that this principle means the formats of representations must be specified in any messages sent between clients and resources. If this style is adopted for implementing the PLM services then the ‘content-type’ Meta data header must be used to specify which data format is returned by a resource. Pautasso [9] also explains that the Meta data can also be used for detecting errors and authenticating requests. It is clear that if data types other than XML are required by consumers of the PLM Web services then this can be requested using Meta tags.
Although the resources are stateless; Pautasso [9] and Vinoski [11] agree that a stateful application can be implemented by allowing applications to navigate URI's returned from resources and using them to access related resources. This is called the hypermedia driving application state principle. A disagreement with the stateless principle of a SOA occurs here. Although it was earlier discussed that stateful SOA applications can be achieved by passing the management of states to another component in the architecture, the hypermedia principle implies that temporary URI's can be returned from resources which are only available to an application in a certain state. This emphasises the point that REST is an architectural style and some principles can therefore be interpreted differently. In this project, if the REST style is used and a stateful application is required then the decision has been made to ensure each resource is stateless and any information required for the Web service to operate will be provided by the consumer.

2.4.3 SOAP and REST

SOAP and REST are the two different methods available to implement Web services. A decision must be made to determine which method will be used to implement the service-based PLM system. Before that decision is taken, a comparison of the two methods needs to be undertaken. Based on the comparison method used by Pautasso [9] the two approaches will be compared in three areas; Principles, concepts and technologies.

SOAP and REST share numerous principles; both approaches use existing Web standards but both differ in how they use them to implement services. REST uses URI’s to address resources; Pautasso [9] explains how this results in RESTful applications becoming part of the Web in the way that typical websites are. SOAP on the other hand uses existing Web standards specifically for transporting messages between clients and Web services. Pautasso [9] describes how SOAP applications use the Web only for interaction and are not part of the Web in the way that REST applications are. Both SOAP and REST adhere to the ‘loose coupling’ principle of a SOA. However there does exist a difference; SOAP messages can be queued if the service provider is unavailable at any time whereas a REST service cannot due to its synchronous nature. Coulouris [4] explains how SOAP messages have a setting that can be set to specify if the message should be sent several times if the eservice is currently unavailable. A RESTful Web service can be said to have a uniform interface whereas SOAP does not. A REST interface only permits ‘CRUD’ operations to be performed on the resource whereas the interface for a SOAP Web service is specified in development and can be changed over time.
When the two implementation choices are compared it can be said they are very different in terms of their underlying concepts. Pautasso [9] confirms that the two methods have a different style of integration with clients and other resources. It is fair to say that REST is simpler to implement as the interface for a RESTful resources is already defined in existing Web standards.

SOAP and REST can also be compared on their respective underlying technologies. The use of the HTTP protocol in a RESTful implementation is compulsory whereas in a SOAP implementation messages can be exchanged using other transport protocols such as TCP and SMTP. SOAP services must make use of SOAP envelopes for message passing whereas this is certainly not the case for a RESTful resource; Vinoski [11] identifies that resource responses are possible in many formats including XML which will be used in the service-based PLM system produced in this project. Another technological difference between the two implementations is how they address resources; REST uses the URI method which can also be used by SOAP but SOAP implementations generally use ‘WS-Addressing’ which Coulouris [4] describes as an endpoint reference containing location information. Finally, SOAP Web services will often use ‘WSDL’ as a method for describing their services, as of yet there is no accepted equivalent for REST due to the fact it uses the standard HTTP interface and would only require some form of API for client developers to understand what resources are available.

2.4.4 Summary

Based on the research carried-out into the two implementation choices for Web services the decision has been made to implement RESTful Web services as part of the SOA PLM system. The major factor in this decision was that RESTful services can be said to be simpler due to their strict use of HTTP ‘CRUD’ operations. The RESTful PLM services will allow third-parties to develop clients to consume the services quickly and efficiently and the stateless nature of the services will ensure the PLM system is scalable.

2.5 Tools

As the distributed nature of a SOA can be viewed as an object-oriented problem, the decision has been made to implement the Web service in the object-oriented language of
Java. Sierra [12] describes how Java allows portability and has user friendly syntax. Experience of Java was also gained from the development of previous projects which meant time would not have to be spent learning the basic implementation language. The eclipse integrated development environment was chosen to be used due to its well documented support for Java.

As Java had been selected as the implementation language for Web services it was a sensible to decision to use Java Servlets to expose the PLM services over HTTP. Sierra explains that Servlets are “Java programs that run on a HTTP Web server” [12].

As Java programs require the Java Virtual Machine to run on desktops, a similar technology is needed to run the Servlets on the server. For this; the Web container Tomcat was chosen as it had been previously used in the Distributed Systems module.

The decision was made to use the Java API for RESTful Web Services (JAX-RS) as it is an official part of the Java Platform. To assist in development of the Web services the decision was made to use a reference implementation of the JAX-RS API. The choice was between two implementations; ‘Jersey’ and ‘Reslet’. The decision was made to use Jersey as it is used in tutorials on the Java EE 6 website [13] and seemed to be the most widely used API.

As Java technologies were being used for the majority of the service implementation; research was carried out to discover any other related technologies that may be useful during development. The Java Architecture for XML Binding (JAXB) which is part of the Java platform was chosen as a possible method to assist in the output of XML from the PLM services. Also, Java Script Object Notation (JSON) was identified as a possible alternative to XML. Haverbeke [14] explains how JSON is more minimalistic than XML. The decision was made to implement JSON as an optional alternative to XML for data exchange if possible. While investigating the Java Naming and Directory Interface (JNDI) the benefits of connection pooling were discovered. The main benefit was that enabling several methods to use the same connection reduced the number of expensive connections to a data source. It was decided that connection pooling would be implemented if possible when connecting to the PLM data.

As the services were going to RESTful; the decision was made to use basic HTTP and CSS to develop the user interface of the client application and to use Asynchronous JavaScript and XML (AJAX) technologies to send requests to the Web server. The reason for this was that XML and JavaScript (in the form of JSON) had already been chosen as implementation tools; it made logical sense to bring these technologies together to create the client application.
To store product data; MySQL was chosen as the technology to be used to implement a database in the PLM system. The reason for this was that it had been used in previous modules in the Degree and there were a wide range of tutorials and resources available on the internet. To make the installation and access to the database simpler; a Windows, Apache, MySQL and PHP (WAMP) stack was installed on the development environment.

2.6 Methodology

There are two development methodologies that will be investigated before a decision is made on which methodology to use; the waterfall model and the rapid prototyping model. The reason these two methodologies have been selected is because of their wide-use throughout the industry.

The waterfall method is a traditional development process that follows a sequential design. It consists of the required phases in development that must occur in order. The phases are; requirements specification, design, implementation testing and maintenance. The rapid prototyping method consists of several iterations of prototypes that will be discarded and not become part of the final application.

Carr [15] identifies that as opposed to the several iterations in the rapid prototyping model; the waterfall model is a single process with only one output stage. If this methodology was adopted only one version of the PLM system could be produced with no room for revisions for design or implementation.

Kelly [16] discusses how rapid prototyping allows for end-user involvement throughout development whereas the waterfall method only permits this during one stage and eliminates development problems early on in the process. It is clear that this methodology would allow frequent involvement of the external company throughout the project.

Service reusability is a key principle of a SOA and Steigerwald [17] explains how in addition to this; software reuse would be promoted by following a rapid prototyping methodology. It is therefore clear; any software developed during iterations of the PLM system could perhaps be re-used in the next iteration before they are discarded.

Based on this research it is clear that although the key steps of the waterfall method will be followed to produce the PLM system, an iterative style of development based on rapid prototyping is favourable. Kelly [16] explains how the front-end client and data processing in
a system should be developed separately during iterations. Therefore, the decision has been made to perform iterations on the apparel PLM services before development commences on the front-end client application.

2.7 Evaluation

The PLM system will need to be evaluated so the success of the project can be measured against its objectives and aims. Functionality will be specified in the next chapter so a potential evaluation method is to evaluate the system against these requirements.

Another method of evaluation could be a user evaluation using typical individuals who would eventually use the system. Noyes [18] describes how conducting a user study in the field is an effective method for gaining an insight into the success of the system.

