**FEASIBILITY REPORT:**

**TECHNICAL FEASIBILITY:**

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical

analysis.

1. **Understand the different technologies involved in the proposed system:**

Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system.

1. **Find out whether the organization currently possesses the required technologies:**
   * Is the required technology available with the organization?
   * If so is the capacity sufficient?

For instance –

“Will the current printer be able to handle the new reports and forms required for the new system?”

**OPERATIONAL FEASIBILITY:**

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

* Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.
* Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.
* Have the user been involved in the planning and development of the project?
* Early involvement reduces the chances of resistance to the system and in
* General and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

**ECONOMIC FEASIBILITY:**

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

**SYSTEM REQUIREMENT SPECIFICATION**

**NUMBER OF MODULES:**

Modules:

1. Admin Module
2. Users Module
3. Image Water Marking Module
4. Security Authentication

**Admin Module:**

1. View registered users.

2. Delete registered usrs.

3. View Requested users.

4. Update status (or) Accept Requested users

5. View Inbox mails

6. View Inbox Water Mark Images.

7. View Outbox mails

8. View outboxes water mark images.

**Users Modules**

1. Delete.read and download water mark images with private key.

2. Image Water markng with text.

3. Image water marking with another image.

4. Watermarking with image type file to any type of other files (video, audio, txt, image, pdf etc).

5. Send these watermark attachments to registered users.

6. File encryption and decryption with user’s private keys and send mails to these files to registered users.

7. Composing mails.

8. Chating Assistence with remaing users.

**Image Water Marking:**

Rich user interface developed in order to select the image and select the watermarking settings based on the application. It also helps to view the extracted watermark from that electronic commerce application.

**Text Watermarking:**

Many paper documents (e.g., contracts, wills, etc.) are more valuable than multimedia like sound clips and images. Digital libraries and archives distribute copyrighted articles, journals, and books in electronic form. Watermarking of text documents provides a means of tracing documents that have been illegally copied, distributed, altered, or forged. Raw text, such as an ASCII text file or computer source code, cannot be watermarked because there is no “perceptual headroom” in which to embed hiddeninformation. However, final versions of documents are typically formatted (e.g., PostScript, PDF, RTF), and it is possible to hide a watermark in the layout information (e.g., word and line spacings) and formatting (e.g., serifs). Although *optical character recognition* (OCR) can theoretically remove any layout information, OCR is expensive, imperfect, and often requires manual supervision.

Brassil *et al.* [20–22] have investigated text watermarking and proposed varietyof methods for embedding hidden messages in PostScript documents. The work ofBrassil *et al.* currently does not use SS embedding, but it could be added to thesystem to strengthen robustness and security.

In [20–22], the message is embedded by altering different parts of the document.Line shifting moves entire lines of text up or down by a small amount, typically1/150 or 1/300 inch (0.170 or 0.085 mm). Similarly, word shifting may horizontallyshift individual words or blocks of words; words at the ends of a line are not shiftedto preserve justification. Figure 4 provides an example of word shifting. Finally, feature coding modifies small parts of characters themselves Recovery of the message from a printed or photocopied document requires a numberof post-processing steps (scanning, skew correction, and noise removal). Afterpost-processing, the message receiver automatically measures line shifts, wordshifts, and/or feature alterations to detect the message.In experiments, these methods have shown promise. Line shifts could be correctlydetected even after photocopying ten times. Word shifts on a single page werecorrectly detected 75 percent of the time, after photocopying four times or afterfax transmission. With simple ECC, 26–30 of 30 embedded message bits per pagecould be decoded, depending upon the amount of degradation (e.g., photocopyingmultiple times).

**Image Watermarking:**

Digital images can be produced from many sources, such as everyday photographs, satellite pictures, medical scans, or computer graphics. Watermarks for natural images typically modify pixel intensities or transform coefficients, although it is conceivable that a watermark could alter other features such as edges or textures. An image may be viewed for an extended period of time, and it may also be subjectto a great deal of manipulation, such as filtering, cropping, geometric transformations, compression, and compositing with other images, and hostile attacks. Thus, imperceptibility, robustness, and security are usually the most important propertiesof image watermarks; speed and complexity are often secondary. Also, since manyimages are compressed (e.g., JPEG or GIF), watermarking algorithms that operatein the transform or wavelet domain may be useful. One potential difficulty in image watermarking is the finite bandwidth available.As the image size decreases, the permissible message length decreases unless E isincreased (weakening imperceptibility) or N is decreased (weakening robustness).The example watermarking system in Figures 2 and 3 has been directly developedinto a system that embeds a spatial-domain DSSS image watermark. Figure5 compares an original, unmarked image with its watermarked counterpart. The two images are indistinguishable perceptually. The original image is a 256 \_ 2568-bit grayscale image. The watermark was embedded a chip rate of N = 4096and per-pixel amplitude qE=N = 2. Hence, the total power per message bit wasE = 16384.

