**UNIVERSITI TUNKU ABDUL RAHMAN**

# REPORT STATUS DECLARATION FORM

**Title**: ________________________________________________

________________________________________________

________________________________________________

**Academic Session**: _____________

I ________________________________________________

**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________

(Author’s signature)

**Address**:

_________________________

_________________________

_________________________

**Date**: _____________________

_________________________

(Supervisor’s signature)

_________________________

Supervisor’s name

**Date**: ____________________

# AUTHENTICATION TOOL FOR PICTURE BASED PASSWORDS

BY

GOH WEN BIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS) COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2014

# DECLARATION OF ORIGINALITY

I declare that this report entitled "**Authentication Tool for Picture Based Passwords**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ____________________

Name : ____________________

Date : ____________________

# ACKNOWLEDGEMENTS

I would like to take this opportunity to offer my special thanks to my supervisor, Mr. Sohail Safdar who has been very patient and helpful to me throughout this project. His guidance and encouragement have given me the motivation and confidence to overcome the issues that I had encountered in this project.

Next I would also like to express my appreciation to my moderator, Mr. Koon Kim Peh for his valuable suggestions and feedbacks for this project. His advice and assistance have helped a lot in identifying and correcting the problems in my project report.

Lastly, I would like to thank to my family who are being very supportive all the time. Without their support, I will not be able to achieve this far.

Thank you.

# ABSTRACTS

Information security has been a prime concern as the technology advances. Various authentication mechanisms are in place for ensuring the authorized access to the information. One of the most commonly used authentication schemes is based on textual passwords. There are various limitations associated to the textual passwords. To overcome these limitations and strengthening the authentication more, picture-based passwords are introduced. Picture-based password involves using of images as password. This type of password is easier to memorize as picture is easier to remember compared to words according to a paper written by Angeli (n.d., pp. 3-4). Unfortunately, most of the existing picture-based passwords suffer from shoulder surfing attacks as most of the picture-based password schemes do not contain anti-shoulder surfing mechanisms. The main objective of this project is to devise and develop a new picture-based authentication algorithm that consists of anti-shoulder surfing mechanism. The proposed algorithm is then implemented in web-based environment. The proposed picture-based password scheme consists of password creation and password application. The user can select from the predefined image objects to create a picture-based password. The object selection must be categorized in three ways during password creation that is defined as Pass Object, Flag Object and Skipping Object. For authentication, the created password is applied using a specialized mechanism that is composed of multiple steps. Each step is controlled by defined rule based on the categorization of objects registered for that user. The methodology used for this project is evolutionary prototyping and the web-based application is developed using PHP, HTML5, JavaScript and WampServer.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| CAPTCHA | Completely Automated Public Turing Test To Tell Computers and Humans Apart |

# 1. INTRODUCTION

This project involves the development of a new picture-based password scheme with a new algorithm. The criteria and features of the system will be based on the analysis of multiple existing graphical password schemes. The newly developed graphical authentication system will consist of anti-shoulder surfing mechanism which will prevent shoulder surfing attack towards the picture-based password scheme.

## 1.1 Motivation and Problem Statement

There are many techniques have been introduced to create a strong textual password. However, most of the methods are not practiced by users as strong textual password is complex and difficult to remember. Therefore another type of password is created which is the picture-based password also known as graphical password. Picture-based password is easier to remember but most of the picture-based password schemes are vulnerable to shoulder surfing attacks due to the lack of anti-shoulder surfing mechanism.

Currently there are some proposed picture-based password schemes which include anti-shoulder surfing mechanism such as the Triangle Scheme and Intersection Scheme. However, these schemes require up to 1000 image objects in each attempt in order to be secure. As a result, it is time consuming and inconvenient as user will need to locate his Pass Object out of many other image objects. In this project, a new picture-based password scheme with anti-shoulder surfing mechanism will be developed. This password scheme only requires 12 image objects in each attempt and also implements new methods called the Flag Object approach and Skipping Object approach.

## 1.2 Project Scope

This project will develop a picture-based authentication system which is a web based application. The type of algorithm used during the authentication process is a newly developed algorithm based on the research of multiple existing graphical password schemes. The picture-based authentication system also includes anti-shoulder surfing mechanism which prevents shoulder surfer from guessing the user’s password. In addition, the system will provide an optimal level of security and requires lesser effort for the user to remember his password. On top of that, the system developed will only require minimal of time for the user to finish the authentication process.

## 1.3 Project Objectives

The main objective of this project is to develop a web based graphical authentication system which is secure from shoulder surfing attack. This can be achieved by the following sub-objectives.

1. To investigate and analyse the existing graphical password schemes which include both schemes with and without anti-shoulder surfing mechanism.
2. Devise a new algorithm based on conducted analysis to avoid shoulder surfing attack.
3. Implement the algorithm using web-based technology.
4. To test and improve the developed system to ensure that it provides an optimal level of security and usability.

## 1.4 Project Significance and Contribution

This project is mainly focused on developing a new picture-based authentication system. The existing textual password schemes are found to be vulnerable to various attacks such as key logging attacks and dictionary attacks. Besides that, a strong textual password is hard to be remembered as it is complex. As for the existing picture-based password schemes, majority of the schemes do not consist of anti-shoulder surfing mechanism. Even though some graphical password schemes consists of anti-shoulder surfing mechanism, the processes involved are time consuming and inconvenient to users. Therefore, a new picture-based password scheme will be developed in this project in order to solve these issues.

In this newly developed picture-based password scheme, it is secure from other attacks such as key logging attacks and dictionary attacks. Besides, since it mainly involves of pictures, user will be able to remember his password easier compared to complicated textual password. In addition, compared to other existing graphical password schemes which have anti-shoulder surfing mechanism, this newly developed system only requires 12 image objects per attempt instead of 1000 of them. The number of image objects during each attempt is crucial as it will greatly affect the duration of the authentication process. User may not want to spend too much of time during the authentication process.

## 1.5 Background Information

In modern times, information can be easily obtained through the internet. In general, information can be classified into two types which are information which is meant to be accessed by public and information which is only meant to be accessible by authorized people. The value of the private information can range from low to very high depending of what type content being stored. An example of valuable private information would be a company financial data. In order to protect the private information from being accessed by unauthorized people, the most commonly method used nowadays is the textual password.

There are many methods used to strengthen the textual password such as changing the password regularly and using alphanumeric password. However, according to Adams and Sasse (1994, p. 42), since users may need to have multiple passwords for more than one applications, users tend to perform insecure work practices for example writing down password and using simple password such as "password". As a result, the role of textual password has failed as those insecure work practices have greatly reduce the strength of the textual password and vulnerable to most computer attacks. The main factor which causes such phenomena is due to the difficulty in memorizing a complicated textual password.

To counter such problems, another method introduced to secure information is the picture-based password which is also known as graphical password. Based on the paper done by Angeli (n.d., pp. 3-4), picture is easier to remember compared to words. Besides that, since picture is not a form of any word or sentences, it is secure from dictionary attack which is one of the commonly used techniques to crack a textual password.

Although picture-based password is memorable, most of the picture-based password does not include anti-shoulder surfing mechanisms. As a result, attacker is able to guess the user's password by shoulder-surfing or by recording the user's authentication process.

In this paper, both existing graphical authentication techniques with and without anti-shoulder surfing mechanisms will be investigated based on their usability and security in order to produce a new graphical authentication scheme.

The newly developed graphical authentication scheme requires user to register five image objects which include three Pass Objects, a Skipping Object and a Flag Object. The Pass Object basically acts as the user's password while Skipping Object and Flag Object act as informative objects which will provide instruction to the user. Generally for this scheme, user is required to go through three attempts and also needs to select six image objects in each attempt. Each password screen is a square panel which consists of 12 image objects. In each attempt, the password screen will have different image objects and arrangement based on certain display criteria. The Skipping Object, it acts as a noise which will greatly reduce the chance of shoulder surfer from guessing the user's Pass Objects. As for the Flag Object, it will determine whether the user should select his Pass Object or not in each attempt. This method is also used to prevent shoulder surfer from guessing the user's Pass Object easily.

## 2. LITERATURE REVIEW

Currently, textual password is the most commonly used technique for user authentication. Textual password usually works together with a username whereby the users are required to enter both of their username and password. The username could be in a form of the users' email address and any other names. There are many guidelines on how to create a strong textual password such as creating an alphanumeric password, prevent from using dictionary words and so on. However, due to the complexity of the strong password, users tend to perform some insecure practices such as writing down their passwords or using a simpler password instead Adams and Sasse (1994, p42). Besides that, they also mentioned that users will only change their password if they know that their passwords are cracked.

According to Kotadia (2005), he stated it is meaningless to ban users from writing down their password as they will tend to use a simpler password in the end. Based on this statement, it clearly shows that the strength of the password is linked to the rate of difficulty to memorize a password. A strong password will be a difficult password for the users to remember. In addition, in real life, users are required to remember more than one password for each of their applications which also increase their difficulty in memorizing all of the passwords.

Due to the issues of textual password, another technique is introduced which is the picture-based password. According to Angeli (n.d., pp. 3-4), it is easier for people to remember picture than a plain text. Picture-based password which is also known as graphical password is an authentication mechanism which uses images as password.

Graphical password is secure from dictionary attack as there is no dictionary which stores graphical information (Sobrado & Birget, 2002).

## 2.1 Graphical Password Schemes

Graphical password schemes are password which may include pictures and patterns. The followings are explanations on some graphical password schemes.

### 2.1.1 Triangle Scheme

This scheme is proposed by Sobrado and Birget (2002) which is used to solve the shoulder surfing issue. In this scheme, the user will need to identify three Pass Objects on a password screen which consists of N numbers of objects. The user needs to click within the triangle area formed by the three Pass Objects during the authentication process. This process will be repeated few times with different images and those objects will be reassigned to a new position.

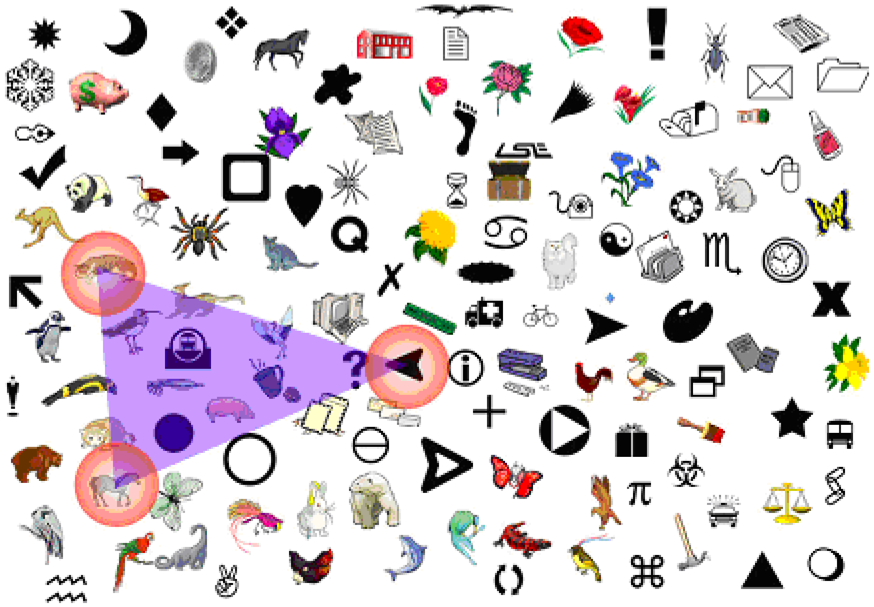


**Figure 2-1-1:** Triangle Scheme (Sobrado & Birget, 2002)

### 2.1.2 Intersection Scheme

Another scheme proposed by Sobrado and Birget (2002) to solve the shoulder surfing problem is the intersection scheme. In this scheme, the user needs to identify four Pass Objects on a password screen which consists of N numbers of objects. The user will need to click on the area of the point of intersection formed by those four Pass Objects. This process will be repeated few times with different images and those objects will be assigned a new position.

