

ALEX LAIN

AlexGeorgeLain/portfolio alexgeorgelain.github.io Alex Lain

MENU ☰

OBJECT-ORIENTED INFORMATION SYSTEMS

IN THIS MODULE, STUDENTS WILL CONTEXTUALISE AND DEVELOP INFORMATION SYSTEMS USING OBJECT-ORIENTED APPROACHES WITH VARIOUS PROGRAMMING LANGUAGES, SUCH AS PYTHON AND SQL. THIS INVOLVES EQUIPPING STUDENTS WITH THE PRACTICAL SKILLS REQUIRED TO BE ABLE TO ANALYSE AN INFORMATION SYSTEM PROBLEM, DESIGN AND IMPLEMENT A SOLUTION USING VARIOUS INDUSTRY STANDARD TOOLS AND TECHNIQUES.

LEARNING OUTCOMES

I will learn to:

- Appraise and evaluate critically the concepts and principles of information systems.
- Design or modify and document an object-oriented information system using appropriate tools.
- Develop an object-oriented information system design, implementing this knowledge in applicable programming languages, such as Python and SQL.
- Develop, implement and evaluate critically information system solutions to facilitate business decisions.

ASSIGNMENTS & GRADES

Overall module grade: TBC

Assignment	Weighting	Submission Unit	Grade
Mid Module Assignment: System Design	30%	Unit 7	58
System Implementation	30%	Unit 11	-
End of Module Assignment: e-Portfolio	40%	Unit 12	-

Object-oriented Information Systems

In this module, students will contextualise and develop information systems using object-oriented approaches with various programming languages, such as Python and SQL. This involves equipping students with the practical skills required to be able to analyse an information system problem, design and implement a solution using various industry standard tools and techniques.

Learning Outcomes

I will learn to:

- Appraise and evaluate critically the concepts and principles of information systems.
- Design or modify and document an object-oriented information system using appropriate tools.
- Develop an object-oriented information system design, implementing this knowledge in applicable programming languages, such as Python and SQL.
- Develop, implement and evaluate critically information system solutions to facilitate business decisions.

Assignments & Grades

Overall module grade: TBC

Assignment	Weighting	Submission Unit	Grade
Mid Module Assignment: System Design	30%	Unit 7	58
System Implementation	30%	Unit 11	-
End of Module Assignment: e-Portfolio	40%	Unit 12	-

ALEX LAIN

REFLECTIONS AND ARTEFACTS

REFLECTIONS ON VARIOUS ASPECTS OF THE WORK COMPLETED IN THE OBJECT-ORIENTED INFORMATION SYSTEMS MODULE, AS WELL AS THE ASSIGNMENTS SUBMITTED DURING THE COURSE OF THIS MODULE.

UNIT 1: INTRODUCTION TO INFORMATION SYSTEMS

Learning Objectives:

- Understand the core elements of an information system.
- Appreciate the common issues that can arise when deploying a system.
- Gain an understanding of the SDLC, its phases and their importance.

Required Reading:

- Sommerville, I. (2016) Software Engineering. Harlow: Pearson. Chapter 1.
- BBC (2015) US Prisoners Released Early by Software Bug.
- Barrett, L. (2003) Hospital Revives its Dead Patients

Collaborative Discussion: Information System Failure - Initial Post:

Name	Post
Alexander	The Queensland Health Payroll System
George	
Lain:	Background

In December 2007 IBM signed a contract worth AU\$98 million to provide a standardized payroll

ALEX LAIN

AlexGeorgeLain/portfolio

alexgeorgelain.github.io

Alex Lain

MENU ☰

Name	Post
Alexander George Lain:	The Queensland Health Payroll System
	Background

In December 2007 IBM signed a contract worth AU\$98 million to provide a standardized payroll software solution to all Queensland State government departments. The objective of this shared services initiative was to consolidate technology and resources to produce several key benefits including increased workforce mobility, easier visibility of service costs, licensing costs and personnel reductions, and consistency of information throughout government agencies. By the end of 2008 IBM had already been paid AU\$32 million, but had failed to achieve any of the "contracted performance criteria". This lead to the unified payroll system being cancelled, and the scope of the project being reduced to building a payroll system solely for Queensland Health which was using a system which was soon to become obsolete.

Due to the specific need of Queensland Health including concurrent employment and a complex rewards structure, a number of customizations had to be made to the standard Workbrain and SAP platforms. Initial testing in July 2009 on 10% of the employee roster resulted in a AU\$1.2 million discrepancy in the biweekly payroll when compared with the soon-to-be obsolete system. A second round of testing in February 2010 produced a AU\$30,000 discrepancy. Despite the risks highlighted in testing, Queensland Health decided to put the system live without full testing being completed in March 2010. By 2012 this system had resulted in over 35,000 payroll mistakes and had cost the state more than AU\$400 million to operate, and a further AU\$112 million in overpaid salaries by May 2012.

System Issues and Failures

- During the cut-over period there were issues with the availability of the system to staff which reduced the processing time available. This created a backlog of payroll forms and unprocessed adjustments which took roughly 8 months to process.
- Queensland Health allowed staff to submit forms for work completed up to six years prior which created significant performance issues.
- System bug fixes and improvements were on a fixed release schedule which lead to delays in fixing issues which could have impacted staff pay.

References

Dolfing, H. (2019) Case Study 9: The Payroll System That Cost Queensland Health AU\$1.25 Billion.
Available From: <https://www.henricodolfing.com/2019/12/project-failure-case-study-queensland/>

ALEX LAIN

AlexGeorgeLain/portfolio

alexgeorgelain.github.io

Alex Lain

MENU ☰

issues which could have impacted staff pay

References

Dolfing, H. (2019) Case Study 9: The Payroll System That Cost Queensland Health AU\$1.25 Billion. Available From: <https://www.henricodolfing.com/2019/12/project-failure-case-study-queensland-health.html>

Anrich Potgelter:

Hi Alexander

Thank you for sharing this information system failure; the Queensland Health payroll failure will be studied for years to come and is likely one of the greatest failures of its kind in information system history. I thoroughly enjoyed researching this failure, and I will unpack some of my additional findings below.

As you mentioned in your initial post, Queensland Health was in dire need of innovation in its information management infrastructure. The intentions of this project were more significant than its intended outcome. Additionally, I discovered that the previous payroll system developed by an organisation called LATTICE had reached end of life and LATTICE had indicated that they would no longer support the service level agreement initially signed with the government. As a result, Queensland Health was in a precarious situation where they needed a solution fast.

Queensland Health opened a series of bidding processes in which IBM was awarded the contract to consult and facilitate the new payroll system's development. When IBM joined the project, their indicative timeline put them within the same month as the previous system would reach end of life. After an audit by KPMG, it was found that the 8-month projection was unattainable and unrealistic, to say the least.

Along with the impossible deadlines it was later found that Queensland Health had decided to perform a hard cut over to the new system without running both systems in parallel to ensure any risks of system failure were mitigated. As a result of the decision to transition to the new system without sufficient testing Queensland Health was thrown into a downward spiral of missed deadlines and incomplete systems, leading to backlogs in the payroll system which would amount to \$422mil in additional costs to rectify the problems caused.

How could this have been prevented?

- A development lifecycle plan would have saved Queensland Health millions as they would have planned for the evolution of their systems rather than leaving them to become antiquated.
- A strict testing regime should have been in place to prevent a situation as we saw here with the hard

ALEX LAIN

the problems caused.

How could this have been prevented?

- A development lifecycle plan would have saved Queensland Health millions as they would have planned for the evolution of their systems rather than leaving them to become antiquated.
- A strict testing regime should have been in place to prevent a situation as we saw here with the hard cut over to a new untested system.
- A greater understanding of the organisation and the IT assets were clearly required to understand the complexities of the organisation.
- Sufficient planning is required for a system of this size, and this was clearly not done, as similar endeavours by other government organisations have taken up to three years to complete.

References

Glass, R. (2013) 'The Queensland Health Payroll Debacle', *Information Systems Management*, 30(1), pp. 89-90.

Paterno, P. and Zhao, S. (2018) 'Queensland Health: Australia's Healthcare IT Catastrophe', *Proceedings of the Northeast Business & Economics Association*, pp. 245-248.

Silva, A. R. and Rosemann, M. (2012) Integrating organisational design with IT design: The Queensland health payroll case. *Lecture Notes in Computer Science*.

Man Sze Wong:

Hi Alexander,

I understand that payroll system can be one of the important systems in an organisation because it can affect the staff pay and may have an impact in the financial area of the company.

Other than the financial area, the payroll system can affect the reputation of an organisation as well. A payroll system can actually be one of the elements to maintain a good reputation of an organisation because a good payroll system can ensure the employee get the salary on time. If the employee can get the salary on time and without any mistake on the salary calculation then the employee in the organisation will think that the organisation is reliable and a good payroll system can reduce the contingent liabilities of an organisation because the no employee will sue the organisation due to the labor dispute.