The decision has therefore been made that as a minimum standard the system should be evaluated against the requirements that were specified and that if possible; an evaluation involving real users should be carried out.

2.8 Conclusion

The research carried out has been extremely useful for gaining a good understanding into previously unknown architectures, methods and technologies. Decisions have been made throughout this chapter based on information gained from the research.

Some important decisions have been made that will be vital throughout the project. An apparel PLM system based on a SOA will be developed that consists of RESTful PLM Web services which has the ability to manage the data, people and process required during each stage of the product lifecycle. This system is intended to be cheaper for apparel organisations to implement and be quicker to install as time consuming customisation operations will not need to be undertaken.
3 Requirements Analysis

3.1 Introduction

This final year project is an external project for PDP Limited. It is part of a larger project that I will continue working on in the future post graduation. The purpose of undertaking the process of Requirements Analysis is to put forward a ‘Business Case’ detailing the reasoning behind the project as a whole and to ascertain exactly what the PLM system should do in comprehensible terms for this project. This chapter will be split into two sections; the Business Case and System Requirements. The Business Case will analyse the current Apparel PLM market and existing systems based on research undertaken by PDP Limited to identify problem areas that could be solved with a PLM system based on a Service-oriented architecture. The System Requirements section will detail the scope of the project, the discovery of requirements, the construction of Use Cases and the specification of the requirements.

3.2 Business Case

This business case has been written for the overall project that I will continue working on after I graduate from University. The goals and schedule set out in this section are for the overall project and do not change those which have been stated in the previous chapter relating to this final year project.

3.2.1 Problem Statement

On the whole, customers of Apparel PLM systems have previously been large organisations with more than five thousand employees. Recently, the emergence of ‘Out Of The Box’ solutions has prompted previously unseen interest in Apparel PLM solutions from small and medium enterprises (SMEs). However, research undertaken by PDP Limited has discovered common issues exist that prevent many SMEs from purchasing a PLM solution. The
research also suggests that these issues do not only prevent potential SMEs purchasing a solution but also cause those who have already adopted a PLM system, including large organisations, significant complications.

PLM systems make use of their own proprietary data format for data exchange between various components of the system. Each vendor of Apparel PLM has their own file format and there exists no accepted standard for data in the industry. Many users of PLM software often possess the desire to extend their solution to enable them to work with other software packages or systems that were not in use by the organisation at the time of implementation. In this situation proprietary data formats are an obstacle as the data will be incompatible with these new systems or software preventing the customer from extending their solution how they wish. The practice of using proprietary data formats by vendors leads to vendor lock for the customer and is therefore very unlikely to change in the foreseeable future as the PLM vendor can create guaranteed revenue through being the only entity capable of extending the customer’s solution.

Despite the emergence of ‘Out Of The Box’ solutions in recent times, high cost remains a contributing factor to the decision of organisations not to purchase a PLM system. This high cost does not only consist of a monetary cost but also includes a time cost. Vendors often tailor solutions to the client and will customise their solution for each potential customer; justifying the high price set for the system and subsequent licenses. A vendor will also encourage a customer to purchase a support contract, adding to the total price of a PLM system. The prolonged length of time taken while implementing a PLM system in an organisation with the vendor also further justifies a high monetary cost. A typical Apparel PLM system implementation takes between four and six months, but often delays can push this up to eight months. Taking into consideration other necessary steps like the customer selecting a PLM system, negotiations on price, the implementation and user training with the new system the whole process can take over a year. This high monetary and time cost of purchasing a PLM system will inevitably prevent some potential customers from taking the decision to implement a PLM system in their organisation.

Prices for PLM systems will also include a price per user. Each user that the organisation wishes to utilise the PLM system requires an annual licence, priced by the vendor. In the apparel industry products are developed at least four times a year with short lead times. This causes the number of users of a PLM system to fluctuate greatly. For example, development of a summer season apparel product may require a large percentage of the design team using the system whereas for the following season this percentage may be reduced greatly if there is considerable ‘carry-over’ (where an old design is reused with different colours or
materials). In this scenario a customer is paying for annual licences for every user when there may be only a small percentage of users actually utilising the PLM system. An apparel organisation may have up to one hundred suppliers for their products, unlike organisations in other industries that use PLM systems who may only have one or two closely vetted suppliers. Ideally these suppliers require access to the system but again users will fluctuate. Apparel organisations will often only use a given supplier for one product in one season, however this requires purchasing a licence for this supplier, also, organisations arrange reserve suppliers for any unforeseen problems with their chosen supplier. If the reserve supplier needs to be utilised a new licence must be purchased or the old licence needs to be transferred over, at a cost to the customer. Taking into consideration that an organisation may have to arrange licences for up to one hundred suppliers it is accurate to say this process is complicated and logistically problematic.

Customers of a PLM system will customise their system over time to help streamline their business processes. However, a situation may arise when the customer desires to upgrade to a new version of their current system or to a completely new PLM system from the vendor. Customisations can often be ported over to the new system but not always without problems Upgrading is not a simple task for organisations especially when there may be projects underway using the current system involving many parts of the product lifecycle. Also, a major system upgrade will require subsequent investment in further staff training so they can learn to use the new system. As upgrading is seen by customers as a complicated process, it is a common occurrence for updates to be neglected resulting in one large update being implemented which can mean a PLM system is unusable for a period of time.

### 3.2.2 Project Description

The overall goals of the project are:

- To remove the dependency on proprietary data formats within the apparel PLM industry.
- To reduce the total monetary cost of purchasing and implementing a PLM system so apparel organisations of all sizes can realise the benefits of implementing a PLM system.
- To reduce the time required for organisations to implement a PLM system within their business.
To abolish complicated licensing agreements for the use of PLM systems by organisations and their associated partners.

To transfer the burden of upgrading a PLM system from the user to the vendor.

The estimated completion date for the project is August 2013.

The resources required for the project are:

- Access to existing PLM systems for research.
- Access to experienced users of PLM for research.
- The regular availability of experienced internal PLM Consultants.
- One full time developer.
- A stable Web server.
- A local development server.
- A reliable hosting server.

### 3.2.3 Solution Description

The solution to the discussed problems will be the development of an Apparel Product Lifecycle System that is based on a set of core PLM services. The system should:

- Be based on a Service-oriented architecture.
- Make use of standard data transport methods throughout.
- Have a Web browser-based front-end.
- Be based on a ‘Software as a Service’ (SaaS) pricing model.
- Provide its data and software over the internet (hosted in the “Cloud”).

Central to the Apparel PLM System will be a core Product Data Management System (PDM System) that will control product data. The PLM system should integrate; data from the PDM, tools, processes and people that are required in each of the following stages of a product’s life:

- Conception (Design Inspiration).
- Design (Illustrations, Storyboards).
• Realisation (Manufacture, Quality control).
• Service (Sell, Deliver).

3.2.4 Assumptions and Risks

• Assumptions:
  o Each function of the PLM system will be exposed as a service that can be consumed not only by PDP Limited’s planned front end but additionally by third parties who will adhere to a SaaS pricing model.
  o The core PDM System will be developed by PDP Limited.
  o The tools, processes and people required for each stage in the Product Lifecycle are already in standard use throughout the apparel industry so will not be developed by PDP Limited.
  o Example data for the system will be provided by PDP Limited.

• Risks:
  o The SaaS pricing model may fail to recoup development costs.
  o Apparel organisations may be reluctant to store their product data with another organisation.
  o The major vendors in the apparel PLM industry with whom we have an affable relationship may be averse to change.

3.2.5 Conclusions and Recommendations

The problems discussed in this document are widely acknowledged across the apparel PLM industry. Resolving these issues will enhance the apparel industry as a whole and promote the continued improvement that apparel organisations realise through the adoption of a PLM system. While it is important to maintain the company’s trusted and impartial standing to protect other revenue streams; PDP Limited is in good position to bring about beneficial change for the industry and an increase in the uptake of apparel PLM systems will be advantageous for all parties involved.
Recommendations:

- The benefits of the PLM system outweigh the risks and costs.
- The project should go ahead.

