Web-Based Chat Application Using AES Cryptographic Algorithm

By

Ronnie Nguru Gitau

145988

CNS

Submitted in Partial Fulfilment of the Requirements of the Bachelor of Science in Computer Networks and Cybersecurity at the Strathmore University

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

June 2023

Declaration

I hereby declare that this work has not been previously submitted and approved by this or any other University. To the best of my knowledge and belief, the report contains no material. previously published or written by another person except where due reference is made in the research proposal itself.

Student's Name: **Ronnie Nguru Gitau**

Sign: _____ Date: _____

Approval

The Computer Networks and cybersecurity project by Ronnie Nguru Gitau was reviewed and approved by:

Supervisor: **Bruce Totona**

Sign: _____ Date: _____

Abstract

The project aims to develop a web-based chat application that leverages the AES (Advanced Encryption Standard) cryptographic algorithm to ensure secure and confidential communication between users. The application will provide a user-friendly interface for real-time messaging, emphasizing data protection and privacy. Through the implementation of AES, all user messages will be encrypted before transmission, making them unreadable to unauthorized individuals. User authentication and access control mechanisms will be integrated to verify user identities and prevent unauthorized access to the chat platform. Using flask, the project aims to have an aesthetically appealing interface, and will be hosted on the cloud using AWS. The cloud will host a Gunicorn server where the code will end up and Nginx be the reverse proxy server. The web-based chat application will offer essential features such as send message, receive message Authentication, file sharing and confirmation message upon signup Emphasis will be placed on scalability and performance optimization to accommodate a growing user base while maintaining efficient message delivery. Throughout the project, rigorous testing and vulnerability assessments will be conducted to identify and mitigate potential security weaknesses. The successful completion of this project will result in a web-based chat application that prioritizes data security using the AES cryptographic algorithm. Users will enjoy a secure and private communication experience, fostering trust and confidence in the application's ability to protect their sensitive information.

Table of Contents

List of Figures

List of tables

List of Abbreviations

AES - Advanced Encryption Standard

AWS - Amazon Web Services

APT - Advanced Persistent Attack

DES - Data Encryption Standard

IDE - Integrated Development Environment

HTTPS - Hypertext Transfer Protocol Secure

NIST - National Institute of Standards and Technology

IPSEC - Internet Protocol Security

SSL - Secure Sockets Layer

TLS – Transport Layer Security

TCP - Transmission Control Protocol

WSGI - Web Server Gateway Interface

Chapter 1: Introduction

1.1 Background

Web applications have become one of the preferred means for users to perform crucial and security-sensitive operations such as communication like chatting or managing bank accounts, official documents, personal health records. The pervasive adoption of such web applications calls for an extensive security analysis in order to avoid attacks (Ajami & BagheriTadi, 2013).

Web security is not a problem of today it is a concern that was raised even 20 years ago by (Dima Alden et al.,1999) although now we have stronger ways to protect data back in the day they looked into discussions about the use of cryptographic module validation programs for increasing and maintaining the security of web applications. Stating examples like, an application may use cryptographic modules to generate passwords. Also, it speaks of the use of COTS components as they may cause serious threats to the security aspect of the application (Lee & Krodel, 2006).

The fact that only the most severe cases of data breaches have negative consequences on a company's intangible capital shows that the existing market may lack the discipline and incentives for cybersecurity investments. For example, Yahoo experienced a massive data breach in 2014, which resulted in the disclosure of 500 million records that became public knowledge in 2016. The company received criticism for their lax approach to cybersecurity and negligence. They failed to make password resets mandatory, delayed communicating with customers, and were vague in their public statements. As a result, Yahoo faced numerous lawsuits and their reputation suffered. Verizon Communications' plans to acquire Yahoo were impacted, resulting in a $350 million decline in the final price in the company's value. Analysts believe that Yahoo!'s failure to prioritize security and poor customer-server interaction were major factors contributing to the decline in reputation (Makridis, 2021).

Businesses not only have to consider the technical aspect but also the implementation as well. For instance, web application, updating the encryption algorithm for a database requires not only a change in algorithms but also re-encryption to prevent data loss. Considering if a database is hashed, it may not be possible to simply upgrade to a new hashing algorithm, but instead use

both the old and the new algorithm in serial, as the existing data cannot be rehashed. Some technical solutions may not be practical to implement due to scalability concerns.

Like any distributed application, web services require robust security mechanisms to ensure secure data transfer. Because web services can be accessed across different platforms, there is a need for heightened security concepts. Because of the compatibility, capability and cross platform access of web services, it demands a greater attention to security concepts The fundamental security requirements for any web-based application are Authentication, Authorization, Confidentiality, Integrity, Availability, and Non-Repudiation. Authentication verifies the user's identity, Authorization checks permissions for access, Confidentiality focuses on the requirement for data in transit between two communicating parties not to be available to a third party, to avoid snooping encouraging the use of VPNs or encryption Integrity detects tampering using hashing algorithms, Non-Repudiation ensures the message sender cannot deny sending it, and Availability guarantees the services are accessible to authorized parties and prevents Denial of Service attacks.

In conclusion, data breaches pose an ongoing threat to personal and financial security, and they are costly for the organizations that hold large collections of personal data. In addition, because so much of our daily lives is now conducted online, it is becoming easier for criminals to monetize stolen information. This problem is especially acute for individual citizens, who have no direct control over the fate of their private information. Therefore, the project looks towards addressing the gaps in different communication systems in terms of real-time communication and ensuring speed in a chatting web application. A key factor will be security. The project intends to adopt the most secure and efficient encryption methods and increase confidentiality and user trust.

1.2 Problem Statement

According to Layton and Watters, (2014), concerns are increasing that data breaches are occurring more frequently, as technologies like the web, social media and the cloud become more integrated within standard business processes, often without rigorous security controls. The problem arises when cyber criminals are able to explore data that is not secured well or secured with a weak encryption method like DES, which can also lead to reputational damage, this is when sensitive data is exposed, and it can damage the reputation of individuals or organizations

which can lead to blackmail and ransom. The data stored on premise i.e., Server logs and databases need to be encrypted with hashing to prevent attackers from viewing the original data, if they managed to break into the server using sophisticated attacks such as APT (Advanced Persistent Attacks).

Developing and maintaining a chatting web application can be costly, and monetizing the application can be challenging. Finding a sustainable revenue model that meets the needs of both users and the business can be a significant challenge. Through this project, there will be a more real-time, secure and efficient way to transfer data between two clients through the AES encryption mode.

1.3 Aim

This project's goal is to develop a secure and real-time chatting web application system that will utilize AES encryption to safeguard users' data when it is stored and sent over a network. With SHA-256 hashing algorithm to hash user passwords saved on the server and add on security.

