

Final IT Report Template

Complete this document and submit as a file (Word doc or PDF) in the **Final IT Report Submission** Assignment. Part C should be copied and pasted into the Text box of the same **Final IT Report submission** Assignment. Note that your work may be checked for originality.

Please see the **Final IT Report Instructions & Marking Rubric for details** including Engineers Australia's Stage 1 Competencies and Elements of Competency and the Marking Rubric.

Name:	Abanob Tawfik
zID:	Z5075490
School:	School of Computer Science and Engineering

Part A: Description of Placements

- Provide the following information per placement:
 - Name of each company or institution you were placed with and the title of your position
 - The date and duration of each placements
 - A brief description of the company's or institution's core business or function
 - A brief description of your role and responsibilities
- Refer to the marking rubric for Part A: Description of Placement
- Maximum 300 words per placement

During my placement I worked at Maia Financial and Credabl as a software engineer intern. I began work on the 3rd of June and finished on the 30th of August. Maia Financial and Credabl are both specialist finance companies owned by a parent company Monash. Maia Financial is focused on asset financing for businesses. They provide loans to businesses that want to acquire assets for a fixed period. Credabl is focused on lending for medical, dental and veterinary professionals. During my placement my role was focused on improving technology infrastructure, developing API integrations, re-designing existing services to provide a better user experience and developing new internal technology. I was responsible for communicating between different departments to gather specific requirements, maintaining documentation for code design and interaction, analyzing existing services for potential improvement, determining the feasibility of new products and implementing solutions from design into clear code that can be understood by my fellow co-workers.

Part B: Indicators of Attainment

- a) From the Engineers Australia Stage 1 Competencies and Elements of Competency document, select 3* or more indicators of attainment listed under: -
 - Table 1 Knowledge and Skill base (at least 1)
 - Table 2 Engineering Application Ability (at least 1)
- b) Provide specific examples of how the task or role undertaken by you during your placement exemplifies each indicator of attainment
- c) Refer to the marking rubric for Part B: Indicators of attainment
- d) Maximum 300 words per indicator of attainment

Table 1 Knowledge and Skill base

Engages with the engineering discipline at a phenomenological level, applying sciences and engineering fundamentals to systematic investigation, interpretation, analysis and innovative solution of complex problems and broader aspects of engineering practice.

During my placement I was able to gain a comprehensive theory-based understanding of the underlying science and engineering fundamentals of a software engineer. Throughout the entire placement I was given programming tasks that I had no prior experience with in a new technological environment. I had used the skills I learnt throughout my studies to carry out these tasks in a timely and efficient manner. I had first defined the problem at hand into a set of requirements. Sometimes this required gathering requirements from non-technical people, in one case I had to design a balance sheet from data and did not know how the accountants wanted their end product, so I explained to the lead in their team in simplistic terms the data that we have available and together we molded this into a mockup of a template. After these requirements were gathered, I began documenting and creating uml diagrams to represent the data structures required. My supervisor and I then began refining my proposed ideas in my design documents into a final set of requirements and a final design. Only then did I begin the programming tasks knowing exactly what I had to do. I began applying the fundamentals of data structures and algorithms and object-oriented design to help me turn the proposed design into code that produced the desired output efficiently.

Develops and fluently applies relevant investigation analysis, interpretation, assessment, characterization, prediction, evaluation, modelling, decision making, measurement, evaluation, knowledge management and communication tools and techniques pertinent to the engineering discipline.

I was able to gain a conceptual understanding of the relevant mathematics, numerical analysis, statistics and computer and information sciences involved within software engineering. In one of my tasks I was required to flatten a tree structure into a list of nodes that contained their parent elements. This was to help recreate financial reports from data returned from QuickBooks API. My supervisor first told me to design an elegant solution before I began coding away, and with this he also wanted me to compute the run-time complexity of all algorithms and functions used. I decided to use a recursive approach over an iterative approach as I was not the only person working on this functionality, and most of the team were happier to take a recursive approach for simplicity and readability. We had decided that the data we retrieve is not feasibly large enough to cause stack-overflow errors. My supervisor and I had run through a numerical analysis on the data structure returned and the largest possible size and branching possible. From this we had determined that there was a low risk involved. My team and I used an Azure DevOps repository to collaborate on our code and each person in the team had their tasks defined in the Jira board. During our weekly sprints we would discuss our progress and any issues involved as a team and understand where the team stands.