So, system bug fixes and improvement have to be performed as soon as possible once the staffs in IT department figured out some problems of the payroll system because it can have a huge impact to an organisation.

ALEX LAIN

AlexGeorgeLain/portfolio alexgeorgelain.github.io Alex Lain

MENU ☰

Glass, R. (2013) 'The Queensland Health Payroll Debacle', *Information Systems Management*, 30(1), pp. 89-90.

Paterno, P. and Zhao, S. (2018) 'Queensland Health: Australia's Healthcare IT Catastrophe', *Proceedings of the Northeast Business & Economics Association*, pp. 245-248.

Silva, A. R. and Rosemann, M. (2012) Integrating organisational design with IT design: The Queensland health payroll case. *Lecture Notes in Computer Science*.

Man Sze Wong: Hi Alexander,

I understand that payroll system can be one of the important systems in an organisation because it can affect the staff pay and may have an impact in the financial area of the company.

Other than the financial area, the payroll system can affect the reputation of an organisation as well. A payroll system can actually be one of the elements to maintain a good reputation of an organisation because a good payroll system can ensure the employee get the salary on time. If the employee can get the salary on time and without any mistake on the salary calculation then the employee in the organisation will think that the organisation is reliable and a good payroll system can reduce the contingent liabilities of an organisation because the no employee will sue the organisation due to the labor dispute.

So, system bug fixes and improvement have to be performed as soon as possible once the staffs in IT department figured out some problems of the payroll system because it can have a huge impact to an organisation.

Reflections:

The main focus of the first week's work centred around information system failures. The articles assigned for required reading highlighted two instances in which the failure of an information system had serious consequences. In addition to these two incidents, in the Collaborative Discussion which ran from Unit 1 to Unit 3, each participant on the course had to research an information system failure (not listed here <https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html>) and write a short post to inform the rest of the group. My initial post can be seen above, along with two of my peers' responses. This discussion demonstrated that a wide range of failures can occur throughout the entire process of planning, building, implementing, and operating a system. It was interesting to see that no industry seemed to be immune to the mistakes or oversights which can cause a system to fail, regardless of how damaging a failure had the potential to be.

Unit 1: Introduction to Information Systems

Learning Objectives:

- Understand the core elements of an information system.
- Appreciate the common issues that can arise when deploying a system.
- Gain an understanding of the SDLC, its phases and their importance.

Required Reading:

- Sommerville, I. (2016) *Software Engineering*. Harlow: Pearson. Chapter 1
- BBC (2015) US Prisoners Released Early by Software Bug.
- Barrett, L. (2003) Hospital Revives its Dead Patients

Collaborative Discussion: Information System Failure - Initial Post:

Alexander George Lain:

The Queensland Health Payroll System

Background

In December 2007 IBM signed a contract worth AU\$98 million to provide a standardized payroll software solution to all Queensland State government departments. The objective of this shared services initiative was to consolidate technology and resources to produce several key benefits including increased workforce mobility, easier visibility of service costs, licensing costs and personnel reductions, and consistency of information throughout government agencies. By the end of 2008 IBM had already been paid AU\$32 million, but had failed to achieve any of the “contracted performance criteria”. This lead to the unified payroll system being cancelled, and the scope of the project being reduced to building a payroll system solely for Queensland Health which was using a system which was soon to become obsolete.

Due to the specific need of Queensland Health including concurrent employment and a complex rewards structure, a number of customizations had to be made to the standard Workbrain and SAP platforms. Initial testing in July 2009 on 10% of the employee roster resulted in a AU\$1.2 million discrepancy in the biweekly payroll when compared with the soon-to-be obsolete system. A second round of testing in February 2010 produced a AU\$30,000 discrepancy. Despite the risks highlighted in testing, Queensland Health decided to put the system live without full testing being completed in March 2010. By 2012 this system had resulted in over 35,000 payroll mistakes and had cost the state more than AU\$400 million to operate, and a further AU\$112 million in overpaid salaries by May 2012.

System Issues and Failures

- During the cut-over period there were issues with the availability of the system to staff which reduced the processing time available. This created a backlog of payroll forms and unprocessed adjustments which took roughly 8 months to process.

- Queensland Health allowed staff to submit forms for work completed up to six years prior which created significant performance issues.
- System bug fixes and improvements were on a fixed release schedule which lead to delays in fixing issues which could have impacted staff pay.

References

Dolfing, H. (2019) Case Study 9: The Payroll System That Cost Queensland Health AU\$1.25 Billion. Available From: <https://www.henricodolfing.com/2019/12/project-failure-case-study-queensland-health.html>

Anrich Potgeiter:

Hi Alexander

Thank you for sharing this information system failure; the Queensland Health payroll failure will be studied for years to come and is likely one of the greatest failures of its kind in information system history. I thoroughly enjoyed researching this failure, and I will unpack some of my additional findings below.

As you mentioned in your initial post, Queensland Health was in dire need of innovation in its information management infrastructure. The intentions of this project were more significant than its intended outcome. Additionally, I discovered that the previous payroll system developed by an organisation called LATTICE had reached end of life and LATTICE had indicated that they would no longer support the service level agreement initially signed with the government. As a result, Queensland Health was in a precarious situation where they needed a solution fast.

Queensland Health opened a series of bidding processes in which IBM was awarded the contract to consult and facilitate the new payroll system's development. When IBM joined the project, their indicative timeline put them within the same month as the previous system would reach end of life. After an audit by KPMG, it was found that the 8-month projection was unattainable and unrealistic, to say the least.

Along with the impossible deadlines it was later found that Queensland Health had decided to perform a hard cut over to the new system without running both systems in parallel to ensure any risks of system failure were mitigated. As a result of the decision to transition to the new system without sufficient testing Queensland Health was thrown into a downward spiral of missed deadlines and incomplete systems, leading to backlogs in the payroll system which would amount to \$422mil in additional costs to rectify the problems caused.

How could this have been prevented?

- A development lifecycle plan would have saved Queensland Health millions as they would have planned for the evolution of their systems rather than leaving them to become antiquated.
- A strict testing regime should have been in place to prevent a situation as we saw here with the hard cut over to a new untested system.
- A greater understanding of the organisation and the IT assets were clearly required to understand the complexities of the organisation.

- Sufficient planning is required for a system of this size, and this was clearly not done, as similar endeavours by other government organisations have taken up to three years to complete.

References

Glass, R. (2013) 'The Queensland Health Payroll Debacle', *Information Systems Management*, 30(1), pp. 89-90.

Paterno, P. and Zhao, S. (2018) 'Queensland Health: Australia's Healthcare IT Catastrophe', *Proceedings of the Northeast Business & Economics Association*, pp. 245-248.

Silva, A. R. and Rosemann, M. (2012) Integrating organisational design with IT design: The Queensland health payroll case. *Lecture Notes in Computer Science*.

Man Sze Wong:

Hi Alexander,

I understand that payroll system can be one of the important systems in an organisation because it can affect the staff pay and may have an impact in the financial area of the company.

Other than the financial area, the payroll system can affect the reputation of an organisation as well. A payroll system can actually be one of the elements to maintain a good reputation of an organisation because a good payroll system can ensure the employee get the salary on time. If the employee can get the salary on time and without any mistake on the salary calculation then the employee in the organisation will think that the organisation is reliable and a good payroll system can reduce the contingent liabilities of an organisation because the no employee will sue the organisation due to the labor dispute.

So, system bug fixes and improvement have to be performed as soon as possible once the staffs in IT department figured out some problems of the payroll system because it can have a huge impact to an organisation.

Reflections:

The main focus of the first week's work centred around information system failures. The articles assigned for required reading highlighted two instances in which the failure of an information system had serious consequences. In addition to these two incidents, in the Collaborative Discussion which ran from Unit 1 to Unit 3, each participant on the course had to research an information system failure (not listed here <https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html>) and write a short post to inform the rest of the group. My initial post can be seen above, along with two of my peers' responses. This discussion demonstrated that a wide range of failures can occur throughout the entire process of planning, building, implementing, and operating a system. It was interesting to see that no industry seemed to be immune to the mistakes or oversights which can cause a system to fail, regardless of how damaging a failure had the potential to be.

ALEX LAIN

AlexGeorgeLain/portfolio alexgeorgelain.github.io Alex Lain

MENU ☰

UNIT 2: INFORMATION SYSTEMS AND THEIR IMPORTANCE

Learning Objectives:

- Understand the core components of an information system and their importance.
- Evaluate a range of issues and appropriate solutions or mitigations.

Required Reading:

- Bourgeois, D. (2014) Welcome to Information Systems for Business and Beyond. Saylor Academy. Chapter 1: Introduction to Information systems.