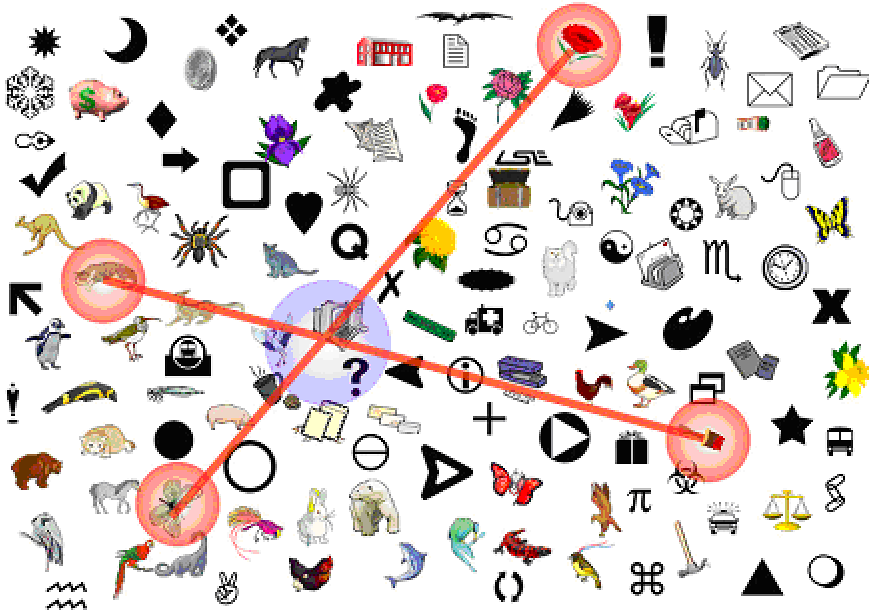


**Figure 2-1-2:** Intersection Scheme (Sobrado & Birget, 2002)

### 2.1.3 Movable Frame Scheme

The next is known as the movable frame scheme which is also proposed by Sobrado and Birget (2002) which is resistance to shoulder surfing. For this scheme, the user needs to identify three Pass Objects whereby two Pass Objects will be on a static frame while another one will be located on a movable frame surrounding the static frame. The user is required to move the frame until the object on the frame lines up with the other two predefined objects which are at the static frame. N numbers of random images will also be included in both frames. This process will be repeated few times with different images and those objects will be assigned a new position.

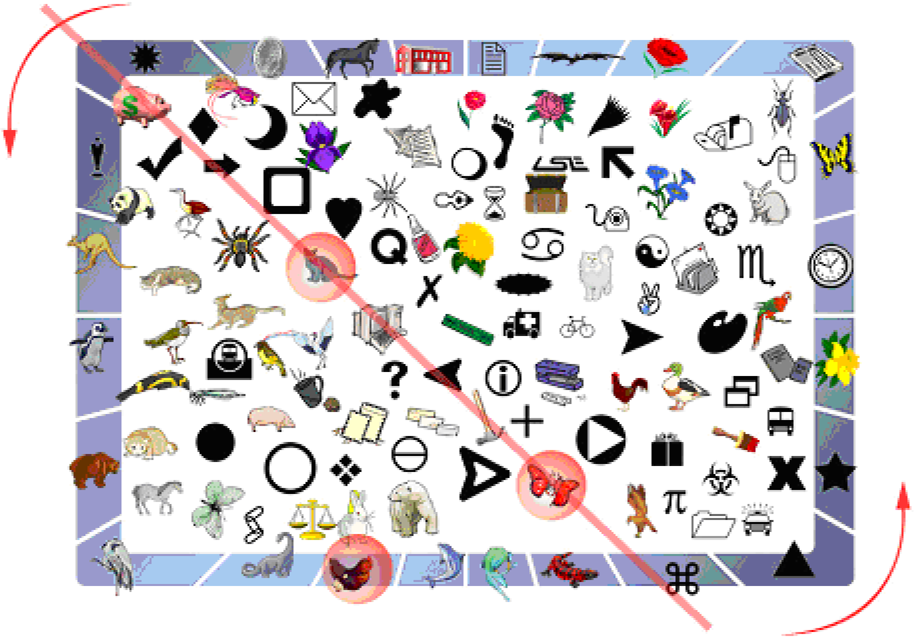


**Figure 2-1-3:** Movable Frame Scheme (Sobrado & Birget, 2002)

### 2.1.4 Passfaces™

It is a commercial product by Passfaces Corporation which requires user to select previously seen human faces picture as password. During the login process, the user is required to choose the correct face out of other faces to login.

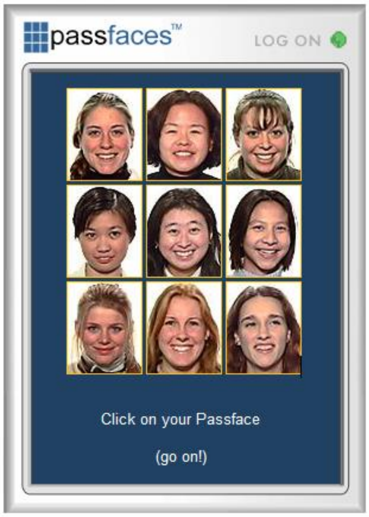
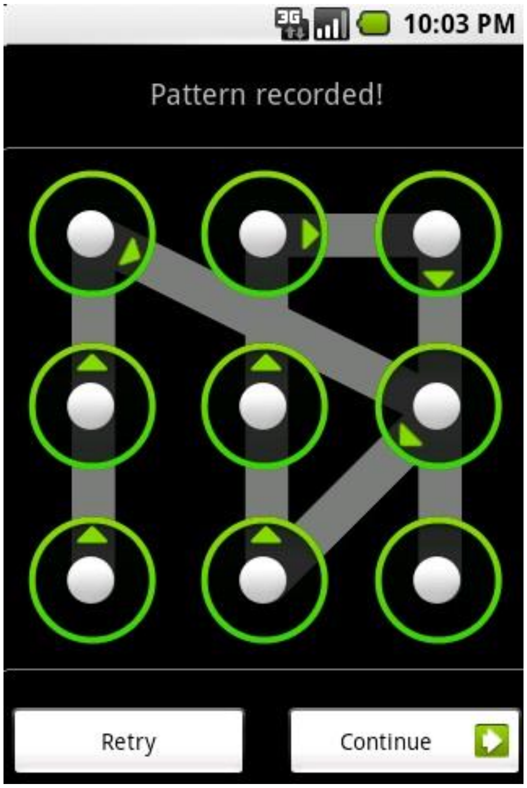


**Figure 2-1-4:** Passfaces$^{TM}$ (passfaces$^{TM}$, 2006)

### 2.1.5 Pattern Lock

This scheme can be mainly found on smart phones whereby user is required to form a pattern out of nine points as his password. When the pattern is registered, each time when user logs in, he is required to form the same pattern on the log in screen which consists of nine points.



Pattern recorded!

**Figure 2-1-5:** Pattern Lock (freeanimationhub, 2012)

## 2.2 Compare and Contrast of the Graphical Password Schemes

According to Sobrado and Birget (2002), the triangle scheme and the intersection scheme is hard to be attacked using the brute-force attack when there are 1000 objects and 10 pass-objects. However, based on the paper written by Muhammad Daniel Hafiz et al. (2008, pp. 397-398), it is stated that such method will increase the difficulty for the user to detect their own pass-objects since the user is required to find them out of 1000 objects. (Lai 2009, pp. 23-24) stated that reducing the number of objects while increasing the size of each objects at the same time will shorter the time required for the user to detect their pass-objects. On the other hand, Muhammad Daniel Hafiz et al. (2008, p. 398), stated that these two schemes will be easier to crack if the numbers of objects are reduced.

(Lai 2009, pp. 22-24) wrote that for the triangle scheme, if those pass-objects are distributed in a corner of the screen, the convex hull of the triangle formed by the three pass-objects will be very small and the password can be guessed easier. Other than that, if those pass-objects are distributed further from each other, this will form a very big convex hull of triangle which increases the chance of random guessing. In order to prevent such issues, Sobrado and Birget (2002) suggested that the triangle scheme, intersection scheme and movable frame scheme require user to repeat the login process for example 10 times per successful login. For each login process, the objects will be placed randomly and independently from the previous display.

An experiment has been conducted by (Lim 2011) shows that the anti-shoulder surfing mechanism of triangle scheme and intersection scheme are not reusable on mobile devices as they are vulnerable to shoulder-surfing attack. The reason is due to the size of the display of a mobile device which is much smaller than a general device and impossible to fit in 1000 objects. The major factor that causes these two schemes to be vulnerable to shoulder-surfing attack is the limited number of objects display on the mobile device which enables shoulder-surfer to guess the password easier or performing random guessing. However, in this experiment, the user is required to perform the login process for only once unlike the method suggested by Sobrado and Birget which is 10 times. As a result, this experiment result is only applicable to mobile device which uses a single login attempt.

As for the movable frame scheme, Muhammad Daniel Hafiz et al. state that the method suggested by Sobrado and Birget which involves user to repeat the login process multiple times is unlikable, confusing and time consuming as the user require to detect his pass-objects out of many non-pass objects.

As for both Passfaces$^{TM}$ and pattern lock, both of them do not have anti-shoulder surfing mechanism and can be easily guessed by shoulder surfer.

In Table 2-1, it shows the comparison of multiple graphical password schemes based on several parameters such as resistance to shoulder surfing, memorability of password, easiness of login, vulnerability to dictionary attack or key loggers, usability in mobile devices and general devices. Each scheme will be ranked based on the scale below.

**Rank Scale: LOW** 1 --------- 3 **HIGH**

**Table 2-1:** The comparison of multiple graphical password schemes

| **Graphical Password Scheme** | **Anti-shoulder surfing** | **Memorability of password** | **Easiness of login** | **Dictionary attack** | **Key-logging attack** | **Usability in Mobile Device** | **Usability in General Device** |
|---|---|---|---|---|---|---|---|
| Triangle Scheme | Yes **(3)** | Yes, only 3 objects to remember **(3)** | Moderate, need to repeat N times **(2)** | No **(3)** | No **(3)** | No, more space required for more objects **(1)** | Yes **(3)** |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Intersection Scheme | Yes **(3)** | Yes, only 4 objects to remember **(3)** | Moderate, need to repeat N times **(2)** | No **(3)** | No **(3)** | No, more space required for more objects **(1)** | Yes **(3)** |
| Movable frame Scheme | Yes **(3)** | Yes, only 3 objects to remember **(3)** | Moderate, need to repeat N times **(2)** | No **(3)** | No **(3)** | Moderate, smaller object size on movable frame **(2)** | Yes **(3)** |
| Pattern Lock | No **(1)** | Depends on the pattern (1.5) | Yes **(3)** | No **(3)** | No **(3)** | Yes **(3)** | No **(1)** |
| Passfaces | No **(1)** | Yes, only 3 faces to remember **(3)** | Yes, repeat only 3 times **(3)** | No **(3)** | No **(3)** | Yes **(3)** | No **(1)** |

From Table 2-2, it shows that movable frame scheme has the highest score which is 19 followed by triangle scheme and intersection scheme which both scored 18. Lastly, Passfaces$^{TM}$ scored 17 while pattern lock obtained 15.5. Therefore, the movable frame scheme is currently the best scheme out of the five graphical password schemes.

**Table 2-2:** The ranking of the graphical password schemes

| **Rank** | **Graphical Password Scheme** | **Total Score** |
|---|---|---|
| 1 | Movable Frame Scheme | 19 |
| 2 | Triangle Scheme | 18 |
| 3 | Intersection Scheme | 18 |
| 4 | Passfaces$^{TM}$ | 17 |
| 5 | Pattern Lock | 15.5 |

# 3. METHODOLOGY AND TOOLS

## 3.1 Methodology

The methodology used in this project is the evolutionary prototyping model. This model will involve five stages as shown in Figure 3-1-1.

