
Work Term Report 5

COMP 3999A

FOR THE WINTER AND SUMMER 2015 PLACEMENT AT THE
OFFICE OF THE AUDITOR GENERAL OF CANADA

Ankush Varshneya (100853074) AnkushVarshneya@cmail.carleton.ca

Supervisor: Alain Roy (Director of IT Applications)



Ankush Varshneya
415 Cresthaven Drive
Nepean ON, K2G 6Z4

Friday, August 14, 2015

Dr. Christine Laurendeau
School Of Computer Science
Carleton University
Ottawa, Ontario K1S 5B6

Work Term Report 5 COMP 3999A

Dear Madam,

This is my work term report for the 4th and 5th work term and was prepared in connection with course COMP 3999A for my Co-op placement at the Office of the Auditor General of Canada (OAG). I worked as a Developer in the IT Applications team lead by Alain Roy. The first term was mainly spent on the Product Code Management (PCM) Project lead by project manager Erin Larsen, while the second term was mainly spend on the Time Sheet and Reporting System (TRS) Project lead by project manager Erin Larsen.

The Office of the Auditor General of Canada (OAG) is an independent audit office and a world leader in legislative and environmental auditing. They conduct independent audits and studies that provide objective information, advice, and assurance to Parliament, territorial legislatures, boards of crown corporations, government, and Canadians. The IT Application team provides various software tools which helps the office to carry out all necessary tasks.

This report highlights some of the key concepts I learned during my placement as well as my contributions to OAG's products.

During the first term I mainly worked on front-end related tasks which were a continuation from a past work term at OAG. Primarily I was assigned to refactor parts of a web application called Product Code Management (PCM). Following the refactoring task, I was given the task of creating additional maintenance related screens which allow for the administrators of the application to manage internal data and automate notifications from within the PCM application.

During the second term I mainly worked on back-end related tasks. Primarily I was assigned the task of verifying and making changes to the data model of an application called Time Sheet and Reporting System (TRS). Next, I was given the task of creating scripts to help create the tables and preload them with the appropriate data. To complement my learning I was given various small task which in turn helped me learn a lot about the system, hence making it easier to work on my main goal.

This report has been prepared and written by me and has not received any previous academic credit at this or any other institution. I would like to thank Alain Roy (Director of IT Applications) for helping in reviewing this report.

Sincerely,
Ankush Varshneya
Student number 100853074

2 ABSTRACT (EXECUTIVE SUMMARY)

Originally, in a past term, I was hired by the Office of the Auditor General of Canada as a Developer to help implement maintenance related screen for a future release of the PCM Project. I was then rehired for an additional 2 terms back to back. In the first term the PCM project was in a maintenance phase, so I was given the task of refactoring the maintenance related screens I had worked on in a previous term as well as adding new maintenance screens. In the second term, I was assigned on the TRS project where I primarily dealt with back-end related tasks. This report begins with an introduction to the Office of the Auditor General of Canada, and then goes on to document the key concepts I learned about while developing there.

During my first term I mainly worked on front-end related tasks which were a continuation from a past work term at OAG. Primarily I was assigned to refactor parts of a web application called Product Code Management (PCM). Following the refactoring task, I was given the task of creating additional maintenance related screens which allow for administrators of the application to manage internal data and automate notifications from within the PCM application.

During the second part of my work term I mainly worked on back-end related tasks. Primarily I was assigned the task of verifying and making changes to the data model of an application called Time Sheet and Reporting System (TRS). Next, I was given the task of creating scripts to help create the tables and preload them with the appropriate data. To complement my learning I was given various small task which in turn helped me learn a lot about the system, hence making it easier to work on my main goal.

In addition to working on the PCM and TRS projects, this document also includes details about some of the other tasks I was assigned. I was assigned the task of doing quality assurance (QA) on another proprietary application at OAG.

This report also highlights my work experience, day to day activities, some of the tools I used, how I applied my academic knowledge in my position and further my knowledge by getting work experience, different type of problems faced at the work, and the type of approaches I used to tackle those problems.

Lastly this document elaborates on how I related my work experience to my academic studies, my accomplishments, and any abbreviations used in my work term report.

