

**FRAMEWORK TO ANALYSE SCRATCH PROJECT**

**BY**

**NGEOW BIH MAN**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfillment of the requirements**

**for the degree of**

**BACHELOR OF INFORMATION SYSTEMS (HONS)**

**INFORMATION SYSTEMS ENGINEERING**

**Faculty of Information and Communication Technology**

**(Perak Campus)**

**MAY 2016**

## REPORT STATUS DECLARATION FORM

**Title:** FRAMEWORK TO ANALYSE SCRATCH PROJECT

**Academic Session:** MAY 2016

I NGEOW BIH MAN 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)

\_\_\_\_\_  
(Supervisor's signature)

**Address:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

\_\_\_\_\_  
Supervisor's name

**Date:** \_\_\_\_\_

**Date:** \_\_\_\_\_

**FRAMEWORK TO ANALYSE SCRATCH PROJECT**

**BY**

**NGEOW BIH MAN**

**A REPORT**

**SUBMITTED TO**

**Universiti Tunku Abdul Rahman**

**in partial fulfillment of the requirements**

**for the degree of**

**BACHELOR OF INFORMATION SYSTEMS (HONS)**

**INFORMATION SYSTEMS ENGINEERING**

**Faculty of Information and Communication Technology**

**(Perak Campus)**

**MAY 2016**

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**FRAMEWORK TO ANALYSE SCRATCH PROJECT**” 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 acknowledge the support and guidance of my supervisor, Mr. Sun Teik Heng in this project. I would also like to express my deep sense of gratitude to him for given me this opportunity to learn and development a framework related to project analysis.

Last but not least, I am extremely thankful to my family and friends for their support and love throughout the course.

## **ABSTRACT**

This project is an application development project which establish a framework to analyse Scratch projects. The improvement of existing Scratch analysis tools is in need because the lack of result stability, data storage, and advance features in the existing systems. This proposed project develops a framework called Scratch School to analyse Scratch projects which involves new algorithm design to increase the accuracy of the result. The project scope includes database development, question development, analysis tool development, recommendations for analysis results, and website development. Agile methodology is used in the system development of the project. The project will break to several parts such as web page layout design, file storing and extracting, and database design. Subscription of users to the framework is proposed to ease the users store or retrieve their analysis results. The proposed project analysis methods are divided into two main groups, which are basic analysis and analysis based on question. Basic analysis is done based on general analysis of the projects. The requirements in the selected question is considered when perform analysis based on question. The questions are created by users and other users can submit their answers to perform analysis based on question. Users need to submit answer based on question's requirements to achieve high mark. The details of the project will be discussed further in this report.

# TABLE OF CONTENTS

<b>TITLE</b>	<b>i</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>viii</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xi</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Problem Statement	1
1.2 Background	1
1.3 Motivation	4
1.4 Project Objectives	4
1.5 Proposed Approach	5
1.6 Project Scope	6
1.7 Impact, Significance and Contribution	6
1.8 Report Organization	7
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>8</b>
2.1 Background Information	8
2.2 Hairball Architecture	8
2.3 Dr. Scratch Web Application	11
2.4 Computational Thinking Concepts	14
2.5 Critical Review	15
2.6 System Comparison	16
<b>CHAPTER 3 SYSTEM DESIGN</b>	<b>18</b>
3.1 Requirements Determination	18
3.1.1 Functional Requirements	18
3.1.2 Non-functional Requirements	18

3.2	Block Diagram	19
3.3	Flowchart	20
3.4	Use Case Diagram	21
3.5	Sequence Diagram	22
3.5.1	Sequence Diagram of User Login	22
3.5.2	Sequence Diagram of Analyse Project	23
3.6	Data Flow Diagram	24
3.6.1	Data Flow Diagram Level 0	24
3.6.2	Data Flow Diagram Level 1	25
3.7	Scratch Project Analysis Marking Scheme	25
<b>CHAPTER 4</b>	<b>METHODOLOGY AND TOOLS</b>	<b>30</b>
4.1	Agile Methodology	30
4.2	Technology Involved	31
4.2.1	Hardware Requirements	31
4.2.2	Software Requirements	31
4.2.3	Programming Language	32
4.2.4	Markup Language	32
4.2.5	Layout Template	32
4.3	System Performance Definition	32
4.4	Timeline	33
4.5	Implementation Issues and Challenges	36
<b>CHAPTER 5</b>	<b>SYSTEM IMPLEMENTATION</b>	<b>37</b>
5.1	System Publish	37
5.2	Screenshot	38
<b>CHAPTER 6</b>	<b>TESTING</b>	<b>44</b>
6.1	Navigation Bar	44
6.2	Home Page	44
6.3	About Page	45
6.4	Quiz Page	45
6.5	Create Question Page	45



6.6	Register Page	46
6.7	Login Page	46
6.8	Analysis Result Page	47
6.9	Complexity Level Scoring	47
6.9.1	Motion Score	47
6.9.2	Looks Score	48
6.9.3	Sound Score	48
6.9.4	Pen Score	48
6.9.5	Data Score	49
6.9.6	Event Score	49
6.9.7	Control Score	49
6.9.8	Sensing Score	50
6.9.9	Operators Score	50
6.10	CT Concepts Level Scoring	51
6.10.1	Logic Score	51
6.10.2	Algorithms Score	51
6.10.3	Decomposition Score	52
6.10.4	Patterns Score	52
6.10.5	Abstraction Score	52
6.10.6	Synchronisation Score	53
6.11	System Testing	53
<b>CHAPTER 7</b>	<b>CONCLUSION</b>	<b>55</b>
7.1	Future Development	55
<b>BIBLIOGRAPHY</b>		<b>57</b>

## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 1-1	A sample Scratch project.	3
Figure 1-2	Analysis result of the sample project.	3
Figure 2-1	Different competence levels of data representation: basic (a), developing (b) and proficient (c).	13
Figure 3-1	Block diagram of Scratch School.	19
Figure 3-2	Flowchart of Scratch School.	20
Figure 3-3	Use case diagram of Scratch School.	21
Figure 3-4	Sequence diagram of user login.	22
Figure 3-5	Sequence diagram of project analysis process.	23
Figure 3-6	Level 0 Data Flow Diagram (DFD).	24
Figure 3-7	Level 1 Data Flow Diagram (DFD).	25
Figure 4-1	Flow of Agile development.	30
Figure 4-2	Part 1 of Gantt chart.	33
Figure 4-3	Part 2 of Gantt chart.	34
Figure 5-1	Scratch School publish in Microsoft Azure.	37
Figure 5-2	Publishing project in Visual Studio.	37
Figure 5-3	Home page of Scratch School.	38
Figure 5-4	Quiz page of Scratch School.	38
Figure 5-5	Create Question page of Scratch School.	39
Figure 5-6	Register page of Scratch School.	39
Figure 5-7	Login page of Scratch School.	40
Figure 5-8	Part 1 of Analysis Result page 1 of Scratch School.	40
Figure 5-9	Part 2 of Analysis Result page of Scratch School.	41
Figure 5-10	About page of Scratch School.	41
Figure 5-11	My Question page of Scratch School.	42
Figure 5-12	Upload Answer page of Scratch School.	42
Figure 5-13	User Profile page of Scratch School.	43

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 2-1	Example of relative and absolute modification blocks in five categories of initial state.	9
Table 2-2	Competence level for each CT concept.	12
Table 2-3	Comparison between benchmarked systems and proposed system.	16
Table 3-1	Analysis marking scheme in complexity level.	26
Table 3-2	Analysis marking scheme in CT concepts level.	28
Table 3-3	Overall score required to achieve different project levels.	29
Table 6-1	Navigation bar testing results.	44
Table 6-2	Home Page testing results.	44
Table 6-3	About Page testing results.	45
Table 6-4	Quiz Page testing results.	45
Table 6-5	Create Question Page testing results.	45
Table 6-6	Register Page testing results.	46
Table 6-7	Login Page testing results.	46
Table 6-8	Analysis Result Page testing results.	47
Table 6-9	Motion score testing results.	47
Table 6-10	Looks score testing results.	48
Table 6-11	Sound score testing results.	48
Table 6-12	Pen score testing results.	48
Table 6-13	Data score testing results.	49
Table 6-14	Event score testing results.	49
Table 6-15	Control score testing results.	49
Table 6-16	Sensing score testing results.	50
Table 6-17	Operators score testing results.	50
Table 6-18	Logic score testing results.	51
Table 6-19	Algorithms score testing results.	51
Table 6-20	Decomposition score testing results.	52
Table 6-21	Patterns score testing results.	52

Table 6-22	Abstraction score testing results.	52
Table 6-23	Synchronisation score testing results.	53
Table 6-24	Analysis results of random picked students.	53

## **LIST OF ABBREVIATIONS**

<i>CS</i>	Computer Science
<i>CT</i>	Computational Thinking
<i>DFD</i>	Data Flow Diagram
<i>JSON</i>	JavaScript Object Notation
<i>URL</i>	Universal Resource Locator

## CHAPTER 1 INTRODUCTION

### 1.1 Problem Statement

At present, analysis tool such as Hairball automated system is used to analyse Scratch projects automatically. Hairball system is used to analyse Scratch projects as well as determine potential errors or unsafe practices in the projects. However, not all of the plug-ins provided by Hairball are fully tested and their outputs are not yet standardised. A reliable framework with standardised results to help with the analysis of Scratch projects is needed to solve the problem. Another problem is the existing systems do not allow previous analysed results to store or retrieve by user. The previous results are required for users to perform self-learning. The database that is used to store prior results is needed to solve this problem. Furthermore, the existing systems only evaluate Scratch projects based on projects uploaded by users. Since Scratch project is widely used in educational purpose, availability of questions that test on students' programming skills is important. Therefore, a framework that is specifically designed to analyse projects based on properly designed questions is needed to help the educators in examining students' programming skills.

### 1.2 Background

Many people are facing a hard time when they learn to develop computer programs by using text-based programming language such as Java, C++ and C#. They need to study and follow the structure and keywords of these programming languages before designing a program that will perform their desired tasks. Therefore, Scratch programming is initiated and it is designed especially for students who ages 8 to 16. Scratch is a visual programming language that creates a fun and easy way of learning computing and software development. It enables users to create interactive stories, games and animations and share their works through internet. Program is developed by using a simple "drag and drop" interface (Colleen n.d.). It is used by people around the world as it is available in more than 40 languages. Scratch programming is suitable to use as the first programming language for beginners and no prior programming knowledge is needed.

Scratch projects can develop through online or offline. Scratch 1.4 and Scratch 2.0 are available for users to install in their computer to construct Scratch projects offline. The blocks are grouped into several types of scripts such as motion, looks and sound which carry out different functions. The function of the blocks is easy to be understood by using simple words to represent the use of the blocks. The objects in Scratch project are called sprite which users can increase the interactivity of the project by adding events to sprite. Sprite, sound and backdrop can be imported to the project instead of using provided image or sound from the build-in library. However, the evaluation of visual projects is harder than traditional text-based programming projects.