Collaborative Discussion: Information System Failure - Peer Response 2:

Name	Post
Antonios	My case study includes an IT failure that was a result of a combination of problematic hardware and software.
Kalaitzakis	

[Introduction](#)

On the 29th of October 2018, a Boeing 737-Max 8 operating the domestic Lion Air flight JT610 from Jakarta to Pangkal Pinang went down resulting in the death of 189 people on-board (BBC, 2018). On the 10th of March 2019, the Boeing's 737 Max 8 flight ET302 operated by Ethiopian Airlines from Addis Ababa to Nairobi in Kenya went down, killing all 157 people on-board (BBC, 2019). As a result of these two accidents, aviation authorities globally banned the operation of the Boeing 737 Max 8 and opened an investigation to find out the reasons that led to these accidents.

[Technical Theory and Reasons of the Accidents](#)

The Boeing 737 aircraft has come a long way after its first flight in April 1967. Back then it was a small aircraft with small engines and relatively simple IT systems. Over the years, high market demand and technological advances have shifted the Boeing 737 into a bigger and much more sophisticated aircraft (Travis, 2019). As the years go by, and competition becomes more fierce, aircraft manufacturing companies are trying to make their aircraft more efficient and more eco-friendly by

ALEX LAIN

AlexGeorgeLain/portfolio

alexgeorgelain.github.io

Alex Lain

MENU ☰

Technical Theory and Reasons of the Accidents

The Boeing 737 aircraft has come a long way after its first flight in April 1967. Back then it was a small aircraft with small engines and relative simple IT systems. Over the years, high market demand and technological advances have shifted the Boeing 737 into a bigger and much more sophisticated aircraft (Travis, 2019). As the years go by, and competition becomes more fierce, aircraft manufacturing companies are trying to make their aircraft more efficient and more eco-friendly by cutting down carbon emissions. In May 2017, Boeing released the updated version of its 737 model, the 737 Max 8.

The 737 Max 8 compared to previous models had larger engines, which in theory would result in higher efficiency and usage of less fuel (Travis, 2019). Having said that, as the 737 Max was fitted with larger engines, the space between the ground and the lower part of the engine got tighter (Travis, 2019). In order to overcome this problem, Boeing changed the shape of the engines from circular to oval and moved the engines higher and in front of the wings. However, by changing the placing of the engines, the aircraft's centreline changed. As a consequence of this change, whenever pilots would apply power to the engines, the aeroplane would tend to raise its nose. While at first, this might not sound like a serious problem, it can actually result in an aeroplane stall. Aeroplane stall is when the aeroplane's nose angle is very high resulting in a non-sufficient amount of air travelling under the aeroplane's wings to keep it up. As a result, the aeroplane drops endlessly, until it is correctly adjusted (Monroe Aerospace, 2019).

With the intention of all-avoiding an aeroplane stall in the most inexpensive way, Boeing installed a sensor accompanied with its own computer software called MCAS (Travis, 2019). Two sensors were placed on the front and two on the side of the aeroplane, measuring the speed of the air and the direction/angle on which the air was "hitting" the aeroplane. MCAS' software would then analyse the data flowing out of these sensors and if it finds that the angle of the nose is too high, it would automatically command the aircraft's system to "push down" both the nose of the aeroplane and the pilot's control wheel (Travis, 2019).

While the decision of installing a completely automatic system to correct any angle problems seemed clever, there was a chain of problems that would mark the system unreliable. First, even though an aeroplane has numerous different systems checking that everything works as it should from which the MCAS could input data from, the MCAS system was only getting input from these four sensors, which were prone to providing false data. Additionally, the hardware and software of MCAS were configured in a way in which pilots were not able to override it, even if they knew the data flowing in the MCAS system was wrong. After the investigation of both accidents, it was made clear that the MCAS system was lowering the nose of the aeroplane even when it was cruising, heading the aeroplane to the

ALEX LAIN

AlexGeorgeLain/portfolio alexgeorgelain.github.io Alex Lain

MENU

pilot's control wheel (Travis, 2019)

While the decision of installing a completely automatic system to correct any angle problems seemed clever, there was a chain of problems that would mark the system unreliable. First, even though an aeroplane has numerous different systems checking that everything works as it should from which the MCAS could input data from, the MCAS system was only getting input from these four sensors, which were prone to providing false data. Additionally, the hardware and software of MCAS were configurated in a way in which pilots were not able to override it, even if they knew the data flowing in the MCAS system was wrong. After the investigation of both accidents, it was made clear that the MCAS system was lowering the nose of the aeroplane even when it was cruising, heading the aeroplane to the ground (Travis, 2019).

Impact of failure

We should not underestimate the power of IT systems. Even the smallest error can cause a huge tragedy. In Boeing's case, an executive decision to rely on a fully automatic system resulted in two air accidents and 346 deaths.

In addition, Boeing suffered huge losses, not only economic but also in the trust of its customers (Dyke and Urry, 2020).

In 2021, after correcting all faulty systems, the Boeing 737 Max 8 was cleared to fly again almost globally (BBC, 2021). However, one in four travellers would still avoid flying with it, even if they would have to pick a layover flight (D'Ambrosio, 2021).

References:

BBC. (2018) Lion Air crash: Boeing 737 plane crashes in sea off Jakarta Available from: <https://www.bbc.com/news/world-asia-46014463> [Accessed 08 February 2021].

BBC. (2019) Ethiopian Airlines: 'No survivors' on crashed Boeing 737 Available from: <https://www.bbc.com/news/world-africa-47513508> [Accessed 08 February 2021].

BBC. (2021) Boeing 737 Max cleared to fly in UK and EU after crashes Available from: <https://www.bbc.com/news/business-55827358> [Accessed 08 February 2021].

IFFF. (2019) How the Boeing 737 Max Disaster Looks to a Software Developer Available from: <https://medium.com/@alexlain/how-the-boeing-737-max-disaster-looks-to-a-software-developer-10a2a2a2a2a2> [Accessed 08 February 2021].

ALEX LAIN

AlexGeorgeLain/portfolio

alexgeorgelain.github.io

Alex Lain

MENU ☰

BBC. (2019) Ethiopian Airlines, 'No survivors' on crashed Boeing 737 Available from: <https://www.bbc.com/news/world-africa-47513508> [Accessed 08 February 2021].

BBC. (2021) Boeing 737 Max cleared to fly in UK and EU after crashes Available from: <https://www.bbc.com/news/business-55827358> [Accessed 08 February 2021].

IEEE (2019) How the Boeing 737 Max Disaster Looks to a Software Developer Available from: <https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer> [Accessed 07 February 2021].

Mondaq (2020) UK: The Impact Of The Boeing 737 Max Grounding Available from: <https://www.mondaq.com/uk/aviation/884752/the-impact-of-the-boeing-737-max-grounding> [Accessed 09 February 2021].

Monroe Aerospace (2019) What happens when an airplane stalls Available from: <https://monroeaerospace.com/blog/what-happens-when-an-airplane-stalls/> [Accessed 08 February 2021].

Travelmarketreport (2019) Many Consumers Don't Want to Fly the 737 Max When it Returns Available from: <https://www.travelmarketreport.com/articles/Many-Consumers-Dont-Want-to-Fly-the-737-Max-When-it-Returns> [Accessed 09 February 2021].

Alexander George Lain:

Hi Antonios,

This is an interesting and, as you said, tragic case that demonstrates the severe consequences that an IT failure can have. I found it interesting that 1/4 of travellers would still avoid flights using this aircraft as this highlights the impact this event had on the public consciousness when compared with other information system failures such as data leaks are often lost quite quickly due to today's fast news cycle. Another interesting point I found was that during a review of a proposed update a new software issue was found which further delayed the plane's return to service. This issue further demonstrated how easily errors can occur, even on projects that are as high profile as this.

References

Shepardson, D. (2020) Boeing addresses new 737 MAX software issue that could keep plane grounded longer. Available from: https://www.reuters.com/article/uk-boeing-737max-software-idUKKBN1ZG289?edition=redirect&utm_source=feedburner&utm_medium=RSS&utm_campaign=Feed%2B-%2BREUTERS+Technology+News

ALEX LAIN

AlexGeorgeLain/portfolio alexgeorgelain.github.io Alex Lain

MENU ☰

Collaborative Discussion: Information System Failure - Peer Response 2:

Name	Post
Uzyar Parak:	<p>Incident</p> <p>Adobe, in 2017, had announced its decision to remove support from its Asboe Flash player, and discontinue its use. This was in part, due to many vulnerabilities in its design, which lead to malware such as RIG Exploit Kit, Astrum Exploit Kit, Matrix Ransomware, Angler Exploit Kit and Ramnit, as well as others leading to information being insecure(1). Adobe's plan was for complete discontinuation from the 12th January 2021, so as to allow enough time for companies to migrate to new and more secure systems(2).</p> <p>The South African Revenue Services, collects tax revenue from approximately 5 million tax paying citizens in South Africa, as well as over 700 000 companies. VAT and customs duty collections are also managed by SARS(3).</p> <p>Despite having a three-year warning, SARS was apparently caught unawares regarding migrating forms on its electronic filing site to the more secure and updated HTML5 software.</p> <p>Consequences</p> <p>As a result of this incompetence, many users of the e-filing site have been unable to complete documents and submit said documents to SARS. To mitigate the impact, SARS has developed a special browser for users to file their tax returns. The browser is specially designed to use the outdated Adobe Flash player, to enable tax filing submission, against the advice of Adobe(4).</p> <p>This may lead to increased exploitation by malicious parties. It is an astounding failure of the ICT team, which may impact revenue collection, or expose private citizens to unnecessary risk.</p> <p>References</p> <ol style="list-style-type: none">1. Donnelly S. 'Soft Target: The Top 10 Vulnerabilities Used by Cybercriminals [Internet]. [cited 2021 Feb 11]. Available from: www.recordedfuture.com2. Adobe will finally kill Flash in 2020 - The Verge [Internet]. [cited 2021 Feb 11]. Available from: https://www.theverge.com/2017/7/25/16026236/adobe-flash-end-of-support-20203. This is who pays taxes in South Africa [Internet]. [cited 2021 Feb 11]. Available from: [REDACTED]

ALEX LAIN

AlexGeorgeLain/Portfolio

Alex George Lain

MENU ☰

References

1. Donnelly S 'Soft Target: The Top 10 Vulnerabilities Used by Cybercriminals [Internet]. [cited 2021 Feb 11]. Available from: www.recordedfuture.com
2. Adobe will finally kill Flash in 2020 - The Verge [Internet]. [cited 2021 Feb 11]. Available from: <https://www.theverge.com/2017/7/25/16026236/adobe-flash-end-of-support-2020>
3. This is who pays taxes in South Africa [Internet]. [cited 2021 Feb 11]. Available from: <https://businesstech.co.za/news/finance/458708/this-is-who-pays-taxes-in-south-africa/>
4. SARS launches new eFiling browser to solve Flash problems [Internet]. [cited 2021 Feb 11]. Available from: <https://mybroadband.co.za/news/business/384210-sars-launches-new-efiling-browser-to-solve-flash-problems.html>

Alexander George Lain: Hi Uzayr,

This is an interesting case, similar to the Queensland Health payroll issue that I posted about. In both cases there seems to have been a slow response to replace software that had been deprecated, despite having several years over which to implement a solution. In the SARS case, the decision to develop a browser based on an out of date version of Chromium as a way to work around recent builds of Flash which stopped working on 12 January 2021 is astounding. This solution also seems to only be available for Windows users, with no alternatives being provided to the Mac or Linux users that make up a non-zero portion of the South African tax paying population.

References

Salter, J. (2021) Flash is dead - but South Africa didn't get the memo. Available from: <https://arstechnica.com/gadgets/2021/02/flash-no-longer-works-in-browsers-so-south-africa-made-its-own-browser/>

Reflections:

This week revolved around learning to understand the core components of an information system, as well as a range of issues that can arise when using a system, and the steps which can be taken to solve or mitigate the problem. This included continuing with the Collaborative Discussion and providing responses to two of our peers' initial posts. Their post and my responses can be seen above. The week's reading also gave a brief and interesting overview of the history and evolution of information systems.

The first seminar of the module was also held this week, with the topic of En Route Automation Modernization.

Unit 2: Information Systems and their Importance

Learning Objectives:

- Understand the core components of an information system and their importance.
- Evaluate a range of issues and appropriate solutions or mitigations.

Required Reading:

- Bourgeois, D. (2014) *Welcome to Information Systems for Business and Beyond*. Saylor Academy. Chapter 1: *Introduction to Information systems*.

Collaborative Discussion: Information System Failure - Peer Response 1:

Antonios Kalaitzakis:

My case study includes an IT failure that was a result of a combination of problematic hardware and software.

Introduction

On the 29th of October 2018, a Boeing 737-Max 8 operating the domestic Lion Air flight JT610 from Jakarta to Pangkal Pinang went down resulting in the death of 189 people on-board (BBC, 2018). On the 10th of March 2019, the Boeing's 737 Max 8 flight ET302 operated by Ethiopian Airlines from Addis Ababa to Nairobi in Kenya went down, killing all 157 people on-board (BBC, 2019). As a result of these two accidents, aviation authorities globally banned the operation of the Boeing 737 Max 8 and opened an investigation to find out the reasons that led to these accidents.

Technical Theory and Reasons of the Accidents

The Boeing 737 aircraft has come a long way after its first flight in April 1967. Back then it was a small aircraft with small engines and relative simple IT systems. Over the years, high market demand and technological advances have shifted the Boeing 737 into a bigger and much more sophisticated aircraft (Travis, 2019). As the years go by, and competition becomes more fierce, aircraft manufacturing companies are trying to make their aircraft more efficient and more eco-friendly by cutting down carbon emissions. In May 2017, Boeing released the updated version of its 737 model, the 737 Max 8.

The 737 Max 8 compared to previous models had larger engines, which in theory would result in higher efficiency and usage of less fuel (Travis, 2019). Having said that, as the 737 Max was fitted with larger engines, the space between the ground and the lower part of the engine got tighter (Travis, 2019). In order to overcome this problem, Boeing changed the shape of the engines from circular to oval and moved the engines higher and in front of the wings. However, by changing the placing of the engines, the aircraft's centreline changed. As a consequence of this change, whenever pilots would

apply power to the engines, the aeroplane would tend to raise its nose. While at first, this might not sound like a serious problem, it can actually result in an aeroplane stall. Aeroplane stall is when the aeroplane's nose angle is very high resulting in a non-sufficient amount of air travelling under the aeroplane's wings to keep it up. As a result, the aeroplane drops endlessly, until it is correctly adjusted (Monroe Aerospace, 2019).

With the intention of all-avoiding an aeroplane stall in the most inexpensive way, Boeing installed a sensor accompanied with its own computer software called MCAS (Travis, 2019). Two sensors were placed on the front and two on the side of the aeroplane, measuring the speed of the air and the direction/angle on which the air was "hitting" the aeroplane. MCAS' software would then analyse the data flowing out of these sensors and if it finds that the angle of the nose is too high, it would automatically command the aircraft's system to "push down" both the nose of the aeroplane and the pilot's control wheel (Travis, 2019).

While the decision of installing a completely automatic system to correct any angle problems seemed clever, there was a chain of problems that would mark the system unreliable. First, even though an aeroplane has numerous different systems checking that everything works as it should from which the MCAS could input data from, the MCAS system was only getting input from these four sensors, which were prone to providing false data. Additionally, the hardware and software of MCAS were configurated in a way in which pilots were not able to override it, even if they knew the data flowing in the MCAS system was wrong. After the investigation of both accidents, it was made clear that the MCAS system was lowering the nose of the aeroplane even when it was cruising, heading the aeroplane to the ground (Travis, 2019).

Impact of failure

We should not underestimate the power of IT systems. Even the smallest error can cause a huge tragedy. In Boeing's case, an executive decision to rely on a fully automatic system resulted in two air accidents and 346 deaths. In addition, Boeing suffered huge losses, not only economic but also in the trust of its customers (Dyke and Urry, 2020).

In 2021, after correcting all faulty systems, the Boeing 737 Max 8 was cleared to fly again almost globally (BBC, 2021). However, one in four travellers would still avoid flying with it, even if they would have to pick a layover flight (D'Ambrosio, 2021).

References:

BBC. (2018) Lion Air crash: Boeing 737 plane crashes in sea off Jakarta Available from: <https://www.bbc.com/news/world-asia-46014463> [Accessed 08 February 2021].

BBC. (2019) Ethiopian Airlines: 'No survivors' on crashed Boeing 737 Available from: <https://www.bbc.com/news/world-africa-47513508> [Accessed 08 February 2021].

BBC. (2021) Boeing 737 Max cleared to fly in UK and EU after crashes Available from: <https://www.bbc.com/news/business-55827358> [Accessed 08 February 2021].

IEEE (2019) How the Boeing 737 Max Disaster Looks to a Software Developer Available from: <https://spectrum.ieee.org/aerospace/aviation/how-the-boeing-737-max-disaster-looks-to-a-software-developer> [Accessed 07 February 2021].

Mondaq (2020) UK: The Impact Of The Boeing 737 Max Grounding Available from: <https://www.mondaq.com/uk/aviation/884752/the-impact-of-the-boeing-737-max-grounding> [Accessed 09 February 2021].

Monroe Aerospace (2019) What happens when an airplane stalls Available from: <https://monroeaerospace.com/blog/what-happens-when-an-airplane-stalls/> [Accessed 08 February 2021].

Travelmarketreport (2019) Many Consumers Don't Want to Fly the 737 Max When it Returns Available from: <https://www.travelmarketreport.com/articles/Many-Consumers-Dont-Want-to-Fly-the-737-Max-When-it-Returns> [Accessed 09 February 2021].