3 ACKNOWLEDGMENTS

First and foremost, I would like to thank Bill Moeller, the software architect, for helping me understand my tasks and for helping me understand the underlying technologies used by the Office of the Auditor General of Canada and lastly for guiding me throughout the term.

I would also like to thank Erin Larsen for helping me throughout the term with my primary task and helping me understand a bit more about the system as well as some of the other things I needed to know about OAG.

I would like to thank Joe Tacconelli for helping me with some of the problems I encountered which were related to the front end portion of the applications.

I would like to thank Branislav Mogin for helping me with some of the problems I encountered which were related to the back end portion of the applications.

I would like to thank Louise Richard, Lise McGillis, and Julie McCoy for helping me understand the back end of an existing application which intern helped me with my second term.

I would also like to thank Eugenio Lo for helping me with some of my tasks.

Lastly I would like to thank the business users, Alain Roy, Patrick Dumoulin, Erin Larsen and other members of her team for helping me review and influencing my mock-ups and design idea as well as providing me with the needed requirements.

4 TABLE OF CONTENTS

1	Title Page.....	1
2	Abstract (Executive Summary).....	3
3	Acknowledgments.....	4
4	Table of Contents	5
5	Introduction	6
6	Work Experiences	7
6.1	Tools	7
6.2	Objectives and Contributions.....	9
6.2.1.1	Product Code Management Project.....	9
6.2.1.2	Refactoring Maintenance Screens.....	9
6.2.1.3	Adding Additional Maintenance Screens	10
6.2.2	Timesheet Reporting System Project	10
6.2.2.1	Data Modeling	11
6.2.2.2	Scripting the tables and preloading Data	11
6.2.2.3	Other TRS related tasks	11
6.2.3	Other Tasks.....	12
7	Reflection	13
7.1	Challenges and Solutions.....	13
7.2	Relation to Academic Studies and Career Development	14
8	Abbreviations	15

5 INTRODUCTION

The Office of the Auditor general of Canada is an independent audit office with head office in Ottawa and four regional offices in Vancouver, Edmonton, Montréal and Halifax. The office is headed by the Auditor General of Canada, Michael Ferguson, who audits federal government departments and agencies, most Crown corporations, and many other federal organizations, and reports directly to Parliament. The Auditor General of Canada is also the auditor for the governments of Nunavut, the Yukon, and the Northwest Territories, and reports directly to their legislative assemblies.

Being a company focused on audits, the OAG is naturally keen on knowing as to how its own resources are being spent. To really be organized in this aspect, OAG uses an internal software called MIS2000 where employees use time codes as a means to show what activity and project they are working on. For example a time code may be in the format XXXXXX-YY where XXXXXX is a code for the Product and YY is a project. So If I were working on developing I would be using one code and if I were in meetings I would be using another code to time myself and so on. This approach really helps the company to better track its resources, as well it helps one to be able to see what they had spent time on and reflect accordingly.

In the IT application department, employees are usually assigned to a project for an application, for example I was assigned to the Product Cost Managements project and then the Timesheet reporting system application. Each project consists of a project leader and other employees such as developers. Everyone in the project team reports to the project leader on progress related to the project. An employee is usually swapped in between projects as needed by project leaders. Every employee in the IT application department also reports to the director (DX), Alain Roy. All the DXs of various departments throughout the company report to the Principal (PX), the PX of the IT department is Jean-Charles Parisé. The PXs report to an Assistant Auditor General (AAG), The AAG for IT is Sylvain Ricard. All the AAGs report to the Auditor general of Canada, Michael Ferguson.

The IT application department makes and maintains applications for the entire office to help in various tasks such as audits. The Product Cost Management application (PCM) is a web application, designed to replace components of an existing legacy terminal based application called MIS2000. PCM is used to track projects and their budgets which are organized by product Lines, Product Categories, and Topics. The Timesheet reporting system (TRS) is a web application, also designed to replace components of an existing legacy terminal based application called MIS2000. TRS is used to track time of employees spent based on a Product/Project combination.

