**INTRODUCTION**

**1.1 OVERVIEW**

Today, graphical codes, such as EAN-13 barcode, Quick Response (QR) code, Data Matrix, PDF417, are frequently used in our daily lives. These codes have a huge number of applications including: information storage (advertising, museum art description), redirection to web sites, track and trace (for transportation tickets or brands), identification The popularity of these codes is mainly due to the following features: they are robust to the copying process, easy to read by any device and any user, they have a high encoding capacity enhanced by error correction facilities, they have a small size and are robust to geometrical distortions.

However, those undeniable advantages also have their counterparts:

1) Information encoded in a QR code is always accessible to everyone, even if it is ciphered and therefore is only legible to authorized users (the difference between “see” and “understand”).

2) It is impossible to distinguish an originally printed QR code from its copy due to their sensitivity to the Print-and-Scan (P&S) process.

In this project, we propose to overcome these shortcomings by enriching the standard QR code encoding capacity. This enrichment is obtained by replacing its black modules by specific textured patterns. Besides the gain of storage capacity, these patterns can be designed to be sensitive to distortions due to the P&S process. These patterns, that do not introduce disruption in the standard reading process, are always perceived as black modules by any QR code reader. Therefore, even when the private information is degraded or lost in the copy, the public information is always accessible for reading.

**1.2 APPROACH**

The proposed two level QR (2LQR) code contains of: a first level accessible for any standard QR code reader, therefore it keeps the strong characteristics of the QR code; and a second level that improves the capacities and characteristics of the initial QR code. The information in the second level is encoded by using *q*−ary (*q* ≥ 2) code with error correction capacities. This information is invisible to the standard QR code reader because it perceives the textured patterns as black modules. Therefore, the second level can be used for private message sharing. Additionally, thanks to textured pattern sensitivity to P&S distortions, the second level can be used to distinguish the original 2LQR code from its copies.

The paper is organized as follows. We start with an introduction of QR code features and existing rich graphical codes in Section II. In addition, the distortion added during the P&S process will be discussed there. The proposed two level QR (2LQR) code as well as the proposed recognition method are presented in Section III. In Section IV, the experimental results show the efficiency of the proposed recognition methods and analyze the capacities of the proposed 2LQR code. Finally, Section V represents conclusions and perspectives.

The QR code was invented for the Japanese automotive industry by Denso Wave1 corporation in 1994. The most important characteristics of this code are small printout size and high speed reading process. The certification of QR code was performed by International Organization of Standardization (ISO), and its whole specification can be found in. A QR code encodes the information into binary form. Each information bit is represented by a black or a white module.

The Reed-Solomon error correction code [15] is used for data encryption. Therefore, one of 4 error correction levels has to be chosen during QR code generation. The lowest level can restore nearly 7% of damaged information, the highest level can restore nearly 30%. Today, 40 QR code versions are available with different to storage capacities. The smallest QR code version (version V1) has a 21 × 21 module size. It can store 152 bits of raw data at the lowest correction level. The biggest QR code version (version V40) has a 177 × 177 module size. It can store a maximum of 7089 bits of raw data at its lowest correction level.

As illustrated in, the QR code has a specific structure for geometrical correction and high speed decoding. Three position tags are used for QR code detection and orientation correction. One or more alignment patterns are used to code deformation adjustment. The module coordinates are set by timing patterns. Furthermore, the format information areas contain error correction level and mask pattern. The code version and error correction bits are stored in the version information areas.

The QR code generation algorithm consists of information encoding using Reed-Solomon error correction code, information division on code words, application of mask pattern, placement of code words and function patterns into the QR code. The QR code recognition algorithm includes the scanning process, image binarization, geometrical correction and decoding algorithm.

**LITERATURE SURVEY**

**2.1 IMAGE EMBEDDING IN QR CODE:**

The QR (Quick Response) code is a two-dimensional barcode developed by the Japanese company Denso-Wave in 1994, and was approved as an ISO International Standard and Chinese National Standard in 2000. The QR code has been widely used due to its good features such as large data capacity, high speed scan, and small printout size. Increase in number of smart phones is the reason behind popularity of QR code. Smart phones are capable of decoding and accessing on line resources as well as it has high storage capacity and high speed of decoding. QR codes are used in a various application, such as accessing websites, initiate phone calls, reproduce videos or open text documents and data storing purposes. An important problem of QR codes is its noisy looks. To improve the appearance of QR code and to reduce noisy black and white random texture has generated great interest for algorithms capable of embedding QR codes into images without losing decoding robustness. There have been many efforts to improve the appearance of such embedding. The main challenge of any embedding method is the embedded result should be decodable by standard applications. The embedding introduces changes in the luminance of the code, distorting the binarization thresholds and thus increasing the probability of detection error. The second challenge is the problem of using the entire area of the code in which the image or logo is to be embedded. This cannot be done by simply replacing information modules with the desired image. A good embedding method should decrease the number of corrupted modules and uses the utmost area. The proposed method is based on the selection of a set of pixels using genetic algorithm. The concentration of pixels and its corresponding luminance are optimized to minimize a visual distortion. Distortion metric is subject to a constraint in the probability of error.

QR code consists of black and white square blocks called as modules of a QR code. Each module is assigned a single bit value. Information is encoded into the QR modules. A dark module is binary one and a light module is binary zero. A code word contains 8 bits of information. There are 40 versions of QR code. A QR code with version V have (17 + 4V) × (17 + 4V) number of modules. Therefore, version 1 has 21 × 21 modules whereas version 40 corresponds to 177 × 177 modules. Fig. 1 shows the structure of a QR code. Finder pattern contains three identical square shape located at the three corners of QR code. Finder pattern is the most important pattern which enables the detection of QR code. Alignment patterns are also essential to locate, rotate and aligning the QR code. Finder pattern, timing pattern and alignment pattern are collectively known as function pattern region of QR code. Alignment patterns are observed with version number 2 and onwards however version number 1 does not have any alignment pattern. Encoding region within the green color consists of data and error correction code words. Data code words are of two types i.e. information code words which stores the actual information and the second is padding code words. Encoding region stores the data, parity modules and decoding information in the form of a code words. A code word consists of a block of 8 modules. Quite Zone is the guard region of QR code. QR code utilizes RS (Reed Solomon) codes for error correction. A QR code contains multiple RS codes where one RS code is sufficient to store the message. The remaining RS codes are usually used to store non meaningful messages [2]. There are 4 types of error correction level i.e. L, M, Q and H which can recover 7%, 15%, 25% and 30% of errors in the code words respectively.