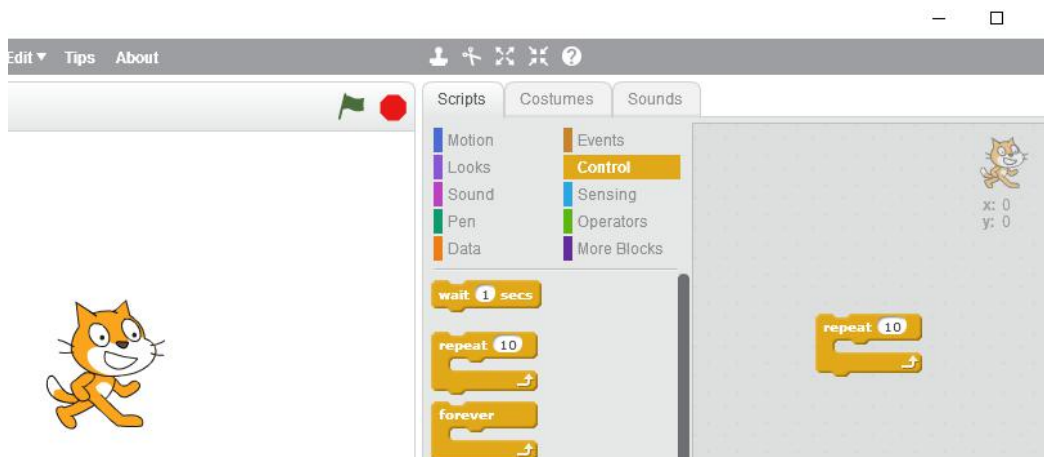
The program will start to run from main structure in text-based programming projects whereas there is no main structure exist in Scratch projects which increase the difficulty of evaluation. Each project need to be individually opened and run to evaluate Scratch project (Boe et al. 2013). Different combinations of blocks can generate the same result which make the evaluation process hard and complex. In one study (Moffat & Wilson 2010), Scratch projects were evaluated manually at primary school and grading based on marks rewarded in a test. The test was conducted to evaluate students whether they can move the sprite to pass through specific places. Students need to write down their instruction in a worksheet. Marks were given based on the progress of the students managed to achieve. The students' projects are evaluated as follows:

- Able to draw a line from the starting point to the ending point
- Able to identify correct instructions that get the sprite to the end
- Perform correct turns with correct direction and degrees

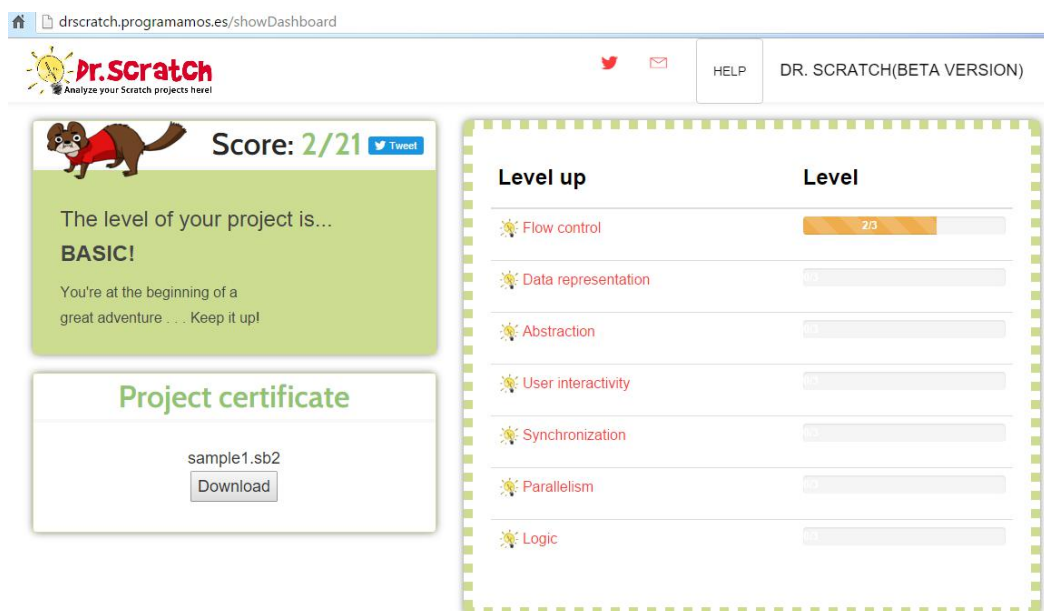
An analysis tool that evaluates Scratch projects is difficult to develop as visual programming gives freedom to users by using different programming structure methods.

Dr. Scratch is a website that is used to analyse Scratch projects from projects' URL or project files uploaded by users (Dr. Scratch 2014). The grading system is based on computational thinking skills such as logical thinking, flow control and user interactivity. Feedback will be given after the analysis completed to improve user's coding skill. Unfortunately, users cannot subscribe to the website to

compare their current works with previous results. A sample Scratch project has been created as shown in Figure 1-1, which only consists of an empty repeat loop. This project is uploaded to Dr. Scratch and obtained 2 points in flow control as shown in Figure 1-2. The evaluation method is formed based on identifying specific blocks of statement for each element in the project. This activity has proven Dr. Scratch's evaluation method is not reliable as 2 points can be rewarded for an empty repeat statement.



**Figure 1-1: A sample Scratch project.**



**Figure 1-2: Analysis result of the sample project.**

Another analysis tool called Hairball is a plugin-able framework uses to analyse Scratch projects. There were plug-ins provided by Hairball to perform the analysis but some plug-ins such as checks.Animation, checks.SaySoundSync and



initialization. Variable Initialization are not fully tested and their outputs are not yet standardised (GitHub 2016). Therefore, improvement on analysis tool of Scratch projects is highly in need.

### 1.3 Motivation

The motivations to initiate the project are due to the following issues:

- The benchmarked system analyses Scratch project in 4 areas, which are initiation, broadcast and receive, say and sound synchronization, and animation. Scores are rewarded without checking on the empty block. For example, 1 mark is rewarded when an if block is detected under logical thinking concept. However, the correct usage of if block is not examined. Therefore, users cannot rely on existing system to improve their Scratch projects.
- The result of evaluation is shown by computing computational thinking (CT) score and classify the analysed project in 3 different programming levels in benchmarked system. Even though the system is able to show CT score in 3 different programming levels, but it does not allow the results to be saved and retrieved by user. So, they cannot compare their current performance with their past performance.
- Educators like to check students' project based on specific rules or themes but existing systems do not evaluate projects based on specific question. Therefore, framework that is use to evaluate students' work based on specific question is in need.

### 1.4 Project Objectives

- To provide users a reliable analysis tool for Scratch projects by designing a new grading algorithm that generate standardised results.
- To allow users to store and retrieve their previous analysis results for self-learning purpose.
- To allow users to create questions and analyse the Scratch projects based on the specific question.

## 1.5 Proposed Approach

The main delivery of this project is a project analysis web application system which can be accessed easily by public. This system only support projects of Scratch 2.0, which the project information will be encoded in a JSON file. This system uses the ease of reading JSON file to evaluate Scratch projects. The system convert the .sb2 file which is the extension of Scratch 2.0 projects to zip file format to retrieve the project's JSON file.

Users need to connect to the internet to access the web application system. The system will be hold in the Microsoft Azure cloud computing platform to manage the web service and related databases. This approach help in eliminate the connection failure problem because Microsoft Azure will manage the system connection.

The proposed system allow users to build their own Scratch questions in the provided web form. Users are able to create the question by selecting category of block, and action needed to perform by the block, and the desired outputs. However, the inputs are limited for some statements to increase the accuracy of the project evaluation. For example, only integer inputs are allowed for the move statement during question creation.

Project is evaluated by detect the availability of specific blocks in the project. The system will check for the null blocks which the block containing empty necessary input, and no mark will be rewarded by these blocks. This approach is used to encourage the users to focus on the correct usage of block. The system will only detect the fully null blocks, not the partially null blocks, which means if the block need two necessary inputs, only one input is available will be considered correct.

## 1.6 Project Scope

This project develops a framework to analyse Scratch projects which involves new algorithm design to increase the accuracy of the result. The project's scopes include:

- Website development  
A website will be created to allow user to upload their Scratch projects for evaluation and displaying of the results.
- Developing an analysis tool  
An analysis tool will be developed to analyse Scratch projects based on the classification of blocks in Scratch and CT concepts.
- Provide recommendations/ suggestions for Scratch projects  
Suggestions will be given on how to improve users' Scratch projects based on evaluation score.
- Developing questions for Scratch projects  
User will be able to create question for Scratch project and the project will be evaluated based on fulfilling the question's requirements.
- Development of database  
Subscribed users are able to save and retrieve analysis results. The system enables users to retrieve their past analysis results for self-benchmarking purpose.

## 1.7 Impact, Significance and Contribution

This project can benefit Scratch users by providing them a better solution to analyse their projects. This enhanced analysis tool that can check the correctness of program logic and provide more accurate and objective scoring system. Users are allowed to generate their own test questions. Therefore, users can examine their understanding skill, problem solving skill, and Scratch programming skill by developing projects for specific question. Moreover, students can learn from their historical results by login to their profile. By having this framework, students will have a reliable analysis tool of Scratch project. The Scratch project can be studied, improved and analysed easily by using this proposed framework. The availability of a good analysis tool makes improvement of Scratch project easier and it helps in future enhancement on the existing system.

## **1.8 Report Organization**

This project consists of 7 chapters as below:

Chapter 1: Introduction

Chapter 2: Literature Review

Chapter 3: System Design

Chapter 4: Methodology and Tools

Chapter 5: System Implementation

Chapter 6: Testing

Chapter 7: Conclusion

## CHAPTER 2 LITERATURE REVIEW

### 2.1 Background Information

Nowadays, there are various ways proposed to develop the interest in computer science among the youths and students. Visual programming languages such as Scratch are proposed to attract youths to learn programming skill in a fun way. Scratch provided an easy method for beginners of programming to create stories, animations and games by the use of drag and drop logic blocks. However, educators are facing difficulty in evaluating student programs and assessing their CT skills due to the lack of capability from the existing tool (Moreno-León, Robles & Román-González 2015). CT is a skill that involves problem solving, systems design and understanding of human behaviour which introduce the fundamental concepts to computer science (Moreno-León & Robles 2015). The advantages of automated checking are consistent, thorough and efficient (Isong 2001). As a result, same level of efficiency and same standards of resulting assignment grade will be given to all checked programs (Isong 2001). Several authors have come out with different approaches to evaluate the Scratch projects for learners. Some of the approaches are Hairball architecture and Dr. Scratch web application.

### 2.2 Hairball Architecture

Hairball is an automated system to detect errors or unsafe practices within a Scratch project and analyse the implementation of Scratch projects. An initial set of Hairball plugins have been designed by Boe et al that distinguish and mark instances of initialization of Scratch state, “say and sound” blocks synchronization, “broadcast and receive” blocks synchronization, and timing and iteration control for complex animation (Boe et al. 2013). The object-oriented features of Python are used to create a major class from which Hairball plugins can be derived. Extending the major class and overloading a single method are required to implement a Hairball plugin. In general, anyone with basic Python programming skills can implements any Hairball plugins that perform various type of static analysis of a Scratch program.

Hairball is not implemented to replace the manual analysis but it’s used to fasten the analysis by identify “easy” cases automatically. This is because Scratch

project is sensory in nature so some analysis still need to perform manually to obtain better results. This study showed that Hairball is very useful when operate with manual analysis. Moreover, Hairball system is able to identify whether Scratch programs containing certain targeted constructs or blocks.