1.4 Specific Objectives

i. To review the different encryption algorithms used in web applications.
ii. To investigate the impacts of data and privacy breaches in communication systems.
iii. To review the existing applications used for communications and examine their gaps.
iv. To implement the proposed secure and real-time chatting web application.
v. To validate the developed chat web application.

1.5 Research Questions

i. How do the different types of encryptions algorithms function?
ii. What are the impacts of data and privacy breaches?
iii. What are the gaps in the existing chat web applications?
iv. Which is the method to implement a secure and real-time chatting web application?
v. How valid is the implemented solution?

1.6 Justification

This project is consequential in that it provides a system through which users can communicate without fear of data breaches, leaks, or intelligence agencies accessing their information. This

increases user trust when communicating because they know no one else is spying on their messages.

It will be secure through encryption to ensure security of messages and files storage systems and no other third parties can cause a security threat. It is a study that can be implemented by companies that would like a secure chatting space. This model can also help an organization save money since it requires less maintenance and is efficient.

1.7 Scope and Limitation

The project is based on a three-tier client-server architecture: any browser in the world having access to internet connectivity being the client, our server at AWS hosting our application (server) and the relational database. Our application shall be listening on port 80 of a nginx reverse proxy server. All requests received via port 80 shall be redirected to a Unicorn server listening on port 8000 of the local host. Responses shall be redirected to port 80 of our nginx server before being sent to the appropriate client.

The application shall perform encryption of all data at the application level before storage on the relational database. Decryption shall be performed on all encrypted data from the database to the application. The encryption methods to be used is AES encryption algorithm adding hashing security using SHA256.

Tokenization and session contexts shall be employed to ensure that conversations are isolated, and users experience seamless communication. Usage of open SSL will be needed to enable adoption of the HTTPS protocol. Open because you need certification for SSL.

Limitations

Cost of hosting application on the AWS server.

Server Space, AWS only gives you limited space before they can upgrade your space at a cost.

Another limitation is the acquisition of a domain. With the current configuration, a client must know the public address of our server, which changes from time to time.

Chapter 2: Literature Review

2.1 Introduction

Data security and confidentiality are one of the most important aspects of information systems today. One attempt to secure data by using cryptography. This chapter investigates the different encryption algorithms and how they work their benefits and drawbacks. It also looks at the existing systems, their gaps and impacts on data and privacy breaches.

2.2 Encryption Algorithms for Web Applications.

Block ciphers operate on fixed-size blocks of data, typically 64 or 128 bits, of data and can be symmetric or asymmetric, while stream ciphers encrypt and decrypt data in a continuous stream and can also be symmetric or asymmetric. Each type of cipher has its own characteristics and applications, and the choice between block ciphers and stream ciphers depends on the specific security requirements and constraints of the system.

2.2.1 AES Encryption Algorithm

The Advanced Encryption Standard (AES) is the most widely used symmetric cipher today. Even though the term “Standard” in its name only refers to US government applications, the AES block cipher is also mandatory in several industry standards and is used in many commercial systems. Among the commercial standards that include AES are the Internet security standard IPsec, TLS, the Wi-Fi encryption standard IEEE 802.11i, the secure shell network protocol SSH (Secure Shell), the Internet phone Skype and numerous security products around the world. To date, there are no attacks better than brute-force known against AES (Paar et al., 2010).

The AES algorithm incorporates several key transformations to ensure secure encryption. The Byte Substitution Transformation, also known as the S-box transformation, replaces each byte of the state using an S-box. This reversible nonlinear operation adds complexity to the encryption process. The Row Shift Transformation involves transposing bytes within each row of the state matrix. It forms part of the SP network structure, confusing the relationship between the plaintext and the key by shifting each row of the state matrix by different offsets. The Column Mixed Transformation operates on columns using Galois field operations. Treating each column as a quadratic polynomial, this transformation adds further diffusion and complexity to the AES algorithm. Lastly, the Round Key Transformation XORs the end state of each round with the

round key, introducing key-dependent elements into the encryption process. The initial seed key used in AES is extended through a key expansion algorithm to meet the key bit requirement for each iteration. The more rounds that are encrypted and decrypted, the larger the extended bit number needed. To determine the extended key size, the length of the component group is multiplied by the number of rounds plus one. (Chen et al., 2020). The decryption algorithm is basically the inverse operation of the encryption algorithm, and the subkey is used in the opposite order.

AES (Advanced Encryption Standard) is highly regarded for its strong security, making it widely used in both hardware and software applications. It offers key sizes of 128, 192, and 256 bits, which provide increased resistance against hacking attempts. Breaking AES encryption with a 128-bit key requires around 2^128 attempts, making it extremely difficult to crack. This level of security makes AES the preferred choice for various applications, including wireless communication, financial transactions, e-business, and encrypted data storage. It is also widely adopted in both commercial and open-source solutions, establishing its credibility and trustworthiness (Koshy, 2020).

One drawback of AES is that it relies on algebraic structures that are relatively simple and straightforward, which may introduce vulnerabilities. Additionally, AES encrypts all blocks in the same manner at all times, which could potentially be exploited by attackers. Implementing AES in software can be challenging, especially when considering both performance and security requirements. Specifically, in counter mode, AES can be difficult to implement efficiently while ensuring adequate security measures are in place (Koshy, 2020).

2.2.2 DES Encryption Algorithm

The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). For DES, data is encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output (Zeebaree, 2020).

DES takes input of sixty-four-bit simple text meaning it encrypts the facts in blocks of size sixty-four-bit every. DES makes use of 16 rounds of substitution and permutation and every round is a feistel round (Kaur & Sodhi, 2016).

The drawback of DES is that it is not as fast as AES the DES set of rules spends the most time on encryption (Kumar Tiwari et al., 2022). Because of its lack of surety, it is being phased out and has been reported that it is not even secure to be used by the military or the government. The bottom line is that DES is not used to protect security and privacy of sensitive or confidential data (Sharma & Garg, 2016).

2.2.3 Blowfish Encryption Algorithm

Blowfish, developed by Bruce Schneier in 1993, is a symmetric-key cryptography technique designed as a replacement for DES and IDEA. It utilizes a variable-length key ranging from 32 bits to 448 bits. One notable advantage of Blowfish is its relatively faster encryption compared to DES, making it a viable alternative. Moreover, Blowfish is freely available for public use, allowing widespread adoption and implementation (Sharma et al., 2021).

Key features of the Blowfish algorithm include a block size of 64-bits, a set of 18 subkeys known as the P-array, and a variable key size. The 64-bit block size ensures efficient processing and compatibility with existing systems, while the 18 subkeys contribute to the algorithm's robustness and security. The variable key size allows for flexibility and customization, enabling users to adapt the encryption strength to their specific needs (Sharma et al., 2021).