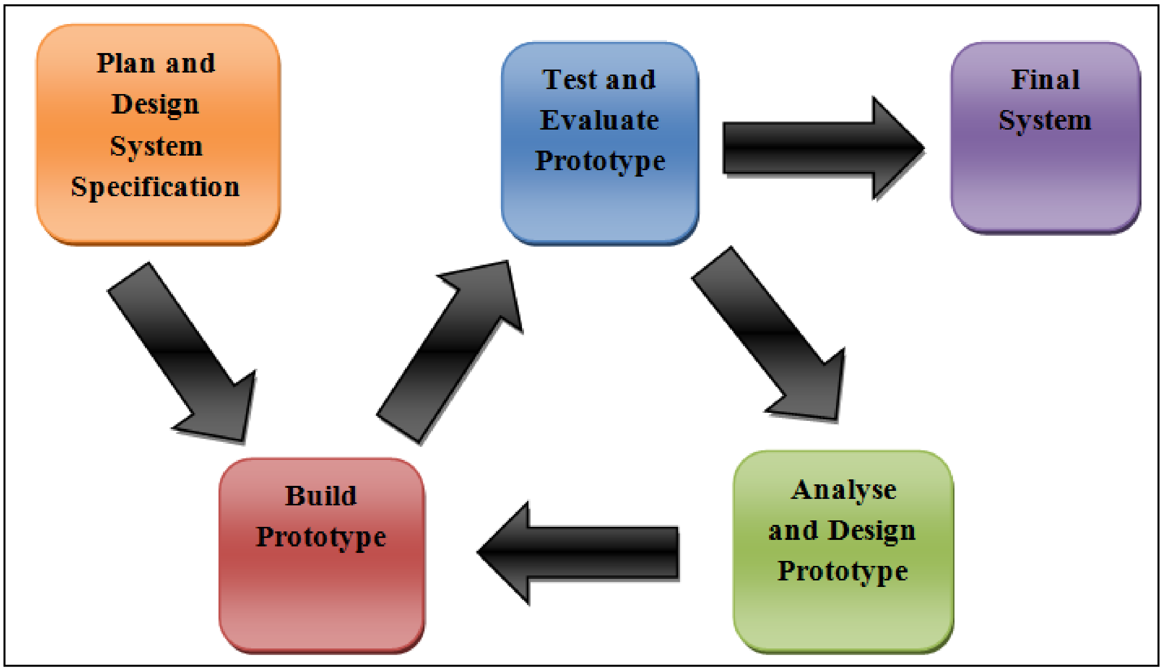

Plan and Design System Specification
Test and Evaluate Prototype
Final System
Build Prototype
Analyse and Design Prototype

**Figure 3-1-1:** Evolutionary Prototyping Model

The proposed picture-based password scheme is suitable for this methodology because via the prototype developed, a better understanding of the system can be achieved and any additional requirement can be added once it is identified in order to improvise the system. In addition, since the system is a picture-based password scheme, it will involve a lot of user interactions during the authentication process. Therefore, a prototype is needed so that user is able to interact with the actual system in order to gain more requirements.

### 3.1.1 Plan and Design System Specification

In the initial stage, the general approach of this project is through exploring the literature review to validate concepts. This method is conducted by reviewing multiple past research papers written by others regarding the graphical password schemes. Through this method, those schemes will be compared and contrasted in order to find out their strengths and weaknesses. Those schemes will be compared based on few predefined parameters in terms of anti-shoulder surfing mechanism, memorability of password, usability and so on. After that, each password scheme will be ranked accordingly and an algorithm will be devised based on the concepts gained from the past researches.

### 3.1.2 Build Prototype

After the initial stage when the algorithm is fully devised, the prototype of the system will be developed. The project will reach this stage again if any improvement needs to be done to the prototype.

The figure shown in Figure 3-1-2 is the screenshot of the current prototype developed which is the generated password screen of the proposed picture-based password scheme. The user is required to select six image objects in order to proceed to the next password screen which will be total of three times in one authentication process.

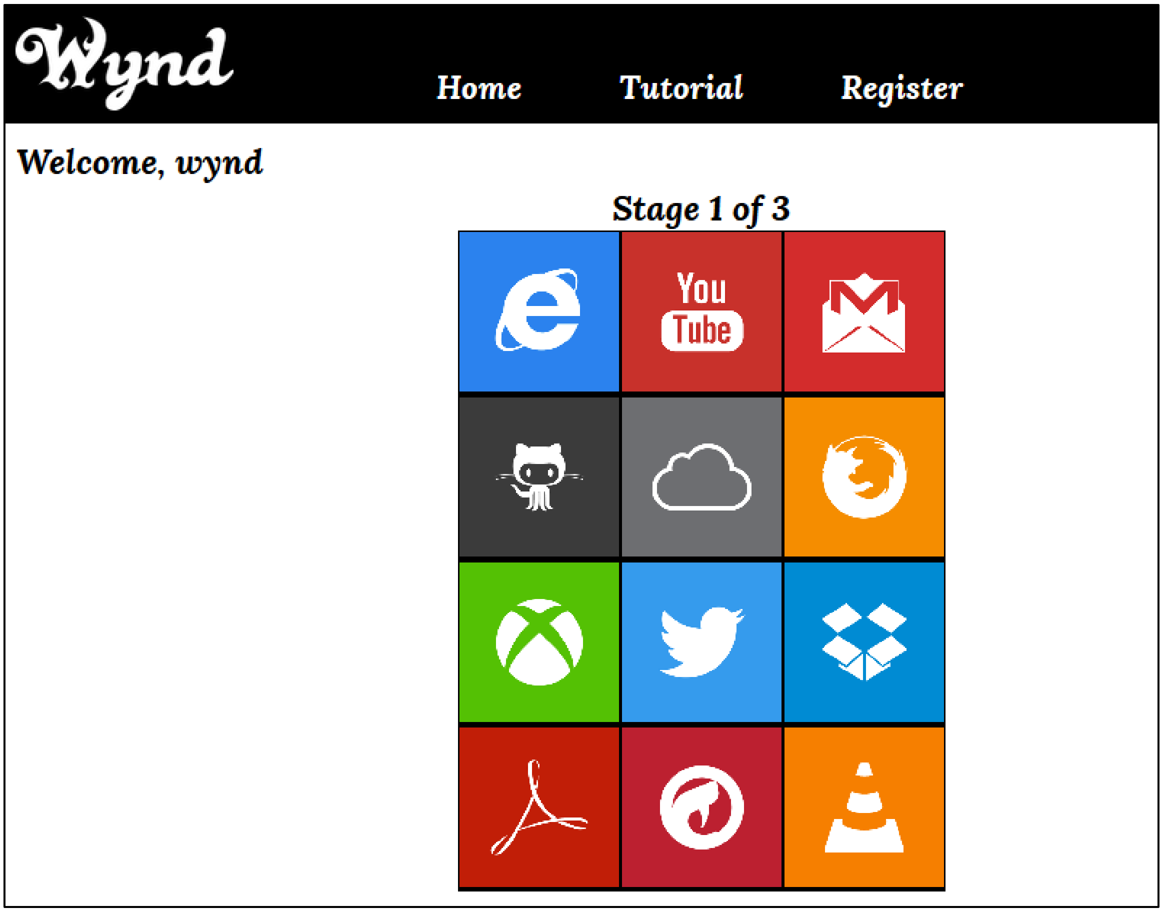


**Figure 3-1-2:** Screenshot of the Password Screen

### 3.1.3 Test and Evaluate Prototype

At this stage, the prototype will be tested and evaluated based on certain criteria such as its usability and level of security. This stage is also used to detect any bug present in the system.

### 3.1.4 Analyse and Design Prototype

Once the prototype is tested, at this stage, the prototype will be analysed to determine what kind of changes to be made to the system. Once the requirement is determined through the analysis, the prototype will be designed based on the specification.

### 3.1.5 Final System

This is the stage where all the requirements and improvements have already been made to the prototype and is also well tested. The prototype is ready to represent the final fully functional system.

## 3.2 Proposed Algorithm

### 3.2.1 Defining Concepts

This section will explain the key components of the system.

**1. Pass Objects**

Pass objects are also known as the user's password. A total of three pass objects will be registered by each user. The number of pass object selected in each authentication process will determine the validity of the picture-based password.

**2. Special Purpose Objects**

Two additional distinctive objects other than the pass objects will also be registered by the user. These two objects are used to convey some special messages to the user which are only understood by the user himself.

### 2.1 Flag Object

Presence of flag object indicates to include only one valid pass object in the selection of image objects for that particular authentication attempt. This means that the user is required to select only one valid pass object and five other non-pass image objects.

### 2.2 Skipping Object

Presence of skipping object indicates to include the skipping object in the selection of image objects for that particular authentication attempt. This means that the user is required to select the skipping object and any five image objects. The authentication will be based on the skipping object method even though the flag object is present in the same stage.

### 3.2.2 Base Idea

The system involves three stages during the authentication process. The first stage will start once the user has entered a valid username to the system. Each stage will display a password screen which consists of 12 image objects and user is required to select six image objects to proceed to the next stage. In order to gain access to the user account, the user is required to perform a valid selection for all three stages. The pass objects will not be selected in any stage unless signalled by special purpose objects. The special purpose objects will be generated randomly in any stage. The three stages varied from each other in terms of the presence of the special purpose objects. The types of the stage are stage with flag object, stage without flag object or skipping object and stage with skipping object.

### 3.2.3 Algorithm Steps

1. System applies the username submitted by the user.
2. System validates the username.
   - i. If the username is invalid or does not exist, return to step 1.
   - ii. If the username is valid, proceed to step 3.
3. System sets screen counter to 0.
4. System checks value of screen counter.
   - i. If screen counter is less than 3, proceed to step 5.
   - ii. If screen counter is not less than 3, proceed to step 10.
5. System generates password screen based on three criteria.
   - i. Without Flag Object and Skipping Object
   - ii. With Flag Object
   - iii. With Skipping Object
6. User selects six image objects from the password screen.
7. Value of screen counter increment by 1.
8. System determines the type of password screen generated and validates image objects selected by user.
   - i. If selection is valid, proceed to step 9.
   - ii. If selection is invalid, proceed to step 4.
9. Value of validating counter increment by 1 and proceed to step 4.
10. System checks the validating counter.
    - i. If value of validating counter is 3, proceed to step 11.
    - ii. If value of validating counter is not 3, proceed to step 1.
11. System grants access to the user account.

