Quantum Computing Algorithmic Design

Applications in Encryption

Samarjeet Saluja

Year 12 Innovative Project

Contents

History of Computing and Programming (Description of Trend / Analysis of Design)

The word, ‘computer’, was first used by Richard Brathwait in 1963, not in reference to a machine at all, but to a job title. In the pre-digital era, a ‘computer’ was a person who did calculations for predominantly military purposes, such as accurately firing artillery shells . This title persisted till the late 19th century, when the meaning slowly shifted to the modern definition of ‘devices capable of performing a sequence of operations according to variable set of procedural instructions’. Purpose of computing was, as is with any machine, to make work done by and for society more efficient and life more pleasant.

English polymath, Charles Babbage, in his paper, ‘Note on the Application of Machinery to the Computation of Astronomical and Mathematical Tables’, in year 1822, proposed a new mechanical device called ‘The Difference Engine’. This was a complex machine that could approximate polynomials (mathematical equations describing relationships between several variables), theoretically being able to greatly increase the efficiency of calculations such as that of the artillery shell firing. He later proposed an even more complex machine, a general purpose computer called the “Analytical Engine”. English mathematician, Ada Lovelace, wrote hypothetical programs for the ‘Analytical Engine’, giving her the title of the first programmer. Though it was never constructed, this idea of an automatic computer inspired the first generation of computer scientists. Thus, Babbage is often considered the father of computing.

Inspired by Babbage’s designed, one of the first general-purpose computers was the ‘Harvard Mark I’, built for military purposes, during WW2. Using mechanical relays as switches, this computer was comparatively slow and prone to wear and tear. Since then, computing power has grown exponentially, allowing computers to get smaller and more powerful simultaneously. The process of developing new switches, from the mechanical relays mentioned above, to vacuum tubes to transistors, has allowed for this exponential growth in power and reduction in size. This idea is represented in Moore’s Law which states that, “the number of transistors in a dense integrated circuit doubles about every two years”.

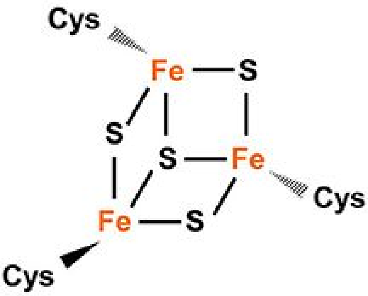
Modern computers, consist of simple mechanisms to represent and process data, as well as control mechanisms. The transistor is the simplest form of data processing of computers, being a switch that controls the flow of electrical signals (bits - smallest unit of information - 0 or 1) through it. However, with the transistor approaching the size of an atom, this process is process is nearing its physical limits, due to the laws of quantum mechanics. As transistors shrink further, electrons may transfer themselves through a closed transistor through the process of quantum tunnelling, hence making it impossible for this to increase the number of transistors in IC’s. This violation of Moore’s Law indicates that a physical barrier for the increase in computing power is being approached. As a solution to this problem, physicists are attempting to harness these quantum properties for an advantages - to develop quantum computers.

Introduction to Quantum Computation

As a means of simulating the evolution of quantum systems over time, the idea of a quantum computer was first proposed by Richard Feynman in 1981. Over the next decade, research allowed for the development of a theoretical framework of a quantum computer, until in 1994, there was a major discovery regarding its practical applications, providing further incentive for one's physical construction. Peter Shor had developed an quantum algorithm for the purposes of quickly factoring large integers, a problem considered previously impossible in classical computers. This idea of quantum supremacy acted as a driving force for the development of quantum computer over the next few decades.

Classical computers have revolutionised the lifestyle of humanity, empowering the digital age, by connecting the world through the internet, enabling faster scientific progress through faster calculations to even automation of laborious procedures following the development of Artificial Intelligence. Yet, there still remain limitations to classical computers, simply being unable to conduct certain tasks. For these certain situations, quantum computers outperform even the most powerful classical computers. Today, after multiple decades of research following the discovery of Shor’s algorithm, quantum computers are performing calculations that have never done before.

Two of the major tasks that breach the limits of classical computers include problems to do with optimization and simulation of quantum systems. Optimisation problems involve finding the best possible solutions among a range of many possible solutions. Whereas a simple application of this is calculation is permutations around objects such as a table, a more applicable example of an optimisation problem is pathfinding (shortest route). Although classical computer have found solutions to this problem over a small range of options, quantum computers have the capability to be exponentially faster with an exponentially larger number of ‘possible routes. In addition, scientific research has continuously been impeded due to the lack of computing power to simulate quantum systems such as the workings of an organic molecules. In order to simulate the workings of a larger nitrogenase enzyme, IBM attempted to simulate the Iron-Sulfide clusters within. However, the largest cluster that could be simulated is the molecule shown above, with a mere 8 atoms. The reason for this is the vast amount if information to do with each electron and its interactions (probability wave distributions, spin, orbital energy states). The commonality in these two problems of optimisation and simulation is the idea of exponential scaling, where the addition of a single object greatly increases the data needed to represent the system.



The need for quantum computers arises here - their workings and hardware allowing to store and manipulate exponentially greater amounts of information in the quantum equivalent of the classical system. Due to this difference in internals, quantum computers are able to solve such problems much faster than a classical computer, even at such an early phase in their development.

Workings of a Quantum Computer

Quantum computing is one of the leading applications of quantum physics, having the capability to solve the world's most complex problems. The great difference in the methods of operation of quantum computers allow it to to perform calculations deemed impossible in the field of quantum computing. In order to do this, these computers exploit properties of quantum mechanics, in particular spin, interference and entanglement, to make calculations much more efficient.