The encryption steps in the Blowfish algorithm involve generating subkeys, initializing substitution boxes, and performing the encryption process. Firstly, subkeys are generated from the main key to be used in subsequent operations. Then, the substitution boxes are initialized with specific values. Finally, the encryption process is carried out, where the input data is divided into blocks and subjected to mathematical operations, resulting in the generation of ciphertext (Nie & Zhang, 2009).

The decryption steps in the Blowfish algorithm mirror the encryption process. Generation of subkeys from the main key, followed by the initialization of substitution boxes. The decryption process itself involves reversing the encryption operations, allowing the recovery of the original plaintext from the ciphertext. Both encryption and decryption rely on the generation of subkeys and the initialization of substitution boxes to ensure consistency and maintain the security of the Blowfish algorithm.

Benefits of implementing the Blowfish algorithm include reduced memory requirements, fast and secure encryption, and ease of implementation. The algorithm's efficient memory usage makes it suitable for resource-constrained environments, minimizing the impact on system resources. Additionally, Blowfish offers a desirable balance between encryption speed and security, ensuring efficient processing of data while maintaining a high level of protection. Furthermore, its straightforward implementation simplifies the development and integration process for both software and hardware implementations, saving time and resources (Nie & Zhang, 2009).

However, the Blowfish algorithm does have some drawbacks to consider. One limitation is the requirement for unique sender and receiver identities. To maintain security, it is crucial to ensure that both parties involved in the encryption process are distinct and verifiable. Failure to establish uniqueness may compromise the integrity and confidentiality of the encrypted data. Additionally, managing keys in the Blowfish algorithm can become complicated, especially when dealing with variable key sizes. Proper key management practices, including secure key storage and distribution, are essential to maintain the algorithm's security.

2.2.4 RSA Encryption Algorithm

There are different types of cryptographic algorithms that can be utilized to establish a secure data system, one of which is the Rivest Shamir Adleman (RSA) cryptography algorithm. RSA algorithm utilizes asymmetric algorithms that consist of a pair of distinct keys used for encryption and decryption. The security level of the RSA encryption algorithm is significantly influenced by the key size of the password. If the key size is smaller, there is a higher possibility for the locks to be broken using the brute force attack method that involves examining the combinations of keys one by one. The key size of the password in the RSA algorithm is affected by the prime numbers generated during the process (Preetha & Nithya, 2013).

Modern-day RSA encryption involves four main steps: key generation, encryption, decryption, and security. The key generation process involves selecting two large prime numbers and using a mathematical algorithm to generate the public and private keys. The sender then uses the recipient's public key to encrypt the message, which is first converted into numerical values using a predetermined system. The recipient can then decrypt the message using their private key and a series of mathematical operations. The security of RSA encryption depends on the

difficulty of factoring two large prime numbers, and longer key lengths are used to enhance the security of modern-day RSA encryption (Rowland, 2016).

RSA encryption has several advantages including high security due to the use of large prime numbers that are difficult to factorize. Additionally, it offers authentication and digital signature capabilities that are important for secure communication and electronic transactions. RSA encryption is also widely used in various applications such as online banking, email encryption, and VPNs. Another advantage of RSA encryption is that it does not require a secure channel for the exchange of keys. Compared to public-key ciphers, such as RSA, the structure of DES and most symmetric ciphers is very complex and cannot be explained as easily as RSA and similar algorithms (Rowland, 2016).

RSA Encryption has some drawbacks that must be considered. According to (Kak, 2016), RSA encryption can be slow compared to other encryption methods, making it unsuitable for time-critical applications. Another disadvantage is key management, which can be complex and time-consuming to ensure system security. Additionally, RSA encryption is vulnerable to attacks such as side-channel attacks and key generation attacks. Lastly, the security of RSA encryption is related to key size, with longer keys requiring more processing power and memory, which can be a disadvantage in resource-limited environments (Preetha & Nithya, 2013).

Table 2.1 Summary of Comparison of existing Encryption algorithms and their features.

	Name	Block Size	Key size	Security	Speed	Speed depends on key size?
AES	Advanced Encryption Standard	128 bits	128,192,256 bits	Secure	Fast	Yes
DES	Data Encryption Standard	64 bits	56 bits	Insecure	Slow	-
Triple DES	Triple Data Encryption Standard	64 bits	56 - 168 bits	Moderately Secure	Very Slow	No
Blowfish	-	64 bits	32 - 448 bits	Moderately Secure	Fast	No

2.3 Impacts of Data and Privacy Breaches

2.3.1 Reputational Damage

Reputation damage impacts both individuals and organizations. When confidential data is accessed without authorization, it undermines public trust and confidence. This loss of trust can have detrimental effects on an organization's reputation, resulting in decreased customer confidence and potential business loss. Recovering from a tarnished reputation can be a daunting task, requiring substantial effort and time. In 2018, Facebook faced a major scandal involving Cambridge Analytica, a political consulting firm that improperly obtained and misused the personal data of millions of Facebook users. The incident raised concerns about user privacy and data protection, leading to a decline in trust and a tarnished reputation for the social media giant.

2.3.2 Financial Loss

Direct financial cost is any cost directly attributed to the breach, including all work time, overtime, external cost, and equipment and legal costs, whereas indirect financial cost is not directly related to the breach but incurs as a result of the breach, such as costs of lost productivity and penalties from clients in the case of a denial-of-service attack (Wang & Wood, 2019). According to sokodirectory.com, In 2020, the Equity Bank reported that it had lost KES 1.3 billion ($13 million) to a cyber-attack. The bank did not disclose the details of the attack, but it was reported that the attackers used phishing emails to access the bank’s system and transfer the money to various accounts. Also, Yahoo experienced a massive data breach in 2014, which resulted in the disclosure of 500 million records that became public knowledge in 2016. As a result, Yahoo faced numerous lawsuits and their reputation suffered. Verizon Communications' plans to acquire Yahoo were impacted, resulting in a $350 million decline in the final price in the company's value (Makridis, 2021).

2.3.3 Provides an entry point to propagate other complex attacks.

If an attacker gains control of a communication system with weak security measures, they can exploit it to carry out various additional attacks. These attacks may include distributing malware to unsuspecting victims or monitoring their communications to gather more valuable information like banking credentials, passwords, and user locations. As a result, victims may experience financial losses due to unauthorized access to their sensitive data. In 2017, Equifax, a major credit reporting agency, suffered a significant data breach due to hackers exploiting a

vulnerability in their web application software. The breach exposed the personal information of around 147 million individuals, including sensitive details like names, Social Security numbers, and credit card information. This security breach not only compromised the personal data but also provided a gateway for the hackers to carry out further sophisticated attacks within Equifax's network.