### 3.3 System Requirements

#### 3.3.1 Project Scope

Based on research carried out during background reading, the business case analysis and discussions with both my project supervisor and external company manager, it has become apparent that to produce a fully functional apparel PLM system for my final year project is too ambitious due to the limited time that is available for working on the project. Therefore, the scope of this project will be focused on the key component that is the core of all PLM systems; the Product Data Management (PDM) system. Development of a PDM system based on services will provide a basic Service-oriented apparel PLM system that can then be further developed into a fully functional PLM system, after the project is completed.

#### 3.3.2 Requirements Gathering

Requirements gathering is the process of extracting knowledge from individuals who will be involved in a given system (stakeholders) and converting this knowledge into requirements that the system must meet. For the basic apparel PLM system I have identified the following stakeholders who are employed by the external company:

- The Managing Director and Technical Director have both been involved in apparel PLM systems since they first emerged around 2003 and can be considered domain experts. They will provide key knowledge that will give valuable insight into the expectations organisations have regarding a PLM system’s functionality.
- My Manager has experience of managing apparel PLM implementations in organisations of various sizes and also has a Computer Science degree. This individual will give insight into how the PLM system should differ from those already available in order to solve the problems identified.
Several Consultants who work at my external company currently use or have previously used apparel PLM systems. These individuals will provide me with an understanding of what end-users will expect from a basic PLM system.

The Chief Technical Consultant has experience of designing and implementing various components of a PLM system and will be able to provide a technical understanding of a basic apparel PLM system.

These individuals have been selected as I believe they can provide a comprehensive understanding of what the basic apparel PLM system should be able to do. Included in the selection are individuals that provide insight from various perspectives as opposed to one perspective and individuals who are able to give technical descriptions due to their current and previous roles.

The key method used to extract knowledge from the stakeholder's was informal interviews. Each individual was asked to give their understanding of the PLM business methodology used in the apparel industry system at a basic high level. Specifically, individuals were asked for the various processes that are commonly carried out, the key individuals and their involvement and any data that is passed between processes. From the discussions a process flowchart was produced. This can be seen in Figure 3.

![Figure 3: PLM Process Flowchart](image-url)
The domain experts that I had access to were presented with this flowchart and confirmed that it was accurate. Using the flowchart, discussions were held with each individual focusing on exactly how they would expect a basic PLM system to facilitate each process. Specifically, information was requested on who would use the system, exactly how they would use it and what, if anything was produced. This information has been summarised into a table of use-cases which can be seen in Appendix D.

Again, due to time restrictions and the types of example data that are available to me it is not possible to produce every use-case discovered. Therefore the decision has been made to choose a subset of these use-cases to implement as a Web service. A simple browser based client to consume the service and demonstrate its functionality will also be developed. Functionalities selected for implementation have been chosen on the basis that they differ technically in their basic operation, this is so the PLM service can provide a range of functionality that is as close to a basic PLM system as possible in the time available.

Knowledge extracted from stakeholders, information discovered during background research and features from the business case have been converted into the following functional (services the system should provide) and non-functional (global constraints on the system) requirements:

- **Functional Requirements:**
  - The ability for a buyer to insert a new style into the PLM.
  - The ability for a designer to search for a specific style by code.
  - The ability for a manager to see how many styles require approval.
  - The ability for a merchandiser to retrieve all style details for a collection book.
  - The ability for a manufacturer to remove a style that will not become a full order.
  - The use of XML for data transport.
  - The use of JAXB for XML manipulation.
  - A MySQL database to persist style data.
  - A browser-based example client to consume the service and allow user input where appropriate.
• **Non-functional Requirements:**
  - The service should be a RESTful service.
  - The service should be stateless.
  - The service should be scalable.
  - The service should be loosely coupled.
  - The service should use a standard interface.

The following requirements are not essential for a basic service based PLM system but are desirable and will be beneficial to any further development:

- User types will have differing access permissions.
- The ability to use JSON as opposed to XML for data transport.
- The use of a JNDI connection pool to maintain performance.
- The ability to input complex data types, for example JPEG thumbnails.

The requirements that have been discussed and specified in this chapter will be designed and implemented in order to obtain a basic apparel PLM system that is based on a Service-oriented architecture.
4 Methodology and Design

4.1 Introduction

This chapter will discuss the methodology and schedule that will be adopted to facilitate the achievement of the project objects. Also, the design of the solution to the problems identified in the previous chapter will be detailed using UML and other diagrams. As concluded in the background research UML has been selected due to its ability to break up a system into easily understandable elements and due to its widespread use.

4.2 Methodology

4.2.1 Chosen Methodology

During the literature search and background reading, a requirements analysis was carried out with the external company to obtain requirements for the service-based apparel PLM system. This chapter will detail a design that is intended to act as a guide for implementation of the proposed solution. Once implementation is completed an Evaluation will be carried out to measure the success of the solution in solving the problems identified.

Based on background research the decision has been taken to undertake an iterative style of development based on the rapid prototyping methodology. As this project is intended to be continued as a long term project a prototyping methodology is ideal due to its speed for the short term development and frequent input from the stakeholders.

Originally, the decision was made to perform two iterations for the PLM Web service and two iterations for the front-end client, with iterations containing a design, implementation and test phase. However after gaining feedback from the mid-project report it was clear that there would not be enough time to perform two iterations for each. Therefore the decision has been made to perform two iterations of development for the PLM Web service and one iteration of development for the client application. This will enable feedback to be received regarding the Web service from my external company Manger and allow any necessary amendments to be made before development begins on the front-end client.
4.2.2 Schedule

The original schedule which can be seen in Appendix E did not assign enough time to the first iteration of the Web Service as the time required to understand how to use the selected technologies was not factored into the two weeks assigned to implementation. The schedule also lacked key milestones.

The revised schedule which can also be seen in Appendix E reflects the revised iteration plan. Extra time has been assigned to the implementation of the PLM Web service and the second iteration of the Web service development has been scheduled to begin after the first. Implementation of the client is scheduled to overlap with the second iteration of the Web service implementation to enable the service to be tested with a working client application. Key milestones have also been included in the revised schedule.

4.2.3 Milestones

Shown below is a list of Milestones and their respective completion dates, they can be used to determine if the project is on schedule and to identify the key objectives that need to be achieved in order to fulfil the objectives of the project.

1. Complete a REST ‘Hello World’ example using the selected technology - 22nd Feb
2. Present to the research group progress to date and upcoming challenges – 23rd Feb
3. Design the database and insert the example data – 29th Feb
4. Submit Mid-Project report – 2nd March
5. Submit Draft Chapter and Table of Contents – 9th March
6. Finish development of solution – 30th March
7. Complete Progress Meeting – 20th April
8. Submit final report – 9th May
9. Submit electronic copy of report – 11th May
4.3 Design

4.3.1 Traditional Apparel PLM System

Figure 4 is a high level abstract diagram of the components and people that are managed by a traditional apparel PLM system. In using the PLM system, various teams within an organisation can access and manipulate product data as they require to carry out their respective jobs. The diagram shows that typically, the responsibility for manufacturing the product and the logistics of the product lies with external teams or organisations that are outside of the organisation that has implemented a PLM system. These external
organisations require access to the product data and must do so through the use of the same PLM system; this is where extra licence costs are incurred by the apparel organisation. The diagram also demonstrates the importance of the Product Data Management System to a PLM system and how teams at each stage of the product lifecycle use it to access product data regardless of their position in or out of the apparel organisation.

4.3.2 Apparel PLM System Based on a Service-oriented Architecture

Figure 5: High Level Architecture of a Service-oriented Architecture Apparel PLM System

Figure 5 is the proposed high level architecture of a Service-oriented apparel PLM system. This shows how the final completed system should operate. The product data for several organisations or users will be stored on the product data database. Through use of the front-end client, apparel organisations or individual users can have access to the PLM Web service. Other third-party apparel PLM systems may use the Web service directly to access up-to-date trend or points-of-measure data.
4.3.3 PLM Web Service Design

Figure 6: PLM Web service design

Figure 6 is the design for the Web service that is intended to meet the requirements set out in the previous chapter. Using the Uniform Resource Identifier (URI) of the RESTful PLM Web service the client application will make a HTTP request using one of the following standard HTTP methods; GET, POST or DELETE along with the necessary parameters if the request is a POST request. The applicable method of the Web service will be called for the request where processing will be performed to build a query that will be executed on the MySQL product database. Once the query has been executed successfully and a result received the data or response will be converted to XML and sent back to the client application via HTTP. The products in the database will be termed as “Styles”.

