CHAPTER IV
ANALYSIS AND DESIGN

4.1 Analysis

4.1.1 The processes

1. Encryption

At this stage, the text is converted to binary. Then the result is padded to 8 bits binary. After that, the image is converted to binary and padded to 8 bits binary. To make the message more secure, the channel that being used is only red. Get the red channel by taking the 8^{th}-15^{th} RGB bits then convert it to hexadecimal and binary. This will increase the security of the data because usually the encryption use all color channel in RGB.

To encrypt the text binary inside the image binary, separate the 8 bits binary of the text and place bit by bit on the last bit of the image. One character needs 8 pixels. Loop through image binary until the text is all placed.

2. Decryption

The encrypted image is converted to binary and the last bit of every pixel is taken one by one. Then the bits taken is joined to one array of string. To get the hidden message, separate the string by cut it every 8 bits then convert it to ASCII characters.

4.2 Design

4.2.1 Use Case Diagram

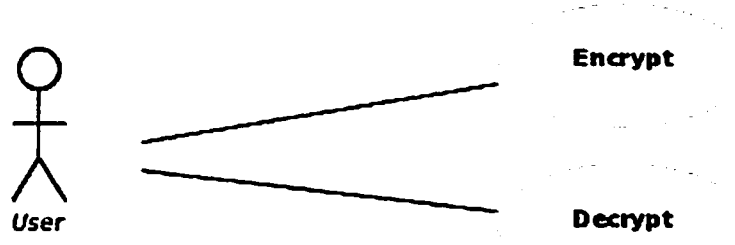

Figure 4.1 Use Case Diagram

On the above diagram shows that user can choose to do two works on this application. First is encryption, second is decryption.

4.2.2 Flowchart

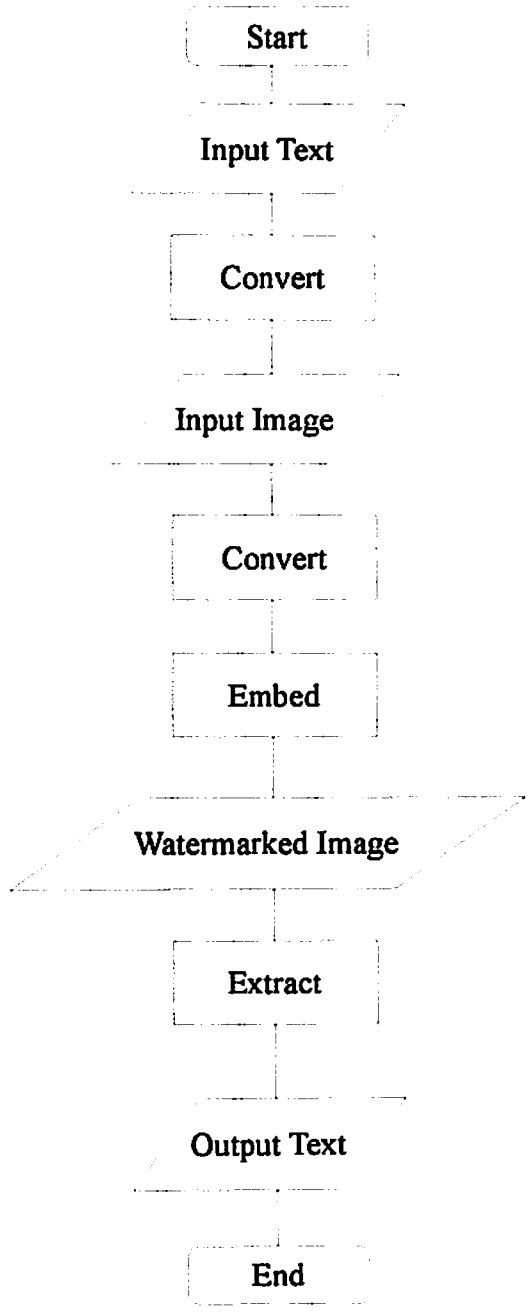

Figure 4.2 Flowchart

This simulation starts with encryption where the text is converted to binary and then the image is converted to binary. Then the image binary last bit is replaced by the text binary.

Decryption starts with conversion of the encrypted image to binary and extraction of the last bit of every pixel. Then the extraction is converted to text.

4.2.3 Class Diagram

Figure 4.3 Class Diagram

Embed class consists of input text with type of String, input image with type of String, the class of convertText(), convertImage(), and embed().

Extract class consists of extract() class.