Hairball can be used to check for the following instances (Boe et al. 2013):

- Detect unmatched “broadcast and receive” blocks
- Identify infinite loops of broadcast/ receive events in the programs
- Make sure proper initialisation of initiate state in the programs
- Make sure proper implementation of complex animations

The initialization plugin is used to analyse the initialization of Scratch state. The initiate state is started to evaluate when “when green flag clicked” block encounter. The set of blocks which are defined as initialization blocks are determined and restricted to appear in initialization zone only. The initialization zone starts from the beginning of the script until a relative block or a broadcast block is encounter. Attribute modifying Scratch blocks are divided into two categories which are relative and absolute. Relative Scratch blocks change the attribute value based on its existing value whereas absolute Scratch blocks set the attribute value directly. Initialization blocks consist of absolute blocks only. Table 2-1 shown categorization of attribute modifying Scratch blocks based on relative and absolute characteristics. When an absolute block for the same attribute occurs in the initialization zone, a modified attribute of a sprite is said to be initialised correctly by initialization plugin.

Category	Relative	Absolute
Costume	Next costume	Switch to costume x
Visibility		Show/hide
Orientation	Turn clockwise x degrees	Point in direction x
Position	Move x steps, go to x, y, glide z seconds to x, y, etc.	Go to x, y
Size	Change size by x%	Set size to x%
Background	Next background	Switch to background x

**Table 2-1: Example of relative and absolute modification blocks in five categories of initial state.**

Furthermore, the say/sound synchronization plugin is used to detect instances of synchronization between “say and sound” blocks in the project. The synchronization between “say and sound” blocks is desired while a message appears, and an audio plays at the same time. The speech bubble disappears as the sound is completed. The plugin detects instances of this concept by searching for sequential “say and sound” blocks and verify whether the instances are implemented using the appropriate method. The appropriate method is referring to use a “say” block to show the message, followed by a “play sound until done” block, and ending with a blank “say” block to clear the prior speech bubble. These three blocks are arranged in a proper order as stated above is considered a correct instance.

Next, the broadcast and receive plugin is used to detect the proper trigger of the execution of other sprites’ scripts by using “broadcast and receive” blocks. The plugin verifies that there is a broadcast block and one or more corresponding receive block for each broadcast or receive event, and it is recognised as a correct instance.

Finally, complex animation plugin is used to identify whether complex animation is implemented properly in the project. Rotation blocks or motion blocks combined with costume change blocks are necessary components of a complex animation instance. A complex animation instance is defined as a loop or a sequence that containing these necessary components. An instance must also include Scratch block that introduces a delay to be labelled correct. The study shown Hairball is extremely useful in identifying correctly implemented instances. Hairball is proposed as an addition to, but not replacement of, manual analysis as Hairball can decrease analysis duration.

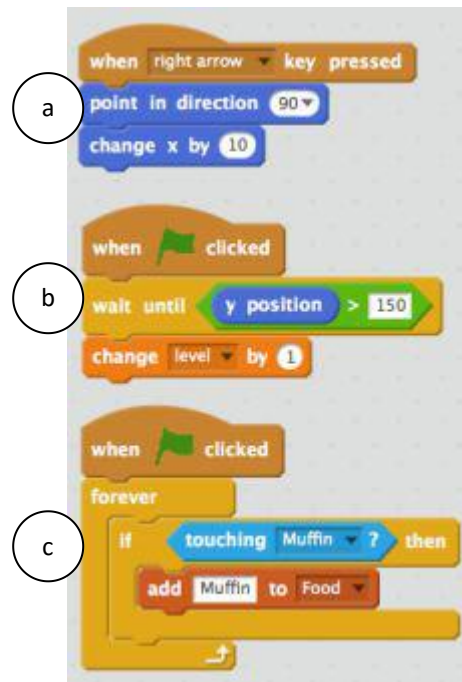
### 2.3 Dr. Scratch Web Application

Dr. Scratch is an open source web application which enables students to analyse their Scratch projects easily by using Hairball plugins. Besides that, users can obtain feedback from the application to improve their codes and develop their CT skills. It is also designed to encourage students to improve their programming skills. A .sb or .sb2 file is uploaded to analyse project by using Dr. Scratch. The alternative way is users directly insert a project URL into provided text box for project analyse. Degree of development of CT and CT score are assigned to user's project when the project is analysed. It also detects certain bad habits of programming or potential errors such as code repetition, and code that is never executed throughout the project based on Hairball plugins. CT score is assigned based on 7 concepts which are abstraction and problem decomposition, logical thinking, synchronisation, parallelism, algorithmic notions of flow control, user interactivity and data representation (Moreno-León, Robles & Román-González 2015). The CT score is calculated by adding up the partial scores of each CT concept (Moreno-León & Robles 2015). The project level is determined by the amount of CT score achieved. A CT score of 7 points and below are classified as Basic level. A CT score between 8 and 14 points are classified as Developing level. A CT score of 15 points and above are evaluated as Proficient level. The competence level of these concepts is evaluated by following the rules listed in Table 2-2. Dr. Scratch will show basic information only for the users to improve their code if the CT level is low. If the CT level is high, users can obtain all available information in terms of CT skills and bad programming habits. Example of different project competence levels were shown in Figure 2-1. The script labelled (a) categorised as basic level, the script labelled (b) categorised as developing level and the script labelled (c) categorised as proficient level.



CT Concept	Competence Level			
	Null (0)	Basic (1 point)	Developing (2 points)	Proficiency (3 points)
<b>Abstraction and problem decomposition</b>	-	More than 1 script and more than 1 sprite	Definition of blocks	Use of clones
<b>Parallelism</b>	-	2 scripts on green flag	2 scripts on key pressed, 2 scripts on sprite clicked on the same sprite	2 scripts on when I receive message, create clone, 2 scripts when %s is > %s, 2 scripts on when backdrop change to
<b>Logical thinking</b>	-	If	If else	Logic operations
<b>Synchronisation</b>	-	Wait	Broadcast, when I receive message, stop all, stop program, stop programs sprite	Wait until, when backdrop change to, broadcast and wait
<b>Flow control</b>	-	Sequence of blocks	Repeat, forever	Repeat until
<b>User interactivity</b>	-	Green flag	Key pressed, sprite clicked, ask and wait, mouse blocks	When %s is > %s, video, audio
<b>Data representation</b>	-	Modifiers of sprites properties	Operations on variables	Operations on lists

**Table 2-2: Competence level for each CT concept.**



**Figure 2-1: Different competence levels of data representation: basic (a), developing (b) and proficient (c).**

A study carried out by Moreno-León, Robles and Román-González showed that students are able to improve their code by reading the information provided in the feedback report. Most of the students agreed that the website is attractive and easy to use. The students are able to understand the information displayed in the result page and are willing to improve on their programming skills after using Dr. Scratch. The analysis shows that there is an increase in the CT score of students for those who read the feedback report. This had proved Dr. Scratch has the ability to improve programming skills of students.

## 2.4 Computational Thinking Concepts

Computational thinking is a technique used to solve a problem by using a computer way. A computer cannot think and act by itself. We need to give instructions to the computer to let it complete the work for us. So, it is important for the student to improve their CT skill while developing Scratch projects. The CT skill level of the student affects the quality of the project produced.

There are several CT concepts introduced by Barefoot Computing, which are listed below (Barefoot Computing 2014):

- Logical reasoning: predict and analyse problems
- Algorithms: forming step-by-step instructions and rules
- Decomposition: break a large problem into smaller parts
- Abstraction: eliminate unnecessary details, simplify the problem
- Patterns and generalisation: identify similarities
- Evaluation: making decision

The output of a computer is predictable. Two different computers which are being set, and given the same inputs will produced the same outputs at the end. Students can use the logical reasoning skill by giving the specific instruction to the computer to obtain the specific outcome. Possible problems can be predicted by having a good logical reasoning skill.

It is important to having a clear instructions and rules in mind to develop a project to reduce possible error occur. Flowchart is a formal diagram which show the sequence of instruction the system should produced to achieve the desired output. Students can improve their algorithm skill by listing down the instructions needed to produce a project.

Decomposition is important when dealing with large project. The large project need to break down into smaller parts to enable teamwork, and make task easier. Work can be divided among students to let them work as a team to increase project quality, and decrease project duration.

The complexity of project is managed by abstraction skill. Abstraction skill enable students to focus on the key events of a project. It simplify complex project by get rid of details of the project activities.

By finding similarities or common in a project can make the project processes easier. The patterns and generalisation of a project is important for the students to perform repeat instructions or tasks.

CT concepts are necessary for a student to produce quality projects. CT concepts can provide students correct, and easier ways to solve a problem by using computer. These concepts will ease the students to learn text-based programming languages with strong logical thinking skill.

## **2.5 Critical Review**

Hairball is an extendable framework that can analyse Scratch projects automatically. Initial set of plugins are provided to analyse Scratch based on competence in computer science (CS) concepts. Potential errors and bad practices in the project are identified by Hairball. The plugins architecture of Hairball is in Python, so evaluators who are not familiar with Python will find it hard to perform further analysis on the students' projects by creating additional plugins. Therefore, this proposed project, Scratch School will provide a web platform to allow users to upload Scratch project for evaluation which users can evaluate their projects without any programming skills.

Dr. Scratch is a web tool that can analyse Scratch projects automatically. CT score and potential errors are calculated and detected as a result of evaluation, which learners can improve their coding skill and CT skills by learning from their mistakes. The website is attractive to students and information display is understandable by them. However, learners who initially obtained high score in Dr. Scratch seem cannot improve their skills by learning from result report (Moreno-León, Robles & Román-González 2015). The information in the feedback report need to be enhanced to help advance users to develop their skills. Scratch School will provide recommendation to users based on CT score of their

projects. The evaluation method proposed in this project will include CT concepts with advance checking on the content in each block.

Scratch School is inspired by Dr. Scratch. Recommendation will be given in the feedback report and potential errors in the code are detected. An interesting web interface will be considered to attract primary and secondary schools' students to use the proposed analysis tool. The grading system of Scratch School is based on classification of blocks in Scratch and CT concepts. Scratch School allows users to create user accounts to keep the log of their analysis. Scratch project questions are designed by users. Users can select and answer these questions from Scratch School to examine their programming skill, and problem solving skill. Scratch School will enhance existing analysis tools and advance the system by enable question development, and evaluation by educators.

## 2.6 System Comparison

After reviewing the benchmarked systems, a comparison has been made on the reviewed systems and proposed system as shown in Table 2-3.

Criteria	Hairball	Dr. Scratch	Scratch School (Proposed system)
Free	Yes	Yes	Yes
User Interactivity	No	Yes	Yes
Connectivity	No	Yes	Yes
Database development	No	No	Yes
Scratch project question	No	No	Yes
Analysis Recommendation	No	Yes	Yes
CT concepts	No	Yes	Yes

**Table 2-3: Comparison between benchmarked systems and proposed system.**

Proposed system and benchmarked systems are free to use by download from website or access through the internet. Dr. Scratch and Scratch School provide user interactivity through web services whereas Hairball does not provide user

interactivity. Dr. Scratch and Scratch School need internet connection to execute whereas Hairball can be used in offline mode. Both of the benchmarked systems do not provide database for user to store data whereas Scratch School provide database to store user account and analysis result. Scratch School enable users to create Scratch project questions whereas benchmarked systems do not provide this service. Both Dr. Scratch and Scratch School provide analysis recommendation, and analysis result include CT concepts.

## CHAPTER 3 SYSTEM DESIGN

### 3.1 Requirements Determination

#### 3.1.1 Functional Requirements

- The system should be able to evaluate Scratch project for subscribed users and unsubscribed users.
- The system should enable users to select question before project analysis.
- The system should store the analysis results of subscribed users.
- The system should display recommendations or suggestions in the analysis results.
- The system should enable users to create Scratch questions.
- The system should display analysis result after evaluation.
- The system should display project level after evaluation.