4.3.4 Database Design

The database will be a MySQL database consisting of one table to persist product data and is designed to incorporate the basic string data that is required for a basic PLM system. Ideally, data stored regarding a product would be vast and incorporate other data types. For example, a thumbnail image would be stored alongside a product to assist users in understanding what the style is at a glance. However, string data is the only available data type that exists in the example product data received from the external company. This String data will be used primarily for the database but if possible, other examples of data types will
be obtained. The data fields selected do not match exactly to the example data supplied as the decision was taken to remove fields that were similar as they add no value to the basic PLM system based on the fact that data schemas will be different for every user.

Figure 8 Error! Reference source not found. shows the fields that will be included in the items table; they will all be ‘varchar’ fields. Item ID’s will follow a pattern that is common in the industry and the possible values for status will be; “Concept”, “Design”, “Approved”, “Build”, “Testing” and “Sale”. Figure 7 Error! Reference source not found. shows an item long with example data in the XML form that it will be transported in.

4.3.5 Web Service Methods

4.3.5.1 Add Style

Figure 9 shows the first method that must be available to a Buyer when using the Web service. To use this method the client application will send a HTTP request using the POST method to the URI:
The data required for the new style will be included in the request's POST parameters, a MySQL INSERT query will then be executed and a response received. The data fields that will be required are:

- itemID
- itemCategory
- itemSubcat
- itemDescription
- itemStatus

4.3.5.2 Search for Style

Figure 10 shows the method that will be available to a Designer when using the Web service. To use this method the client application will send a HTTP request using the GET method to the URI:

- http://<ServerURL>:<Port>/testProject/rest/items/<itemID>

The data required to search for a style will be the itemID of the desired item. This must be included in the URI of the HTTP request. A MySQL SELECT query will be performed and the full details of the desired item will be returned to the client.
4.3.5.3  **Count Styles requiring approval**

Figure 11 shows the method that will be available to a Manager when using the Web service. To use this method the client application will send a HTTP request using the GET method to the URI:

- `http://<ServerURL>:<Port>/testProject/rest/items/count`

No data will need to be sent as parameters for this request, a MySQL SELECT and COUNT query will be performed and the number of items that are at the design or concept stage will be returned.

4.3.5.4  **Retrieve all Styles**

Figure 12 displays the method that will be available to a Merchandiser when using the Web service. To use this method the client application will send a HTTP request using the GET method to the URI:

- `http://<ServerURL>:<Port>/testProject/rest/items/`
No parameter data will be required for this method, a SELECT ALL MySQL query will be performed and the full details of all styles will be returned to the client.

### 4.3.5.5 Remove Style

![Figure 13: Remove Style Use Case](image)

Figure 13 shows the method that will be available to a Manufacturer when using the Web service. To use this method the client application will send a HTTP request using the DELETE method to the URI:

- `http://<ServerURL>:<Port>/testProject/rest/items/<itemID>`

The data required to remove a style will be the itemID of the desired item, this must be included in the URI of the HTTP request. A MySQL DELETE query will then be performed on that item.

### 4.3.6 Front-end Client

As discussed during the background research and requirements analysis chapters, the basic client application that will be used to demonstrate the functionality of the PLM Web service will be browser based. Figure 14 shows the intended layout for the browser. The simple user-interface will be produced using HTML and CSS and the client-side processing will be developed using AJAX related technologies.
When a button is pressed by the user a XMLHttpRequest will be made to the PLM Web service where the relevant method will be called. Any data parameters that are required for the desired method will also be sent using the GET or POST variables. Once a response is received from the server, the returned XML will be parsed and displayed in the table. AJAX enables the JavaScript to update the table without reloading the whole web application.

4.3.7 Revised Front-end Client

Due to an issue that arose during development of the client application, a change to the design of the client application was necessary; the reason for this change is discussed in the next chapter of the report. A discussion with my external company Manager regarding this issue concluded that for this project a simple desktop-based client application would be a suitable replacement for the Web-based client application. It was agreed that the main purpose of the client application is to demonstrate the functionality of the PLM Web service
and that a desktop client application would still be able to achieve this goal. Therefore, the new design can be said to be low-fidelity in terms of the user interface quality but high-fidelity in the sense that it allows the functionality of the Web service to be demonstrated thoroughly.

![Figure 15: Revised Menu Design](image)

Figure 15 shows the revised layout for the client application. The user will be presented with a simple window with the four possible methods that are available on the PLM Web service. Selecting Count Items will send a HTTP Request to the relevant method on the Web service and the result will be displayed in a pop-up alert box. Elements from the original design have been used for the remaining methods. Selecting Retrieve All will open a new window displaying the data, as will Search Style; although first an input box will pop up for the user to enter the Style ID. Similarly, selecting Delete Style will present the user with an input box for the Style ID of the item they wish to remove. Finally, selecting Add Style will display a window with the relevant fields that need to be completed for a new style and a confirmation button. The application will be developed in Java as the Jersey reference implementation that is used for implementing the RESTful PLM Web service contains API’s for developing a RESTful client application, this will hopefully make up for the time lost while partially implementing the original browser-based client application.
5 Implementation

5.1 Introduction

This chapter will detail the implementation stage of the project. Based on background research, requirements analysis and previous experience the decision was made to implement a RESTful PLM Web service using Java Servlets and the Jersey API's. A back-end MySQL database would be used to store product data and XML would be used for any data exchange. The Servlets will operate within a Tomcat Web container running on a 'localhost' server on a machine that has a 'WAMP' stack installed (Windows, Apache, MySQL and PHP). After completion of the required methods in the first iteration of the PLM service implementation a second iteration of development would be undertaken to make any necessary changes and if possible implement the use of a JNDI data source and JAXB binding.

5.2 Hello World

Previous experience with the tools and technologies selected for implementation based on background research is limited, therefore the decision was made that a “Hello World” RESTful Web service should be implemented, based on a tutorial that demonstrates how the selected technologies for the solution work at basic level [19].

The first step was to update the Eclipse Integrated Development Environment with the Web Tools Platform via the built in update manager. The reason for this update is so that Dynamic Web Projects can be created as opposed to a standard Java projects. Dynamic Web projects have the ability to contain dynamic Java EE resources, some of which are used during implementation; for example Servlets and Metadata.

A Dynamic Web Project was created called 'HelloWorld' which produced a hierarchical folder structure for various elements to be placed. To use the Jersey API's the relevant JAR files need to be available to the project so these were copied into the library in the ‘WebContent’ folder. A file called web.xml is also located in the ‘WebContent’ folder which is used to specify parameters and settings required for the Servlet. In this file details were entered to
enable the Jersey Servlet container to be initiated when the service was launched. These three actions were always performed when creating a new project and will be referred to as the 'initial setup' in this report. One class was created in the project called HelloWorld with one method that returned some example XML.

For deployment of the RESTful Web service, the eclipse IDE had a built in function called 'Run on Server'. This enabled the service to be deployed on a Tomcat Web server and be accessible through the URI that was constructed in the web.xml file and the HelloWorld class. This URI and the output received from the Web service can be seen in Figure 16. The successful completion of the Hello World example provided a good understanding of how the various technologies interact and work together. Also, an understanding of the key elements required for a RESTful Java Web service was obtained, for example the Java annotations of; '@GET' and '@Produces' which indicate the type of method and the data type of the output.

5.3 PLM Web Service – First Iteration

The first iteration of the Web service implementation will be the development of the methods that need to be available for users of the Web service. The implementation for each of the methods will be explained.

5.3.1 Methods

A new project was created and the initial setup was performed. To develop the Web service, example data was required so a data model class and data provider were created. The data model was a class called ‘Item’ which required an ID and description to be passed as parameters when a new instance of the class was created. A get and set method for both
the ID and description were added to the Item class. The data provider was a class called 'ItemDAO'. The class created three example items using hard-coded data and returned as a List of the three items.

Two classes were required for the Web service methods; the 'ItemResource' class and the 'ItemsResource' class. The Items resource class is the main base resource and contains most of the methods of the Web service, the Item resource class is used when operations are called on existing items.