There have been a lot of efforts to improve the appearance of QR code. The base strategy of such work is to find the best group of QR modules to substitute by the image or logo in the QR code. The method presented in [3] proposed that, there are three areas to replace the QR module by the image or logo. These areas include data code words, padding code words and the error correcting code words. Depending on the error correction level of QR code, pad characters have been changed. The size of the embedding image in the QR code is identified and then the image is implanted in the identified region of QR code. The size of the image which is to be embedded is increased and tested the readability of QR embedding to find largest size of which the image could be embedded except the finder pattern of QR code. Author concludes that if the numbers of characters in the QR code is decreased then the larger image can be embedded. The second approach [12] of embedding is based on the modification of the pixel’s luminance. The luminance of central pixels is modified since this is the area usually sampled by the decoder. This approach uses the entire area of the code for embedding except the finder and alignment pattern. The approach in [10] performs the blending which combines the color image C and the QR code Q based on the luminance of color image and the binary value of QR code. The blending of C and Q to produce an output B is accomplished by replacing pixels of Q with those of C. Author assumes that pixels of Q are normalized so that white pixels have a luminance of 1, and black pixels have a luminance of 0. This algorithm ensures that the blended output image preserves the bright part of color image when the pixel value of the QR code equals to 1, and dark part of the color image when pixel value is 0. Cox proposed a complicated algorithm [19] to embed a binary image into a QR code during the data encoding stage of generating the code. He carefully investigated the internal structure of QR code and the logic behind data encoding, and designed an algorithm to encode image content as redundant numeric strings appended to the original data. However, this technique works only for URL type data string and the quality of embedded image is limited by the length of encoded URL.

**2.2 DISTORTION MODELING AND INVARIANT EXTRACTION FOR DIGITAL IMAGE PRINT-AND-SCAN PROCESS:**

After an image is printed-and-scanned, it is usually filtered, rotated, scaled, cropped, contrast-and luminance adjusted, as well as distorted by noises. This paper presents models for the print-and-scan process, considering both pixel value distortion and geometric distortion. We show properties of the discretized, rescanned image in both the spatial and frequency domains, then further analyze the changes in the Discrete Fourier Transform (DFT) coefficients. Based on these properties, we show several techniques for extracting invariants from the original and rescanned image, with potential applications in image watermarking and authentication. Preliminary experiments show the validity of the proposed model and the robustness of the invariants.

Today the print-and-scan (PS) process is commonly used for image reproduction and distribution. It is popular to transform images between the electronic digital format and the printed format. The rescanned image may look similar to the original, but may have been distorted during the process. For some image security applications, such as watermarking for copyright protection, users should be able to detect the embedded watermark even if it is printed-and-scanned. In image authentication cases, the rescanned image may be considered as authentic, because it is a reproduction of the original. Little work has been done to understand the changes that digital images undergo after the PS process. Most work discusses individual models of printing or scanning. In this paper, we begin with the characteristics of the PS process. Then, in Section 3, we propose a model that can be used to analyze the distortion of a discretized digital image after the PS process in the spatial and frequency domain. Then, we will analyze the variations of DFT coefficients, leading to important properties for extracting invariants. In Section 4, we discuss several methods that can be used to extract invariants of the PS process. Some experimental results, including an analysis of the feature vector proposed in [5], are shown in Section 5. In Section 6, we make a summary and discuss some future work.

Distortion occurs in both the pixel values and the geometric boundary of the rescanned image. The distortion of pixel values is caused by (1) the luminance, contrast, gamma correction and chromnance variations, and (2) the blurring of adjacent pixels. These are typical effects of the printer and scanner, and while they are perceptible to the human eye, they affect the visual quality of a rescanned image. Distortion of the geometric boundary in the PS process is caused by rotation, scaling, and cropping (RSC). Although it does not introduce significant effects on the visual quality, it may introduce considerable changes at the signal level, especially on the DFT coefficients of the rescanned image. It should be noted that, in general image editing processes, geometric distortion cannot be adequately modeled by the well-known rotation, scaling, and translation (RST) effects, because of the design of today’s Graphic User Interface (GUI) for the scanning process. From Figure 1, we can see that users can arbitrarily select a range for the scanned image. We use “cropping” to describe this operation, because the rescanned images are cropped from an area in the preview window, including the printed image and background. The RST model, which has been widely used in pattern recognition, is usually used to model the geometric distortion on the image of an observed object. In those cases, the meaning of RST is based on a fixed window size, which is usually pre-determined by the system. However, in the PS process, the scanned image may cover part of the original

picture and/or part of the background, and may have an arbitrarily cropped size. These changes, especially that of image size, will introduce significant changes of the DFT coefficients. Therefore, instead of RST, a RSC model is more appropriate to represent the PS process.

**2.3 SURVEY ON INFORMATION HIDING TECHNIQUES USING QR BARCODE:**

Nowadays, the information processing system plays crucial part in the internet. Online information security has become the top priority in all sectors. Failing to provide online information security may cause loss of critical information or someone may use or distribute such information for malicious purpose. Recently QR barcodes have been used as an effective way to securely share information. This paper presents the survey on information hiding techniques which can share high security information over network using QR barcode.

Due to tremendous growth in communication technology, sharing the information through the communication network has never been so convenient. Nowadays information is processed electronically and conveyed through public networks. Such networks are unsecured and hence sensitive information needs to be protected by some means. Cryptography is the study of techniques that allows us to do this. In order to protect information from various computer attacks as well as network attacks various cryptographic protocols and firewalls are used. But no single measure can ensure complete security.

Nowadays, the use of internet and sharing information are growing increasingly across the globe, security becomes a vital issue for the society. Security attacks are classified as passive attacks and active attacks [11, 12]. In passive attacks, attacker monitors network traffic and looks for sensitive information but does not affect system resources. Passive attacks include traffic analysis, eavesdropping, Release of message contents [11, 12]. In active attack, attacker breaks protection features to gain unauthorized access to steal or modify information. Active attacks

include masquerade, replay, modification of messages, and denial of service [11, 12]. Therefore, security threats (such as eavesdropping, data modification, phishing, website leaks etc.) force us to develop new methods to counter them. Considering QR barcodes as an effective media of sharing information, many researchers have proposed information/data hiding methods [6,7, 8, 9.] as well as online transaction systems [1,2,3,4,5] using QR barcode. In this paper, we describe different information hiding schemes using QR barcode.

**PROPOSED SYSTEM**

In this system, we propose a two-level QR code. These two levels are public and private level. These levels are used for storage.

The public level is the same as the standard QR code storage level; therefore, it is readable by any classical QR code application.

The private level is constructed by replacing the black modules by specific textured patterns. It consists of information encoded using q-ary code with an error correction capacity.

This allows us not only to increase the storage capacity of the QR code, but also to distinguish the original document from a copy. This authentication is due to the sensitivity of the used patterns to the print-and-scan (P&S) process.

The pattern recognition method that we use to read the second-level information can be used both in a private message sharing and in an authentication scenario. It is based on maximizing the correlation values between P&S degraded patterns and reference patterns.

The storage capacity can be significantly improved by increasing the code alphabet q or by increasing the textured pattern size.

**DETAILED DESIGN OF THE PROJECT**

**4.1 FEASIBILITY STUDY**

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operational Feasibility
* Economic Feasibility

**4.2 REQUIREMENTS SPECIFICATION**

**4.2.1 System requirements specification:**

A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* Business requirements describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify specific methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement); a requirement that the product be maintainable (a Product requirement) often is addressed by imposing requirements to follow particular development styles.