Once inside the system, the hackers were able to move through the network and conduct malicious activities. The stolen data had the potential to be used for identity theft, financial fraud, and other harmful purposes. The incident underscored the critical need for strong web application security measures, including regular patching and effective vulnerability management.

2.4 Existing Applications and their Gaps.

There are many defenselessness and attack virus for web-based application. This includes both web-specific (i.e. Cross-Site Scripting), as well as generic (i.e. Password Sniffing), all of which leave the user susceptible to being victims of identity theft means the security issue is much greater problem in on the network some solution will be research or find but it's not sufficient such as SSL may fully protect an application due to the introduction of new web concepts such as cloud computing and the growing complexity of attack vectors (Sharif, 2022).

2.4.1 Adverts and hacking on Discord

One of the most effective ways for hackers to wreak havoc is to place malware or a virus in the ads that they send out to legitimate sites. To the publishers and the ad reviewers, these ads look just like any other. To other certain groups of viewers, which are specifically targeted by the virus creator, a click, or simply a view, is all it takes for the evil wheels to be in motion. This vector of spreading can have potentially hundreds of thousands of computers infected by the malicious code. The malware can be as simple as adding countless pop-ups or load more ads, or as destructive as having the ability to spy, wipe or steal data. For example, while many chat applications are free to use, they equip the application with built-in processes which track every single movement of their users, and they use this type of information to sell to 3rd party advertisement agencies (Dashtinejad, 2015).According to Szabo Discord allows file sharing, meaning that anyone can easily share a picture, video, link, or anything of that caliber on a server or through a private message. This makes it easy for someone to share an IP Grabber, which can

be used to track users' IP addresses for a variety of reasons, such as targeted advertising or identifying the location of a user (Szabó, 2023).

2.4.2 Insufficient Encryption on Viber

Research has revealed that media files such as pictures or videos which are transferred between the users have no encryption and the data is stored on the Viber server unencrypted which can be accessed without any authentication mechanism. These vulnerabilities give the ability to a malicious user to simply launch MitM attacks and capture unencrypted data either over the wired or wireless networks. Viber does not support end-to-end encryption. (Dashtinejad, 2015).

Sufficient encryption is key to adequate security most of the chatting application look for ethical. hackers to verify their systems and ensure that the security is good enough, in comparison to Viber, Kaspersky (2022), tested the signal application and said "First of all, the cybercriminals did not gain access to correspondence. Signal uses end-to-end encryption with the secure Signal Protocol.

2.4.3 Insufficient Session Management

Session management attacks are those attacks against web applications in which an attacker gets control of a user's browser in order to execute malicious script (usually an HTML) within the context of trust of the web application's site (Nagpal et al., 2014).

As a result, and if the embedded code is successfully executed, the attacker might then be able to access, passively or a masquerade attack by IP packet substitution (such as stream cipher attack). Suppose the Victim wants to log on a web site. Victim sends username and password. Web Site verifies the couple. If an attacker can listen to the information transferred Sniffer (unencrypted) / Trojan (encrypted). He can log-in to the system using Username and Password (Nagpal et al., 2014). Rocket. Chat through 3.4.2 allows XSS where an attacker can send a specially crafted message to a channel or in a direct message to the client which results in remote code execution on the client side (CVE, National Vulnerability Database, 2020).

Figure 2.2 summarizes common vulnerabilities on the chat web applications and how it takes place.

Attack Type	How it Takes Place	Examples of Chat Web Applications
Brute Force Attack	Repeatedly guessing usernames and passwords to gain unauthorized access	Chat Gum, Chat Crypt
Session Spotting	Observing and identifying active session information, such as session tokens or cookies, to impersonate a user.	Discord, Slack
Replay Attack	Capturing and reusing network traffic to impersonate a user or execute unauthorized actions.	WeChat
Session Hijacking	Unauthorized access to an active session by stealing session identifiers or manipulating session management mechanisms	Facebook Messenger, WhatsApp

2.5 Conceptual Framework

Application will be hosted on AWS, and the data stored in MySQL. Coded using Python, HTML, JavaScript and CSS on a Linux machine.

AES encryption will be implemented on the client machine using java script encryption module before HTTPS request is sent to the nginx server. Additionally, AES decryption will be implemented on the client machine upon using JavaScript AES module upon reception of HTTPS response. Hence, end to end encryption facilitated. SHA256 encryption and decryption will be implemented prior to data storage in the database and after retrieval, respectively.

Flask application will be hosted on AWS and run locally on http://127.0.0.1:8000 using Gunicorn. However, we shall set a reverse proxy listening at port 80 or 443 that will redirect any incoming traffic to our Flask Application. This will be using nginx (nginx) server. Configuration of TCP ports will be implemented at the cloud level using IAM Manager.

