

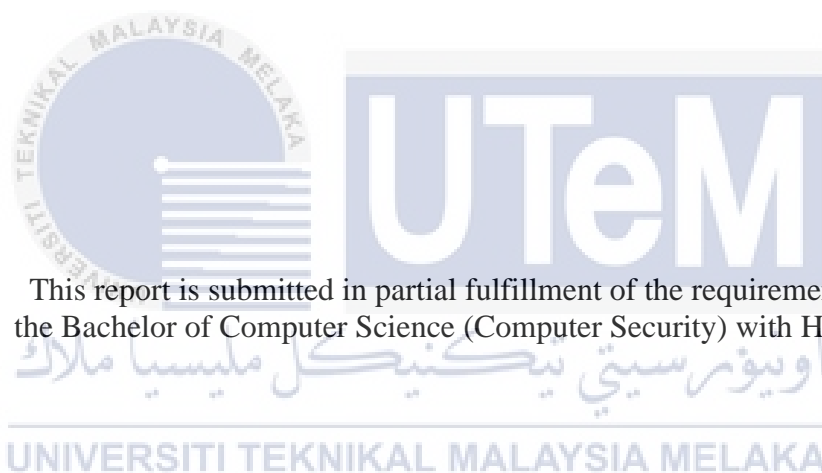
HYBRID DOCUMENT COPYRIGHT PROTECTION



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

HYBRID DOCUMENT COPYRIGHT PROTECTION

NOR FATIN SHAZWANI BINTI ADNAN



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2021

DECLARATION

I hereby declare that this project report entitled

HYBRID DOCUMENT COPYRIGHT PROTECTION

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : NOR FATIN SHAZWANI BINTI ADNAN DATE : 8 SEPT 2021



I hereby declare that I have read this project report and found
this project report is sufficient in term of the scope and quality for the award of
Bachelor of Computer Science (Computer Security) with Honours.

DEDICATION

This research is dedicated to my beloved father, who taught me to keep learning even in a difficult situation as long as there is a chance because knowledge is something precious we can have. It is also dedicated to my dearest mother, who taught me to never give up on what I am currently doing. She taught me that even the smallest progress is still considered progress and with the smallest progress I will be able to complete the task.



ACKNOWLEDGEMENTS

All praises to Allah with His Permission and Grace, I am able to complete this final year project report.

First of all, I would like to express my deepest gratitude to my supervisor, Prof. Madya TS Dr. Siti Rahayu Binti Selamat for her guidance, advice, encouragement, and supportive comments throughout the process of completing this project. Without my supervisor's help, I might not be able to complete this report and project successfully.

In addition, my utmost appreciation goes to my beloved parents, Mr. Adnan Bin Husain and Mrs. Norasila Binti Sanip, who have been giving me continuous support and motivation throughout my project. Indeed, without their ongoing support and encouragement, I would not be here.

Lastly, I would also like to take this opportunity to thank all my friends for lending hands every time I need it. Especially, Muhammad Hafiz Bin Jamil, Wan Nurin Jazmina Binti Wan Omar, and Amirah Nadhirah Binti Kamarulzaman, thank you for all the reminders, encouragement, care, guidance, and support. Thank you for inspiring me to finish this project. Words cannot express my gratitude for all your love and support.

ABSTRACT

The E-Learning system has made it easier for lecturers to share their digital lecture notes with the students. Students can access them from their lecturers anywhere and anytime. However, these digital lecture notes are vulnerable to illegal copy and unauthorized distribution because they do not have copyright protection. Therefore, to solve the problem, a QR code technology and a steganography technique have been applied as a mechanism of document protection. In this implementation, the information of the owner's file and the person who downloaded the file are stored and hidden in the QR Code and steganography image. In addition, a UTeM logo also was embedded as a watermark to provide multiple protection to the digital lecture notes. With this information, if the digital lecture notes are misused or distributed on a public platform without the permission of their lecturer, the user who distributed the files can be identified by scanning the QR code contained in that particular file. With that information, the owner of the digital lecture notes also can be proven.

ABSTRAK

Sistem *E-Learning* memudahkan pensyarah berkongsi nota kuliah digital mereka dengan pelajar. Pelajar boleh mendapatkannya dari pensyarah di mana sahaja dan pada bila-bila masa. Walau bagaimanapun, nota kuliah digital ini terdedah kepada salinan haram dan pengedaran yang tidak dibenarkan kerana mereka tidak mempunyai perlindungan hak cipta. Oleh itu, untuk menyelesaikan masalah ini, teknologi Kod QR dan teknik steganografi telah digunakan sebagai mekanisme perlindungan dokumen. Dalam pelaksanaan ini, maklumat fail pemilik dan orang yang memuat turun fail disimpan dan tersembunyi dalam Kod QR dan imej steganografi. Di samping itu, logo UTeM juga dimasukkan sebagai tera air untuk memberi perlindungan berganda kepada nota kuliah digital. Dengan maklumat ini, jika nota kuliah digital disalahgunakan atau diedarkan di platform awam tanpa kebenaran pensyarah mereka, pengguna yang mengedarkan fail boleh dikenal pasti dengan mengimbas kod QR yang terkandung dalam fail tersebut. Dengan maklumat itu, pemilik nota kuliah digital juga boleh dibuktikan.

TABLE OF CONTENTS

DECLARATION.....	ii
DEDICATION.....	iii
ACKNOWLEDGEMENTS.....	iv
ABSTRACT	v
ABSTRAK	vi
LIST OF TABLES	xi
LIST OF FIGURES	xiv
LIST OF ABBREVIATIONS	xviii
CHAPTER 1: INTRODUCTION.....	1
1.1. Introduction	1
1.2 Project Background	1
1.3 Problem Statement (PS)	2
1.4 Project Question (PQ)	2
1.5 Project Objective (PO)	2
1.6 Project Scope	3
1.7 Project Contribution	3
1.8 Report Organization	4
1.9 Summary	5
CHAPTER 2: LITERATURE REVIEW.....	6
2.1 Introduction	6
2.2 Digital Document	7
2.3 Protection Technique.....	8
2.3.1 Watermarking Technique.....	10
2.3.2 Steganography Technique.....	11
2.4 Basic of QR Code.....	12
2.5 Analysis of QR Code Applications in Information Security Perspective ...	16
2.6 Proposed solution	23
2.7 Summary	23
CHAPTER 3: PROJECT METHODOLOGY	24

3.1	Introduction	24
3.2	Methodology	24
3.2.1	Literature Review	25
3.2.2	Analysis.....	25
3.2.3	Formulate Hybrid Protection Technique.....	25
3.2.4	Embedded Protection Techniques into document.....	26
3.2.5	Implementation	26
3.2.6	Testing.....	26
3.3	Project Schedule and Milestone	27
3.3.1	Gantt Chart	27
3.3.2	Milestone.....	30
3.4	Summary	31
CHAPTER 4: ANALYSIS AND DESIGN.....		32
4.1	Introduction	32
4.2	Requirement	32
4.2.1	Software Requirement.....	32
4.2.2	Hardware Requirement	33
4.3	System Architecture Design.....	34
4.4	QR Code Generator Design.....	37
4.4.1	Data Input.....	38
4.4.2	Analyzing Input Data	39
4.4.3	Data Encoding.....	40
4.4.4	Creating Error Correction Codewords	46
4.4.5	Structuring Final Data	47
4.4.6	Converting Block into QR Matrix	47
4.4.7	Apply Mask Pattern	48
4.4.8	Apply Version and Format Information.....	48
4.5	Steganography Image Generator Design.....	48
4.5.1	Determine data and image.....	49
4.5.2	Data Encoding.....	49
4.5.3	Determine Image Properties.....	50
4.5.4	Extract RGB color from the image	50
4.5.5	Inject data to image	51
4.6	Design for Embedding Images into Document	51

4.6.1	Import Document	51
4.6.2	Determine Document Page to Insert The Images	52
4.6.3	Import Images	52
4.6.4	Embed The Images Into Document	53
4.7	Summary	53
CHAPTER 5: IMPLEMENTATION.....		54
5.1	Introduction	54
5.2	Software Development Environment Setup	54
5.2.1	Web Application Manager	54
5.2.2	Database Manager	55
5.2.3	Visual Studio Code (VS Code) Setup	55
5.3	Implementation.....	57
5.3.1	QR code Generator.....	58
5.3.2	Steganography Image Generator.....	71
5.3.3	Embedding The Images into Document.....	76
5.4	Summary	88
CHAPTER 6: TESTING.....		89
6.1	Introduction	89
6.2	QR Code Generator Testing	89
6.3	Steganography Image Generator Testing	96
6.4	Embed Image into Document Testing	101
6.5	QR Code Image Testing	106
6.5.1	Completeness of data.....	107
6.5.2	Usability.....	109
6.6	Steganography image Testing	114
6.6.1	Completeness of data	114
6.6.2	Imperceptibility	117
6.7	Summary	119
CHAPTER 7: PROJECT CONCLUSION.....		120
7.1	Introduction	120
7.2	Project summarization	120
7.3	Project contribution	121
7.4	Project limitation	121
7.5	Future works.....	122

7.6 Summary	122
REFERENCES.....	124



LIST OF TABLES

	PAGE
Table 1. 1: Summary of problem statement	2
Table 1. 2: Summary of project question.....	2
Table 1. 3: Summary of the project objective.....	3
Table 1. 4: Summary of the project contribution	3
Table 2. 1: Digital document protection technique	8
Table 2. 2: Data capacity of QR code (Espejel-Trujillo et al., 2012)	13
Table 2. 3: Summary of the QR code application in information security perspective	20
Table 3. 1: Gantt chart	27
Table 3. 2: Project Milestone.....	30
Table 4. 1: The list of software used with the description	33
Table 4. 2: The list of hardware used with the description	33
Table 4. 3 : Pseudocode for inserting required information into the QR code ..	39
Table 4. 4: Alphanumeric table.....	39
Table 4. 5: ECC level (Denso Wave Incorporated, 2021)	41
Table 4. 6: QR code version with maximum allowable capacity (Denso wave Incorporated, 2021)	41
Table 4. 7: Indicator mode for respective data type (Denso wave Incorporated, 2021)	42
Table 4. 8: Character count indicator according to the version and data type (Thonky, 2021).....	43
Table 4. 9: Break phrase into pairs	43
Table 4. 10: Data encoding based on respective data type	44
Table 4. 11: Current bit string of the example	44
Table 4. 12: Error correction codewords (Thonky, 2021)	44

Table 4. 13: Terminator is added.....	45
Table 4. 14: Arranged encoded data	45
Table 4. 15: Required bytes for the example	46
Table 4. 16: Encoded data in decimal and polynomial	47
Table 4. 17: ASCII Table.....	49
Table 4. 18: Example to encode data	50
Table 5. 1: Pseudocode for Data Input Process	59
Table 5. 2: Pseudocode for Analyzing Alphanumeric Data type	60
Table 5. 3: Pseudocode for Alphanumeric Encoding.....	60
Table 5. 4: Pseudocode for creating error correction codewords.....	61
Table 5. 5: Pseudocode to structure final data in a block	63
Table 5. 6: Pseudocode placement of finder pattern and separator	64
Table 5. 7: Pseudocode to place timing pattern and alignment pattern	65
Table 5. 8: Pseudocode to mask data.....	66
Table 5. 9: Pseudocode to apply version and format information.....	67
Table 5. 10: Pseudocode resizing logo image and set logo image transparency .	69
Table 5. 11: Pseudocode for determine data and image	72
Table 5. 12: Pseudocode for encoding message	73
Table 5. 13: Pseudocode to reset image properties	73
Table 5. 14: Pseudocode for extract rgb color of image inject data to image.....	75
Table 5. 15: Javascript Pseudocode to pass material and user id.....	79
Table 5. 16: Pseudocode to obtain user id and material location	80
Table 5. 17: Pseudocode to check the file information	80
Table 5. 18: Pseudocode for determining page to insert the images	81
Table 5. 19: Pseudocode for passing QR code image path and the resizing value	82
Table 5. 20: Pseudocode for checking image file type.....	82
Table 5. 21: Pseudocode for checking image file	83
Table 5. 22: Pseudocode to read png image stream	84
Table 5. 23: Pseudocode to determine color type of the image	85
Table 5. 24: Pseudocode define path for embedded document.....	87
Table 5. 25: Pseudocode for calling Output() function.....	87
Table 5. 26: Pseudocode to return output as pdf document to a new window ...	87

Table 6. 1: Properties for data completeness.....	107
Table 6. 2: Data completeness of QR code analysis	109
Table 6. 3: User Usability Testing.....	110
Table 6. 4: Analysis of user usability testing on QR code.....	113
Table 6. 5: Data completeness properties	114
Table 6. 6: Data completeness of steganography image analysis.....	117
Table 6. 7: Visibility Testing Analysis.....	118



LIST OF FIGURES

	PAGE
Figure 2. 1: Overview of literature review	6
Figure 2. 2: Classification of Watermarking Technique (Tiwari & Sharmila, 2017)	11
Figure 2. 3: Component of QR code (Pal & Kumar, 2021)	13
Figure 2. 4: Overview of QR code process (Tiwari, 2016)	14
Figure 2. 5: Encoding step (Tiwari, 2016)	14
Figure 2. 6: Decoding step (Tiwari, 2016)	15
Figure 3. 1: Methodology	24
Figure 4. 1: System Architecture	34
Figure 4. 2: Flowchart of the main module	35
Figure 4. 3: User interface of view student assignment submission	35
Figure 4. 4: User interface of student assignment submission	36
Figure 4. 5: User interface that contains lecture note and lab sheet	36
Figure 4. 6: ERD of the system	37
Figure 4. 7: QR code generator design	38
Figure 4. 8: Inject data into image design	48
Figure 4. 9: Embed images with document design	51
Figure 5. 1: Laragon setting	54
Figure 5. 2: Add phpMyAdmin folder to Laragon File	55
Figure 5. 3: VS Code extensions	56
Figure 5. 4: Setup Database Connection in VS Code	56
Figure 5. 5: Established Database Connection in VS Code	57
Figure 5. 6: System architecture	57
Figure 5. 7: Flowchart main process for the QR code generation	58
Figure 5. 8: Data module is structured in QR matrix	66

Figure 5. 9: Generated QR code	69
Figure 5. 10: QR code image with UTeM logo	71
Figure 5. 11: Flowchart of steganography image generation.....	71
Figure 5. 12: Flowchart for Embedding the images into a document	76
Figure 5. 13: Student user interface to download lecture note	77
Figure 5. 14: Student user interface to download lab material and submit assignment.....	77
Figure 5. 15: Lecturer user interface to manage their material	78
Figure 5. 16: Lecturer user interface to review open submission.....	78
Figure 5. 17: Lecturer user interface to review student submission	79
Figure 5. 18: Downloaded file	88
Figure 6. 1: QR code generator testing process.....	89
Figure 6. 2: Download lecture note from the student side	90
Figure 6. 3: Generated QR code for lecture note (student).....	90
Figure 6. 4: Generated QR code with UTeM's logo for lecture note (student) ..	90
Figure 6. 5: Download Lab document from the student side.....	91
Figure 6. 6: Generated QR code for lab document (student).....	91
Figure 6. 7: Generated QR code with UTeM's logo for lab document (student)	91
Figure 6. 8: Download student submission document from the student side	92
Figure 6. 9:Generated QR code for student submission document (student)	92
Figure 6. 10: Generated QR code with UTeM's logo for student submission document (student)	92
Figure 6. 11: Download lecture note from lecturer side	93
Figure 6. 12: Generated QR code for lecture note (lecturer)	93
Figure 6. 13: Generated QR code with UTeM's logo for lecture note (lecturer)	93
Figure 6. 14: Download Lab document from lecturer side	94
Figure 6. 15: Generated QR code for lab document (lecturer).....	94
Figure 6. 16: Generated QR code with UTeM's logo for lab document (lecturer)	94
Figure 6. 17: Download student submission document from lecturer side.....	95
Figure 6. 18: Generated QR code for student submission document (lecturer).	95
Figure 6. 19: Generated QR code with UTeM's logo for student submission document (lecturer).....	95
Figure 6. 20: Steganography image generator testing process	96

Figure 6. 21: UTeM's logo used as a container of hidden information	96
Figure 6. 22: Downloadable lecture note document UI from student's perspective	97
Figure 6. 23: Generated steganography image for lecture note document from student's perspective	97
Figure 6. 24: Downloadable lab document UI from student's perspective.....	97
Figure 6. 25: Generated steganography image for lab document from student's perspective	98
Figure 6. 26: Downloadable student's assignment document UI from student's perspective	98
Figure 6. 27: Generated steganography image for student's assignment document from student's perspective.....	98
Figure 6. 28: Downloadable lecture note document UI from lecturer's perspective	99
Figure 6. 29: Generated steganography image for lecture note document from lecturer's perspective	99
Figure 6. 30: Downloadable lab document UI from lecturer's perspective.....	99
Figure 6. 31: Generated steganography image for lab document from lecturer's perspective	100
Figure 6. 32: Downloadable student's assignment document UI from lecturer's perspective	100
Figure 6. 33: Generated steganography image for student's assignment document from lecturer's perspective.....	100
Figure 6. 34: Embed QR code image and steganography image into document testing process.....	101
Figure 6. 35: The user interface for a lecture note document that can be downloaded from the student's viewpoint	101
Figure 6. 36: Lecture note document with QR code image, watermark image, and steganography image embedded within it from the student's viewpoint ..	102
Figure 6. 37: The user interface for a lab document that can be downloaded from the student's viewpoint	102
Figure 6. 38: Lab document with QR code image, watermark image, and steganography image embedded within it from the student's viewpoint.....	103

Figure 6. 39: The user interface for student’s submission document that can be downloaded from the student's viewpoint	103
Figure 6. 40: Student’s submission document with QR code image, watermark image, and steganography image embedded within it from the student's viewpoint	104
Figure 6. 41: The user interface for a lecture note document that can be downloaded	104
Figure 6. 42: Lecture note document with QR code image, watermark image, and steganography image embedded within it from the Lecturer’s viewpoint	104
Figure 6. 43: The user interface for a lab document that can be downloaded from the lecturer’s viewpoint.....	105
Figure 6. 44: Lab document with QR code image, watermark image and steganography image embedded within it from the lecturer’s viewpoint.....	105
Figure 6. 45: The user interface for a student’s assignment document that can be downloaded from the lecturer’s viewpoint	106
Figure 6. 46: Student’s assignment document with QR code image, watermark image, and steganography image embedded within it from the lecturer’s viewpoint	106
Figure 6. 47: Information of 'Adlina Kadir' from Database.....	107
Figure 6. 48: Information of owner file and submission date from System	108
Figure 6. 49: Information of source file	108
Figure 6. 50: Student submission document named 'assignment_adlina.pdf'	108
Figure 6. 51: Result after scan the QR code	109
Figure 6. 52: User interface for message extraction.....	114
Figure 6. 53: Information of lecturer 'Nor Azman Abu' from Database.....	115
Figure 6. 54: File owner and submission date information from System	115
Figure 6. 55: The file source information.....	115
Figure 6. 56: The student’s submission document named 'assignment_adlina.pdf'	116
Figure 6. 57: Outcome of the QR code scanning	116
Figure 6. 58: Steganography Image.....	116
Figure 6. 59: Extraction of message in steganography image.....	117
Figure 6. 60: Zoom in document.....	118

LIST OF ABBREVIATIONS

UTeM	-	Universiti Teknikal Malaysia Melaka
UiTM	-	Universiti Teknologi Mara
Covid-19	-	Coronavirus Disease 2019
QR-code	-	Quick Response Code
LSB	-	Least Significant Bits
ISO/IEC	-	International Organization For Standardization/International Electrotechnical Commission
2D code	-	2 Dimensional Code
RSA	-	Rivest-Shamir-Adleman Encryption
VSS	-	Visual Secret Sharing
DWT	-	Discrete Wavelet Transform
HH	-	Higher Highs
HL	-	Higher Lows
LH	-	Lower Highs
LL	-	Lower Lows
DCT	-	Discrete Cosine Transform
HE	-	Histogram Attack
JPEG	-	Joint Photographic Experts Group
SVD	-	Singular Value Decomposition
ID	-	Identification
ERD	-	Entity Relationship Diagram
PHP	-	Personal Home Page
SQL	-	Structured Query Language
RAM	-	Random Access Memory

LCD	-	Liquid Crystal Display
IPS	-	In-Plane Switching
GHz	-	Gigahertz
PnP	-	Plug and Play
ECC	-	Error Correction Capability
Char	-	Character
RGB	-	Red, Blue, Green



CHAPTER 1: INTRODUCTION

1.1. Introduction

Nowadays, online learning is a common way used by the educational industry to deliver their contents to their students because all the country has been hit by the Covid-19 Pandemic. Therefore, all universities have their own online learning platform. For example, U-Learn and i-Learn are the online platforms used by UTeM and UiTM respectively. With the online learning platform, lecturers can upload their material to be used by their students to be accessed anywhere and anytime. This also provides a convenient environment for both of them. However, several issues should be considered to ensure the materials provided are protected and not misused by the students. Therefore, this project is proposed to protect the materials provided by lecturers from any misuse by their students. Hence, this chapter will explain the background, problem, objectives, and significance of the project.

1.2 Project Background

Online learning has long been practiced by many universities, but its uses have increased since the hit of Covid-19 Pandemic. As a result, teaching and learning have been done digitally and using an online learning platform. According to Abdul Rahman et al. (2020), the implementation of teaching and learning activities online can help reduce the risk of Covid-19 infection because it is carried out virtually. As such, the use of digital documents has also increased as it is facilitating the teaching and learning process. However, although the use of digital documents in an online learning platform facilitates all parties, some issues can occur, such as copyright issues and misuse of the lecturer's teaching materials. Therefore, it is important to digitally secure those digital documents from these issues. In the meantime, this project will use a QR code and a steganography image to store information about who download the

document, the owner of the document, and the source of the document as it is simple and reliable just for academic purposes.

1.3 Problem Statement (PS)

Since the Covid-19 pandemic, teaching and learning have been done online. So, all notes and study materials from the lecturer have been uploaded to the online learning platform to make it easier for students to download and study in their respective places. However, the possibility for cases such as misuse of the lecturer's material by distributing it on the public platform without the permission of their lecturer can occur. Nowadays, there are also many public online platforms to share assignments and notes like Coursehero, Quizlet, and Quizizz. Apart from that, some of the lecturers also less consent about the copyright issue because they trust if the material is shared in that platform (such as U-Learn), it will be just between the student only, not with other people. The problem here is, we do not know who is the real person that shares the lecturer's material, where it might cause some problem for the lecturers later. For example, someone can publish a book using the lecturer's study material content.

Table 1. 1: Summary of problem statement

PS	Problem Statement
PS1	Current online learning situation is causing the misused of the lecturer's material and do not know who is behind it.

1.4 Project Question (PQ)

Based on the problem statements listed in Table 1.1, three project question (PQ) are constructed as shown in Table 1.2.

Table 1. 2: Summary of project question

PQ	Project Question
PQ1	What copyright protection technique can be used for documents?
PQ2	How to protect the digital documents from any illegal activities?
PQ3	How to measure the effectiveness of the proposed protection technique?

1.5 Project Objective (PO)

The aim of this project is to secure the lecturer's material. Therefore, to be able to

solve the problem identified in Table 1.1 and to achieve the aim of this project, three project objectives (PO) are derived as shown in Table 1.3.

Table 1. 3: Summary of the project objective

PQ	PO	Project Objective
PQ1	PO1	To analyze copyright protection techniques for documents
PQ2	PO2	To formulate hybrid copyright protection technique
PQ3	PO3	To evaluate the effectiveness of the proposed protection technique

1.6 Project Scope

The main purpose of this project is to generate a QR code and a steganography image that contains information about the user that downloads the document and embeds it into the lecturer material in order to find out who holds that document in case the document is misused by someone. The document also will be embedded with UTeM's logo as a watermark. A system has been developed for the testing part. The document supported to upload to the system is digital materials and the format of the document is .pdf. This project is targeting students and lecturers as this system is an online learning platform. This project will focus on how to solve the problem as stated in the problem statement with the use of QR code technology, steganography, and watermarking technique.

1.7 Project Contribution

Based on the problem statement, project question, and project objectives listed in Table 1.1, Table 1.2, and Table 1.3 respectively, three project contributions (PC) are constructed as shown in Table 1.4.

Table 1. 4: Summary of the project contribution

PS	PQ	PO	PC	Project Contribution
PS1	PQ1	PO1	PC1	Techniques to protect the digital documents from illegal activities or misuse activities
	PQ2	PO2	PC2	A protected document that embedded with QR code, watermark image and steganography image
	PQ3	PO3	PC3	The identity of the document's ownership

1.8 Report Organization

This report consists of seven chapters namely Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 Project Methodology, Chapter 4 Analysis and Design, Chapter 5 Implementation, Chapter 6 Testing, and Chapter 7 Conclusion.

Chapter 1: Introduction

This chapter explains the introduction of this project. It consists of a problem statement, project question, project objective, project scope, and project contribution. This chapter also includes report organization which will summarize each of the chapters in this project.

Chapter 2: Literature Review

This chapter contains previous research and project explanations with supporting papers, journals, books, and websites.

Chapter 3: Project Methodology

In this chapter, the research methodology is explained. The processes of the methodology used in this project are described including project milestones and Gantt chart.

Chapter 4: Analysis and Design

This chapter contains the system architecture design, some user interface design, ERD design, the QR code generator design, inject ownership information to an image design, and embedding images into document design. The details of each design are explained.

Chapter 5: Implementation

This chapter will explain about the project implementation including the activity involved in the implementation process.

Chapter 6: Testing

This chapter involves testing on the QR code generator, steganography image generator, and testing on the images with several parameters.

Chapter 7: Conclusion

This chapter provides the conclusion of the project. It includes the project summarization, project contribution, project limitation, and suggestions for future works.

1.9 Summary

This chapter introduced the project background, problem statement, project question, project objective, project scope, project contribution, and report organization. The background study explains the use of online platforms and their relationship with the increased use of digital documents and why it is necessary to protect the digital materials from the problems that can be faced later as well as what technology and technique can be applied to prevent that problem from occurring. Next, Chapter 2 will elaborate on the literature review on the related topics of the project.



CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Chapter 1 explained the problem, objectives, and expected contributions of the project. In this chapter, the literature review related to digital document protection, the application of QR Code technology, and document protection techniques will be discussed. The information will be obtained from related journal articles, proceedings, books, and websites. This chapter aims to provide the information that is related to the document copyright protection including information on the digital document, the protection technique, the basic of QR code and its process, analysis of the QR code application, and to proposed a solution for the problem statement stated as in Chapter 1. The overview of this chapter is illustrated in Figure 2.1.

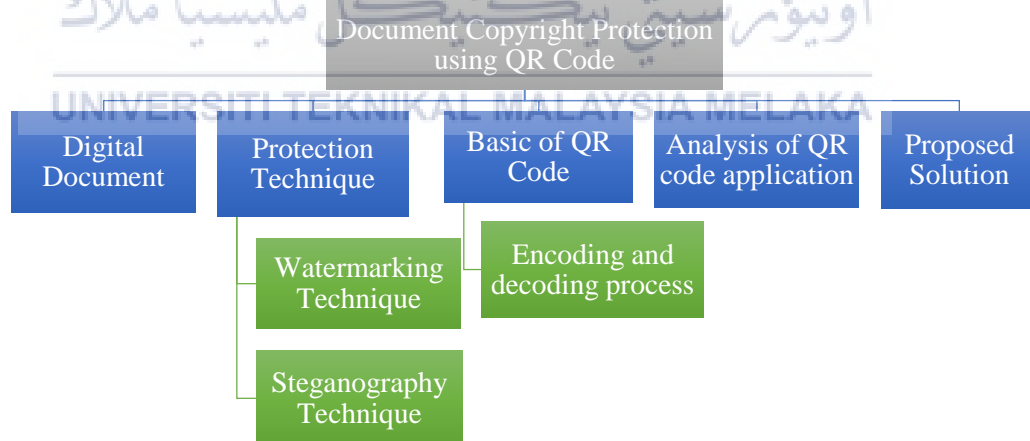


Figure 2. 1: Overview of literature review

2.2 Digital Document

Nowadays, digital documents are a common document used by many organizations because it is paperless and can be accessed anywhere such as through a computer or even smartphone. Apart from that, the use of digital documents can save a lot of time, especially when searching for an important document. Besides, physical space also can be reduced. According to the Society of American Archivists (2020), a digital document is “Information created on none electronic media, typically text or images on paper or film, and converted to an electronic format that can be stored and manipulated by a computer.”.

Particularly, it is a text document that is stored online and can be opened with any electronic device that supports the digital document type. There are several digital document types which are Word Documents (.doc or .docx), Portable File Document (PDF), Spreadsheet (.xls or .xlsx) and Powerpoint (.ppt or .pptx). Due to the exponential growth of the internet, there are many issues related to digital documents are arisen.

Zhou and Yang (2010) stated that most academic organizations provide resources to their reader via a network as it is a better way for sharing nowadays. Although there are platforms that easy to share resources, the issue that occurs is copyright protection of the resources. Besides, Kim et al. (2014) voiced out an issue related to scanned books where there are many illegal copies of the scanned book are distributed. This shows that not only the academic industry is having this issue but other industries such as the publishing industry also facing this kind of issue. Seeing the use of digital documents or products has been arising, there are also issues such as manipulated information, copyright infringement (Dang et al., 2019), document forgery (Ahvanooy et al., 2018), and digital copies (Aru and Ananaba, 2018).

In view of the fact that there are many issues related to digital documents, it is crucial to protect than let them continue to happen. If these issues continue to take place, many parties can suffer losses. Hence, there are some protection techniques that can be applied to digital documents which will be described in the next section.

2.3 Protection Technique

There are various issues related to the digital document that has been voiced out. Therefore, it is important for it to be protected from letting it continuing to occur which can cause significant losses by some party. To protect the digital document, there are several techniques that have been used by researchers. The techniques for digital document protection are summarized in Table 2.1.

Table 2. 1: Digital document protection technique

Research Title and Authors	Technique	Advantages	Disadvantages
Robust Visible Digital Stamp for Instant Documents Authentication and Verification. (Hassan and Hussein, 2020)	Watermarking	<ul style="list-style-type: none"> • No network required. • Secure users' information privacy because no third party is used. • Secure document content. 	<ul style="list-style-type: none"> • Susceptible to attack if not implemented properly.
Copyright Protection and Distribution System for Scanned Books/Comics. (Kim et al., 2014).	Watermarking	<ul style="list-style-type: none"> • Prevent illegal distribution • Prevent copyright infringement • Protect authors' rights. 	<ul style="list-style-type: none"> • Required many processes and modules.
Copyright Protection of E-Government Document Images Using Digital Watermarking. (Al-Haj and Barouqa, 2017)	Watermarking	<ul style="list-style-type: none"> • Provide invisible watermark • Hardly to detect by naked eyes. • Better for protect the document ownership. 	<ul style="list-style-type: none"> • Low embedding watermark capacity, only small image can be a watermark image
Steganography of Encrypted Messages Inside Valid QR Codes. (Alajmi et al., 2020).	Steganography	<ul style="list-style-type: none"> • The simplicity of the method. • Better information security. • Better for hiding credential data. 	<ul style="list-style-type: none"> • Susceptible to attack if not implemented properly.

Research Title and Authors	Technique	Advantages	Disadvantages
QR code Authentication System for confidential (digital Mark sheet) Encrypted data hiding and retrieval (Decryption). (Chavan et al., 2016).	Steganography	<ul style="list-style-type: none"> Requires password from the owner. Only an authorized person can know the mark. Better for information hiding and protect the data. 	<ul style="list-style-type: none"> Unfamiliar algorithm Complicated because use a combination of three types symmetric key.
Detection Of Forgery and Fabrication In Passports and Visas Using Cryptography and QR Codes. (Chemana Shaik, 2021).	Cryptography	<ul style="list-style-type: none"> Apply two types of encryption methods. Provide robustness. 	<ul style="list-style-type: none"> Susceptible to attack if not implemented properly.
QR code and transport layer security for licensing documents verification. (Wibiyanto and Afrianto, 2018)	Cryptography	<ul style="list-style-type: none"> Provide a layer of security. License documents are harder to be forged. 	<ul style="list-style-type: none"> Susceptible to attack if not implemented properly.
An Improved Digital Watermarking Technology Based on QR Code. (Zhang and Meng, 2012)	Watermarking	<ul style="list-style-type: none"> Increases robustness of arbitrary rotation angle and several attacks. 	<ul style="list-style-type: none"> The quality image of the QR code is reduced.
Copyright Protection for Online Text Information. (Mir and Khan, 2020)	Watermarking and Cryptography	<ul style="list-style-type: none"> Watermark is generated along with author id and encrypted using AES encryption. Use the most secure encryption. Better for text document copyright protection. 	<ul style="list-style-type: none"> Complex encryption. Challenging to implement in software.

Research Title and Authors	Technique	Advantages	Disadvantages
A Robust Digital Watermarking Algorithm for Text Document Copyright Protection based on Feature Coding (Iqbal et al., 2019)	Watermarking	<ul style="list-style-type: none"> • Robust on the different attack. • Have improvement in embedding capacity (from bytes to kilobytes). • Better for text document copyright protection. 	<ul style="list-style-type: none"> • complicated architecture.

Based on the analysis of the digital document protection technique in Table 2.1, it is found that there are three protection technique which are Watermarking, Steganography and Cryptography. Among the three, the well-known techniques in addressing this copyright-related issue are a Watermarking technique and a Steganography technique. These techniques will be discussed more in the next subsection.

2.3.1 Watermarking Technique

Watermarking is one of the techniques to prove ownership of the document or asset. According to Patel and Tahilraman (2016) watermarking is the technique of inserting the data in the host where the host can be an image, video, audio, speech, and text. Watermarking is used in various places such as documents, images, audio, and video. Since nowadays everything is store digitally, a watermark is one of the important things because, without a watermark, all value documents or assets can be susceptible to unauthorized use. There are several classifications of watermarking which are domain-based, perception-based, and document-based as depicted in Figure 2.2.