**4.2.2 Software Requirements Specification (SRS)**

A requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

Software Requirements:

Operating System: Windows 7/Windows 8

IDE: Net Beans 8.0/7.3(Based on JDK, select its compatible NetBeans version)

Software setup : Jdk8/JDK 7/JDK 6

Technology: Java Standard Edition

Note: As per Operating system setting need to Install JDK (32bit/64 bit)

Hardware Requirements (Minimum)

Processor: Any Processor above 500 MHz

RAM: 2 GB

Hard Disk: 10 GB

Compact Disk: 650 Mb

Input device: Standard Keyboard and Mouse

**4.3 SYSTEM DESIGN**

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided.

During, Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules. In other works, in system design the attention is on what components are needed, while in detailed design how the components can be implemented in software is the issue.

Design is concerned with identifying software components specifying relationships among components. Specifying software structure and providing blue print for the document phase. Modularity is one of the desirable properties of large systems. It implies that the system is divided into several parts. In such a manner, the interaction between parts is minimal clearly specified.

During the system design activities, Developers bridge the gap between the requirements specification, produced during requirements elicitation and analysis, and the system that is delivered to the user.

Design is the place where the quality is fostered in development. Software design is a process through which requirements are translated into a representation of software.

**4.4 SYSTEM MODEL**

**4.4.1 Introduction to UML**

The unified Modeling Language (UML) is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document the artifacts of software-intensive system.

The goal of UML is to provide a standard notation that can be used by all object - oriented methods and to select and integrate the best elements. UML is itself does not prescribe or advice on how to use that notation in a software development process or as part of an object - design methodology. The UML is more than just bunch of graphical symbols. Rather, behind each symbol in the UML notation is well-defined semantics.

The system development focuses on three different models of the system.

* Functional model
* Object model
* Dynamic model

**4.4.2 Use Case Model**

Use case diagrams represent the functionality of the system from a user point of view. A Use case describes a function provided by the system that yields a visible result for an actor. an actor describes any entity that interacts with the system. The identification of actors and use cases results in the definition of the boundary of the system, which is, in differentiating the tasks accomplished by the system and the tasks accomplished by its environment. The actors outside the boundary of the system, whereas the use cases are inside the boundary of the system

A Use case contains all the events that can occur between an actor and a set of scenarios that explains the interactions as sequence of happenings.



Fig:4.1

**4.4.3 Class Diagram**

Class Diagrams are used to describe the structure of the system. Classes are abstractions that specify the common structure and behavior of a set of objects. Objects are instances of classes that are created, modified and destroyed during the execution of a system. An object has state that includes the values of its attributes and links with other objects.

The class diagram is used to refine the use cases diagrams and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the classes. Apart from this, each class may have certain "attributes" that uniquely identify the class. In the class diagram these classes are represented with boxes which contain three parts.

****

Fig:4.2

**4.4.4 Sequence Diagram**

Sequence diagrams are used to formalize the dynamic behavior of the system and to visualize the communication among the objects. They are useful for identifying the additional objects that participate in the use case. Sequence diagram represent the objects participating in the interaction horizontally and time vertically.

Sequence diagrams typically show a user or actor and the objects and the components they interact with the execution of the use case. Each column represents an objects that participate in the interaction. Message is shown by solid arrows. Labels on the solid arrows represent the message names. Activations are depicted by vertical rectangles. The actor who initiates the interaction is shown in the left most columns. The messages coming from the actor represent the interactions described in the use case diagrams.

****

Fig:4.3

**4.4.5 Collaboration Diagram**

****Fig:4.4

**4.4.6 State Chart Diagram**

UML State chart is notation for describing the sequence of states an object goes through in response to external events. Objects have behavior and state. The state of an object depends on its current activity or condition. A state chart diagram shows the possible states of the object ad the transitions that cause a change in state.

State chart describes the dynamic behavior of an individual object as a number of states. A state is a condition satisfied by attributes of objects. Given a state, a transition represents a future state the object can move to and the conditions associated with the change of state.

A state is depicted by a rounded rectangle A transition is depicted by open arrows connecting two states. States are labeled with their names. A small solid black circle indicates the initial state and a circle surrounding the small solid circle indicates the final state.



Fig:4.5

**4.4.7 Activity Diagram**

An Activity diagram describes the behavior of the system in terms of activities. Activities are modeling elements that represent the execution of set of operations. The completion of these operations triggers a transition to another activity. Activity diagrams similar to flowchart diagrams in that they can be used to represent control flow and data flow. Activities are represented by rounded rectangles and arrows are represented transition between activities. Think bars represent the synchronization of the control flow.

****

Fig:4.6

**4.4.8 Data Flow Diagrams(c)**

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts. The Basic Notation used to create a DFD’s are as follows:

**1. Dataflow:** Data move in a specific direction from an origin to a destination.

**2. Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

**3. Source:** External sources or destination of data, which may be People, programs, organizations or other entities.

Message

2LQR Code

Fig:4.7

**IMPLEMENTATION**

**5.1 Technology Description**

**5.1.1 Java Technology**

The Java platform consists of the Java application programming interfaces (APIs) and the Java virtual machine (JVM). The following Java technology lets developers, designers, and business partners develop and deliver a consistent user experience, with one environment for applications on mobile and embedded devices. Java meshes the power of a rich stack with the ability to deliver customized experiences across such devices.

Java APIs are libraries of compiled code that you can use in your programs. They let you add ready-made and customizable functionality to save you programming time. Java programs are run (or interpreted) by another program called the Java Virtual Machine. Rather than running directly on the native operating system, the program is interpreted by the Java VM for the native operating system. This means that any computer system with the Java VM installed can run Java programs regardless of the computer system on which the applications were originally developed. In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris TM Operating System (Solaris OS), Linux, or Mac OS.

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine1 (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

**5.1.2 Swings**:

Swings is an extension of AWT. The swing components do not depend upon the files of the local os in creating the GUIs and thereby, they decrease the overhead on the JVM.

Thus, swings are known as Light-Weight Components.

The following is the class hierarchy of java.awt package.

Picture1

Fig:5.1

**SYSTEM TESTING**

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time and quality.

**6.1 Levels of Testing**

Since the errors in the software can be injured at any stage. So, we have to carry out the testing process at different levels during the development. The basic levels of testing are Unit, Integration, System and Acceptance Testing.