At the beginning of the Items Resource class the Java annotation '@Path' was used. This annotation is used by the compiler to set the URI of the Web service along with the base URL which is obtained from the application name and URL pattern that are specified in the web.xml file. The Path annotation was set to '/items' meaning the full URI of the service to be used was:

- http://localhost:8080/testProject/rest/items/

The first method implemented was the 'retrieve all styles' method. The '@GET' annotation was used to identify that this method should potentially be the method called when a HTTP GET request is made to the service. Use of the '@Produces' annotation allowed the method to be identified as a method that returns XML to the client. The combination of specifying the method as a GET method and one with a return type of XML results in this method being called for any HTTP Get requests that specify a return type of XML. A list of items is created by the method and then populated by data from the previously implemented data provider.

The next method to be implemented was the 'count styles requiring approval' method. As the final data to be used was not yet accessible by the service, this method was temporarily implemented to count the total number of items in the data provider. As for the previous method the @GET annotation was used as no parameters were required and the @Produces annotation was set to XML. However, in addition to the two annotations, another annotation was introduced; '@Path'. The purpose of the annotation on this method was to change the URI so the method can be called using the following URI:

- http://localhost:8080/testProject/rest/items/count

The 'add style' method was annotated with the '@POST' annotation to specify that it should be called when a HTTP Post request is made to the service. Inside the class definition the annotation '@FormParam' was used to get the Post variables of ID and description from the HTTP request and assign them to local variables. A new item was created with the local variables and added to the data provider.
The final two methods were contained in the second class; Item Resource. The ‘get style’ method was annotated with the @GET annotation and the @Produces annotation which specified XML will be produced. The method retrieves the desired item from the data provider by using a global variable which holds the desired ID. This global variable is created by the Items Resource class when an item is specified.

Similarly, the ‘delete style’ method obtains the desired style ID from the global variable. The method was annotated with the ‘@DELETE’ annotation that specifies it should be used for any HTTP request using the delete method. The item with the ID that is specified by the global parameter is removed from the data provider when this method is called.

As the base URI was specified in the Items Resource class, this resource class will be used by default whenever the service is used. To enable the ‘get style’ and ‘delete style’ methods to be accessible a method was created in the Items Resource class called ‘getItem’ which returns a new instance of the Item Resource class. To create this new instance an ID is required of the desired item to initialise the class. The @Path annotation was used to specify that any further strings added to the URI of the service when consumed should be treated as a @PathParam, which is then used to obtain the desired ID and to create a new instance of the ItemResource class meaning the ‘get style’ and ‘delete style’ methods are accessible. To use the methods a HTTP Get or Delete request must be made to the URI:

- http://localhost:8080/testProject/rest/items/<ID>

5.3.2 Testing

The ‘select all styles’ method and the ‘count styles requiring approval’ method could be tested using a standard Web browser as navigating to a URL or URI in a browser results in a HTTP Request being sent using the Get method. Once the Web service had been deployed using the IDE’s inbuilt functionality it was available on the ‘localhost’ server. Navigating to the URI for the ‘select all’ method displayed all the information from the item data provider in the browser in XML form as expected. Similarly navigating to the URI for the ‘count styles’ method produced the same result; XML was displayed indicating the number of items in the data provider.

As the remaining methods required parameters to operate; browser-based testing was not sufficient as it lacked the ability to send GET, POST and DELETE methods with custom parameters. Two options for testing the services were explored. The first was to use a
Firefox ‘add-on’ called ‘Tamper Data’ which allowed HTTP Requests to be modified. This method worked to an extent but the application is designed to modify requests not initiate them so it was quite cumbersome to use.

![Figure 17: HTML Post Form](image)

The second method explored for testing the services was to build a simple HTML form that would specify a Post, Get or Delete method. Figure 17 shows the form used to test the ‘add style’ method. Selecting Submit after entering an ID and description would send a HTTP Post request to the Web service where the ID and description parameters were added to the data provider. Similar HTML forms were used to successfully test the ‘get style’ and ‘delete style’ methods.

5.4 PLM Web Service – Second Iteration

On completion of the first iteration, the services were demonstrated to the external company Manager; positive feedback was received on progress and the request was made that the product ID should be described in XML as an attribute of an ‘Item’ as opposed to its current form where it was displayed as an element of an ‘Item’. The reason for this request was stated to be that in the Manager’s experience of parsing XML, storing an ID as an attribute is much more efficient requiring less processing. It was agreed that this change would be applied when implementing JAXB binding. The second iteration of the Web service implementation covers replacing the temporary ‘hard-coded’ data with data from the MySQL database, the use of JAX-B for XML output and the implementation of alternative JSON methods.
5.4.1 MySQL Database

A MySQL table named ‘items’ was created following the design specified in the previous chapter. The example data received from the external company which can be viewed in Appendix B was in a Microsoft Excel file. This was edited to match the database design and saved as a CSV file. The ‘WAMP’ stack used for implementation had an application called ‘phpMyAdmin’ that could be used to manage the database. The import function in this application was used to upload the data from the CSV file to the items table.

Based on background research, it was clear that the use of the Java Naming and Directory Interface (JNDI) to implement connection pooling was advantageous for performance and would be necessary for the final application. Therefore the decision was made to use this method to connect the database as opposed to connecting to the database in each method each time it was called. The data source was specified in the web.xml file and connection settings were obtained from a file called context.xml. In the constructors of the ‘Items Resource’ and ‘Item Resource’ class the connection to the database is made. The JNDI data source is initialised once meaning any queries made to the product database use the connection that has already been established, this improves the overall speed of the PLM Web service.

For each method available on the Web service, instead of using the ‘hard-coded’ data provider changes were necessary to utilise the MySQL database. Any references to the data provider were changed; a connection was first obtained from the previously initialised data source, a MySQL query was then formed taking into consideration any necessary parameters and finally, the statement was executed and a result was received. The ‘count styles’ method was also changed to count the number of styles in the database that had a status of ‘concept or ‘design’.

Finally, the ‘Item’ class was updated to model the database schema, including the necessary get and set methods for all the fields of a product style. Also, a new class ‘Items’ was produced to return a collection of Items to make the output of XML more efficient.

5.4.2 Java Architecture for XML Binding

Previously, XML output was obtained by manually returning a long string that included XML mark-up. This method was suitable for small amounts of data such as the result of the ‘count
styles’ method, but for methods such as the ‘select all styles’ which were now returning sizeable amounts of data from the database, the manual method was inconvenient and prone to errors.

Therefore, based on these factors and information obtained from the background research the decision was made to use the Java Architecture for XML Binding (JAXB) tool for manipulating XML. So that the tool could be used; Java annotations were added to the data model ‘Item’ class. These were; ‘@XmlRootElement’ to specify that an item was to be the root node in the XML structure and ‘@XmlElement’ along with ‘@XmlAttribute’ to specify how data fields should be manipulated for XML output. The addition of these annotations to the data objects enabled return statements in the methods of the PLM service to return an Item object or a collection of Items without manually adding any XML mark-up tags. The objects would automatically be returned as XML.

5.4.3 JavaScript Object Notation

Although XML is widely used and has become the standard method for data interchange there are some advantages in using JavaScript Object Notation (JSON) as an alternative, which are discussed in the background research. As specified in the requirements analysis, giving the consumer of the PLM Web service a choice over which data interchange format they desire is advantageous and good practice.

Therefore the Web service methods were refined to enable the consumer to request JSON as opposed to XML. To achieve this, the methods were re-implemented with a change in Java annotations. Instead of the ‘@Produces’ annotation stating: ‘TEXT_XML’, they were also changed to include ‘APPLICATION_JSON’. Both XML and JSON could then be returned from the method and it would be up to the consumer of the PLM Web service to specify which they desire.

5.5 Client Application

The purpose of developing an example client application is to demonstrate the functionality of the PLM Web service in a scenario that is similar to how the final service based apparel PLM system will be used once it is completed. This client application can represent a client
developed by PDP Limited to enable access to the PLM services and allow them to create a revenue stream from its use. Also the client application could represent a third-party apparel PLM system making use of the services.