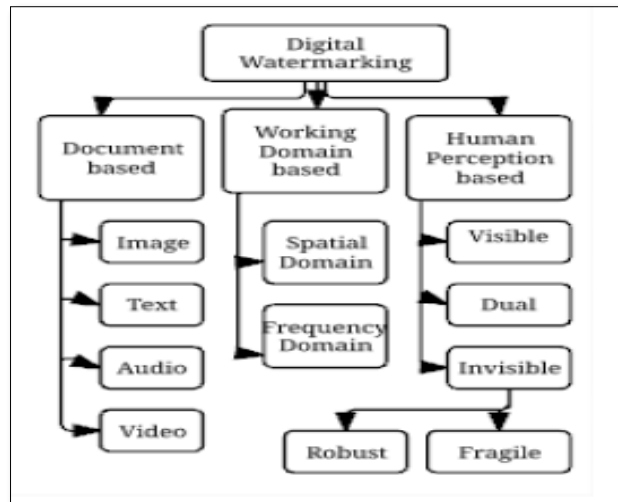


Figure 2. 2: Classification of Watermarking Technique (Tiwari & Sharmila, 2017)

Figure 2.2 depicts the document-based classification on the document types such as image, text, audio, and video. In working domain-based, the classification is based on the spatial domain and frequency domain while in human-perception-based, the classification is based on visible, dual, and invisible wherein the invisible classification is divided into robust and fragile. In watermarking technique, an image is usually used to be inserted into their document or asset as a watermark. This includes a QR code method because once the QR code is generated, it will be formatted in an image format such as .jpeg or .png. The use of QR code in the watermarking technique is more on a public matter and the owner wants people or outsiders who have that document or asset to know the real owner of it. Basically, the use of a QR code as a watermark is to prove the ownership of a document publicly and to protect the carrier (Rhazlane et al., 2017). Another technique for protecting a document is a steganography technique which is will be described in the next subsection.

2.3.2 Steganography Technique

The phrase “Steganography” is come from Greek words which are “Stegano” and “Graphy” where when it is combined, the meaning will be “cover writing” (Kadhim et al., 2018). Basically, steganography is a technique to cover the presence of the messages and more on hiding confidential data from an outsider or unauthorized users. There are several types of steganography which are image, audio, and video steganography.

Image steganography is hiding information which can be an image, video, or text into another image that will be the cover of the information (Subramanian et al., 2021). This type of steganography technique can protect the information from being tampered (Amarendra et al., 2019). Kumar et. al. (2021) stated that higher capacity is demanded in an image steganography technique for better imperceptibility. There are several types of data hiding methods in image steganography which are least significant bits (LSB), masking, filtering, and transformation (Thampi, 2014). LSB is the most common method used by many researchers because of its simple approach to hide the information in a cover image. Image steganography can also be attacked. For example, known carrier attack, steganography only attack, known message attack, and known steganography attack.

Just like the watermarking technique, the steganography techniques can also apply the QR code technology because the QR code can be used as a message carrier (Hassanein, 2014) and it is also usually used as a container of a secret message (Alajmi, 2020). Unlike the watermarking technique, the steganography technique is more about hiding the secret message and protecting the hidden message from disclosure but this technique can be used in this project to hide the ownership information in an image. In the next section, the basics of QR codes and their process will be discussed.

2.4 Basic of QR Code

QR code stands for Quick Response Code and is referred as two dimensional (2D) Code. QR code is made up of a square-shaped pattern which contains data such as link or character. It was invented by Denso Wave, which was one of the Japanese Group (Toyota) in 1994 and was approved by the International Standard (ISO/IEC 18004) in June 2000 (Mantoro et al., 2015). QR code is also made up of black and white modules where the encoded data is represented (Suwito et al., 2017). Nowadays, QR Code is widely used in daily life as it only needs a mobile device that has a QR code reader to redirect the user to a website, video, or anything anywhere and anytime. The main feature of the QR Code is it has a high capacity of data encoding where the data can be store vertically and horizontally. The data capacity of the QR code is represented in Table 2.2.

Table 2. 2: Data capacity of QR code (Espejel-Trujillo et al., 2012)

QR Code Data capacity	
Numeric only	Max. 7,089 characters
Alphanumeric	Max. 4,296 characters
Binary (8 bits)	Max. 2,953 bytes
Kanji, full-width Kana	Max. 1,817 characters

A QR code can store information or payload according to the data type. Besides, a QR code is more size efficient compared to other containers such as colored images because it is presented in a binary image (Alajmi et al., 2020). Apart from that, there are several components of the QR Code which are represented in Figure 2.3.

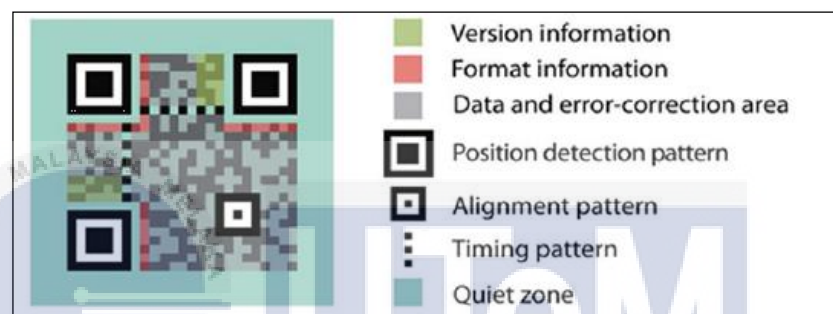
**Figure 2. 3: Component of QR code (Pal & Kumar, 2021)**

Figure 2.3 shows the components of the QR code and these components can be described as follows:

- Version information contains information of which QR code version is the QR code.
- Format information contains encoded pattern information that allows the rest of the region to be decoded.
- Data and error-correction areas are the areas where the encoded actual data have been located.
- Position detection pattern or also known as finder pattern (Ali and Farhan, 2020) is used to find the correct direction of the QR Code.
- An alignment pattern ensures that the QR code can be read, even if it is skewed or at a different angle.
- A timing pattern is an L-shaped line that lies between the three squares (position detection pattern) that helps the QR code reader find the width of the code or data matrix dimension.

- A quiet zone is an empty border around the outside of the QR code where it is used to ensure the QR code can be readable.

Apart from that, a QR code also consists of an encoder and decoder which is used to encode and decode the data. Figure 2.4 shows an overview of the QR code process for a text message.



Figure 2. 4: Overview of QR code process (Tiwari, 2016)

Figure 2.4 shows the general process of encoding and decoding for a text message. During the encoding process, a text message will go through the encoding process where the encoder will transform the text message into a QR code and after the QR code is scanned by a QR reader, the QR code will go through the decoding process where the decoder will decode the message into a readable text. The details of the process will be described in Figure 2.5 and Figure 2.6.

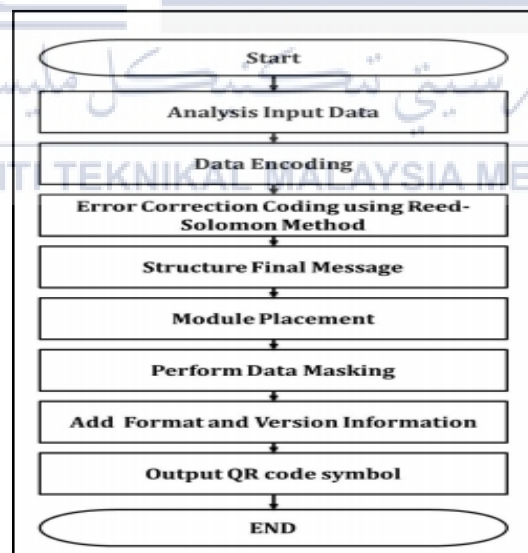


Figure 2. 5: Encoding step (Tiwari, 2016)

To transform data into a QR code, there are some steps that need to be followed. Figure 2.5 shows the encoding step to encode the data into a QR code. The details of the steps are explained below:

- Analysis input data: During this step, the input data will be analyzed according to data type whether it is numeric, alphanumeric, binary, or kanji. Then, the input will be transformed into bits.
- Data encoding: In this step, the transformed bits in the previous step are encoded.
- Error correction coding: A Reed-Solomon error correction is used here to generate error correction codewords. This process is to ensure that if the data cannot be read properly, the errors can be corrected.
- Structure final message: In this process, the data codewords and error correction codewords in the previous step are being structured in a block.
- Module placement: After the data codewords and error correction codewords have been structured in a block, they must be organized in a matrix form.
- Data masking: In this step, an eight-mask pattern will be applied to the QR code matrix to make the QR code more readable by the QR reader.
- Format and version information: This is the last step of the encoding process. In this step, the QR code is being formatted and becomes a readable QR code as an output.

If there is an encoding process, there must be a decoding process too. The steps of the decoding process are shown in Figure 2.6.

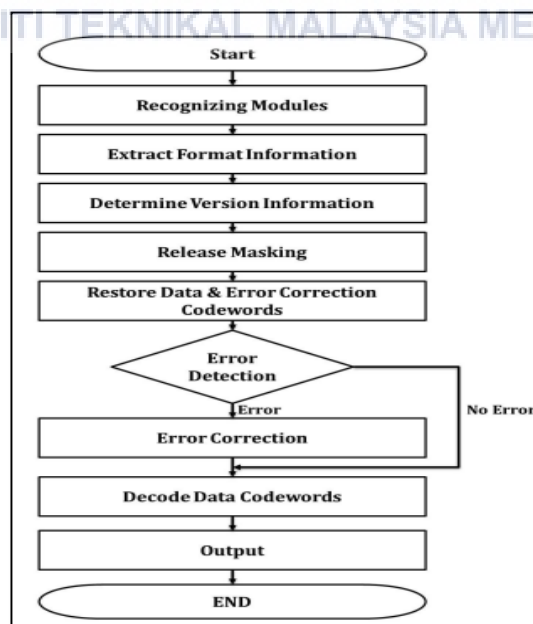


Figure 2. 6: Decoding step (Tiwari, 2016)

Figure 2.6 shows how the QR code is being decoded. Firstly, after the QR code is being scanned by a QR reader, the QR reader will:

- Recognize module: Each QR code has its own module and the module size depends on how many characters are encoded. In this process, the module is being recognized as either it is dark or light. The dark will represent 1 and white will represent 0.
- Extract format information: In this step, the format information is being extracted, masking pattern is released and error correction is applied to the format information.
- Determine version information: Each QR code has its own version and it depends on how long the character is being stored in the QR code. In this step, the information version is being determined.
- Release Masking: In this step, the mask is released by XORing the encoding section bit pattern with the mask pattern. The mask pattern here is referred from the format information.
- Restore data and error correction codewords: the codewords of data and error correction are restored here.
- Error detection and correction: In this step, the errors are identified based on the error correction codewords. If there is any error detected, it will be corrected.
- Decode data codewords: Lastly, the data codewords will be divided into segments according to the data type (numeric, alphanumeric, etc) and character count indicator. Finally, the output is decoded into readable text.

The QR code is the simplest to use as we just need a smartphone with a QR reader application. Apart from that, it is also reliable for many purposes such as academic, advertising, marketing (Asare et al., 2015), and many more. The next section will analyze the QR code applications in information security perspective.

2.5 Analysis of QR Code Applications in Information Security Perspective

Hassan and Hussein (2020) stated that QR code is the fastest way and consumes less cost in transferring data. It is widely utilized on a daily basis since almost all people in the world are having a smartphone. With a smartphone, a QR reader can easily be downloaded and the user can simply decode the information embedded in the

QR code by scan the QR code. There are many applications of QR code being used by researchers. The QR code application in an information security perspective will be described in next the paragraphs.

Manimekalai and Bakkiyalakshmi (2017) proposed a way to hide data in QR code with the combined concepts of steganography and cryptography. Firstly, a secret message is created and is embedded in a QR code. Then, the QR code with the hidden message is encrypted and embedded in a cover image by using the least significant bits (LSB) insertion technique which later will be a stego image. The advantages of this research are they provide better confidentiality and security for the message while the disadvantages are the hidden data may be lost if the format of the cover image is changed.

Mendhe et al. (2018) proposed a 3-layered architecture system for information security. In the first layer, the RSA encryption is used to encrypt the secret message. In the second layer, the encrypted message is inserted into a QR code which later will be converted as an image and lastly, the QR code image will be encoded behind mask image where a random initialized pixel image will be the cover of the QR code image. The researchers believe that RSA is the best for encryption of information in terms of security, flexibility, and performance among others. Even though there are other algorithms that are competent but the majority of them have a memory usage and encryption performance trade-off. This system provides security in each of the layers where the researchers able to enhance the security of the digital information with the combination of steganography and cryptography.

Ashwini et al. (2021) proposed a standard multi-color QR code based on texture patterns and text steganography to hide data. The researchers stated that some of the data was store directly to the QR code where it is not secure. In order to secure the data, a visual secret sharing scheme (VSS) is used before it is stored in the QR code. VSS is a method that allows for secret image sharing. As a result, the researchers able to improve two aspects in their proposed idea, which are security and partitioning technique.

Dang et al. (2019) proposed an invisible blind watermarking where a QR code is embedded into document images type based on discrete wavelet transform (DWT). The researchers use the DWT approaches because they believe that almost all invisible types of watermarking techniques rely on DWT. In their proposed idea, the original document image will firstly go through the noise reduction process before proceed to level 2 which is HH sub-band process by DWT. Sub-band is used because the human visual system is more sensitive towards LL sub-band. The use of HH sub-band also to ensures that the embedded watermark retains better image quality. The QR code watermark is then embedded into HH2 by modifying the HH2 coefficients. Lastly, the watermarked document is obtained after applying the inverse DWT. As a result, the researcher's proposed idea can work well in every document image type and their idea are also robust towards various digital image watermarking attacks.

Huang et al. (2020) proposed the use of two watermarking types with the application of QR code. In this paper, the DCT is applied where it is the key component of image compression and there are two watermarks are embedded into the color images. The first watermark is the QR code that contains the desired copyright information and the second watermark is a binary random sequence. There are several attacks tested on the watermark image to check for the watermark robustness. The attack that they have applied to the watermarked color image is histogram equalization (HE) attack and JPEG compression attack. According to the attack simulation, it is found that the QR code is still readable even after the color image is applied with several attacks. This shows that the copyright information still remains in the QR code and will be visible to the user once the QR code is scanned. The proposed method has successfully created a robust watermarked and improved copyright protection mechanism where it can be used in other document-based such as text documents.

Arkah et al. (2019) proposed a solution for document authenticity. In their research, a QR code will be generated multiple which each will contains a digital signature of the document. The researchers used a color map as a digital signature. The digital signature is extracted from the color contains in the document. The extracted color then will go through some process to be a color map and it will be a digital signature of the document. Once the digital signature has been inserted into the

multiple QR codes, the QR code is then stamped to the document. As a result, the proposed method can identify whether the document has been tampered with or not.

According to Li et al. (2017), the application of QR code for the anti-counterfeit scheme is an effective method that can be utilized. The approach used by the researchers along with the QR code application is DWT-SVD. This approach can give a better image quality. In their research, the data about the copyright owner is produced in the form of an image. The image is then will be inserted into the QR code and lastly become a watermark to overcome the counterfeiting issue. As a result, it is found that this method still able to read copyright information after several attacks are applied. This shows that this method is resistant to some attacks.

Saraswati et al. (2017) proposed QR Code watermarking with DWT and Counterlet Transform approach for authentication purposes. In their research, a logo or a watermark image is transformed into a binary image. The binary image is then will be embedded into the QR code image and produced a watermarked QR code image. The watermarked QR code image is produced in low quality but it is still in an acceptable quality as the QR reader can still read the QR code image. As a result, the watermarked QR image is still able to be read by the QR reader, and this idea able to perform authentication operations.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The analysis can be summarised in Table 2.2. In Table 2.2, the title, author, including technique, a summary of the paper, advantages, and disadvantages are listed. With this analysis, it is found that both techniques are suitable for this project.

Table 2. 3: Summary of the QR code application in information security perspective

Title and Author	Technique	Summary	Advantages	Disadvantages
Hide and Seek: A New Way to Hide Encrypted Data in QR Code Using the Concepts Steganography and Cryptography. (Manimekalai and Bakkiyalakshmi, 2017)	Steganography	Apply a combination concept of steganography and cryptography where a secret message is embedded into the QR code. The QR code is then encrypted and embedded once again in an image as another container.	<ul style="list-style-type: none"> • Provide a layer of protection. • More on hiding an information 	<ul style="list-style-type: none"> • If the encryption system is revealed, the steganography system will fail
Secure QR-Code Based Message Sharing System Using Cryptography and Steganography. (Mendhe, Gupta, and Sharma, 2018)	Steganography	3-layered architecture is used where RSA is used to encrypt a secret message, the encrypted message is embedded into a QR code and the QR code image will be covered with a random initialized pixel image.	<ul style="list-style-type: none"> • Provide some protection on the hidden message. 	<ul style="list-style-type: none"> • The applied steganography will fail if the encryption system is exposed
A Survey On Novel Approach For Data Hiding Under Qr Code Using Visual Secret Sharing. (Ashwini et al., 2021).	Steganography	First, texture patterns and text steganography is used to hide the data, and then, a visual secret sharing (VSS) scheme is used to encode the secret QR code.	<ul style="list-style-type: none"> • New technology • Better in data concealment. • Better for data hiding. 	<ul style="list-style-type: none"> • Information may loss due to the changes in the aspect ratio. • Unfamiliar scheme

Title and Author	Technique	Summary	Advantages	Disadvantages
A Blind Document image watermarking approach based on Discrete Wavelet Transform and QR code embedding. (Dang et al., 2019).	Watermarking	A QR code is embedded into a digital image using a DWT approach. The original digital image will go through the DWT process until a watermarked document is obtained.	<ul style="list-style-type: none"> • Almost impossible to detect the invisible watermark with naked eyes. 	<ul style="list-style-type: none"> • Longer compression time. • A low compression rate may result in a blurry image.
Multi-Purpose Watermarking with QR Code Applications. (Huang et al., 2020)	Watermarking	Two watermark type is embedded to a colored image. One is a QR code that contains information about the copyright and the second watermark is a binary random sequence. It is found that the QR code is still readable after several attacks are tested on the colored image and this proposed idea is successful in protecting the image document.	<ul style="list-style-type: none"> • Error correction of the QR code makes the QR code still readable. • Better for verifying the ownership. 	<ul style="list-style-type: none"> • The QR code might be exploited since the simulation attack on the QR code is not tested.
Research on Anti-counterfeiting Technology Based on QR Code image Watermarking Algorithm (Li et al., 2017)	Watermarking	In their research, the data about the copyright owner is produced in the form of an image. The image is then will be inserted into the QR code and lastly become a watermark to overcome the anti-counterfeiting issue.	<ul style="list-style-type: none"> • Strong robustness. • Able to protect copyright information • Better for proofing the ownership and prevent counterfeiting 	<ul style="list-style-type: none"> • Complexity in algorithm

Title and Author	Technique	Summary	Advantages	Disadvantages
Digital Color Documents Authentication Using QR Code Based on Digital Watermarking. (Arkah et al.,2020)	Watermarking	Multiple QR code is generated which each will contains a digital signature. The multiple QR code is then stamped to the document and as a result the proposed method able to identify whether the document is authentic or not.	<ul style="list-style-type: none"> • Better for alteration or tampered detection. Able to detect document authenticity 	QR code simulation attack is not conducted.
Research on Anti-counterfeiting Technology Based on QR Code image Watermarking Algorithm (Li et al., 2017)	Watermarking	In their research, the data about the copyright owner is produced in the form of an image. The image is then will be inserted into the QR code and lastly become a watermark to overcome the anti-counterfeiting issue.	<ul style="list-style-type: none"> • Has strong robustness.. • Able to protect copyright information Better for proofing the ownership and prevent counterfeiting 	Complexity in algorithm
QR Code Watermarking Algorithm Based on DWT and Counterlet Transform for Authentication. (Saraswati et al., 2017)	Watermarking	A logo or watermark image is transformed into a binary image. The binary image is then embedded into the QR code and becomes a watermarked QR image. The watermarked QR image can be read by the QR reader even the visual quality is low.	<ul style="list-style-type: none"> • Able to prove authenticity 	The DWT approach degrades the QR code image quality.

2.6 Proposed solution

Based on the related work, the suitable method for this project is watermarking and steganography technique. According to Mir and Khan (2020), an effective method for copyright protection is digital watermarking. Besides that, Li et al. (2017), Huang et al. (2020), Rhazlane et al. (2017), and Al-Haj and Barouqa (2017) also utter the same opinion. While Vyas and Dudul (2020) stated image steganography entails concealing secret information in a cover image so that it cannot be easily identified. Hence, a suitable technique to protect the material in the developed system is a watermarking and steganography technique. The steganography technique is chosen because it can be as another way to protect the ownership information in case the watermark image is removed. The image that will be used as a watermark is the QR code image which will contain the ownership information and information of the user who downloads the document and a UTeM's logo. The image that will be a steganography image is also the UTeM's logo but will be manipulated to contains the information as in the QR code image. In addition, the application of the QR code itself is the simplest to used and it is also reliable just for academic purposes. Just by using a smartphone, the ownership of the document can be proven.

2.7 Summary

This chapter describes details and related information of the technique to protect the digital document. Several techniques have been defined, the basic of the QR code has been explained, the challenges related to the online document has been discussed and an analysis of the document protection technique with the QR code application has been conducted. A proposed solution has been given based on previous research. The next chapter will discuss the methodology that is going to be used in this project.

CHAPTER 3: PROJECT METHODOLOGY

3.1 Introduction

Chapter 2 explained the document copyright protection, digital document, the protection technique, the basics of QR code with its process, analysis of the QR code application, and a proposed solution. In this chapter, the methodology of the project will be explained. The methodology describes the processes that are carried out in this project. The processes involved in this project are literature review, analysis, formulate hybrid copyright protection technique, embedded protection techniques into document, implementation, and testing. The milestones and Gantt chart of the project are also included in this chapter.

3.2 Methodology

A methodology is one of the important things in a project as it helps in what to do after one process is done. Table 3.1 shows the processes involved in the project.

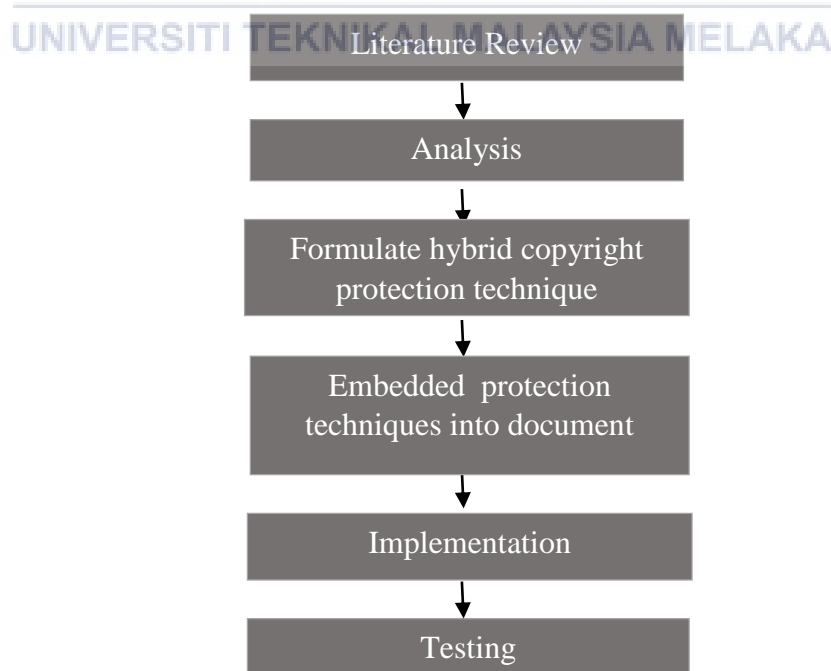


Figure 3. 1: Methodology

Figure 3.1 shows six processes to be carried out to complete this project. The processes are Literature Review, Analysis, Formulate hybrid copyright protection technique, Embedded protection techniques into document, Implementation, and Testing. The details of each of the processes are explained in the next section.

3.2.1 Literature Review

This is the initial activity and preparation for protecting copyright document. During this process, a literature review regarding this project title is conducted. Since this project is about document copyright protection, a study related to this matter is needed to get more knowledge on how and what the system should have in order to protect the digital document there. The protection technique such as watermarking and steganography technique, the properties of the QR code are also studied since it is the main module, even how the QR code can be generated, how the data can be converted to QR code, and many more are studied.

3.2.2 Analysis

In this process, all the gathered information in the literature review process is analyzed. The analysis that is conducted in this process is an analysis of the document copyright protection technique. During this analysis, it is found that there are three techniques that can be used for document protection which are Watermarking, Steganography, and Cryptography. The second analysis is the analysis of the Watermarking and Steganography technique with the QR code application. This analysis is conducted because the two stated techniques are the most technique that uses the QR code technology and during the analysis, it is found that both Watermarking and Steganography technique are the suitable technique to be used since its purpose is among the most related to this project.

3.2.3 Formulate Hybrid Protection Technique

During this process, the QR code generator process, and inject the ownership information into an image has been designed. This include which user interface of the system that this module is going to be located, what is the required information that is going to be used in the module, how to generate the QR image and steganography image, where to saved the images, and how to pass the images to the next module.

3.2.4 Embedded Protection Techniques into document

This is another important process after formulating the QR code generator and injecting ownership information into an image. To make a document with a QR code image, steganography image and watermark image embedded within it, it first needs to be structured. There are several information that needed to be collected which are current user information who is downloading the document, for example, ID number, user name, and user email, material selected to download by the current user, for example, lecture note or lab sheet, material selected information and properties for example, the lecture note and lab sheet owner name, its source, width and size of the QR code image, steganography image and watermark image to embed into the selected material, position to locate the images, page to locate the images and where to stored the embedded material.

3.2.5 Implementation

In this activity, the implementation of the system module is conducted. A PHP programming language will be used to write the code and all the system modules are created including integrating the QR code generator module, inject ownership information into an image module and embedded the QR code image, watermark image and steganography image into the document module with the system. During this process, it is important to make sure all the systems function and integrated function works well without a problem to avoid problem occur later in the testing part.

3.2.6 Testing

In this activity, each module of the system will be tested. The success of generating the QR code image, generating the steganography image and embedding the generated QR code image, generated steganography image and watermark image into the digital document is tested to ensure that it works as planned. Testing with several parameters is also performed.

3.3.2 Milestone

Milestone is used to observing and managing the project from the beginning until the end of the project. Table 3.2 shows the milestones for this project.

Table 3. 2: Project Milestone

Activity	Output	Completion Date
Literature Review <ul style="list-style-type: none"> Find and study for related work on the document copyright protection and QR code 	Literature review	Week 2 (22 March 2021)
Analysis <ul style="list-style-type: none"> Research about document copyright protection technique. Research about QR code. 	Identify suitable document copyright protection technique	Week 4 (5 April 2021)
Design <ul style="list-style-type: none"> Design a user interface, database Design QR code & steganography image generator Formulate QR Code & steganography image Generator Identify required information to generate the QR code & steganography image Design embedding images to document Identify required information to embed QR code & steganography image to the document Structure the images to be embedded into the document 	QR generator process design, Steganography image generator Embed QR image, steganography image and watermark image to document process design	Week 7 (26 April 2021)

Activity	Output	Completion Date
<p>Implementation</p> <ul style="list-style-type: none"> ▪ Develop prototype system ▪ Integrate the QR generator with the system ▪ Integrate the steganography image generator with the system • Integrate the generated QR code image, steganography image and watermark image with the document in the system 	Development of the module in the system	Week 20 (27 August 2021)
<p>Testing</p> <ul style="list-style-type: none"> ▪ Successfulness to generate the QR code ▪ Successfulness to generate the steganography image ▪ Successfulness to embed the QR code image, steganography image and watermark image into the document • Accuracy of the information in the QR code and steganography image 	QR code image, steganography image and watermark image embedded into document, readable QR code, successfully extract steganography image	Week 21 (3 September 2021)

3.4 Summary

This chapter explains in detail the methodology used to implement the project. The activity by activity method is used as this project's methodology because it is the simplest and easiest to understand. The methodology includes six processes which are literature review, analysis, formulate hybrid protection techniques, embedded protection techniques into document, implementation, and testing. Lastly, the project milestones and Gantt chart are also included which show the timeline of this project. In the next chapter, the design for this project will be explained.

CHAPTER 4: ANALYSIS AND DESIGN

4.1 Introduction

Chapter 3 describes the processes that are carried out in this project. Among the processes are literature review, analysis, formulate hybrid protection technique, embedded protection technique into document, implementation, and testing. Milestone and Gantt Chart are also included as it help in organizing the project. This chapter describes the plan and acts as a guideline before starting the project. It shows the design and requirements needed to develop the system module. This chapter contains software requirements, hardware requirements, system architecture, some of user interfaces design in the system, entity relationship diagram (ERD) design, QR code generator process design with details, steganography image generator process design with details and embedded images into document process design with details.

4.2 Requirement

The requirement is vital to be gathered before starting the system development as it is the process of determining what the system needs in order to successfully develop it. There are two requirements part, which are software requirement and hardware requirement. The next subsection will describe these two requirements.

4.2.1 Software Requirement

There is some software needed to be used in the project development. The main software used are VScode for code writing, MySQL as a database of the system and Google Chrome as a place to view the output or find errors during the code writing. Table 4.1 shows the list of software used and its description.

Table 4. 1: The list of software used with the description

Software	Description
Microsoft Windows 7 Operating System	Windows is one of the operating system that are commonly used by computer users as the user interface makes it easy to use no matter what type of computer the user are using.
Microsoft Word 2016	Microsoft word is a software used for writing a report for this project.
Draw.io	Draw.io is a software used to design a module of the system, entity relationship diagram, system architecture, flowchart and other several diagram.
Pencil	Pencil is an open source software that is used to design logical user interface for the system.
Laragon	Laragon is an open source software consisting Apache web server, MySQL database and PHP interpreters.
Visual Studio Code	VScode is a free source code editor software that is use to write the code for the system.
Google Chrome	Google Chrome is a browser to view the output or user interface of the code from VScode.
QR Reader	QR Reader is a software downloaded in mobile phone to read the QR image embedded in the document.

4.2.2 Hardware Requirement

Apart from that, hardware requirement is also needed to be gathered to ensure that it is compatible with the software listed in Table 4.1. Table 4.2 shows the hardware used for this project and its description.

Table 4. 2: The list of hardware used with the description

Hardware	Specs
Laptop	<ul style="list-style-type: none"> • Intel Core i3 2.13GHz • 4GB RAM • Windows 7 x64-bit Operating System

Hardware	Specs
Mobile Phone	<ul style="list-style-type: none"> • Octa-core (4×2.3 GHz Cortex-A53 & 4×1.8 GHz Cortex-A53) • 3GB RAM • Android 8.1.0

4.3 System Architecture Design

System architecture design provides a view on where the modules are located. There are three main modules in this project which are a QR code generator, steganography image generator and embedded images into a document. The architecture is illustrated in Figure 4.1.

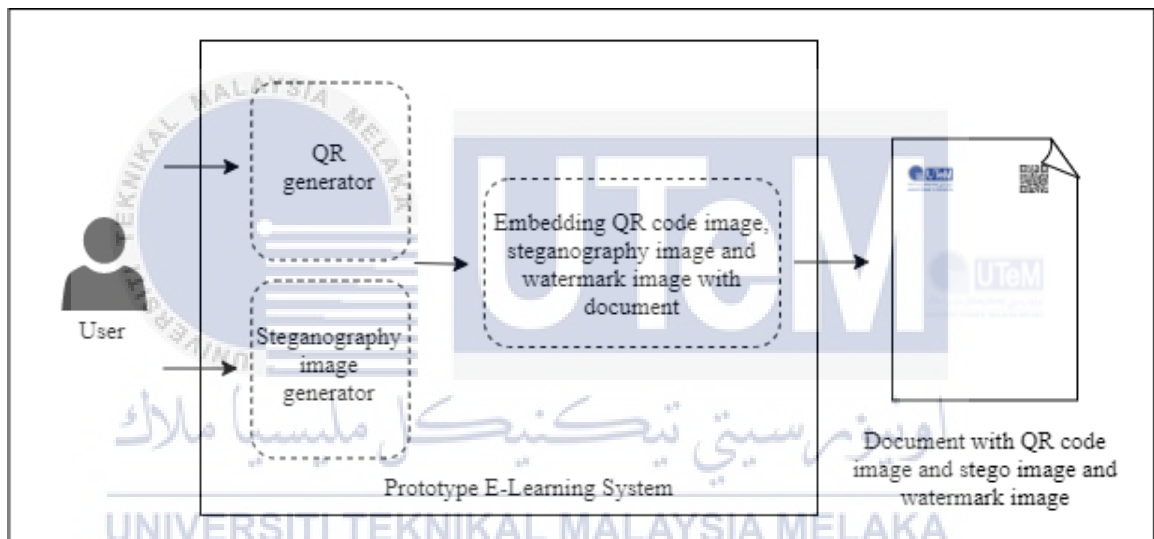


Figure 4. 1: System Architecture

Figure 4.1 shows the overview of the document with QR code image, steganography image and watermark image embedded within it is produced. Before the system proceeds to the two image generator module and the embedded module, the user first needs to login into the system. After the user has the right to access the system, the user can do anything there such as download lecture notes, lab sheets and submit an assignment. The flowchart is shown in Figure 4.2.