6 WORK EXPERIENCES

OAG follows agile project management methodologies, as a result many project teams consisting of cross-functional teams of the project leader, developers and key business users meet on a daily basis in stand-up SCRUM meetings. These meetings are arranged by the project manager, where the developers discuss the progress of the project as a whole with the business, as well as gather the users' feedback. Developers also discuss their progress with the project leader and may also use this time to get opinion and help from their fellow developers. Additionally, new feature implementations and improvements are also discussed here. These meetings are a great opportunity to find out what's going on around you, what your fellow colleagues are working on, it gives you a great understanding of the project and it's a great opportunity to follow up on any questions you may have. Since developers are sometimes switched around from project to project, these meetings are a great place to find out about what's going on in the project you may have just been assigned to.

Moreover, the project manager also arranges a meeting at the end of a sprint, every 2 weeks, where senior management, clients, developers and other interested parties meet to get updates on the project. This is less of a discussion group and more meant as an information session to get a broad level overview of the project and address any key concerns of the clients.

In addition to meetings organized by the project leader, a developer may also organize meetings with the key parties to discuss requirements and concerns related to the component of the project they may be working on. This is what I did at times to discuss information related to my task and gather feedback throughout the term so that I could provide my level best output to the projects I was assigned.

Lastly, an optional biweekly stand-up is scheduled by the director of IT application department, which is exclusive to all developers managed by that director. This is where developers from different projects discuss their progress and issues with the director and with each other. This is a key place to find out what is happening in other projects and the entire development team as a whole.

6.1 TOOLS

At the Office of the Auditor General of Canada, they always use the latest and greatest technologies which means that projects that originated at different time periods may use different technologies that were current to the time they were instantiated. The Product Cost Management application involves heavy use of .NET VB programming, while the Timesheet Reporting System application involves heavy use of .NET C# programming; both projects use ASP.NET MVC architecture and entity framework ORM. Its IDE consists of Microsoft Visual Studio and it uses Microsoft SQL Server as a back-end store, which is managed through the Microsoft SQL Server Management Studio. IIS, also referred to as Internet Information Services, is used as a web server to deploy the application.

Microsoft Team Foundation Server (TFS) is another key component used by the application. It is used as a means of version control and as a product tracker. TFS is also used for compiling builds and deploying them.

From a front-end prospective HTML5 is used to display the UI and jQuery/JavaScript are used for client side scripting. Twitter bootstrap and CSS3 are used for styling the web application. In contrast to previous work terms I found the front – end to be really flexible and easy to use, it was easy to add virtually any jQuery plug-ins to easily enhance the functionality. I utilized the DataTable plug-in throughout the maintenance related screens to address a lot of the requirements requested by the business users for the user interphase.

From a back-end prospective, the application uses a Service Oriented Architecture (SOA) and Windows Communication Foundation framework (WCF), which is used for building a service oriented application in a secure manner. .Net Domain object and .Net Services in combination with the entity framework are used to communicate with the back-end database. Furthermore, each Project contains a database subsystem which consists of SQL code to generate and prepopulate the tables with the necessary data.

.Net MVC (Model View Control) are the architecture used to bind the front-end to the back-end. First, the entity framework uses the DATA Access Layer (DAL) to query the database with in the Business Layer (BL) which produced domain objects out of the data it is given. The .Net Services are used to send the domain objects to the front-end, through service call methods. The convention is that the Controller and Model, or viewModel as its more commonly called, part of “MVC” makes calls to the service. The ViewModel is created from using data from the domain objects. The methods of the ViewModel manipulate the inner Domain object based on user events. Finally the domain object is sent back to the server through a service to be persisted. Using HTML Helpers, a tool included in .NET MVC, the View can bind the UI components to the properties of the Model; so if a UI component is used to get user input then that input will get sent to the ViewModel which will change the Domain Objects. The controller part of MVC is used to route web pages to the correct view as well as to instantiate the model with the correct settings needed by the view. The URL convention used by the PCM application is as follows: www.NOTAREALWEBSITE.com/LANGUAGE/NAME_OF_CONTROLLER/NAME_OF_VIEW. The name of the controller is used to route to the correct controller, the name of the view is used to tell the controller that view it needs to route to and what model to execute. The language is used as an internationalization mechanism, it tells the controller what language to render the views in.

From talking to other students and fellow developers, other newer projects use a similar layout to the one mentioned above with the exception that all other projects, except PCM, use .NET C# instead of .NET VB.