In classical computers, binary digits (bits) are the smallest units of information, representing the presence or absence of an electrical signal (0 or 1). Quantum computers use quantum binary digits (qubits), bits that exploit quantum properties of their physical existence. A qubit can be any two-level quantum system, such as an electron in a magnetic field, or a single photon. In this system, 1 and 0 are this main possible states, corresponding to the photon's horizontal or vertical polarisation. In the quantum world, the qubit can be in any proportion of both states at once - a superposition of these states. However, as soon as you test its value, by sending the photon to a filter, it has to decide whether to be horizontally or vertically polarised. The qubit is in a superposition of probabilities of 0 and 1, until the instant you measure it, when it collapses into one of the definite states. This property of superposition allow qubits to represent an exponentially greater amount of data than classical bits. For example, four classical bits can be in one of 2^4 different configurations at a time. Four qubits in superposition, however, can be in all of these 16 configurations at the same time. This number grows exponentially with each qubit, such that 20 can store over 1 000 000 values in parallel.

Quantum entanglement is another unique property of qubits, allowing them to react to a change in the other's state instantaneously, now matter how far apart. This allows for deducing the properties of a bits partner, by simply measuring the state of one entangled qubit, speeding up processing time. Furthermore, this benefits the way quantum logic gates work, making it so that less qubits have to manipulated to obtain the same net result. Whereas a normal logic gate takes in a simple set of inputs and produces one definite input, a quantum gate manipulates an input of superpositions, manipulates probabilities and produces another superposition as an output, later collapsing into an actual sequence of 0's and 1's. This allows quantum computers to complete an entire lot of calculations that are possible with the setup, at the same time. This is exponentially more efficient than a classical computer.

"Quantum decoherence" is a concept used in the traditional "Copenhagen interpretation" of quantum mechanics, de-emphasizing the role of the observer. In quantum mechanics, particles such as electrons are described by a wave function, a mathematical description of the probabilistic nature of the wave. Coherence is a fundamental property of quantum mechanics, necessary for the functioning of quantum computers. However, when a quantum system is not perfectly isolated, coherence will decay over time. This process is called quantum decoherence. Due to issues such as this, for the foreseeable future, quantum computers are unlikely to replace your everyday desktop PC, but will bring major progress in research in universities and large organisations.

Further Innovation - Shor’s Algorithm (Products Relation to Trend)

Workings of RSA

The fundamental definition of a composite number is having the ability to be divided into equal numbers greater than one to add to the larger number (eg. 28 = 7+7+7+7). A prime number on the other hand does not have this property (eg. 13 = 6+7). An integral component of RSA cryptography is ‘The Fundamental Theorem of Arithmetic’, describing that every number has a unique set of prime numbers that multiply to give a larger number. No two numbers would have the same set of prime factors.

With the advent of computer networking, cryptography was becoming necessary as a means of cybersecurity. However, there was a problem, as at the time, encryption required to parties to share a secret key (usually a number), which was impossible due to the nature of long-distance computer networking. Simply sending the key prior to the transmission would be redundant as it could be just as easily intercepted as any other transmitted data. In order to solve this problem, a one-way mathematical function was devised, allowing for data encryption through a decryption (private) key that was separate to the encryption (public) key (performing inverse functions to one another).

Without certain information kept private by each party, it would be extremely difficult (almost impossible) to decrypt the mixture of private and public keys that was transmitted. This one-way function was known as modular-exponentiation. The function to the left: m^e*mod(N) = c, was used, where the ’m’ is a message that has been numerically encoded, ‘e’ is a public exponent, and N is a random number (also transmitted), transmitting the value of ‘c’. Without having any information other information, calculating m^e*mod(N) = c, where e, N, and c are known is extremely difficult, whereas calculating c knowing the other pronumerals is possible.

In RSA, the mathematical public key (?^e*mod(N)) is transmitted from person A to person B, where person B inserts their own value for m, transmitting the result from the modulus of equation. However, to calculate the original message (m), a second one-way function is needed for a decryption key (d) to undo the modular equation (m^(ed)*mod(N) = c). For this second one-way function, prime-factorisation was introduced, being a fundamentally difficult problem to do, having an exponential time complexity. Through the use of prime factorisation in relation with Euler’s Totient Function (Phi), as well as Euler’s Theorem, relating the phi function to modular exponentiation, a method of calculating a decryption key was made apparent. The value of the decryption key (d) = (k*Φ(N)+1)/e, where k is an integer used in the relation between prime factorisation and modular exponentiation. Thus, the decryption is easy to do, provided that the prime factorisation is known, which would remain unknown to people intercepting the key exchange. This encryption algorithm is the most used in networking and communications, making any potential solution to the one-way functions involved a threat to modern-day cybersecurity.

Quantum Computers and Decryption

Another famous application of quantum computing is the breaking of encryptions, rendering cyber-security encryption algorithms useless. Current networking data is kept secure by the RSA encryptions system, which involves presenting a public key to encode messages that can only be decoded by the user. However, this public key can actually be used to calculate the private key through the process of prime factorisation of extremely large numbers. However, the necessary mathematical calculations required to do this would be almost impossible on even the most powerful classical computers, required years of computation. With the advent of Shor's algorithm, quantum computers could theoretically break RSA encryption, through prime factorisation, theoretically rendering modern-day cybersecurity useless. With further development to quantum computation, this algorithm will be used by intelligence agencies with access to quantum computers, to further their investigations, breaching consumer privacy.

A major component of RSA is relying on the fact that there are no efficient algorithms to factorise numbers into 2 prime factors. In the future, the architecture of quantum computing will allow it factor large numbers exponentially faster than classical computers. My product aims to highlight the difference in computation capability between quantum and classical computers for the purpose of prime factorisation. It is a demonstration of the practical applications of 'Quantum Computation', and the role it plays in future cybersecurity. My product runs Shor's algorithm on a simulated quantum computer, demonstrating the difference in capability between classical and quantum computers. However, due to the early stage in research, and the limitations of the simulation, classical computing is shown to be quicker at factorising in this case. However, in a case of a key which is 100's of digits long, quantum computers, after further development, will outperform classical computers every time.