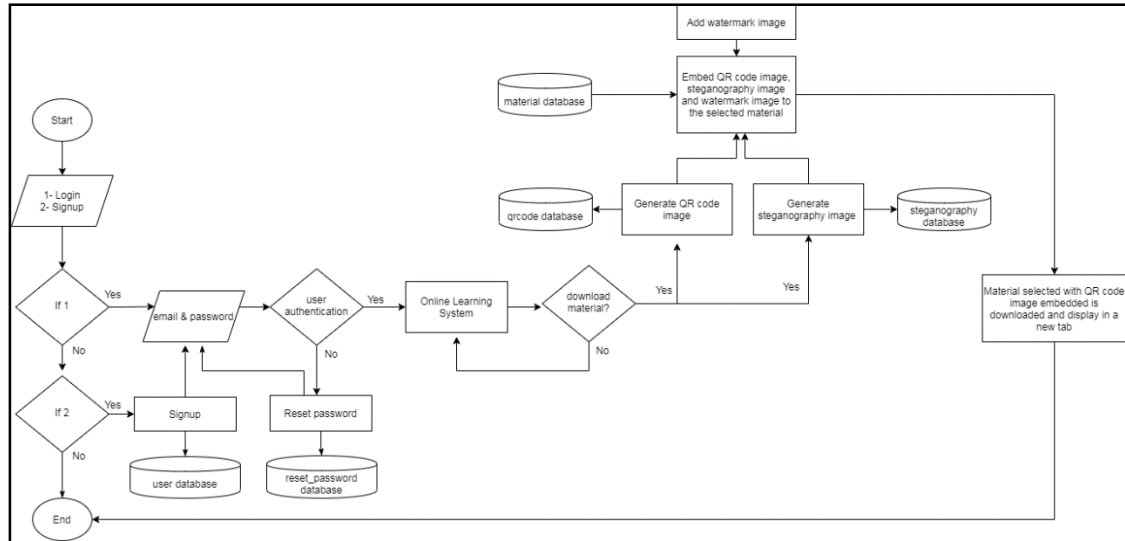


Figure 4. 2: Flowchart of the main module

Figure 4.2 shows the flowchart of the main module. The QR code generator and steganography image generator will be triggered once the user clicks any material that can be downloaded from the system. The QR code generator will generate the QR code image, the steganography image generator will generate the steganography image and then the program will compress the generated images with one watermark image into the selected document. As a result, the user will download the material that contains the three images stated. The “System” in Figure 4.1 is referred to the interface as in Figure 4.3, Figure 4.4, and Figure 4.5. It displays some of the user interfaces where the user can download the material.

Student submission status				
Dashboard				
STUDENT SUBMISSION				
Navigation bar	<input type="text" value="Search"/> <input type="button" value="Search"/>			
	Name	Email	Submission status	
	Muhammad Abu Bakar	abu@student.com	submitted	lab1_abu.pdf
	Student 2	matric2	None	
	Student 3	matric3	None	
	Student 4	matric4	None	

Figure 4. 3: User interface of view student assignment submission

Figure 4.3 shows the user interface of student assignment submission on the lecturer side. Every student assignment file that is uploaded to the system can be seen by their lecturer.

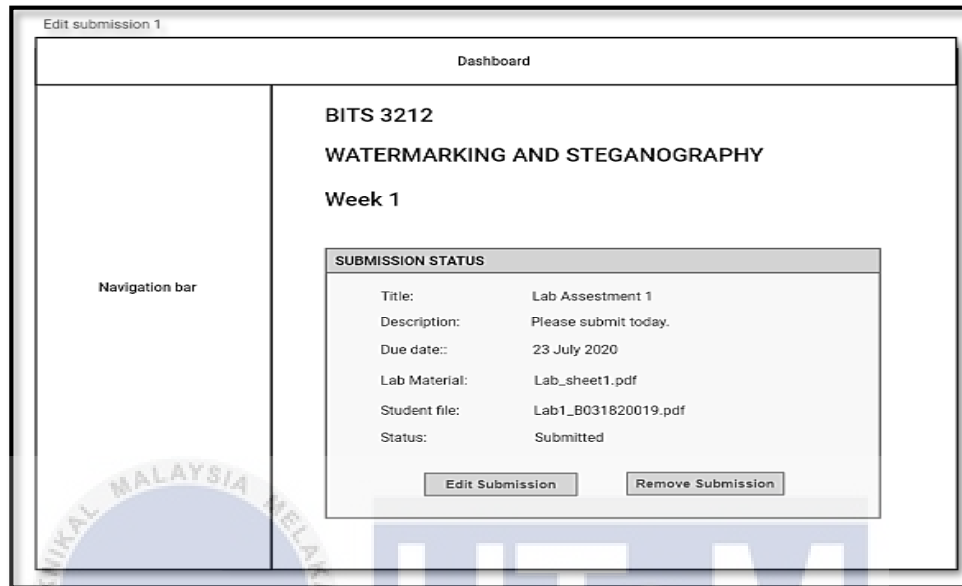


Figure 4. 4: User interface of student assignment submission

Figure 4.4 shows the user interface where student can submit their assignment. In this user interface, the student can download the lab sheet file and after students submit their assignment file, their file also can be downloaded by themselves.

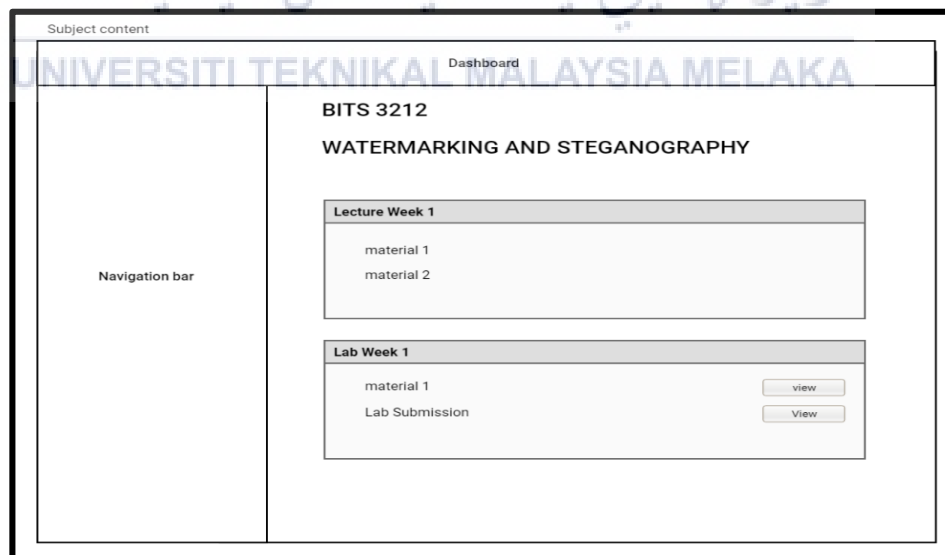


Figure 4. 5: User interface that contains lecture note and lab sheet

Figure 4.5 shows the user interface of lecture notes and lab sheets where the user can download. Figure 4.3, Figure 4.4, and Figure 4.5 indicate where the three modules as illustrated in Figure 4.1 are located. In addition, the entity-relationship diagram (ERD) of the system is designed and shown in Figure 4.6.

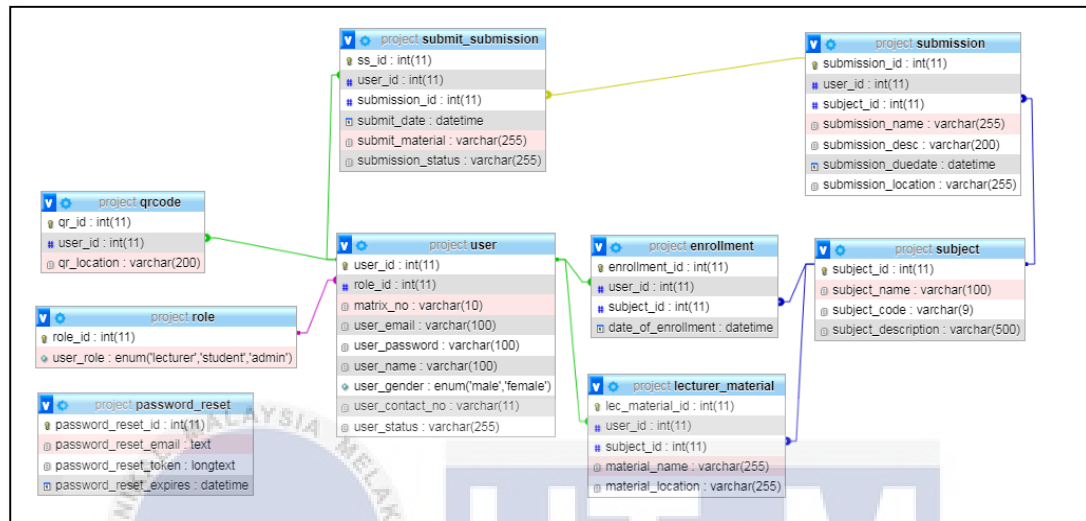


Figure 4. 6: ERD of the system

Figure 4.6 shows the database to store the data used for this project and the relationship between each entity. In the next section, the details of the main module will be explained.

4.4 QR Code Generator Design

In this section, the design of the QR code is discussed. There are eight processes in this design which are data input, analyzing input data, data encoding, creating error correction codewords, structuring final data, converting block into QR matrix, apply mask pattern, apply version and format information as shown in Figure 4.7.

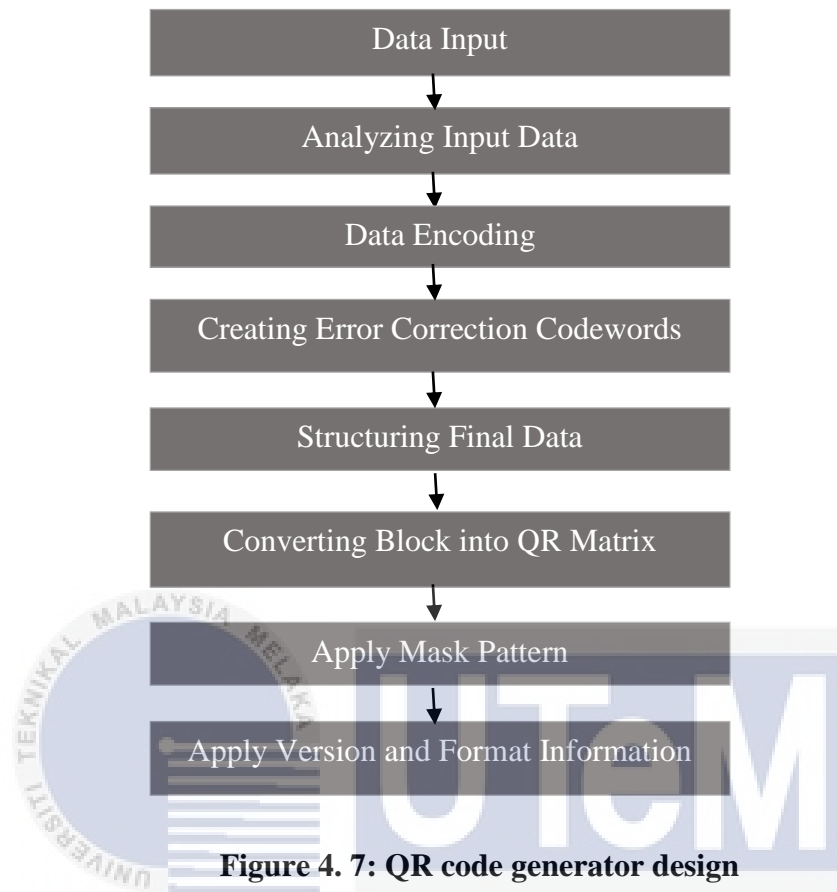


Figure 4. 7: QR code generator design

Figure 4.7 shows the QR code generator design. These eight processes are the processes in generating the QR code image. In the next subsection, the details of each of the processes will be discussed.

4.4.1 Data Input

Data input here means the information that has been gathered either from the system or from the program. The information gathered from the system is the material choose by the user, for example, lecture note, lab sheet, or the user's assignment document. Other informations are the user name, user matrix number, user email, owner of the downloaded document, and source of the downloaded document. While the information gathered from the program is error correction capability (ECC) level which are either level low, medium, quartile, or high, then, pixel size and frame size. The pseudocode for inserting required input data is shown in Table 4.3:

Table 4. 3 : Pseudocode for inserting required information into the QR code

Begin
Pass In: user_id, material_location
Get information from database
Found:= matrix_no, user_name, user_email, material_location
Initialize matrix = matrix_no
Initialize name = user_name
Initialize email = user_email
Initialize owner = user_name
Initialize material = material_location
Intialize ecc = H
Initalize pixel_size = 10
Initialize frame_size = 10
Generate QR code
Output: QR code image
Save QR code image to local server
Call: combine_qr_pdf (user_id, material_location)
Return call

4.4.2 Analyzing Input Data

During this process, the input data type from the previous process is being analyzed. There are several data types available which are numeric, alphanumeric, byte, and kanji. If the input data consist of digit numbers, for example (0-9) only, the data type is considered numeric. If the input data consist of space, number, symbol, and characters it is considered alphanumeric. For alphanumeric data type, each character has its value, for example, for character A, the value is 10. The alphanumeric value is shown in Table 4.4.

Table 4. 4: Alphanumeric table

Char	Code	Char	Code
0	0	N	23
1	1	O	24

Char	Code	Char	Code
2	2	P	25
3	3	Q	26
4	4	R	27
5	5	S	28
6	6	T	29
7	7	U	30
8	8	V	31
9	9	W	32
A	10	X	33
B	11	Y	34
C	12	Z	35
D	13	space	36
E	14	\$	37
F	15	%	38
G	16	*	39
H	17	+	40
I	18	-	41
J	19	.	42
K	20	/	43
L	21	:	44
M	22		

If there is no character found in Table 4.4, for example, lowercase () it is considered used the byte data type. If the input data contains Japanese or Kanji characters, the data type will be Kanji. After analyzing the data, it then will go through the next process.

4.4.3 Data Encoding

In this process, the input data that has been analyzed will be encoded. The input data will be transformed into a string of bits according to the respective data type. Before encoding the input data, the program first will identify the ECC level. The ECC level is important as it is the capability for the QR code to be restored if the code is damaged. There are four levels which are high (H), quartile (Q), medium (M), and low (L). According to Lotlikar et. al. (2013) the higher the level of the ECC, the lesser the storage capacity to store the data. Table 4.5 shows the percentage of the damaged QR code that can be restored for each level of the ECC.

Table 4. 5: ECC level (Denso Wave Incorporated, 2021)

ECC Level	Percentage can Restored Data
Level L	About 7%
Level M	About 15%
Level Q	About 25%
Level H	About 30%

The ECC level has been determined in the input data process. Based on this project, the information that is going to be stored in the QR code is only consists of user id, user name, user matrix number, user email, name of the document owner, download date, downloaded by and document source. In this project, level H is the best ECC level to use as it can result faster scanning.

After that, the smallest version of the QR code also has been determined in this process. It is determined by calculate the encoded input data and combine with the ECC level. The result of the calculation will show the smallest version that the encoded data can be inserted in the QR code. For example, the input data is 'NAME: FATIN', this phrase has 11 characters and it is encoded with **level L** of ECC, so the capacity and the version in Table 4.6 will show **Version 1** as its smallest version which the capacity for the QR code can store data is **maximum 25 characters**. There are 40 versions of the QR code. Each version has its own capacity and modules. Table 4.6 shows some of the QR code versions with their maximum allowable capacity.

Table 4. 6: QR code version with maximum allowable capacity (Denso wave Incorporated, 2021)

Version	Modules	ECC level	Numeric	Alphanumeric	Byte	Kanji
1	21x21	L	41	25	17	10
		M	34	20	14	8
		Q	27	16	11	7
		H	17	10	7	4

Version	Modules	ECC level	Numeric	Alphanumeric	Byte	Kanji
2	25x25	L	77	47	32	20
		M	63	38	26	16
		Q	48	29	20	12
		H	34	20	14	8
3	29x29	L	127	77	53	32
		M	101	61	42	26
		Q	77	47	32	20
		H	58	35	24	15

The next process is determining the mode indicator. Each data type has its mode indicator. For instance, Numeric is indicated as 0001, Alphanumeric is 0010, Byte is 0100 and Kanji is 1000. From the example, the phrase ‘NAME: FATIN’, has the **indicator mode** of **0010** because the data type is alphanumeric. This indicator mode will be added to the start of the encoded data. Table 4.7 shows the indicator mode for the respective data type.

Table 4. 7: Indicator mode for respective data type (Denso wave Incorporated, 2021)

Data type	Indicator Mode
Numeric	0001
Alphanumeric	0010
Byte	0100
Kanji	1000

After the indicator mode is determined, the next item that needs to be determined is the character count indicator. The character count indicator is added after the mode indicator is being determined. In this process, the characters counting will start from the original input data and then convert it into binary. The character count indicator’s length is based on the encoding type and the QR code version used. For example, from the same phrase, ‘NAME: FATIN’ which consists of 11 characters with QR code version 1 and encoding data type alphanumeric, then the character count indicator must have a length of 9 bits. The 11 characters need to be converted into binary which becomes 1011. To make

it fits in the length of 9 bits, it will become 000001011 which the 0s will be filled on the left side if the binary is not enough 9 bits. This character count indicator result will be added after the mode indicator which will be 0010 000001011. Table 4.8 shows the character count indicator according to the version and data type.

Table 4. 8: Character count indicator according to the version and data type
(Thonky, 2021)

Version	Data type	Character count indicator
1 - 9	Numeric	10 bits
	Alphanumeric	9 bits
	Byte	8 bits
	Kanji	8 bits
10 - 26	Numeric	12 bits
	Alphanumeric	11 bits
	Byte	16 bits
	Kanji	10 bits
27 - 40	Numeric	14 bits
	Alphanumeric	13 bits
	Byte	16 bits
	Kanji	12 bits

After the character count indicator has been determined, the input data then will be encoded according to the selected data type encoding. In the encoding process, the input data will be breaks up into pairs. The example is shown in Table 4.9.

Table 4. 9: Break phrase into pairs

Phrase	Phrase in pairs					
NAME: FATIN	NA	ME	: SPACE	FA	TI	N

To begin the encoding calculation, each of the characters first is converted into code according to the alphanumeric table (because the data type is alphanumeric). Then, for the calculation, the first character will be multiplying with 45 and add with the second code. The result of the calculation is in decimal then convert it into binary. If the phrase or the input data is odd, the character will be converted into a 6-bit binary (in the example the

odd character is the last N). The calculation on how the data is being encoded is shown in Table 4.10.

Table 4. 10: Data encoding based on respective data type

	Phrase in pairs					
Character	N A	M E	: SPACE	F A	T I	N
Code	23 10	22 14	44 36	15 10	29 18	23
Calculation	$(23*45) + 10$	$(22*45)+14$	$(44*45)+36$	$(15*45)+10$	$(29*45)+10$	23
Decimal	1045	1004	2016	685	1315	23
Binary	10000010101	1111101100	11111100000	1010101101	10100100011	010111

The example is in the alphanumeric data type so the data is encoded with the alphanumeric encoding and the current bit of string is:

Table 4. 11: Current bit string of the example

Mode Indicator	Character count indicator	Encoded data		Total bit string
0010	000001011	10000010101	1111101100	72 bits
		11111100000	1010101101	
		10100100011	010111	

Then, the next process is the current bit string will be broken up into 8-bit codewords and bytes will be added if necessary. In this process, the required number of bits for the QR code is determined. Table 4.12 show some of the error correction codewords for QR code version 1.

Table 4. 12: Error correction codewords (Thonky, 2021)

Version - EC Level	Total Number of Data Codewords for this Version and EC Level
1-Low	19 bits
1-Medium	16 bits
1-Quartile	13 bits
1-High	9 bits

To determine the bits required for a QR code, first need to refer on the QR code version that has been determined before and EC level. According to the example, phrase ‘NAME: FATIN’ with the alphanumeric data type, QR code version 1, and ECC level L, the total number of data codewords is 19 bits. Therefore, the total bits required for the QR code is $19 * 8$ bits which equal to 152 bits.

Next, the indicator of 0s is needed and is added to the right side of the encoded data if the total bit string is shorter than the total of bits required in QR code. For instance, the current example bits string is 72 bits, but the required bits for the QR code is 152 bits long. Because of that, the terminator is needed but the terminator can only be at most 4 bits long, so four 0s is added to the right side of the encoded data and will be, 0010 000001011 10000010101 1111101100 1111100000 1010101101 10100100011 010111 0000. Table 4.13 shows the terminator added to the right of the encoded data.

Table 4. 13: Terminator is added

Mode Indicator	Character count indicator	Encoded data	Terminator	Total bits string
0010	000001011	10000010101 1111101100 11111100000 1010101101 10100100011 010111	0000	76 bits

Next, the string of bits is arranged into 8-bit. If the string bits are not enough of 8-bits, then add more 0s on the right side of the data bits. Table 4.14 shows the bits string of the encoded data that is arranged in 8-bit by 8-bit. The zero value is added on the right side to completed the 8-bit.

Table 4. 14: Arranged encoded data

Bit string in 8-bit	Total bits string
00100000 01011100 00010101 11111011 00111111 00000101 01011011 01001000 11010111 00000000	80 bits

According to the current bit string (80 bits), it does still not reach the total required bits (152 bits). To achieve the 152 required bits for the QR code, the following bytes can be added repetitively until the total bits string achieve the required bits:

- 11101100
- 00010001

236 and 17 are the equivalents of these bytes. These bytes are needed by the QR code if the total bits string after encoding is not enough. To know how many bytes to add to complete the required string bits, we can calculate it by subtracts the total required string bits with the total current bits string. In this example, the required bits string for QR code is 152 bits and the current string bits after terminator has been added is 80 bits, so $152 \text{ bits} - 80 \text{ bits} = 72 \text{ bits}$ left. Then divided by 8 to get how many bytes are left, $72/8 = 9$ bytes. Therefore, 9 bytes are required and must be added to the end of the data string. Table 4.15 shows the required bytes for the example.

Table 4. 15: Required bytes for the example

Required bytes	Total bits string
00100000 01011100 00010101 11111011 00111111 00000101	152 bits
01011011 01001000 11010111 00000000 11101100 00010001	
11101100 00010001 11101100 00010001 11101100 00010001	
11101100	

After the data codewords have been obtained and meet the required bits for the QR code, the next process is generating an error correction codewords for the data.

4.4.4 Creating Error Correction Codewords

In this process, the encoded data will be converted back into decimal and will be transformed into the polynomial form. Table 4.16 shows the polynomial for the encoded data.

Table 4. 16: Encoded data in decimal and polynomial

Item	Data
Encoded Data	00100000 01011100 00010101 11111011 00111111 00000101 01011011 01001000 11010111 00000000 11101100 00010001 11101100 00010001 11101100 00010001 11101100 00010001 11101100
Decimal	32 92 21 251 63 5 91 72 215 0 236 17 236 17 236 17 236 17 236
Polynomial	$32x^{18} + 92x^{17} + 21x^{16} + 251x^{15} + 63x^{14} + 5x^{13} + 91x^{12} + 72x^{11} +$ $215x^{10} + 0x^9 + 236x^8 + 17x^7 + 236x^6 + 17x^5 + 236x^4 + 17x^3 + 236x^2$ $+ 17x^1 + 236$

After the encoded data has been converted into the polynomial form there are other several processes such as multiplying and XORing and the final result is will be converted back into decimal and the error correction codewords are generated. After the data codewords and error correction codewords have been generated, the next process is to structure the final data.

4.4.5 Structuring Final Data

In this process, the final data is being structured. How many the encoded data will be in a block and how many error correction codewords are required are determined here. After the final data has been structured in a block, it will go through the next process.

4.4.6 Converting Block into QR Matrix

During this process, the structured data in the previous process will be place in the QR code matrix along with several patterns such as finder pattern, timing pattern, alignment pattern, and separators. The top left, top right and bottom left of the QR code are the locations for the finder pattern. No matter what version the QR code is, the position is always the same. The separators are white modules and will be located around the finder pattern and then the alignment pattern and timing pattern are also located in the matrix. Lastly, the remaining space will be filled up with the encoded input data.

4.4.7 Apply Mask Pattern

In this process, the data masking is applied to the data that have been placed in the matrix. This is to ensure that the QR code is readable by the QR scanner.

4.4.8 Apply Version and Format Information

This is the last process. In this process, the version and the format information are applied to the QR matrix. Lastly, after the required information has been placed in the QR matrix, a quiet zone is applied around the QR matrix which will become a complete QR code.

4.5 Steganography Image Generator Design

The next module will be about injecting the ownership information into an image where that image will look like a watermark image and will be embedded into the document. There are five processes involved which are determine data and image, encode data, determine image properties, extract RGB color of the image, and inject data into image.

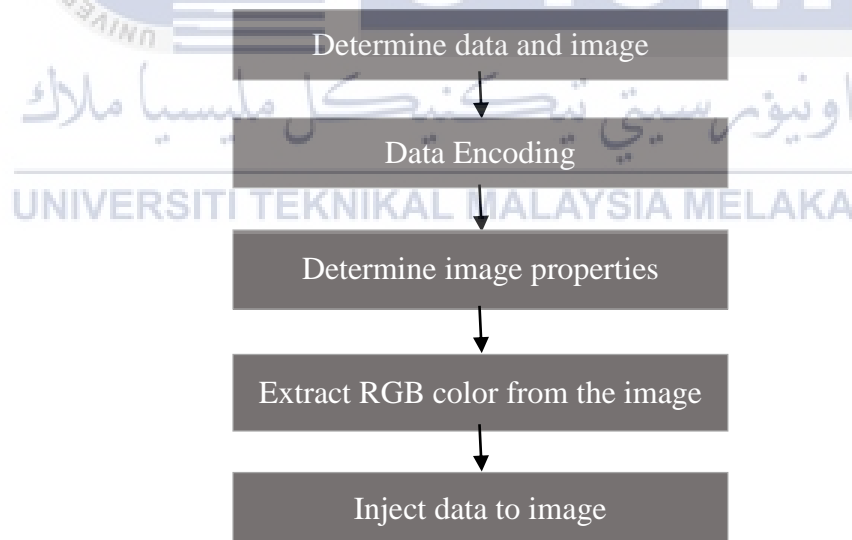


Figure 4. 8: Inject data into image design

Figure 4.8 shows the design for injecting ownership information into the image. In the next subsection, the details of each of the five processes will be discussed.

4.5.1 Determine data and image

In this process, the data is being determined the same as the data to be encoded in the QR code image. The data to be included are user id, date of the download file, name of the person who downloaded the file, matrix number, email, source file, and file owner name. This data has been store in the display variable and will be called in this process. Next is determining which image will be a container to store the data. The image is determined by giving the path of the image itself. The image chosen by the developer is the UTeM's logo.

4.5.2 Data Encoding

After the data and the image as a container have been determined, the data needs to be encoded before inject into the image. Firstly, each of the data's character will be converted to ASCII in binary form. For example, a letter U will have an ASCII value of 85 and 1010101 in binary.

Table 4. 17: ASCII Table

Dec	Binary	Char	Dec	Binary	Char	Dec	Bin	Char
3	00000011	[end of text]	78	01001110	N	103	01100111	g
32	00100000	[Space]	79	01001111	O	104	01101000	h
45	00101101	-	80	01010000	P	105	01101001	i
46	00101110	.	81	01010001	Q	106	01101010	j
47	00101111	/	82	01010010	R	107	01101011	k
58	00111010	:	83	01010011	S	108	01101100	l
64	01000000	@	84	01010100	T	109	01101101	m
65	01000001	A	85	01010101	U	110	01101110	n
66	01000010	B	86	01010110	V	111	01101111	o
67	01000011	C	87	01010111	W	112	01110000	p
68	01000100	D	88	01011000	X	113	01110001	q
69	01000101	E	89	01011001	Y	114	01110010	r
70	01000110	F	90	01011010	Z	115	01110011	s
71	01000111	G	95	01011111	_	116	01110100	t
72	01001000	H	97	01100001	a	117	01110101	u
73	01001001	I	98	01100010	b	118	01110110	v
74	01001010	J	99	01100011	c	119	01110111	w
75	01001011	K	100	01100100	d	120	01111000	x
76	01001100	L	101	01100101	e	121	01111001	y
77	01001101	M	102	01100110	f	122	01111010	z

Figure 4.17 shows some of the ASCII table attribute. To convert each letter of the data into binary from the ASCII table, ord() function can be used for a php language. This function will give an integer value for each of the character and then the integer value will be converted to binary using decbin() function where it return a string of the binary number. The decbin() function will return its value to 7 bit number, so in order to force it value to return to 8 bit number, the str_pad() function is used. Table 4.18 shows the example on how the data is converted based on ASCII table into 8 bit by 8 bit.

Table 4. 18: Example to encode data

Msg	U	s	e	r	[space]	I	D	:	[end of text]
Int	85	115	101	114	32	73	68	58	3
Bin	01010101	01110011	01100101	01110010	00100000	01001001	01000100	00111010	00000011
Output	01010101011100110110010101110010001000000010000001001001010001000011101000000011								

Apart from that, the 'end of text' also will inserted into the string so that during the decode process (extract message) from the image, the end of the message can be detected.

4.5.3 Determine Image Properties

The next process is determining some of the image properties. For example set the image opacity and transparent of the image before the encoded message is injected into the image. In this process, the image opacity and the transparent is set to the lowest value to ensure that image will not disturb the content of the image.

4.5.4 Extract RGB color from the image

After the image properties has been setting as supposedly, the Red, Blue, Green (RGB) channel of the image will be extracted. The reason to extract this RGB value is because the encoded data before will be injected into the least significant bit (LSB) of the blue channel. The blue channel is used because human eyes are less sensitive towards that color. According to Vaishnavi and Subashini (2014), there are three channel where the eyes can sense more which are red (R), green (G), and blue (B) and they stated that 65% of the eyes are sensitive to red color, 33% are sensitive to green color and 2% are sensitive to blue color. So, blue channel is the lesser can be detected by eyes if there is information injected in that channel.

4.5.5 Inject data to image

After the RGB color from the image has been extracted, the value of the blue channel will be converted into binary form. Then, each binary number of the data, for example, character 'U', in binary is 01010101, this binary number will be injected to the LSB of the image's blue channel starting from the image's upper left corner and, process each image's pixel row until the entire message has been injected. The stego image is then will be saved in the local storage and will be used in the next module.

4.6 Design for Embedding Images into Document

Next, the system will go through this process once the QR code image and the steganography image in the previous section is successfully generated. In this section, there are four processes which are import document, determine page in the document to put the images, import images, and embed images into the document.

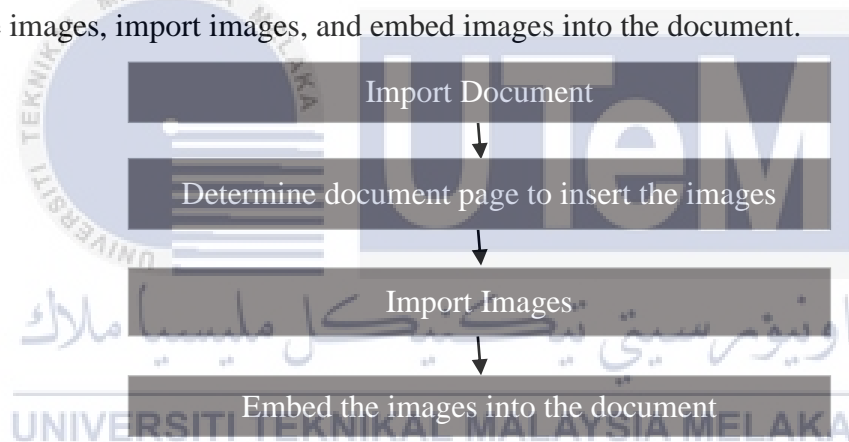


Figure 4.9: Embed images with document design

Figure 4.9 shows the process design for embedding the QR code image, steganography image, and UTeM's logo as a watermark image into the document. There are four processes and the details of each of the processes will be discussed in the subsection.

4.6.1 Import Document

In this process, the host of the QR code image, the steganography image, and the watermark image are being imported. The document imported in this process is got from the system. The user interface for this process to be triggered is shown in Figure 4.3, Figure 4.4, and Figure 4.5. From these user interfaces, once the user clicks the material (eg: lecture note, lab sheet, student's assignment document) in the system, the program

will get the material location and pass the material location to this process. The imported document is determined by knowing the material location. In this process, the program will get information on the imported pdf document such as the pdf version, pdf pages, pdf reader id. The document will be imported using the function of `setSourceFile()`.

4.6.2 Determine Document Page to Insert The Images

In this process, the page to insert the three images is determined. In this project, the page to insert the images is set to all pages. The `importPage()` function is used to determine which pages in the document the QR code image, the steganography image, and the watermark image will be inserted.

4.6.3 Import Images

The next process is importing the three images stated. In this process, which QR code image and which steganography image to be imported is based on the current user id. In the QR Code generator module and steganography image generator module, the QR code image, and the steganography image are automatically generated once the user clicks the material from the system as in Figure 4.3, Figure 4.4, and Figure 4.5. The generated images are saved in the local server and the location of the images will be called during this process. In this process, the function used to get the imported QR code image, steganography image and watermark image is `Image()`. The size and position to be inserted into the document also have been defined in that function. During this process, the program will check for file extension, for example, `.png` or `.jpg`. Then, the program will check for file signature to confirm the file type. The program then checks for other information such as header chunk, palette, transparency, and image data block to determine whether the images imported are a grayscale image or RGB image. After determined, the images is then will be compressed. Next, the program will determine the position and the size of the image to be stored in the document. The value of these two properties will be defined in the `Image()` function.

4.6.4 Embed The Images Into Document

After the images have been compressed, the program will embed the three images into the imported document (in the import document process). As a result, the program produces a pdf document with a QR code image, a steganography image, and a watermark image embedded within it and saves it on the local server. Lastly, this document is the material that is going to be downloaded by the user in the system.

4.7 Summary

This chapter has discussed the design and analysis for the main module in the system. The requirement for software and hardware are gathered to ensure this project can be developed smoothly, the system architecture is drawn to show where is the main module are located in the system, the user interfaces is design before implement the real interface, ERD is designed to visualize how the data in the system is connected, the module of QR generator, steganography image generator and images embedded into the document are described including the module process design, explanation on the module design, and the module process. The implementation of the module will be discussed in the next chapter.