#### 3.1.2 Non-functional Requirements

- Look and feel  
The system shall appear to be simple to use by students.
- Usability and humanity  
The system shall be easy to use by primary and secondary school students.
- Performance  
The system shall generate accurate analysis results.
- Operational  
The system shall be able to execute in day and night with internet connection.
- Maintainability and support  
The system shall be able to support Internet Explorer, Mozilla Firefox, and Google Chrome web browsers.
- Cultural  
The system shall not display religious symbols or words.
- Legal  
The system shall not break any international laws and regulations.

### 3.2 Block Diagram

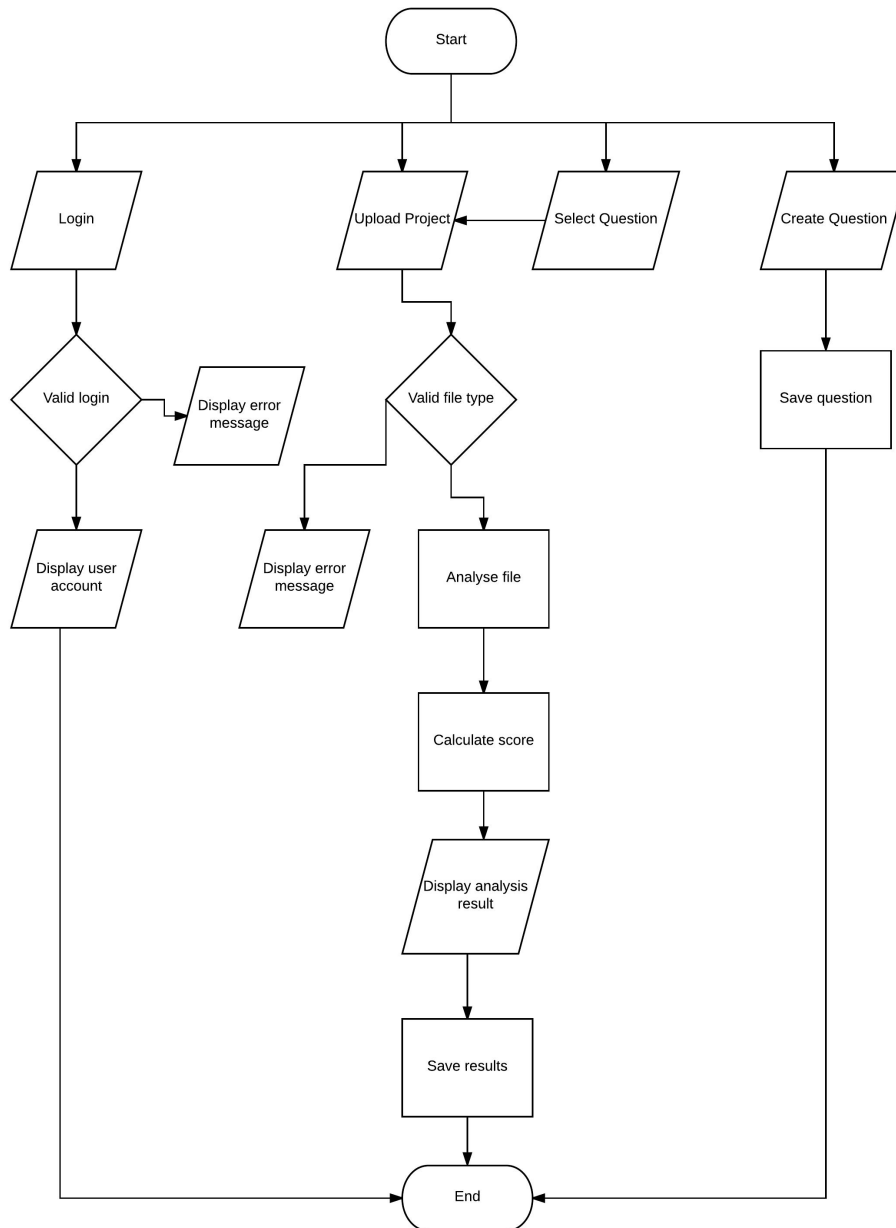


**Figure 3-1: Block diagram of Scratch School.**

Figure 3-1 shows the block diagram of this proposed framework. The block diagram shows the main functions performed by the framework. Users are able to select a question and upload a project based on the selected question. The project uploaded is analysed and an analysis result will be generated. Users also can perform analysis of Scratch projects without selecting any question. There is a side function which enables users to create Scratch questions in the proposed system.



### 3.3 Flowchart

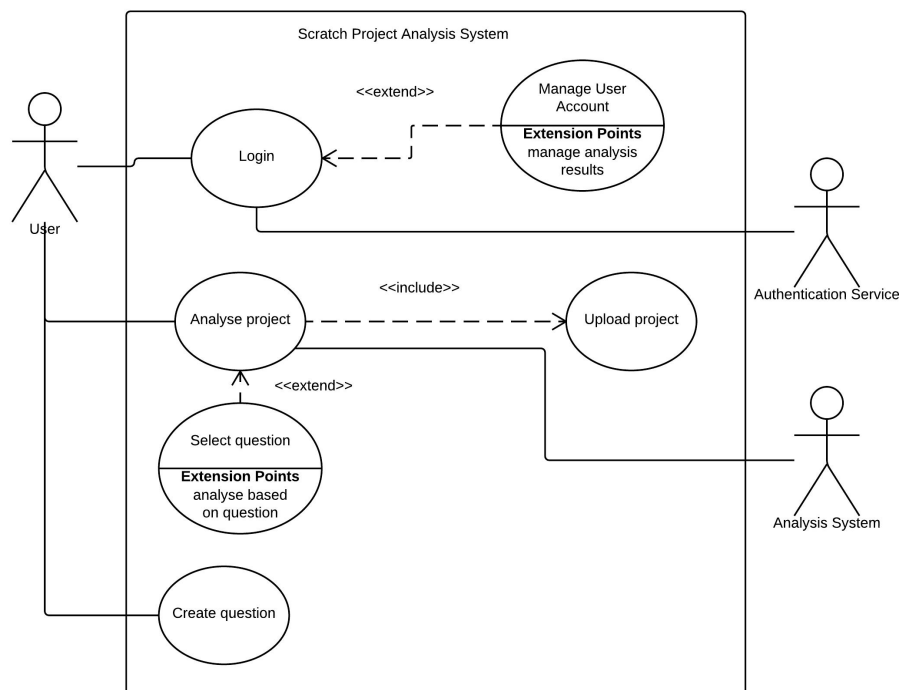


**Figure 3-2: Flowchart of Scratch School.**

Figure 3-2 shows the flowchart of the proposed system. Users can register or login as subscribed users or directly access the framework as anonymous. Subscribed and unsubscribed users can perform the analysis for Scratch project. The analysis can be divided into two groups, which are basic analysis and

analysis based on selected question. Basic analysis is done based on general analysis of the projects. The requirements in the selected question is considered when perform analysis based on question. Users can pre-select a question displayed in the framework and upload the answer to perform the second analysis method. The file type of the uploaded project is checked before performing analysis. The analysis results of subscribed users are stored in their profiles. Users can create question and question created will store in the system database.

### 3.4 Use Case Diagram

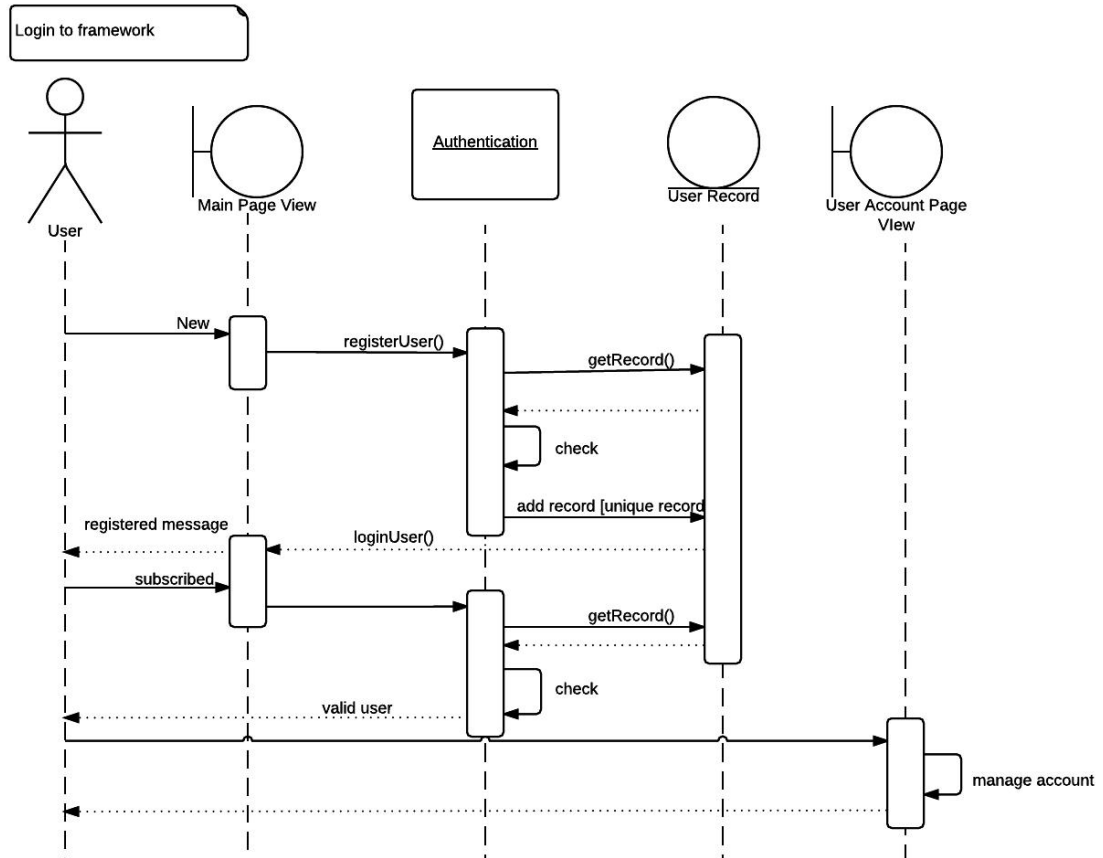


**Figure 3-3: Use case diagram of Scratch School.**

Figure 3-3 shows the use case diagram of proposed system. Users login to the framework and they can manage their account after successful login. Users can analyse their Scratch projects by upload their projects to the framework. They need to select a Scratch question if they wish to evaluate their projects based on specific question. Users can create question and display it in the system for question selection.