Proficiently applies advanced technical knowledge and skills in at least one specialist practice domain of the engineering discipline.

Throughout my placement one of my main goals was to learn full-stack web development. This was an area in software engineering that always interested me and allowed for more flexibility and a wider set of attainable skills. I have learnt how to design RESTful API services in ASP.NET to handle HTTP request and response procedure from our backend systems. I've also learnt how to use Angular to design modular and reusable front end, user facing systems that also communicates with a backend to modify the view on screen. Another thing I've learnt was how to use the .NET LINQ library to handle database changes and queries. Part of my experience was learning how the components of the stack interact with

each other. In a specific instance I had to create a button on the front end for an API integration with QuickBooks accounting software. When clicked, this would trigger the OAuth2 procedure to authenticate the user and provide us with a request code, all handled within the front end in Angular. The request code will then be passed to our backend API controller which will acquire an access token using the code and pull the user's balance sheet and profit and loss data. The data was then processed and mapped into a form that could be placed into a database and maintain its hierarchical structure using backend logic my supervisor and I had designed. After this we used the .NET LINQ library to add these reports into our MYSQL database. This would then send a notification to the front end again to inform the user their data was successfully acquired, or an error message if any step failed.

Identifies and applies systematic principles of engineering design relevant to the engineering discipline.

During my placement I had learnt the importance of software design in real world scenarios. After I had finished the API integrations from QuickBooks, Myob and Xero, I was tasked to re-design and re-implement the functionality of one of their web application flow. With the help of my supervisor we had began the process of requirements engineering. We had gathered requirements, documented them and began designing our solution based on this document. Often, other departments would ask for extra features which were unfeasible due to our time restraints so one of my biggest challenges was communicating across what was possible based on our resources and what could not be done. My supervisor and I had then constructed user stories and writing design documentation to plan out how to efficiently fulfill our requirements. I was tasked with mapping how each system would interact with each other, and then draw the code flow and uml designs of my proposed design. Together with the help of my supervisor we were able to refine my design into a more concise form that could capture all the requirements and follow many of the design patterns of software design including, the observer pattern, canonical model, dependency injection, strategy pattern, composite pattern, decorator pattern and factory pattern. We had also made sure our design followed the principles of SOLID. In many cases my boss would make me decouple Angular components to allow more flexibility and re-usability. Throughout each stage of the process my supervisor would always make me take a step back and look at what I'm doing now to see the bigger picture. This valuable life lesson has helped me stop dwelling on the little things and making every tiny feature perfect, and rather achieve the functionality and then refine.

Table 2 Engineering Application Ability

Partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the paramount consideration.

During my placement, all of the front-end work for applications were designed on Angular which embodies the practice of partitioning problems into smaller manageable components and then combining it all in the end to form your final product. This also came with the bonus of components being re-usable and encouraging modular design. In specific the application flow I had to re-design and re-implement was a single page application form that spanned over multiple "pages" with complex mechanisms. The only feasible way to achieve this was to break down the functionality of each "page" into separate components, for example we had a form-input-component which was a reusable component that allowed us to enter form details and bind them to an ngModel. This component was tested, to make sure the behavior is what is expected from both the UI and the internal models. We would then plug this component into any form that required a text input, for example a form with the field "first name". Another example would be during the creation of our in-house document generation service which takes JSON data and a word document template from an API request and will replace the placeholder tags/run the code in the template to create a pdf output document. A very complex problem was the parsing of our template, as we had to parse and process our xml code and run the C# equivalent of what was inside the placeholder. My boss and I had broken the problem down into parsing assignment tags first, then parsing loop tags, then parsing conditional tags. This allowed us to isolate and unit test each section and combine them all in the end to a correct, unit tested and performant document generation tool.

Proficiently applies technical knowledge and open-ended problem-solving skills as well as appropriate tools and resources to design components, elements, systems, plant, facilities and/or processes to satisfy user requirements.