Due to the complex nature of RSA, a modified version is demonstrated, where a prime number is assumed to have been transferred through the 'Diffie-Hellman Exchange'. Multiplications of this common prime factor and second private prime factor are transferred in this modified form of encryption. This program demonstrates how a classical and quantum computer would crack this encryption, through prime factorisation of integers. Below is the explanation of the components of the modified encryption scheme:

--

'Public' Prime Factor - Has been already privately exchanged between computers through the 'Diffie-Hellman Exchange' (modified to only transfer a prime key)

Key 1 - A multiplication between User 1's private key (also prime) and the common prime factor

Key 2 - A multiplication between User 2's private key (also prime) and the common prime factor

Final Key - A multiplication of the 2 private keys.

--

Since the 2 computers exchanging information already know the prime factors, private keys can easily be determined through division, allowing to calculate the final key. However, in order for an external entity, intercepting this exchange, to calculate the private keys would be through prime factorisation, in turn allowing it to achieve the final key. My program demonstrates the prime factorisation an external entity would have to do, on a classical and simulated quantum system.

Social and Ethical Issues

Social Issues

Lack of access of quantum computers to the public is a social issue of concern. Although at this early stage, it is understandable that there is a lack of access, in the future when the technology has developed enough to have practical applications in the world. Current and future lack of widespread access would restrict availability to economically advantages individuals and large institutions in wealthier countries. This would not only create create an international, but disparities with the nation due to the unequal opportunity. For example, militaries making use quantum computers would have a social advantage over less economically well-off countries. Furthermore, the right to use of these as education tools is also limited, due to expense and social connections. To solve this problem, quantum computers should be made available for student use at every major university, providing access to as many people as possible. For the international accessibility problem, major quantum computers must be made available online for free use by anyone, internationally.

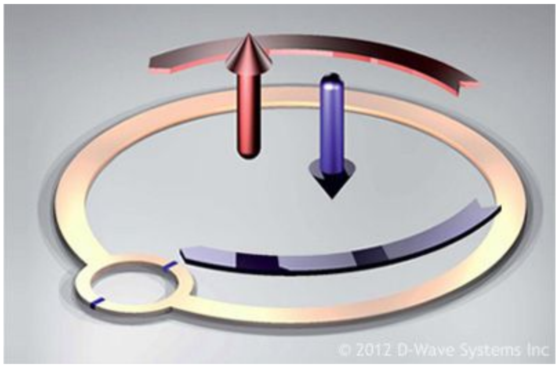
Ethical Issues

The ability of quantum computers to completely break the current RSA encryption protocol, commonly used over the internet will have serious ethical issues. Even currently, the NSA has been known to conduct mass-surveillance, breaching the privacy of many people. The advent of the break of RSA will make it even easier for anyone with access to quantum computers to access any information exchanged across networks causing privacy to be completely diminished. Furthermore, this brings us back to the issue of accessibility, where certain people have access to the technology, hence the ability to intercept, decrypt and read anyones information. However, running Shor's algorithm to break current cryptography would require thousands or even millions of qubits, depending on diminishing the effects of quantum decoherence and accuracy of qubits. When this happens however, military organisations, governments and even major companies would not have secure data anymore, completely rendering all forms of privacy and security obsolete. However, as a solution to this, a new form of encryption must be developed, that involves not prime factorisation, but some other one way function for the exchange of keys. However mathematically difficult to do, it must be done to reinstate cybersecurity in the future.

Relation to Hardware

Within the Quantum Processing Unit

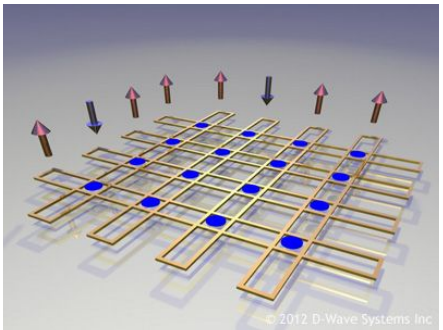
Two forms of quantum computation exist; major ones including; quantum annealing and adiabatic quantum computation. Quantum annealing computation makes use of the intrinsic effects of quantum physics, allowing to solve problems relating to optimization and probabilistic sampling problems. The reason such problems can be solved using physics, is the ability to have them reframed as energy minimisation problems. A fundamental aspect of physics is that everything aims to find it minimum energy state - meaning such problems are easy to solve, lowest energy state being the best solution. Here lies the fundamental definition of quantum annealing - solving problem by finding the lowest energy state of a system. This brings forth many applications in the field of machine learning, where attempting to build a probabilistic representation of the world is greatly benefitted of this method of finding the lowest energy state - finding the best neural network model. Below is described the hardware use in the quantum annealing approach to quantum computation.



Modern digital computers are encoded by adjusting and monitoring voltages, present on transistors inside integrated circuits. Each transistor is addressed by bus, either set to a state of 0 (low voltage) or 1 (high voltage). The basic building block of a quantum computer is a superconducting qubit or SQUID (Superconducting QUantum Interference Device). Electrons are the fundamental basis of having a superposition of states, as in reality they are not a particle, but a probabilistic wave function - being subject to interference patterns. The properties of the material used to construct such superconducting qubits is Niobium, supporting the electrons wave like structure to be used for computation. When this metal is cooled down, it becomes what is known as a superconductor, and it starts to exhibit quantum mechanical effects.

Whereas a regular transistor, based off silicon and electrical signals, allows for 2 states to be encoded (1 or 0), the superconducting qubit structure, encoded through magnetic fields, allows for a superposition between the state of spin up (+1) and spin down (-1). Thus, until the state is measured, the quantum computer can theoretically be at infinite superpositions between these states, though not possible due to the physical restrictions of the hardware.

To create multi-qubit processors, the processors are connected together, through the use of superconducting loops called couplers. By putting many qubits and couplers together, it is possible to build a programmable quantum processing unit (usually as a wafer), to run previously theoretical algorithms. The loop in the diagram above, is stretched out into long rectangles, points



where the rectangles cross being schematically represented as blue dots.