### 3.2.4 Flow Chart of Proposed Algorithm

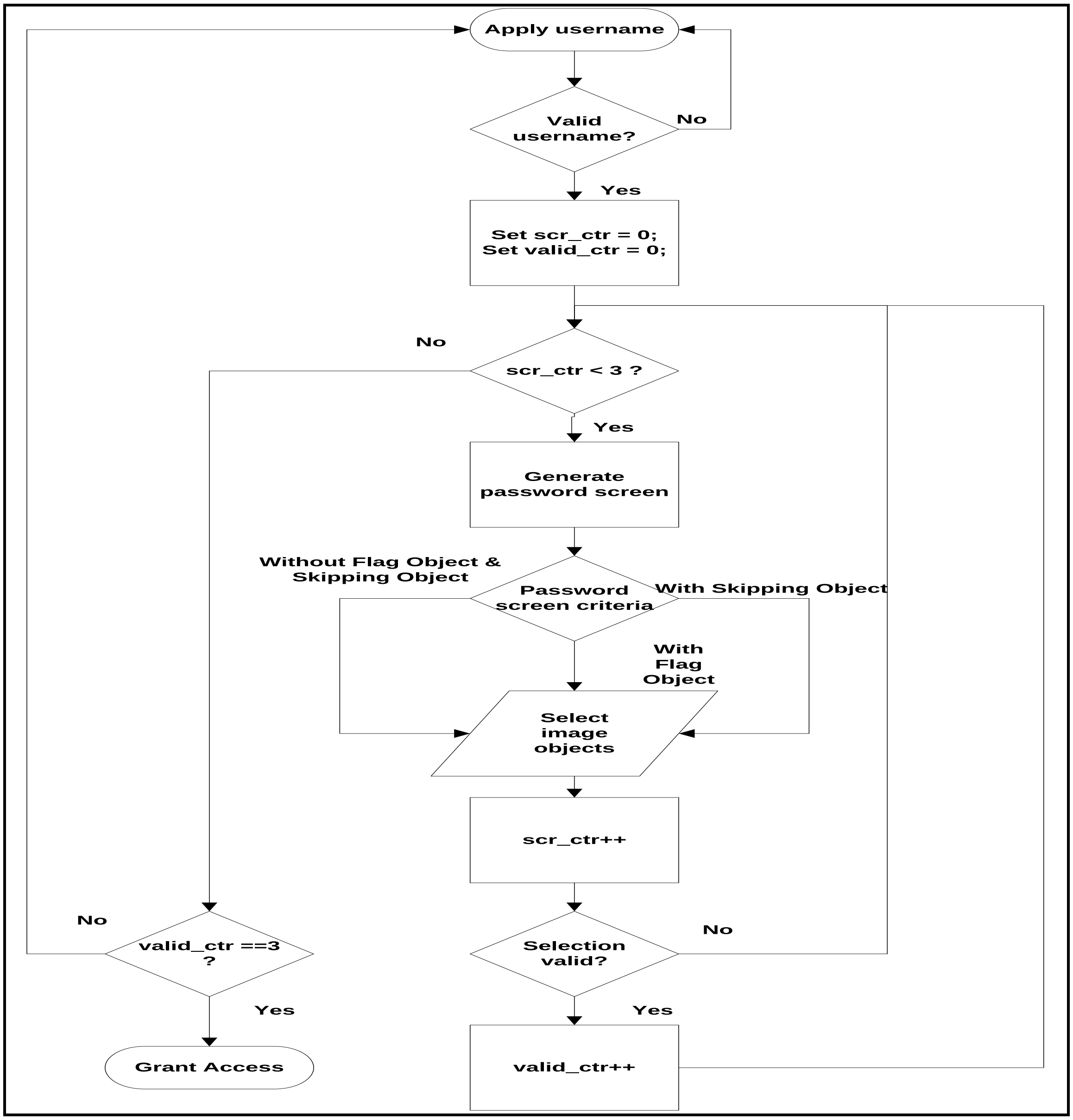
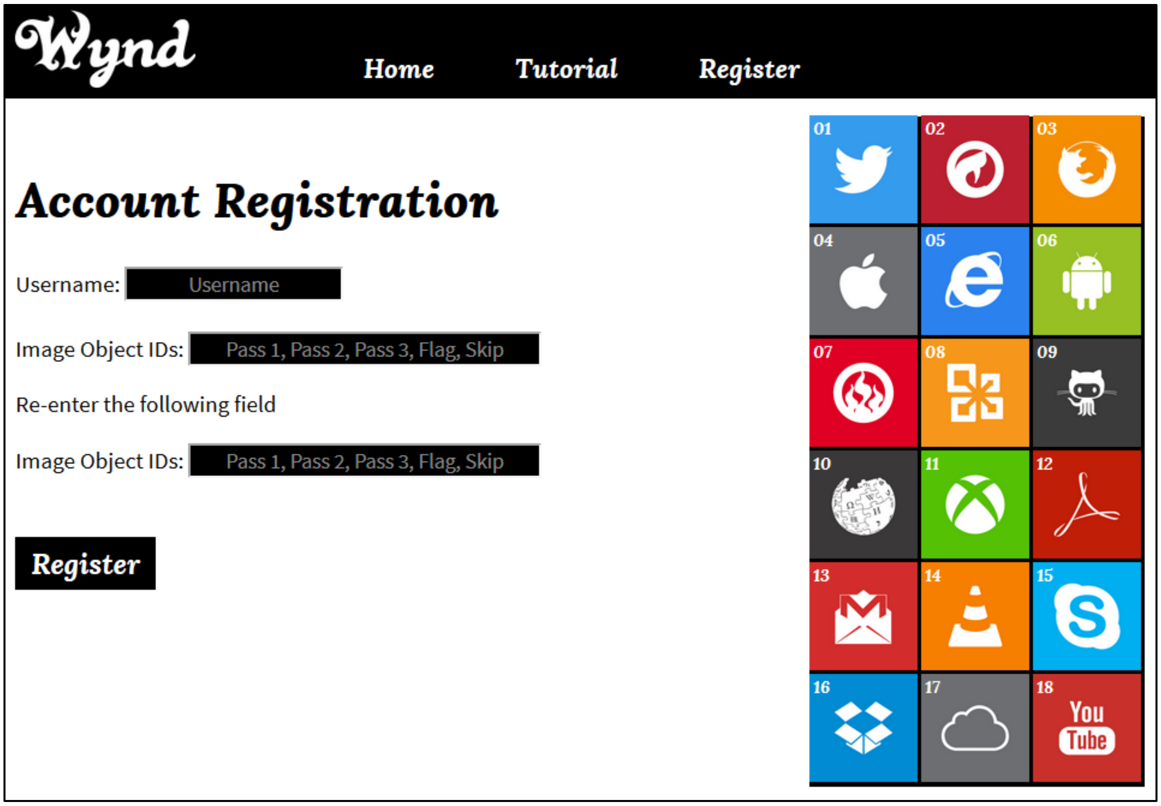


**Figure 3-2-4:** Flow chart of the proposed algorithm

### 3.2.5 Account Registration

In order to complete the registration process, user is required to register a username and three types of image objects which are Pass Objects, Flag Object and Skipping Object. At the registration page, a screen is generated which shows all the available images to be registered. Each image has its own unique identification (ID) and is displayed to the user. After the user has entered his username, the next step is to register his Pass Objects, Flag Object and Skipping Object. The user needs to have three Pass Objects, one Flag Object and one Skipping Object. All of these image objects must be unique from each other. After the user has entered the image object IDs, he is required to re-enter the IDs again in the next field in order to confirm his selection and to ensure that both fields are similar. The account registration page is shown in Figure 3-2-5. Any input on the image object ID fields is masked as asterisk or circle in order to prevent shoulder surfing attack during registration.



Account Registration

Username: Username

Image Object IDs: Pass 1, Pass 2, Pass 3, Flag, Skip

Re-enter the following field

Image Object IDs: Pass 1, Pass 2, Pass 3, Flag, Skip

**Figure 3-2-5:** Screenshot of the registration page

## 3.3 Requirement Specifications

### 3.3.1 Functional Requirements

1. The system will validate to check whether the username entered by the user is a registered username.

2. The system generates the password screen based on certain criteria which will be randomly selected such as display criteria with or without Skipping Object or the Flag Object.
3. For each submission of the password attempt, the system is able to validate the password and also compute the validating counter.

### 3.3.2 Non-functional Requirements

1. The system will not allow user to refresh or use the back function during the authentication process. If these activities are detected, user will be redirected back to the initial user login page.
2. Any unauthorised access to protected web pages will be redirected to the user login page.
3. During the registration process, the system will ensure that the new username entered does not exist in the database.
4. System will ensure that the five registered image objects are unique from each other.

### 3.4 Use Case Diagram

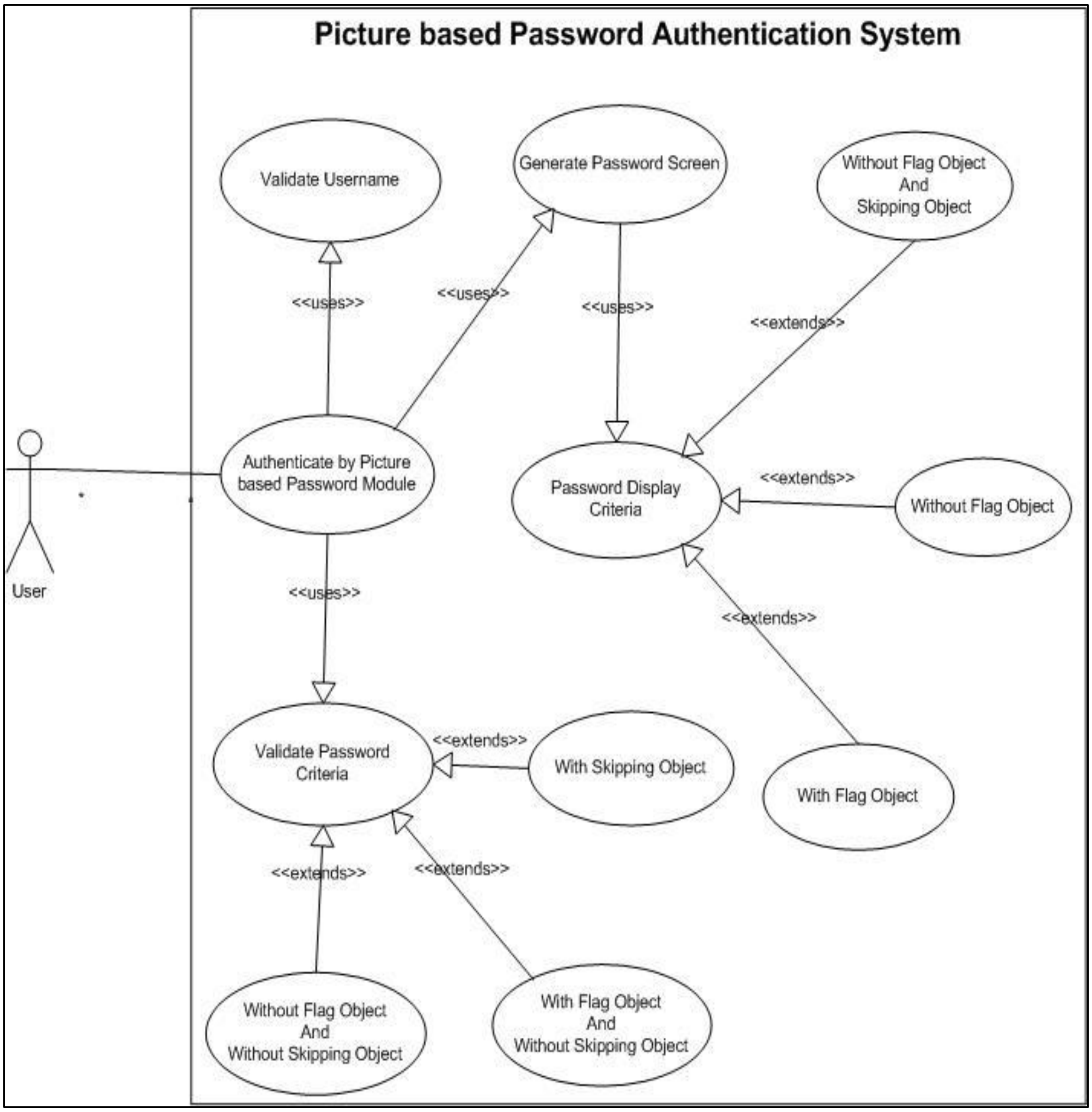

Picture based Password Authentication System
Validate Username
Generate Password Screen
Without Flag Object And Skipping Object
<<uses>>
<<uses>>
<<uses>>
<<extends>>
User
Authenticate by Picture based Password Module
Password Display Criteria
<<extends>>
Without Flag Object
<<uses>>
<<extends>>
Validate Password Criteria
<<extends>>
With Skipping Object
With Flag Object
<<extends>>
<<extends>>
Without Flag Object And Without Skipping Object
With Flag Object And Without Skipping Object

**Figure 3-4-1:** Use case diagram of the Authentication System

Based on Figure 3-4-1, it shows the use case diagram of the picture-based password authentication system. The system will validate the username submitted by the user in order to ensure that it is a registered username. After that, it will generate the password screen which will be based on three criteria. For each attempt, once the user has submitted his password, the system will validate the password based on three criteria. In one complete authentication process, the user is required to go through three attempts of password validation.