The first step in producing the client application was to re-use the HTTP form: 'create_item.html' that was designed for testing the first iteration of the Web services. This would provide a basis to start building the client application. A script containing an ‘XMLHttpRequest’ using the POST method along with appropriate variables was created. The script was initiated when the ‘Submit’ button was pressed on the HTML form. However, the Web service request was not successful and the server did not receive any communication from the example client application after several attempts.

After a period of trial and error debugging, the reason became clear as to why the client application could not consume the Web service. The client was violating the cross-domain security policy of JavaScript and although cross-domain issues had been experienced in previous projects with alternative technologies, the security policy was overlooked because the Web service and client were in fact running on the same domain ('localhost') but not on the same port.

Due to the time that would be required to change the development environment and the fact that knowledge of how to solve the issue was limited, the decision was taken, with consultation from the external company Manager that it would be acceptable to produce the example client application in the form of a desktop application.

### 5.6 Revised Client Application

The time spent partially developing the original client and investigating the problem that arose meant that development of the revised client application needed to begin straight away in order to minimise delays. It became clear that the client application should therefore be developed in Java utilising the Jersey API’s that assist in developing RESTful client applications, as those technologies had already been used for the Web service and therefore would require less familiarisation time.

The application contained a class called ‘Client’ in which there were several functions to consume each of the methods available on the PLM Web service. To consume the Web the service methods each function obtained the ‘client’ object created in the constructor and set
its resource URI to that of the Web service. The desired HTTP method was specified calling the appropriate method on the resource and passing any parameters that were required for the method to operate.

![Figure 18: Insert Style Input](image)

The application enabled the user to input parameters for the methods through the use of on-screen prompts; Figure 18 shows the main menu and resulting pop-up window for the ‘insert style’ method. Methods that returned results that were of interest to the user were displayed in a table or alert box; Figure 19 displays the result window for the ‘view all styles’ method.

![Figure 19: Results Window](image)
Due to the delay caused by the problem encountered while implementing the original client design, the run-over time scheduled was used up implementing the revised client application and using the application to test the Web service further. During testing, a problem arose while attempting to consume the ‘delete style’ method on the PLM service; an HTTP ‘405 Error’ was being returned from the service, which meant the DELETE method was not allowed. With the run-over time having already been used, the decision was made not to spend any extra time solving the issue. The reason for this was that limited investigation had discovered the problem was most likely caused by the current settings of the local Tomcat server which denied HTTP DELETE or PUT requests, and that there was no problems with the PLM service itself. Therefore, development of the solution was complete and Milestone 7 had been achieved through use of the pre-planned run-over time.
6 Evaluation

6.1 Introduction

The purpose of this chapter is to obtain an understanding of how successful the project has been and to assess how well the solution that has been produced solves the problems identified towards the beginning of the report. The main sections in this chapter will aim to; evaluate the solution from the perspective of the external company, evaluate the solution against the requirements set out in Chapter 3, evaluate the solution against alternative systems and to evaluate the project as a whole.

6.2 Steering Development of an Apparel PLM System towards a Service-oriented Architecture

The business case in Chapter 3 identified overall goals that will hopefully be achieved by producing a Service-oriented apparel PLM system. Although the goals specified are long term and not compulsory for this final year project, the decision has been made to re-assess these goals taking into consideration the project experience and challenges discovered.

The first goal was to ‘remove the dependency on proprietary data formats within the apparel PLM industry’. The data used in this basic apparel PLM system was taken from real example data used by apparel PLM systems, however, the amount and type of data used was suited to a basic Product Data Management component of a PLM system. Full apparel PLM systems not only transfer basic product data types but rely on more complex types. Typically an ‘Adobe Illustrator’ file will always accompany a product in the PDM containing a technical sketch of the product. Taking into consideration the fact that there will be several revisions to this file and other attached components such as thumbnails accompanying products in the PDM it is sensible to compile these into one master file. The basic Service-oriented PLM system in this project uses XML and JSON for data exchange with a data object that has the potential to describe the file paths to ‘Adobe Illustrator’ files or thumbnails just as the proprietary PLM data formats do. The key difference is that the proprietary files store the paths inside the file whereas this project shows how data can be obtained from a data store
and provided as the result of a service. Therefore it can be said that this overall goal is achievable; storing data in a database and using a service to obtain and return it to the user means the requirement for a custom file format is eradicated in a Service-oriented apparel PLM system.

The goal of ‘reducing the total monetary cost of purchasing and implementing a PLM system’ is certainly achievable based on the experiences of this project. The major cost when adopting a PLM system is paying for lifecycle processes to be implemented within an organisation. As the processes will already be available as a Web service the main cost will be for using those services, which will certainly be less. In this project, developing a Client to consume already existing services was not an awkward or inconvenient task so it can be said that if an organisation was to develop their own client to consume the services this would also be cheaper than implementing a typical PLM system.

The goal to ‘reduce the time required for organisations to implement a PLM system’ has been demonstrated to an extent in this project. Even though a problem occurred during development of a Client application the time required to do so was relatively low. If an organisation wishes to develop their own Client application to consume the PLM services it can be said that this would be quicker than implementing a traditional apparel PLM system. Use of the planned Web-based client will reduce the time required even further.

The very nature of the PLM Web service demonstrates that the goal of ‘abolishing complicated licence agreements for the user of PLM system’ is certainly achievable. Consumers of the services will pay only when they need to use them and although use of the service may be restricted to a specific purpose through a licence agreement, agreements that specify minimum lengths of use will be non-existent.

Finally, as the service and possibly the client application are based ‘on the cloud’; the apparel organisation will not need to consider any of the challenges that occur when updating a PLM system. PDP Limited will be responsible for updated versions of the PLM services and client application.
6.3 System Requirements Evaluation

This section will evaluate to what extent the solution meets the system requirements stated in Chapter 3. Each requirement will be stated with a discussion on how the solution does or does not meet this requirement. The decision has been made to use the functional and non-functional requirements as evaluation criteria. This is because the functional requirements describe the fundamental processes that the system must be able to perform and the non-functional requirements describe the benefits of using a RESTful Service-oriented apparel PLM system.

6.3.1 Functional Requirements

- Can a buyer insert a new style into the PLM?

The apparel PLM system allows a user to insert a new product style into the basic PLM system through use of the ‘add style’ method. This can be done using the example client application which requests information about the style such as the ID and description to be entered. The style is added to a product database. Also the service can be used without the client by sending a HTTP POST request with the correct parameters to the service URI.

- Can a designer search for a specific style by code?

The apparel PLM system allows a user to search for a specific style that already exists in the PLM system through use of the ‘get style’ method. The method can be consumed using the example client application which requests the user to input the desired style ID, if it exists in the database the details of the product will be returned. The service can also be used by sending a HTTP GET request with the desired style ID as a parameter.

- Can a manager obtain the number of styles requiring approval?

The apparel PLM system allows a user to determine how many styles are in the PLM system that require approval through use of the ‘count styles’ method. This can be done using the example client application which displays in a pop-up alert box the number of styles that currently have a status of ‘concept’ or ‘design’. The service can also be consumed without the client by sending a HTTP GET request to the correct service URI.

- Can a merchandiser retrieve all style details for a collection book?
The apparel PLM system allows a user to retrieve all styles that exist in the PLM system through use of the 'get all styles' method. The method can be consumed using the example client application which returns the details of all styles and presents it in a table. The service can also be consumed by sending a HTTP GET request to the service URI.

- Can a manufacturer remove a style that will not become a full order?

The apparel PLM system does not allow a user to remove a style through use of the example client application. As previously discussed this is due to security settings in the development environment. However this service can be used without the client by sending a HTTP DELETE request with the desired style ID to the service URI.

- Is XML used for data transport?
- Is JAXB used for XML manipulation?

Each method that exists on the Web service is capable of consuming XML. Through the use of JAXB; any output is also delivered in XML format. The data objects in the system are annotated using JAXB notations which ensure correctly formatted XML is returned from the service.

- Is data persisted in a MySQL database?

Product data was implemented in the second iteration of development. The data was uploaded to the MySQL database which was then hooked up to the Web services using a JNDI pooled data source.

- Does a browser-based example client that can consume the service and allow user input where appropriate exist?