Outside the Quantum Processing Unit

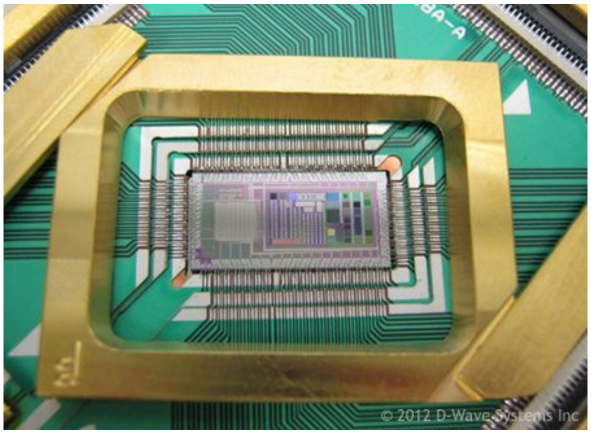
To build the quantum computer, one of these QPUs is selected from the wafer, and placed in the center of the QPU packaging system, as shown in Figure 6. This image shows the QPU area open, just after it has been wire bonded to connect it to the signal lines. It is possible to see the signal lines on the surrounding printed circuit board. There are far fewer incoming lines than there are programmable elements on the QPU, which is made possible by additional circuitry - in the form of demultiplexers and signal routing and addressing - all implemented in superconducting logic circuitry on the chip.

To construct a fully functioning quantum computed, a QPU is placed in the centre of a QPU packaging system, connecting it to signal lines leading to external components. There are far fewer incoming lines than there are programmable elements on the QPU, which is made possible by additional circuitry - in the form of demultiplexers and signal routing and addressing - all implemented in superconducting logic circuitry on the chip.

An extremely low temperature is required due to the nature of qubits, as differentiating between a spin up or spin down state is only possible when the energy difference between the two states is less than the surrounding thermal energy of the environment. Hence, a reduction of temperature of the computing environment to below approximately 80mK is required for the processor to function. Further, parts of the IO system, comprising of roughly 10 KG of material, is cooled also cooled to this temperature, causing most of the physical volume quantum computer systems to be part of the dilution refrigerator. To reach the near-absolute zero temperatures at which the system operates, the refrigerators use liquid helium as a coolant. The system commonly used allows for liquid helium to reside inside a closed-cycle system, allowing it to be recycled and recondensed.

As discussed above, quantum computers are heavily subject to quantum decoherence, collapsing all superpositions and calculations, with interfered with or accidentally observed. Hence, to prevent stray magnetic fields, there exists a magnetic shielding system, achieving fields less than 1 nanoTesla in three dimensions surrounding the QPU. Integrated on the QPUs are magnetic sensors that measure the ambient field such that inverse magnetic fields are applied that zero the field at these sensors, through destructive interference. Essentially, this is achieved through a large Faraday Cage surrounding the components of the computer.

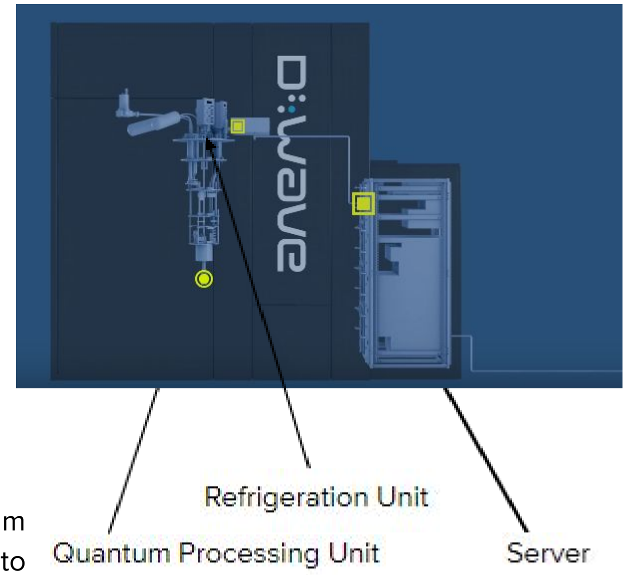
The many components described above are the major parts comprising a quantum computer. Over the next decade, with further research and development into quantum computing, the power expecting to double every year, through an increase in qubits and better efficiency.



D-Wave Systems D-Wave 2X - Real Life Example

Founded in 1999, D-Wave Systems is the world's first quantum computing company, delivering quantum computing systems and software. They provide systems for organizations and institutions including Lockheed Martin, Google, and NASA. In 2015, D-Wave's 'D-Wave 2X' was released, the most advanced quantum computer yet to be released with a 1000 qubit quantum-annealing processor. Running at an even colder temperature than before at 15 mK, the quantum annealing algorithm better stays at low energy states, improving solution accuracy. There is also an increased precision of control circuitry and a large reduction in qubit noise. The new quantum processor is the most complex superconductor integrated circuit to be successfully used in production systems. Being used in the NASA Research Centre California, this computer is being used for the purposes of machine learning and optimisation by Google, NASA and USRA. However, these are also accessible at D-Wave's headquarters, available for use at request.

The infrastructure of this computer makes use of a Quantum Processing Unit, present in a low temperature and magnetic environment. Within the system, there is; a quantum server, allowing users to access and program the processor, a cooling system to maintain the low temperature, and multiple layers of shielding creating a low-noise, magnetic environment from the processor, reducing the effects of quantum decoherence. The quantum server allows users from across the world to interact with QPU, by receiving commands from the user and converting it into the machine language of the quantum processor for it to run. Once executed, the signal is then sent back to the server, converted back into classical machine language, and sent back to the user.



Storing information through the manipulation of magnetic fields, thus meaning the quantum processor is extremely sensitive to noise, to combat which there is 16 layers of shielding against magnetic waves, between the processor and the environment. Within the outermost shielding, there is also radiation shielding to further protect the processor.