Throughout my placement I had to constantly apply my technical knowledge of software design and infrastructure in order to solve new and dynamic problems. For example, one task given to me was to create a web-service that would automatically send a confirmation email to a user after they have finished their application for a loan. Using the principles of software design, I made sure that the service was flexible and re-usable and designed so that it can be easily integrated into many other projects. One problem I ran into was finding out how to send an email with the images, colors and formatting through my backend code. Using my knowledge of how emails work from networking I had remembered that email content was sent as HTML and plaintext in a HTTP request to the email server. After planning and discussion with my supervisor, we had decided to use SendGrid API to send the emails across due to the requirement to send a large amount of emails. We initially were looking into using Microsoft outlook API, however this was not feasible as the azure gateway configuration limited the amount of emails we could send, whereas SendGrid allows us to send as many as we want on a pay for use subscription. The decision to use SendGrid also allowed us to quickly integrate this core feature as they provided a simple and easy to understand API that required a low amount of resources to get into production.

Contributes to and/or manages complex engineering project activity, as a member and/or as the leader of an engineering team.

During my placement there was major emphasis on teamwork and working within a team of engineers. I was taken to weekly sprints and stand ups to discuss my current work and get insights from other people in the team who have different areas of specialization. I would also learn from these weekly meetings the importance of communication and collaboration in an engineering team. In one sprint I had to communicate my changes to the data engineer and explain the table structure I created to hold my data and what kind of data am I storing. After discussion we had discovered there was a simpler way to store the information and reduce the amount of redundant data stored by using other table data to link together. I was also tasked to gather requirements from other departments who had no technical background, and with my supervisors help we were able to gather requirements from non-technical people and transform them into technical requirements. Another part of working in a software engineering team was using Azure DevOps git repository to collaborate in code maintenance and development. I had learnt throughout the experience the importance of commit messages and pull requests. All changes required me to document what work I have done, and my supervisor and I would discuss changes before they were approved and merged into staging. Communicating in a technical and formal manner in an engineering team has helped build my communication skills I was lacking before my placement.

Part C: Reflection on Learning and Workplace Experience

- a) Write a reflection that compares and contrasts what you have learned at UNSW with your experiences as an engineer in the workplace
- b) Refer to the marking rubric for Part C: Reflection on Learning and Workplace Experience for examples
- c) Maximum 1,500 words for Part C

Compares and contrast industrial placement experiences and academic knowledge, and illuminate differences, as well as similarities, between the former and the later. The student also demonstrates deep understanding of fields of study and broaden own points of view due to the industrial placement experiences.

During my experience at Maia Financial and Credabl, I was in a small team consisting of an Irish software engineer who was my supervisor, a database engineer, an operations manager, software developers, UX designers and our team manager. These people were from different disciplines, cultural backgrounds and different age groups from all around the world. Whereas my teams in university were all fellow students mostly studying the same degree.

Working in a diverse team that had a wide skillset allowed us to share new insights, focus on our areas of specialization and count on the other teammates to complete their sections in order to bring it all together and have a well-polished final product. It also highlighted the importance of communication, documentation and design.

From my experience with SENG2011, I worked in a group to create a new product that satisfied certain requirements. I had to work with my team to create problem statements, walkthroughs, UML diagrams, ER diagrams and mockups before we began creating our final product. This was all facilitated and communicated through mandatory weekly meetings and further communication outside of that. Before my experience I had always told myself, this process was tedious and pointless. After this experience I have learnt the importance of documentation and design and how it is central to any successful product. This is because we are required to have one unified view of our final product. It also taught me the importance of communication within a team, and each person understanding their role. This course proved to be very practical and I found myself applying skills I gained from that course into challenges I faced during my placement.

When it came to programming or analysis during my time at Maia Financial and Credabl, security was something greatly emphasized and put as the highest priority. This is because any vulnerability can be exploited and cause potential damage to the business.

A key difference I found to programming in the workplace and programming at university is that, in university the main priority is to get the solution working and presentable to either autotests or internal staff. However, in the workplace we are presenting to clients and customers who could also have malicious intent and we needed to assume that any security flaw will be exploited. When I first began writing logic to perform SQL queries, my boss emphasized that allowing user input into our queries opens the risk of SQL injections or users running code which could cause damage to the business.

Through this experience it has encouraged me to take up COMP6441 in 2020 term 2 to understand security.

My communication skills were greatly strengthened during my experience at Maia financial and Credabl. I had to learn how to communicate with other software engineers and developers using Jira, git and having weekly sprints. I had also learnt how to communicate with non-technical people to gather requirements or present a product.

During my studies the importance of communication was always emphasized, but in the workplace, it was a necessity. Each week the team would gather for about an hour to discuss their progress on their tasks, get feedback from other members in the team and make sure everyone was on the right track. By communicating weekly amongst developers, it allowed us to complete tasks in a timely and efficient manner.