### 3.5 Use Cases

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC01 | | |
| **Use Case Name:** | User authenticate by picture-based password module. | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | User | | |
| **Description:** | Authenticate and permit access to the user account. | | |
| **Preconditions:** | User has accessed the login page. | | |
| **Trigger:** | User wanted to login to his account. | | |
| **Normal Flow:** | 1. User enters username in one form.<br>2. User submits the form.<br>3. System generates password display criteria queue. | | |

| | |
|---|---|
| | 4. System generates password screen.<br>5. User selects six objects.<br>6. User submits the password attempt.<br>7. System validates and increases the validating counter.<br>8. Repeat steps 3 to 6 for two times.<br>9. User login successfully to his account. |
| **Alternative Flows:** | 2.1 Invalid username<br>8.1 Login failed |
| **Exceptions Handling:** | 2.1.a Refer to use case UC02<br>8.1.a Repeat step 1. |
| **Frequency of Use:** | Moderate (used only when user wants to login to his account) |
| **Post-conditions:** | User logged into his account. |
| **Secondary Actors:** | Authentication server, System database |
| **Includes:** | UC02, UC03, UC07 |
| **Special Requirements:** | 1. The user’s internet connection should be stable without any disconnection.<br>2. The password screen cannot be refreshed and is non-reversible. |
| **Assumptions:** | Web service is available at all time. |
| **Notes and Issues:** | 1. What is the maximum amount of invalid username attempt allowed? |

| | |
|---|---|
| | 2. What is the maximum length of the username allowed? |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC02 | | |
| **Use Case Name:** | Validating the username | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th July 2013 |
| **Primary Actor:** | Authentication server | | |
| **Description:** | Check whether the username entered is a registered username or not. | | |
| **Preconditions:** | User has accessed the login page. | | |
| **Trigger:** | User submits the form with his username. | | |
| **Normal Flow:** | 1. System compares the username with the usernames stored in the system database.<br>2. System allows the user to proceed to the password screen. | | |
| **Alternative Flows:** | 1.1 No username matches found in the system database. | | |
| **Exceptions Handling:** | 1.1.a Return to use case UC01 step 1. | | |
| **Frequency of Use:** | Moderate (used only when user wants to login to his account) | | |
| **Post-conditions:** | User accesses the password screen. | | |

| | |
|---|---|
| **Secondary Actors:** | User, System database |
| **Includes:** | UC01 |
| **Special Requirements:** | 1. The user's internet connection should be stable without any disconnection.<br>2. The password screen cannot be refreshed and is non-reversible. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC03 | | |
| **Use Case Name:** | Generating password screen | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | Authentication server | | |
| **Description:** | System generates the password screen based on username. | | |
| **Preconditions:** | 1. A valid username is submitted to the system.<br>2. Password display criteria queue is generated. | | |
| **Trigger:** | 1. System validated the username submitted by the user.<br>2. Previous password screen is submitted by the user. | | |

| | |
|---|---|
| **Normal Flow:** | 1. System generates the password screen based on the current password display criteria queue.<br>2. System points to the next password display criteria queue.<br>3. System proceeds to step 5 in use case UC01. |
| **Frequency of Use:** | Moderate (three times) |
| **Post-conditions:** | Password screen is generated. |
| **Secondary Actors:** | User, System database |
| **Includes:** | UC01 |
| **Special Requirements:** | 1. The user's internet connection should be stable without any disconnection.<br>2. The password screen cannot be refreshed and is non-reversible. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC04 | | |
| **Use Case Name:** | Password display criterion 1: Without Flag Object and Skipping Object | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |

| | |
|---|---|
| **Primary Actor:** | Authentication server |
| **Description:** | Generating password screen without Flag Object and Skipping Object. |
| **Preconditions:** | A valid username is submitted to the system. |
| **Trigger:** | Selected by step 1 in UC03. |
| **Normal Flow:** | 1. System selects all three Pass Objects.<br>2. System randomly selects 9 objects which are not a Flag Object, Skipping Object or unselected Pass Object.<br>3. System randomly places all the selected objects in 12 locations. |
| **Frequency of Use:** | Low (One time) |
| **Post-conditions:** | Password screen generated with all three Pass Objects and 9 objects which are not Flag Object, Skipping Object or unselected Pass Object. |
| **Secondary Actors:** | System database |
| **Includes:** | UC03 |
| **Special Requirements:** | 1. The user’s internet connection should be stable without any disconnection. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC05 | | |
| **Use Case Name:** | Password display criterion 2: With Flag Object and without Skipping Object | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | Authentication server | | |
| **Description:** | Generating password screen with Flag Object and without Skipping Object. | | |
| **Preconditions:** | A valid username is submitted to the system. | | |
| **Trigger:** | Selected by step 1 in UC03. | | |
| **Normal Flow:** | 1. System selects all three Pass Objects.<br>2. System selects the Flag Object.<br>3. System randomly selects 8 objects which are not Skipping Object or unselected Pass Object.<br>4. System randomly places all the selected objects in 12 locations. | | |
| **Frequency of Use:** | Low (One time) | | |
| **Post-conditions:** | Password screen generated with all three Pass Objects, a Flag Object and 8 objects which are not Skipping Object or unselected Pass Object. | | |

| | |
|---|---|
| **Secondary Actors:** | System database |
| **Includes:** | UC03 |
| **Special Requirements:** | 1. The user's internet connection should be stable without any disconnection. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC06 | | |
| **Use Case Name:** | Password display criterion 3: With Skipping Object | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | Authentication server | | |
| **Description:** | Generating password screen with Skipping Object. | | |
| **Preconditions:** | A valid username is submitted to the system. | | |
| **Trigger:** | Selected by step 1 in UC03. | | |
| **Normal Flow:** | 2. System selects the Skipping Object.<br>3. System randomly selects 11 objects out of the unselected | | |

| | |
|---|---|
| | objects.<br>4. System randomly places all the selected objects in 12 locations. |
| **Frequency of Use:** | Low (One time) |
| **Post-conditions:** | Password screen generated with a Skipping Object and 11 random objects. |
| **Secondary Actors:** | System database |
| **Includes:** | UC03 |
| **Special Requirements:** | The user's internet connection should be stable without any disconnection. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC07 | | |
| **Use Case Name:** | Validating password criterion 1: Without Flag Object | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | Authentication server | | |
| **Description:** | Validate the password submitted by user and increase the validating counter. | | |

| | |
|---|---|
| **Preconditions:** | A valid password is submitted by user. |
| **Trigger:** | User selected six objects and submitted the password attempt. |
| **Normal Flow:** | 1. Validate the presence of Skipping Object.<br>2. Validate the presence of Flag Object.<br>3. Validate the number of Pass Object selected.<br>4. Proceed to step 8 in use case UC01. |
| **Alternative Flows:** | 1.1 Skipping Object is present.<br>2.1 Flag Object is present.<br>3.1 One or more Pass Object is selected.<br>3.2 No Pass Object is selected. |
| **Exceptions Handling:** | 1.1.a Proceed to use case UC09.<br>2.1.a Proceed to use case UC08.<br>3.2.a Increase the Validating Counter by one. |
| **Frequency of Use:** | Moderate (three times) |
| **Post-conditions:** | Value of Validating Counter is modified. |
| **Secondary Actors:** | System database |

| | |
|---|---|
| **Includes:** | UC01, UC08, UC09 |
| **Special Requirements:** | The user's internet connection should be stable without any disconnection. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC08 | | |
| **Use Case Name:** | Validating password criterion 2: With Flag Object | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th July 2013 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | Authentication server | | |
| **Description:** | Validate the password submitted by user and increase the validating counter. | | |
| **Preconditions:** | A valid password is submitted by user. | | |
| **Trigger:** | The password attempt is forwarded from use case UC07 | | |
| **Normal Flow:** | 1. Validate the number of Pass Object selected.<br>2. Proceed to step 8 in use case UC01. | | |

| | |
|---|---|
| **Alternative Flows:** | 1.1 Only one Pass Object is selected.<br>1.2 Three Pass Objects are selected.<br>1.3 Two Pass Objects are selected<br>1.4 No Pass Object is selected. |
| **Exceptions Handling:** | 1.1.a Increase the Validating Counter by one. |
| **Frequency of Use:** | Low (One time) |
| **Post-conditions:** | Value of Validating Counter is modified. |
| **Secondary Actors:** | System database |
| **Includes:** | UC01 |
| **Special Requirements:** | The user's internet connection should be stable without any disconnection. |
| **Assumptions:** | Web service is available at all time. |

| | | | |
|---|---|---|---|
| **Use Case ID:** | UC09 | | |
| **Use Case Name:** | Validating password criterion 3: With Skipping Object | | |
| **Created By:** | GWB | **Last Updated By:** | GWB |
| **Date Created:** | 20th March 2014 | **Date Last Updated:** | 20th March 2014 |
| **Primary Actor:** | Authentication server | | |

| | |
|---|---|
| **Description:** | Validate the password submitted by user and increase the validating counter. |
| **Preconditions:** | A valid password is submitted by user. |
| **Trigger:** | The password attempt is forwarded from use case UC07 |
| **Normal Flow:** | 1. Validate the number of Skipping Object selected.<br>2. Proceed to step 8 in use case UC01. |
| **Alternative Flows:** | 1.1 Skipping Object is selected.<br>1.2 No Skipping Object is selected. |
| **Exceptions Handling:** | 1.1.a Increase the Validating Counter by one. |
| **Frequency of Use:** | Low (One time) |
| **Post-conditions:** | Value of Validating Counter is modified. |
| **Secondary Actors:** | System database |
| **Includes:** | UC01 |
| **Special Requirements:** | The user's internet connection should be stable without any disconnection. |
| **Assumptions:** | Web service is available at all time. |

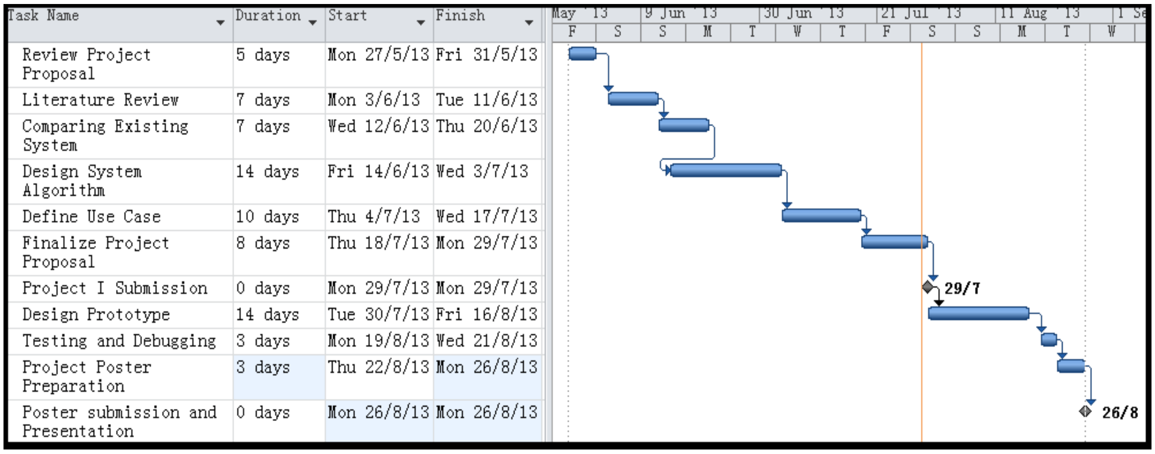
### 3.6 Tools and Technology

In this project, Microsoft Visual Studio Express 2012 for Web and Notepad++ will be used for coding. HTML (Hyper Text Mark-up Language) will design the basic document structure of the web page. CSS (Cascading Style Sheets) will be used to design the outlook of web page in order to produce an attractive user interface. Besides that, to handle users' input, JavaScript will be used as it is able to perform event handling for example when user performs mouse click on the image objects. Since the system is required to access to the server database and also store the session data, server-side scripting language which is PHP will be used.

As for the database language, MySQL will be used as it allows the administrator of the system to create or modify database structure, query information from the database and so on. Besides, MySQL is open source which is free of charge. As for the web server, Apache2 will be used since it is also open source and free of charge. The web application will be mainly running use WampServer as it provides a Windows web development environment integrated with Apache2, PHP and a MySQL database.