Effect on Commerce / Economics

With the approaching end of Moore's Law, technological development is to see a significant decrease as we approach the physical limit of how small transistors can get. Having made a transistor 1 atom large, classical computer will reach there limits, only having minute upgrades from then on. With no incentive for consumers to purchase electronic goods - the largest industry in the world, there would be a drastic impact on the economy, humanity hitting a global recession. A halt in the upgrade in technology would cause an economic stand still, as consumers would stop buying new technology that has minimal upgrades. In order to combat this, a continuation of electronics development required an architectural rehaul. That is where quantum computers come in, with major companies investing millions of dollars already, quantum computing will be the area of technological development post-digital era, catapulting humanity into a possible quantum era.

The increase in jobs for maintenance, programming and research would reduce the unemployment rate, positively impacting the economy. Society will become much more efficient, possibly supporting larger populations, and reducing time for calculations. Just as Herman Hollerith's tabulating machine, enabled greater efficiency in the 1890 US Census, quantum computing will exponentially increase efficiency, possibly solving world problems and increasing production of resources. However, in the long run, there may be an increase in unemployment rate, as quantum computing increases the rate at artificial intelligence develops, training neural network models in exponentially faster.

Quantum computing will also result in an increased efficiency in resource allocation, machine learning, and development of new drugs and materials; reducing waste, freeing up time, increasing profit, curing diseases. Although these seem to be positive factors, this technology will not be widely accessible. This could lead to a more unequal distribution of power and wealth: between America and the rest of the world, and between a few big companies and the rest of society. Major companies having already have start to invest heavily in quantum computing, the majority of which are American. These companies are already patenting many of the ideas, including some based on freely-available academic research. This could lead to American businesses monopolizing commercial quantum computing, similar to the way Silicon Valley dominates much of current classical computing. This would weaken competition in our free-market economy, shifting power from individuals and small businesses to one or a few large companies, hence, leading to a growing inequality in society.

The global market for financial assets (stocks, bonds, securities) is estimated to be $294 trillion, including the $69 trillion stock market. Whereas with current technology, problems such as; dynamic portfolio optimisation and regression analysis are classically time consuming and expensive, people with access to quantum computation will have a commercial advantage. For example, in the case of an asset manager, every time they rebalance their portfolio, their investors lose money. With the computational power of quantum computers, the asset manager can decide when to rebalance their portfolio less frequently.

<u>IPO Diagram</u>

Classical Encryption

Input	Process	Output
Click Encrypt Button + Key_1 + Key_2 + Text_Input	Ensures valid input is present in all fields. - *if (string.IsNullOrEmpty(user_text_input.Text) || string.IsNullOrEmpty(NUM1.Text) || string.IsNullOrEmpty(NUM2.Text))* Computes final key, using input of two keys. - *Each key is factored into 2 prime factors* - *The common prime factor is disregarded.* - *The unique prime factors of each key, are the private key of each 'computer'.* - *These private keys are multiplied to give the final key.* Encrypts the Inputted Text, According to Final Key - *Shifts each char in array by the 'final_key%36', due to the 36 characters in the array* Displays Encrypted Text and Final Key	Outputs the Encrypted Text on Screen, as well as Final Key
Click Decrypt Button + Key_1 + Key_2 + Text_Input	Ensures valid input is present in all fields. - *if (string.IsNullOrEmpty(user_text_input.Text) || string.IsNullOrEmpty(NUM1.Text) || string.IsNullOrEmpty(NUM2.Text))* Computes final key, using input of two keys. - *Each key is factored into 2 prime factors* - *The common prime factor is disregarded.* - *The unique prime factors of each key, are the private key of each 'computer'.* - *These private keys are multiplied to give the final key.* Decrypts the Inputted Text, According to Final Key - *Shifts each char in array by the '-final_key%36', due to the 36 characters in the array* Displays Decrypted Text and Final Key	Outputs the Decrypted Text on Screen, as well as Final Key