### 3.5 Sequence Diagram

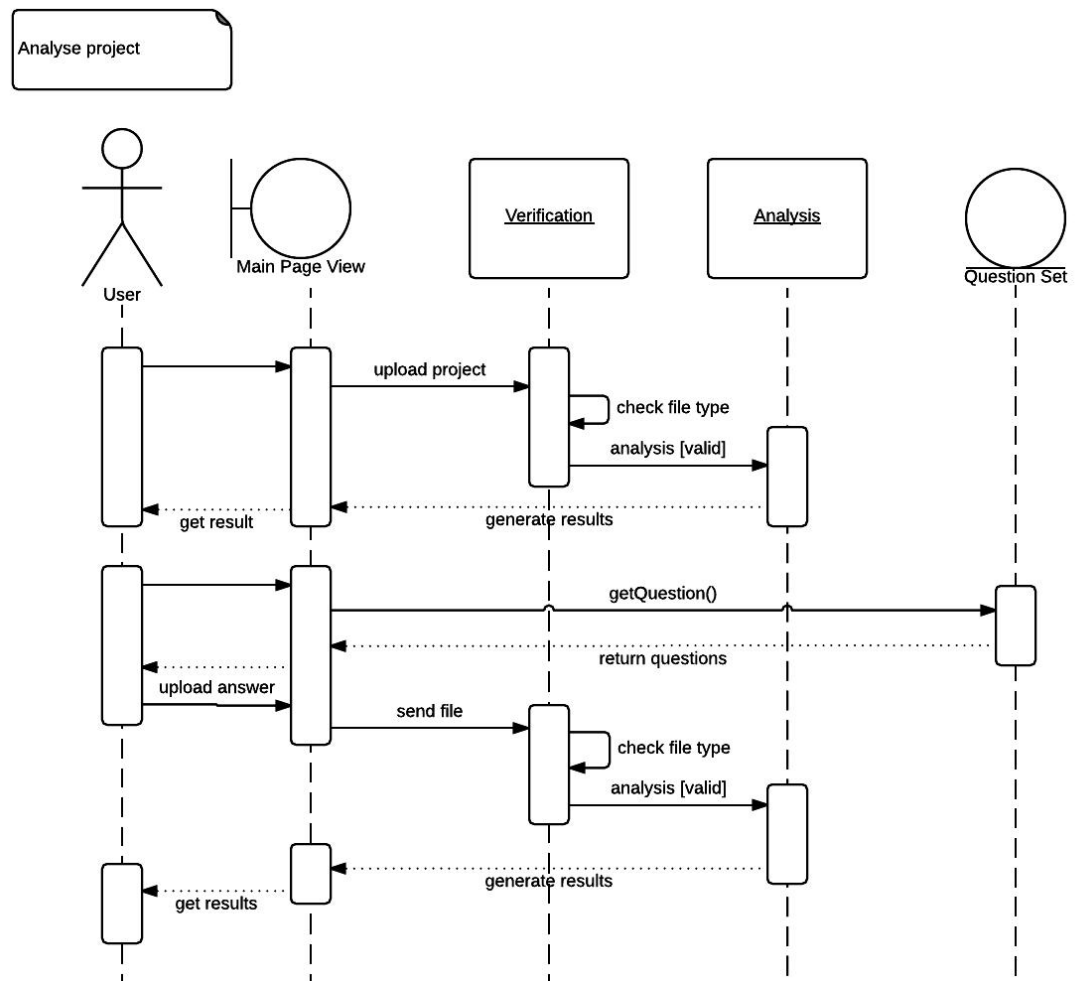
#### 3.5.1 Sequence Diagram of User Login



**Figure 3-4: Sequence diagram of user login.**

A new user will need to register to create new user account. Email needs to verify as unique and proper format email before a new user record is created. Existing users will login by providing email and password registered in this framework. Users can manage account details and view previous analysis results in the user account page after successful login. The sequence diagram of user login process is shown in Figure 3-4.

## 3.5.2 Sequence Diagram of Analyse Project

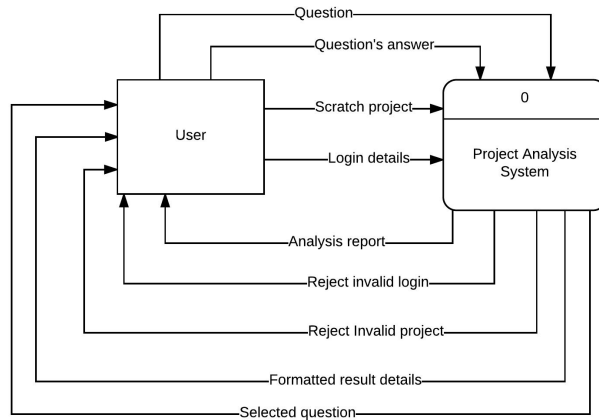


**Figure 3-5: Sequence diagram of project analysis process.**

The system need to verify the file type of the file uploaded by users before starting to analyse the project. Analysis will be carried out for valid file type. Analysis results will be generated and displayed to the users. Users can choose a question from the question set and upload project as an answer to the question. The analysis will be performed specifically based on the question's requirements. A sequence diagram is shown in Figure 3-5 to illustrate the project analysis process.

### 3.6 Data Flow Diagram

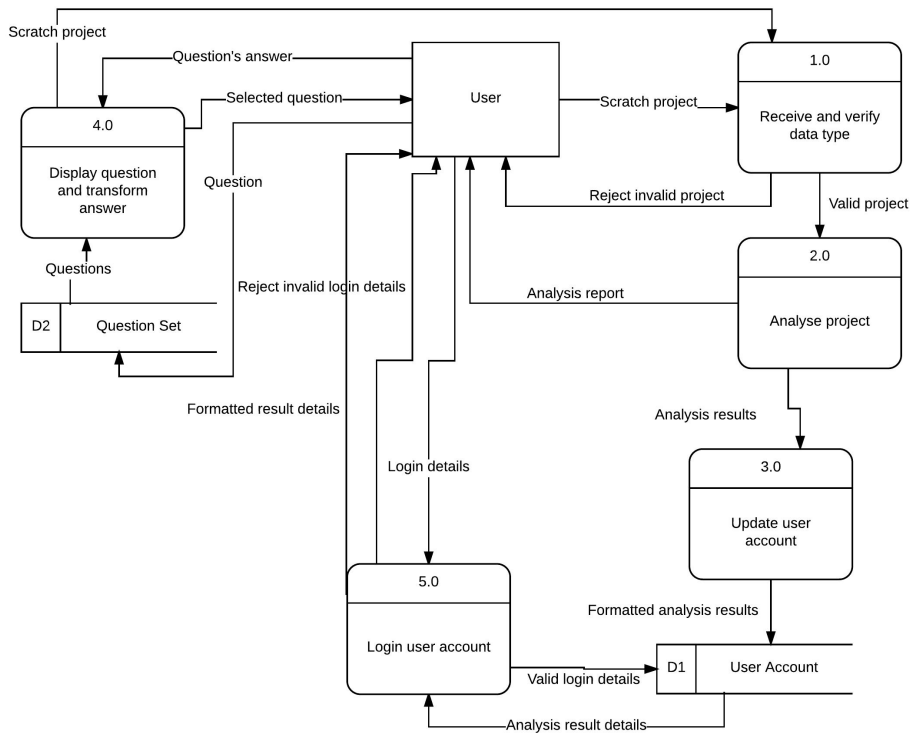
#### 3.6.1 Data Flow Diagram Level 0



**Figure 3-6: Level 0 Data Flow Diagram (DFD).**

Figure 3-6 shows the context diagram of DFD of proposed system. Users can insert four items to the project analysis system which are question, question's answer, Scratch project, and login details. Moreover, users can receive analysis report, formatted result details, selected question, reject invalid login and invalid project messages.

### 3.6.2 Data Flow Diagram Level 1



**Figure 3-7: Level 1 Data Flow Diagram (DFD).**

Figure 3-7 shows level 1 DFD of proposed system which is the detail view of context diagram of DFD. System will verify file type uploaded by users before analysis process is performed. An analysis report will be displayed to user and stored in user account after project analysis. A successful login will direct user to the user account page. Users can select question from the question set and upload their answers to the analysis system.

### 3.7 Scratch Project Analysis Marking Scheme

The evaluation method is created based on classification of blocks in Scratch to determine the complexity of projects. Moreover, CT concepts level of the students are determined based on their projects. Table 3-1 shows the analysis marking scheme in complexity level whereas Table 3-2 shows the analysis marking scheme in CT concepts level. It also shows the determinants allocated for each elements.

<b>Elements</b>	<b>Code</b>	<b>Marks</b>
<b>Motion</b>	move, turn left, turn right, point in direction, point towards	1
	go to, go to mouse-pointer, glide to, change x, set x, change y, set y	1
	set rotation style, bounce on edge	1
<b>Looks</b>	say, say for, think, think for, show, hide	1
	switch costume, next costume, switch backdrop	1
	change effect, set effect, clear graphic effect, change size, set size, go to front, go to back	1
<b>Sound</b>	play sound, play sound until done, stop all sound	1
	play drum, rest for, play note, set instrument	1
	change volume, set volume, change tempo, set tempo	1
<b>Pen</b>	clear, stamp, pen down, pen up	1
	change pen color, set pen color	1
	change pen shade, set pen shade, change pen size, set pen size	1

<b>Data</b>	set variable, change variable, show variable, hide variable	1
	add to list, delete from list, insert to list, replace item in list, item of list, length of list, list contains?, show list, hide list	1
<b>Events</b>	when “green flag” clicked	1
	when key pressed, when this sprite clicked, when backdrop switches, when larger	1
	when I receive, broadcast, broadcast and wait	1
<b>Control</b>	wait, repeat, forever	1
	if, if else, repeat until, wait until, stop	1
	when I start as a clone, create clone, delete this clone	1
<b>Sensing</b>	touching, touching color, color is touching, distance to	1
	ask and wait, key pressed?, mouse down?, mouse x, mouse y	1
	turn video, set video transparency, reset timer, days since 2000, username	1
<b>Operators</b>	+, -, *, /, pick random, mod, round or any mathematical calculation	1
	<, =, >, and, or, not	1
	join, letter of, length of	1
<b>Total</b>		<b>26</b>

**Table 3-1: Analysis marking scheme in complexity level.**



Elements	Mark of Codes		
	3	2	1
<b>Logic</b>	and, or, not	if else	if
<b>Algorithms</b>	repeat until	repeat, forever	use 2 or more sprites
<b>Decomposition</b>	add to list, delete from list, insert to list, replace item in list, item of list, length of list, list contains?, show list, hide list	set variable, change variable, show variable, hide variable	change x, set x, change y, set y, point in direction, go to, set rotation style, show, hide
<b>Patterns</b>	set x, set y, set rotation style, set effect, clear graphic effect, set size, set instrument, set volume, set tempo, clear, set pen color, set pen shade, set pen size, set video transparency, reset timer	when key pressed, when this sprite clicked, when backdrop switches, when larger, when I receive, broadcast, broadcast and wait	when “green flag” clicked
<b>Abstraction</b>	go to front, go to back	show, hide	switch costume, next costume, switch backdrop
<b>Synchronisation</b>	wait until, when backdrop switches, broadcast and wait	when I receive, broadcast, stop	wait
<b>Total</b>	<b>18</b>		

**Table 3-2: Analysis marking scheme in CT concepts level.**

The analysis result is divided into two groups, which are complexity level and CT concepts level. Each elements in complexity level has maximum 3 points to obtain except Data element for a maximum mark of 2 points. If the correct usage of code listed in the element category is detected, 1 mark will be allocated

to the respective element. A total of 26 marks can be obtained by the users in the complexity level section. Users need high understanding of the usage of each Scratch block to achieve high score in this section. Users can only skilled themselves in using Scratch blocks to combine different category of blocks in a project. The CT concepts levels are determining based on 6 CT concepts which are logic, algorithms, decomposition, patterns, abstraction, and synchronisation. Each elements in CT concepts level has maximum 3 points to obtain. A total of 18 marks can be obtained by the users in the CT concepts level section. The maximum overall mark will be 44 marks which is the addition of complexity level mark and CT concepts level mark. The project level will be determined by the total overall score obtain by the users which shown in Table 3-3.