### 3.7 Timeline

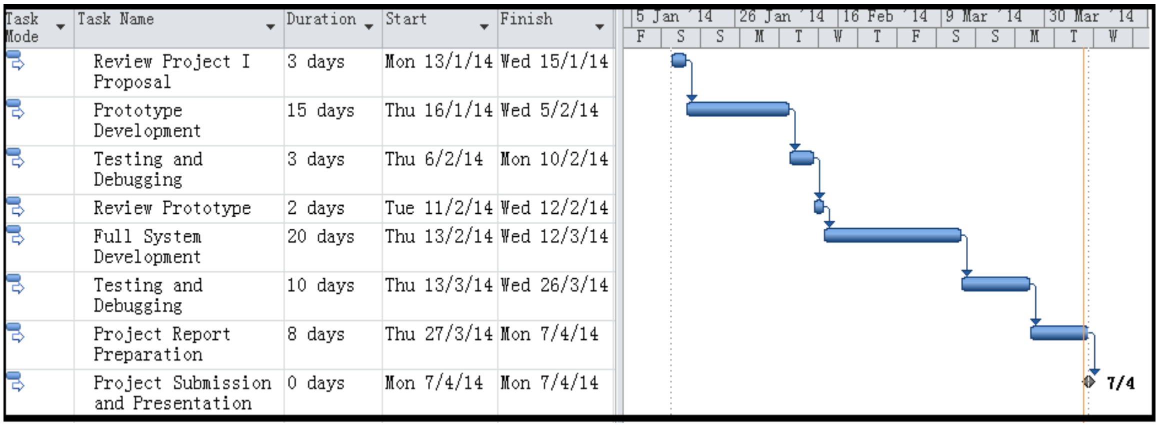
The Figure 3-8-1 shows the Gantt chart for the timeline in Project I. It consists of two milestones which are the project proposal submission which is on Week 10 and the submission of poster and oral presentation which will be on Week 14.



| Task Name | Duration | Start | Finish |
|---|---|---|---|
| Review Project Proposal | 5 days | Mon 27/5/13 | Fri 31/5/13 |
| Literature Review | 7 days | Mon 3/6/13 | Tue 11/6/13 |
| Comparing Existing System | 7 days | Wed 12/6/13 | Thu 20/6/13 |
| Design System Algorithm | 14 days | Fri 14/6/13 | Wed 3/7/13 |
| Define Use Case | 10 days | Thu 4/7/13 | Wed 17/7/13 |
| Finalize Project Proposal | 8 days | Thu 18/7/13 | Mon 29/7/13 |
| Project I Submission | 0 days | Mon 29/7/13 | Mon 29/7/13 |
| Design Prototype | 14 days | Tue 30/7/13 | Fri 16/8/13 |
| Testing and Debugging | 3 days | Mon 19/8/13 | Wed 21/8/13 |
| Project Poster Preparation | 3 days | Thu 22/8/13 | Mon 26/8/13 |
| Poster submission and Presentation | 0 days | Mon 26/8/13 | Mon 26/8/13 |

**Figure 3-7-1:** Gantt Chart of Project I

The Figure 3-8-2 shows the Gantt chart for the timeline in Project II. It is mainly focused on developing and testing the system. The project milestone is the submission and presentation of the final system and report.



| Task Mode | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| | Review Project I Proposal | 3 days | Mon 13/1/14 | Wed 15/1/14 |
| | Prototype Development | 15 days | Thu 16/1/14 | Wed 5/2/14 |
| | Testing and Debugging | 3 days | Thu 6/2/14 | Mon 10/2/14 |
| | Review Prototype | 2 days | Tue 11/2/14 | Wed 12/2/14 |
| | Full System Development | 20 days | Thu 13/2/14 | Wed 12/3/14 |
| | Testing and Debugging | 10 days | Thu 13/3/14 | Wed 26/3/14 |
| | Project Report Preparation | 8 days | Thu 27/3/14 | Mon 7/4/14 |
| | Project Submission and Presentation | 0 days | Mon 7/4/14 | Mon 7/4/14 |

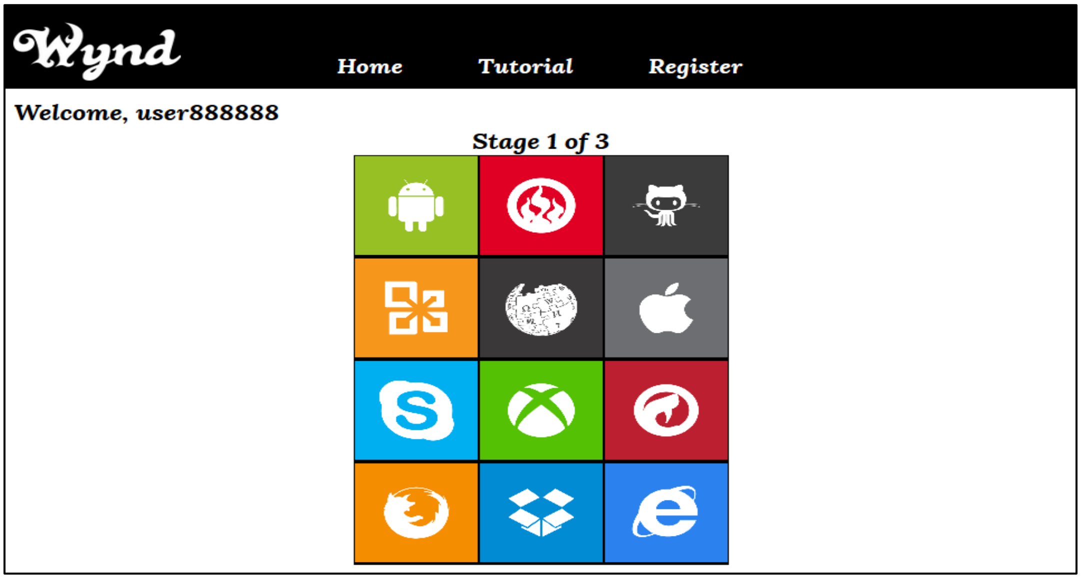
**Figure 3-7-2:** Gantt Chart of Project II

## 3.8 Design and Verification Plan

After the algorithm and system is implemented, some tests are conducted to verify the design of the system.

### 3.8.1 Username Validation

The system consists of a class function which will validate the username entered by the user during the login process. Any invalid username entered by the user will be flagged by the system to perform certain tasks such as sending an error message to the user or generating a random password screen. The code of this class function can refer to Appendix A-1. As shown in Figure 3-8-1, the system generated a random password screen for a non-existing user to prevent the system from exposing the valid usernames.



Welcome, user888888

Stage 1 of 3

**Figure 3-8-1:** Screenshot of random password screen

### 3.8.2 Account Validation

During account registration, user is required to register a valid and non-existing username. Besides that, the system will also ensure that all the image objects registered by the user are unique and both image object ID field and the re-enter ID field are similar. The screenshot shown in Figure 3-8-2 is the case when user registered an invalid image object ID to the system. The class function which handles the validation of the registration process can refer to Appendix A-2.

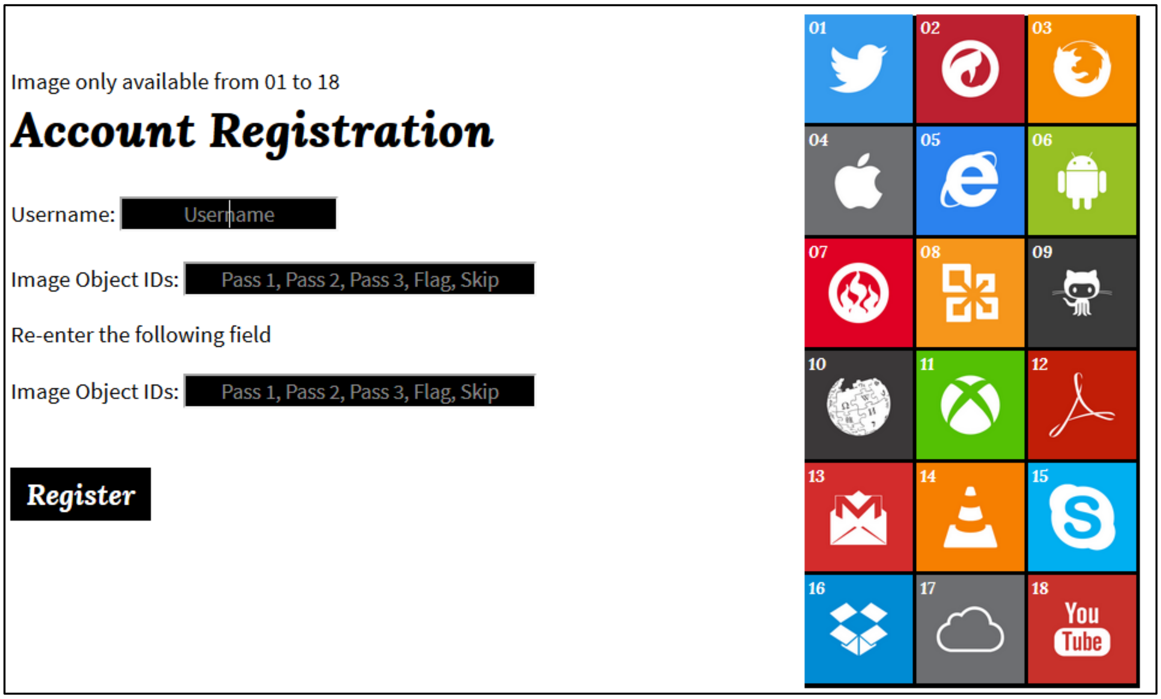


Image only available from 01 to 18

Account Registration

Username: Username

Image Object IDs: Pass 1, Pass 2, Pass 3, Flag, Skip

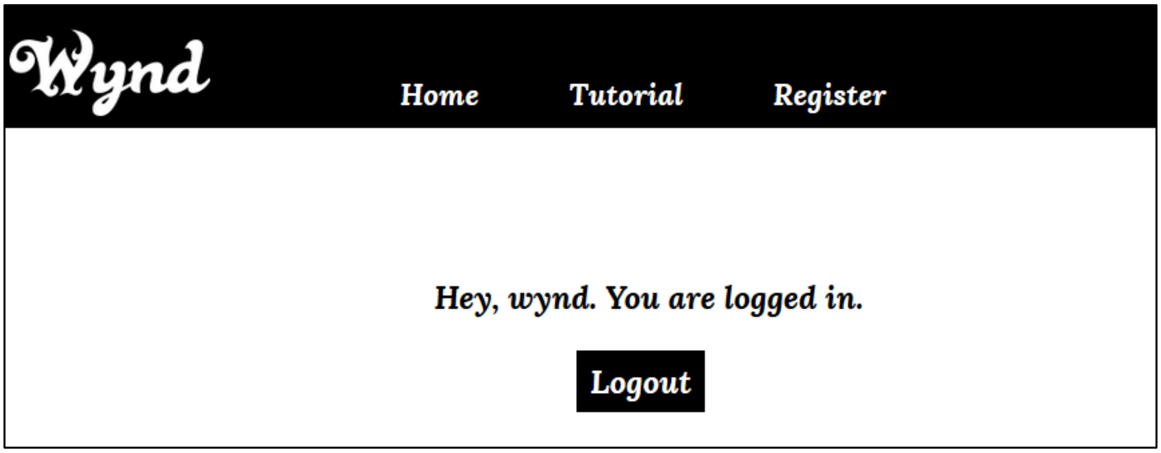
Re-enter the following field

Image Object IDs: Pass 1, Pass 2, Pass 3, Flag, Skip

**Figure 3-8-2:** Screenshot of invalid image ID input

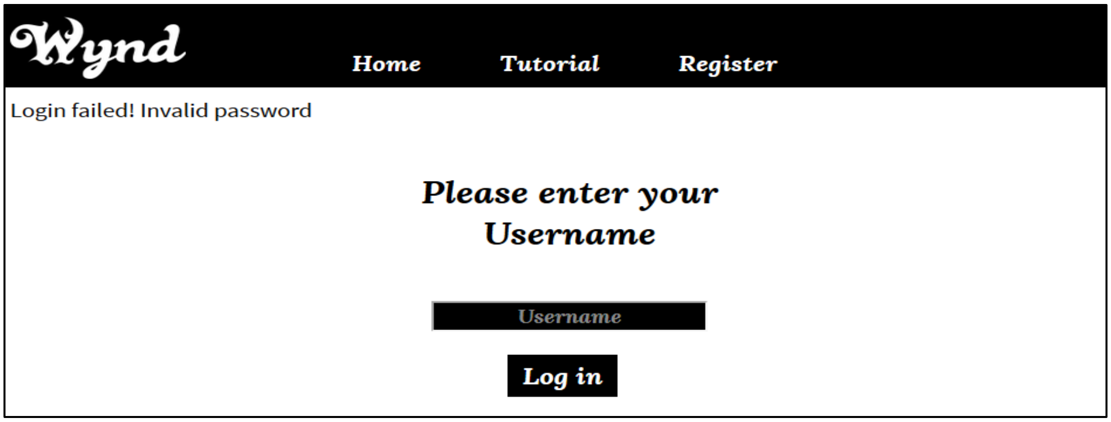
### 3.8.3 Image Login Validation

The system is able to validate the selections of the user to determine whether the user has selected the correct combination image objects or not. If the user has completed all stages with the correct selections, he will be able to login successfully as shown in Figure 3-8-3a. If the user fails to select the correction combination of images in any stage, he will be redirected back to the initial login page with an error message as shown in Figure 3-8-3b. The class function which handles this image login validation is shown in Appendix A-3.