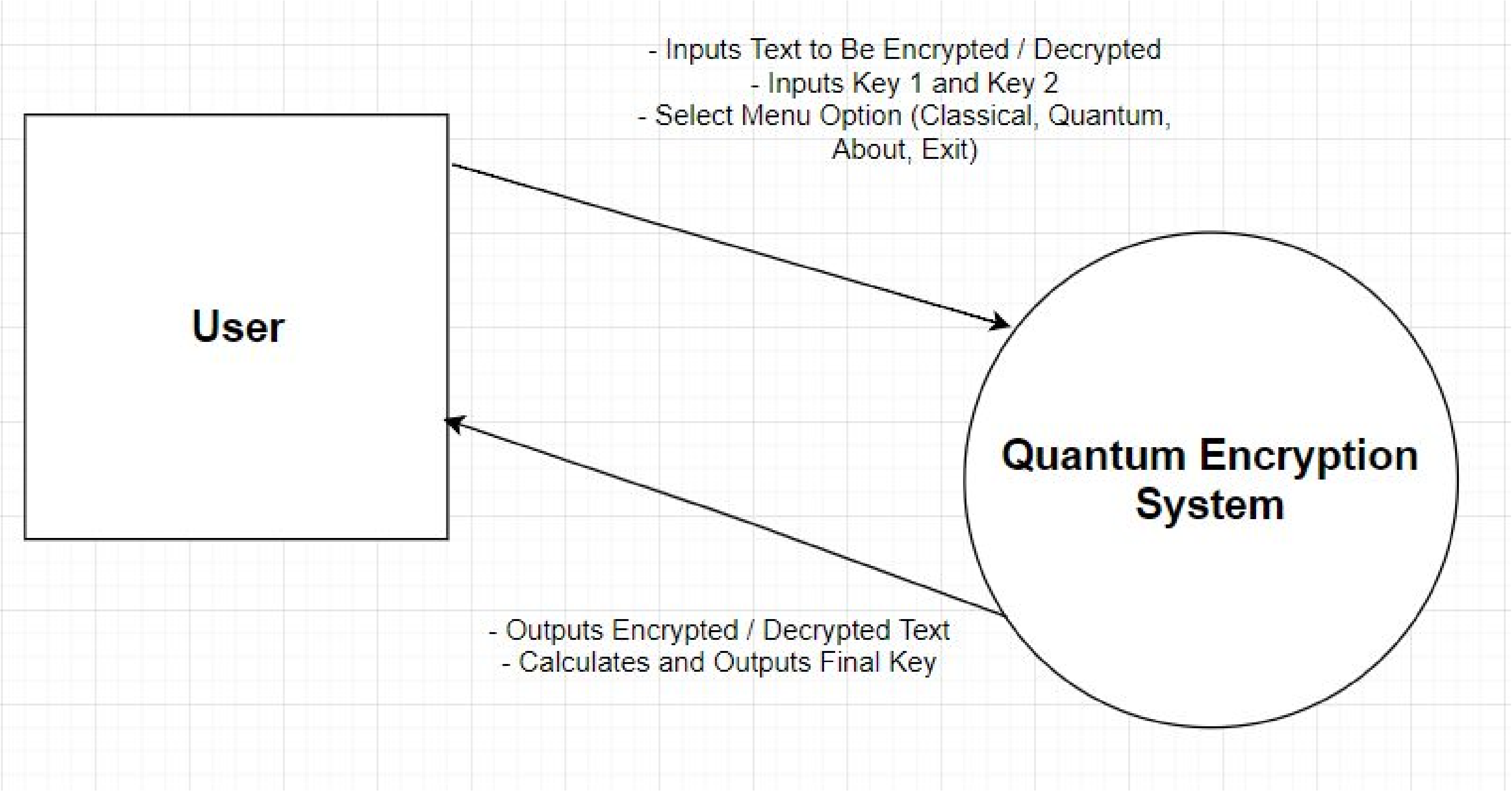
Click Quantum Button	Changes Prime Factorisation Method to Quantum Changes Screen to Quantum Version - *Changes Selected Button to Quantum* - *'Change Key' Button Appears* - *Key_1 / Key_2 Text-Boxes Disappear*	Displays Quantum Screen
About Button	Changes Screen - *Hides All Boxes* - *Shows Text*	Changes to About Screen
Exit Button	Exits Program	Closes Program
Change Key Button (Quantum Only)	Displays Message ("Enter the keys by running the following program.") Opens Q# Program to be Run	Shows Message and Opens Q# Factorisation Program
Click Encrypt Button (Quantum)	Reads Factors from External Database, ensuring that they Exist Computes final key, using input of two keys. - *The common prime factor is disregarded.* - *The unique prime factors of each key, are the private key of each 'computer'.* - *These private keys are multiplied to give the final key.* Encrypts the Inputted Text, According to Final Key - *Shifts each char in array by the 'final_key%36', due to the 36 characters in the array* Displays Encrypted Text, Key_1, Key_2 and Final Key	Outputs the Encrypted Text on Screen, as well as Encrypted Text, Key_1, Key_2 and Final Key

Click Decrypt Button (Quantum)	Reads Factors from External Database, ensuring that they Exist Computes final key, using input of two keys. - *The common prime factor is disregarded.* - *The unique prime factors of each key, are the private key of each ‘computer’.* - *These private keys are multiplied to give the final key.* Encrypts the Inputted Text, According to Final Key - *Shifts each char in array by the ‘-final_key%36’, due to the 36 characters in the array* Displays Decrypted Text, Key_1, Key_2 and Final Key	Outputs the Decrypted Text on Screen, as well as Encrypted Text, Key_1, Key_2 and Final Key

Quantum Program

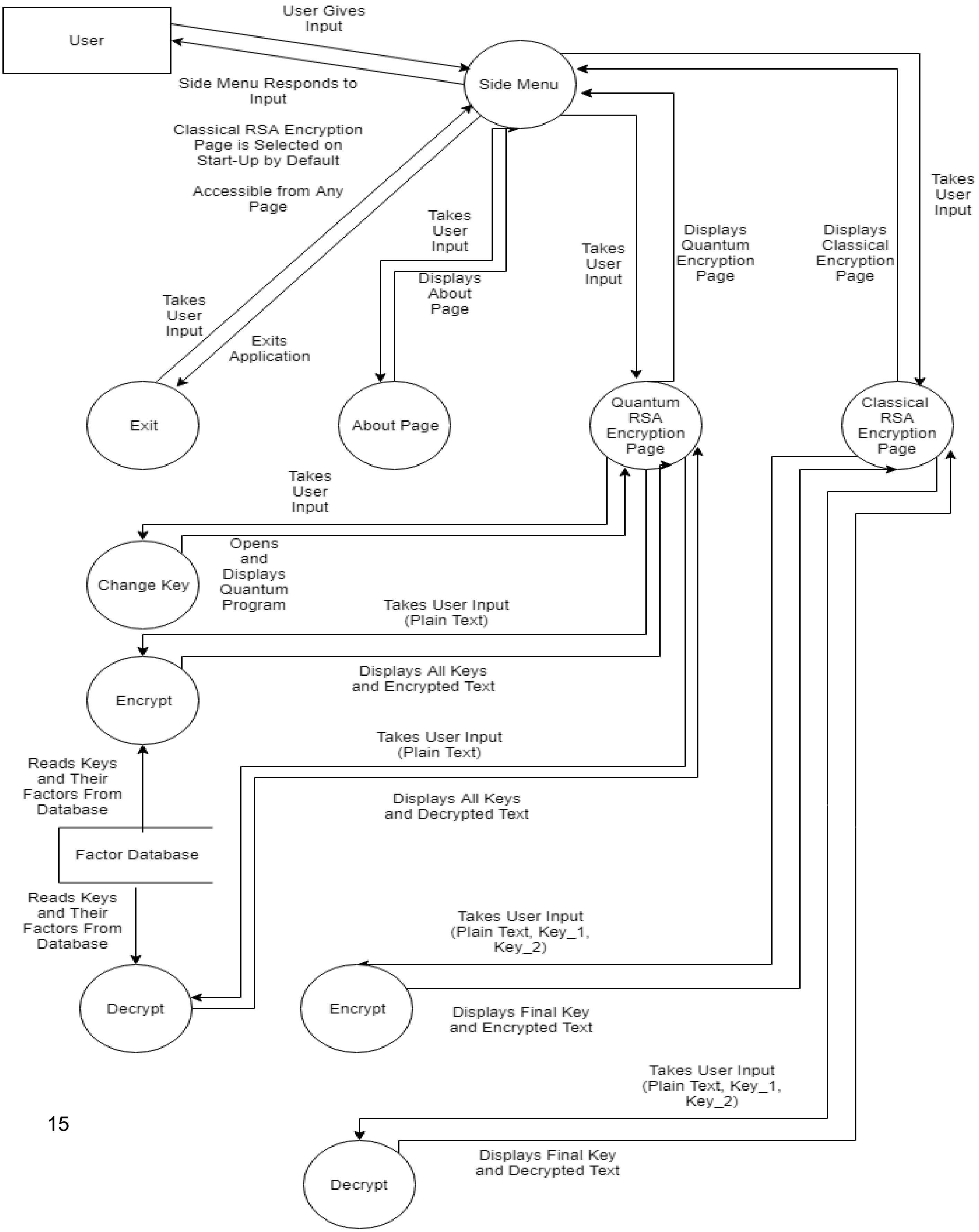
Input	Process	Output
Enter Key_1	Ensures valid input is present in all fields. - *if (!int.TryParse(Convert.ToString(temp_input), out l))* Computes final key, using input of two keys. - *Factors using Shor’s Algorithm* Displays and Saves Key and Factors into External Database - *Shifts each char in array by the ‘final_key%36’, due to the 36 characters in the array*	Saves factors and Displays Factors on Screen (with ongoing working)
Enter Key_2	Ensures valid input is present in all fields. - *if (!int.TryParse(Convert.ToString(temp_input), out l))* Computes final key, using input of two keys. - *Factors using Shor’s Algorithm* Displays and Saves Key and Factors into Same External Database - *Shifts each char in array by the ‘final_key%36’, due to the 36 characters in the array*	Saves factors and Displays Factors on Screen (with ongoing working)