The marked image was then subjected to a number of different attacks. Examplemarked images after attack appear in Figure 6. The watermark message remainedrecoverable after addition of Gaussian noise with variance 400, a sinusoidal patternwith amplitude 30, and a constant offset of 30. The watermark also survived JPEGcompression with a quality factor of 20 percent (compressed to 4959 bytes, or 13:1Compression).A free software program called StirMark [19,23] is available for testing watermarkrobustness. StirMark simulates printing and rescanning of an image, and its producersclaim that it can defeat several commercial watermarking systems. The SSwatermark, however, survived StirMark and the embedded message was recoveredwithout any bit errors.

**Insertion of Watermark Image into Cover Work:**

**Algorithm Explanation**

The aim of the program is to replace the LSB of the base image with the MSB of the watermark. First, the program asks the user for the images to be read. The user will then enter the name of the images, both the base and the watermark, with their extension. Both these images will be read. . The program will then change the image size to double. This is done so as inform to provide double data-type space for the images. The reason for doing so, is to provide decimal storage for the subsequent additional operations which will be performed on the base and watermark image

The next step is to assign the number of most significant bits of the watermark which will be used to overwrite on the least significant bit spaces of the base signal. Once the user provides this, the watermark signal bits are shifted to the right by the specified bits.  An exact number of these bits will be made zero in the LSB of the base image. The reason for making them zero, is to provide space for the MSB of watermark to be stored.  The bits of both the images, i.e. watermark image which now has its MSB shifted to its LSB, and the base image, which has its LSBs are set zero, are added. The added image gives us the watermarked signal. o The final image obtained.  

**Results:**

** **



Watermarked Image , using bits=1. Since it is an invisible watermarking scheme, only the base image is visible.

**RSA:**

The RSA algorithm was publicly described in 1977 by [Ron Rivest](http://en.wikipedia.org/wiki/Ron_Rivest), [Adi Shamir](http://en.wikipedia.org/wiki/Adi_Shamir), and [Leonard Adleman](http://en.wikipedia.org/wiki/Leonard_Adleman)

The RSA algorithm involves three steps:

[key](http://en.wikipedia.org/wiki/Key_%28cryptography%29) generation, encryption and decryption.

### Key generation

RSA involves a **public key** and a [private key](http://en.wikipedia.org/wiki/Private_key)**.**

The public key can be known to everyone and is used for encrypting messages.

Messages encrypted with the public key can only be decrypted using the private key.

The keys for the RSA algorithm are generated the following way:

1. Choose two distinct [prime numbers](http://en.wikipedia.org/wiki/Prime_number) *p* and *q*.
   * For security purposes, the integers *p* and *q* should be chosen at random, and should be of similar bit-length.
2. Compute *n* = *pq*.
   * *n* is used as the [modulus](http://en.wikipedia.org/wiki/Modular_arithmetic) for both the public and private keys. Its length, usually expressed in bits, is the [key length](http://en.wikipedia.org/wiki/Key_length).
3. Compute φ(*n*) = (*p* – 1)(*q* – 1), where φ is [Euler's totient function](http://en.wikipedia.org/wiki/Euler%27s_totient_function).
4. Choose an integer *e* such that 1 < *e* < φ(*n*) and [greatest common divisor](http://en.wikipedia.org/wiki/Greatest_common_divisor) gcd(*e*, φ(*n*)) = 1; i.e., *e* and φ(*n*) are [coprime](http://en.wikipedia.org/wiki/Coprime).
   * *e* is released as the public key exponent.
   * *e* having a short [bit-length](http://en.wikipedia.org/wiki/Bit-length) and small [Hamming weight](http://en.wikipedia.org/wiki/Hamming_weight) results in more efficient encryption – most commonly 216 + 1 = 65,537. However, much smaller values of *e* (such as 3) have been shown to be less secure in some settings.[[4]](http://en.wikipedia.org/wiki/RSA_%28algorithm%29#cite_note-Boneh-4)
5. Determine *d* as *d* ≡ *e*−1 (mod φ(*n*)), i.e., *d* is the [multiplicative inverse](http://en.wikipedia.org/wiki/Modular_multiplicative_inverse) of *e* (modulo φ(*n*)).

* This is more clearly stated as solve for *d* given *de* ≡ 1 (mod φ(*n*))
* This is often computed using the [extended Euclidean algorithm](http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm).
* *d* is kept as the private key exponent.