Hey, wynd. You are logged in.

**Figure 3-8-3a:** Screenshot of successful user login



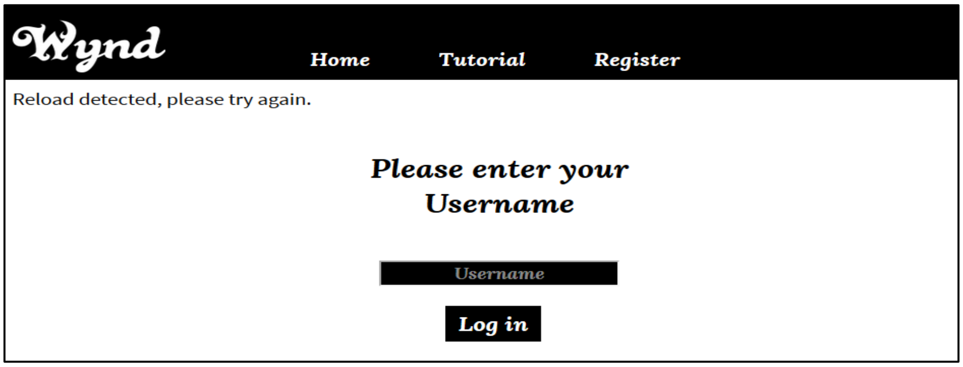
Login failed! Invalid password

Please enter your Username

**Figure 3-8-3b:** Screenshot of failed user login

### 3.8.4 Reload Detection

The system will prevent user from reloading the page or using the back function during the image login process. If these activities are detected, the system will redirect the user to the initial login page and display an error message as shown in Figure 3-8-4. The function which performs this detection task is shown in Appendix A-4.



Reload detected, please try again.

Please enter your Username

**Figure 3-8-4:** Reload detection

# 4. PROJECT IMPLEMENTATION

In this section, it will be mainly focused on the final system implementation. It will cover the achievements, issues, future development and also discussion on the final system.

## 4.1 Implementation Issues and Challenges

### 4.1.1 Reload Prevention during Authentication

In this system authentication process, unlike other typical login system which usually requires only one step to complete, this system requires at least three steps as the user needs to complete three different types of password screens. This will actually cause an issue to the system if the user reloads or uses the back function of the browser in the middle of his authentication. Problems or issues which may occur if the user reloads in the middle of the authentication are submission of the duplicated forms and also system crash since the reload function has modified values of the variables in the system.

In order to prevent the reload and back functions, the method being used in this system is focused on the conditions to generate the password screen. The system will only generate a new password screen if the required conditions are met or else the user will be redirected back to the username login page.

### 4.1.2 Registration with Anti-shoulder Surfing Mechanism

This system is designed to provide a login system using image and at the same time consists of anti-shoulder surfing mechanism in order to prevent shoulder surfer from spying the users' passwords. Therefore, it is important to ensure that the account registration process is also protected from shoulder surfer. The reason is because that, if the user's password is already exposed to the shoulder surfer in the initial stage during account registration, this will actually destroy the anti-shoulder surfing mechanism of the system authentication process since the shoulder surfer already knows the user's password.

In order to solve this issue, a text-based registration method is implemented. In this method, all the image objects will be displayed on the screen and each of them will have a unique identification. In order to register those image objects, the user needs to type in each of the image object identification. In addition, in order to prevent shoulder surfer from viewing the inputted image object identifications, the input field of the image objects identification will be masked and is shown as asterisks or circles.

### 4.1.3 Brute Force Attack on Image Objects

In this system, when it receives a valid username, it will immediately generate the password screen based on the user's registered image objects. However, this will actually create an issue regarding the username and account. A username is usually public to others as its main function is not to secure account but to serve as identification. As a result, an attacker can easily get a username and input to the system. The system does not authenticate who is the actual owner of that particular username and it will generate the user's password screen as long as it receives a username. This issue gives the attacker the opportunity to conduct brute force attack. Besides that, there are also chances that the attacker may figure out the type of image objects registered by the user since he can generate unlimited numbers of password screens and based on them to predict the user's registered image objects.

Brute force attack usually cannot be prevented, but it can be limited in terms of time, process and availability.

In order to limit brute force attack, two methods can be used which are by implementing CAPTCHA and a temporary account lock. CAPTCHA is a program which functions to prevent bots from attacking the website by implementing a test which is only understandable by human. When the system detects multiple failed login attempts on a particular user account, the system might trigger the CAPTCHA to prevent bots attack. Besides that, even though that the hacker is a human, he will need to spend some time in order to pass the CAPTCHA and this actually increases the time for the brute force attack. As for temporary lock, it can be a timed-based temporary lock or non-time-based lock. A time-based lock will be triggered after few failed login attempts and the user account will be lock for a period of time before the next login. As for non-timed-based temporary lock, the user account will be locked after certain amount of failed login attempts and then it requires to owner of the account to unlock the account through a user validation process for example via the user's email account. Currently the system has not been implemented with these features but it will be implemented in future.

### 4.1.4 Exposure of Valid Usernames

When the system receives a valid username, it will automatically generate the password screen for that username and if the system receives an invalid username, an error message will be displayed. This feature actually consists of the risk of exposing valid usernames since the attacker is able to input random username in order to determine its validity and a valid username will cause the system generates a password screen.

To prevent such issue occur, a method is implemented on the system whereby the system will always generate a password screen even though that username does not exist. If an invalid username is entered to the system, will system will generate a random password screen. Therefore, the attacker might not be able to know to validity of the username.

## 4.2 Project Achievements

### 4.2.1 Image based Password Algorithm

In the beginning of this project, it involves analysis of multiple existing graphical password schemes. The initial target of this task is to determine the best existing graphical password schemes and then improvise its algorithm. However, in later stage this project has achieved a better goal whereby it successfully devised a new non-existing graphical password algorithm.

In addition, the newly devised algorithm shows a significant lead compared to those analyzed existing graphical password schemes. This can be shown in terms of the number of image objects in a login screen. The devised algorithm only requires 12 images object per login screen compared to other schemes such as the Triangle Scheme and Intersection Scheme which requires up to 1000 image objects. Besides that, this algorithm also consists of the anti-shoulder surfing mechanism whereby some graphical password schemes do not have for example the pattern lock. The new algorithm also requires up to three stages only compared to some schemes which require up to 10 attempts.

Other than that, this newly devised algorithm is also suitable to be implemented on mobile devices as the password screen does not require much space since only 12 image objects are needed.

### 4.2.2 Web-based Graphical Password Application

Another achievement of this project is that, the newly devised algorithm has been developed to a web-based application. The web-based authentication system is now a fully functional system which will perform according to the devised algorithm. The web-based application is also developed with user interfaces which will help user to understand better on the authentication system. Besides that, the system is also able to provide an optimal level of security and usability.

## 4.3 Discussion

### 4.3.1 Flag Object VS Skipping Object

In this subtopic will be discussing about the main purpose of introducing Flag Object and Skipping Object into this system. The flag object in this system aims to prevent the Pass Objects being guessed easily by the shoulder surfer. If flag object is not implemented in the system, the pass objects can easily be guessed by others since the pass objects will never be selected in a non-flag password screen. The attacker only needs to identify those unselected image objects in order to predict the user's pass objects. However, when flag object comes in place, the attacker will not be able to guess the pass objects as easily as the previous case. This is because during each flag object password screen, any one of the three pass objects has the possibility to be selected which makes the pass objects of the user become more complex to be guessed.

The structure of the password screens of non-flag and flag have a similarity which is both screens will display all three pass objects. This can be an issue whereby the attacker only needs to guess the three pass objects out of the 12 images objects in a password screen.

Therefore, to counter such issue, Skipping Object is introduced to the system. The aim of the skipping object is to cover up the similarity of the structure of the password screens for flag and non-flag. During skipping object step, the password screen will consist of the skipping object with the remaining 11 images objects which are selected randomly. As a result, there are a lot of combinations of password screen which can be formed and it causes the possibility of the attacker to guess the pass object to be very low. In addition, skipping object also functions as part of the authentication validation criteria as the selection of skipping object will affect the validation counter. Lastly, skipping object has a higher priority compared to flag object. This means that, if both skipping object and flag object appear on the same password screen, the skipping object will be selected as the validation criterion while the flag object is treated as a random non-functional object.

### 4.3.2 Dynamicity of the Algorithm

The devised image based password algorithm for this project is a dynamic algorithm. Part of its structures can be adjusted in order to meet different requirements. The types of requirements can be classified to three types which are the security, anti-shoulder surfing capability and usability.

For higher security requirement, this can be achieved by manipulating the number of attempts needed for each authentication. For the current system, the number of attempt is set to three and it can be increased for example up to five attempts for better security. Besides that, increasing the number of image objects in a password screen also able to increase the system security level. This can be achieved by increasing the current number of the image object per password screen from 12 to 16 image objects. However, there is a trade-off between security and usability. A system with better security level will usually result a lower usability level of the system.

As for the anti-shoulder surfing capabilities, increasing this capability means that the user registered image objects will be more secure from being guessed by the attacker. This can be achieved by increasing the number of image objects in a password screen and also increase the number of image object to be selected in each attempt. For example, the number of image object in the current system password screen is increased from 12 to 16 and the number of image object to be selected in each attempt is increased from six to 10. This method will increase the number of image objects which are possible to be the image object registered by the user. As a result, the attacker will need to use a lot of time to guess the possible image object registered by the user since there are more possibilities to be filtered. Similarly as the security level, increasing the anti-shoulder surfing capabilities of the system will also result in a less usability system for the user.

To increase the usability of the system, it involves the process of making the system to become simpler so that the user only require less effort and time to complete each authentication. The usability of the system can be increased by reducing the number of attempt for one complete authentication, for example reducing the number of attempt of the current system from three steps to one step. Besides that, the number of image object per password screen can also be reduced for example from the current system 12 to 8 image objects. However, increase in usability of the system will result a less secure and more vulnerable system in terms of security and anti-shoulder surfing capabilities.

In short, these three requirements must be determined and prioritized according in order to match the needs of the system also provides the optimal level of security and usability.

## 4.4 Future Work

### 4.4.1 Brute Force Attack

In order to minimize brute force attack, several methods can be implemented to the system which will increase the time needed for each attack. The system can be implemented with CAPTCHA and account lock mechanism. These methods will be triggered if several failed login attempts are detected by the system.

### 4.4.2 Integration of Textual and Picture-based Password

The proposed algorithm can be further developed in order to integrate with textual password in order to enable two-steps verification mechanism. The two-steps verification mechanism will greatly improve the security of the system as it consists of both textual and picture-based password algorithm features. As a result, the attacker is required to crack two types of passwords which are totally different from each other in order to login to the user's account.

### 4.4.3 Mobile Device Application

The proposed picture-based password scheme can be developed into a mobile application such as mobile device screen lock. This mobile application can replace the existing screen lock applications such as pattern lock since they do not have anti-shoulder surfing mechanism. Besides that, existing textual screen locks can also be vulnerable to shoulder surfing attack as the password can be guessed by spying on the key pressed by the user on the mobile device keyboard.

# 5. CONCLUSION

In a nutshell, information security is getting more important as there are more highly sensitive data which will be stored on the network. Strong textual password can takes a lot time to be cracked but not many users are committed to use strong textual password as it is complex. Picture-based password is said to be easier to remember compared to textual password as image is easier to remember compared to words. Besides that, picture-based password is secure from key logging attacks unlike textual password.