Context Diagram

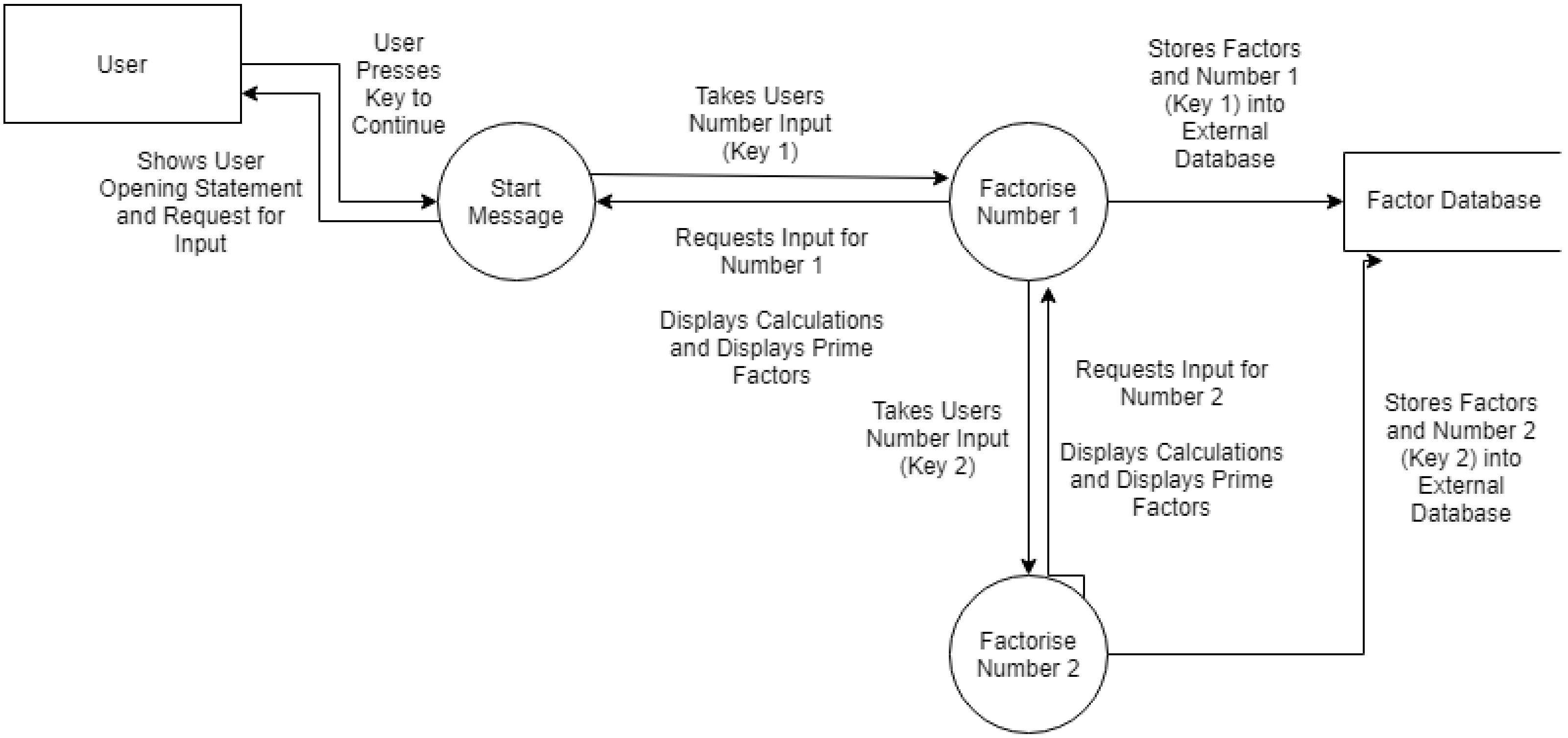

- Inputs Text to Be Encrypted / Decrypted
- Inputs Key 1 and Key 2
- Select Menu Option (Classical, Quantum, About, Exit)
User
Quantum Encryption System
- Outputs Encrypted / Decrypted Text
- Calculates and Outputs Final Key