By construction, *d*⋅*e* ≡ 1 (mod φ(*n*)). The **public key** consists of the modulus *n* and the public (or encryption) exponent *e*. The **private key** consists of the modulus *n* and the private (or decryption) exponent *d*, which must be kept secret. *p*, *q*, and φ(*n*) must also be kept secret because they can be used to calculate *d*.

* An alternative, used by [PKCS#1](http://en.wikipedia.org/wiki/PKCS1), is to choose *d* matching *de* ≡ 1 (mod λ) with λ = lcm(*p* − 1, *q* − 1), where lcm is the [least common multiple](http://en.wikipedia.org/wiki/Least_common_multiple). Using λ instead of φ(*n*) allows more choices for *d*. λ can also be defined using the [Carmichael function](http://en.wikipedia.org/wiki/Carmichael_function), λ(*n*).
* The [ANSI X9.31](http://en.wikipedia.org/w/index.php?title=ANSI_X9.31&action=edit&redlink=1) standard prescribes, [IEEE 1363](http://en.wikipedia.org/wiki/P1363) describes, and [PKCS#1](http://en.wikipedia.org/wiki/PKCS1) allows, that *p* and *q* match additional requirements: being [strong primes](http://en.wikipedia.org/wiki/Strong_prime), and being different enough that [Fermat factorization](http://en.wikipedia.org/wiki/Fermat_factorization) fails.

### Encryption

[Alice](http://en.wikipedia.org/wiki/Alice_and_Bob) transmits her public key (*n*, *e*) to [Bob](http://en.wikipedia.org/wiki/Alice_and_Bob) and keeps the private key secret. Bob then wishes to send message *M* to Alice.

He first turns *M* into an integer *m*, such that 0 ≤ *m* < *n* by using an agreed-upon reversible protocol known as a [padding scheme](http://en.wikipedia.org/wiki/RSA_%28algorithm%29#Padding_schemes). He then computes the ciphertext *c* corresponding to

 c \equiv m^e \pmod{n} .

This can be done quickly using the method of [exponentiation by squaring](http://en.wikipedia.org/wiki/Exponentiation_by_squaring). Bob then transmits *c* to Alice.

### Decryption

Alice can recover *m* from *c* by using her private key exponent *d* via computing

 m \equiv c^d \pmod{n} .

Given *m*, she can recover the original message *M* by reversing the padding scheme.

(In practice, there are more efficient methods of calculating *cd* using the precomputed values below.)

### A working example

Here is an example of RSA encryption and decryption. The parameters used here are artificially small, but one can also [use OpenSSL to generate and examine a real keypair](http://en.wikibooks.org/wiki/Transwiki:Generate_a_keypair_using_OpenSSL).

1. Choose two distinct prime numbers, such as

p = 61and q = 53.

1. Compute *n* = *pq* giving

n = 61 \times 53 = 3233.

1. Compute the [totient](http://en.wikipedia.org/wiki/Totient) of the product as φ(*n*) = (*p* − 1)(*q* − 1) giving

\varphi(3233) = (61 - 1)(53 - 1) = 3120.

1. Choose any number 1 < *e* < 3120 that is [coprime](http://en.wikipedia.org/wiki/Coprime) to 3120. Choosing a prime number for *e* leaves us only to check that *e* is not a divisor of 3120.

Let e = 17.

1. Compute *d*, the [modular multiplicative inverse](http://en.wikipedia.org/wiki/Modular_multiplicative_inverse) of *e* (mod φ(*n*)) yielding

d = 2753.

The **public key** is (*n* = 3233, *e* = 17). For a padded [plaintext](http://en.wikipedia.org/wiki/Plaintext) message *m*, the encryption function is *m*17 (mod 3233).

The **private key** is (*n* = 3233, *d* = 2753). For an encrypted [ciphertext](http://en.wikipedia.org/wiki/Ciphertext) *c*, the decryption function is *c*2753 (mod 3233).

For instance, in order to encrypt *m* = 65, we calculate

c \equiv 65^{17} \equiv 2790 \pmod{3233} .

To decrypt *c* = 2790, we calculate

m \equiv 2790^{2753} \equiv 65 \pmod{3233}.

Both of these calculations can be computed efficiently using the [square-and-multiply algorithm](http://en.wikipedia.org/wiki/Square-and-multiply_algorithm) for [modular exponentiation](http://en.wikipedia.org/wiki/Modular_exponentiation). In real life situations the primes selected would be much larger; in our example it would be relatively trivial to factor *n*, 3233, obtained from the freely available public key back to the primes *p* and *q*. Given *e*, also from the public key, we could then compute *d* and so acquire the private key.

Practical implementations use the [Chinese remainder theorem](http://en.wikipedia.org/wiki/Chinese_remainder_theorem) to speed up the calculation using modulus of factors (mod *pq* using mod *p* and mod *q*).