CHAPTER 5 : IMPLEMENTATION

5.1 Introduction

Chapter 4 explained the design and requirements needed to develop a prototype system for testing and proposed QR code along with steganography image and watermark image for protecting document ownership. The chapter also elaborated on the software and hardware requirements, system architecture, and the design of QR code generators, design of steganography image generator and design of image embedded into documents. Chapter 5 explain the software development setup and the implementation of the proposed copyright protection. This chapter also presents and explains the pseudocode for each process involved in protecting document ownership.

5.2 Software Development Environment Setup

In this project, web application and database managers are installed and configured. Hence, this project selected Laragon as the web application manager and Apache web server as the database manager.

5.2.1 Web Application Manager

Laragon is software used for managing web application. After installing Laragon software, several changes need to be done and are shown in Figure 5.1.

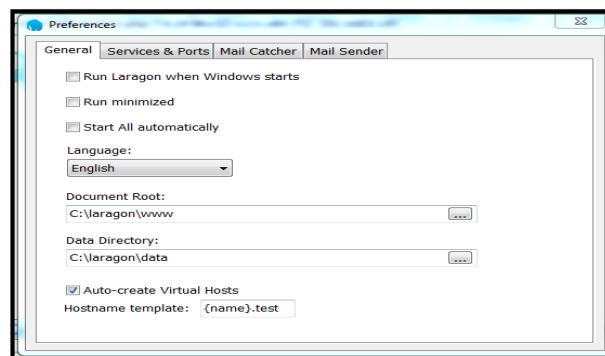


Figure 5. 1: Laragon setting

Figure 5.1 shows the general information that setting according to the project, such as document root, data directory, and virtual hostname. The path for the document root and the data directory must be correct for the software to function correctly. Document root contains all files related to this project, and the data directory is the path of the database information of this project located. The virtual hostname is automatically created based on the project's name in the root document. Therefore, the name display on the web browser consists of the project name and hostname template, which will be e-learning.test.

5.2.2 Database Manager

There are several web servers can be added to the Laragon and the one used for this project is the Apache web server. The phpMyAdmin is used to manage the database and can be downloaded from the internet. To integrate the web server with the Laragon software, the PhpMyAdmin folder need to be added to the Laragon folder which located in the 'laragon\etc\apps' as depicted in Figure 5.2.

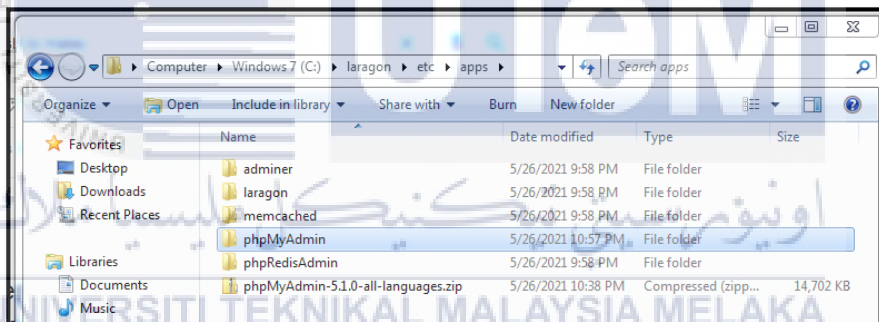


Figure 5. 2: Add phpMyAdmin folder to Laragon File

Figure 5.2 depicts the phpMyAdmin folder that contains many files that are related to the database application. It has been unzipped, rename, and added to the Laragon folder. To avoid any error related to permission, the name of the folder must be *phpMyAdmin*.

5.2.3 Visual Studio Code (VS Code) Setup

Visual Studio Code is a software used for scripting. There are several extensions that can be added to the VS code. The important extension in this project is the 'SQLTools' where it is used to set up the database, and 'SQLTools MySQL/MariaDB' for the database driver. Other extension is just an option. 'Open PHP/HTML/JS In Browser' is used for opening the code in the browser, 'Php cs fixer', 'PHP Intelephense',

and 'PHP IntelliSense' are extensions that help in giving several words or ideas to complete the code. Figure 5.3 shows the extensions that are used in the VS code for this project.

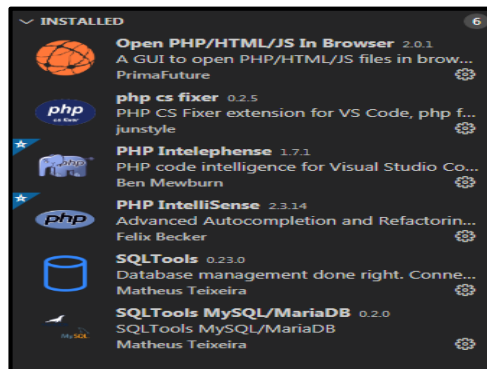


Figure 5. 3: VS Code extensions

After extensions has been added, database connection must be setup. In this setup, the connection name, the medium to connect to the database known as connect using, server address, port number, database name, and username is set as shown in Figure 5.4. This setting aims to ensure the VS code is connected to the database (phpMyAdmin).

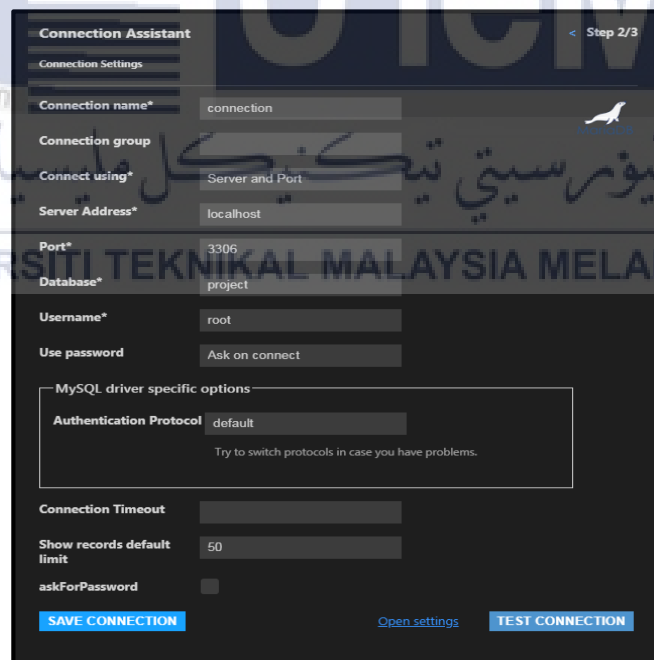


Figure 5. 4: Setup Database Connection in VS Code

Figure 5.4 shows how to set up the database connection between the database and Vs code. The next thing to do is to ensure that the connection between the database and the VS code

is established by testing the connection in a new file and run on an active connection. If the result is shown, then the connection is working.

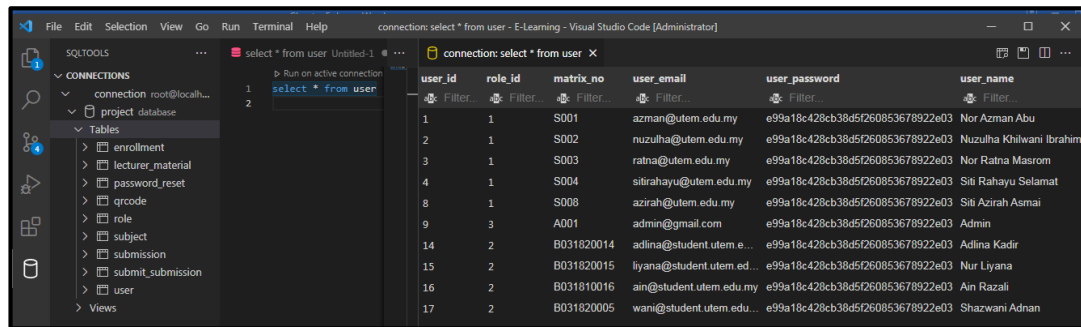


Figure 5. 5: Established Database Connection in VS Code

Figure 5.5 shows the connection that has been established between the VS code and the database. After all the required installation, setup, and configuration has been made, the implementation of the project can be started and will be explained in the next section.

5.3 Implementation

This section elaborates on how the QR code is generated through the QR code generator, how the steganography image is generated through the steganography image generator, and how the images are embedded into the document. Figure 5.6 shows the diagram for the whole process and how it was connected each other.

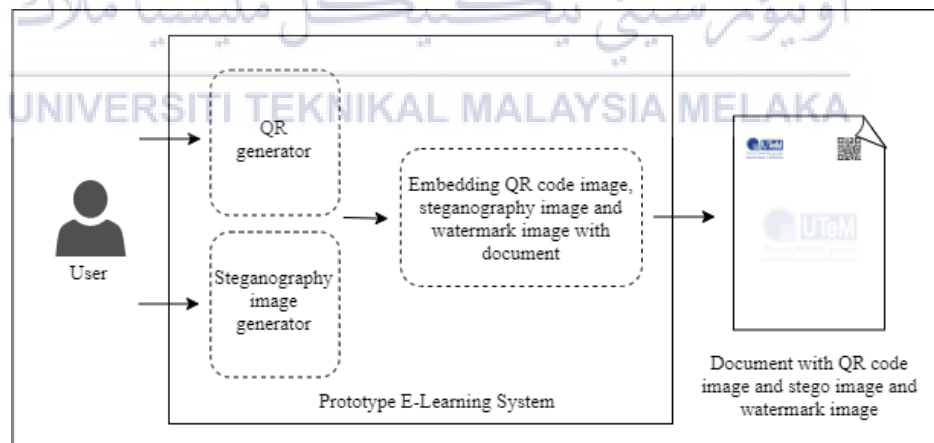


Figure 5. 6: System architecture

Based on Figure 5.6, there are three main modules in this project which are the QR code generator, steganography image generator, and embedding the images into the document. Before the system proceeds to the three modules, the user must first log in to the system. Once the user has been granted access to the system, the user can download and upload a file into the system.

The first module that will trigger once the user clicks any file in the system is the QR code generator and the steganography image generator. After these two images have been generated, the system will proceed with the embedding process which another one image is added called a watermark image that will be embedded into the document along with the two images. The three images are positioning on the top left, on the top right, and center of the file. The file containing the three images will be the one that the user downloads.

5.3.1 QR code Generator

QR code generator module consists of eight main processes namely 1) Data Input, 2) Analyzing Input Data, 3) Encoding data, 4) Creating Error Correction Codewords, 5) Structuring Final Data, 6) Converting Block into QR Matrix, 7) Applying Mask Pattern and, 8) Applying Information of Version and Format. The flow of the QR code generation process is shown in Figure 5.7.

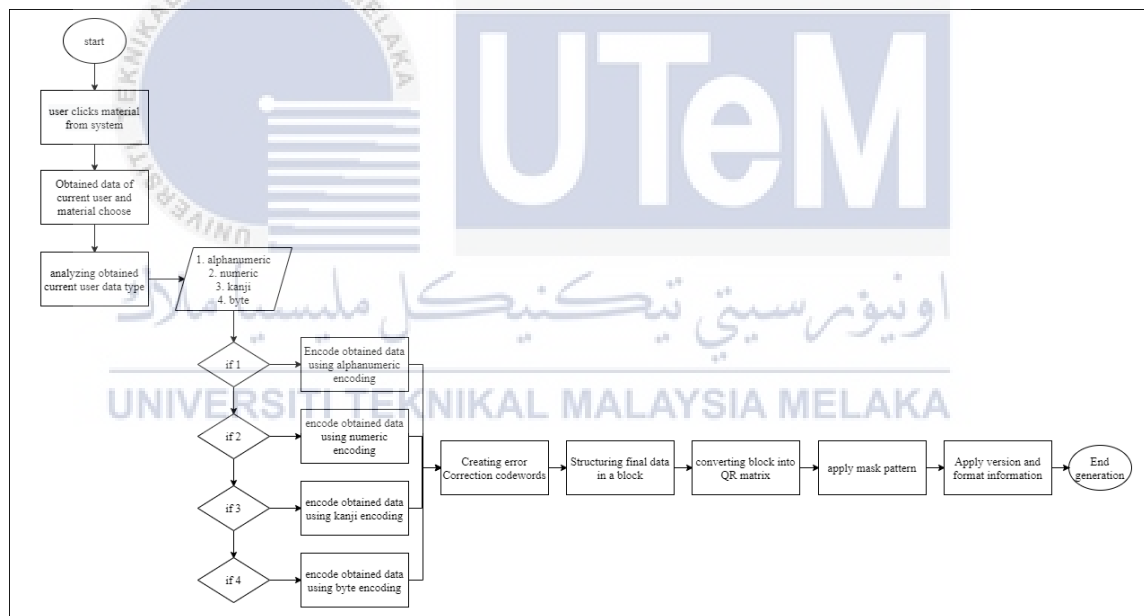


Figure 5. 7: Flowchart main process for the QR code generation

Figure 5.7 illustrated the main process for the generation of the QR code. The implementation of the QR code generation will be described in the subsection.

5.3.1.1 Data Input

The data input is the information that is going to be placed inside the QR code. It includes the current user id, the date of the file downloaded, the name of the user who

downloads the file, matrix number, email, owner name, and source of the file. Table 5.1 shows the pseudocode of the data input process.

Table 5. 1: Pseudocode for Data Input Process

<pre> Begin Pass In: user_id, material_location Get information of user and material from database Found:= user_id, matrix_no, user_name, user_email, file_owner, material_name id = get user_id from database matrix = get matrix_no from database name = get user_name from database email = get user_email from database owner = get user_name from database url = get material_name from database download_date = date('d-m-Y H:I A') display = "User ID: " + id + "Download date:" + download_date + "Downloaded by:" + name + "Matrix no:" + matrix + "Email:" + email + "File owner:" + owner + "Source File:" + url ecc = H pixel_size = 10 frame_size = 10 access class method QRCodepng(display, ecc, pixel_size, frame_size) </pre>
--

Table 5.1 shows the process of getting the information to be input into the QR code.

5.3.1.2 Analyzing Input Data

In this process, the mode of the data obtained from the previous process is identified as either numeric, alphanumeric, kanji, or byte. However, the size of the data is verified before the data can be sent for processing. If the input data size is less than zero, which means no data, then it is considered invalid but if the data is more than zero, then the data will be sent to the next function, for example, check alphanumeric mode function for alphanumeric mode, check numeric mode function for numeric mode, check kanji mode function for kanji mode. Table 5.2 shows some of the pseudocode to analyze the alphanumeric data.

Table 5. 2: Pseudocode for Analyzing Alphanumeric Data type

```

Function check alphanumeric mode
Pass In: size, data
size = data size
data = input data
i = moveCount
  FOR each moveCount less than size, validate if the data is alphanumeric
    IF character from data == -1
      return false
    END IF
  Increment moveCount
  END FOR
  return true
END function

```

Table 5.2 shows pseudocode to analyze the input data for the alphanumeric mode.

5.3.1.3 Data encoding

The next process is to encode the input data. In this process, the program will obtain the length indicator of the input data and determine the version of the QR code that the data can become. The program will go through the encode mode function for each mode. For example, in the encoding mode function for alphanumeric mode, the input data will be sent to the look alphanumeric table function to refer to the alphanumeric table. In that function, there are several calculations have been done as described in Chapter 4, so that the input data can be encoded. The encoding process will be based on the data input modes. Table 5.3 shows pseudocode for alphanumeric encoding.

Table 5. 3: Pseudocode for Alphanumeric Encoding

```

Function encode mode alphanumeric
Pass In: version
version = qr code version
data = input data
size = size of input data
bs = bitsream
val = value of encode
i = moveCount
TRY
  words = size/2
  CREATE a new instance of QRbitstream into bs

```



```

Calling a method of the bs object named appendNum
FOR i = 0 less than words
    val = calling function look alphanumeric table (data [i*2] * 45)
    val += calling function look alphanumeric table (data[i*2+1] )
    Calling a method of the $bs object named appendNum
END FOR
IF size is not equal to 0
    val = calling function look alphanumeric table (data[words*2])
    calling a method of bs object named appendNum
END IF
Bstream = bs

```

Table 5.3 shows some of the processes for encoding the alphanumeric input data type.

5.3.1.4 Creating Error Correction Codewords

The next process is creating error correction codewords. In this process, there are some mathematic calculations involved, for example, division, multiplication, addition, modulo operation, and XOR operation. There are also algebra expressions involved such as Galois Field (GF), generate powers of 2, logs, antilogs, exponents, and alpha notation. To create the error correction codewords, the encoded data need to go through all the calculations involved, including converting those binaries (encoded data) into decimal where it is the coefficients of the message polynomial, dividing the message polynomial with generator polynomial, multiplying the generator polynomial with the lead term of the message polynomial, XORing the result with the message polynomial, and other calculations until the remainder are found. The remainder that results from the calculation is the error correction codewords for the original message polynomial. Table 5.4 shows some of the pseudocode code for this process.

Table 5. 4: Pseudocode for creating error correction codewords

```

function init_rs_char
Pass in: symsize, gfpoly, fcr, prim, nroots, pad
symsize = symbol size
gfpoly = galois field polynomial
fcr = first consecutive root in index form
prim = primitive element in index form
nroots = number of generator root
pad = padding bytes
alpha_to = log lookup table
index_of = antilog lookup table

```

```

genpoly = generator polynomial
mm = bits per symbol
nn = symbols per block
rs = reed-solomon codec
SET rs = null
IF (sysmsize is less than 0 OR sysmsize is more than 8) return rs
IF (fcr is less than 0 OR more than or equal to (shift the bits of 1, sysmsize steps to the left)) return
rs
IF (prim is less than or equal to 0 OR prim is more than or equal to (shift the bits of 1, sysmsize
steps to the left)) return rs
IF (nroots is less than 0 OR nroot more than or equal to (shift the bits of 1, sysmsize steps to the
left)) return rs
IF (pad is less than 0 OR pad nroot more than or equal to ((shift the bits of 1, sysmsize steps to the
left)-1-nroots)) return rs
CREATE a new instance of QRrsItem into rs
SET property in the rs object called mm = sysmsize
SET property in the rs object called nn = (shift the bits of 1, sysmsize steps to the left)-1)
SET property in the rs object called pad = pad
SET property in the rs object called alpha_to = array_fill(0, nn+1, 0)
SET property in the rs object called index_of = array_fill(0, nn+1, 0)
NN = NN bitwise AND nn
A0 = A0 bitwise AND NN
set property in the rs object called index_of[0] = A0
set property in the rs object called alpha_to[A0] = 0
sr = 1
FOR i=0 less than property in rs object called nn
    SET property in the rs object called index_of[sr] = i
    SET property in the rs object alpha_to[i] = sr
    shift the bits of sr, 1 steps to the left
    IF (sr & shift bits of 1, sysmsize steps to the left
        sr = sr bitwise XOR gfpoly
    END IF
    Sset sr = sr bitwise AND nn
END FOR
IF (sr is not equal to 1)
    SET rs = null
    return rs;
END IF
SET property in the rs object called genpoly = generator polynomial from its roots
SET property in the rs object called fcr = fcr
SET property in the rs object called prim = prim

```

```

SET property in the rs object called nroots = nroots
SET property in the rs object called gfpoly = gfpoly
SET iprim = 1;
for each prim-th root of 1 modulus prim not equal to 0
    SET iprim = iprim + property in the rs object called nn
SET property in the rs object called iprim = iprim/prim
SET property in the rs object called genpoly[0] = 1
FOR each root = for multiply prim, i=0 less than nroots
    SET property in the rs object called genpoly[i+1] = 1
FOR each j= i more than 0
    IF (property in the rs object called genpoly[j] is not equal to 0)
        SET property in the rs object called genpoly[j] = genpoly[j-1] XOR
        alpha_to[modnn(index_of[genpoly[j]] + root)]
    ELSE
        SET property of rs object called genpoly[j] = genpoly[j-1]
    END ELSE
END IF
SET property of rs object called genpoly[0] can never be zero
FOR each i=0 less than or equal to nroots
    convert property of rs object called genpoly[] to index form
END FOR
return rs
END function

```

Table 5.4 shows the pseudocode for the process of creating the error correction codewords.

5.3.1.5 Structuring Final Data

Next, the final data will be structured in a block. The program will determine how many blocks are required which include the error correction capability (ECC) level information, QR code version information, data length, ecc length, and how many error correction codewords are required to be structured in the block. Some of the pseudocode for this process is shown in Table 5.5.

Table 5. 5: Pseudocode to structure final data in a block

```

input = input data
ret = return
datacode = input byte stream
version = qr version based on input
b1 = block 1

```

```

data length = length of input data
ecc length = length of error correction codewords
SET datacode = get byte stream of the input
IF datacode is null
    Throw exception null input string
END IF
access QRspec class method get ecc spec of the input
SET version = get version of the input
SET b1 = access QRspec class method rsBlockNum1
SET data length = access QRspec class method rsDataLength
SET ecc length = access QRspec class method rseccLength
SET ecc code = fills an array with ecc length starting at index 0
SET blocks = access QRspec class method rsBlocknum
SET ret = object in 'this' method named init
IF ret less than 0
    Throw exception block allocation error
ELSE
    Allocate data in block
END IF

```

Table 5.5 shows the pseudocode for the process of structuring final data in a block.

5.3.1.6 Converting Block into QR Matrix

After the data has been structured in a block, it will be placed into the QR matrix along with the alignment pattern, finder pattern, timing pattern, and separator. The finder pattern must be three as it is a QR code standard and will be located on the top left, top right, and bottom left to make the QR code readable. The position of these three patterns has been determined in the put finder pattern function. This pattern will always be located at that location no matter what version the QR code is because that is the standard for QR code. The separator position also has been determined which will be located around the finder pattern and is defined in the set() function for each finder pattern. The pseudocode to place finder pattern and separator is shown in Table 5.6

Table 5. 6: Pseudocode placement of finder pattern and separator

```

Function create frame
Pass in: version
yOffset = separator
v = version

```

```

vinf = version information
SET width = width of QR according to version
SET frame line = repeat string of "\0" according to width
SET frame = filling array of 0 with width and frame line
// Finder pattern location
access self's class method putFinderPattern(frame,0,0)
access self's class method putFinderPattern(frame, width - 7, 0)
access self's class method putFinderPattern(frame, 0, width - 7)
// Separator
SET yOffset = width - 7
FOR y= 0 to 6
SET frame[y][7] = "\0"
SET frame[y][width - 8] = "\xc0"
    SET frame[yOffset][7] = "\xc0"
    increment y
END FOR
SET pattern = repeat "\xc0", 8 times
access QRstr's class method set (frame, 0, 7, setPattern)
access QRstr's class method set (frame, width - 8, 7, setPattern)
access QRstr's class method set (frame, 0, width-8, setPattern)

```

The next item to be placed into the QR code matrix is the timing pattern. It is a two-line in a horizontal and a vertical form which will be located between the three finder patterns. The horizontal timing pattern will be placed on the 6th row of the QR code between the separators while the vertical timing pattern will be located on the 6th column of the QR code between the separators. Table 5.7 shows pseudocode for the timing pattern and alignment pattern placement.

Table 5. 7: Pseudocode to place timing pattern and alignment pattern

```

width = width of QR according to version
// Timing pattern
FOR i=1 to (width - 15)
    SET timing pattern frame[6][7+i] = a single byte string from (0x90 or (i bitwise AND 1))
    SET timing pattern frame[7+i][6] = a single byte string from (0x90 or (i bitwise AND 1))
    increment i
END FOR
// Alignment pattern
access self's class method putAlignmentPattern(version,frame,width)

```

As shown in Table 5.7, all the data that have been encoded in the previous process is placed in the QR matrix. The data then is located in the QR matrix starting from the

right-bottom of the matrix and proceeding upward (zig-zag) as depicted in Figure 5.8. Figure 5.8 depicts how the data module is arranged inside the QR matrix.

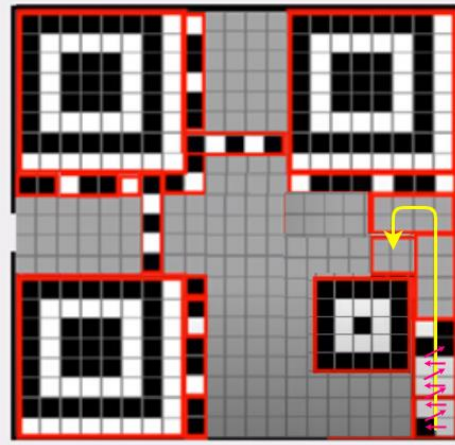


Figure 5. 8: Data module is structured in QR matrix

5.3.1.7 Apply Mask Pattern

Data masking is a process that is applied to the matrix after the data module is structured in the QR matrix as elaborated in section 5.3.1.6. This process is known as applying mask pattern. Applying a mask pattern is the process of changing the color of the data module. For example, the data module in the QR matrix is in a light module, it should be changed to a dark module, same with the dark data module, it should be changed to a light module. In other words, toggle the data module color. This mask pattern should only be applied to the data modules and error correction module, other than that such as finder pattern, alignment pattern, and others should not be masked. Some of the pseudocode is shown in Table 5.8.

Table 5. 8: Pseudocode to mask data

```

maskObj = mask object
CREATE a new instance of QRmask into maskObj
If (mask is less than 0)
  If (QR_FIND_BEST_MASK)
    SET masked = a method of maskObj object named mask(width, frame, calling a method
    of input object named getErrorCorrectionLevel)
  END IF
ELSE
  SET masked = a method of maskObj object named makeMask (width, frame, (remainder
  of QR_DEFAULT_MASK divided by 8), calling a method of input object named

```



```

        getErrorCorrectionLevel)
    END ELSE
END IF
ELSE
    SET masked = a method of maskObj object named makeMask(width, frame, mask, calling
        a method of input object named getErrorCorrectionLevel)
END ELSE
IF (mask equal to NULL)
    Return NULL
END IF

```

5.3.1.8 Apply Version and Format Information

In this process, the format and version information will be placed in the QR matrix. The version and format information are generated in a string form. The format information will contain the encoded error correction capability (ECC) level and mask pattern used in the QR code. It will be located below the topmost finder patterns and to the right of the leftmost finder pattern. While the version information will contain the version of the QR code and it is always placed beside the finder pattern without considering the size of the QR code. There are some mathematical calculations involved in this process. Table 5.9 shows some of the pseudocode for these two processes.

Table 5. 9: Pseudocode to apply version and format information

```

//Version Information
v = version
vinf = version information
If version is more than or equal to 7
    SET vinf = access self's class method getVersionPattern(version) to check version
    SET v = vinf
    FOR x=0 to 5
        FOR y=0 to 3
            SET frame[(width - 11) + y][x] = a single byte string from (0x88 or (v bitwise AND 1))
            SET v = shift the bits of v, 1 steps to the right
            Increment 1
        END FOR
    END FOR
SET v = vinf
FOR y=0 to 5
    FOR x=0 to 3
        SET frame [y][x+ (width - 11)] = a single byte string from (0x88 or (v bitwise AND 1))
        SET v = shift the bits of v, 1 steps to the right
    END FOR
END FOR

```

```

    END FOR
  END FOR
  SET frame[width - 8][8] = "\x81";
  return frame
// Format Information
Function writeFormatInformation
Pass In: width, frame, mask, level
v = version
SET blacks to 0
SET format = access QRspec class method getFormatInfo(mask, level)
FOR i=0 to 7
  If(format & true)
    SET blacks = blacks + 2
    SET v = 0x85
  END IF
  ELSE
    SET v = 0x84
  END ELSE
  SET frame[8][width - 1 - i] = character of v
  IF(i less than 6)
    SET frame[i][8] = character of v
  END IF
  ELSE
    SET frame[i+1][8] = character of v
  END ELSE
  SET format = shift the bits of format, 1 steps to the right
END FOR
FOR i=0 to 6
  If (there is format & true)
    SET blacks = blacks + 2
    SET v = 0x85
  END IF
  ELSE
    SET v = 0x84
  END ELSE
  SET frame[width - 7 + i][8] = character of v
  IF(i is equal to 0)
    SET frame[8][7] = character of v
  END IF
  ELSE
    SET frame[8][6 - i] = character of v
  END ELSE
  SET format = shift the bits of format, 1 steps to the right

```



```

END FOR
RETURN blacks
END function

```

Then, the quiet zone is applied around the QR matrix to form a complete QR code. Figure 5.9 shows the generated QR code.



Figure 5. 9: Generated QR code

To combine a logo inside the QR code, there are some properties that need to be added. Table 5.10 shows the pseudocode to resize the UTeM's logo, set position and set image transparency.

Table 5. 10: Pseudocode resizing logo image and set logo image transparency

```

Filename = name of the file
SET path_qr-code = 'qr-image/' append filename
SET path_logo = 'qr-generator/logo-utem.png'
SET qr = function imagecreatefrompng
    Pass In: path_qr-code
SET logo = method imagecreatefrompng
    Pass In: path_logo
//Get qr-code and logo size
SET qr_width = imagesx
    Pass in: qr
SET qr_height = imagesy
    Pass in: qr
SET logo_width = imagesx
    Pass in: logo
SET logo_height = imagesy
    Pass in logo
SET target = imagecreatetruecolor
    Pass In: width, height
Calling method imagesavealpha
    Pass In: target, true

```

```

SET divisor = 5
// Resize image size
SET cal_logo_width = qr_width/divisor
SET scale = logo_width/cal_logo_width
SET cal_logo_height = logo_hight/scale
SET from_width = (qr_width – cal_logo_width)/2
// Set transparency of image
SET transparent to imagecolorallocatealpha
    Pass In: target, value of red component, value of green component, value of blue
    component, value of alpha component
Calling method imagefill
    Pass In: target, qr, x-coordinate of start point, y-coordinate of start point, transparent
Calling method imagecopy
    Pass in: target, qr, x-coordinate of target, y-coordinate of target, x-coordinate of qr, y
    coordinate of qr, qr_width, qr_height
Calling method imagecopyresampled
    Pass in: target, logo, from_width, from_width, 0,0, cal_logo_width, cal_logo_height,
    logo_width, logo_height
Calling method imagepng
    Pass in: target, path of folder to save

```

In this process, there are 5 subprocesses involved. First, the path will be defined to save the generated QR code and the UTeM logo to be embedded with the QR code. Second, in order to avoid the original image is altered, a new image of QR code and the logo is created using the function `imagecreatefrompng()`. Third, the program will create a true-color image and assign transparency for the two new images so that it will be easier to blend between the two images. Fourth, to determine the position of the logo inside the QR code, there is some calculations that need to be done, which shows in the ‘//Resize image size’ comment. Fifth, the logo is combined with the QR code, and stored to the specific path defined in the first subprocess. The output of these subprocesses is shown in Figure 5.10. After going through several subprocess, the logo has been combined with the QR code and the program will save the QR code image with the logo above it to the specific path as defined in the program. Figure 5.10 shows the QR code with UTeM logo in the middle.



Figure 5. 10: QR code image with UTeM logo

Figure 5.10 shows a new QR code image that will be embedded in the document. The embedding into document process will be explained in Section 5.3.3.

5.3.2 Steganography Image Generator

This process is another alternative that act as a watermark and at the same time it is actually a steganography image that contains the ownership information. The ownership information will be encoded and will be injected into the image using the least significant bit (LSB) method. This steganography image then will be embedded into the document which is described in Section 5.3.3. Figure 5.11 shows the flowchart of the main process for the steganography image generation.

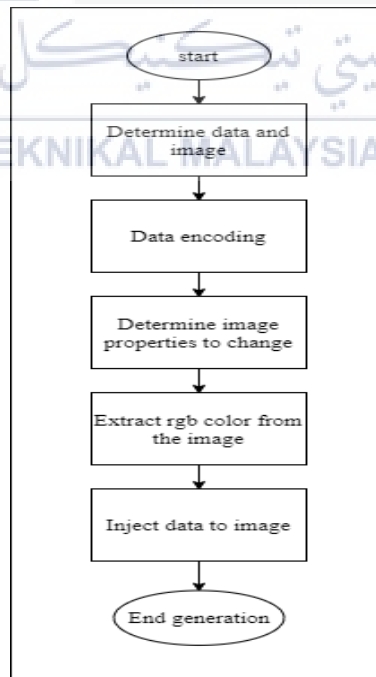


Figure 5. 11: Flowchart of steganography image generation

The detailed implementation each of the processes for the steganography image generation is described in the subsection.

5.3.2.1 Determine data and image

The data to be injected into the image is determined by taking the same data as the data that have been encoded in the QR Code generator module. The data includes user id, download date, download by, matrix number, email, file owner, and source of the file. While the image that will be the carrier for this information is decided to be UTeM logo. The data and image are then will be passed to one function which is called as encode message. Table 5.11 shows the pseudocode for this process.

Table 5. 11: Pseudocode for determine data and image

Function combine qr pdf
Pass In: user_id, material_location, last_qr_id, display
SET message to display
SET file to 'utem.png'
Calling function encode message
Pass In: message, file
Return function

5.3.2.2 Data Encoding

In this process, each character of the data will be converted to ASCII code equivalent integer value. Then, from that integer value, it will be converted again into a binary number in 8 bit for each character. In PHP, the ord() function can be used to convert the character that represents its ASCII code. This function will return an integer value and this value will be converted into binary using decbin() function. The only problem when using the decbin() function is it only returns a binary number with the most significant digit. For example, the binary should be **01110101** (8 bits) but with this function, it will return **1110101** (7 bits) where it will remove the zero on the left side if available. In order to force the result to return a total of 8 bits per character, the str_pad() function can be used. Then, the binary value of **00000011** which represent as **'end of text'** in ASCII table will be appended with the last binary string for easier in decode process to find the end

text of the message. Table 5.12 shows the pseudocode to encode the data and inject the 'end of text' binary value to the end of the binary message string.