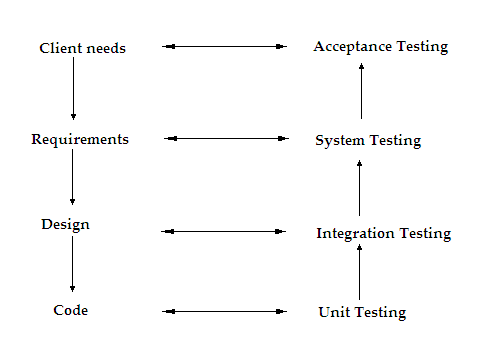


Fig:6.1

**6.2 TEST CASES**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test No** | Input | Expected Behavior | **Observed**  **Behavior** | **Status** |
| 1 | Enter the details of Public Data | Name:Ram  D.O.B:06/08/1992  Address:Bobbili | After Enter details submit Generate QR-code | True |
| 2 | Enter the Private message | Enter any message | It Navigates to ECC | True |
| 3 | Show Code-Words | Observe those codeword’s and it navigate to Texture Pattern | Observe those codeword’s and it navigate to Texture Pattern | True |
| 4 | 2LQR-code Generator | P&S Navigates to 2LQR-code | P&S Navigates to 2LQR-code | True |

Table :6.1

**CONCLUSION**

In this paper a new rich code called two level QR (2LQR) code is proposed. This 2LQR code has two levels: a public level and a private level. The public level can be read by any QR code reading application, while the private level needs a specific application with specific input information. This 2LQR code can be used for private message sharing or for authentication scenarios. The private level is created by replacing black modules with specific textured patterns. These textured patterns are considered as black modules by standard QR code reader. Thus the private level is invisible to standard QR code readers. In addition, the private level does not affect in anyway the reading process of the public level. The proposed 2LQR code increases the storage capacity of the classical QR code due to its supplementary reading level. Experiment results show that the storage capacity is improved by up to 28% (transition from message size equal to 272 bits to a message length of 380 bits). The storage capacity of the2LQR code can be improved by increasing the number of textured patterns used or by decreasing the textured pattern size. All experiments show that even with a pattern size of 6×6 pixels and with an alphabet dimension q = 8, it is possible to obtain good pattern recognition results, and therefore a successful private message extraction. However, we are facing a trade-off between the pattern size, the alphabet dimensions and the quantity of stored information during the 2LQR code generation. One important feature of the textured patterns used is their sensitivity to the P&S process. To take advantage of this sensitivity, we use a pattern recognition method based on maximization of correlation values among the P&S degraded versions and characterization patterns. We have tried three different types of characterization patterns: mean patterns, median patterns (for the private message sharing scenario) and original patterns (for the document authentication scenario). The mean and median characterization patterns give almost the same results of pattern detection. Therefore, either of them can be used in the private message sharing scenario. The best pattern recognition results were obtained, when the original patterns are used as characterization patterns. The original patterns can be also used for the private message sharing scenario, but in this case the blind method for pattern detection cannot be performed. The suggested textured patterns can be distinguished only after one P&S process. Therefore, we can use the detection method with original patterns in order to ensure good document authentication results.

**FUTURE ENHANCEMENTS**

In our future work, we will address five different paths. The first path will concern the improvements of the pattern recognition method. The second will cover the textured pattern analysis to automate its combination process. The third will deal with message recovering and authentication attacks, such as cropping and code reconstruction. The forth path will concern the study of the second level recovery problems in the 2LQR code images captured by a camera. In the last path, the storage capacity of 2LQR code will be increased by replacing also the white modules with textured patterns, which have small density than black pixels.