Alexander George Lain:

Hi Antonios,

This is an interesting and, as you said, tragic case that demonstrates the severe consequences that an IT failure can have. I found it interesting that 1/4 of travellers would still avoid flights using this aircraft as this highlights the impact this event had on the public consciousness when compared with other information system failures such as data leaks are often lost quite quickly due to today's fast news cycle. Another interesting point I found was that during a review of a proposed update a new software issue was found which further delayed the plane's return to service. This issue further demonstrated how easily errors can occur, even on projects that are as high profile as this.

References

Shepardson, D. (2020) Boeing addresses new 737 MAX software issue that could keep plane grounded longer. Available from: <https://www.reuters.com/article/uk-boeing-737max-software-idUKKBN1ZG289?edition-redirect=uk>

Collaborative Discussion: Information System Failure - Peer Response 2:

Uzyar Parak:

Incident

Adobe, in 2017, had announced its decision to remove support from its Asboe Flash player, and discontinue its use. This was in part, due to many vulnerabilities in its design, which lead to malware such as RIG Exploit Kit, Astrum Exploit Kit, Matrix Ransomware, Angler Exploit Kit and Ramnit, as well as others leading to information being insecure(1). Adobe's plan was for complete discontinuation from the 12th January 2021, so as to allow enough time for companies to migrate to new and more secure systems(2).

The South African Revenue Services, collects tax revenue from approximately 5 million tax paying citizens in South Africa, as well as over 700 000 companies. VAT and customs duty collections are also managed by SARS(3).

Despite having a three-year warning, SARS was apparently caught unawares regarding migrating forms on its electronic filling site to the more secure and updated HTML5 software.

Consequences

As a result of this incompetence, many users of the e-filing site have been unable to complete documents and submit said documents to SARS. To mitigate the impact, SARS has developed a special browser for users to file their tax returns. The browser is specially designed to use the outdated Adobe Flash player, to enable tax filing submission, against the advice of Adobe(4).

This may lead to increased exploitation by malicious parties. It is an astounding failure of the ICT team, which may impact revenue collection, or expose private citizens to unnecessary risk.

References

1. Donnelly S. 'Soft Target: The Top 10 Vulnerabilities Used by Cybercriminals [Internet]. [cited 2021 Feb 11]. Available from: www.recordedfuture.com
2. Adobe will finally kill Flash in 2020 - The Verge [Internet]. [cited 2021 Feb 11]. Available from: <https://www.theverge.com/2017/7/25/16026236/adobe-flash-end-of-support-2020>
3. This is who pays taxes in South Africa [Internet]. [cited 2021 Feb 11]. Available from: <https://businesstech.co.za/news/finance/458708/this-is-who-pays-taxes-in-south-africa/>
4. SARS launches new eFiling browser to solve Flash problems [Internet]. [cited 2021 Feb 11]. Available from: <https://mybroadband.co.za/news/business/384210-sars-launches-new-efiling-browser-to-solve-flash-problems.html>

Alexander George Lain:

Hi Uzayr,

This is an interesting case, similar to the Queensland Health payroll issue that I posted about. In both case there seems to have been a slow response to replace software that had been deprecated, despite having several years over which to implement a solution. In the SARS case, the decision to

develop a browser based on an out of date version of Chromium as a way to work around recent builds of Flash which stopped working on 12 January 2021 is astounding. This solution also seems to only be available for Windows users, with no alternatives being provided to the Mac or Linux users that make up a non-zero portion of the South African tax paying population.

References

Salter, J. (2021) Flash is dead - but South Africa didn't get the memo. Available from: <https://arstechnica.com/gadgets/2021/02/flash-no-longer-works-in-browsers-so-south-africa-made-its-own-browser/>

Reflections:

This week revolved around learning to understand the core components of an information system, as well as a range of issues that can arise when using a system, and the steps which can be taken to solve or mitigate the problem. This included continuing with the Collaborative Discussion and providing responses to two of our peers' initial posts. Their post and my responses can be seen above. The week's reading also gave a brief and interesting overview of the history and evolution of information systems.

The first seminar of the module was also held this week, with the topic of En Route Automation Modernization.

AlexGeorgelain/Portfolio alexgeorgelain.github.io Alex Lain

UNIT 3: FUNDAMENTALS OF OBJECT-ORIENTED DESIGN

Learning Objectives:

- Identify the appropriate objects within a system.
- Develop an object-oriented design for a system.
- Correctly apply composition and inheritance where appropriate.

Required Reading:

- Philips, D. (2018) Python 3 Object-Oriented programming. 3rd ed. Packt Publishing. Chapters 1&5.

Collaborative Discussion: Information System Failure - Summary Post:

Name	Post
Alexander George Lain:	Over the past 3 weeks I've come to understand that information systems include the people who interact with them and not only the technology used in them. While doing my own research, and by reading through the research posted here by my peers, it has become clear that many information systems fail in some way. These are sometimes failures of technology, but often these are preceded by management or organisational failures.

In the case of the Queensland Health payroll, the failure was initially brought on by the need to replace the LATTICE system which was announced to reach end of life on June 30th 2008. This eventually led to a rushed rollout of a replacement system which ran over budget and cost hundreds of millions of dollars (AUD) in salary over payments. In a similar vein, the South African Revenue Services' failure (discussed by Uzay) to react in a timely manner after Adobe announced the discontinuation of Flash led a temporary solution being put in place which was unavailable on all operating systems. This temporary solution was a browser built on older technologies to enable the use of Flash after the discontinuation in January 2021 which lead to concerns over security.

In both of these cases the failure to react when a technology that was being used was discontinued led to rushed and less than satisfactory solutions being implemented. One way in which this can be avoided is for organisations to develop a framework for replacing parts of system that can be built into an Information System project that can then be implemented if any of the technologies being used are discontinued. This can be implemented as part of a larger risk management strategy which has been highlighted by several other members of this discussion as an area of information systems which needs improvement.

ALEX LAIN

AlexGeorgeLain/reportable

Required Reading

Alex Lain

MENU ☰

- Philips, D. (2018) Python 3 Object-Oriented programming. 3rd ed. Packt Publishing. Chapters 1&5.

Collaborative Discussion: Information System Failure - Summary Post:

Name	Post
Alexander George Lain:	<p>Over the past 3 weeks I've come to understand that information systems include the people who interact with them and not only the technology used in them. While doing my own research, and by reading through the research posted here by my peers, it has become clear that many information systems fail in some way. These are sometimes failures of technology, but often these are preceded by management or organisational failures.</p> <p>In the case of the Queensland Health payroll, the failure was initially brought on by the need to replace the LATTICE system which was announced to reach end of life on June 30th 2008. This eventually led to a rushed rollout of a replacement system which ran over budget and cost hundreds of millions of dollars (AUD) in salary over payments. In a similar vein, the South African Revenue Services' failure (discussed by Uzay) to react in a timely manner after Adobe announced the discontinuation of Flash led a temporary solution being put in place which was unavailable on all operating systems. This temporary solution was a browser built on older technologies to enable the use of Flash after the discontinuation in January 2021 which lead to concerns over security.</p> <p>In both of these cases the failure to react when a technology that was being used was discontinued led to rushed and less than satisfactory solutions being implemented. One way in which this can be avoided is for organisations to develop a framework for replacing parts of system that can be built into an Information System project that can then be implemented if any of the technologies being used are discontinued. This can be implemented as part of a larger risk management strategy which has been highlighted by several other members of this discussion as an area of information systems which needs improvement.</p>

Reflections:

This week's work introduced the concept of objects and object-oriented design. It was interesting to see what classes are and how to use them, having heard them discussed in interviews and articles in the past but never had any of my own experience using them as I only had a basic knowledge of programming concepts before I began this course. Seeing how an object can store its own attributes and then use these attributes when a method is called on it seemed particularly clean and useful when compared with storing many global variables and passing them into a function as I had done previously.

Unit 3: Fundamentals of Object-Oriented Design

Learning Objectives:

- Identify the appropriate objects within a system.
- Develop an object-oriented design for a system.
- Correctly apply composition and inheritance where appropriate.

Required Reading:

- Philips, D. (2018) *Python 3 Object-Oriented programming*. 3rd ed. Packt Publishing. Chapters 1&5.

Collaborative Discussion: Information System Failure - Summary Post:

Alexander George Lain:

Over the past 3 weeks I've come to understand that information systems include the people who interact with them and not only the technology used in them. While doing my own research, and by reading through the research posted here by my peers, it has become clear that many information systems fail in some way. These are sometimes failures of technology, but often these are preceded by management or organisational failures.

In the case of the Queensland Health payroll, the failure was initially brought on by the need to replace the LATTICE system which was announced to reach end of life on June 30th 2008. This eventually led to a rushed rollout of a replacement system which ran over budget and cost hundreds of millions of dollars (AUD) in salary over payments. In a similar vein, the South African Revenue Services' failure (discussed by Uzayr) to react in a timely manner after Adobe announced the discontinuation of Flash led a temporary solution being put in place which was unavailable on all operating systems. This temporary solution was a browser built on older technologies to enable the use of Flash after the discontinuation in January 2021 which lead to concerns over security.