6.2 OBJECTIVES AND CONTRIBUTIONS

6.2.1.1 *Product Code Management Project*

In a previous term at the Office of the Auditor General of Canada I had been hired to implement maintenance related screens which allowed the administrators of the PCM application to easily maintain data, related to the application within the application itself.

For the duration of the first term, the PCM Application Project was in the maintenance phase. This phase involved fixing software and UI related defects, and refactoring the application to be more consistent with newer guide lines set by the IT application department as well as addressing any new features requested by the business users.

6.2.1.2 *Refactoring Maintenance Screens*

Initially, I was assigned the responsibility of refactoring maintenance related screens for administrators to manage the PCM application as well as refactoring other screens which made use of the data managed by the maintenance screens. The refactoring phase consisted of 3 sub phases, fixing software and UI defects, changing sub components of the application to be more consistent with other parts of the application, and addressing any feature changes requested by the business users.

The application follows a client server like architecture, traditionally the business logic was present in both the server and client subsystems of the application. This meant that input was being validated partially on the client and partially on the server. As part of the refactoring phase the business logic and input validation code were to be fully moved to the server sub system as part of the Domain object. Furthermore, any objects related to business logic were fully removed from the client subsystem. This left the client subsystem with the view objects, controller objects, and viewModel Objects. The view objects contained HTML code, JavaScript code, and CSS code. Controller objects only contained logic to redirect to the appropriate view. And lastly, the viewModel object were used as wrapper objects to contain and manipulate data as needed by the view objects. As mentioned previously, this refactoring was applied, by me, to the maintenance screens as well as the screens that used data from the maintenance screens, and other screens that had previously not been refactored.

In parallel to the refactoring phase the other task I was assigned, was to address software and UI defects and feature updates. One of the software defects was to do with the fact that certain data created from within the maintenance screen was not detected by the find component of the application. With the help of a fellow co-worker, I was eventually able to find out that this was largely due to one of the columns in an SQL view being hardcoded. I was able to communicate this issue with a fellow back end developer, and together we added the functionality to have this column be generated from user input rather than hard coding them.

Promptly after fixing the software defects, I started fixing UI defects. This involved frequent consultations with the Usability Architect at the office. This mainly involved cleaning up the UI, making it consistent throughout the web application, and eliminating unnecessary white space without sacrificing

usability. In addition to this I also altered the UI to allow for additional features requested by the business users.

6.2.1.3 Adding Additional Maintenance Screens

Next, I was given the task of implementing 2 new maintenance screen as requested by the business users. This task was very similar to what was done in a past work term at OAG.

One of the screens had to do with the ability to maintain the fiscal year table in the backend. This task involved making screens to allow an administrator to close or freeze an existing fiscal year as well as to add future fiscal years.

The other screen had to do with adding functionality to the application, to automate the sending of emails to interested parties from within the PCM application, if a pre specified event occurs. This has done through Simple Mail Transfer Protocol (SMTP) and a helper class with in .NET.

For each of the tasks I first analysed the mock up and requirements and made a list of what functionality will be needed. For example, both of the screens required a way to retrieve data, create data and amend pre-existing data. After the list was made, I begun making the needed API's to full fill the listed functionality. The first screen evolved querying the database for the correct information and as well as persisting information to the database. The other screen was a bit trickier to develop, as it required the use of SMTP as well as a mechanism to trigger the emails.

Furthermore, then I started to piece together the MVC part of the screens. First, I developed the ViewModel as well as the domain part; this is where I made service calls to the back-end to get the needed properties and implement the needed methods as part of the model. This involved reverse engineering the mock-ups to see what data I needed and to organize it accordingly.

Additionally, I created the View part, based on the ViewModel and mock-ups. This is where the majority of my development effort went into. Initially I used the view as a high fidelity prototype to show off to the business; I would organize review meetings and use my view to solicit feedback from users on a couple of occasions. The view was highly based on HTML5 programming and I used CSS3 to easily style the view based on user feedback.

Finally, I developed the controller which as mentioned in the tools section, is used to route web pages to the correct view as well as to instantiate the model with the correct settings needed by the view. Thus, this gave me somewhat of a finished product.