**APPENDICES I**

**9.1 SAMPLE SCREENS:**

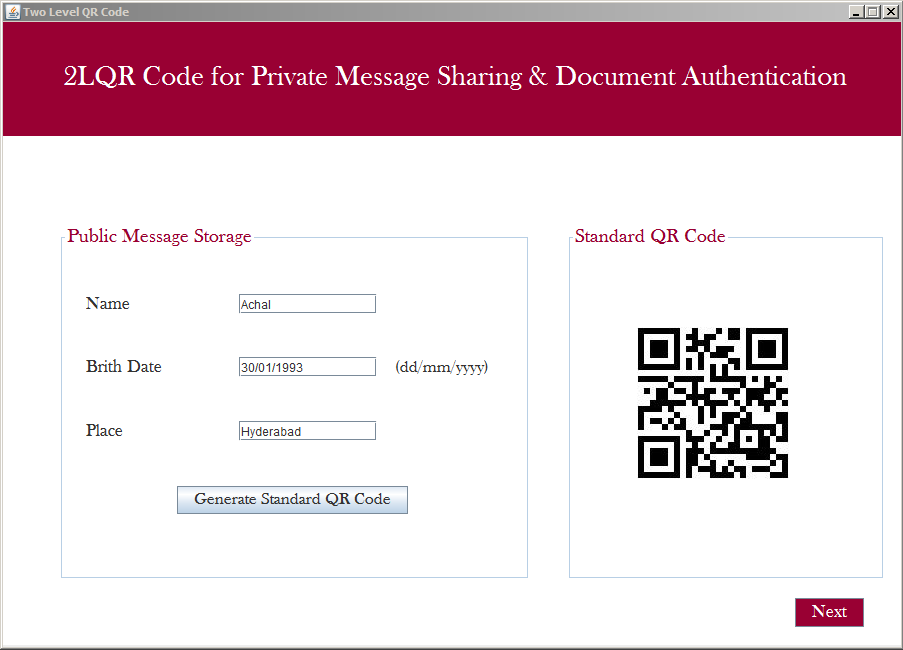


Fig:9.1

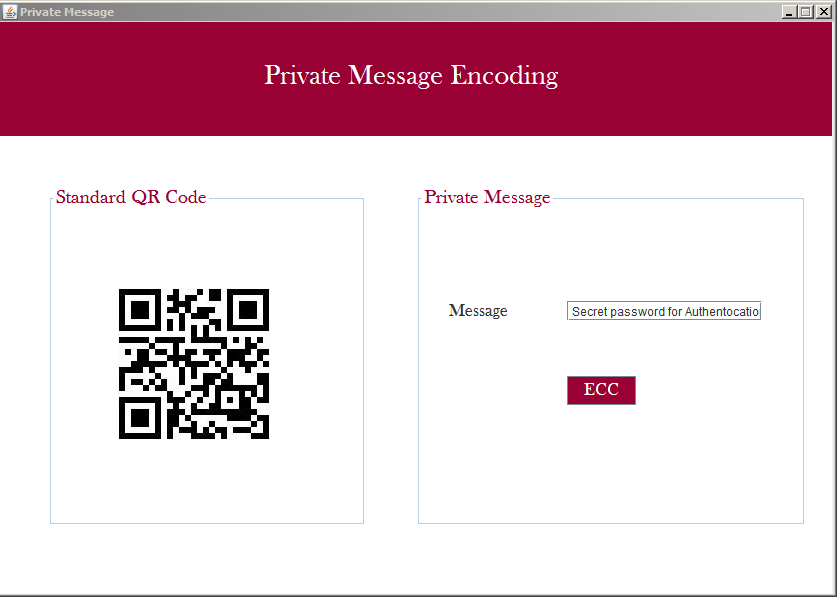


Fig:9.2

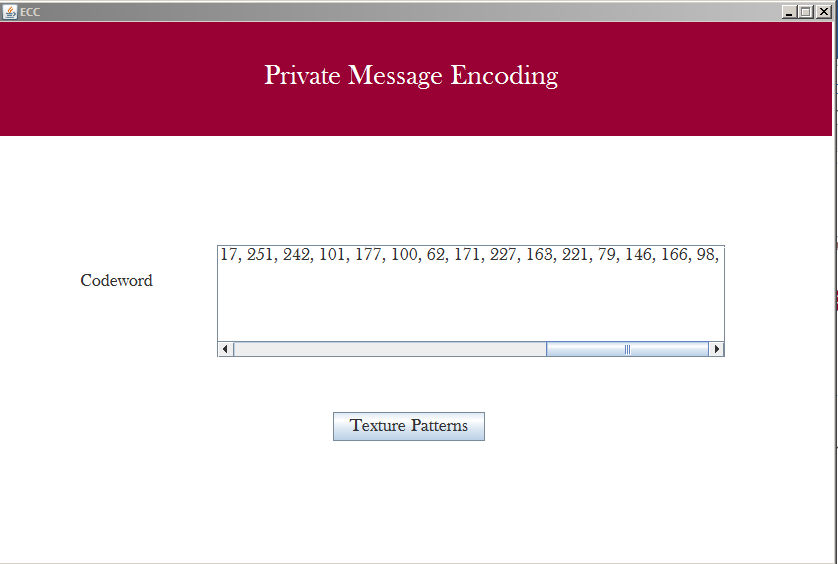


Fig:9.3

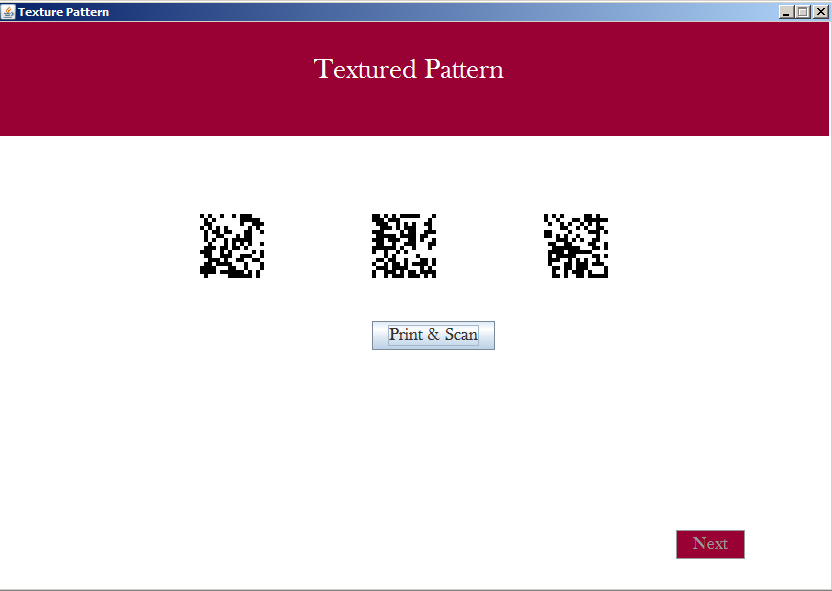


Fig:9.4

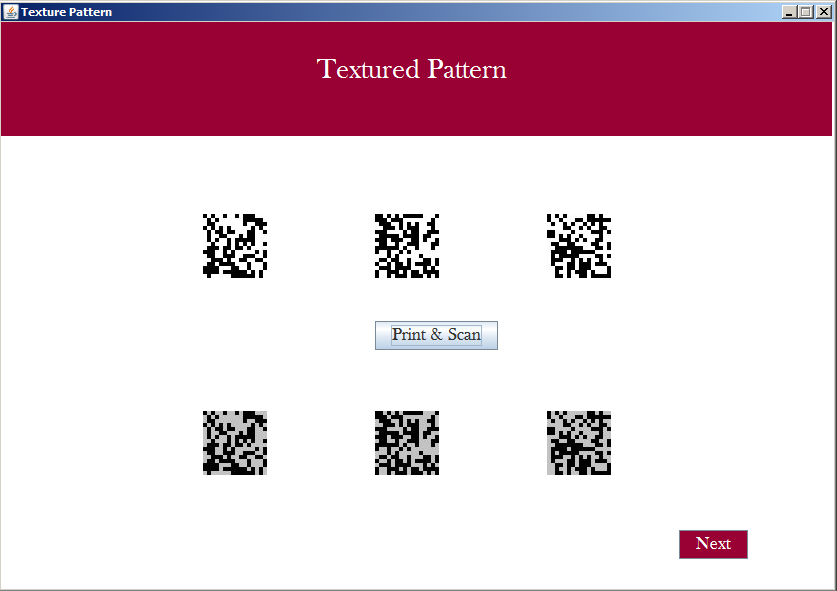


Fig:9.5

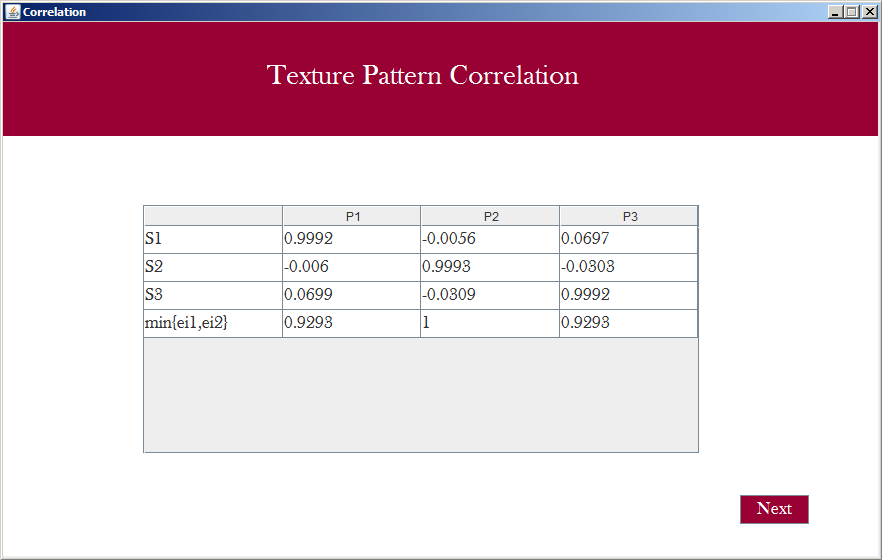


Fig:9.6



Fig:9.7



Fig:9.8

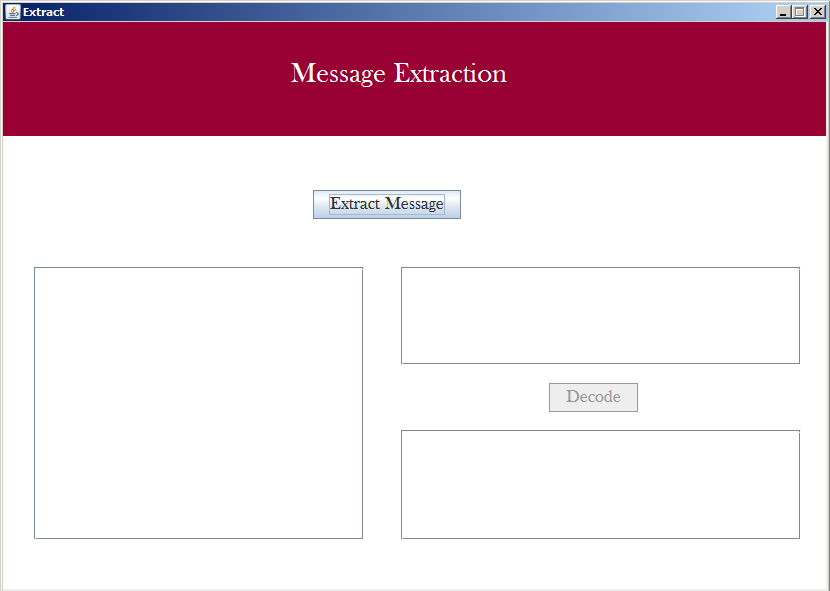


Fig:9.9

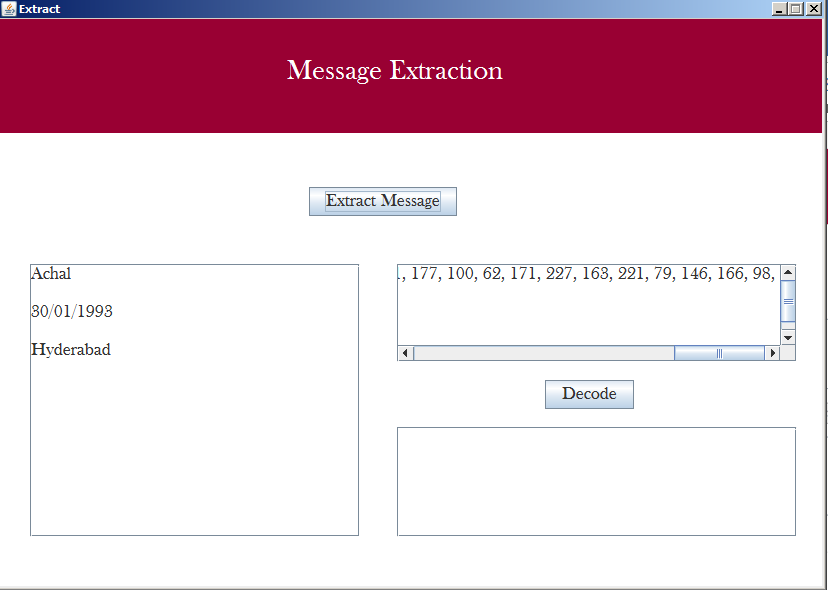


Fig:9.10

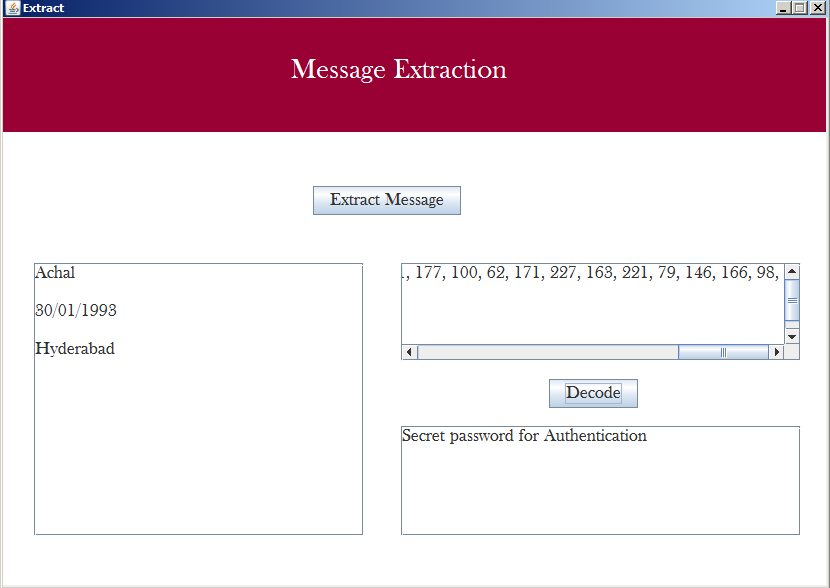


Fig:9.11

**APPENDICES II**

**10. SAMPLE CODE:**

**Main.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package qrcode;

public class Main {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

MainFrame mf=new MainFrame();

mf.setVisible(true);

mf.setResizable(false);

mf.setTitle("Two Level QR Code");

}

}

**MainFrame.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package qrcode;

import javax.swing.JOptionPane;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

import java.io.File;

import javax.swing.ImageIcon;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.BinaryBitmap;

import com.google.zxing.EncodeHintType;

import com.google.zxing.MultiFormatReader;

import com.google.zxing.MultiFormatWriter;

import com.google.zxing.NotFoundException;

import com.google.zxing.Result;

import com.google.zxing.ResultPoint;

import com.google.zxing.WriterException;

import com.google.zxing.client.j2se.BufferedImageLuminanceSource;

import com.google.zxing.client.j2se.MatrixToImageWriter;

import com.google.zxing.common.BitMatrix;

import com.google.zxing.common.HybridBinarizer;

import com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;

public class MainFrame extends javax.swing.JFrame {

/\*\*

\* Creates new form LoginFrame

\*/

Details dt=new Details();