In both of these cases the failure to react when a technology that was being used was discontinued led to rushed and less than satisfactory solutions being implemented. One way in which this can be avoided is for organisations to develop a framework for replacing parts of system that can be built into an Information System project that can then be implemented if any of the technologies being used are discontinued. This can be implemented as part of a larger risk management strategy which has been highlighted by several other members of this discussion as an area of information systems which needs improvement.

Reflections:

This week's work introduced the concept of objects and object-oriented design. It was interesting to see what classes are and how to use them, having heard them discussed in interviews and articles in the past but never had any of my own experience using them as I only had a basic knowledge of programming concepts before I began this course. Seeing how an object can store its own attributes and then use these attributes when a method is called on it seemed particularly clean and useful when compared with storing many global variables and passing them into a function as I had done previously.

ALEX LAIN

AlexGeorgeLain/portfolio alexgeorgelain.github.io Alex Lain

MENU ☰

UNIT 4: OBJECT-ORIENTED DEVELOPMENT AND PYTHON

Learning Objectives:

- Design object-oriented models of a system.
- Develop object-oriented software using the Python programming languages.

Required Reading:

- Philips, D. (2018) Python 3 Object-Oriented programming. 3rd ed. Packt Publishing. Chapters 2&3.

Reflections:

The work in this unit was to begin to write code using objects which were introduced in the previous unit. This was done in Codio through a series of exercises. The first of these exercises involved defining a "Circle" class with attributes for the centre (x, y) and radius. A circle object was then instantiated and passed into functions which determined whether a point, rectangle, or part of a rectangle lay inside the circle. The second exercise involved working with the datetime module and altering the attributes and methods on a pre-defined class to meet a new specification. The final set of exercises involved drawing a UML diagram for a given set of classes to demonstrate the parent/child relationship between them, and then writing a new method for a parent class which would instantiate a number of its child objects.

This was an interesting and challenging week of work. The exercise provided allowed me to put in to practice the theory which I had seen in the reading over the last two weeks. Working on these exercises also demonstrated a number of easy-to-make mistakes when using object oriented programming, such as giving `__init__()` variables a predefined value and then finding that that value has been altered after multiple instances of that class have been made. Learning to work with documentation while using the datetime module was also a new challenge, and I can see that being able to efficiently search documentation for packages and modules will be a vital skill to develop going forward.

The work this week culminated in the second seminar of this module, "Python Programming".

UNIT 5: UNDERSTANDING UML

Learning Objectives:

- Use UML to develop an object-oriented system design.
- Develop a sequence diagram to model the interactions between objects.
- Identify and use the correct elements of UML to design a system.

Unit 4: Object-Oriented Development and Python

Learning Objectives:

- Design object-oriented models of a system.
- Develop object-oriented software using the Python programming languages.

Required Reading:

- Philips, D. (2018) *Python 3 Object-Oriented programming*. 3rd ed. Packt Publishing. Chapters 2&3.

Reflections:

The work in this unit was to begin to write code using objects which were introduced in the previous unit. This was done in Codio through a series of exercises. The first of these exercises involved defining a “Circle” class with attributes for the centre (x, y) and radius. A circle object was then instantiated and passed into functions which determined whether a point, rectangle, or part of a rectangle lay inside the circle. The second exercise involved working with the datetime module and altering the attributes and methods on a pre-defined class to meet a new specification. The final set of exercises involved drawing a UML diagram for a given set of classes to demonstrate the parent/child relationship between them, and then writing a new method for a parent class which would instantiate a number of its child objects.

This was an interesting and challenging week of work. The exercise provided allowed me to put in to practice the theory which I had seen in the reading over the last two weeks. Working on these exercises also demonstrated a number of easy-to-make mistakes when using object oriented programming, such as giving `__init__()` variables a predefined value and then finding that that value has been altered after multiple instances of that class have been made. Learning to work with documentation while using the datetime module was also a new challenge, and I can see that being able to efficiently search documentation for packages and modules will be a vital skill to develop going forward.

The work this week culminated in the second seminar of this module, “Python Programming”.

AlexGeorgelain/alexgeorgelain.github.io Alex Lain

ALEX LAIN MENU ☰

The work this week culminated in the second seminar of this module, "Python Programming".

UNIT 5: UNDERSTANDING UML

Learning Objectives:

- Use UML to develop an object-oriented system design.
- Develop a sequence diagram to model the interactions between objects.
- Identify and use the correct elements of UML to design a system.

Required Reading:

- Fowler, M. (2013) *UML Distilled: a brief guide to the standard object modeling language*. 3rd ed. Boston, MA: Addison Wesley. Chapters 3, 4, 10 &11.
- Ambler, S. (2003) *Elements of UML Style*. Cambridge: Cambridge University Press. Chapters 1, 2, 3.

Reflections:

This week's work was an introduction to the Unified Modelling Language (UML). Learning to distinguish between the different UML relationships (association/inheritance/aggregation/composition) was a challenge initially when trying to visualise how different objects would interact with each other. This became easier after trying to develop my own diagram and also seeing examples of my peers' work.

UNIT 6: HANDS-ON WITH UML

Learning Objectives:

- Identify the potential classes in a system brief.
- Develop a set of UML documentation for a specific scenario.

Required Reading:

- Ambler, S. (2003) *Elements of UML Style*. Cambridge: Cambridge University Press. Chapters 4, 6, 8 & 9.
- Lucidchart (n.d.) *UML Class Diagram Tutorial*.
- Banas, D. (2012) *UML 2.0 Tutorial*.

Reflections:

Over the course of this week it was interesting to see how the different various UML diagrams are structured, and how

Unit 5: Understanding UML

Learning Objectives:

- Use UML to develop an object-oriented system design.
- Develop a sequence diagram to model the interactions between objects.
- Identify and use the correct elements of UML to design a system.

Required Reading:

- Fowler, M. (2013) *UML Distilled: a brief guide to the standard object modeling language*. 3rd ed. Boston, MA: Addison Wesley. Chapters 3, 4, 10 & 11.
- Ambler, S. (2003) *Elements of UML Style*. Cambridge: Cambridge University Press. Chapters 1, 2, 3.

Reflections:

This week's work was an introduction to the Unified Modelling Language (UML). Learning to distinguish between the different UML relationships (association/inheritance/aggregation/composition) was a challenge initially when trying to visualise how different objects would interact with each other. This became easier after trying to develop my own diagram and also seeing examples of my peers' work.

ALEX LAIN

AlexGeorgeLain/Portfolio alexgeorgelain.github.io Alex Lain

UNIT 6: HANDS-ON WITH UML

Learning Objectives:

- Identify the potential classes in a system brief.
- Develop a set of UML documentation for a specific scenario.

Required Reading:

- Ambler, S. (2003) Elements of UML Style. Cambridge: Cambridge University Press. Chapters 4, 6, 8 & 9.
- Lucidchart (n.d.) UML Class Diagram Tutorial.
- Banas, D. (2012) UML 2.0 Tutorial.

Reflections:

Over the course of this week it was interesting to see how the different various UML diagrams are structured, and how they are used in tandem to illustrate the design of a system. It was also quite interesting to see read that different elements of the UML guidelines often seem to conflict with each other, and that it is down to the individual organisations to choose the style which they prefer. Personally I would have expected more uniformity across all uses of UML than this seems to suggest.

This week contained the third seminar of the module, titled "Working with UML", during which the steps taken in drawing out class, sequence, and activity diagrams were talked through and demonstrated.

UNIT 7: DATABASE DESIGN

Learning Objectives:

- Develop the knowledge and skills to apply database design principles.
- Design a database that has been correctly normalised.

Required Reading:

- Connolly, T. & Beg, C. (2015) Database Systems: A Practical Approach to Design, Implementation, and Management. Global Edition. Edinburgh: Pearson. Chapters 1, 4 & 14.

Reflections:

Unit 6: Hands-on with UML

Learning Objectives:

- Identify the potential classes in a system brief.
- Develop a set of UML documentation for a specific scenario.

Required Reading:

- Ambler, S. (2003) *Elements of UML Style*. Cambridge: Cambridge University Press. Chapters 4, 6, 8 & 9.
- Lucidchart (n.d.) UML Class Diagram Tutorial.
- Banas, D. (2012) UML 2.0 Tutorial.

Reflections:

Over the course of this week it was interesting to see how the different various UML diagrams are structured, and how they are used in tandem to illustrate the design of a system. It was also quite interesting to see read that different elements of the UML guidelines often seem to conflict with each other, and that it is down to the individual organisations to choose the style which they prefer. Personally I would have expected more uniformity across all uses of UML than this seems to suggest.

This week contained the third seminar of the module, titled “Working with UML”, during which the steps taken in drawing out class, sequence, and activity diagrams where talked through and demonstrated.

ALEX LAIN

AlexGeorgelain/portfolio alexgeorgelain.github.io Alex Lain

MENU ☰

UNIT 7: DATABASE DESIGN

Learning Objectives:

- Develop the knowledge and skills to apply database design principles.
- Design a database that has been correctly normalised.