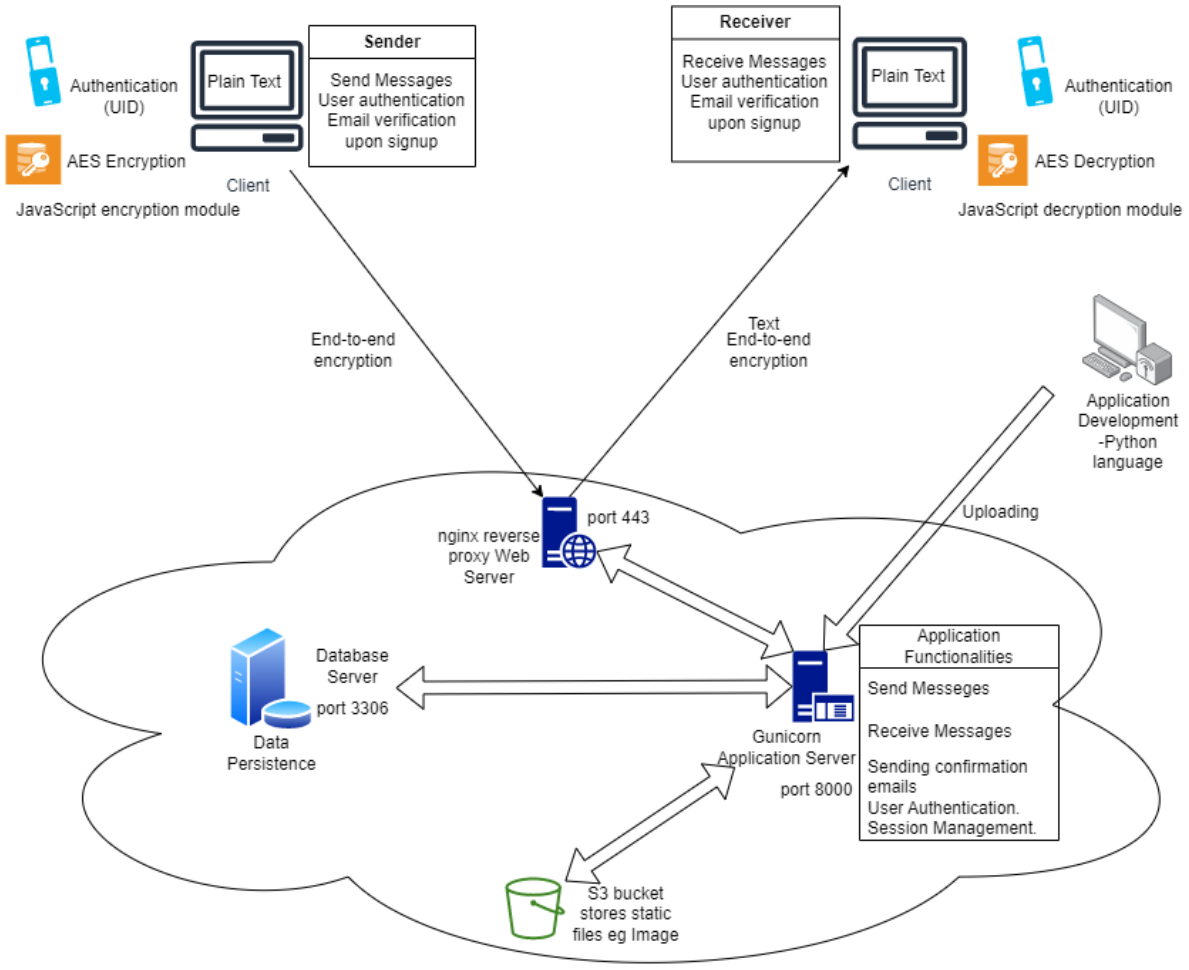

Figure 2.1 Conceptual Framework Diagram

Chapter 3: Methodology

3.1 Introduction

This chapter provides an in-depth overview of the system development process, including the design components utilized during the coding phase. It incorporates several essential element and multiple key concepts such as Sequential Development Methodology, Systems Tools and Techniques, and the Approaches employed for testing the developed system.

3.2 Methodology

Rapid application development (RAD) is a widely utilized agile project management strategy in the field of software development. The primary advantage of employing a RAD approach is its ability to expedite project completion, making it an appealing option for developers operating in fast-paced environments such as software development.

By reducing planning time and emphasizing prototype iterations, RAD allows project managers and stakeholders to accurately measure progress and communicate in real-time on evolving issues or changes. This results in greater efficiency, faster development, and effective communication. Below is a sample figure that shows the Rapid application development cycle and the steps it takes from planning to testing focusing most of the time on the prototype cycle.

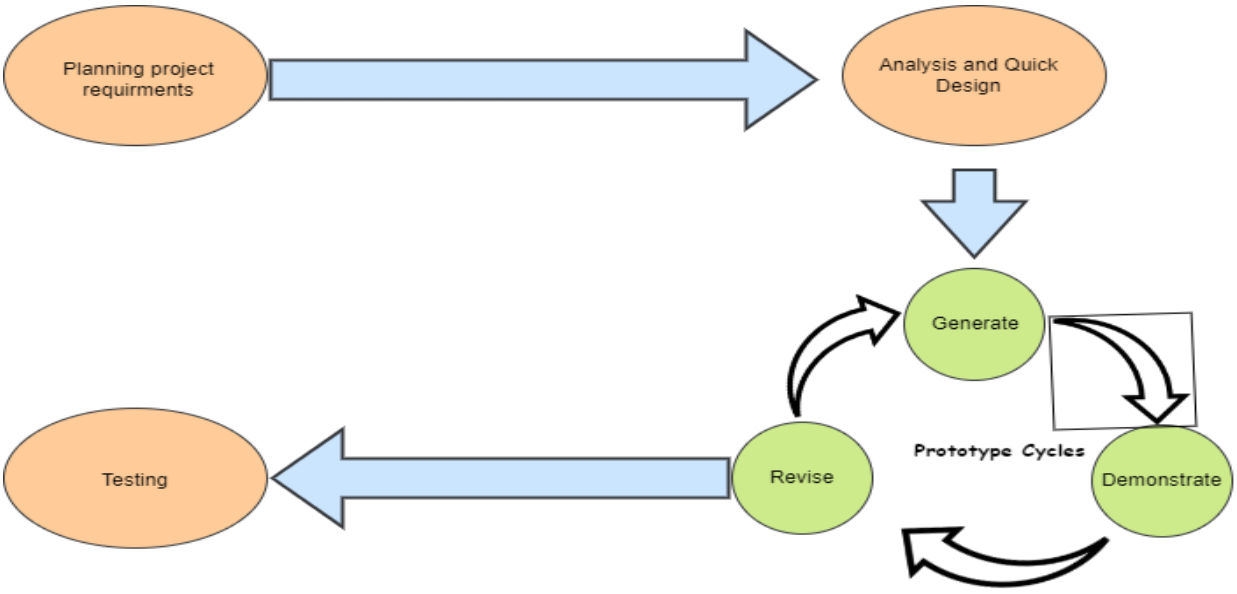

Figure 3.1 Rapid application development cycle

3.2.1 Planning and Requirements Gathering

During this phase, meticulous attention will be given to the information and requirements-gathering process, which is crucial for the successful implementation of the project. Using online search look at exisiting documentation and review papers of work that has already been done to guide me through my project. Compare exisiting tools and decide on the ones that best fit my project.

3.2.2 Analysis and Design

Upon completion of the system planning phase, an extensive system analysis will be undertaken to determine the most effective logic for the system's complete implementation. This analysis will be carried out utilizing advanced system analysis tools, including use-case diagrams and flowcharts, to ensure a thorough and precise understanding of the system's requirements and functionalities.

3.2.3 Implementation

In this phase, the project shall focus on the actual development of the chat application using Flask. The application will facilitate communication among registered members stored in our database through the utilization of user sessions.

Firstly, the PyCharm IDE will be installed, providing a robust development environment for efficient coding and debugging. Simultaneously, the Python programming language will be installed to ensure compatibility and enable the execution of the application. Furthermore, Flask, a popular web framework, will be installed to facilitate the development of the chat application.

To host the application, an Amazon Web Services (AWS) account will be set up, utilizing student credits to minimize costs. An EC2 instance will be provisioned to host the chat application, while static images will be stored in an S3 bucket, ensuring efficient and reliable storage.

To ensure optimal performance and scalability, the installation of the Gunicorn application server and Nginx web server will be carried out. Gunicorn will act as a WSGI HTTP server, serving the Flask application, while Nginx will handle proxying and load balancing. The utilization of port 443 will ensure secure communication through HTTPS.