public MainFrame() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jLabel5 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jPanel4 = new javax.swing.JPanel();

jLabel6 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel1.setEnabled(false);

jPanel2.setBackground(new java.awt.Color(153, 0, 51));

jLabel1.setFont(new java.awt.Font("Baskerville Old Face", 0, 28)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("2LQR Code for Private Message Sharing & Document Authentication");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(60, 60, 60)

.addComponent(jLabel1)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(42, 42, 42)

.addComponent(jLabel1)

.addContainerGap(44, Short.MAX\_VALUE))

);

jPanel3.setBackground(new java.awt.Color(255, 255, 255));

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Public Message Storage", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Baskerville Old Face", 0, 20), new java.awt.Color(153, 0, 51))); // NOI18N

jLabel2.setFont(new java.awt.Font("Baskerville Old Face", 0, 18)); // NOI18N

jLabel2.setText("Name");

jLabel3.setFont(new java.awt.Font("Baskerville Old Face", 0, 18)); // NOI18N

jLabel3.setText("Brith Date");

jLabel4.setFont(new java.awt.Font("Baskerville Old Face", 0, 18)); // NOI18N

jLabel4.setText("Place");

jLabel5.setFont(new java.awt.Font("Baskerville Old Face", 0, 17)); // NOI18N

jLabel5.setText("(dd/mm/yyyy)");

jButton1.setFont(new java.awt.Font("Baskerville Old Face", 0, 17)); // NOI18N

jButton1.setText("Generate Standard QR Code");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGap(22, 22, 22)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel2)

.addComponent(jLabel3)

.addComponent(jLabel4))

.addGap(77, 77, 77)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jTextField1)

.addComponent(jTextField2)