This functional requirement was not achieved. Due to unforeseen development issues a browser-based client to consume all the methods of the service was not produced. A simple HTML client was developed to test some of the methods but it was unable to consume all the methods on the PLM service. As the client application's main purpose was to demonstrate the functionality of the Web service while allowing user inputs a Java desktop-based client was developed instead.
6.3.2 Non-functional Requirements

- Is the service a RESTful service?

The Web service implementation strongly adheres to the REST principles and was developed using JAX-RS; the Java API for RESTful Web services.

- Is the service stateless?

The PLM Web service is stateless in the sense that each request to the service does not require the server to obtain any information regarding the client application’s state. The requests made to server are independent of one another and no data is persisted on the server.

- Is the service scalable?

Although the PLM service may not be efficient in the sense that extra data is sent such as HTTP headers and XML mark-up, the service is scalable because it is stateless and does not allow conventional state information to be stored; therefore the service is easier to scale.

- Is the service loosely coupled?

All methods on the PLM Web service are consumed independently of one another, this means that a service is available to a client at any time without requiring data from other methods or services.

- Does the service use a standard interface?

The interface for the Web service consists of the standard HTTP protocol methods. A wide range of information regarding the methods available on the service is not required in order to use them. The methods that are available on the Web service are; GET, POST and DELETE.

6.3.3 Summary

The Service-based apparel PLM system meets all but one of the functional and non-functional requirements to some extent; therefore it can be said that the solution has achieved its aims to a fairly good extent. The browser-based client application requirement
that was not achieved did not detract from the system as a whole; as an alternative application type was decided and agreed upon with the external company Manager.

6.4 Comparison with Traditional PLM Systems

6.4.1 Introduction

As the project has indentified problems with traditional apparel PLM Systems and attempted to solve those problems with a solution, it would be beneficial to evaluate the solution against existing PLM Systems. Ideally a side-by-side comparison of one or more traditional PLM systems against the solution developed for this project would be performed. However, PLM Systems are large implementations that form the core of the organisation in which they are implemented and therefore are unavailable for this type of comparison.

As an alternative, the decision has been made to evaluate the solution with user’s who can be considered experts in the domain of Apparel Product Lifecycle Management Systems. Feedback from this user base should allow a useful comparison to be made to existing systems.

6.4.2 Method

6.4.2.1 Participants

The participants were made up of three males and one female. They are all current employees of the external company and assist in delivering consultation services to organisations that are looking to implement a new PLM system or enhance their existing PLM system. Each of the participants have spent a period of time (two years or more) using PLM systems on a daily basis and some of the participants have experience of developing and testing apparel PLM systems. As the participants can be considered experts in apparel PLM systems; there was a possibility that the solution could be too simplistic and unrecognisable in comparison to existing PLM systems. However, the participants were fully
aware of the overall project and of the goals of this project; therefore the decision was made to go ahead with the evaluation.

6.4.2.2 Materials

The participants were provided with a machine that mirrored the setup of the development environment described in the previous chapter. The PLM Web service was running on the ‘localhost’ server and the example Client application was running. The participants were also provided with an instruction sheet which can be seen in Appendix F.

6.4.2.3 Procedure

A ‘within participant’ design was used for the evaluation; each of the participants performed the same operations. The participants were asked to sit down at the machine in turn and follow the instructions on the sheet provided. In between turns the item that was inserted into the PLM system when following the instructions was manually deleted. After all the participants had completed the instructions a discussion was held on the following points:

- How the types of services available in the system compare to those available in a basic Product Data Management System.
- Apart from more services, what major features are found in existing PLM systems that are not available in this solution?
- Are there any issues that may arise during further development of the solution that would not trouble the development of existing systems?

6.4.2.4 Feedback & Discussion

All the participants identified that existing basic PDM systems provide more services than the basic operations available in the solution. The most common of these discussed was a method to return styles that required action immediately. Although a method was available to count styles requiring approval; one participant stated that this functionality should then lead into returning exactly which styles require approval. Another common service of a PDM
system that was identified as missing from the solution was the ability to edit the data schema of a product. The participants indicated that this was a relatively new feature for PDM’s and the ability to add extra fields to a product ‘on-the-fly’ is a big selling point for existing PDM’s.

The participants agreed that the major feature missing from the basic PLM solution was user access levels. It was stated that although the functionality that existed on the service would be useful to various user types of existing PLM systems the system was missing the ability to control which methods are available to certain users. For example; a designer shouldn’t be able to approve their own design as that responsibility lies with the Manager. Another feature missing from the solution when compared to existing PLM systems was the lack of various data types. One participant explained that for user’s based in the design phase of the product lifecycle; visual representations of data were important. The lack of thumbnails in the solution was identified as the main data type missing when compared to existing systems.

All the participants with a technical background agreed that the main issue that could arise during further development was security. It was stated that potentially storing two rival organisations’ product data could prevent organisations from using the Service-oriented apparel PLM system. As the data is so valuable to an organisation they would be reluctant to let it out of their control; therefore it was stated that security needs to be a top priority in any further development of the system.

6.4.2.5 Conclusion

The feedback obtained from the expert user’s demonstrated that although the basic PLM system is useful to some extent as a service-based PDM system, the services require many more development iterations before they can be used in place of existing PLM systems. The basic functionality was identified as a good foundation for further development of the service but would require a way to restrict certain methods to specific users. Finally, the issue of security was emphasised and it was clear that this would have to be central to any further development of the Service-based apparel PLM system.
6.5 Project Evaluation

The chosen methodology of implementing an iterative style of development enabled a prototype Web service and client application to be produced in a short period of time and therefore can be said to have been a good decision. The method enabled feedback to be obtained from the external company after the first iteration of the Web service was complete and a change was implemented in the second iteration based on this feedback. The iterative style enabled development to be efficient as it allowed the focus of the first iteration to be solely on the implementation of the Web service’s core methods. The second iteration was then able to be used for refining the service and implementing desirable requirements before moving on to development of the client application.

The revised schedule and milestones ensured that the project could be checked to determine if it was on schedule. The schedule was followed fairly well throughout; development of the client application encountered a problem which resulted in the implementation exceeding its allotted time but the previously scheduled ‘run-over’ time was used effectively to prevent a knock-effect on subsequent milestones. It could be said however, that if an extra week had been assigned to the ‘run-over’ time; the issue that was preventing the client application from consuming the DELETE method could have been resolved.

The minimum requirements set out in the introduction were all met; A business case was produced that identified problems with current apparel PLM systems and proposed a Service-based solution, a design of the Web service was produced that adhered to the principles of a Service-oriented Architecture, a basic PLM system was produced consisting of a Web service and client application and an evaluation was produced to discuss the issue of steering PLM system development towards a Service-oriented Architecture.

Two of the possible extensions were met; the Web service had the ability to use JSON in addition to XML for data exchange and a JDNI connection pool was used to manage efficient database connections. As time was limited; the remaining possible extensions were not met. It is fair to say that these possible extensions were too ambitious in hindsight and could possibly warrant complete projects of their own where they would be an enhancement of an existing Service-based PLM system.
6.6 Further Work

From the outset; this project was intended to be continued after the final year project. The experiences gained from this project have provided potential for further work.

The PLM system could be enhanced to take into consideration the access permissions of various user types. For example a manufacturer should only be authorised to see technical data about a product.

Another enhancement that could be made to the system is to include other data types such as JPEG thumbnails or Adobe Illustrator files; both of which are commonplace in the apparel industry.

Due to the issues that arose during development of the client application, it could be re-implemented using server-side technologies that would be free from the cross-domain issues.

In the short term; further work will include improving the quality of services provided in the system and so will produce several more iterations of the PLM Web service.
7 Bibliography


Appendix A: Personal Reflection

The final year project was a very different experience from the normal agenda of attending lectures and tutorials. Working on a piece of work that is unique to you requires a lot of self-discipline and commitment. Whereas a timetable shared by peers and weekly coursework deadlines provided the encouragement to work throughout previous semesters; the final year project requires you to be extremely self-motivated.

As my final year project was an external project I believe my experience was in some ways very different to other students. In the early days of the project I was expected, and rightly so, to have a good understanding of the motivations behind the project. Whereas ideas put forward by experienced members of the department automatically carry the credibility required to be the subject of a final year report; a self-proposed external project does not and requires extra work by the student. I am very thankful that my project supervisor challenged the motivations of my proposal early on in the process so that I was able to clarify the motivations behind the project and justify its credibility.