Required Reading:

- Connolly, T. & Beg, C. (2015) Database Systems: A Practical Approach to Design, Implementation, and Management. Global Edition. Edinburgh: Pearson. Chapters 1, 4 & 14.

Reflections:

The reading materials and lecturecast in this unit centred around database design and normalisation. The idea of first, second, and third normal forms was interesting to see formalised as it is a way I have often tried to structure data when working in spreadsheets before as it provides a clean and logical solution to presenting data. Additionally, seeing how draw out the database structure in a diagram, and then implement it using SQL was very informative and will no doubt be useful throughout the rest of the course and beyond.

This unit also ended with the submission of the first assignment of the module on the topic of system design. My submitted work can be viewed below.

The task was to design a class diagram for an online shopping platform to meet the specified requirements, an activity diagram for the process of a customer completing an order, and a state diagram highlighting the states of an order in the proposed system. This assignment brought together the work done in units 3 - 6, and was quite a challenge. The activity and state diagrams I did not find to be too difficult to draw out, but having read the feedback from my tutor, it is obvious that I could have been more specific in some of my choice of language.

The class diagram I did struggle with somewhat. Selecting the correct, or most suitable relationships, between each objects in my diagram took some thought. The main issue I had was with deciding when something should be its own object, and when something would be better suited simply as a combination of attributes and methods on another class. Having now began working on implementing this design for the assignment in unit 11, it has become clear to me that my initial design is over-complicated, and my implementation will contain fewer classes, and some of the removed classes such as Address and PaymentMethod instead be attributes inside of the Customer class.

> MID MODULE ASSIGNMENT: SYSTEM DESIGN

Unit 7: Database Design

Learning Objectives:

- Develop the knowledge and skills to apply database design principles.
- Design a database that has been correctly normalised.

Required Reading:

- Connolly, T. & Beg, C. (2015) *Database Systems: A Practical Approach to Design, Implementation, and Management*. Global Edition. Edinburgh: Pearson. Chapters 1, 4 & 14.

Reflections:

The reading materials and lecturecast in this unit centred around database design and normalisation. The idea of first, second, and third normal forms was interesting to see formalised as it is a way I have often tried to structure data when working in spreadsheets before as it provides a clean and logical solution to presenting data. Additionally, seeing how draw out the database structure in a diagram, and then implement it using SQL was very informative and will no doubt be useful throughout the rest of the course and beyond.

This unit also ended with the submission of the first assignment of the module on the topic of system design. My submitted work can be viewed below.

The task was to design a class diagram for an online shopping platform to meet the specified requirements, an activity diagram for the process of a customer completing an order, and a state diagram highlighting the states of an order in the proposed system. This assignment brought together the work done in units 3 - 6, and was quite a challenge. The activity and state diagrams I did not find to be too difficult to draw out, but having read the feedback from my tutor, it is obvious that I could have been more specific in some of my choice of language.

The class diagram I did struggle with somewhat. Selecting the correct, or most suitable relationships, between each objects in my diagram took some thought. The main issue I had was with deciding when something should be its own object, and when something would be better suited simply as a combination of attributes and methods on another class. Having now began working on implementing this design for the assignment in unit 11, it has become clear to me that my initial design is over-complicated, and my implementation will contain fewer classes, and some of the removed classes such as Address and PaymentMethod instead be attributes inside of the Customer class.

AlexGeorgeLain/portfolio

alexgeorgelain.github.io

Alex Lain

ALEX LAIN

MENU ☰

UNIT 8: HANDS-ON WITH DATABASE DESIGN

Learning Objectives:

- Develop a structure for a database for a given scenario.
- Create an entity relationship diagram to document your design.

Required Reading:

- Connolly, T. & Beg, C. (2015) Database Systems: A Practical Approach to Design, Implementation, and Management. Global Edition. Edinburgh: Pearson. Chapters 12, 13.
- Lucidchart (2017) What is an Entity Relationship Diagram?

Reflections:

The work this week was on databases. Namely, what to consider when deciding on the tables and columns in the database and how to develop an Entity Relationship Diagram (ERD). ERDs another another useful tool for visualising one of the structures in an information system. These, along with UML diagrams, will be crucial tools for me to use as I continue with this course. Too often I tend to leap into trying to implement a system without laying out a design beforehand and this is something that I will need to work on now that I have began to develop the necessary skills to do so.

The forth seminar of the module, titled "Database Design", was also put on this week. It walked through the design of a database step by step, from the initial scenario through to the final design, and how to go about constructing the ERD. Some of my peers also demonstrated their own work on normalisation from the previous week.

This unit also contained a discussion on alternatives to SQL. This discussion has highlighted a number of NoSQL alternatives such as the document-oriented MongoDB, and the graph database management system Neo4j. I thoroughly enjoyed catching up with this discussion and reading through my peers contributions, although I was unable to contribute myself.

UNIT 9: IMPLEMENTING DATABASE WITH SQL

Learning Objectives:

- Implement a database design using SQL.
- Create appropriate queries using SQL.
- Understand the access and security issues around SQL.

Unit 8: Hands-on with Database Design

Learning Objectives:

- Develop a structure for a database for a given scenario.
- Create an entity relationship diagram to document your design.

Required Reading:

- Connolly, T. & Beg, C. (2015) *Database Systems: A Practical Approach to Design, Implementation, and Management*. Global Edition. Edinburgh: Pearson. Chapters 12, 13.
- Lucidchart (2017) What is an Entity Relationship Diagram?

Reflections:

The work this week was on databases. Namely, what to consider when deciding on the tables and columns in the database and how to develop an Entity Relationship Diagram (ERD). ERDs another another useful tool for visualising one of the structures in an information system. These, along with UML diagrams, will be crucial tools for me to use as I continue with this course. Too often I tend to leap into trying to implement a system without laying out a design beforehand and this is something that I will need to work on now that I have began to develop the necessary skills to do so.

The forth seminar of the module, titled “Database Design”, was also put on this week. It walked through the design of a database step by step, from the initial scenario through to the final design, and how to go about constructing the ERD. Some of my peers also demonstrated their own work on normalisation from the previous week.

This unit also contained a discussion on alternatives to SQL. This discussion has highlighted a number of NoSQL alternatives such as the document-oriented MongoDB, and the graph database management system Neo4j. I thoroughly enjoyed catching up with this discussion and reading through my peers contributions, although I was unable to contribute myself.

AlexGeorgelain/alexgeorgelain.github.io Alex Lain ALEX LAIN MENU

UNIT 9: IMPLEMENTING DATABASE WITH SQL

Learning Objectives:

- Implement a database design using SQL.
- Create appropriate queries using SQL.
- Understand the access and security issues around SQL.

Required Reading:

- Han J., Halihong E., Guan L. & Jian D. (2011) 'Survey on NoSQL database'. 2011 6th International Conference on Pervasive Computing and Applications. Port Elizabeth, South Africa, 26-28 October. IEEE. 363-366. doi: 10.1109/ICPCA.2011.6106531.
- MySQL Training (2020) MySQL 8.0 Reference Manual.

Reflections:

The reading and lecturecast in this unit focused on SQL. It was a very useful revision and expansion on the SQL work covered in the first course module, Launching into Computer Science.

UNIT 10: WORKING WITH SQL

Learning Objectives:

- Implement a table structure based on a specification.
- Perform simple queries in order to extract information.
- Develop queries requiring multiple tables using either subqueries or joins.

Required Reading:

- Connolly, T. & Beg, C. (2015) Database Systems: A Practical Approach to Design, Implementation, and Management. Global Edition. Edinburgh: Pearson. Chapters 6, 7 & 8.

Reflections:

This unit continued on from last week's work with SQL. There were two sets of Codio exercises using SQL which were revision from the previous module, as well as a seminar titled "System Implementation Discussion". This seminar was very useful as it allowed for a good amount of questions to be asked and discussed about the upcoming assignment

Unit 9: Implementing Database with SQL

Learning Objectives:

- Implement a database design using SQL.
- Create appropriate queries using SQL.
- Understand the access and security issues around SQL.

Required Reading:

- Han J., Haihong E., Guan L. & Jian D. (2011) 'Survey on NoSQL database'. 2011 6th International Conference on Pervasive Computing and Applications. Port Elizabeth, South Africa, 26-28 October. IEEE. 363-366. doi: 10.1109/ICPCA.2011.6106531.
- MySQL Training (2020) MySQL 8.0 Reference Manual.

Reflections:

The reading and lecturecast in this unit focused on SQL. It was a very useful revision and expansion on the SQL work covered in the first course module, Launching into Computer Science.

AlexGeorgeLain/Portfolio alexgeorgelain.github.io Alex Lain

ALEX LAIN MENU ☰

UNIT 10: WORKING WITH SQL

Learning Objectives:

- Implement a table structure based on a specification.
- Perform simple queries in order to extract information.
- Develop queries requiring multiple tables using either subqueries or joins.