During implementation, careful attention will be given to coding practices and the configuration of essential tools. The objective is to create an aesthetically appealing, functional, and flexible project. To ensure progress and enhance security, an iterative approach will be employed. For instance, when implementing encryption algorithms, the key size of the AES encryption algorithm will be gradually increased from 128 bits to 256 bits. This step-by-step process allows for monitoring progress while balancing the trade-off between speed and security during encryption and decryption operations.

The iterative loop of development, demonstration, and refinement will continue until a more refined and precise system model is achieved. Throughout the implementation phase, the tools established during the planning and requirements-gathering phase will be effectively utilized, ensuring the project aligns with the established objectives and specifications.

3.2.4 Testing

During the testing phase, the application will undergo rigorous evaluation to ensure its functionality, reliability, and adherence to established requirements. Several testing methods will be employed to assess the system's performance and validate its behavior.

Unit testing will be conducted to verify the individual components or modules within the application. By examining each unit in isolation, the test results will contribute to enhancing the overall quality of the system.

Integration testing will be employed to assess the interactions between different components of the application. By verifying the integrity of the connections and data flow between components, potential issues can be identified and addressed promptly.

System testing will be carried out to evaluate the application. Through systematic test scenarios, the system will be examined in different usage scenarios, and any deviations or anomalies will be identified and resolved.

Performance testing will gauge the application's response time, scalability, and resource utilization under various load conditions. By subjecting the system to simulated user interactions, its ability to handle concurrent requests and maintain optimal performance levels will be

measured. This testing will ensure the application can handle anticipated user volumes without significant degradation.

Security testing will be conducted to identify vulnerabilities and ensure the application's resilience against potential security threats. By examining the application's access controls, data encryption, and protection mechanisms, potential risks will be identified and addressed to safeguard the system and its users' sensitive information.

Throughout the testing phase, meticulous documentation of test cases, procedures, and results will be maintained. This documentation will serve as a valuable reference for future enhancements, bug fixes, and quality assurance activities.

3.3 System Analysis

The system will be constructed using the object-oriented analysis and design (OOAD) technique, which applies object-oriented concepts to thoroughly analyze and design the system. This method, integrated within the software development life cycle, involves the creation of classes, objects, and graphical system models. Within the OOAD approach, significant attention is given to the input data required for system creation during the coding phase of the Rapid Application Development (RAD) methodology. This meticulous focus ensures that the final system, when presented, is constructed with meticulously reviewed and validated data.

The system architecture is formulated based on the system's context, adhering to architectural design principles, and leveraging domain knowledge. Typically, the system is structured into layers, with each layer further decomposed to form subsystems. This systematic approach establishes an organized framework for the system, facilitating efficient implementation and maintainability.

3.3.1 Functional Requirments

Functional requirements provide insights into the operations and capabilities of a system, outlining the specific features and characteristics it should possess. These requirements define the expected behavior and functionalities of the system.

Below are the essential functional requirements for the chat web application, organized in a logical order:

i. Develop and present an aesthetically pleasing and user-friendly front-end interface that effectively displays the application's functionalities and features.
ii. Implement a key exchange mechanism between the two clients, utilizing the Diffie-Hellman algorithm. This process will enable secure encryption and decryption of messages exchanged between the clients.
iii. Encrypt the messages transmitted between the clients using the AES encryption algorithm, further enhancing the security and confidentiality of the communication.
iv. Configure a storage mechanism to hold the messages securely. Employ the SHA-256 hashing algorithm to ensure the integrity and protection of the stored messages.
v. Implement the decryption process using AES decryption to enable the receiving client to decrypt and read the encrypted messages accurately.
vi. Incorporate user authentication and authorization mechanisms to ensure that only registered and authorized users can access the chat application, enhancing its security and privacy features.
vii. Establish a mechanism to gracefully close the session once the communication between the clients is completed, ensuring proper termination, and freeing up system resources.

3.3.2 Non-Functional Requirements

Non-functional requirements are specifications that define the performance and characteristics of a system. These requirements describe the aspects related to how the system should perform rather than its specific functionalities.

Availability: The system should be accessible for utilization in any organization that has a well-established network infrastructure. This accessibility primarily relies on the use of a web server, which can be accessed if there is an internet connection and a web browser. The two clients involved in the system should be easily accessible to users, facilitating efficient real-time communication.

Maintainability: The system should exhibit cost-effective maintenance capabilities, particularly due to its web-based nature and utilization of a server, which facilitates efficient manageability. It is anticipated to have a prolonged lifespan and should be adaptable to accommodate additional requirements, such as modifiability, configurability, extensibility, and interoperability.

Simultaneously, it is crucial to minimize the costs associated with the web server to ensure unrestricted communication without concerns about excessive expenses.

Performance: Each client connected to the system requires a specific bandwidth to enable seamless communication, as different types of network traffic typically have varying bandwidth demands. However, in this prototype, a lower bandwidth requirement is deemed adequate, resulting in minimal latency and few delays. Aspects such as latency, load, and resource utilization must be taken into consideration to ensure a smooth end-user experience. Striking the optimal balance between securing the algorithm and safeguarding against latency-related issues becomes paramount in achieving this objective.

Security: To uphold data integrity and security, it is imperative for the system to implement AES encryption as a means of data encryption and decryption. AES, a symmetric encryption algorithm, is well-suited for efficiently handling substantial data volumes. Moreover, to generate cryptographically robust random numbers vital for managing sensitive data, including passwords, account authentication, security tokens, and other secrets, the system should make use of the secrets module. The secrets module is specifically designed to access the most secure source of randomness, thereby bolstering the overall security of the system. Additionally, the storage incorporates hashing utilizing SHA-256 to safeguard data even in cases where loss or corruption occurs in the database.

Testability: The quality attribute requirement of testability encompasses multiple facets, such as the extent of test coverage, the effectiveness and efficiency of web-based tests, and the quality of testing reporting. It plays a direct role in determining the level of confidence in the system's capability to operate as intended.

3.4 System Development Tools and Techniques

3.4.1 Python

Python, an extensively utilized and interpreted programming language, is renowned for its simplicity and readability. It emphasizes code readability by implementing significant indentation, facilitating developers in comprehending, and maintaining their code efficiently. Python adopts a dynamic typing approach, enabling variables to hold values of different types during program execution. Additionally, Python incorporates automatic garbage collection,

simplifying memory allocation and deallocation tasks and relieving developers from manual memory management burdens.

Regarding web development, Python presents Flask, a highly favored web framework. Flask is lightweight and flexible, enabling the creation of web applications using Python. It equips developers with essential tools and libraries for constructing web services and APIs. Emphasizing simplicity and minimalism, Flask empowers developers to concentrate on specific requirements and customize their applications accordingly. Its modular architecture and extensive documentation contribute to Flask's popularity in web application development using Python. The framework offers scalability and adaptability, accommodating projects of varying sizes and complexities.