The values *dp*, *dq* and *q*inv, which are part of the private key are computed as follows:

* d_p = d\text{ (mod }(p-1)\text{)} = 2753 \text{ (mod } (61-1)\text{)} = 53
* d_q = d\text{ (mod }(q-1)\text{)} = 2753 \text{ (mod } (53-1)\text{)} = 49
* q_\text{inv} = q^{-1} \text{ (mod } p\text{)} = 53^{-1} \text{ (mod } 61\text{)} = 38(Hence: q_\text{inv} \times q \text{ (mod } p\text{)} = 38 \times 53 \text{ (mod } 61\text{)} = 1)

Here is how *dp*, *dq* and *q*inv are used for efficient decryption. (Encryption is efficient by choice of public exponent *e*)

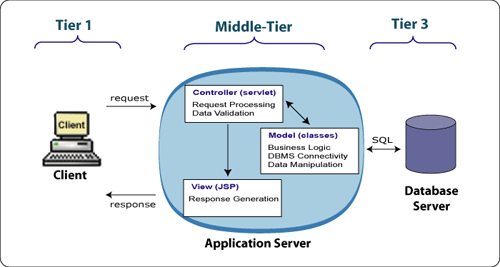
* m_1 = c^{d_p} \text{ (mod } p\text{)} = 2790^{53} \text{ (mod } 61\text{)} = 4
* m_2 = c^{d_q} \text{ (mod } q\text{)} = 2790^{49} \text{ (mod } 53\text{)} = 12
* h = (q_{Inv} \times (m_1 - m_2)) \text{ (mod } p\text{)} = (38 \times -8) \text{ (mod } 61\text{)} = 1
* m = m_2 + h \times q = 12 + 1 \times 53 = 65(same as above but computed more efficiently)

Securiy Authentication:

1. Registeration
2. Login
3. View profile
4. Update Profies
5. Change passwords
6. Logout
7. RecoverPassword

**PROCESS FLOW:**

**ARCHITECTURE DIAGRAM:**



1. **THE PRESENTATION LAYER**

Also called as the client layer comprises of components that are dedicated to presenting the data to the user. For example: Windows/Web Forms and buttons, edit boxes, Text boxes, labels, grids, etc.

1. **THE BUSINESS RULES LAYER**

This layer encapsulates the Business rules or the business logic of the encapsulations. To have a separate layer for business logic is of a great advantage. This is because any changes in Business Rules can be easily handled in this layer. As long as the interface between the layers remains the same, any changes to the functionality/processing logic in this layer can be made without impacting the others. A lot of client-server apps failed to implement successfully as changing the business logic was a painful process

1. **THE DATA ACCESS LAYER**

This layer comprises of components that help in accessing the Database. If used in the right way, this layer provides a level of abstraction for the database structures. Simply put changes made to the database, tables, etc do not affect the rest of the application because of the Data Access layer. The different application layers send the data requests to this layer and receive the response from this layer.

1. **THE DATABASE LAYER**

This layer comprises of the Database Components such as DB Files, Tables, Views, etc. The Actual database could be created using SQL Server, Oracle, Flat files, etc.   
In an n-tier application, the entire application can be implemented in such a way that it is independent of the actual Database. For instance, you could change the Database Location with minimal changes to Data Access Layer. The rest of the Application should remain unaffected.

**SDLC METHODOLOGIES**

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, “A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The steps for Spiral Model can be generalized as follows:

* The new system requirements are defined in as much details as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
* A preliminary design is created for the new system.
* A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
* A second prototype is evolved by a fourfold procedure:

1. Evaluating the first prototype in terms of its strengths, weakness, and risks.
2. Defining the requirements of the second prototype.
3. Planning an designing the second prototype.
4. Constructing and testing the second prototype.

* At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involved development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer’s judgment, result in a less-than-satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

**The following diagram shows how a spiral model acts like:**



**Fig 1.0-Spiral Model**

**ADVANTAGES**

* Estimates(i.e. budget, schedule etc .) become more relistic as work progresses, because important issues discoved earlier.
* It is more able to cope with the changes that are software development generally entails.
* Software engineers can get their hands in and start woring on the core of a project earlier.

**SOFTWARE REQUIREMENT AND**

**HARDWARE REQUIREMENT**

**SOFTWARE REQUIREMENTS:**

Operating System : Windows XP/2003 or Linux

User Interface : HTML, CSS

Client-side Scripting : JavaScript

Programming Language : Java

Web Applications : JDBC, Servlets, JSP

IDE/Workbench : My Eclipse 6.0

Database : Oracle 10g

Server Deployment : Tomcat 6.x

**HARDWARE REQUIREMENTS**

Processor : core 2 duo

Hard Disk : 160GB

RAM : 1GB or more