Table 5. 12: Pseudocode for encoding message

```

Function encode message
Pass In: file, message
SET binary message to null
SET moveCount to 0
  FOR each row of the message
    SET character of message to its equivalent integer ASCII value
    SET integer of character to binary and force return 8 bits using str_pad
    Increment moveCount
  END FOR
SET binary message append with the binary string to 00000011

```

5.3.2.3 Determine image properties

In this process, the image that has been pass in the encode message function can be edited as required. In this project, the image that is passed in this function will act as a background but actually, it will contain a message hidden in it. In order to prevent the image from interfering with the content of the document, the properties such as opacity and transparency of the image can be edited. The opacity and transparency properties for this image are set up to the lowest so that it will not bother the content in the document. There are some other function used which is `Imagecreatefrompng()` function, where it is used to load the image into memory. `Imagealphablending()` function is used to set the blending mode for the image, if the blend mode is true then the blending mode is enabled otherwise not enabled. `Imagesavealpha()` function is used to find out whether to save the alpha channel (transparency information) or not. `Imagefilter()` function is used to apply a filter to the image, for example, the opacity. `Imagecolorallocatealpha()` function is used to allocate the transparent color to the new image and `imageFill()` function to apply the transparency to the image. Table 5.13 shows the pseudocode for this process.

Table 5. 13: Pseudocode to reset image properties

```

Function encode message
Pass In: file, message
/*-- other previous process ---*/
SET image to function imagecreatefrompng
  Pass In: file

```

SET opacity to 0.1
Calling function Imagealphablending
Pass In: image, false
Calling function imagesavealpha
Pass In: image, true
SET transparency to 1 - opacity
Calling function image filter
Pass In: image, image filter type, color appearance parameter, transparency
SET width to function imagesx
Pass In: image
SET height to function imagesy
Pass In: image
SET target to imagecreatetruecolor
Pass In: width, height
Calling function imagesavealpha
Pass In: target, true
SET transparent to imagecolorallocatealpha
Pass In: target, value of red component, value of green component, value of blue component, value of alpha component
Calling function imagefill
Pass In: target, x-coordinate of start point, y-coordinate of start point, transparent

5.3.2.4 Extract RGB color from the image

After the image has been edited as appropriate for the project, the image's RGB color channel will be extracted. The reason to extract this value is to inject the encoded message into the least significant bit (LSB) of the blue channel only since the blue channel is the lesser color where human eyes are sensitive. In this process, the `imagecolorat()` and `Imagecolorsforindex()` function is the main and is used to retrieve the pixel index's color value for red, green, blue, and alpha. The programs will start retrieve the color index value starting from top left to right and then repeat for each row of the image until all the image's pixel has been going through. The Pseudocode is shown in Table 5.14.

5.3.2.5 Inject data to image

After the index color for the RGB has been obtained, the blue channel index color will be converted into binary using the `decBin()` function. Apart from that, the `str_pad()` function is also used to ensure that the result is returned to 8 bits. For each string blue binary value minus 1, will be replaced with the binary message (ownership information

that has been converted). The program will continue in a loop until all the blue binary value has been replaced with all the binary message. A `bindec()` function is then will be used to convert the new binary value that consists of the binary message to decimal. Then, `imagecolorallocatealpha()` and `imagesetpixel()` function is used to inject the new color back into the image and produce a steganography image. The image is then will be saved in the local storage as 'encoded_utm.png' and will be called in the next process which is embedding images into the document. Table 5.14 shows the pseudocode for the extraction of the RGB channel, conversion of the blue channel to binary, and injection of the encoded message to the LSB of the blue channel.

Table 5. 14: Pseudocode for extract rgb color of the image & inject data to image

<pre> Function encode message Pass In: file, message /*-- other previous process ---*/ SET messagePosition to 0 SET moveCount to 0 FOR each row of the height FOR each column of the width IF binary message [message position] is not null Break 2; ELSE SET rgb to function imagecolorat Pass In: image, moveCount height, moveCount, width SET colors to function imagecolorsforindex Pass In: image, rgb SET red to red SET green to green SET blue to blue SET alpha to alpha SET binaryBlue to Convert the blue to binary SET binaryBlue[string length(binary value) - 1] to binaryMessage[messagePosition] SET newBlue to blue color with message SET newColor to imagecolorallocatealpha Pass In: image, red, green, newBlue, alpha Calling function imagesetpixel Pass In: image, moveCount height, moveCount width, newColor Increment messagePosition </pre>

```

    END IF
    END FOR
  END FOR

  SET newImage to 'encoded_utm.png'
  Calling function imagepng
    Pass In: image, newImage, compression level value
  Calling function imagedestroy
    Pass In: image;

```

5.3.3 Embedding The Images into Document

In this process, the document and page for inserting the QR code image, the steganography image, and the watermark image are determined. Then, the images are imported from the local storage and embedded in the selected document. Figure 5.12 shows the flowchart for embedding the images into the document process.

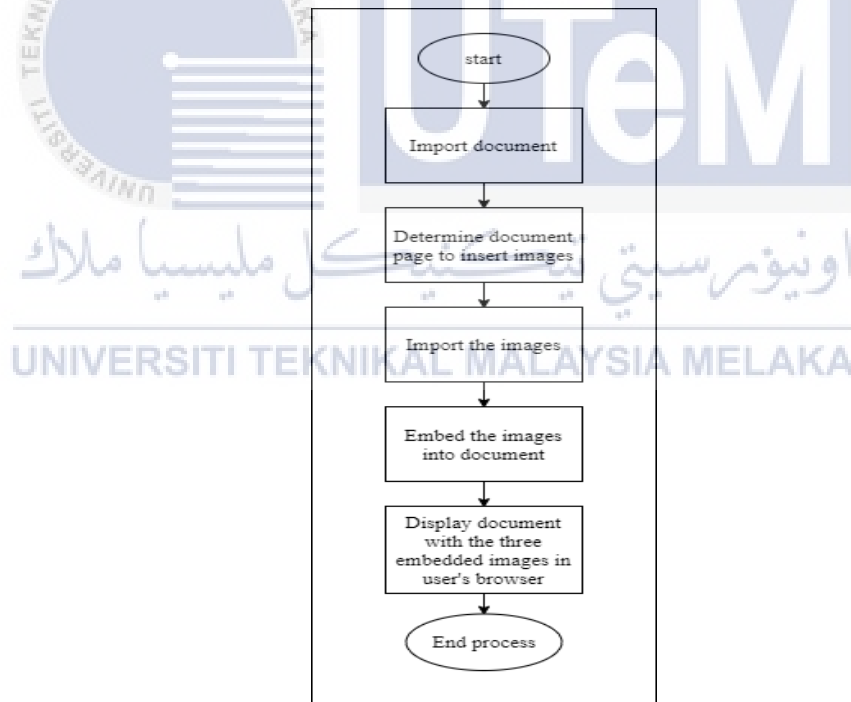


Figure 5. 12: Flowchart for Embedding the images into a document

The process flow for embedding the QR code image, steganography image, and watermark image is shown in Figure 5.12. To illustrate the detailed process of QR code image and steganography image is created with the addition of watermark image and embedded in the document, a prototype is developed as discussed in the subsection.

5.3.3.1 Import Document

The document that the user ‘clicked or ‘choose’ is referred to as the imported document where it can be a lecture note file, lab material file, and student assignment file. The interface where the document can be acquired is shown in Figure 5.13, Figure 5.14, Figure 5.15, Figure 5.16, and Figure 5.17.

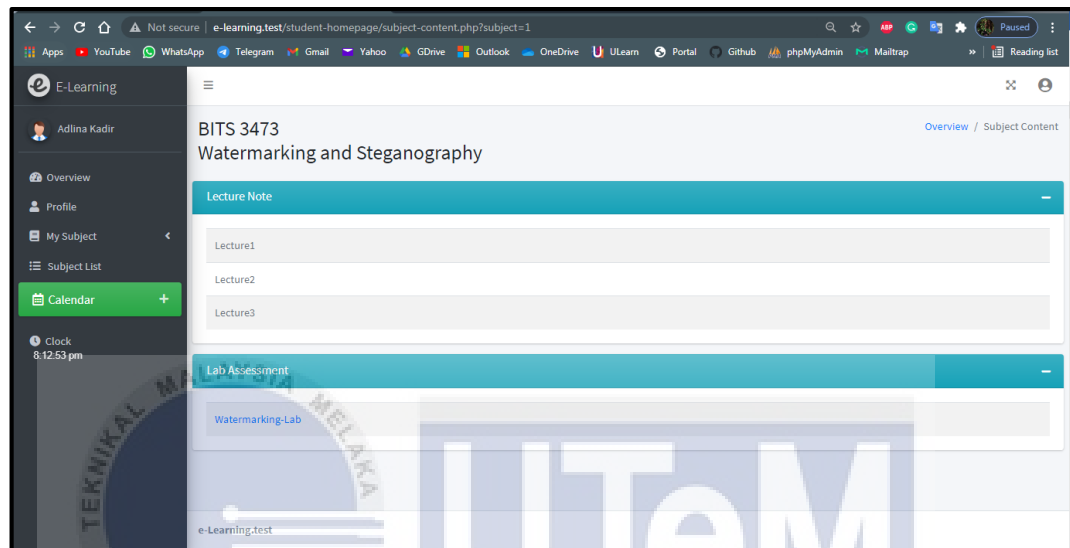


Figure 5.13: Student user interface to download lecture note

Figure 5.13 shows the user interface where the student can download the lecture note. This interface also includes the lab assessment part where students can view the lab submission opened by their lecturer.

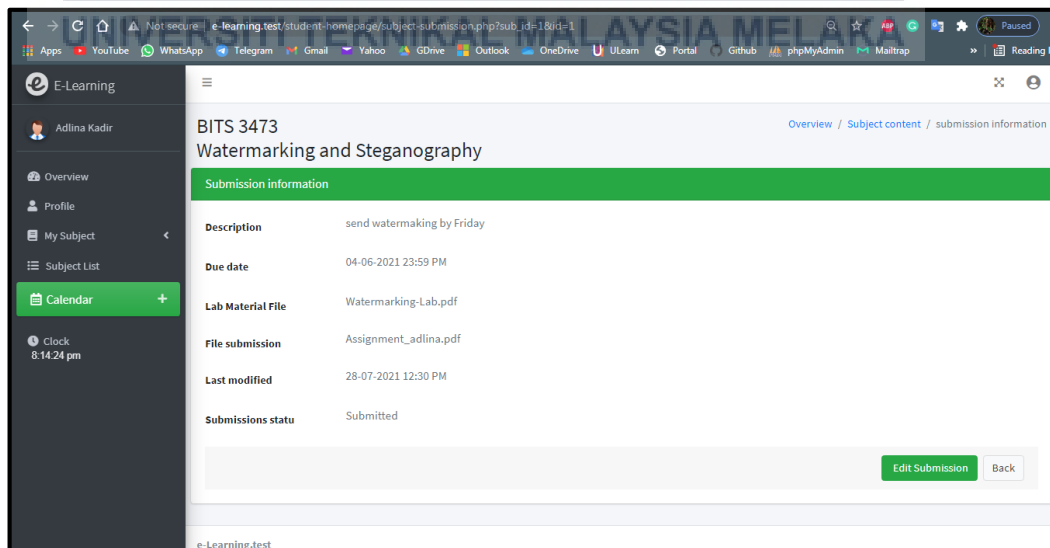


Figure 5.14: Student user interface to download lab material and submit assignment

Figure 5.14 shows the user interface for the student to view the details of the lab submission opened by their lecturer. This interface consists of the submission description, due date of the lab submission, lab material file that can be downloaded, file submission (student submission file), last modified date, and status of the submission. In this user interface also student can edit their submission in case they mistakenly uploaded other files than their real assignment file. The two material that can be downloaded from this interface is the lab material file and student submission file.

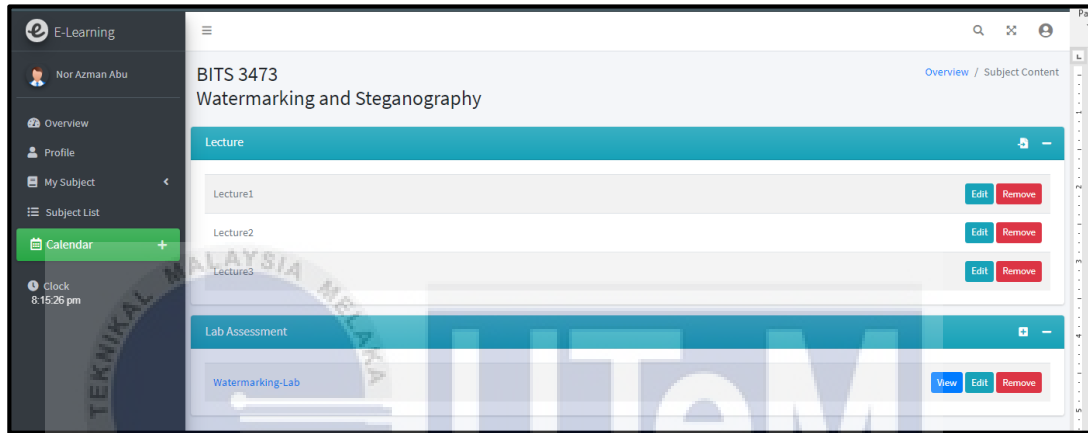


Figure 5. 15: Lecturer user interface to manage their material

Figure 5.15 shows the user interface for lecturer to manage their material. In this user interface, the lecturer can add, edit or delete the lecture note or the lab document. Lecturers also can download the material that they uploaded to the system as in this user interface.

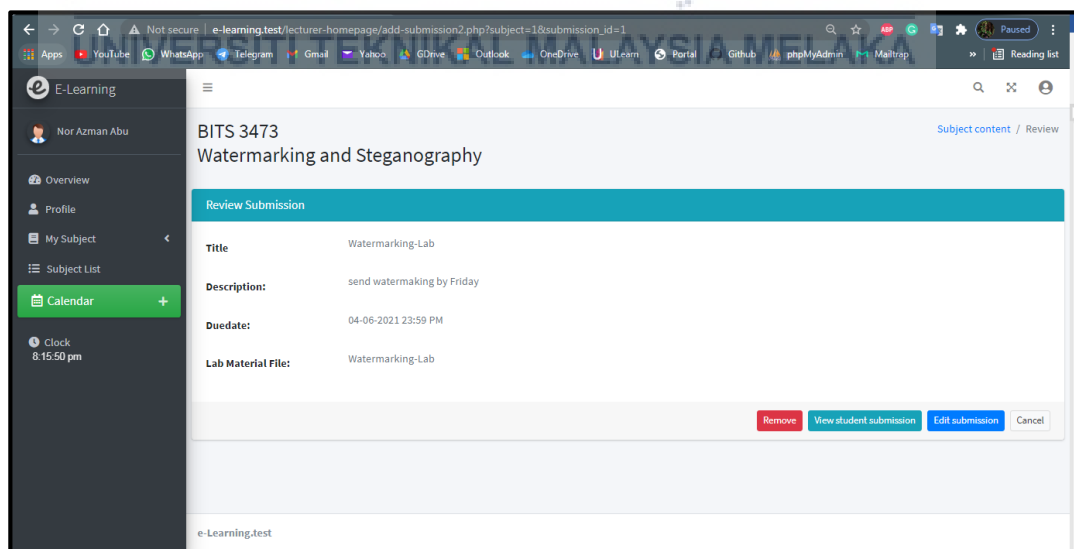


Figure 5. 16: Lecturer user interface to review open submission

Figure 5.16 shows the user interface for the lecturer to edit the lab assessment part. In this user interface, the lecturer can edit the information such as the submission title, submission description, due date and even replace the lab material file. In this user interface, the material that can be downloaded by the lecturer is the lab material file.

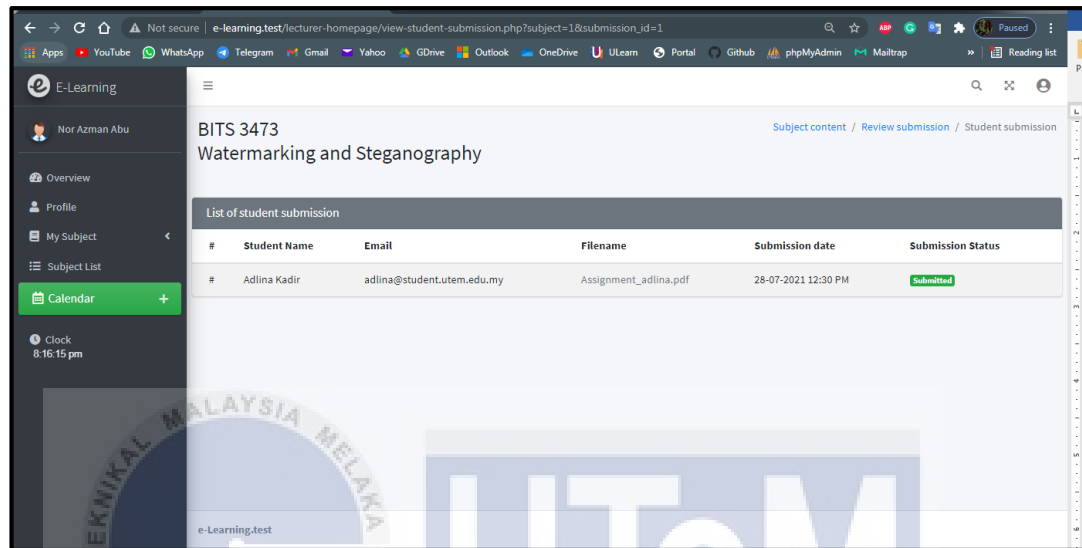


Figure 5. 17: Lecturer user interface to review student submission

Figure 5.17 shows the user interface for the lecturer to review the student who has submitted their assignment. In this user interface, the lecturer can download the student assignment file to review later. A javascript is used to pass the chosen material once the user clicked the material in the system. Table 5.15 shows the pseudocode for the javascript to pass the material and user id to the embed the images into document module.

Table 5. 15: Javascript Pseudocode to pass material and user id

```

Function combine_pdf_qr
Pass in: id
    SET user_id = SESSION['user_id']
    SET material_location = id
Ajax request
    SET url: _pass_data: "../qr-generator/comne-qr-pdf.php"
    SET method: "POST"
    SET data:
        user_id: user_id
        material_location: material_location
    IF success
        Open new windows with downloaded pdf
    END IF
END request
END function

```

Based on javascript pseudocode in Table 5.15, the code will send the current user id and the material chosen by the user to another file (another module) that will embed the images into the chosen document. Table 5.16 depicts the pseudocode for passing the current user id and material selected by the user from the javascript before being obtained in the embed images into document module.

Table 5. 16: Pseudocode to obtain user id and material location

```
SET connect = database connection
SET user_id = POST['user_id']
SET material_location = POST['material_location']
```

In this process, the POST method is used to get that information because that information is actually coming from the database and it is store in the variable stated. The material chosen by the current user then will be passed to the setSourceFile() function to check the file information. Table 5.17 shows some of the pseudocode to check file information.

Table 5. 17: Pseudocode to check the file information

```
Function setSourceFile
Pass in: file
SET currentReaderId = a method getPdfReader
    Pass in: file
SET objectsTocopy of the currentReaderId = a new element on the end of the array
SET pdf reader = getPdfReader of the currentReaderId
SET pdf version of the getPdfVersion
Return number of page
```

Based on Table 5.17, the program first will read the imported pdf document information such as the pdf reader id, pdf version, and pdf pages to ensure that it is a pdf file. Lastly, this program will return the number of pages of the source file, so that the page to insert the images in the next process can be decided.

5.3.3.2 Determine document page to embed the images

After the path of the material is known, the next process is determining on which pages the QR code image, the steganography image, and the watermark image will be inserted. The function used for this process is importPage(). A loop will be made to iterate

through all the pages. Table 5.18 shows the pseudocode for determining pages where to insert the stated images.

Table 5. 18: Pseudocode for determining page to insert the images

```

pageCount = total number of pages of the document
pageNo = page number
SET pageCount = calling function setSourceFile("../path_of_the_material")
FOR pageNo= 1 to <= pageCount
    SET template = importPage
        Pass in: pageNo
    SET size = getTemplateSize
        Pass in Template
    IF (size[0] > size[1])
        SET page as landscape
    END IF
    ELSE
        SET page as portrait
    END ELSE
// some other process

```

To specify which page to insert the QR code image, steganography image and watermark image, any number can be put inside the importPage() function. In this prototype system, the page to put the three images is on all pages of the document and the total pages are store in the variable of page number.

5.3.3.3 Import The Images

After the user's choice document has been imported and the page to insert the three images has been decided, the next process is to select which image will be inserted into the document that has been imported. The function used in this process is Image(). In this function, the path of the QR code image, steganography image, watermark image, the images position, and the size of the images to be inserted into the document are decided. The pseudocode for passing the path of the images and the resizing value is shown in Table 5.19.

Table 5. 19: Pseudocode for passing QR code image path and the resizing value

```

Calling a method of 'pdf' object named Image
Pass In: '/path_of_the_qr_image', 173, 4, 20, 20
Calling a method of 'pdf' object named Image
Pass In: '/path_of_the_steganography_image', 80, 100, 80, 50
Calling a method of 'pdf' object named Image
Pass In: '/path_of_the_UTeM's_logo', 20, 7, 28, 15

```

After the required value has been defined in the function, the program will check for the image file extension for example, .png or .jpg. Table 5.20 shows the pseudocode for checking the image file type.

Table 5. 20: Pseudocode for checking image file type

```

type = file extension type
mtd = method/function
w = width
h = height
x = x-position
y = y-position
empty = no value
info = file extension information
Function Image
Pass in: file path, x, y, w, h
IF file is empty
    Echo error message
IF file is not empty
    IF type is empty
        SET pos = find the position of the first occurrence of a substring "."
        IF there is no string after substring "."
            Echo error message
        END IF
    SET type = lower letter type
    IF type is 'jpeg'
        SET type to 'jpg'
    SET mtd = '_parse' append type
    IF mtd is not exist
        Echo unsupported message
        SET the file to info
    END IF
    SET info = object named images[file]
    IF w equal to 0 && h equal to 0 set automatic width and height
        SET w = -96
        SET h = -96

```

```

END IF
IF w less than 0
    SET w = width of the file in info * 72 / w
If h less than 0
    SET h = height of the file in info * 72 / h
IF w is equal to 0
    SET w = h * width of the file in info / height of the file in info
IF h is equal to 0
    SET h = w * height of the file in info / width of the file in info
IF y is empty
    IF y + h is more than PageBreakTrigger && not InHeader && not InFooter &&
    AcceptPageBreak
        SET x2 = object in 'this' method named x
        Call a method of 'this' object named AddPage
        SET object in 'this method' name x = x2
    END IF
    SET y = y + h
    Set object in 'this' method named y = y + h
END IF
IF x is empty
    SET x = object in 'this' method named x
    Call a method of 'this' object named _out to write the format using sprint
IF link is empty
    Call a method of 'this' object name Link to put the link on the page
END IF
END function

```

After the image file type has been determined, it will pass to the `_parsepng()` or `_parse.filetype()` function to check whether the image file can be opened or not. If the image file can be opened, the image file information will be extracted in the `_parsepngstream()` function. The pseudocode to check whether the file can be open or not is shown in Table 5.21.

Table 5. 21: Pseudocode for checking image file

```

file = image of qr code
rb = parameter mode for open file, r - read only, b - force in binary mode
f = file information in rb mode
info = contain f and file information
function _parsepng
Pass in: file
    Set f = calling fopen function
        Pass in file, rb
    IF file cannot be opened

```

```

        Echo error message
    ELSE
    SET info = calling a method 'this' object name _parsepngstream
        Pass in: f, file
    Calling fclose function
        Pass in: f
    Return info
END function

```

Next, the program will continue with the `_parsepngstream()` function, where the program will check the signature of the image file to confirm the file type by reading the header chunk. This function is also used to obtain other information needed such as palette, image transparency, image data block, and image trailer. The pseudocode to read the png image stream is shown in Table 5.22.

Table 5. 22: Pseudocode to read png image stream

```

file = image of qr code
f = file information in read only and binary mode
bpc = bits per character
ct = color type
type = image type
PLTE = palette
IDAT = image data
tRNS = transparent
IEND = image trailer
Function parsepngstream
Pass in: f, file
IF stream of f doesn't have character string of png signature
    Echo error message
END IF
Read 4 bytes from stream of f
    IF stream of f doesn't have 'IHDR'
        Echo error message
    SET w = read 4-byte integer from stream of f
    SET h = read 4-byte integer from stream of f
    SET bps = ASCII value 1-byte integer from stream of f
    If bpc is more than 8
        Echo error message
    ct = 1-byte integer from stream of f in ASCII value
    IF ct is 0 or 4
        SET colspace = "DeviceGray"
    ELSEIF ct is 2 or 6

```



```

        SET colspace = "Device RGB"
    ELSEIF ct is 3
        SET colspace = "Indexed"
    ELSE
        Echo message unknown color type
    IF 1-byte integer from f not equal to 0
        Echo message unknown compress method
    IF 1-byte integer from f not equal to 0
        Echo message unknown filter method
    IF 1-byte integer from f not equal to 0
        Echo message interlacing not supported
    Read 4-byte integer from stream of f
Do
    SET n = read integer from stream f
    SET type = read 4-bytes from stream f
    IF type is 'PLTE'
        SET pal = read n-bytes from stream f
        Read palette
    END IF
    ELSEIF type is 'tRNS'
        SET t = read n-bytes from stream f
        Read transparency info
    ELSEIF type is 'IDAT'
        SET data = read n-bytes from stream f
        Read image data block
    ELSEIF type 'IEND'
        Break
    ELSE read n-bytes + 4 from stream f
END do
// some other process

```

The information that is obtained from `_parsepngstream()` function also is used to determine whether the imported image's color type is grayscale or RGB type. The color transparency of the image file can be known here. This transparency will allow the images to blend in with the surrounding in the document. The pseudocode to determine the color type is shown in Table 5.23.

Table 5. 23: Pseudocode to determine color type of the image

```

ct = color type
f = image information in read only and binary mode
w = width of f in integer
h = height of f in integer

```

```

data = uncompressed data
IF ct is 4
  // gray image
  SET length = 2 * w
  FOR i=0 to less than h
    SET pos = (1+ length) * i
    SET color append = data for each pos index
    SET alpha append = data for each pos index
    SET line = return string of data, pos+1, length
    SET color append = replace '/(.)/s' pattern in line to '$1'
    SET alpha append = replace '/(.)/s' pattern in line to '$1'
  END FOR
END IF
ELSE
  //RGB image
  SET length = 4 * w
  FOR i=0 to less than h
    SET pos = (1+ length) * i
    SET color .= data for each pos index
    SET alpha .= data for each pos index
    SET line = return string of data, pos+1, length
    SET color .= replace '/(.{3})/s' pattern in line to '$1'
    SET alpha .= replace '/(.{3})/s' pattern in line to '$1'
  END FOR
END ELSE

```

After the image type has been determined, the program will compress the image with function of `gzcompress()`. This function will compress the image string using a ZLIB data format.

5.3.3.4 Embed The Images into Document

After that, the position and the size of the image to be inserted into the document are determined. This value of the position and size of the image has been defined in the previous process which is in the `Image()` function (Table 5.19). The QR code image, the steganography image, and the UTeM's logo as a watermark image will be embedded into the document using that value. Before embedding the three images into the document, the location to store the embedded document is determined. In the program, the embedded document will be stored in the local storage which is in "C:/laragon/www/E-

Learning/lecturer-homepage/material/stampedMaterial/". The pseudocode is shown in Table 5.24.

Table 5. 24: Pseudocode to define path for the embedded document

```
SET outputPath = "../lecturer-homepage/material/stampedMaterial/" append Material-
name append ".pdf"
```

The variable that stores the path of the embedded document is then being called in the Output() function to give the destination where the document needs to be sent. There are several options on what to do with the embedded document, 'I' to send the standard output to the browser, 'D' to download the file, 'F' to save the document to local, and 'S' to return the document as a string. Table 5.25 shows the pseudocode for calling Output() function.

Table 5. 25: Pseudocode for calling Output() function

```
Call a method of the pdf object named Output
Pass In: outputPath, "F"
Return outputPath
```

The 'F' option is chosen in the Output() function because there is already have another function called open() used in the code to open the downloaded file in a new tab. The program will return the result of the embedded process after it has completed all the processes. With the function of open(), the result of the document with an embedded QR code image, steganography image and UTeM's logo within it will be downloaded and displayed on the new tab of the user's browser. The pseudocode to return the result is shown in Table 5.26.

Table 5. 26: Pseudocode to return output as pdf document to a new window

```
Ajax request (
  SET url_to_pass_data: "../qr-generator/comne-qr-pdf.php"
  SET method: "POST"
  SET data:
    user_id: user_id
    material_location: material_location
  IF success
    Open new windows with downloaded pdf
  END IF
END request
```

Table 5.26 illustrates the javascript pseudocode for returning the result that the QR code generation, steganography image generation, and embedding the images into document processes are working fine. Figure 5.18 shows the result file.

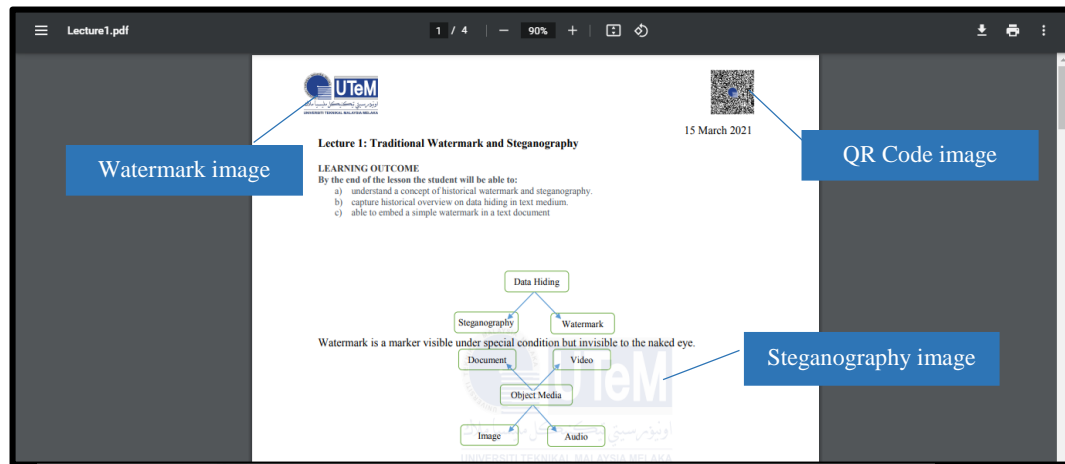


Figure 5. 18: Downloaded file

Figure 5.18 shows the result from embedding images into the document process. Figure 5.18 shows the QR code image, steganography image, and watermark image that is embedded in the document by representing the three images based on the value of the position and size determined.

5.4 Summary

This chapter has explained the configuration and setup of the software that is used in this project. The implementation of the three main modules are elaborated which are QR code generator, Steganography image generator, and embed the images into the document. The purpose of the first module is to generate a QR code image. There are several processes and subprocess in order to successfully generate the QR code image which will contain the ownership information and the downloading activities information. For the second module, the aim is to inject the ownership and downloading activities information in the image carrier, which will be formed into a steganography image. For the third module, the goal is to embed the QR code image, steganography image, and watermark image into the document. In the next chapter, the documentation of the testing part for this project will be described.

CHAPTER 6: TESTING

6.1 Introduction

In the previous chapter, the software development setup, the system architecture, and the implementation of the main module have been described. In this chapter, the testing will be conducted on the generation of the QR code, generation of the steganography image, embedding QR code image, steganography image and watermark image into the document, and testing on the QR code image and the steganography image itself. On the QR code image testing, there will be two parameters for the testing which are data completeness and usability. While on steganography image testing, the parameters for the testing are data completeness and imperceptibility.

6.2 QR Code Generator Testing

In this testing part, the QR code generator will be tested to see if it can successfully generate a QR code image. The process of conducting the QR code generator testing is shown in Figure 6.1.

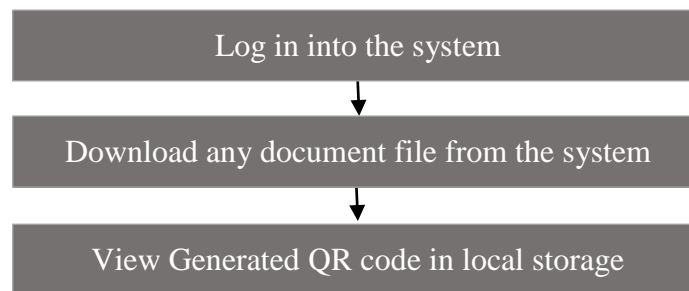


Figure 6. 1: QR code generator testing process

Before starting the process in Figure 6.1, the QR code generator testing is divided into two parts: student and lecturer. There are three tests for each part, which are on the lecture note document, lab document, and student's submission document, which total of six tests. To

start the QR code generator testing, first need to log in to the system and click any lecture note as shown in Figure 6.2. The first testing will be started from the student side, named ‘Adlina Kadir’.

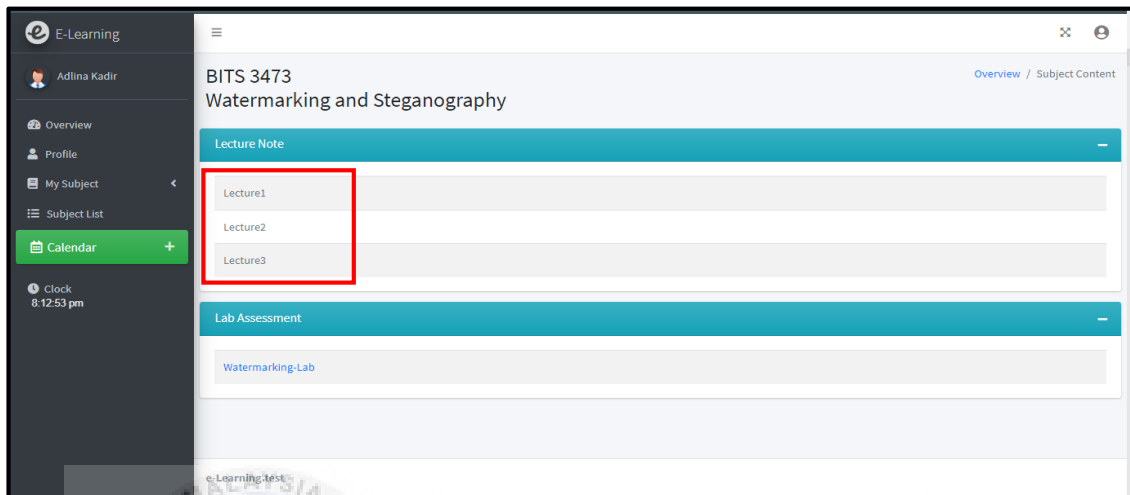


Figure 6. 2: Download lecture note from the student side

From Figure 6.2, once the lecture note in the ‘Lecture Note’ tab is downloaded, the generated QR code should be generated and stored in the local storage, for example, “C:\laragon\www\E-Learning\qr-generator\qr-image”. Figure 6.3 shows the generated QR code of user ‘Adlina Kadir’ is saved in the local storage.