3.4.2 AES encryption modes

AES is a widely used block encryption algorithm that offers secure data protection. In November 2001, five standardized modes of operation were introduced for the AES algorithm: ECB, CBC, CFB, OFB, and CTR. But much later to deal with the upcoming technologies CGM (Galois/Counter Mode) When it comes to implementing AES in a secure manner, it is recommended to use the CGM mode.

AES-GCM (Galois/Counter Mode) is an encryption mode that combines the AES algorithm with a universal hash function called GMAC. It offers both confidentiality and integrity protection, ensuring the security of the data. AES-GCM achieves authenticated encryption by generating an authentication tag that can be used to verify the integrity of the ciphertext during decryption, safeguarding against tampering and unauthorized modifications.

One advantage of AES-GCM is its faster performance compared to AES-CBC. It supports parallel processing, allowing multiple blocks to be processed simultaneously, which improves encryption and decryption speed. However, to maintain the security of the encryption, AES-GCM requires a unique nonce (initialization vector, IV) for each encryption operation. This nonce should never be reused with the same key to prevent security vulnerabilities.

Many modern cryptographic libraries and frameworks, including Flask's cryptography extension, support AES-GCM mode, making it convenient to implement in web applications. However, it's important to consider specific requirements and compatibility constraints. While AES-GCM is

recommended for its confidentiality, integrity, and performance benefits, AES-CBC mode can still be a viable choice if you have specific constraints or if compatibility with legacy systems is a concern. AES-CBC has been widely supported and used in various applications for a long time, making it a reliable option in such cases.

3.4.3 AWS Server

The AWS cloud will be the host and the storage of most of my project it will also assist in the networking and flow of data. It will host the Gunicorn Application server via port 8000 which will be hosting the python code used to run the system as well as the flask used for the aesthetics of the application. Some of the functionalities that will be coded and put into the Gunicorn server include send messages receive messages just to mention a few. S3 bucket will be used to store static data like images whereas the Database Server (Data Persistence) will be used to take care of the other data. Nginx reverse proxy Web server will also be hosted on the cloud and will be key in communication between the clients and the servers.

3.5 Method Used to Test Developed System

To ensure the functionality and reliability of the software components, the incorporation of unit testing is an integral part of the development process. By conducting unit testing, early identification and resolution of any issues or discrepancies can be achieved, avoiding potential setbacks during the final stages of development. This proactive approach not only streamlines the debugging process but also contributes to a reduction in overall development time.

In addition, the system aims to establish secure communication between clients through the implementation of encryption mechanisms. By enabling encrypted communication, an enhanced level of privacy and confidentiality is ensured for the exchanged data. This robust measure safeguards sensitive information from unauthorized access or tampering, establishing a secure communication channel.

As the development phase nears completion, system testing will be employed to comprehensively evaluate the overall system's compliance with specified functional and non-functional requirements. Through this comprehensive testing approach, the system's performance, reliability, and adherence to desired functionalities will be assessed. This rigorous evaluation process allows for the validation of the system's capabilities, identification of any

potential gaps or shortcomings, and necessary improvements to ensure the final system meets the specified requirements.

3.6 Deliverables

Deliverables are key items used to measure the progress of a project. The deliverables of this project includes the following:

i. A fully functional and well-designed chat application developed using Flask, a popular web framework in Python.
ii. An implementation of essential features including user registration and login, real-time messaging, and secure communication between clients.
iii. Documentation of test cases, procedures, and results to serve as a reference for future maintenance and enhancements.
iv. A robust implementation of a database to securely store user data, facilitating seamless user management and retrieval of information.

References

Ajami, S., & Arab-Chadegani, R. (2013). Barriers to implement electronic health records (EHRs). *Materia socio-medica*, *25*(3), 213.

Chen, X. (2020, August). Implementing AES encryption on programmable switches via scrambled lookup tables. In *Proceedings of the Workshop on Secure Programmable Network Infrastructure* (pp. 8-14).

Dashtinejad, P. (2015). Security System for Mobile Messaging Applications.

Dima, A., Wack, J., & Wakid, S. (1999). Raising the bar on software security testing. *IT professional*, *1*(3), 27-32.

Howard, F. M., Kochanny, S., Koshy, M., Spiotto, M., & Pearson, A. T. (2020). Machine learning–guided adjuvant treatment of head and neck cancer. *JAMA network open*, *3*(11), e2025881-e2025881.

Kak, S. (2016). Understanding RSA algorithm and its vulnerabilities. International Journal of Advanced Computer Science and Applications, 7(11), 457-460.

Kaur, N., & Sodhi, S. (2016). Data Encryption Standard Algorithm (DES) for Secure Data Transmission. *IJCA Proceedings on International Conference on Advances in Emerging Technology*. https://www.ijcaonline.org/proceedings/icaet2016/number2/25887-t036

Kumar Tiwari, P., Choudhary, V., & Raj Aman, S. (2022). Analysis and comparison of DES, AES, RSA encryption algorithms. *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. https://doi.org/10.1109/icac3n56670.2022.10073996

Layton, R., & Watters, P. A. (2014). A methodology for estimating the tangible cost of data breaches. *Journal of Information Security and Applications*, *19*(6), 321–330. https://doi.org/10.1016/j.jisa.2014.10.012

Lee, Y. H., & Krodel, J. (2006). FLIGHT-CRITICAL DATA INTEGRITY ASSURANCE FOR GROUND-BASED COTS COMPONENTS. *FLIGHT-CRITICAL DATA INTEGRITY ASSURANCE FOR GROUND-BASED COTS COMPONENTS*.

https://www.faa.gov/aircraft/air_cert/design_approvals/air_software/media/06-2_cots_fltcriticaldata.pdf

Makridis, C. A. (2021). Do data breaches damage reputation? Evidence from 45 companies between 2002 and 2018. *Journal of Cybersecurity*, *7*(1), tyab021

Nie, T., & Zhang, T. (2009). A study of DES and Blowfish Encryption Algorithm. *TENCON 2009 - 2009 IEEE Region 10 Conference*. https://doi.org/10.1109/tencon.2009.5396115

Paar, C., & Pelzl, J. (2010). Chapter 4 The Advanced Encryption Standard (AES). In *Understanding cryptography* (pp. 87–87). essay, Springer Berlin Heidelberg.

Preetha, M., & Nithya, M. (2013, June). *A STUDY AND PERFORMANCE ANALYSIS OF RSA ALGORITHM*. A STUDY AND PERFORMANCE ANALYSIS OF RSA ALGORITHM . https://www.ijcsmc.com/docs/papers/June2013/V2I6201330.pdf

Rowland, H. (2016). The Role Of Prime Numbers in RSA Cryptosystems.