I believe a valuable lesson can be learnt from this project regarding the literature search. Although the library resources for Computing are an invaluable tool when searching for literature; extremely useful information can be discovered by searching within different subject areas. An example from this project was a book that I referenced several times and one that has been in frequent use since the beginning of the project. My advice to any future students would be to think outside the subject of ‘Computing’ to find that crucial piece of the puzzle you are missing.

My project was undertaken for an external company where I had previously been on placement. While on placement I discovered a real interest in the technology that is used throughout the fashion industry and this was a driving factor behind my motivations for wanting to undertake this external project. My advice to future students would be to try and choose a project that genuinely interests you so that if at any time developing or writing becomes monotonous; your interest in the subject motivates you to continue working to a high standard.

The most useful source of guidance in my project was definitely the meetings with my supervisor. In addition to these meetings the presentation, feedback from the mid-project report and the progress meeting were extremely useful. My advice to future students would be to take advantage of these additional sessions by preparing a ‘next steps’ plan.
Occasionally, after these events had taken place I remembered a question that I had meant to ask which could have been answered thoroughly at the time. If I had taken a short term plan for the next Week to the meeting I could have discussed my schedule and the issue that with hindsight I should of asked.

To conclude; I would advise any future students to treat the project as a challenge, not a chore. If you are focused on the goals of the project the experience will be enjoyable and enriching. In additions to the lessons I learnt above, I gained a wealth of technical knowledge and career skills that will assist me in my future career.
Appendix B: Materials used in the solution

Below is a screenshot of the example data that was supplied by the external company to populate the database. This data is owned by the external company and I had permission to use this.

<table>
<thead>
<tr>
<th>Gender</th>
<th>FileName</th>
<th>CatName</th>
<th>TypeName</th>
<th>VarName</th>
<th>Data</th>
<th>FieldName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_001</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long Fly Front Coat</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_001</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long Fly Front Coat with cuff detail and back vent</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_002</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Fly front, back vent, adjustable cuff</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_002</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long Single Breasted Coat</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_002</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long coat with chest pockets and stand collar</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_002</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Single breasted, Patch pockets, flaps, epaulettes,</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_003</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long zip fronted coat</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_003</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long zip fronted coat with low patch pockets</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_003</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Patch pockets, welt pockets, should guards, back y</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_004</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Full length shirt front coat</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_004</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Full length shirt front coat with pocket flaps</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_004</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Shirt placket, pocket flaps, jett pockets, front d</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_005</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Classic double pocket Epsom</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_005</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Collar and revere, single breasted, back vent, fro</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_006</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Double breasted trench coat</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_006</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Double breasted trenchcoat with stand collar.</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_007</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Double breasted Hunter Coat with belt detail</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_007</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Single breasted Hunter Coat with belt detail</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_007</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Single breasted, shoulder flaps, epaulettes, beltc</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_008</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long single breasted epson</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_008</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long single breasted epson with panel seams and ba</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_008</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Collar &amp; revere, welt pockets, centre back vent, t</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_009</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long parka</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_009</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long parka with squared hem and back vent</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_009</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Hood, epaulettes, front placket, six snap fastener</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_010</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long duffle coat</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_010</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long duffle coat with hood and classic toggle fast</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_010</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Hood, welt pockets, chest gun patches, toggle fast</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_011</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long fly fronted coat</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_011</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long fly fronted coat with stand collar and seam d</td>
<td>Long Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_011</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Stand collar, chest patches, front panel seams, cu</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_012</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long double breasted mac</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_012</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long double breasted mac with back belt detail</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_012</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Collar &amp; revere, eight button fastening, welt pocke</td>
<td>Keywords</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_013</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long four pocket parka</td>
<td>Short Desc</td>
</tr>
<tr>
<td>Menswear</td>
<td>M_COAT_CAS_013</td>
<td>Coats</td>
<td>Casual</td>
<td>None</td>
<td>Long four pocket parka with waist belt detail</td>
<td>Long Desc</td>
</tr>
</tbody>
</table>

No other materials were used that were not developed as part of the project.
Appendix C: How ethical issues were addressed

All individuals that were involved in this project other than myself were employees of the external company 'PDP Limited' so there were no ethical issues which arose.
# Appendix D: Summary of Functionality

<table>
<thead>
<tr>
<th>Description</th>
<th>Who</th>
<th>Basic Operation</th>
<th>How PLM is Used</th>
<th>Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buyer visits fashion shows, retail outlets etc. to gain design inspiration</td>
<td>Buyer</td>
<td>Insert style into PLM</td>
<td>Buyer will use the system to insert brief, this includes uploading photos and text. A status of &quot;pre-concept&quot; or similar will be assigned.</td>
<td>Design Brief - text, description of the customer, where the product will be sold, target price, product details like Menswear, shift, short sleeve etc. Photos/Storyboard</td>
</tr>
<tr>
<td>Designer creates illustrations of garments in Adobe Illustrator adhering to design brief</td>
<td>Fashion Designer</td>
<td>Search the PLM for relevant style design brief. Insert Sketch to the PLM</td>
<td>Designer will browse PLM for assigned design brief. Adobe Illustrator used to draw the garment, then upload/link to PLM with an &quot;awaiting approval&quot; status or similar.</td>
<td>Illustrations - Adobe Illustrator files</td>
</tr>
<tr>
<td>Manager provides feedback on designs and selects final designs</td>
<td>Manager</td>
<td>Search the PLM for designs requiring approval. Count items requiring approval, insert comments, update product status.</td>
<td>Manager will view designs, &quot;awaiting approval&quot; and provide comments to change product status to &quot;approved&quot; or similar.</td>
<td>Final Illustrations - Adobe Illustrator files</td>
</tr>
<tr>
<td>Merchandiser selects fabrics and textures for the designs taking into account market trend, costs etc.</td>
<td>Merchandiser</td>
<td>Search the PLM for designs requiring style details. Retrieve all products for collection book. Create/insert collection book. Insert Style details</td>
<td>Merchandiser will view &quot;approved&quot; designs and insert style details.</td>
<td>Collection Book - Designs and material info</td>
</tr>
<tr>
<td>Event</td>
<td>Responsible Party</td>
<td>Action</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------</td>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Technicians produce a technical specification, often using CAD software.</td>
<td>Garment Technicians</td>
<td>Technical Spec – Tech pack contains detailed sketch with washcare label, seam types etc. Measuring instructions and points of measure.</td>
<td>Use PLM to create Tech pack. User will search for POM's/washcare labels/seam types already stored in the system to add to tech spec of a product.</td>
<td>Search PLM for the product and POM's etc. Update product with Technical Spec.</td>
</tr>
<tr>
<td>Manufacturer produces samples.</td>
<td>Manufacturer</td>
<td>Physical Sample of a garment</td>
<td>Manufacturer will log-in to PLM to view the Tech Pack of a product only.</td>
<td>Search PLM for the Tech Pack of a product.</td>
</tr>
<tr>
<td>Manufacturer produces order on Mass.</td>
<td>Manufacturer</td>
<td>Full order</td>
<td>Manufacturer will use PLM to view order details (Qty, due date etc) of a product and again the Tech Spec or remove unwanted orders.</td>
<td>Search PLM for purchase order and Tech pack. Remove unwanted Tech pack.</td>
</tr>
<tr>
<td>Garments are delivered to stores.</td>
<td>Logistics</td>
<td>Invoice for payment</td>
<td>Logistics will use PLM to view delivery details and to invoice the Manufacturer/Design company.</td>
<td>Search PLM for delivery details. Insert invoice to PLM.</td>
</tr>
</tbody>
</table>
Revised Gantt chart:
Appendix F: Participant Instructions

Please follow the instructions below in number order

1. Select ‘View all Styles’ to view a report of all the product data, once you have finished viewing the data close the window.

2. Selecting ‘Items requiring approval; to retrieve a count of the number of styles requiring approval. Select ‘ok’ to close the dialog box.


4. Select ‘Retrieve Style’ to search for a style by ID. Enter the ID: ‘M_COAT_FOR_005’ and click ‘ok’. Check the information received matches the information you previously inserted. Close the window.

Thank you