6.2.2 Timesheet Reporting System Project

During the second part of my work term I was assigned to work on the called Time Sheet and Reporting System (TRS) application which was a remake of a sub component of an existing legacy application called MIS2000. This phase evolved verifying and making changes to the data model of the new TRS application as well as creating scripts to help create the tables and preload them with the

appropriate data; both tasks were done in parallel of one another; as of writing this report both tasks are still on going.

6.2.2.1 Data Modeling

For the TRS project I was assigned the task of verifying the data model as well as making changes to it. This task involved ongoing talks with both the Business as well as the remaining developers of the legacy application this application was meant to replace.

Primary, I analysed the Data schema of the old application to get an understanding of the older application. This evolved analyzing the existing tables, their attributes, and their relationship. During this task I had to frequently ask the developers of the old application the purpose behind the way the old schema was made; this gave me a lot of domain knowledge about the application which intern helped me to better do my task.

Following this I analyzed a draft of the data model of the new TRS application and noted any discrepancies. These discrepancies were then fixed to make a more accurate data model. For this part of the task I had to have frequent meetings with the developers of the old application as well as the developers of the new application. This helped to better understand the goals of the new application.

As of writing this report this task is ongoing as the team has been discovering what needs to move to the new system and what needs to stay in to old system in terms of the data.

6.2.2.2 Scripting the tables and preloading Data

In parallel to helping make the schema I was also assigned the task of making the scripts for the tables as well as preloading them with the appropriate data. This task was pretty straight forward in that it involved making scripts based on the schema.

As of writing this report this task is also ongoing as the team have been discovering what needs to move to the new system and what needs to stay in to old system in terms of the data.

6.2.2.3 Other TRS related tasks

To help me get a better understanding of the system I have been assigned various smaller tasks such as helping develop various calls within the TRS API from a back-end prospective. Initially there was a potential for performance hits with the API I produced, using my analytical skills as well as help from a fellow developer I was able to amend the issue.

6.2.3 Other Tasks

In parallel to working on the PCM and TRS projects, I was assigned the task of doing quality assurance (QA) on another proprietary application at OAG. This task involved comparing an older version of the application to a newer version and making sure all the requirements were still being met.

7 REFLECTION

Reflecting back on the term, it was a fantastic learning experience for me. I learned about a formal software cycle that is present at the OAG. Attending weekly stand-up SCRUM meetings helped me with getting the bigger picture of how the PCM project and TRS project work and how the IT Application department functions at large. It also taught me about how agile project management is used at a company such as OAG.

I learned a great deal about the .NET Visual Basics and C# Platform, how a .NET application works and how it is structured as well as how everything is layered from the database all the way up to the UI presented to the user.

Working with the front-end has taught me how .NET MVC can be used to build fluid applications as well as how it can be connected to the back-end through service calls. I really learned a lot about HTML5, CSS3 and jQuery and all the plug-ins that it can use to easily make a flexible UI which would have made for a daunting task if it was implemented in some of the traditional applications I worked on in a previous work term. Working on the maintenance screens gave me an experience with developing a component from scratch; in contrast to a previous work term, where I was only adding on top of a component rather than developing it completely from scratch.

Working with the back-end has taught me about the effort needed to make a schema and what type of software is needed to make them.

In addition to some of the debugging tools found in Microsoft Visual Studio, I also learned about tools such as Microsoft SQL Server, which can be used to interact with the database outside of the actual application, as well as the developer tools in Internet Explorer which are used for client side debugging directly in the browser.

7.1 CHALLENGES AND SOLUTIONS

While working on my tasks, understanding the technologies used was somewhat of a challenge. With the help of Bill Moeller, I was able to get a good idea of how the technologies work together. With the help of online tutorials and looking at some of the other parts of the project I was able to get a good understanding of how to go forward with my task.

Looking at examples throughout the application involved a large variety of code written by many writers. Looking at such a variety of code helped me explore a variety of coding styles and practices, which in turn influenced my own style by a great degree. Sometimes certain code segments lead me to interact with other developers, which allowed me to explore how other developers contribute to the OAG products and hence helped me greatly increase my productivity and knowledge.