Data Flow Diagram Level 1

Classical Aspect


User
User Gives Input
Side Menu Responds to Input
Side Menu
Classical RSA Encryption Page is Selected on Start-Up by Default
Accessible from Any Page
Takes User Input
Displays About Page
Takes User Input
Displays Quantum Encryption Page
Displays Classical Encryption Page
Takes User Input
Takes User Input
Exits Application
Exit
About Page
Quantum RSA Encryption Page
Classical RSA Encryption Page
Takes User Input
Change Key
Opens and Displays Quantum Program
Takes User Input (Plain Text)
Encrypt
Displays All Keys and Encrypted Text
Takes User Input (Plain Text)
Reads Keys and Their Factors From Database
Displays All Keys and Decrypted Text
Factor Database
Reads Keys and Their Factors From Database
Takes User Input (Plain Text, Key_1, Key_2)
Decrypt
Encrypt
Displays Final Key and Encrypted Text
Takes User Input (Plain Text, Key_1, Key_2)
Decrypt
Displays Final Key and Decrypted Text

Quantum Program


User
User Presses Key to Continue
Shows User Opening Statement and Request for Input
Start Message
Takes Users Number Input (Key 1)
Requests Input for Number 1
Displays Calculations and Displays Prime Factors
Factorise Number 1
Stores Factors and Number 1 (Key 1) into External Database
Factor Database
Takes Users Number Input (Key 2)
Requests Input for Number 2
Displays Calculations and Displays Prime Factors
Stores Factors and Number 2 (Key 2) into External Database
Factorise Number 2

Log Book

Date:	Time Spent:	Work Completed + Learnt:	Challenges Faced + Overcoming:
23/10/2018	40 min	Brainstormed Ideas and Received Notification - Machine Learning - Neural Networks - Quantum Computing - Cryptography - Internet Communications (5G)	Too many Ideas
25/10/2018	80 min	Narrowed Down to: - Machine Learning - Quantum Computing - Neural Networks	All complex topic requiring a lot of research before practical ideas could begin
29/10/2018	80 min	Made machine learning algorithm that reads and recognises characters.	New concept. Different method of thinking.
01/11/2018	80 min	Changed my mind to Quantum Computing and Started Research. Machine learning seemed limited in research opportunity as is extremely overdone.	Challenging topic. A lot of knowledge of math, physics and computer science required. No idea what the practical would be.
05/11/2018	80 min	Learning concepts behind quantum computing and differences with a quantum computer.	New ideas. Difficult concepts to learn. Differ greatly to classical computers. Keep and open mind and absorb information to overcome. A lot of physics knowledge required, but background knowledge is physics was fairly solid.
08/11/2018	80 min	Learning mathematics behind quantum computing. Learnt basics of linear algebra and dirac vector notation etc.	Challenging to learn the math. Completely different from what is in the syllabus. Kept an open mind
12/11/2018	80 min	Chose Q# as my language. Installed it on my computer and did some research.	Were not many languages as quantum computing is an extremely recent innovation. Barely and documentation and tutorials. Found microsoft's documentation on Q# to overcome this.
15/11/2018	80 min	Learn Basic Q#. Read through microsoft's documentation.	None. Similar to C#.
19/11/2018	80 min	Learnt more advanced Q#. Thought of Shor's Algorithm as a practical demonstration.	Shor's Algorithm is a fairly advanced algorithm requiring hard math. Looked at microsoft's version of Shor's Algorithm to understand it better,

20/11/2018	120 min	Started writing report. Did research into the history of computation.	None. Fairly easy.
22/11/2018	80 min	Continued on report. Finished Design of Trend.	A lot of information to talk about. Limited marks and information is not required in criteria. To solve this, the information was included anyways because it was interesting.
23/11/2018	120 min	Researched the workings of Shor's algorithm. Attempting to understand how it works.	Difficult to do. A lot of unfamiliar ideas. But a basic idea has been received.
26/11/2018	80 min	Thought about how to put Shor's Algorithm into practical. Decided to simply demonstrate RSA. Started research into encryption.	RSA turned out to be not so simple. Further research is required.
27/11/2018	120 min	Researched RSA. Understood main idea of it.	RSA is too complex to recreate. Decided to do modified version of RSA,
29/11/2018	80 min	Decided on how to create modified version of RSA.	Hard to do. Required creative thinking and playing around with numbers in the "Diffie-Hellman Exchange". Figured out a pattern that allows for primes to be transferred, Ended up not to always work. Decided to just assume it works.
30/11/2018	80 min	Created Bare User Interface of Program.	None. Easy to do
01/12/2018	180 min	Decided how to structure my program and what to include. Researched into Hardware and finished that section of report.	Had to think about modified RSA and demonstrating Shor's Algorithm for prime factorisation. Prime factorisation is focus so must demonstrate that. Figured it out and wrote it in my report.
02/12/2018	240 min	Wrote 1000 words on Shor's Algorithm and RSA in the report. Researched Social and Ethical Issues and finished that part of report.	No Challenges. Not Hard. Some challenge to simplify RSA and Shor's Algorithm for the report.
07/12/2018	240 min	Made Significant Progress with Program. Finished Classical Aspect of Encryption.	Not Hard. Some troubleshooting required but fairly simple.
08/12/2018	300 min	Started quantum aspect of the program. Made fair progress.	Difficult to do and understand. A lot of random errors being thrown.
09/12/2018	300 min	Finished quantum program and linked it to classical part of the program. Added about section. Finished program.	Difficult to integrate classical and program due to the nature of the language. Q# program did not exist as .EXE. Just linked VS file to solve this

10/12/2018	300 min	Did real world application and effects on commerce section of report. Did IPO Diagram, Context diagram, and Data Flow Diagram. Polished everything and finishing touches.	Data flow was slightly confusing due to dual nature of program. No challenges apart from that.