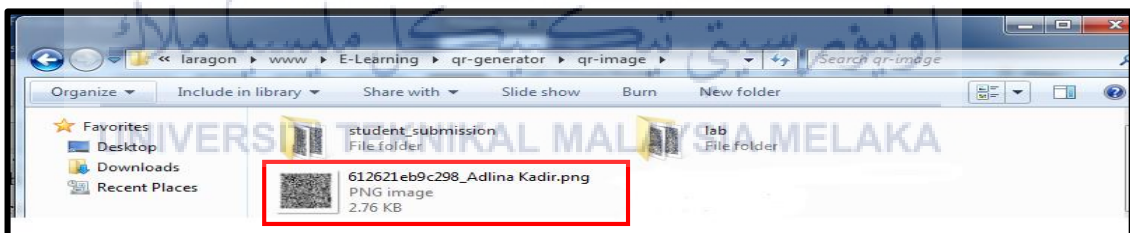


Figure 6. 3: Generated QR code for lecture note (student)

While for the QR code with UTeM’s logo in the middle will be saved, for example in “C:\laragon\www\E-Learning\qr-generator\qr-image-with-logo” as shown in Figure 6.4.

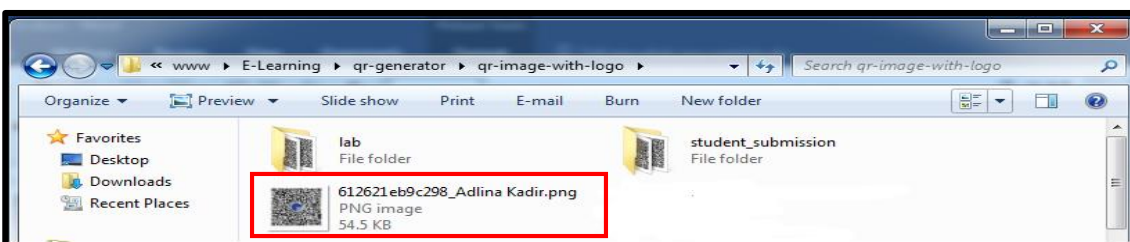


Figure 6. 4: Generated QR code with UTeM’s logo for lecture note (student)

In the second testing, the student will download the lab document as shown in Figure 6.5.

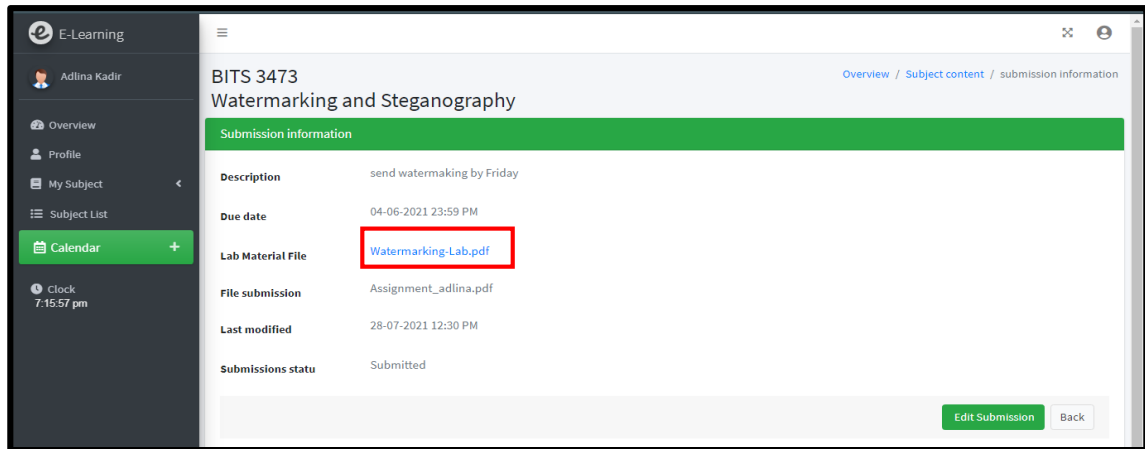


Figure 6. 5: Download Lab document from the student side

Based on Figure 6.5, once the lab document is downloaded, the QR code for that respective student will be generated and will be saved for example in, “C:\laragon\www\E-Learning\qr-generator\qr-image\lab” as shown in Figure 6.6.

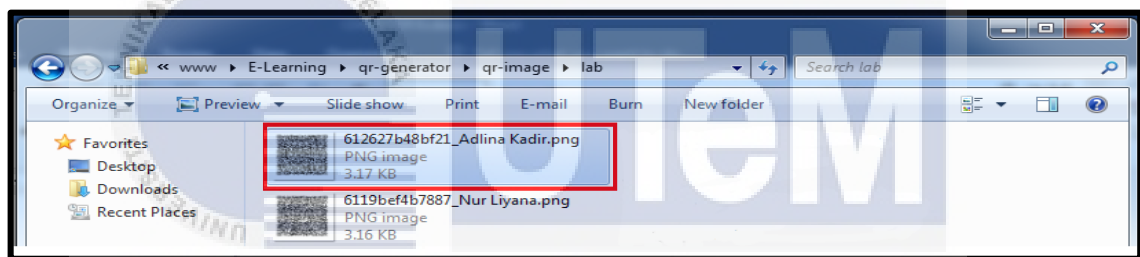


Figure 6. 6: Generated QR code for lab document (student)

For the QR code with UTeM’s logo in the middle, the path of where it is saved in the local storage for example will be “C:\laragon\www\E-Learning\qr-generator\qr-image-with-logo\lab” as shown in Figure 6.7.

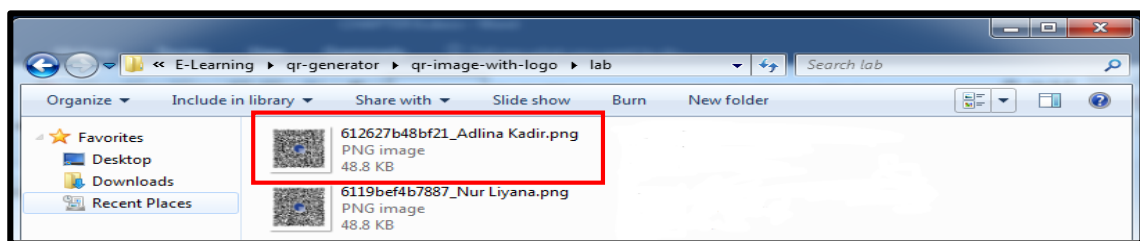


Figure 6. 7: Generated QR code with UTeM’s logo for lab document (student)

In the third testing, the student will download their own submission document as shown in Figure 6.8.

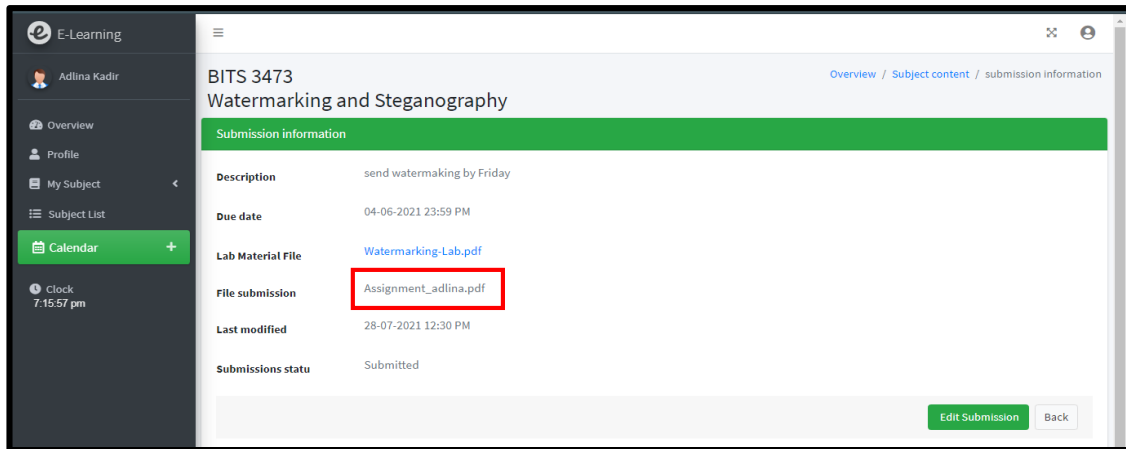


Figure 6. 8: Download student submission document from the student side

From Figure 6.8, after the student’s submission document is downloaded, the generated QR code should be saved in the local storage for example in, “C:\laragon\www\E-Learning\qr-generator\qr-image\student_submission”. Figure 6.9 shows the generated QR code of user ‘Adlina Kadir’ after the user downloads their own submission document.

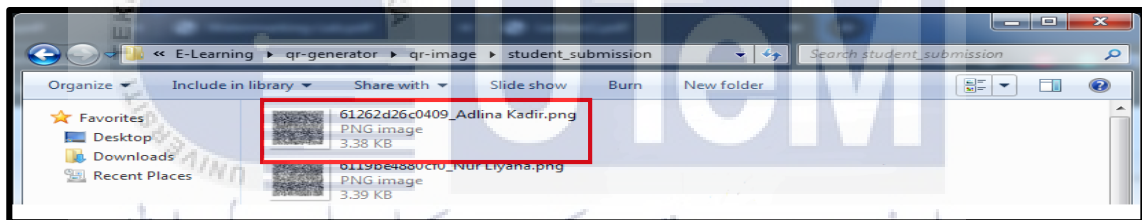


Figure 6. 9:Generated QR code for student submission document (student)

Meanwhile for the QR code with UTeM’s logo in the middle will be saved in the directory for example, “C:\laragon\www\E-Learning\qr-generator\qr-image-with-logo\student_submission” as shown in Figure 6.10.

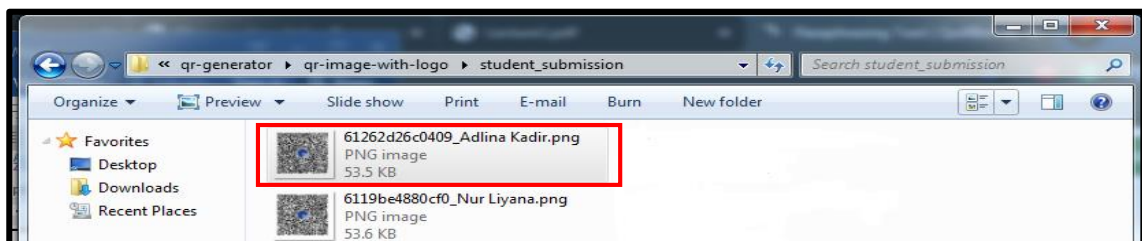


Figure 6. 10: Generated QR code with UTeM’s logo for student submission document (student)

For the fourth testing, the role is changed to the lecturer. The lecturer named ‘Nor Azman Abu’, will download his lecture note document as shown in Figure 6.11.

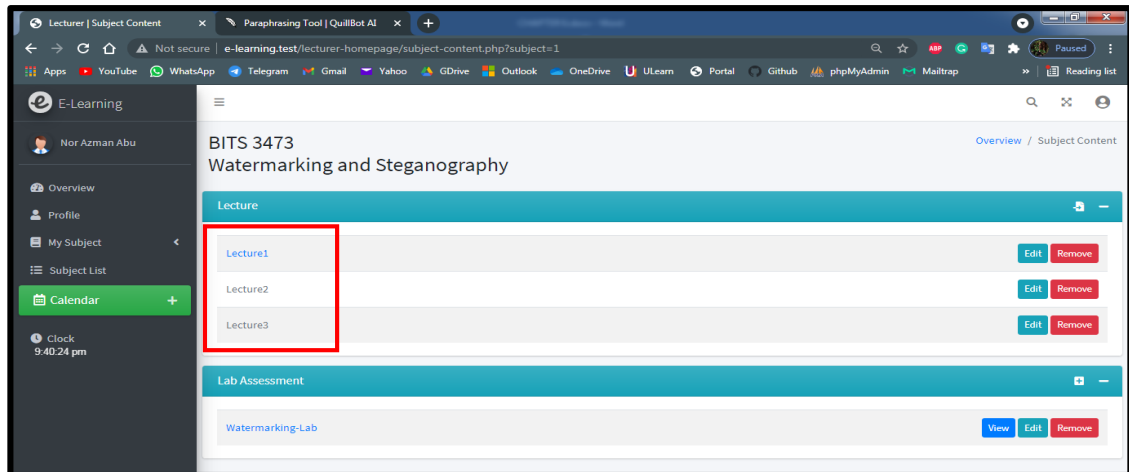


Figure 6. 11: Download lecture note from lecturer side

The generated QR code is saved in the directory such as “C:\laragon\www\E-Learning\qr-generator\qr-image” as soon as the lecture note document in Figure 6.11 is downloaded. The generated QR code for the lecturer is saved in the local storage, as shown in Figure 6.12.

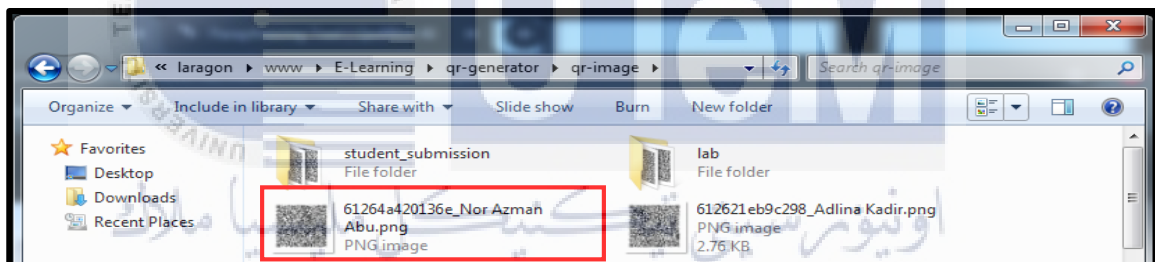


Figure 6. 12: Generated QR code for lecture note (lecturer)

While, as shown in Figure 6.13, the QR code with UTeM’s logo in the middle is saved for example in the “C:\laragon\www\E-Learning\qr-generator\qr-image-with-logo” directory.

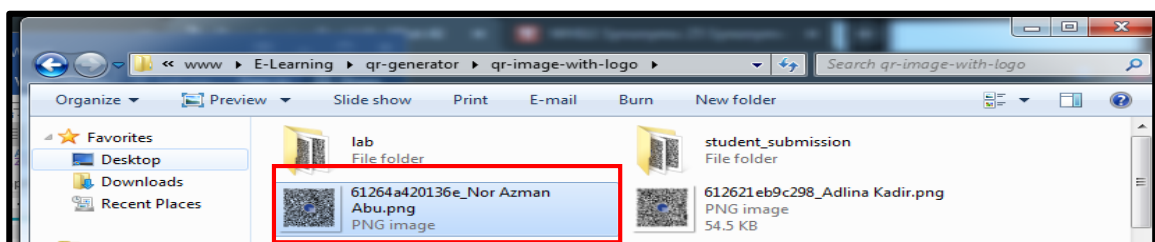


Figure 6. 13: Generated QR code with UTeM’s logo for lecture note (lecturer)

For the next testing, the lecturer will download the lab document, as shown in Figure 6.14.

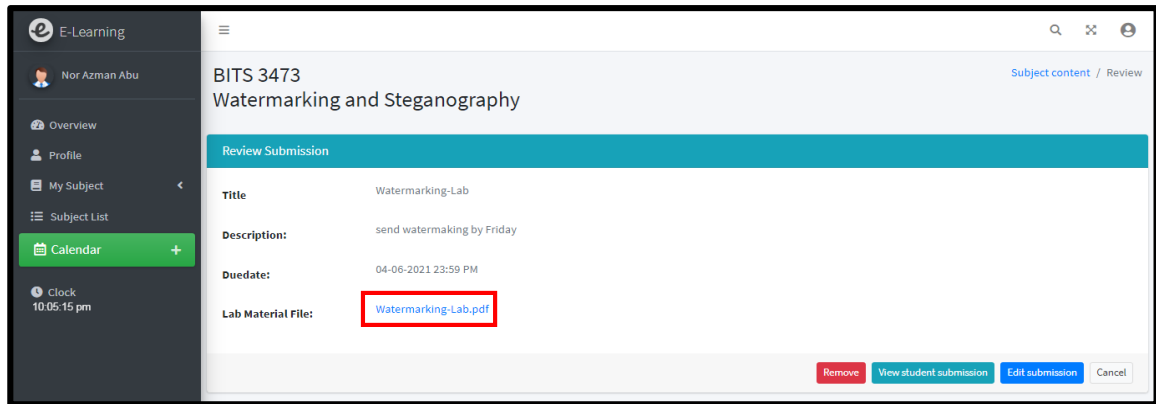


Figure 6. 14: Download Lab document from lecturer side

Based on Figure 6.14, the QR code for the lecturer named 'Nor Azman Abu' will be generated and kept in a directory such as "C:\laragon\www\E-Learning\qr-generator\qr-image\lab" after the lab document is downloaded, as shown in Figure 6.15.

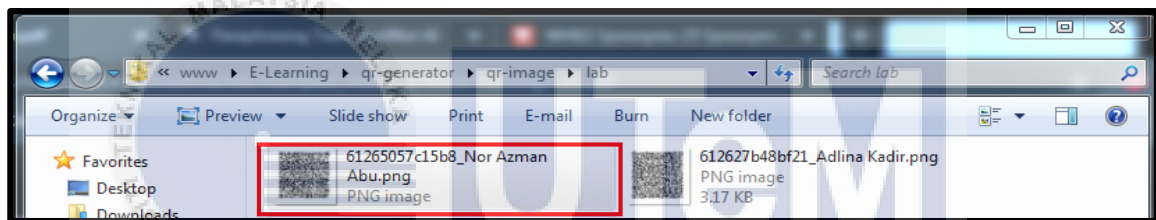


Figure 6. 15: Generated QR code for lab document (lecturer)

Meanwhile for the QR code image with UTeM's logo within it is stored as in "C:\laragon\www\E-Learning\qr-generator\qr-image-with-logo\lab" directory as shown in Figure 6.16.

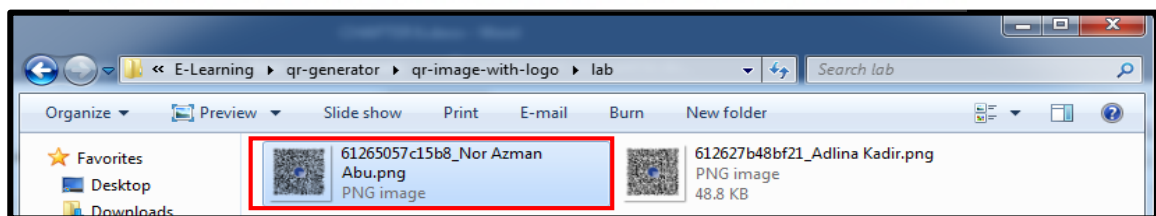


Figure 6. 16: Generated QR code with UTeM's logo for lab document (lecturer)

The last part of the QR code generator testing is, the lecturer will download the document submitted by their student. In this example testing, the lecturer named, 'Nor Abu Azman', subject taught, 'Watermarking and Steganography' and student enrolled in that subject named, 'Adlina Kadir'. Once the student uploads their assignment document into the submission system, the lecturer can see and download the submission document as shown in Figure 6.17.

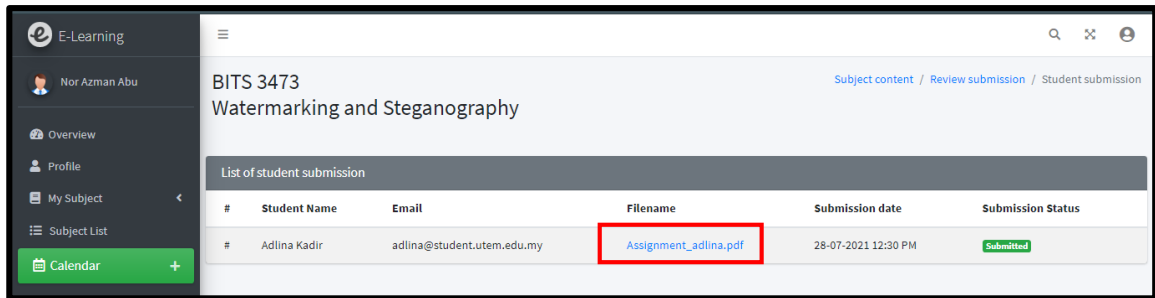


Figure 6. 17: Download student submission document from lecturer side

From Figure 6.17, the moment the student’s submission document is downloaded, the generated QR code is saved in the local storage, for example in the directory “C:\laragon\www\E-Learning\qr-generator\qr-image\student_submission”. Figure 6.18 depicts the generated QR code for the lecturer as soon as the lecturer downloaded the student’s submission document.

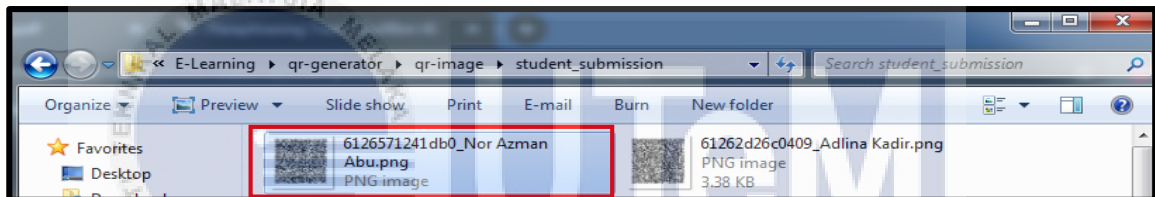


Figure 6. 18: Generated QR code for student submission document (lecturer)

While regarding the QR code containing the UTeM’s logo is stored in the directory such as “C:\laragon\www\E-Learning\qr-generator\qr-image-with-logo\student_submission” as shown in Figure 6.19.

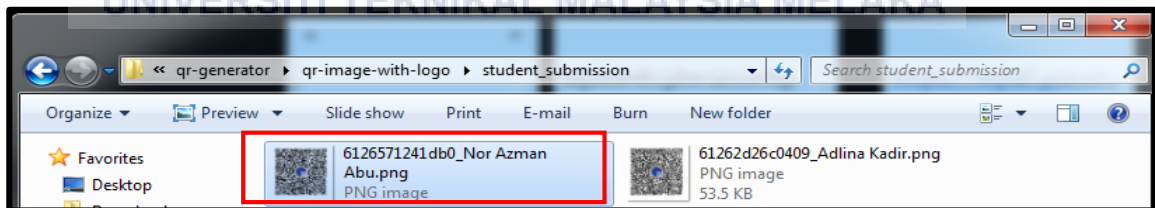


Figure 6. 19: Generated QR code with UTeM’s logo for student submission document (lecturer)

The next testing will be the testing on the steganography image. The details are elaborated in the next section.

6.3 Steganography Image Generator Testing

In this section of testing, the generation of the steganography image will be tested to see if it is successfully generated or not. There are two-part on the steganography image generator testing which are the student part and the lecturer part. Each part has three testings which are on the lecture note document, the lab document, and the student submission document, for a total of six tests. The process of conducting the steganography image generator testing is depicted in Figure 6.20.

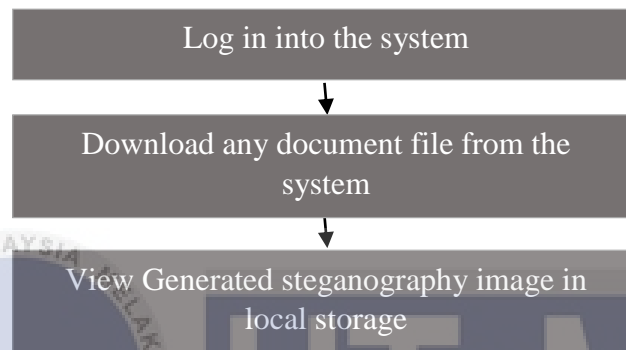


Figure 6. 20: Steganography image generator testing process

Based on Figure 6.20, the steganography image is a UTeM's logo that has been manipulated to hide the information within it. The UTeM's logo to hide the information is shown in Figure 6.21.



Figure 6. 21: UTeM's logo used as a container of hidden information

To begin the steganography image generator testing, logging into the system is required. The steganography image is automatically generated once the user clicked any documents file in the system. This means every time the user (student/lecturer) downloads any document file (lecture note document, lab document, student assignment document), the image as shown in Figure 6.21 will be manipulated by change some of the image properties such as the transparency, and opacity. Then, the information will be injected into that image and as a result, it produced an image with information hidden in it. Figure

6.22 shows the example, student named ‘Adlina Kadir’ downloads a lecture note from the system.

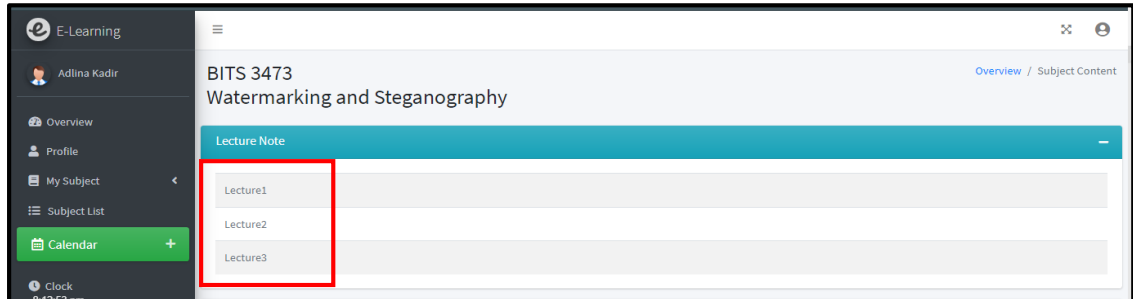


Figure 6. 22: Downloadable lecture note document UI from student's perspective

From Figure 6.22, after downloading the lecture note from the ‘Lecture Note’ tab, the produced steganography image should be stored as in “C:\laragon\www\E-Learning\qr-generator” directory. Figure 6.23 shows the generated steganography image for a student named ‘Adlina Kadir’ is saved in the local storage.



Figure 6. 23: Generated steganography image for lecture note document from student's perspective

The student named ‘Adlina Kadir’ will download the lab document as shown in Figure 6.24 for the second test.

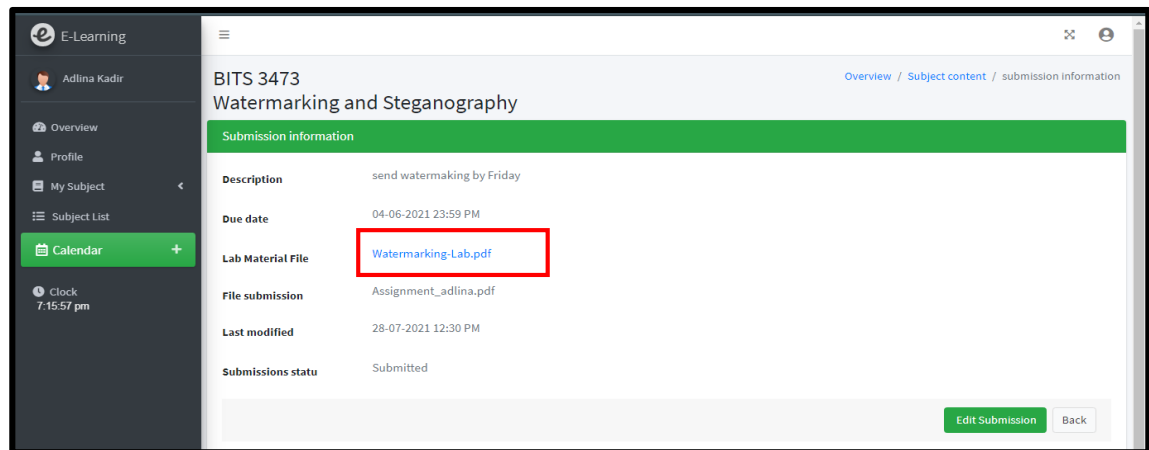


Figure 6. 24: Downloadable lab document UI from student's perspective

Referring to Figure 6.24, the moment of downloading the lab document, a steganography image containing information for that respective student will be generated and will be saved such as in “C:\laragon\www\E-Learning\qr-generator” directory as shown in Figure 6.25.

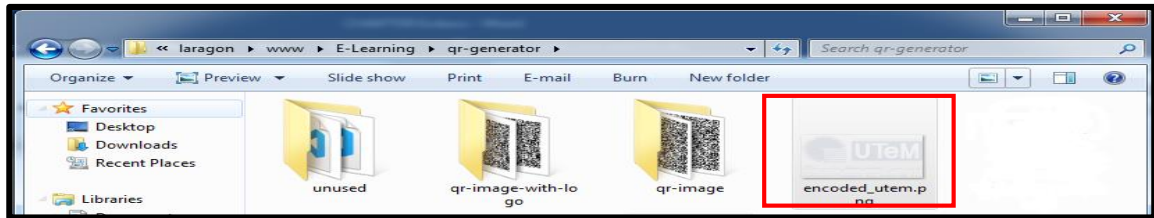


Figure 6. 25: Generated steganography image for lab document from student's perspective

For the third testing, the student’s assignment document will be downloaded by the student itself as shown in Figure 6.26.



Figure 6. 26: Downloadable student's assignment document UI from student's perspective

Based on Figure 6.26, at the time of downloading the student’s assignment document, the steganography image is generated and stored for example in “C:\laragon\www\E-Learning\qr-generator” as shown in Figure 6.27.

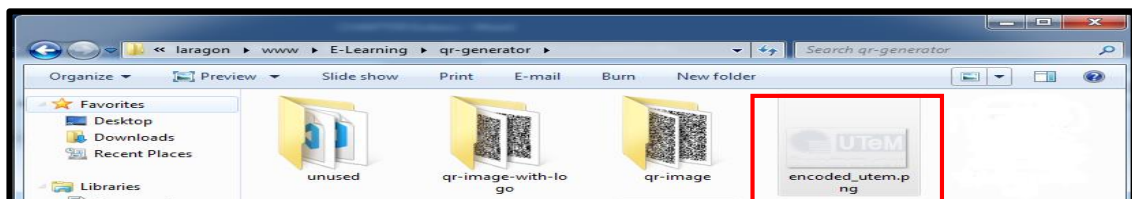


Figure 6. 27: Generated steganography image for student's assignment document from student's perspective

Next, the lecturer will download the lecture note document for the fourth testing as shown in Figure 6.28.

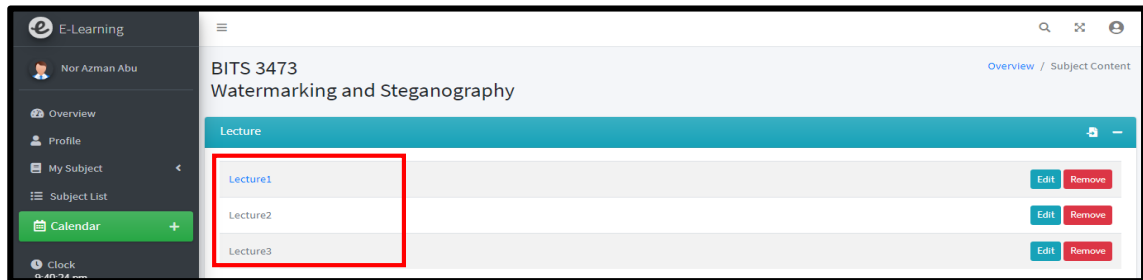


Figure 6. 28: Downloadable lecture note document UI from lecturer’s perspective

Based on Figure 6.28, the moment the lecture note document from the ‘Lecture Note’ tab is downloaded by the lecturer, the generated steganography image is saved for example in “C:\laragon\www\E-Learning\qr-generator”. Figure 6.29 shows the generated steganography image for the lecturer named ‘Nor Azman Abu’ in the local storage.

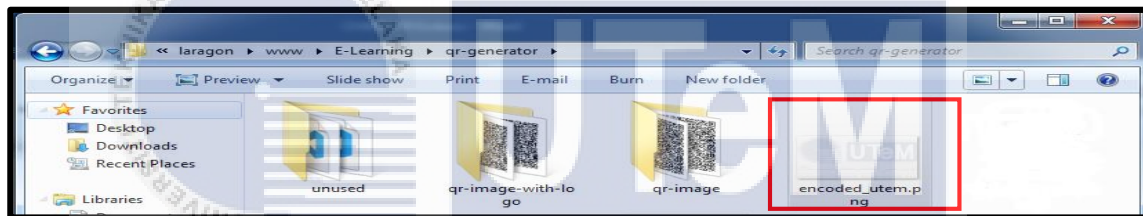


Figure 6. 29: Generated steganography image for lecture note document from lecturer’s perspective

As shown in Figure 6.30, the lecturer will then download the lab document for the fifth testing.