Anti-shoulder surfing mechanism is an important feature of a picture-based password scheme as it will determine whether the picture password is easy to be guessed by others or not. There are some existing graphical password schemes which also implement the anti-shoulder surfing mechanism, however majority of them requires large number of images in order to be secure and this will actually slow down the user authentication process which is not efficient.

This project has developed a new picture-based password scheme which not only has anti-shoulder surfing mechanism implemented and at the same time does not requires large number of images compared to those existing proposed solutions. Due to the dynamicity of the algorithm, this project can be further developed on various applications as it is capable of preventing shoulder surfing attack and overcoming the limitations of textual password.

## REFERENCES

Adams, A. and Sasse, M. A., 1999. Why users compromise computer security mechanisms and how to take remedial measures. *Users are not the enemy*, 42, pp. 41-46.

Ahn, L., Blum, M., Hopper, N. and Langford, J., 2000. *CAPTCHA: Telling Humans and Computers Apart Automatically* [online] Available at: < http://www.captcha.net/> [Accessed 23 March 2014]

freeanimationhub, 2012. Pattern Lock. [image online] Available at: < http://elechub.com/how-to-reset-your-pattern-lock-on-andriod-phones/> [Accessed 7 April 2013].

Haider al-Khateeb, 2011. *Security and Usability in Click-based Authentication Systems.* [pdf] Available at: <http://uobrep.openrepository.com/uobrep/bitstream/10547/142229/1/Haider_al-Khateeb_Thesis_April_2011.pdf> [Accessed 27 March 2013].

Kotadia, M., 2005. *Microsoft: Write down your passwords* [online] ZDNet. Available at: < http://www.zdnet.com/microsoft-write-down-your-passwords-1139193117> [Accessed 3 April 2013].

Lai, H. L., 2009. *Cued recall graphical password system resistant to shoulder surfing.* [pdf] Available at: <http://dspace.fsktm.um.edu.my/bitstream/1812/456/1/Cued%20Recall%20Graphical%20Password%20Resistant%20to%20Shoulder%20Surfing.pdf> [Accessed 27 March 2013].

Lim, K. S., Norafida Ithnin and Hazinah K. Mammi, 2011. *Identifying the Reusability of the Triangle and Intersection Schemes on Mobile Devices* [pdf] Available at: < http://el.trc.gov.om:4000/htmlroot/ENGG/tcolon/e_references/Consolidated/Computer%20Science/Journals/Identifying%20the%20Reusability%20of%20the%20Triangle%20and%20Intersection%20Schemes%20on%20Mobile%20Devices.pdf> [Accessed 8 February 2013].

Lim, K. S., Norafida Ithnin and Hazinah K. Mammi, 2012. *An Anti - Shoulder Surfing Mechanism and its Memorability Test* [pdf] Available at: <http://www.sersc.org/journals/IJSIA/vol6_no4_2012/8.pdf> [Accessed 20 March 2013].

Monrose, F. and Reiter, M. K., 2005. *Graphical Passwords* [pdf] Available at: <http://hornbeam.cs.ucl.ac.uk/hcs/teaching/GA10/lec2extra/ch09-1monrose.pdf> [Accessed 20 March 2013].

Muhammad Daniel Hafiz, Abdul Hanan Abdullah, Norafida Ithnin and Hazinah K. Mammi, 2009. *Towards Identifying Usability and Security Features of Graphical Password in Knowledge Based Authentication Technique*, pp.396-403.

Passfaces, 2005. *Two Factor Authentication for the Enterprise.* [online] Available at: <http://www.realuser.com> [Accessed 7 February 2013].

Sobrado, L. and Birget, J., 2002. *Graphical passwords.* [online] The Rutgers Scholar. Available at: < http://rutgersscholar.rutgers.edu/volume04/sobrbirg/sobrbirg.htm> [Accessed 1 February 2013].

# APPENDIX A SYSTEM VERIFICATION FUNCTIONS

## A-1 Username Login Validation

```
<?php

/**
 * Class login
 * handles the user's login and logout process
 */
class Login
{
…

    /**
     * log in with post data
     */
    private function dologinWithPostData()
    {
        // check login form contents
        if (empty($_POST['user_name'])) {
            $this->errors[] = "Username field was empty.";
        } elseif (!empty($_POST['user_name'])) {

            // create a database connection, using the constants from
config/db.php (which we loaded in index.php)
            $this->db_connection = new mysqli(DB_HOST, DB_USER,
DB_PASS, DB_NAME);

            // change character set to utf8 and check it
            if (!$this->db_connection->set_charset("utf8")) {
                $this->errors[] = $this->db_connection->error;
            }

            // if no connection errors (= working database connection)
            if (!$this->db_connection->connect_errno) {

                // escape the POST stuff
                $user_name = $this->db_connection-
>real_escape_string($_POST['user_name']);

                // database query, getting all the info of the selected
user (allows login via email address in the
                // username field)
                $sql = "SELECT username
                        FROM users
                        WHERE username = '" . $user_name . "'";
                $result_of_login_check = $this->db_connection-
>query($sql);

                // if this user exists
                if ($result_of_login_check->num_rows == 1) {

                    // get result row (as an object)
                    $result_row = $result_of_login_check-
>fetch_object();

                        // write user data into PHP SESSION (a file on
your server)
                        $_SESSION['user_name'] = $result_row->username;
                        $_SESSION['user_login_status'] = 1;

                } else {
                              $_SESSION['user_login_status'] = 1;
                              $_SESSION['user_name'] = $user_name;
                              $_SESSION['invaliduser'] = 1;
                }
            } else {
                $this->errors[] = "Database connection problem.";
            }
        }
    }
…
}
```

## A-2 Account Registration Validation

```
<?php

/**
 * Class registration
 * handles the user registration
 */
class Registration
{
…

    /**
     * handles the entire registration process. checks all error
possibilities
     * and creates a new user in the database if everything is fine
     */
    private function registerNewUser()
    {
        if (empty($_POST['user_name'])) {
            $this->errors[] = "Empty Username";
        } elseif (empty($_POST['pass']) || empty($_POST['rpass'])) {
            $this->errors[] = "Empty Field";
        } elseif (!empty($_POST['user_name'])
                  && !empty($_POST['pass'])
                  && !empty($_POST['rpass'])) {
                        $tmp = str_split($_POST['pass'],2);
                        $tmp2 = str_split($_POST['rpass'],2);

                        $pass1 = intval($tmp[0]);
                        $pass2 = intval($tmp[1]);
                        $pass3 = intval($tmp[2]);
                        $flag = intval($tmp[3]);
                        $skip = intval($tmp[4]);

                        $rpass1 = intval($tmp2[0]);
                        $rpass2 = intval($tmp2[1]);
                        $rpass3 = intval($tmp2[2]);
                        $rflag = intval($tmp2[3]);
                        $rskip = intval($tmp2[4]);

                        if($pass1 != $rpass1) {
                              $this->errors[] = "Field and re-entered
field are not the same";
                        } elseif ($pass2 != $rpass2){
                              $this->errors[] = "Field and re-entered
field are not the same";
                        } elseif ($pass3 != $rpass3){
                              $this->errors[] = "Field and re-entered
field are not the same";
                        } elseif ($flag != $rflag){
                              $this->errors[] = "Field and re-entered
field are not the same";
                        } elseif ($skip != $rskip){
                              $this->errors[] = "Field and re-entered
field are not the same";
                        } elseif ($pass1 < 1 || $pass1 >18) {
                              $this->errors[] = "Image only available
from 01 to 18";
                        }

.
.
.

                        } elseif ($pass3 == $flag) {
                              $this->errors[] = "Each Object must be a
unique image";
                        } elseif ($pass3 == $skip) {
                              $this->errors[] = "Each Object must be a
unique image";
                        } elseif ($flag == $skip) {
                              $this->errors[] = "Each Object must be a
unique image";
                        } elseif (strlen($_POST['user_name']) > 64 ||
strlen($_POST['user_name']) < 2) {
                              $this->errors[] = "Username cannot be
shorter than 2 or longer than 64 characters";
                        } elseif (!preg_match('/^[a-z\d]{2,64}$/i',
$_POST['user_name'])) {
                              $this->errors[] = "Username error: only
a-Z and numbers are allowed, 2 to 64 characters";
                        } elseif (!empty($_POST['user_name'])
                              && strlen($_POST['user_name']) <= 64
                              && strlen($_POST['user_name']) >= 2
                        && preg_match('/^[a-z\d]{2,64}$/i',
$_POST['user_name'])
```

```
                        && !empty($pass1)
                        && !empty($rpass1)
                        && ($pass1 === $rpass1)
                        && !empty($pass2)
                        && !empty($rpass2)
                      && ($pass2 === $rpass2)
                              && !empty($pass3)
                        && !empty($rpass3)
                        && ($pass3 === $rpass3)
                              && !empty($flag)
                        && !empty($rflag)
                    && ($flag === $rflag)
                              && !empty($skip)
                    && !empty($rskip)
                    && ($pass1 != $pass2)
                              && ($pass1 != $pass3)
                              && ($pass1 != $flag)
                              && ($pass1 != $skip)
                              && ($pass2 != $pass3)
                              && ($pass2 != $flag)
                              && ($pass2 != $skip)
                              && ($pass3 != $flag)
                              && ($pass3 != $skip)
                              && ($flag != $skip)
                        ) {

.
.
.
    }
}
```

## A-3 Image Login Validation

```php
<?php

/**
 * Class ImgLogin
 * handles the user's image login process
 */
class ImgLogin
{
    …
      private function validpass()
      {
            $_SESSION['attempt']++;
            unset($_SESSION['rand']);

                  if ($_SESSION['type'] == 3) //invalid user
                  {
                        $_SESSION['user_login_status2'] = 0;
                  }
                  else if ($_SESSION['type'] == 0) //skip type
                  {
                        $_SESSION['skipcheck'] = 0;
                        for($i=1; $i<7; $i++)
                        {
                              if($_POST[$i] == $_SESSION['skip'])
                                    $_SESSION['skipcheck'] = 1;
                        }

                        if($_SESSION['skipcheck'] == 1)
                              $_SESSION['user_login_status2']++;
                  }
                  else if ($_SESSION['type'] == 1) //flag type
                  {
                        $_SESSION['flagcheck'] = 0;
                        for($i=1; $i<7; $i++)
                        {
                              if($_POST[$i] == $_SESSION['img1'] ||
                              $_POST[$i] == $_SESSION['img2'] ||
                              $_POST[$i] == $_SESSION['img3'] )
                                    $_SESSION['flagcheck']++;

                        }
```

```
                         if($_SESSION['flagcheck'] == 1)
                               $_SESSION['user_login_status2']++;
                         else
                               $_SESSION['user_login_status2'] = 0;
                   }
                   else if ($_SESSION['type'] == 2) //no flag type
                   {
                         $_SESSION['xflagcheck'] = 0;
                         for($i=1; $i<7; $i++)
                         {
                               if($_POST[$i] == $_SESSION['img1'] ||
                               $_POST[$i] == $_SESSION['img2'] ||
                               $_POST[$i] == $_SESSION['img3'] )
                                     $_SESSION['xflagcheck']++;

                         }
                         if($_SESSION['xflagcheck'] == 0)
                               $_SESSION['user_login_status2']++;
                         else
                               $_SESSION['user_login_status2'] = 0;
                   }
      }

    /**
     * simply return the current state of the user's login
     * @return boolean user's login status
     */
    public function isUserImgLoggedIn()
    {
        if (isset($_SESSION['user_login_status2']) AND
$_SESSION['user_login_status2'] == 3) {
            return true;
        }
        // default return
        return false;
    }

      …
}
```

## A-4 Reload Detection

```
<?php
// show potential errors / feedback (from login object)
if (isset($imglogin)) {
    if ($imglogin->errors) {
        foreach ($imglogin->errors as $error) {
            echo $error;
        }
    }
    if ($imglogin->messages) {
        foreach ($imglogin->messages as $message) {
            echo $message;
        }
    }
      if (!isset($_SESSION['userinfo']))
            $imglogin->generateImg();
      else if ($_SERVER['REQUEST_METHOD'] === 'POST')
            $imglogin->generateImg();
      else{
            header("Location: index.php?refresh");
            exit;
      }

}
?>
```