Looking at some of the requirements and trying to implement UI features using JavaScript/jQuery was very tough at times. For example, on one of the requirements was, to be able to sort columns of a HTML table, I tried implementing it on my own only to run into a dead end. Finally a

fellow developer, Joe Tacconelli, suggested that I use the Datatable Plug-in. Using the plug-in, not only helped me to implement the user requirement but I was also able to use some of my own extensions and others available in the plug-in to have the UI on my maintenance screens be very intuitive and to the users liking.

Furthermore, understanding the complex schema of an existing application and comparing it to a new applications schema was very tough. With the help of fellow back-end developers, Branislav Mogin Louise Richard, and Lise McGillis I was able to better understand the schema and make suggestions to improve the design of the schema for the new TRS application.

7.2 RELATION TO ACADEMIC STUDIES AND CAREER DEVELOPMENT

This placement helps me enhance the learning I received at university. It helped me see some of the software engineering concepts such as agile project management in action in real life projects

During the Requirement and Design Phase, I found myself using some of the skills I acquired through a Human-Computer-Interaction class I took at the university. Through meetings with the business I was able to conduct user studies which I learned about in class. I made medium and high fidelity prototypes through Balsamiq Mock-ups, as well as the View of my application. This helped me see how concepts I learned about in class are implemented in real life which complements my learning received in university.

During the development phase, I found myself using a lot of concepts I learned about in a web application class at university. For example, when working with jQuery some variable was not being set when it should have been; I remembered learning about closures in java script and using that logic, I understood the flow of my jQuery and understood where my error was coming from. Additionally, I used the basic structure I learned about websites in the web application course and was able to apply it to my work, thus I was easily able to understand as to how the tools work as described in the tools section above.

During the Data modeling and scripting phase, I found myself using a lot of concepts and skills I acquired through a data base class at university. For example I was able to understand the diagrams of the schema based on what I learned about ER diagrams in class. I was able to know what type of relationship patterns are needed for certain entities.

Solving defects on my own screens as well as comparing the schemas and scripting the database touched upon various other components of the application, which really helped me in learning trouble shooting tips, which are easier to learn in a work scenario. In contrast, the courses I have been through in university required you to write your own code with the exception of 1 or 2 courses, where there is very little code written by the professor or teaching assistants, which doesn't offer as large a variety.

While doing group work for courses, the work ends up being divided unequally at times. For example in a group, one member may have expertise in mock-ups and user requirement gathering, hence they may end up doing a good chunk of that work while another member focuses on another aspect. Here at OAG, I was able to get a good experience with each of the aspects. For example in past

courses I did not do much requirement gathering or product reviews with users as they were done by other members of my group. So I lacked those skills even though they may have been taught at a conceptual level in the course. Since I got a lot of exposure to it while working at OAG it really helped me to go back and piece together the conceptual part I learned in class to the practical skills I acquired in this work term. When I first began going in to the development phase of my term it was really overwhelming at times, as I was dealing with such a large system based on a language that was foreign to me. But with some help from fellow developers, looking at examples, and connecting skills acquired from past work terms and classes, I was quickly able to understand the technologies used on the project. Towards the end of the development phase I felt very comfortable and confident working on the project.

In conclusion, I was able to make use of some of the concepts and techniques I acquired from my studies to be able to better adapt myself to the work environment. After gaining the confidence I am now able to better trust myself even when I work on different areas of the product. Working at OAG has given me a great experience as well as a new appreciation for this type of career.

8 ABBREVIATIONS

IT – Information Technology
API – Application Program Interphase
ORM –Object Relational Mapping
UI – User Interface
PCM – Product Code Management
TRS – Time Sheet and Reporting System
OAG – The Office of Auditor General of Canada
DX – Director
PX – Principal
AAG – Assistant Auditor General
VB – Visual Basics
.NET – a frame work developed by Microsoft
ASP – Active Server Pages
IDE – Integrated Development Environment
SQL – Structured Query Language
IIS – Internet Information Services
TFS – Team Foundation Server
HTML – Hyper Text Markup Language
CSS – Cascading Style Sheets
SOA – Service Oriented Architecture
WCF – Windows Communication Foundation
MVC – Model View Control
URL – Uniform Resource Locator
BL – Business Logic
DAL – Data Access Layer
LINQ – Language integrated Query
LMS – Learning Management System
MIS2000 – A legacy application used at the Office of Auditor General of Canada