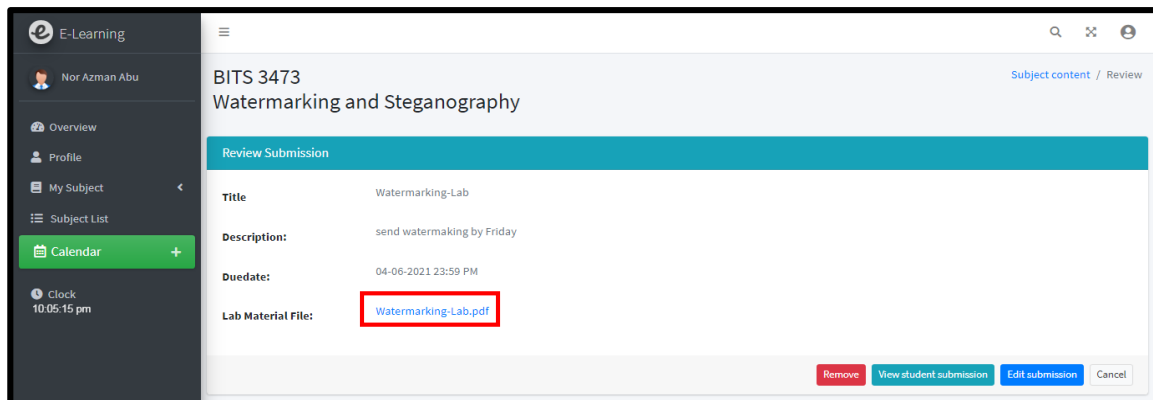


Figure 6. 30: Downloadable lab document UI from lecturer’s perspective

From Figure 6.30, at the time the lab document is downloaded, a steganography image with information for that respective lecturer is created and is saved such as in “C:\laragon\www\E-Learning\qr-generator” directory as shown in Figure 6.31.

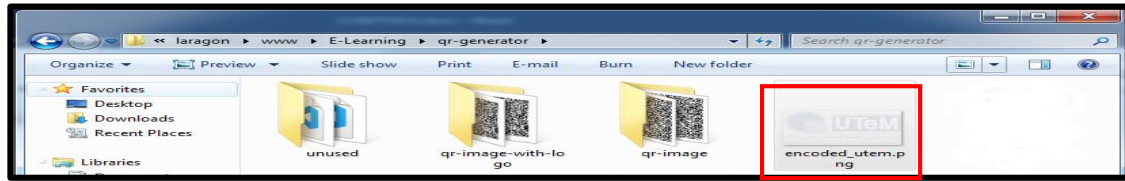


Figure 6. 31: Generated steganography image for lab document from lecturer’s perspective

The last part of the steganography image generator testing is the lecturer will download the document submitted by their student. Based on Figure 6.32, the lecturer named ‘Nor Azman Abu’ is able to see and download the submission document made by the student named ‘Adlina Kadir’.

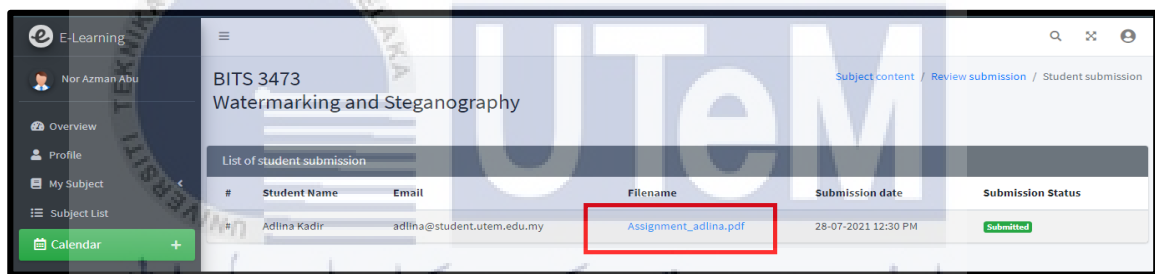


Figure 6. 32: Downloadable student’s assignment document UI from lecturer’s perspective

According to the system as in Figure 6.32, the hidden information will be injected into an image and formed into a steganography image once the lecturer downloads the student’s assignment to protect the ownership document. This information includes the document owner’s name, the source of the file, and the time of the downloading activities. The generated steganography image is saved for example in the directory, “C:\laragon\www\E-Learning\qr-generator” as shown in Figure 6.33.

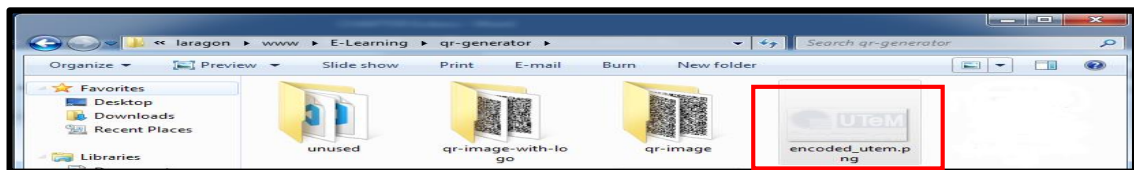


Figure 6. 33: Generated steganography image for student's assignment document from lecturer’s perspective

6.4 Embed Images into Document Testing

This testing section involves the embedding of the three images into the documents, namely QR code image, watermark image, and steganography image. There are two parts: student and lecturer. Each part has three documents to be tested which are lecture note document, lab document, and student assignment document which adds up to six tests. The process of conducting the embedding of the images into the document testing is shown in Figure 6.34.

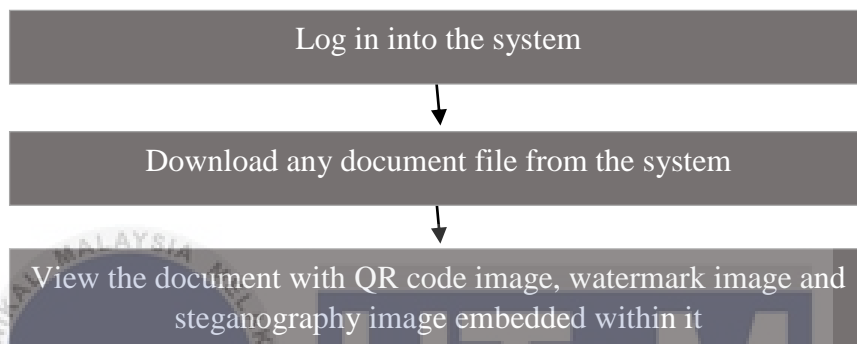


Figure 6. 34: Embed QR code image and steganography image into document testing process

Based on Figure 6.34, logging into the system is required to begin the image embedding into the document testing. The first testing will be conducted from the perspective of the student is namely ‘Adlina Kadir’, who will download the lecture note document as illustrated in Figure 6.35.

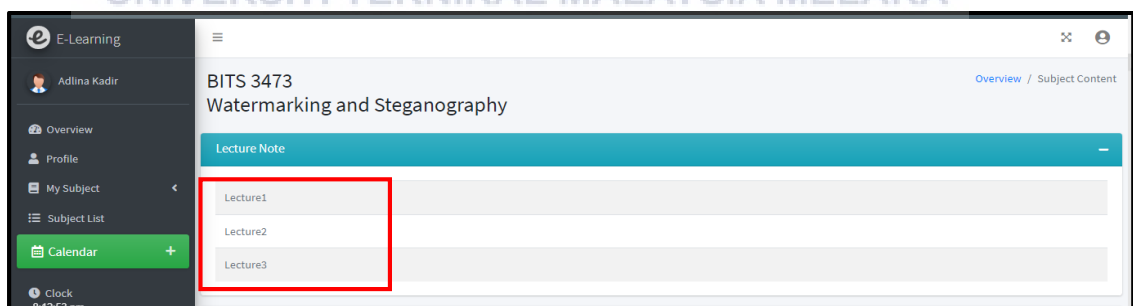


Figure 6. 35: The user interface for a lecture note document that can be downloaded from the student's viewpoint

Based on Figure 6.35, after the document has been downloaded, the lecture note document should be displayed on a new tab of the student’s browser and the lecture note document displayed should contain the three images stated before as portrayed in Figure 6.36.

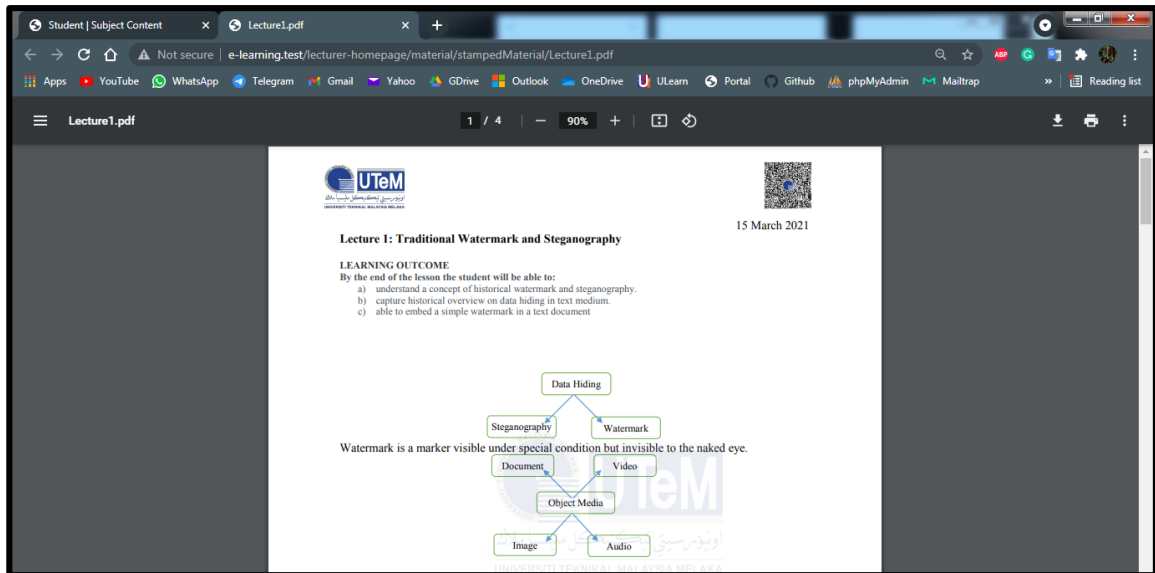


Figure 6. 36: Lecture note document with QR code image, watermark image, and steganography image embedded within it from the student's viewpoint

The second testing is the student will download the lab document from the user interface as shown in Figure 6.37.

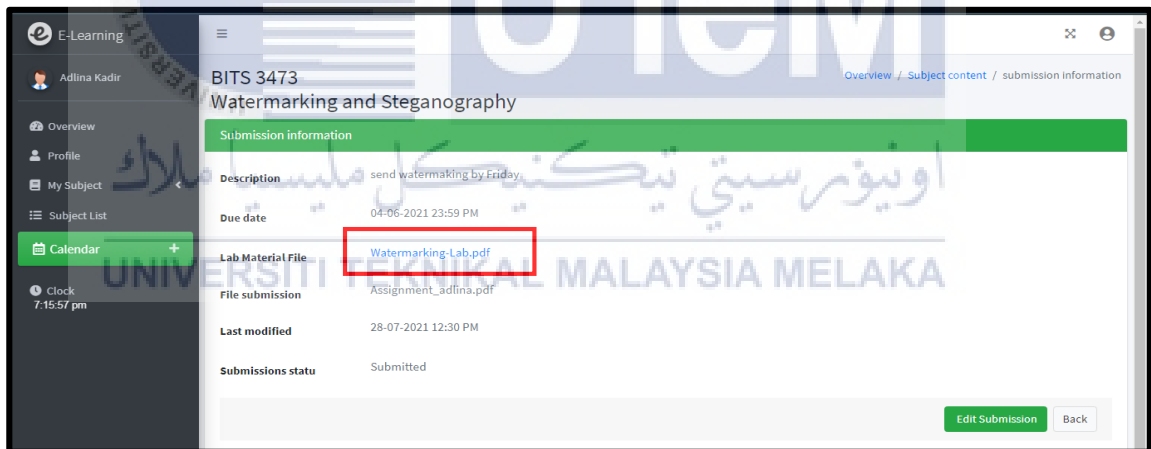


Figure 6. 37: The user interface for a lab document that can be downloaded from the student's viewpoint

Based on Figure 6.37, the lab document should be presented in a new tab in the student's browser after the lab material document has been downloaded. The lab document displayed should contain the watermark image (top left), QR code image (top right), and steganography image (center) as shown in Figure 6.38.

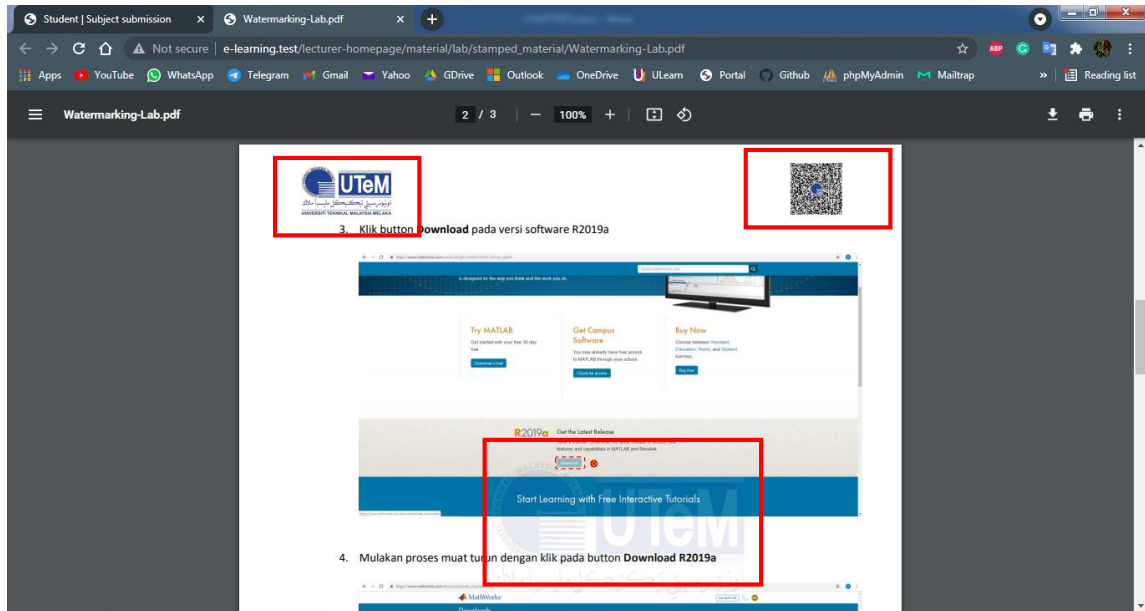


Figure 6. 38: Lab document with QR code image, watermark image, and steganography image embedded within it from the student's viewpoint

In the third testing, the student will download their submission document through the user interface as shown in Figure 6.39.

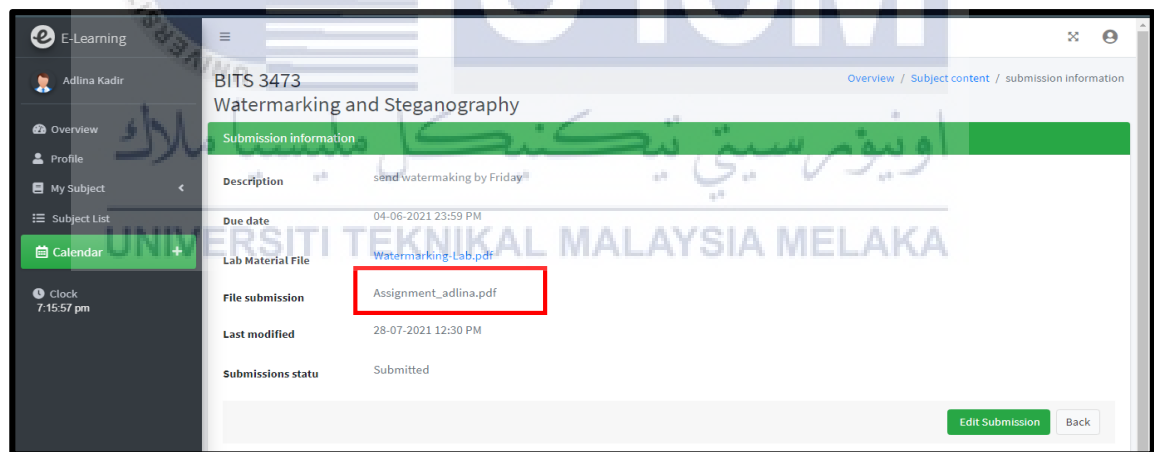


Figure 6. 39: The user interface for student's submission document that can be downloaded from the student's viewpoint

From Figure 6.39, after the submission document has been downloaded, it should appear in a new tab in the student's browser. There are three images that should be visible in the presented submission document which are watermark image (top left), QR code image (top right), and steganography image (center) as shown in Figure 6.40.



Figure 6. 40: Student's submission document with QR code image, watermark image, and steganography image embedded within it from the student's viewpoint

The fourth testing is for the lecturer side. Figure 6.41 illustrated the user interface on where the lecturer can get the lecture note document to download.

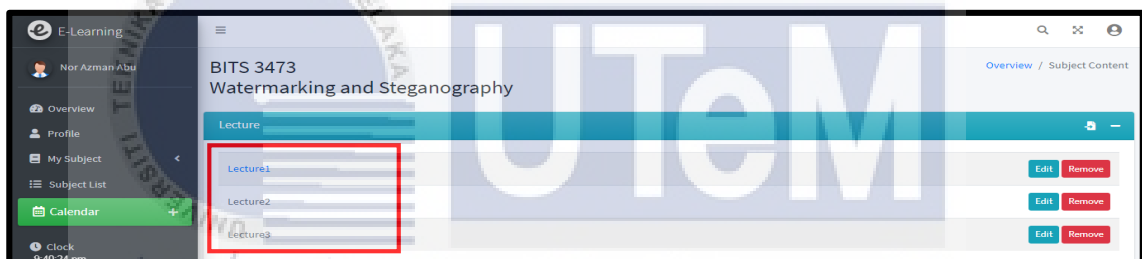


Figure 6. 41: The user interface for a lecture note document that can be downloaded

In the embedding process, three images will be embeds in the document by positioning the image on the top left, on the top right, and the center of the file once the lecturer downloads the notes as shown in Figure 6.42.

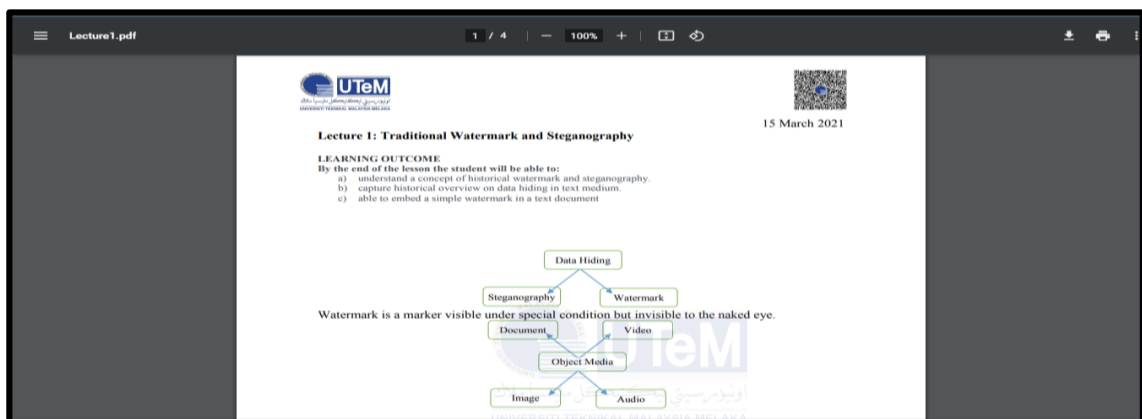


Figure 6. 42: Lecture note document with QR code image, watermark image, and steganography image embedded within it from the Lecturer's viewpoint

The lecturer will then download the lab document for the fifth testing, as indicated in Figure 6.43.

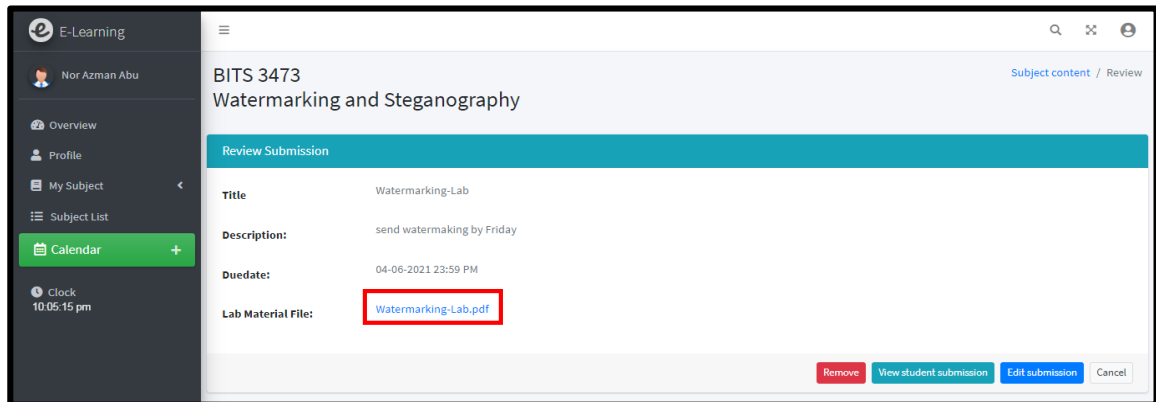


Figure 6. 43: The user interface for a lab document that can be downloaded from the lecturer’s viewpoint

When the lab document in Figure 6.43 is downloaded, a watermark image, a QR code image, and a steganography image with information hidden in it will be embedded into the lab document with the position on the top left, on the top right, and the center as shown in Figure 6.44.



Figure 6. 44: Lab document with QR code image, watermark image and steganography image embedded within it from the lecturer’s viewpoint

The final part in the embed image into document testing is the lecturer will download the assignment document that their student has submitted. As illustrated in Figure 6.45, the lecturer ‘Nor Azman Abu’ can view and download the submission document made by the student ‘Adlina Kadir’.

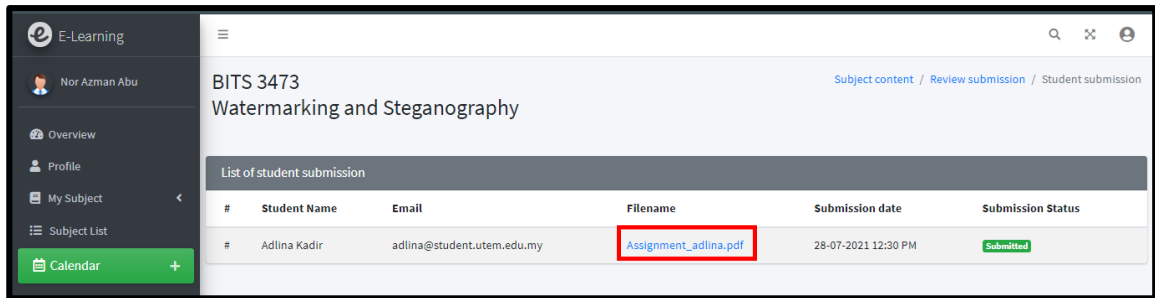


Figure 6. 45: The user interface for a student’s assignment document that can be downloaded from the lecturer’s viewpoint

Based on Figure 6.45, the downloaded student’s submission document will be displayed in the lecturer’s browser once clicked. Three images are embeds in the file by positioning the image on the top left, top right, and center of the file as shown in Figure 6.46.

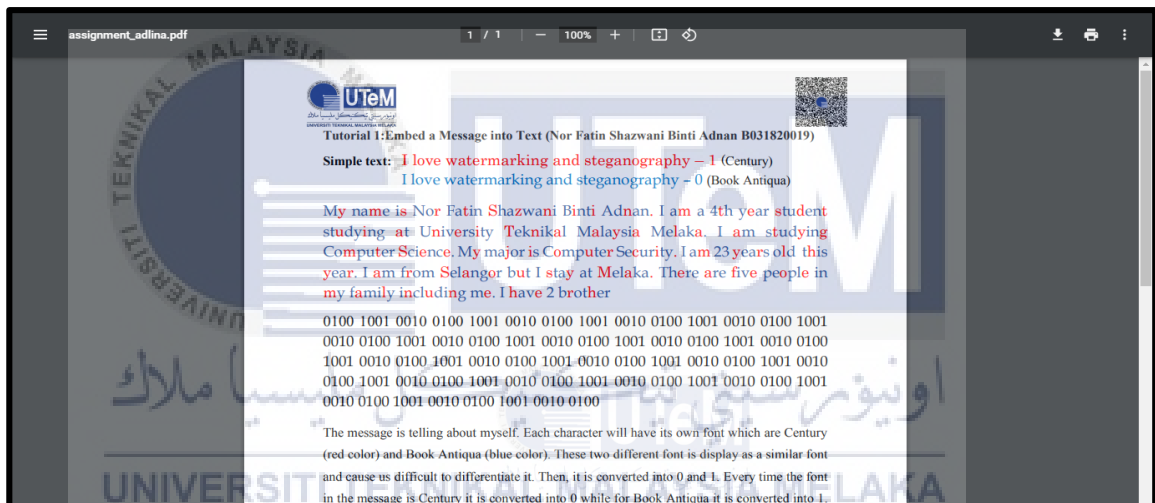


Figure 6. 46: Student’s assignment document with QR code image, watermark image, and steganography image embedded within it from the lecturer’s viewpoint

The next testing is on the QR code image and will be elaborated in the next section.

6.5 QR Code Image Testing

In this testing, the QR code image will be tested. There are two parameters involved in this testing which are the completeness of the data and the usability. The purpose of the data completeness testing is to determine whether the information in the QR code is complete and accurate when compared to information from the system and database. While the goal of the usability testing is to determine whether the QR code is readable.

6.5.1 Completeness of data

For the first testing on the QR code image testing, the first parameter is the completeness of data. This testing will check whether the information stated in the QR code is accurate or not with the information from the system and database. The user who will be the tester from the system in this testing is 'Adlina Kadir', and the PDF document that will be used is the student's submission document named 'assignment_adlina.pdf'. The QR code in that document should have information such as user id, download date, downloaded by, matrix number, email, date of a submission made, file owner, and source file as shown in Table 6.1.

Table 6. 1: Properties for data completeness

Properties	Data
User ID	14
Download date	26-08-2021 [download time]
Downloaded by	Adlina Kadir
Matrix No	B031820014
Email	adlina@student.utem.edu.my
Submission made	26-08-2021 10:51 AM
File owner	Adlina Kadir
Source File.	e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf

According to the database, 'Adlina Kadir' id is '14', matrix number is 'B031820014', email is 'adlina@utem.student.utem.edu.my' as shown in Figure 6.47.

user_id	role_id	matrix_no	user_email	user_password	user_name	user_gender	user_contact_no	user_status
1	1	S001	azman@utem.edu.my	e99a18c428cb38d5f260853678922e03	Nor Azman Abu	male	01132094738	Approved
2	1	S002	nuzulha@utem.edu.my	e99a18c428cb38d5f260853678922e03	Nuzulha Khiwani Ibrahim	female	01132094738	Approved
3	1	S003	norratna@utem.edu.my	e99a18c428cb38d5f260853678922e03	Nor Ratna Masrom	female	01132094738	Approved
4	1	S004	siti@utem.edu.my	e99a18c428cb38d5f260853678922e03	Siti Rahayu Selamat	female	01132094738	Approved
8	1	S008	azirah@utem.edu.my	e99a18c428cb38d5f260853678922e03	Siti Azirah Asmai	female	0127765433	Approved
9	3	A001	admin@gmail.com	e99a18c428cb38d5f260853678922e03	Admin	female		Approved
14	2	B031820014	adlina@student.utem.edu.my	e99a18c428cb38d5f260853678922e03	Adlina Kadir	female	01132094738	Approved

Figure 6. 47: Information of 'Adlina Kadir' from Database

Information for the submission date is '26-08-2021 10:51 AM' as shown in Figure 6.48.

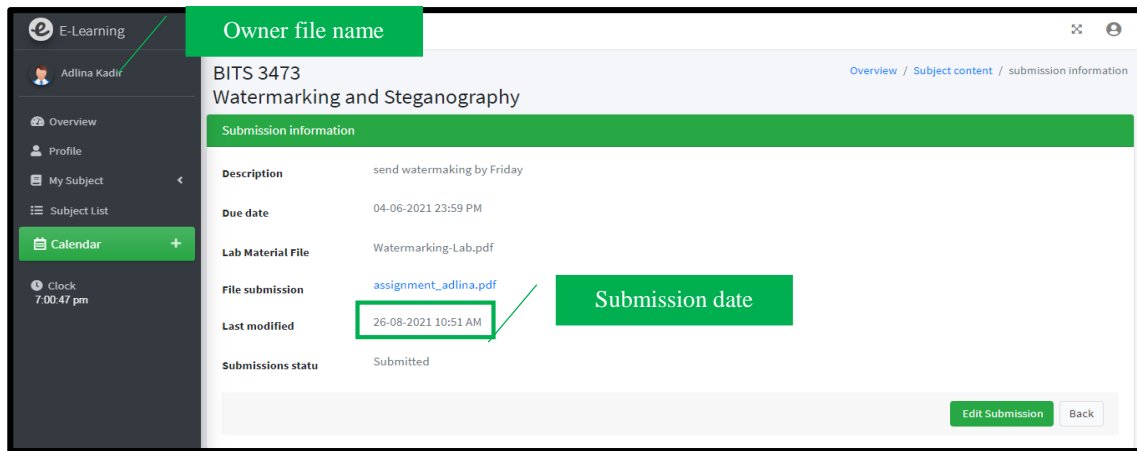


Figure 6. 48: Information of owner file and submission date from System

Information for the source file is ‘e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf’ as shown in Figure 6.49.

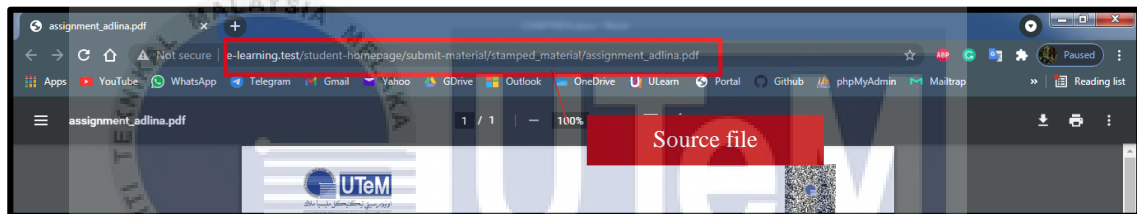


Figure 6. 49: Information of source file

The document tested by the student named ‘Adlina Kadir’ is the student’s submission document itself named ‘assignment_adlina.pdf’ as shown in Figure 6.50.



Figure 6. 50: Student submission document named 'assignment_adlina.pdf'

After the QR code as in Figure 6.50 is scanned, the result is shown as in Figure 6.51. The information stated in the result should be the same as the information from the system and database.

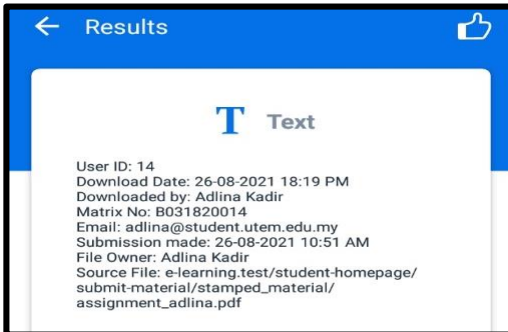


Figure 6. 51: Result after scan the QR code

After the information from the QR scanner in Figure 6.51 has been compared with the information from the system and database, the analysis has been made. The analysis for the data completeness on the QR code image is shown in Table 6.2.

Table 6. 2: Data completeness of QR code analysis


Properties	Data comparison	Advantages	Disadvantages
Accurate	✓	The data is accurate	The data shows after scanning might be too much
Complete	✓	The data is complete	

Based on the analysis in Table 6.2, it can be concluded that the information stated in the QR code is accurate and complete but the ownership information and the downloading activities information from the QR code might be too much. The next section will be the testing on the usability parameter.

6.5.2 Usability

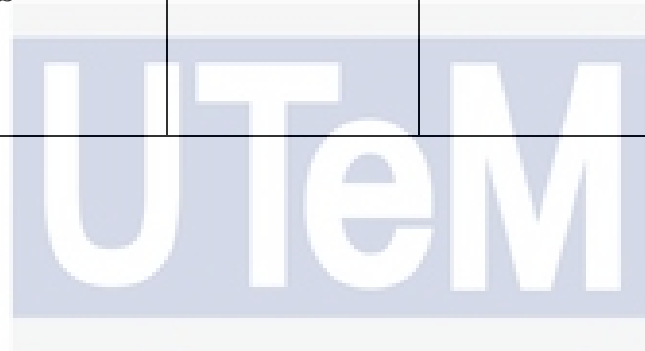
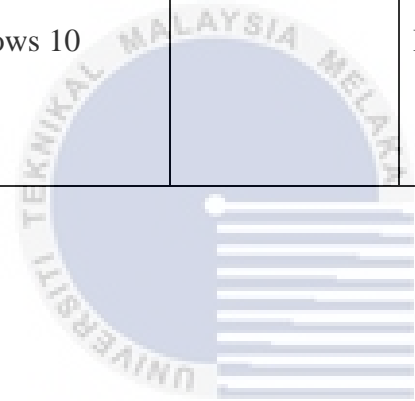
This testing is referred to evaluating the usability of the QR code in the document from the user's perspective. The tester that will test the QR code is the student as they are the easiest to contact. The properties collected in this testing includes the tester name, the device to open the PDF document, the application to open the PDF document, QR code scanner device, QR code scanner application, percentage of the PDF document where the QR code is readable, and the output after scan as shown in Table 6.3.