Sharif, A., Isoaho, J., & Farooq, A. (2022). Threat modelling with UML for cybersecurity risk management in OT-IT integrated infrastructures.

Sharma, V., Chauhan, A., Saxena, H., Mishra, S., & Bansal, S. (2021, October). Secure file storage on cloud using hybrid cryptography. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-6). IEEE.

Szabó, M. S. (2023, May 3). *Using discord? don't play down its privacy and security risks*. WeLiveSecurity. https://www.welivesecurity.com/2023/05/03/using-discord-privacy-security-risks/

VLSI Architecture for AES Encryption. (2020, July 11). *Journal of Xidian University*, *14*(7). https://doi.org/10.37896/jxu14.7/085

Zeebaree, S. R. (2020). DES encryption and decryption algorithm implementation based on FPGA. *Indones. J. Electr. Eng. Comput. Sci*, *18*(2), 774-781.

Zeebaree, S. R. (2020). DES encryption and decryption algorithm implementation based on FPGA. *Indonesian Journal of Electrical Engineering and Computer Science*, *18*(2), 774. https://doi.org/10.11591/ijeecs.v18.i2.pp774-781

Appendices

Appendix 1: Gantt Chart

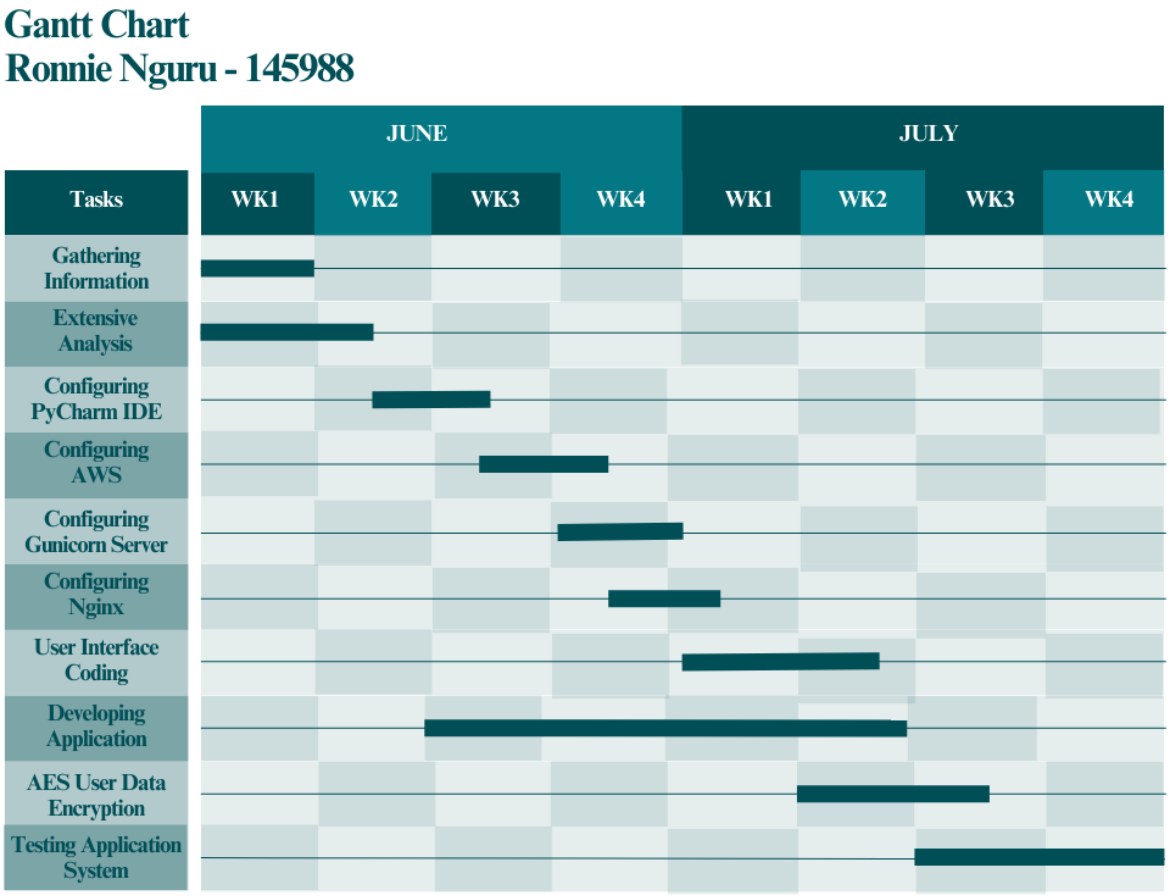

Figure 3.2 Project Gantt Chart

Strathmore University

School of Computing and Engineering Sciences

Project Proposal Assessment Guide

Student Number	
Working Title:	

Evaluation Areas	Weight	Score	Notes
Title page: Informative, concise, and appropriate	**2 pts**		
Abstract To have background, problem, solution, methodology (approach data and tools) outcomes and expectations	**2 pts**		
Introduction Background **(2)** *A clear illustration of issue, context and audience* Problem Statement **(2)** *Pain points, audience, who is affected and how solution comes in to fix the pain.* Objectives (S.M.A.R.T and Linked to Problem Statement) **(2)** Research questions **(1)** *Alignment of questions with objectives* Justification **(2)** *Should be research supported.* Scope of Project **(2)**	**(13 pts)**		

Specify boundaries of people process, HW/SW, data etc. Limitations **(1)** *Challenges Expected* Delimitation **(1)** *To do to counter anticipated challenges*			
Literature Review/Related Work Objectives mapping to Literature Review **(2)** Critique of Theoretical framework and content adequacy (**2**) *Principles, parameters of consideration* Discussion of technologies contextualization for the proposed work **(2)** Citations of content and alignment to work **(2)** Review of at least 3 systems comprehensively the working behind it **(2)** Gaps identification, analysis relative to the proposed solution **(1)** Conceptual Framework clear to communicate how it works, data flows, processing, actors **(3)** *Diagram that's clear; discussion of diagram.* *Describe input process output storage boundaries.*	**(14 pts)**		
Methodology Methodology and justification (**2**) Correct Methodology Application (**1**),	**(8 pts)**		

Design and Development tools (**2**) Deliverables and milestones **(2)** *Examinable bits from ideation* *Proposal, design, test cases documentation doc* *Proof of concept- modules* Gantt Chart that makes sense relative to the project **(1)**			
Proposal Presentation Table of Contents and List of Figures **(2)** Are relevant references provided and formatted correctly? **(2)** Is there a clear and proper use of language? **(1)** Effective report structure (chapters and sections) and layout **(2)**	**(6 pts)**		
Total Marks	**45**		

Verdict (Please tick) ☐ Accept ☐ Reject

Comments (**Reasons for Reject/Accept**)