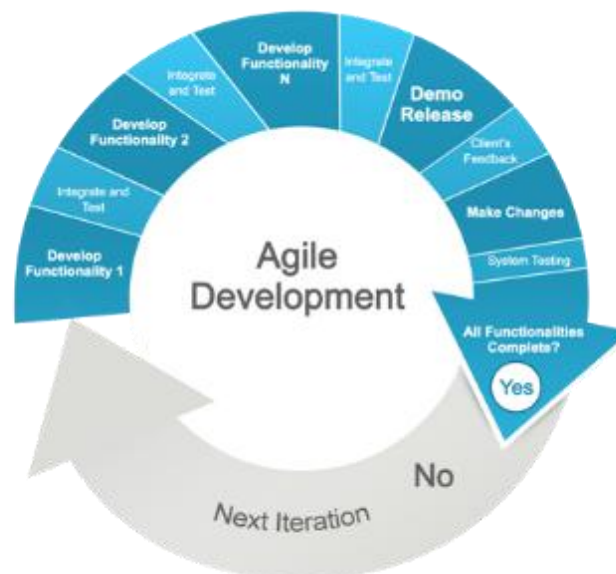
<b>Project Level</b>	<b>Score</b>
Basic	0 - 21
Developing	22 - 34
Proficient	35 - 44

**Table 3-3: Overall score required to achieve different project levels.**

## CHAPTER 4 METHODOLOGY AND TOOLS

### 4.1 Agile Methodology

Agile methodology is used in the system development of the project. It is an alternative to traditional project management which iterative and incremental process models are combined (Tutorialspoint 2016). A diagram as shown in Figure 4-1 to illustrate the flow of Agile methodology. Agile development method breaks the project into small increments and deliver in a defined duration. It provides faster product delivery and allow changes in any stage of the project. The project will break to several parts such as web page layout design, file storing and extracting, and database design. Each part will be completed in defined duration and changes in requirements can be performed. The testing and development of the system will be carried out concurrently to ease the identification of errors and problems of the system.



**Figure 4-1: Flow of Agile development.**

This project begins from study previous works done by other researchers to find out how others perform evaluation on Scratch projects before begin to develop this framework. After study previous works, this project will develop based on positive aspects achieved by others and solve limitations faced by other analysis tools. Problem statements, project objectives and expected innovation are set to foresee the expected outcome. A guideline used to evaluate Scratch projects and questions for Scratch project are form before the evaluation program is

develop. A website which enable users to upload file will be produce. The file received will be save as zip file and extract it when performing evaluation. Search JSON file throughout the extracted zip file and evaluate it by searching keywords listed in the evaluation guideline. Project score will be shown in result display page. A database will be formed to store result details and user's information. Login and registration pages will be formed for user to subscribe to the website. User detail's pages will be developed to display user information and previous result details. Quiz pages are created to display user created questions and enable user to evaluate Scratch project based on the selected question. Recommendations will be display in the result page after each project analysis. Interview and workshop are conducted among primary school's children and collected data is use to obtain the pre-testing results. The pre-testing results are used to show the Scratch programming standards of students before using the proposed system.

## **4.2 Technology Involved**

### **4.2.1 Hardware Requirements**

- Computer  
System type: Laptop  
CPU: 2.3Ghz Intel Core i5 processor  
RAM: NVIDIA GeForce 920M with 4 GB Dedicated VRAM

### **4.2.2 Software Requirements**

- Windows 10  
The operating system used for system development.
- Visual Studio 2015  
Visual Studio is used to code the framework which include website design, link database to website, and construct the functionality of the framework. It is easy to use and support the functions designed in this project.
- Microsoft SQL Server 2016  
Microsoft SQL Server 2016 is used to construct the database needed in this project. It provides all basic functions and features needed to construct a simple database.

### 4.2.3 Programming Language

- C#, Asp.Net

The ASP.Net framework is constructed by using C# programming language. C# programming language is easy to learn with C based programming language knowledge. It can direct the computer to perform necessary performance needed in the framework.

### 4.2.4 Markup Language

- HTML, CSS

HTML and CSS are used to design the user interface of the system. A clear, visible, and attractive layout can be form easily by using these markup languages.

### 4.2.5 Layout Template

- Bootstrap layout template

A simple Bootstrap layout template is used to form the default web form design to create an uniform web form layout throughout the system.

## 4.3 System Performance Definition

The proposed system, Scratch School is designed to meet the performance listed below:

- Accurate analysis result generated based on analysis plan
- Analysed projects results can be retrieved from the database by user and the projects' scoring is displayed clearly
- User able to subscribe to the framework successfully
- Recommendation is generated based on the project's problems or issues
- User can create Scratch project questions and evaluate project based on the question requirements successfully

### 4.4 Timeline

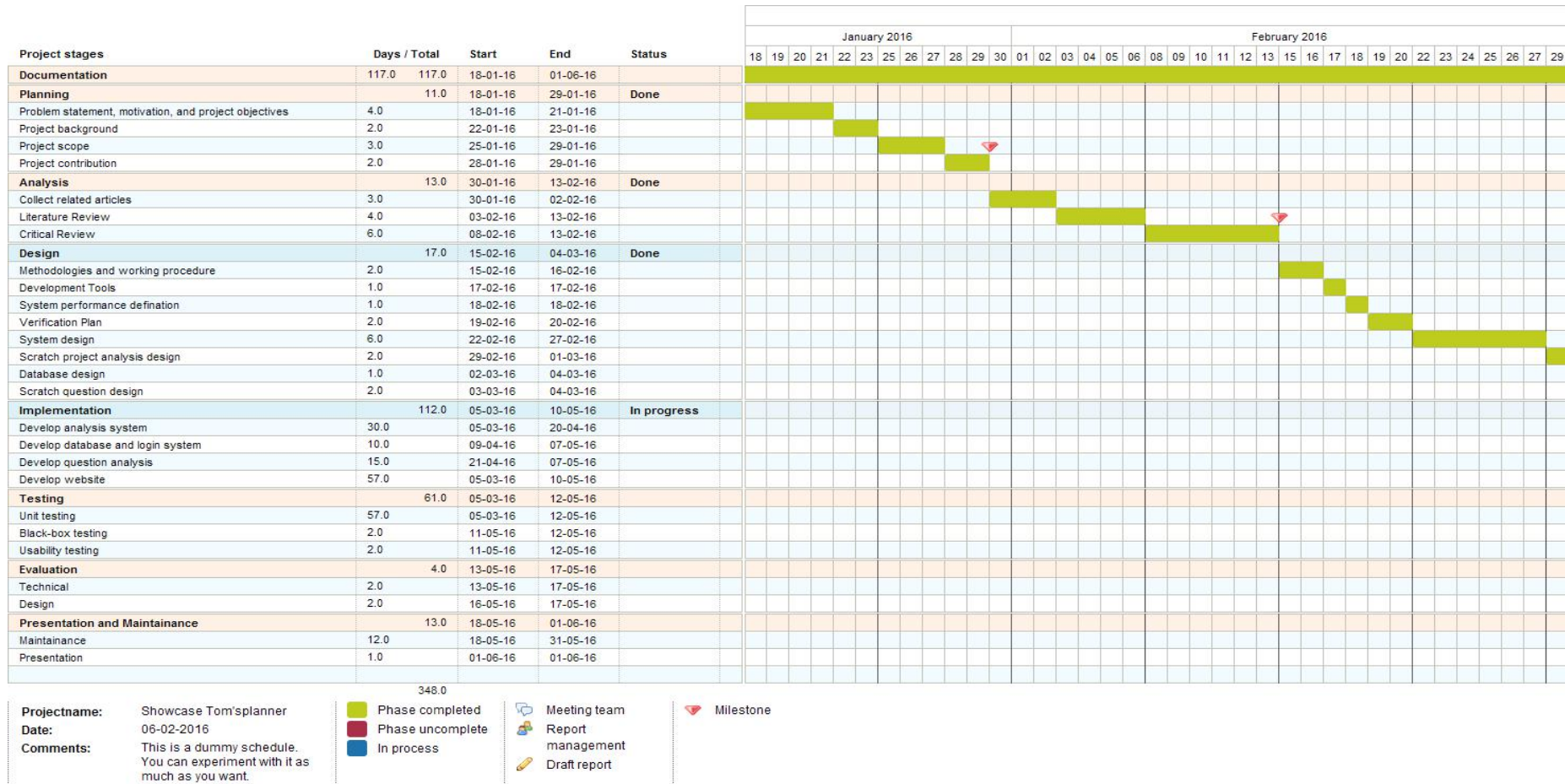
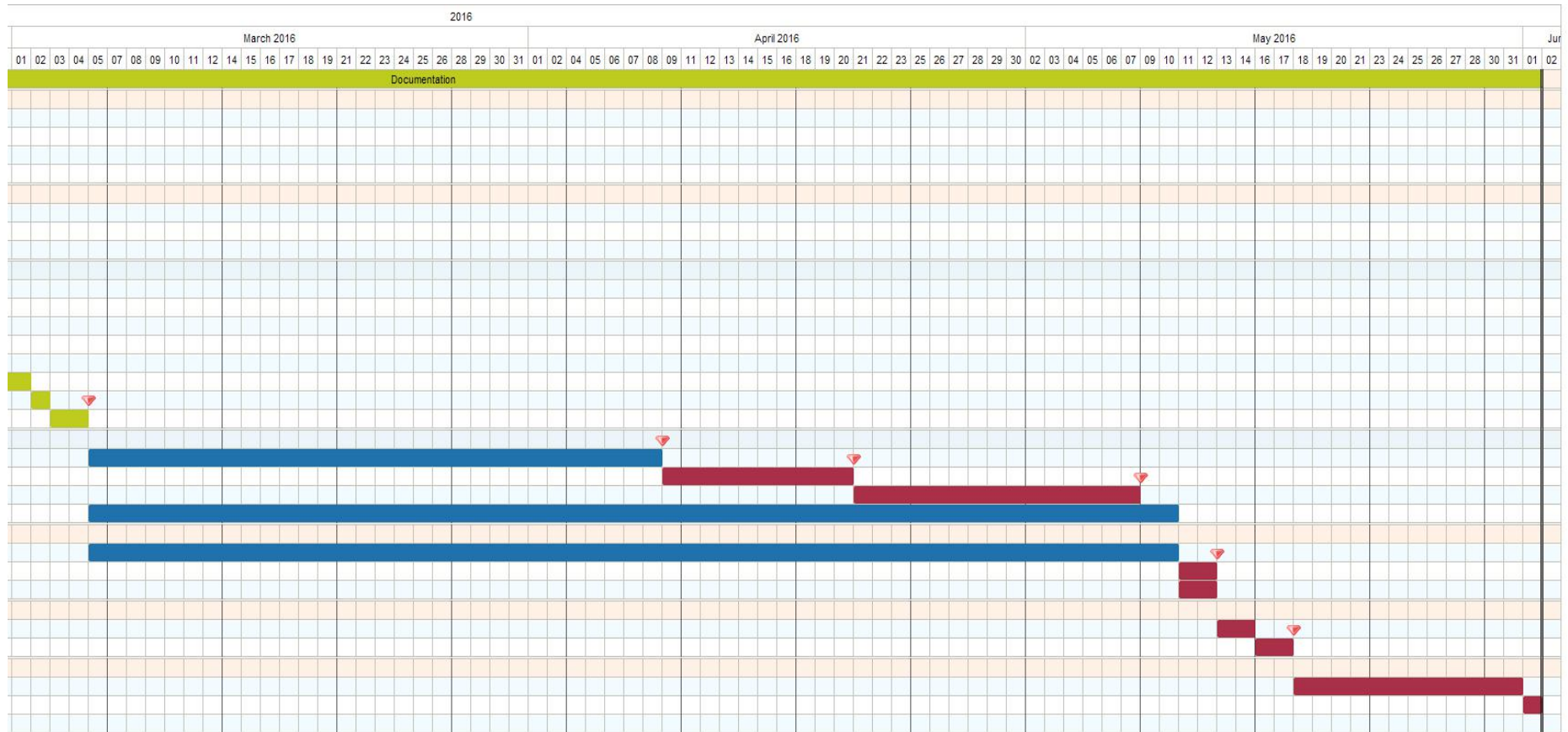


Figure 4-2: Part 1 of Gantt chart.