.addComponent(jTextField3, javax.swing.GroupLayout.DEFAULT\_SIZE, 138, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jLabel5))

.addGroup(jPanel3Layout.createSequentialGroup()

.addGap(113, 113, 113)

.addComponent(jButton1)))

.addContainerGap(36, Short.MAX\_VALUE))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGap(45, 45, 45)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(jLabel2)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(43, 43, 43)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel5))

.addGap(44, 44, 44)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel4)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(45, 45, 45)

.addComponent(jButton1)

.addContainerGap(61, Short.MAX\_VALUE))

);

jPanel4.setBackground(new java.awt.Color(255, 255, 255));

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "Standard QR Code", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Baskerville Old Face", 0, 20), new java.awt.Color(153, 0, 51))); // NOI18N

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(41, 41, 41)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 223, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(44, Short.MAX\_VALUE))

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(51, 51, 51)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 207, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jButton2.setBackground(new java.awt.Color(153, 0, 51));

jButton2.setFont(new java.awt.Font("Baskerville Old Face", 0, 18)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Next");

jButton2.setEnabled(false);

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(56, 56, 56)

.addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(37, 37, 37)

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(14, Short.MAX\_VALUE))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2)

.addGap(37, 37, 37))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(91, 91, 91)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addComponent(jButton2)

.addGap(0, 18, Short.MAX\_VALUE))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 0, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

// TODO add your handling code here:

try

{

String name=jTextField1.getText().trim();

String dob=jTextField2.getText().trim();

String loc=jTextField3.getText().trim();

if(name.equals("")||dob.equals("")||loc.equals(""))

JOptionPane.showMessageDialog(this, "Enter all the Details");

else

{

boolean bool=validateDate(dob);

if(bool)

{

String msg=name+"#"+dob+"#"+loc;

String charset = "UTF-8";

Map<EncodeHintType, ErrorCorrectionLevel> hintMap = new HashMap<EncodeHintType, ErrorCorrectionLevel>();

hintMap.put(EncodeHintType.ERROR\_CORRECTION, ErrorCorrectionLevel.L);

String fpath="Q1.jpg";

BitMatrix matrix = new MultiFormatWriter().encode(new String(msg.getBytes(charset), charset),BarcodeFormat.QR\_CODE, 200, 200, hintMap);

MatrixToImageWriter.writeToFile(matrix, fpath.substring(fpath.lastIndexOf('.') + 1), new File(fpath));

int rec[]=matrix.getEnclosingRectangle();

dt.x1=rec[0];

dt.x2=rec[0]+rec[2];

jLabel6.setIcon(new ImageIcon(fpath));

jButton2.setEnabled(true);

}

else

JOptionPane.showMessageDialog(this, "Invalid Date");

}

}

catch(Exception e)

{

e.printStackTrace();

}

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

// TODO add your handling code here:

try

{

PrivateMsgFrame pf=new PrivateMsgFrame();

pf.setVisible(true);

pf.setTitle("Private Message");

pf.setResizable(false);

pf.jLabel6.setIcon(new ImageIcon("Q1.jpg"));

}

catch(Exception e)

{

e.printStackTrace();

}

}//GEN-LAST:event\_jButton2ActionPerformed

public boolean validateDate(String dat)

{

boolean bool=true;

try

{

String dateFromat="dd/mm/yyyy";

SimpleDateFormat sdf = new SimpleDateFormat(dateFromat);

sdf.setLenient(false);

Date date = sdf.parse(dat);

}

catch(Exception e)

{

bool=false;

}

return bool;

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(MainFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MainFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

// End of variables declaration//GEN-END:variables

}

**Details.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package qrcode;

public class Details

{

static int x1=0;

static int x2=0;

static int ccp[];

static String encd="";

static int inLen=0;

}

**QRFrame.java**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package qrcode;

import java.awt.Graphics2D;

import java.awt.Image;

import java.awt.Color;

import java.awt.image.BufferedImage;

import java.awt.image.ImageObserver;

import static java.awt.image.ImageObserver.ALLBITS;

import java.awt.image.ImageProducer;

import java.awt.image.MemoryImageSource;

import java.awt.image.PixelGrabber;

import java.awt.image.VolatileImage;

import java.io.File;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

public class QRFrame extends javax.swing.JFrame {

/\*\*

\* Creates new form QRFrame

\*/

Details dt=new Details();

public QRFrame() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jPanel4 = new javax.swing.JPanel();

jLabel6 = new javax.swing.JLabel();

jPanel5 = new javax.swing.JPanel();

jLabel7 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));

jPanel2.setBackground(new java.awt.Color(153, 0, 51));

jLabel1.setFont(new java.awt.Font("Baskerville Old Face", 0, 28)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 255, 255));

jLabel1.setText("2LQR Code Generation");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(316, 316, 316)

.addComponent(jLabel1)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGap(35, 35, 35)

.addComponent(jLabel1)

.addContainerGap(51, Short.MAX\_VALUE))

);

jPanel4.setBackground(new java.awt.Color(255, 255, 255));

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "2LQR Code", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Baskerville Old Face", 0, 20), new java.awt.Color(153, 0, 51))); // NOI18N

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(41, 41, 41)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 223, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(44, Short.MAX\_VALUE))

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addGap(51, 51, 51)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 207, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(62, Short.MAX\_VALUE))

);

jPanel5.setBackground(new java.awt.Color(255, 255, 255));

jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "PS 2LQR Code", javax.swing.border.TitledBorder.DEFAULT\_JUSTIFICATION, javax.swing.border.TitledBorder.DEFAULT\_POSITION, new java.awt.Font("Baskerville Old Face", 0, 20), new java.awt.Color(153, 0, 51))); // NOI18N

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);

jPanel5.setLayout(jPanel5Layout);

jPanel5Layout.setHorizontalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addGap(41, 41, 41)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 223, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(44, Short.MAX\_VALUE))

);

jPanel5Layout.setVerticalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel5Layout.createSequentialGroup()

.addGap(51, 51, 51)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 207, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap(54, Short.MAX\_VALUE))

);

jButton1.setFont(new java.awt.Font("Baskerville Old Face", 0, 18)); // NOI18N

jButton1.setText("P&S");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(153, 0, 51));

jButton2.setFont(new java.awt.Font("Baskerville Old Face", 0, 18)); // NOI18N

jButton2.setForeground(new java.awt.Color(255, 255, 255));

jButton2.setText("Next");

jButton2.setEnabled(false);

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton2))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(71, 71, 71)

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33)

.addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 20, Short.MAX\_VALUE)

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(74, 74, 74))

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(45, 45, 45)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(jPanel1Layout.createSequentialGroup()

.addGap(181, 181, 181)

.addComponent(jButton1)))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 19, Short.MAX\_VALUE)

.addComponent(jButton2)

.addGap(29, 29, 29))

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>//GEN-END:initComponents

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed

// TODO add your handling code here:

try

{

BufferedImage bi=new BufferedImage(200,200,BufferedImage.TYPE\_BYTE\_GRAY);

for(int i=0;i<bi.getWidth();i++)