Required Reading:

- Connolly, T. & Beg, C. (2015) Database Systems: A Practical Approach to Design, Implementation, and Management. Global Edition. Edinburgh: Pearson. Chapters 6, 7 & 8.

Reflections:

This unit continued on from last week's work with SQL. There were two sets of Codio exercises using SQL which were revision from the previous module, as well as a seminar titled "System Implementation Discussion". This seminar was very useful as it allowed for a good amount of questions to be asked and discussed about the upcoming assignment which was due in unit 11.

UNIT 11: WEB DEVELOPMENT IN PYTHON

Learning Objectives:

- Implement a simple web server.
- Create dynamic templates using Jinja.
- Connect a database to a Flask application.
- Understand some of the security considerations around web applications.

Required Reading:

- Ginberg, M. (2017) The Flask Mega Tutorial.
- Pallets Projects (2010) Flask Tutorial.

Reflections:

This unit was an introduction to web development using the Python framework Flask. This included a Codio exercise which gave a short tutorial on how to initiate a flask web server and connect it to a database. The final seminar of the

Unit 10: Working with SQL

Learning Objectives:

- Implement a table structure based on a specification.
- Perform simple queries in order to extract information.
- Develop queries requiring multiple tables using either subqueries or joins.

Required Reading:

- Connolly, T. & Beg, C. (2015) *Database Systems: A Practical Approach to Design, Implementation, and Management*. Global Edition. Edinburgh: Pearson. Chapters 6, 7 & 8.

Reflections:

This unit continued on from last week's work with SQL. There were two sets of Codio exercises using SQL which were revision from the previous module, as well as a seminar titled "System Implementation Discussion". This seminar was very useful as it allowed for a good amount of questions to be asked and discussed about the upcoming assignment which was due in unit 11.

ALEX LAIN

AlexGeorgelain/Portfolio alexgeorgelain.github.io Alex Lain MENU

UNIT 11: WEB DEVELOPMENT IN PYTHON

Learning Objectives:

- Implement a simple web server.
- Create dynamic templates using Jinja.
- Connect a database to a Flask application.
- Understand some of the security considerations around web applications.

Required Reading:

- Ginberg, M. (2017) The Flask Mega Tutorial.
- Pallets Projects (2010) Flask Tutorial.

Reflections:

This unit was an introduction to web development using the Python framework Flask. This included a Codio exercise which gave a short tutorial on how to initiate a flask web server and connect it to a database. The final seminar of the module was also run during this week, which was used to run through the requirements for the final e-portfolio submission due by the end of unit 12.

The bulk of the work I completed this week was aimed towards completing the second assignment of the module, the Python implementation of the system design submitted in the Mid-Module Assignment in unit 7. My submitted work can be viewed below.

This assignment was a big learning experience for me as it was the first time I had ever written a full python application to implement a system of my own design. I chose to make this a command line application as it was the simplest way to demonstrate the required functionality. During the course of the development I chose to simplify my initial design slightly by reducing the number of classes slightly as described in my unit 7 reflections. I also made some additional changes which are described in the `readme.txt` file in the linked work.

I found this assignment to be a fantastic way for me to put into practice many of the things I have learned over the course of this module, and highlighted the areas that I will need to put in some additional work to improve my own knowledge. I look forward to being able to return to this assignment in the future and when I am hopefully able to implement this online shopping system in a more complete and satisfactory manner.

SYSTEM IMPLEMENTATION

Unit 11: Web Development in Python

Learning Objectives:

- Implement a simple web server.
- Create dynamic templates using Jinja.
- Connect a database to a Flask application.
- Understand some of the security considerations around web applications.

Required Reading:

- Ginberg, M. (2017) The Flask Mega Tutorial.
- Pallets Projects (2010) Flask Tutorial.

Reflections:

This unit was an introduction to web development using the Python framework Flask. This included a Codio exercise which gave a short tutorial on how to initiate a flask web server and connect it to a database. The final seminar of the module was also run during this week, which was used to run through the requirements for the final e-portfolio submission due by the end of unit 12.

The bulk of the work I completed this week was aimed towards completing the second assignment of the module, the Python implementation of the system design submitted in the Mid-Module Assignment in unit 7. My submitted work can be viewed below.

This assignment was a big learning experience for me as it was the first time I had ever written a full python application to implement a system of my own design. I chose to make this a command line application as it was the simplest way to demonstrate the required functionality. During the course of the development I chose to simplify my initial design slightly by reducing the number of classes slightly as described in my unit 7 reflections. I also made some additional changes which are described in the readme.txt file in the linked work.

I found this assignment to be a fantastic way for me to put into practice many of the things I have learned over the course of this module, and highlighted the areas that I will need to put in some additional work to improve my own knowledge. I look forward to being able to return to this assignment in the future and when I am hopefully able to implement this online shopping system in a more complete and satisfactory manner.

ALEX LAIN

AlexGeorgeLain/eporfolio

alexgeorgelain.github.io

Alex Lain

MENU ☰

SYSTEM IMPLEMENTATION

UNIT 12: THE FUTURE OF INFORMATION SYSTEMS

Learning Objectives:

- Understand the implications of emerging technologies on privacy in information systems.
- Engage with future trends in information systems.

Required Reading:

- Valacich, J. & Schneider, C. (2014) *Information Systems Today: Managing in the Digital World*. 6th ed. Boston, MA: Pearson. Chapters 1 and 10.
- Bourgeois, D. (2014) Welcome to Information Systems for Business and Beyond. Saylor Academy. Chapters 11 - 13.
- Jablonski, J. & Robak, S. (2019) 'Information Systems Development and Usage with Consideration of Privacy and Cyber Security Aspects', in: Ganzha, M., Maciaszek, L. & Paprzycki, M. (eds) *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems ACSIS*. 18: 547–554. doi: 10.15439/2019F261.
- Brooks, N., Greer, T. & Morris, S. (2018) Information systems security job advertisement analysis: Skills review and implications for information systems curriculum. *Journal of Education for Business* 93(5): 213–221. doi: 10.1080/08832323.2018.1446893.

Final Reflections:

» [OOIS BLOG POST](#)

» [E-PORTFOLIO SUBMISSION](#)

© Untitled Inc. All rights reserved. | Design: HTML5 UP

Unit 12: The Future of Information Systems

Learning Objectives:

- Understand the implications of emerging technologies on privacy in information systems.
- Engage with future trends in information systems.

Required Reading:

- Valacich, J. & Schneider, C. (2014) *Information Systems Today: Managing in the Digital World*. 6th ed. Boston, MA: Pearson. *Chapters 1 and 10*.
- Bourgeois, D. (2014) *Welcome to Information Systems for Business and Beyond*. Saylor Academy. *Chapters 11 - 13*.
- Jablonski, J. & Robak, S. (2019) 'Information Systems Development and Usage with Consideration of Privacy and Cyber Security Aspects', in: Ganzha, M., Maciaszek, L. & Paprzycki, M. (eds) Proceedings of the 2019 Federated Conference on Computer Science and Information Systems ACSIS. 18: 547–554. doi: 10.15439/2019F261.
- Brooks, N., Greer, T. & Morris, S. (2018) Information systems security job advertisement analysis: Skills review and implications for information systems curriculum. *Journal of Education for Business* 93(5): 213–221. doi: 10.1080/08832323.2018.1446893.

Final Reflections:

The final unit of this module consisted of writing a blog post about a future trend in information systems, and the submission of this e-portfolio, how it looked as of 19 April 2021. Both of the submitted pieces of work can be viewed at the links below.

This module has been a steep learning curve for me. Over the past 12 weeks I have completed the work listed above, much of which was brand new to me when I begin the module. The two main things that completing this module has highlighted to me are the importance of having an ongoing project, and the importance of my personal time management.

While working on the system implementation assignment in unit 11, it became very apparent that moving forward it will benefit me greatly to have my own ongoing project that I can use to test out the new concepts introduced during the course of a module. While the exercises provided in Codio provide a good introduction, writing the system from the ground up was a fantastic learning experience, I think my submission ultimately left something to be desired as I was still becoming comfortable with the process of writing the implementation as I was doing it. If I had began the assignment having already familiarised myself with the process, I could have produced a more fleshed out submission that didn't merely meet the required specification.

How I managed my time over the last 12 weeks has also been less than optimal. Rather than adhering to a regular work schedule, I completed much of this module in bursts which fit between other commitments. This ultimately led to me having to remind myself of previously covered work, rather than allowing my knowledge to progressively build week after week. Moving forward I am very conscious that I will need to spread my course workload much more evenly to ensure I can complete the work required to an acceptable standard and in a timely fashion.

Finally, having now produced e-portfolio contributions for a complete module, I intend to use the break between modules to become more comfortable with HTML and CSS so that I can produce a website that is more suited to my needs, rather than trying to fit my work into a pre-existing template. Having my own design to maintain and improve will also help to keep me more engaged with the e-portfolio process, which I have struggled with at times.