**Figure 4-3: Part 2 of Gantt chart.**

Project Schedule has planned as shown in Gantt chart in Figure 4-2 and Figure 4-3. The duration needed for the entire project to complete is around 117 days. The schedule is divided into seven groups which are planning, analysis, design, implementation, testing, evaluation, and maintenance and presentation.



#### 4.5 Implementation Issues and Challenges

The major challenge of the development of proposed system is the selection of CT concepts criteria. Since there are different explanation available to describe CT concepts, longer time is needed to understand the concepts and associate them in Scratch program. The analysis plan is then formed based on 6 CT concepts and the blocks which represent the CT concepts are identified. The identification of CT concept blocks are taken more time than expected.

On the other hand, project file need to be extracted after the file has been successfully uploaded through the web page. There are several file extraction methods but it can be hard to select a method which is suitable for this proposed project, Scratch School. After file extraction, searching for keywords in the JSON file in the project to start the analysis. The structure can be complex because there are different keywords for different elements and the sequence of the blocks may affect the final output.

Furthermore, the requirements of questions need to identify and specify in the analysis when performing analysis based on question. A well-planned construction structure is needed to overcome and reduce the possible error faced in the development phase.

Lastly, time management is important while develop this proposed system. The development processes need to be planned carefully to ensure that the system meet the system quality and project is on schedule. Documentation need to keep track regularly to avoid further errors and problems occur.

## CHAPTER 5 SYSTEM IMPLEMENTATION

## 5.1 System Publish

The proposed system, Scratch School is published using Microsoft Azure Cloud Computing Service to access by anyone through the internet. Web service and databases are created in Azure account before publish the system. Visual Studio 2015 need to connect to the Azure account before publish the system. By publish system through Microsoft Azure Cloud Computing Service, the web service is easier to maintain and network traffic can be track and control. Figure 5-1 shows the Scratch School web service after successful publish in the Microsoft Azure. Figure 5-2 shows the Visual Studio when publishing the proposed system, Scratch School.

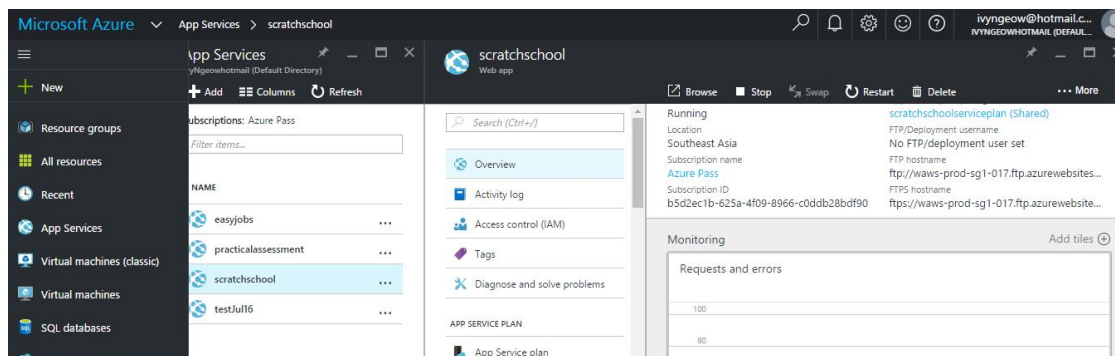


Figure 5-1: Scratch School publish in Microsoft Azure.

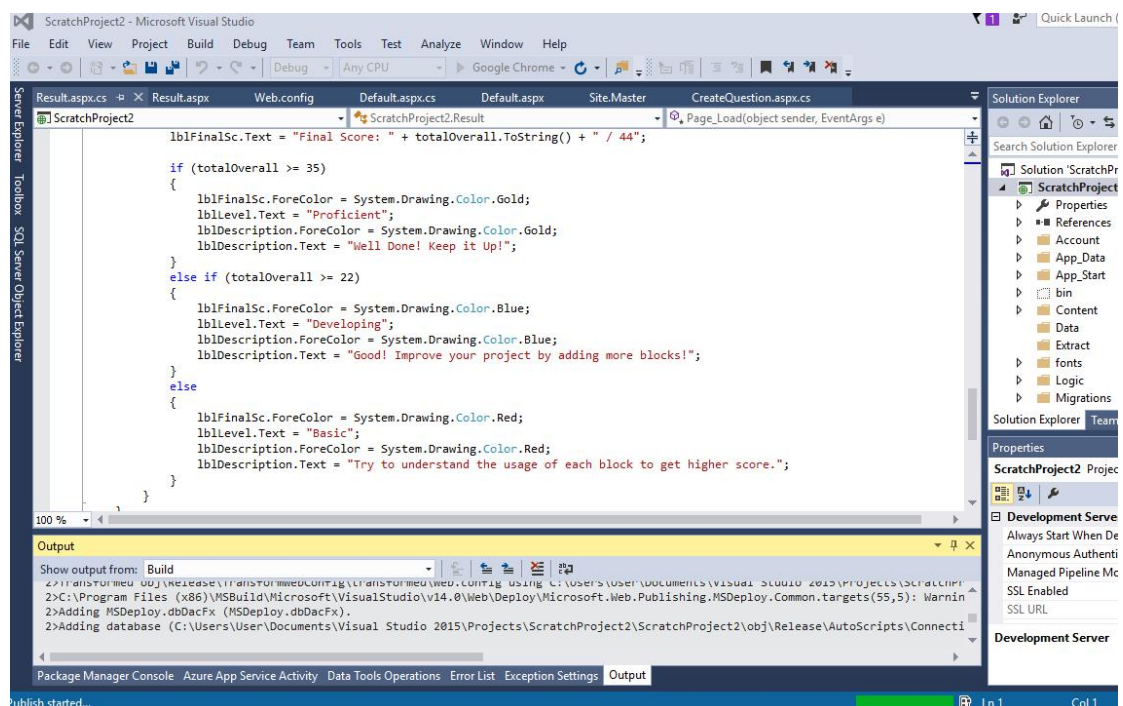
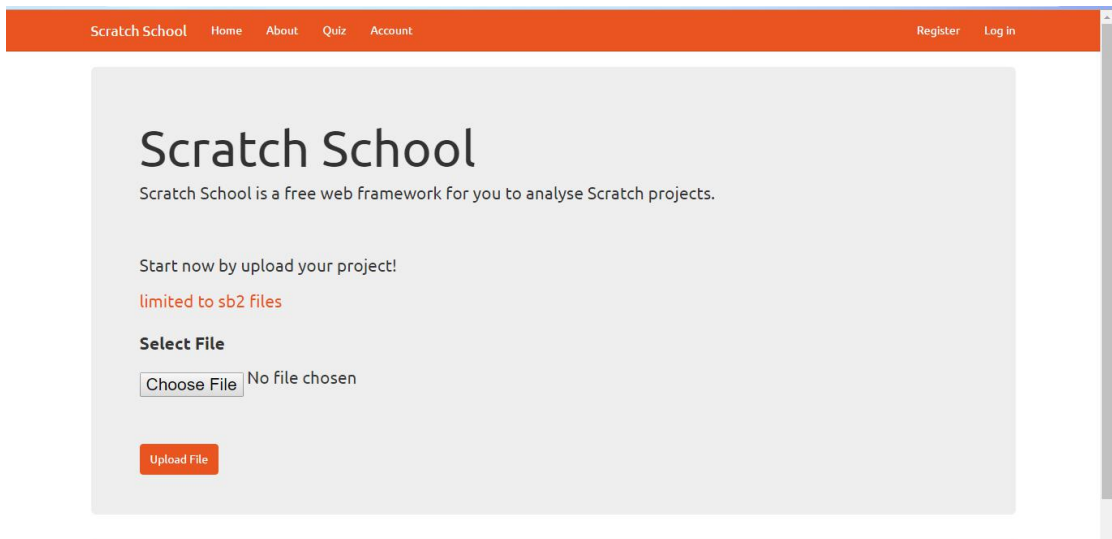


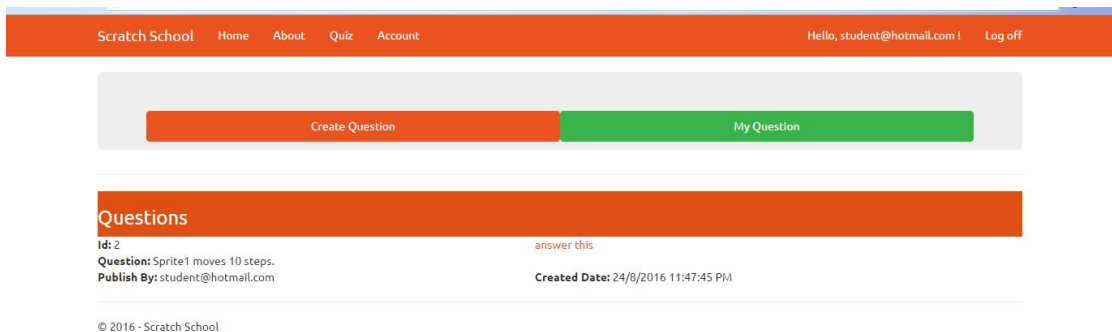
Figure 5-2: Publishing project in Visual Studio.

## 5.2 Screenshot



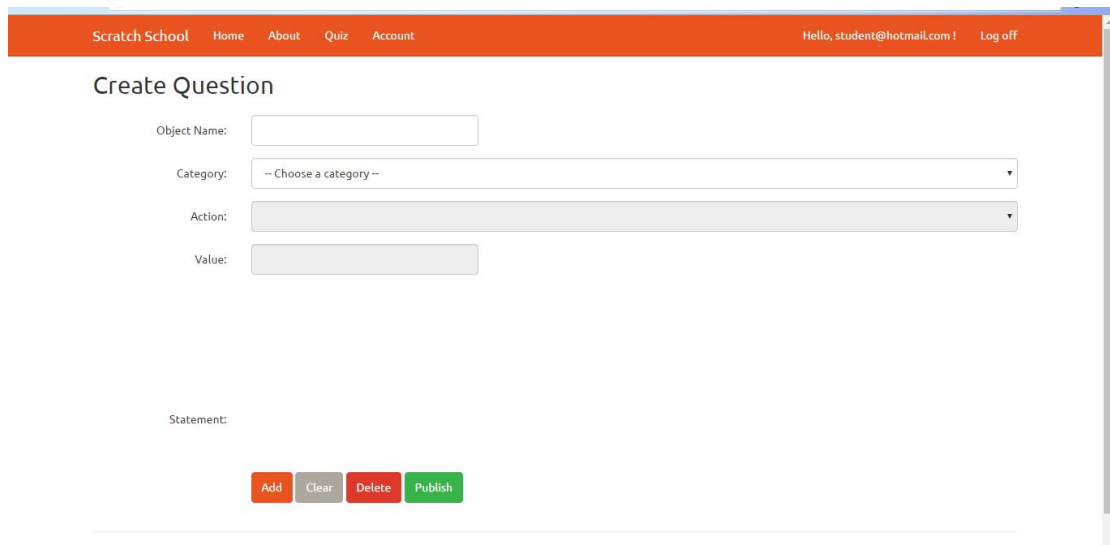
**Figure 5-3: Home page of Scratch School.**

Figure 5-3 shows the home page of the Scratch School which enable user to upload project for general analysis.