{

for(int j=0;j<bi.getHeight();j++)

{

bi.setRGB(i, j, new Color(255,255,255).getRGB());

}

}

for(int i=dt.x1;i<dt.x2;i++)

{

for(int j=dt.x1;j<dt.x2;j++)

{

bi.setRGB(i, j, new Color(195,195,195).getRGB());

}

}

ImageIO.write(bi, "jpg", new File("grQ.jpg"));

ImageIcon ic=new ImageIcon("grQ.jpg");

ImageIcon ic1=new ImageIcon("2LQR.jpg");

Image grimg=ic.getImage();

Image img1=ic1.getImage();

int w=ic.getIconWidth();

int h=ic.getIconHeight();

int pel[]=new int[h\*w];

int pel1[]=new int[h\*w];

PixelGrabber pg = new PixelGrabber(grimg,0,0,w,h,pel,0,w);

pg.grabPixels();

PixelGrabber pg1 = new PixelGrabber(img1,0,0,w,h,pel1,0,w);

pg1.grabPixels();

int newpix1[]=new int[h\*w];

int ct1=0;

int c1=0;

for(int i=0;i<pel.length;i++)

{

newpix1[ct1]=pel[c1] & pel1[c1]; // corr

c1++;

ct1++;

}

ImageProducer ip1= new MemoryImageSource(w, h, newpix1, 0, w);

Image iim1=createImage(ip1);

BufferedImage rd1=toBufferedImage(iim1,1);

ImageIO.write(rd1, "jpg", new File("psqr.jpg"));

jLabel7.setIcon(new ImageIcon(rd1));

jButton2.setEnabled(true);

}

catch(Exception e)

{

e.printStackTrace();

}

}//GEN-LAST:event\_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton2ActionPerformed

// TODO add your handling code here:

ExtractFrame ef=new ExtractFrame();

ef.setVisible(true);

ef.setResizable(false);

ef.setTitle("Extract");

}//GEN-LAST:event\_jButton2ActionPerformed

public BufferedImage toBufferedImage(final Image image, final int type)

{

if (image instanceof BufferedImage)

return (BufferedImage) image;

if (image instanceof VolatileImage)

return ((VolatileImage) image).getSnapshot();

loadImage(image);

final BufferedImage buffImg = new BufferedImage(image.getWidth(null), image.getHeight(null), type);

final Graphics2D g2 = buffImg.createGraphics();

g2.drawImage(image, null, null);

g2.dispose();

return buffImg;

}

private void loadImage(final Image image)

{

class StatusObserver implements ImageObserver

{

boolean imageLoaded = false;

public boolean imageUpdate(final Image img, final int infoflags,

final int x, final int y, final int width, final int height)

{

if (infoflags == ALLBITS)

{

synchronized (this)

{

imageLoaded = true;

notify();

}

return true;

}

return false;

}

}

final StatusObserver imageStatus = new StatusObserver();

synchronized (imageStatus) {

if (image.getWidth(imageStatus) == -1 || image.getHeight(imageStatus) == -1)

{

while (!imageStatus.imageLoaded)

{

try

{

imageStatus.wait();

} catch (InterruptedException ex) {}

}

}

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(QRFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(QRFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(QRFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(QRFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new QRFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify//GEN-BEGIN:variables

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

public javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel4;

private javax.swing.JPanel jPanel5;

// End of variables declaration//GEN-END:variables

}

**REFERENCES**

1. Information Technology—Automatic Identification and Data Capture Techniques EAN/UPC Bar Code Symbology Specification, ISO/IEC Standard 15420:2009, 2009.
2. Z. Baharav and R. Kakarala, “Visually significant QR codes: Image blending and statistical analysis,” in Proc. IEEE Int. Conf. Multimedia Expo (ICME), Jul. 2013, pp. 1–6.
3. C. Baras and F. Cayre, “2D bar-codes for authentication: A security approach,” in Proc. 20th Eur. Signal Process. Conf. (EUSIPCO), Aug. 2012, pp. 1760–1766.
4. T. V. Bui, N. K. Vu, T. T. P. Nguyen, I. Echizen, and T. D. Nguyen, “Robust message hiding for QR code,” in Proc. IEEE 10th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process. (IIH-MSP), Aug. 2014, pp. 520–523.
5. A. T. P. Ho, B. A. M. Hoang, W. Sawaya, and P. Bas, “Document authentication using graphical codes: Reliable performance analysis and channel optimization,” EURASIP J. Inf. Secur., vol. 2014, no. 1, p. 9,2014
6. T. Langlotz and O. Bimber, “Unsynchronized 4D barcodes,” in Proc. 3rd Int. Symp., ISVC 2007, Lake Tahoe, NV, USA, Nov. 26–28, 2007, pp. 363–374.
7. C.-Y. Lin and S.-F. Chang, “Distortion modeling and invariant extraction for digital image print-and-scan process,” in Proc. Int. Symp. Multimedia Inf. Process., 1999, pp. 1–
8. P.-Y. Lin, Y.-H. Chen, E. J.-L. Lu, and P.-J. Chen, “Secret hiding mechanism using QR barcode,” in Proc. IEEE Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS), Dec. 2013, pp. 22–25.
9. J. Picard, “Digital authentication with copy-detection patterns,” Proc. SPIE, vol. 5310, pp. 176–183, Jun. 2004.
10. M. Querini, A. Grillo, A. Lentini, and G. F. Italiano, “2D color barcodes for mobile phones,” Int. J. Comput. Sci. Appl., vol. 8, no. 1, pp. 136–155, 2011.
11. M. Querini and G. F. Italiano, “Facial biometrics for 2D barcodes,” in Proc. IEEE Fed. Conf. Comput. Sci. Inf. Syst. (FedCSIS), Sep. 2012, pp. 755–762.
12. J. Rouillard, “Contextual QR codes,” in Proc. IEEE 3rd Int. Multi-Conf. Comput. Global Inf. Technol. (ICCGI), Jul./Aug. 2008, pp. 50–55.
13. B. Sklar, Digital Communications, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001.
14. K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, “‘Print and scan’ resilient data hiding in images,” IEEE Trans. Inf. Forensics Security, vol. 1, no. 4, pp. 464–478, Dec. 2006.
15. M. Sun, J. Si, and S. Zhang, “Research on embedding and extracting methods for digital watermarks applied to QR code images,” New Zealand J. Agricult. Res., vol. 50, no. 5, pp. 861–867, 2007.
16. I. Tkachenko, W. Puech, O. Strauss, J.-M. Gaudin, C. Destruel, and C. Guichard, “Fighting against forged documents by using textured image,” in Proc. 22th Eur. Signal Process. Conf. (EUSIPCO), Sep. 2014, pp. 790–794.
17. R. Ulichney, Digital Halftoning. Cambridge, MA, USA: MIT Press, 1987.
18. R. Villán, S. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun, “Tamper-proofing of electronic and printed text documents via robust hashing and data-hiding,” in Proc. SPIE, vol. 6505, p. 65051T, Feb. 2007.