Table 6. 3: User Usability Testing

Tester	Device to open PDF Document	Application to open PDF Document	QR code Scanner Device	QR Code Scanner Application	PDF Zoom Percentage	Output
Muhammad Faizal Bin Adnan	Laptop: HP Notebook CQ42 OS type: Windows 7	Google Chrome Browser	Smartphone: Oppo A5S OS type: Android	QR & Barcode Scanner Application	200 %	User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/ submit-material/stamped_material/ assignment_adlina.pdf
Muhammad Faizal Bin Adnan	Laptop: HP Notebook CQ42 OS type: Windows 7	Adobe PDF	Smartphone: Oppo A5S OS type: Android	QR & Barcode Scanner Application	200 %	User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/ submit-material/stamped_material/ assignment_adlina.pdf
Muhammad Hafiz Bin Jamil	Laptop: HP Pavillion Gaming OS type: Windows 11	Adobe PDF	Smartphone: Vivo V5 OS type: Android	Online scanner: https://www.the-qrcode-generator.com/scan	400%	 the-qrcode-generator.com/scan Content of QR Code SCAN AGAIN Time: 10:51 AM User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf

Tester	Device to open PDF Document	Application to open PDF Document	QR code Scanner Device	QR Code Scanner Application	PDF Zoom Percentage	Output
Siti Nurbatrisyia Binti Jalawi	Laptop: Lenovo ideapad 320 OS type: Windows 10	Microsoft Edge Browser	Smartphone: Redmi 9t OS type: Android	Built-in MIUI Scanner	175%	<p>QR code details:</p> <p>User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf</p>
Farzana Binti Ariffin	Laptop: HP Pavillion 15 OS type: Windows 10 Pro	Google Chrome Browser	Smartphone: Samsung Galaxy A72 OS type: Android	Built-in Samsung Browser QR Scanner	175%	<p>< Scan result</p> <p>User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf</p>
Farzana Binti Ariffin	Laptop: HP Pavillion 15 OS type: Windows 10 Pro	Adobe PDF	Smartphone: Samsung Galaxy A72 OS type: Android	Built-in Samsung Browser QR Scanner	200%	<p>< Scan result</p> <p>User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf</p>

Tester	Device to open PDF Document	Application to open PDF Document	QR code Scanner Device	QR Code Scanner Application	PDF Zoom Percentage	Output
Amirah Nadhirah Binti Kamarulzaman	Laptop: HP Pavillion 14 OS type: Windows 10	Adobe PDF	Smartphone: iPhone 8 OS type: IOS	Built-in iPhone 8 QR Scanner	125%	<pre>User ID: 14 Download Date: 26-08-2021 18:19 PM Downloaded by: Adlina Kadir Matrix No: B031820014 Email: adlina@student.utm.edu.my Submission made: 26-08-2021 10:51 AM File Owner: Adlina Kadir Source File: e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf</pre>



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

According to the properties stated in Table 6.3, ‘Device to open the PDF document’ is the tester’s device to open the PDF file, ‘Application to open the PDF’ is the tester’s application to open the PDF file, ‘QR code scanner device’ is the tester’s device used as a QR code scanner, ‘QR code scanner application’ is the tester’s QR code scanner application used in their device, ‘PDF zoom percentage’ is the percentage of the PDF document to zoom in to make the QR code readable and ‘Output’ is the result of the scanning. Table 6.4 shows the analysis for this testing. The properties ‘Application to open PDF document’ and ‘QR code scanner application’ is chosen as it seems to give more impact on the user usability testing.

Table 6. 4: Analysis of user usability testing on QR code

Application to open PDF Document	QR Code Scanner application	Advantages	Disadvantages
Browser	Smartphone Built-in scanner.	<ul style="list-style-type: none"> • Less percentage to zoom in on the document. • Readable QR code. 	<ul style="list-style-type: none"> • The document file still needs to be zoom in.
Browser	Open source scanner.	<ul style="list-style-type: none"> • Readable QR code. 	<ul style="list-style-type: none"> • More percentage to zoom in on the document.
Adobe PDF	Smartphone Built-in scanner.	<ul style="list-style-type: none"> • Readable QR code. 	<ul style="list-style-type: none"> • More percentage to zoom in on the document.
Adobe PDF	Open source scanner.	<ul style="list-style-type: none"> • Readable QR code. 	<ul style="list-style-type: none"> • More percentage to zoom in on the document.

From the analysis in Table 6.4, it can be concluded that the QR code is can be read but the distance to scan the QR code inside the file depends on the application to open the PDF document and the QR scanner application used. Based on the analysis, the QR code inside the file is the best use with open the PDF file with a browser and scan the QR code using a smartphone built-in scanner as it resulted in less percentage to zoom in the pdf file. The next testing will be on the steganography image.

6.6 Steganography image Testing

In this testing, the generated steganography image will be tested. There are two parameters involved in this testing which are the completeness of the data and the imperceptibility. The data completeness aims to ensure that the data after the QR code scanning is complete and the information is accurate while the aim for the imperceptibility is to measure the visibility of the hidden message from the container image.

6.6.1 Completeness of data

For this parameter, the steganography image will be tested by uploading into the decoder of the image in the system to see whether the information injected inside the image is the same as the information from the QR code or not. Figure 6.52 shows the user interface of the message extraction.



Figure 6. 52: User interface for message extraction

For this testing, the document used is still the student's submission document namely, 'assignment_adlina.pdf' but from the lecturer's point of view. Table 6.5 shows the information should have in the steganography image after the user lecturer downloads the student's submission file.

Table 6. 5: Data completeness properties

Properties	Data
User ID	1
Download date	26-08-2021 [download time]
Downloaded by	Nor Azman Abu
Matrix No	S001

Properties	Data
Email	azman@utem.edu.my
Submission made	26-08-2021 10:51 AM
File owner	Adlina Kadir
Source File.	e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf

As shown in Figure 6.53, the lecturer named ‘Nor Azman Abu’, user id is ‘1’, matrix number is ‘S001’, and email is ‘azman@utem.edu.my’.

user_id	role_id	matrix_no	user_email	user_password	user_name	user_gender	user_contact_no	user_status
1	1	S001	azman@utem.edu.my	e99a18c428cb38d5f260853678922e03	Nor Azman Abu	male	01132094738	Approved
2		S002	nuzulha@utem.edu.my	e99a18c428cb38d5f260853678922e03	Nuzulha Khatuni	female	01132094738	Approved
3			ratna@utem.edu.my	e99a18c428cb38d5f260853678922e03	Nor Ratna Masrom	female	01132094738	Approved
4	1	S004	sitirahayu@utem.edu.my	e99a18c428cb38d5f260853678922e03	Siti Rahayu Selamat	female	01132094738	Approved

Figure 6. 53: Information of lecturer 'Nor Azman Abu' from Database

From Figure 6.54, the assignment document named ‘assignment_adlina.pdf’ is from ‘Adlina Kadir’ and submission date is made on ‘26-08-2021 10:51 AM’.

Student Name	Email	Filename	Submission date	Submission Status
Adlina Kadir	adlina@student.utem.edu.my	assignment_adlina.pdf	26-08-2021 10:51 AM	Submitted

Figure 6. 54: File owner and submission date information from System

As indicated in Figure 6.55, the information for the source file from Figure 6.54 is ‘e-learning.test/student-homepage/submit-material/stamped_material/assignment_adlina.pdf’.

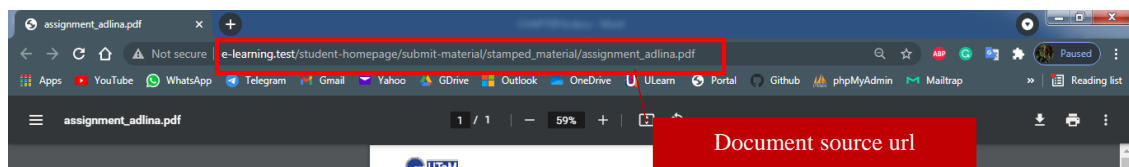


Figure 6. 55: The file source information

Figure 6.56 shows the student’s submission document named ‘assignment_adlina.pdf’ is downloaded by the lecturer, ‘Nor azman Abu’.

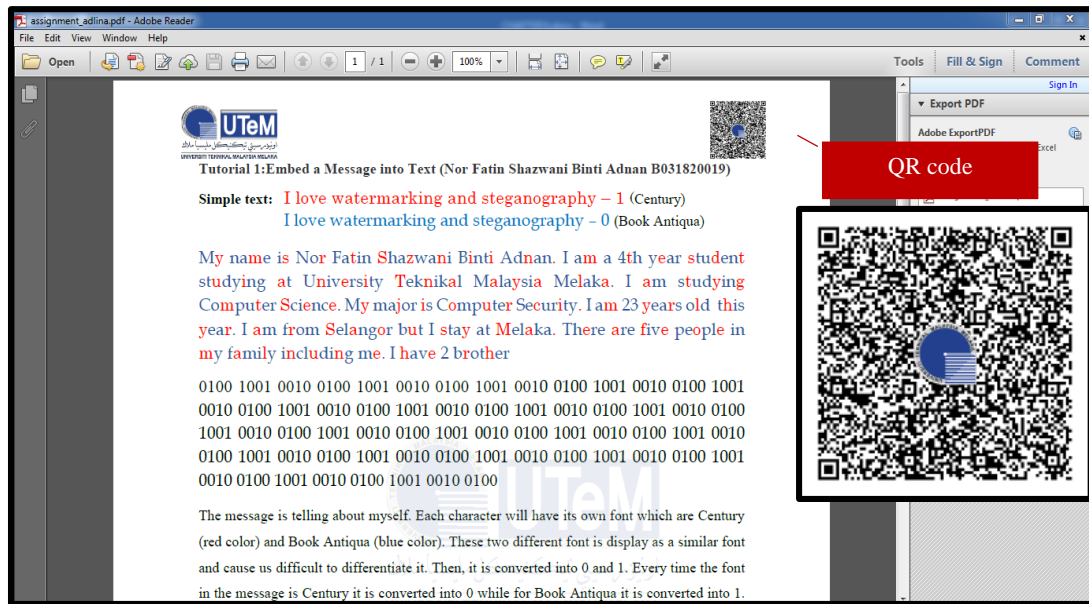


Figure 6. 56: The student's submission document named 'assignment_adlina.pdf'

The outcome of scanning the QR code is illustrated in Figure 6.57. The information from the result should correspond to the information in the system and database.

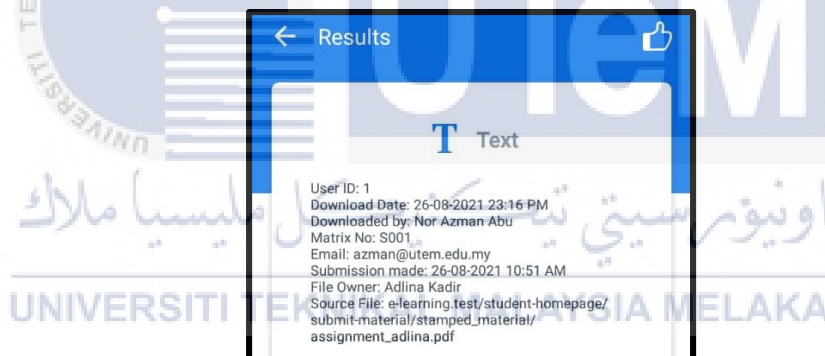


Figure 6. 57: Outcome of the QR code scanning

To extract the information from the steganography image as shown in Figure 6.58, the image must be uploaded into the system on the image decoder part as in Figure 6.52.



Figure 6. 58: Steganography Image

The steganography image in Figure 6.58 will be uploaded into the image decoder and the outcome after the extraction should be the same as the information from the QR code image in the student's submission document, namely 'assignment_adlina.pdf' because these two images are sitting inside the same document. The message extraction is shown in Figure 6.59.

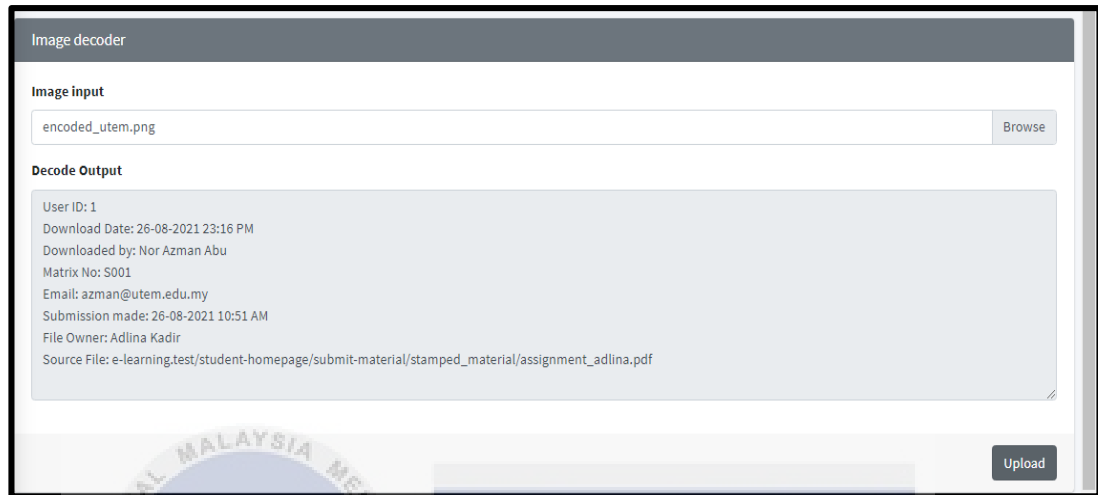


Figure 6. 59: Extraction of message in steganography image

After the extracted information in Figure 6.59 has been compared with the information from the QR scanner, an analysis is made. The analysis for the data completeness on the steganography image is shown in Table 6.6.

Table 6. 6: Data completeness of steganography image analysis

Properties	Data comparison	Advantages	Disadvantages
Accurate	✓	The data is accurate	The data provided is excessive
Complete	✓	The data is complete	

Based on Table 6.6, it can be concluded that the information in the steganography image after the extraction process is complete and precise as the information in the QR code image but the drawback is the data provided might excessive which might reveal too much.

6.6.2 Imperceptibility

Imperceptibility is the property that can measure the visibility of the hidden information in the container and determine whether it can be perceived by the human mind and sense or not. Setiaadi (2020) stated that if the image alteration is extreme,

the human sense of vision will be able to recognize the changes. This means the container of the hidden information should not be manipulated too much and the imperceptibility should be considered for a steganography image. For this parameter, the steganography image is measured by taking a look at the document roughly and evaluate whether the message in the document can be detected or not. The testing for this parameter that can be done is called visibility testing. As shown in Figure 6.60, the document is zoom in for visibility testing.

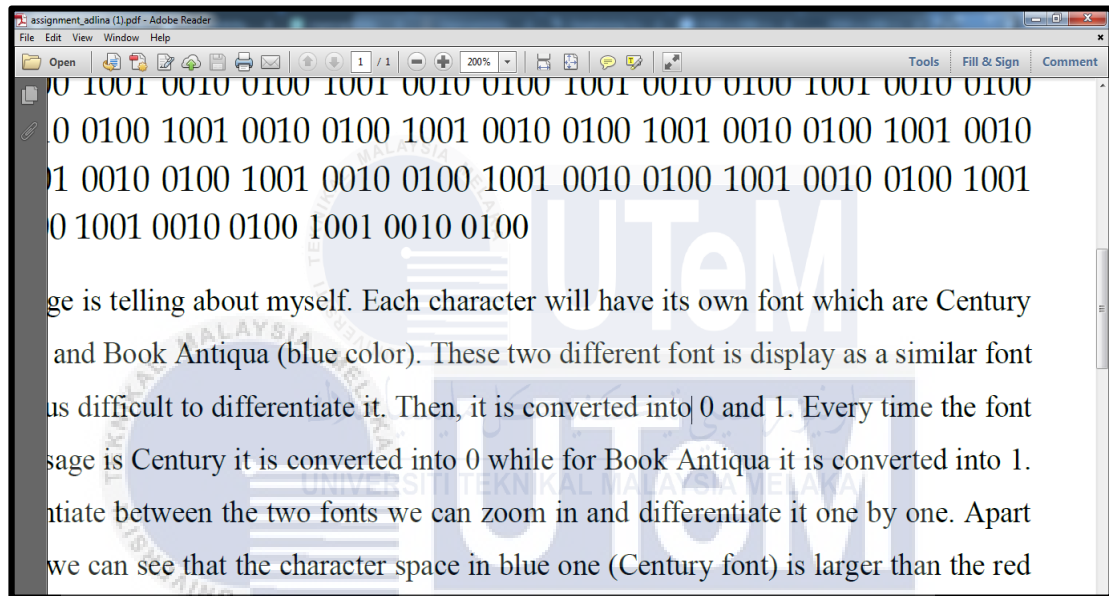


Figure 6. 60: Zoom in document

Based on Figure 6.60, the steganography image can only be seen as a watermark image but not seen as an image that contains a hidden message in it. The analysis for this testing is illustrated in Table 6.7.

Table 6. 7: Visibility Testing Analysis

Properties	Visibility	Advantages	Disadvantages
Visibility of the steganography image	The message cannot be seen but the carrier can be seen	<ul style="list-style-type: none"> Contains ownership information Seen as a watermark image instead of a steganography image 	<ul style="list-style-type: none"> May distract the content of the document. Visible and can cause the viewer to focus on the image instead of the content.

Generally, this evaluation can be concluded that the hidden message inside the steganography image is undetectable by the human naked eye but the carrier is still can be seen as it may result in the steganography image be removed. For further analysis, using a tool to evaluate the visibility of the hidden message can be made.

6.7 Summary

This chapter has explained the testing that has been conducted for this project. The QR code generator testing is conducted to see if the QR code is successfully generated or not, the steganography image generation testing is to see whether the steganography image is successfully created or not, embed the images into document testing is to check whether the QR code image watermark image, and the steganography image is embedded in the document or not, QR code image testing is to evaluate whether the QR code image is readable and contains a 'should have' information or not, and lastly the steganography image testing is to determine the hidden information in the steganography image can be detected or not. Generally, all the testing is conducted to ensure all of the main modules in this project is work as planned. The conclusion of the project will be described in the next chapter, which is Chapter 7.

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

The project's testing was completed in the previous chapter to evaluate the effectiveness of the proposed project and achieved all project objectives. In this chapter the project summarization, project contribution, project limitation, and future works are elaborated. Project summarization will describe the summary of the project, project contribution will describe the researcher's contribution and project limitation will explain the limitation of the project. The suggestions to improve the project will be discussed according to the project limitation found from the project in the future works section.

7.2 Project summarization

There are three objectives in this project which are to analyze copyright protection techniques for documents (PO1), formulate hybrid copyright protection techniques (PO2) and evaluate the effectiveness of the proposed protection techniques (PO3). For PO1, it is found that three copyright protection techniques can be used to protect the document in the system, which are watermarking, steganography, and cryptography. PO1 has been achieved as presented in Chapter 2 that shows the suitable technique for this project is watermarking and steganography. For PO2, the protection technique to be formulated is according to the suitable protection technique that has been found in PO1. There are two techniques used in this project which are watermarking and steganography. For the watermarking protection technique, two images will be used as a watermark. Firstly, the QR code image which contains the ownership information, and secondly, the UTeM's logo image which does not contain anything, it is just a UTeM's logo image. For the steganography protection technique, one image will be a steganography image which is also the UTeM's logo. The UTeM's logo will be the image carrier that will hide the ownership information in it. The

steganography image will contain information the same as in the QR code image. For PO3, the effectiveness of the proposed technique will be evaluated according to the parameter as described in Chapter 6. In this project, a prototype of E-Learning System is developed in order to evaluate the effectiveness of the proposed methods to protect the copyright of the document. Three images will be embedded in a document whenever a document is downloaded by the user. The three images are, UTeM's logo as a watermark, QR code image which contains ownership information, and steganography image which also contains hidden ownership information.

7.3 Project contribution

A document protection technique is important as it can be a step of the protection layer. The first project contribution is a technique to protect digital documents from illegal and misuse activities. There are many protection techniques found and can be used to protect the document. Among the protections are watermarking, steganography, and cryptography. The first contribution can give a sight on what protection technique for a document can be applied. The second contribution is a protected document that embedded with a QR code image, watermark image, and steganography image. In this project, the three images states are embedded with the document each time the document is downloaded. This contribution can protect the document as the three images are embedded. The last contribution is the identity of the document's ownership. The QR code image as stated before contains the ownership information same as the steganography image. Because of that, whenever the document is distributed to other platforms or misuse, the identity of the documents can be known by scan the QR code and extract the message from the steganography image. The UTeM's logo also can give a sight that the document is from UTeM.

7.4 Project limitation

There are several limitations of this project which are:

1. Author of the document: The best for the protection for the ownership of the document should be the author of the document. However, in this project, the author of the document is defined as the person who upload the document into the system.

2. Watermarking image visibility: The image for watermark can be seen by users.
3. Steganography image visibility: The image for steganography can distract the content for the document.
4. Data hidden identification; The hidden information from the steganography image can only identify at the developer side.
5. Document type: The best document type that can be uploaded to the system should be all type of digital document. However, in this project, the document type can be uploaded to the system is only a PDF type.

Although there are limitation, the proposed project however able to provide an effective method on protecting the copyright and ownership of the document.

7.5 Future works

Based on the limitations stated in Section 7.4, the improvements that can be made to overcome the limitations are:

1. The system can be improved by using the 'Author' name as the owner of the file as it is more accurate to know about who is the real owner of the document.
2. An invisible watermark can be made and applied as it cannot be seen by the naked eye.
3. An invisible steganography image can be created since it cannot be seen with the human eye.
4. A mechanism to obtain the steganography image from the document after the document is downloaded can be made to prove the ownership information inside the steganography image from the user's perspective.
5. Add some other document extension type

Therefore, to make the system to be better, it is important to make some changes, which should be continuous. The better the improvement can be made, the more limits can be overcome.

7.6 Summary

Despite everything, the project has been completed and the objectives were met. There are limitations in this project, however it can be improved in the future. This project can be enhanced for better use and better document protection for the E-Learning system. As technology is evolving around the world, the protection of digital

documents also needs to be evolved. It is very advisable to have at least a layer of document protection as the benefits that will be gained from the protection are crucial to prevent the document from illegal or misuse activities.



REFERENCES

- Abdul Rahman, Muhamad & Hassan, Mohd & Sabuddin, Siti. (2020). COVID-19: Kecenderungan Meneruskan Penggunaan Platform Pembelajaran Atas Talian dalam Kalangan Guru Pra Perkhidmatan Semasa Perintah Kawalan Pergerakan. Conference: International Conference On Educational Research (InCER 2020). 916-933.
- Nadzirah Mat Sin, n.d, Online Learning for Future Education, Academia, accessed 18 April 2021, <
https://www.academia.edu/16675692/Online_Learning_for_Future_Education
 on >
- Society of American Archivists 2020, accessed 27 March 2021, <
<https://dictionary.archivists.org/entry/digital-document.html>>
- Zhou, G. and Yang, L. (2010). Application of BHO-based PDF Documents Copyright Protection. 2010 International Conference on Management and Service Science. IEEE Access. 1-4.
- Kim, S., Lee, S., Oh, T., Choi, N., Ryu, J.D., & Kang, H. (2014). Copyright protection and distribution system for scanned books/comics. The International Conference on Information Networking 2014 (ICOIN2014). 352-355.
- Patel, S.K.J and Tahilramani, N.V. (2016). Information Hiding Techniques: Watermarking, Steganography: A Review. International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering. 4. 168-173.
- Tiwari, N. And Sharmila (2017). Digital Watermarking Applications, Parameter Measures and Techniques. IJCSNS International Journal of Computer Science and Network Security. 17. 184-193.

- Kadhim, J.J., Premaratne, P., Vial, P.J. and Halloran, B. (2018). Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. *Neurocomputing*. 335. 299-326.
- Subramanian, N., Elharrouss, O., Al-Maadeed, S. and Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. *IEEE Access*. 9. 23409-23423.
- Amarendra, K., Mandhala, V.N., Gupta, B.C., Sudheshna, G.G, Anusha, V.V. (2019). Image Steganography Using LSB. *International Journal Of Scientific & Technology Research*. 8. 906-909.
- Kumar, K.S., Kumar, Ch.M., Kumar, B.S. and Cristin, R. (2021). Highly imperceptible data hiding technique using MSB in the grayscale image. *Materials Today: Proceedings*.
- Thampi, S.M. (2014). Information Hiding Techniques: A Tutorial Review. *Cryptography and Security (cs.CR)*.
- Balgurgi, P.P, and Jagtap, S.K. (2013). Audio Steganography Used for Secure Data Transmission. *Proceedings of International Conference on Advance in Computing*. *Advances In Intelligent Systems and Computing*. 174. 699-706.
- Dutta, H., Das, R.K., Nandi, S. and Prasanna, S.R.M. (2019). An Overview of Digital Audio Steganography. *IETE Technical Review*. 37. 632-650.
- Banik. B.G. and Bandyopadhyay, S.K. (2018). Blind Key Based Attack Resistant Audio Steganography Using Cocktail Party Effect. *Hindawi Security and Communication Network*. 2018. 1-22.
- Aru, O.E. and Ananaba, C.E. (2018). Detailed Examination of Information Hiding Techniques for Copyright Protection of Text Documents. *IOSR Journal of Applied Chemistry (IOSR-JAC)*. 11. 21-30.

- Mishra, M. Mishra, P. And Adhikari, M.C. (2012). Digital Image Data Hiding Techniques: A Comparative Study. The Journal of F.M. University. 7. 105-115.
- Mantoro, T., Wahyudi, M., Ayu, M.A., & Usino, W. (2015). Real-time Printed Document Authentication Using Watermarked QR Code. Fourth International Conference on Cyber Security, Cyber Warfare, and Digital Forensic. IEEE Access. 68-72.
- Suwito, M.H., Ueshige, Y., Feng, Y., & Sakurai, K. (2017). Integrity Watermarking and QR-Code Techniques for ensuring Printed Document Authenticity Real Time Distribution.
- Espejel-Trujillo A., Castillo-Camacho I., Nakano-Miyatake M., Perez-Meana H. (2012). Identity Document Authentication Based on VSS and QR Codes. The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science. 3. 241-250.
- Alajmi, M., Elashry, I., Hala S.E. and Osama S.F. (2020). Steganography of Encrypted Messages Inside Valid QR Codes. IEEE Access. 8. 27861- 27873.
- Hassanein, M.S. (2014). Secure digital documents using Steganography and QR Code.
- Pal, K. and Kumar, C.R.S. (2021). QR Code Based Smart Document Implementation Using Blockchain and Digital Signature. Advances in Intelligent Systems and Computing. 1174. 449-465.
- Ali, A.M. and Farhan, A.K. (2020). Enhancement Of Qr Code Capacity By Encrypted Lossless Compression Technology For Verification Of Secure E-Document. IEEE Access. 8. 27448- 27458.
- Kaspersky 2021, accessed 28 March 2021, <<https://www.kaspersky.com/resource-center/definitions/what-is-a-qr-code-how-to-scan>>

- Asare, I.T., Asare, D., & Sun, G. (2015). The Effective Use of Quick Response (QR) Code as a Marketing Tool. *International Journal of Education and Social Science*. 2(12), 67-73.
- Hassan, A. and Hussein, A. (2020). Documents Authentication and Verification. *IOP Conference Series: Materials Science and Engineering*, Volume 765, 1st International Conference of Electromechanical Engineering and its Application (ICEMEA-2020). 765. 1-10.
- Manimekalai, M. and Bakkiyalakshmi, R. (2017). Hide and Seek: A New Way to Hide Encrypted Data in QR Code Using the Concepts Steganography and Cryptography. *International Journal of Advanced Research in Computer and Communication Engineering*. 6. 538-540.
- Mendhe, A., Gupta, D.K. and Sharma, K.P. (2018). Secure QR-Code Based Message Sharing System Using Cryptography and Steganography. 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). 2018. 188-191.
- Ashwini, C.M., Dipshikha M.N, Vinay V.K, Kajal S.P. (2021). A Survey On Novel Approach For Data Hiding Under Qr Code Using Visual Secret Sharing. *International Journal Of Advance Scientific Research And Engineering Trends*. 6. 31-34.
- Sawsan, K.T. and Basheer, N.A. (2016). A New Method for Cipherring a Message Using QR Code. *Computer Science and Engineering 2016*. 6. 19-24.
- Dang, Q.B., Louisa, K., Coustaty, M., Luqman, M.M. and Oqier, J. (2019). A Blind Document image watermarking approach based on Discrete Wavelet Transform and QR code embedding. 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). 8. 1-6.

- Ahvanooy, M.T., Li, Q., Shim, H.J. and Huang, Y. (2018). A Comparative Analysis of Information Hiding Techniques for Copyright Protection of Text Documents. *Security and Communication Networks*. 1-22.
- Huang, H.C., Chen, Y.H., Chang, F.C. and Tseng, C.T. (2020). Multi-Purpose Watermarking with QR Code Applications. *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*. 42-45.
- Arkah, Z.M., Alzubaidi, L., Ali, A.A. and Abdulameer, A.T. (2020). Digital Color Documents Authentication Using QR Code Based on Digital Watermarking. *Advances in Intelligent Systems and Computing*. 940. 1094-1100.
- Li, D., Gao, X., Sun, Y., & Cui, L. (2017). Research on Anti-counterfeiting Technology Based on QR Code image Watermarking Algorithm. *12*. 57-66.
- Saraswati, M., Maroti, M., Sainath, M., Prakash, S., & Fadewar, D.H. (2017). QR Code Watermarking Algorithm Based on DWT and Counterlet Transform for Authentication.
- Rhazlane, S., El Ouazzani, A., Harbi, N., Nadia, K. And Hassan, B. (2017). Data Alteration: A Better Approach to Securing Cloud Data with Encryption. *Conference: The 13th Conference EDA: BI & Big Data*.
- Al-Haj, A. and Barouqa, H. (2017). Copyright protection of e-government document images using digital watermarking. *2017 3rd International Conference on Information Management (ICIM)*. 2017. 441-446.
- Kadhim, J.J., Premaratne, P., Vial, P.J. and Halloran, B. (2018). Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research. *Neurocomputing*. 335. 299-326.

- Chavan, S., Gadakh, S., Kanchan, G. Surabhi, S. and Shinkar, D.V. (2016). QR code Authentication System for confidential (digital Mark sheet) Encrypted data hiding and retrieval (Decryption). International Journal of Advanced Research in Computer and Communication Engineering. 5. 88-92.
- Chemana Shaik (2021). Detection Of Forgery And Fabrication In Passports And Visas Using Cryptography And Qr Codes. Advanced Computing: An International Journal (ACIJ). 12. 1-11.
- Wibiyanto, A. And Afrianto, I. (2018). QR code and Transport Layer Security for Licensing Documents Verification. Journal: IOP Conference series material science and engineering.
- Zhang, W., and Meng, X. (2015). An improved digital watermarking technology based on QR code. 2015 4th International Conference on Computer Science and Network Technology (ICCSNT). 1. 1004-1007.
- Mir, N. and Khan, M.A.U. (2020). Copyright Protection for Online Text Information: Using Watermarking and Cryptography. 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS). 2020. 1-4.
- Iqbal, M.M, Khadam, U., Han, K.J., Han, J. and Jabbar, S. (2019). A Robust Digital Watermarking Algorithm for Text Document Copyright Protection based on Feature Coding. 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). 2019. 1940-1945.
- Dey, A.S., Nath, B.J. And Nath, C.A. (2012). A New Technique to Hide Encrypted Data in QR Codes. Proceedings of The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing.
- Arief , A.T., Wirawan, W. and Suprpto, Y.K. (2019). Authentication of Printed Document Using Quick Response (QR) Code. 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA). IEEE Access. 2019. 228-233.

Nayak, J. K. and Singh, P. (2021). Fundamentals Of Research Methodology: Problems And Prospects. Daryaganj, New Delhi, India. SSDN Publishers and Distributors. 1-336.

Phpqrcode.sourceforge.net. 2021, PHP QR Code - QR code generator, an LGPL PHP library, accessed 3 June 2021 <<http://phpqrcode.sourceforge.net/>>

Denso Wave Incorporated 2021, accessed 8 June 2021, <https://www.qrcode.com/en/about/error_correction.html>

Scanova Blog 2021, QR Code Minimum Size: Find the ideal size for your use case, accessed 5 June 2021, <<https://scanova.io/blog/qr-code-minimum-size/>>

Hotexamples.com. 2021, PHP FPDI::Output Examples, accessed 5 June 2021 <<https://hotexamples.com/examples/-/FPDI/Output/php-fpdi-output-method-examples.html>>

Grotta, S. and Grotta, D., 2021, Pixels: Size Matters, IEEE Spectrum: Technology, Engineering, and Science News, accessed 6 June 2021 <<https://spectrum.ieee.org/geek-life/tools-toys/pixels-size-matters>>

Pal, K. and Kumar, C.R.S. (2021). QR Code Based Smart Document Implementation Using Blockchain and Digital Signature. Advances in Intelligent Systems and Computing. 1174. 449-465.

Lotlikar, T., Kankapurkar, R., Parekar, A., & Mohite, A. (2013). Comparative study of Barcode , QR-code and RFID System. International Journal of Computer Technology and Applications. 4. 817-821.

Thonky.com. 2021, QR Code Tutorial, accessed on 21 June 2021, <<https://www.thonky.com/qr-code-tutorial/>>

Setiadi, D.I.M. (2021). Psnr vs SSIM: imperceptibility quality assessment for image steganography. Multimed Tools and Applications. 80. 8423-8444.

Al-Mohammad, A. (2010). Steganography-based secret and reliable communications: Improving steganographic capacity and imperceptibility. Brunei University, school of Information systems, Computing and Mathematics Theses.

Vyas, A.O. and, Dudul, s.v. (2020). A Novel Approach of Object oriented Image Steganography Using LSB. ICDSMLA 2019. Lecture Notes in Electrical Engineering. 601. 144-151.

PNG specification: Chunk specifications, accessed 24 August 2021, <<https://www.w3.org/TR/PNG-Chunks.html>>

The PHP Group, PHP Manual, accessed 11 August 2021, <<https://www.php.net/manual/en/index.php>>

Philip Norton, 2021, Steganography with images in PHP, accessed 11 August 2021, <<https://www.hashbangcode.com/article/steganography-images-php>>

BoiteAKlou, 2018, Steganography Tutorial: Least Significant Bit (LSB), accessed 14 August 2021, <<https://www.boiteaklou.fr/Steganography-Least-Significant-Bit.html>>