One of the courses I completed was COMP2511, where one assessment was a group project. The lecturer of the course had pushed us to use the same communication techniques I have been using almost every day in my placement. I used the documenting techniques I learnt from COMP2511 for almost all programming components of my placement, and whenever I was given a task, I would always follow the same process I learned from the course. This experience has highlighted and given me a greater understanding to the importance of the communication techniques required for software engineering.

Evaluates changes in learning before and after each industrial placement, recognizing complex contextual factors (e.g. works with ambiguity and risk, deals with frustration), demonstrate self-awareness and envisions a future self or develop plans that build on experiences or learnings from each industrial placement.

Being placed within a team of engineers opened my eyes to the amount of practical knowledge I was still lacking. I laid the foundations of software design and working within a team through lectures, tutorials, practice during labs and software engineering workshops at university. During my studies, time management was emphasized through strict punishment of late submissions, however I never understood the risk of not meeting deadlines in the real world, the consequences of poor time management and how much impact it has on a business.

For instance, before my training I would often spend an unreasonable amount of time dwelling on an incorrect solution. Now through my gained experience, I realize this wastes time and so in the future I will make sure to manage my time properly, and if I get stuck on a problem, I will be more open to starting fresh with brand new approaches. To do this I will begin using Trello to manage all my tasks, the details associated with them and set smaller achievable sub-goals that build into a final goal.

Whilst programming within a team, I realized that my documentation skills were not up to standards before my placement. During my course work, documentation was always encouraged and its importance in the workplace was made known, but in most cases, it was never assessed so I didn't take it as seriously as I should have.

Before my placement I would only take documentation seriously if it was an assessed component and never understood the practicality behind creating a design. I would often create and design as I went. This experience has taught me that documentation is out there to not only benefit myself, but other people who work with me in a team. It's a communication technique that allows everyone on the team to see on the same page and follow a guideline.

In the future, despite what project or assessment I am working on, I will make sure to create and maintain documentation that contains, the current problem I am working on, walkthroughs, ER diagrams and UML class diagrams. This will also help me make sure I stick to the right path when working and assist in my time management.

Something that I had learnt in my placement, that I had no prior experience to, was communicating technical requirements or products to non-technical people. Through one of the courses I had done, COMP3511 (Human Computer Interaction), I had learnt the importance as a software engineer to be able to communicate requirements to both technical and non-technical people.

Throughout my time with Maia financial and Credabl, my communication skills have developed, specifically with non-technical people from other departments. In a specific example, one department required some changes to a webpage to push out for marketing. My supervisor told the department lead to communicate with me and that he'd observe and assist as needed. In this experience, I asked a lot of questions at first that seemed confusing from a non-technical point of view, and my supervisor hinted at me to abstract all the programming jargon, draw diagrams and ask for exact behavior expected through a walkthrough. This experience had really boosted my confidence for my communication skills, and I feel a lot more comfortable communicating with non-technical people.

In the future, to help improve this skill, I plan to show all my family and friends any work I've done and try to explain at a high level what it does, whilst hiding the inner complexity and specifics. I now know that a large portion of being an engineer hinges around a set of requirements. Being able to gather

requirements from non-technical people and convert them to a technical specification is a skill I will continue to work on throughout all courses I continue to do at university.

In summary, what I learnt at university have been the building blocks for the workplace. Throughout my placement, I was always scared I was not prepared and that I couldn't complete the tasks. With the amazing help of the team I was with, they helped bring out the best in me and show me that I was capable of performing. I always thought beforehand that what I learnt in university would not be practical in the workplace, due to how many people have told me the real world is very different. What I discovered myself doing throughout my placement was falling back to my prior experience and work I had done at university. During times I was unsure of how to progress or what to do, I would go back to the fundamentals, create a design document and follow it. This experience has helped push me even further and shown me the work and responsibilities of a software engineer. I am really looking forward to getting back into university and applying all the skills I have learnt to all my future course work.

Appendix

Include all of your Employer Evaluation Forms in the appendix of your report.

- a) The Employer Evaluation Forms must total 60 days
- b) Have been signed off by your placement supervisor

Please insert Employer Evaluation Forms here:

Please see the next page, I have attached the pdf file of my employer evaluation form on the next page.