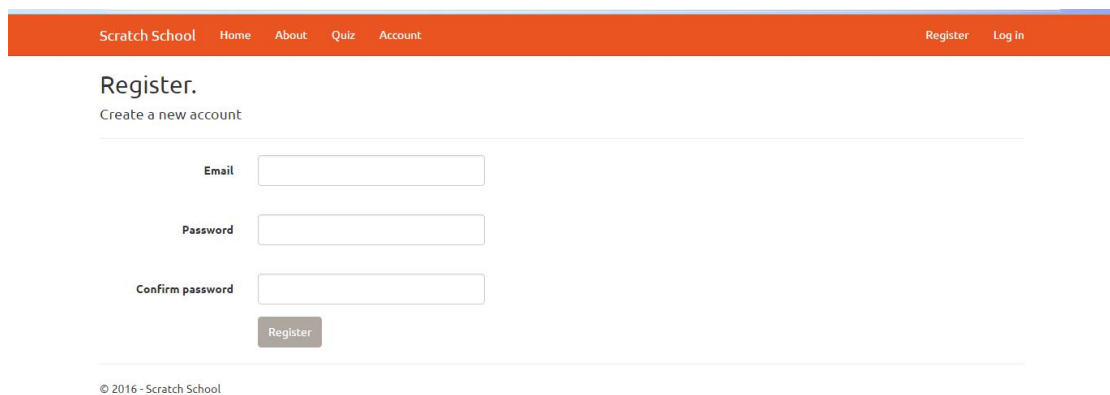
**Figure 5-4: Quiz page of Scratch School.**

Figure 5-4 shows the quiz page of Scratch School which enable user to view, select and create questions. My Question button enable users to view their created questions.



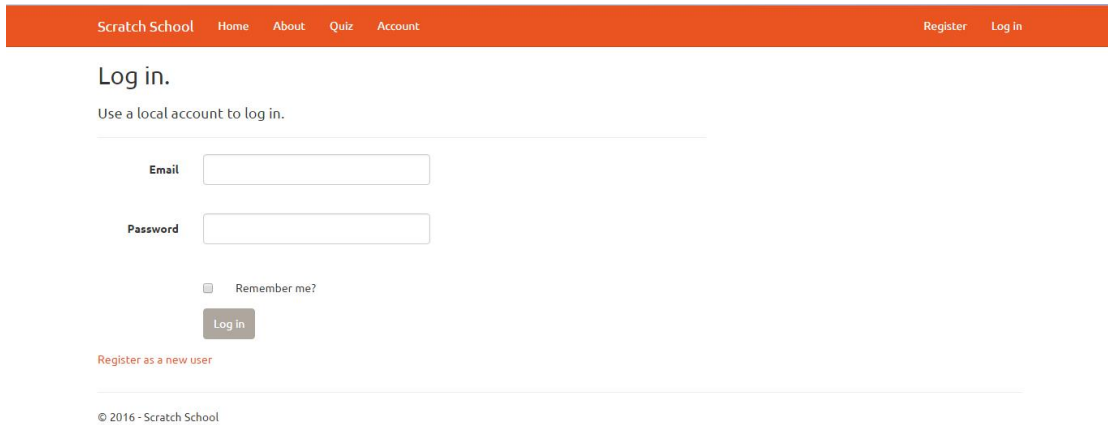
**Figure 5-5: Create Question page of Scratch School.**

Figure 5-5 shows the Create Question page of Scratch School which enable students to create Scratch question. Add button let user to add the created statement. Clear button clear out all the text in the text boxes and selection of the drop down list. Delete button let users to delete the created statement. Publish button let the users to save and publish the question in the Quiz page.



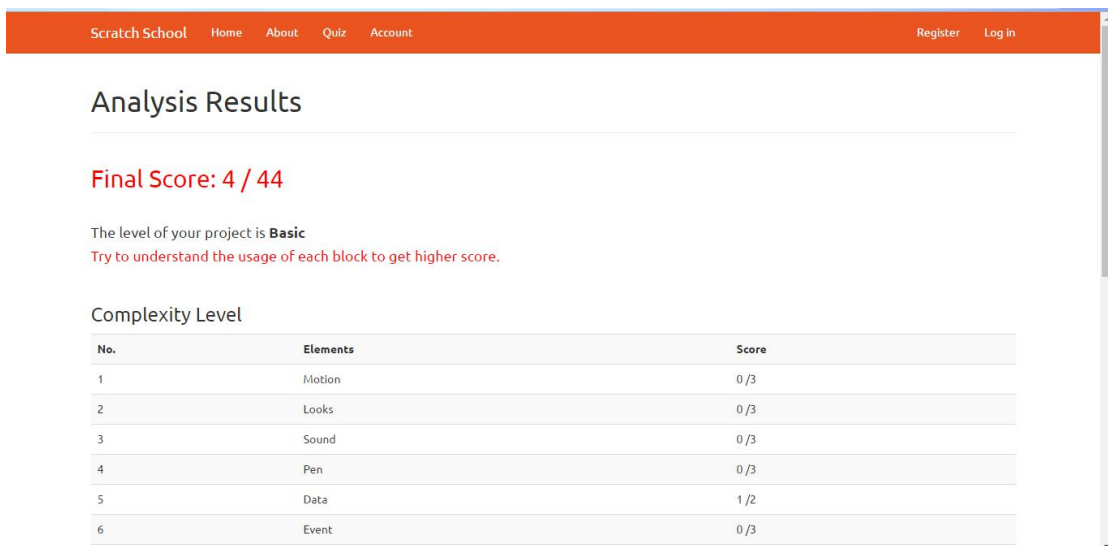
**Figure 5-6: Register page of Scratch School.**

Figure 5-6 shows the Register page of Scratch School which enable user to register a new user account.



**Figure 5-7: Login page of Scratch School.**

Figure 5-7 shows the Login page of Scratch School which enable users to log in to their user accounts.



**Figure 5-8: Part 1 of Analysis Result page 1 of Scratch School.**

Scratch School			Home	About	Quiz	Account	Register	Log in
7	Control	0/3						
8	Sensing	0/3						
9	Operators	0/3						
<b>Total</b>		<b>1/26</b>						

CT Concept Level

No.	CT Concepts	Score
1	Logic	0/3
2	Algorithms	0/3
3	Decomposition	3/3
4	Patterns	0/3
5	Abstraction	0/3
6	Synchronisation	0/3
<b>Total</b>		<b>3/18</b>

Warning: better use "When Green Flag Clicked" to start your project!

**Figure 5-9: Part 2 of Analysis Result page of Scratch School.**

Figure 5-8 and Figure 5-9 show the Analysis Result page of Scratch School which will display the project evaluation results.

How we evaluate your project?

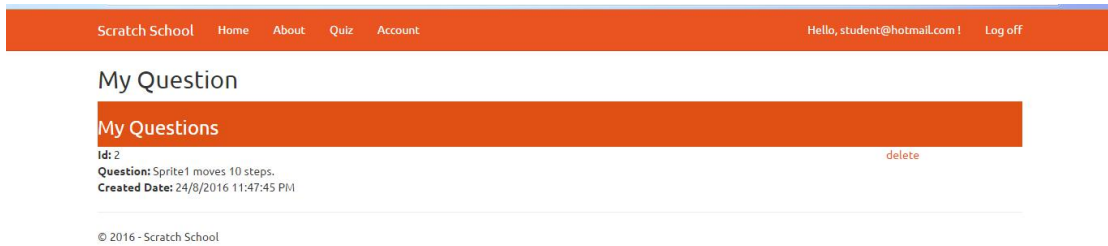
Complexity Level

The complexity level of blocks in the project is evaluated based on the availability of blocks. The marking scheme of complexity level is shown in the table below.

Elements	Marks allocated for blocks			Total
	1	1	1	
<b>Motion</b>	-move...steps -turn right...degree -turn left...degree -point in direction... -point towards...	-go to x... y... -go to... -glide...secs to x... y... -change x by... -set x to... -change y by... -set y to...	-if on edge, bounce -set rotation style...	3
<b>Looks</b>	-say...for...secs -say... -think...for...secs -think... -show -hide	-switch costume to... -next costume -switch backdrop to...	-change...effect by... -set...effect to... -clear graphic effects -change size by... -set size to...% -go to front -go back...layers	3

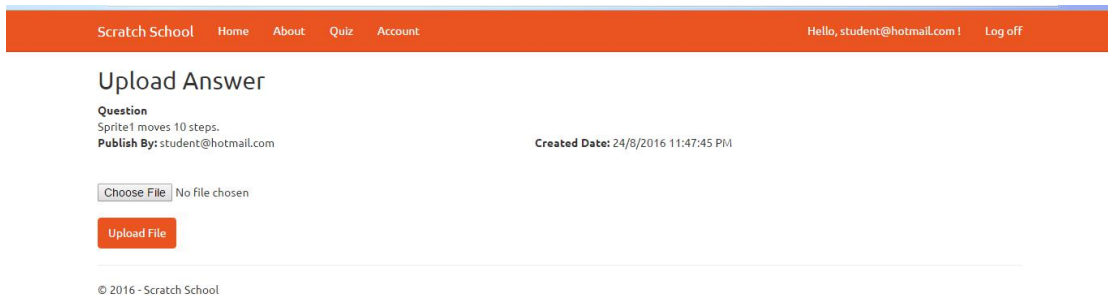
**Figure 5-10: About page of Scratch School.**

Figure 5-10 shows the About page of Scratch School which the marking criteria of Scratch project is displayed.



**Figure 5-11: My Question page of Scratch School.**

Figure 5-11 shows the My Question page of Scratch School to enable users to review and manage their created questions.



**Figure 5-12: Upload Answer page of Scratch School.**

Figure 5-12 shows the Upload Answer page of Scratch School which enable users to upload project for their selected question.

The screenshot shows the 'User Profile' page of Scratch School. At the top, there is a navigation bar with 'Scratch School', 'Home', 'About', 'Quiz', and 'Account'. On the right, it says 'Hello, student@hotmail.com!' and 'Log off'. Below the navigation bar, the title 'User Profile' is displayed. The main content is a table with three columns: 'Project Name', 'Upload Time', and 'Result'. The table lists various project names and their corresponding upload times and results.

Project Name	Upload Time	Result
askAndWait	24/8/2016 9:18:11 AM	Result
askAndWait	24/8/2016 9:18:29 AM	Result
testingmove	24/8/2016 9:28:37 AM	Result
2sprites	24/8/2016 9:31:25 AM	Result
addList	24/8/2016 9:49:54 AM	Result
and	24/8/2016 9:51:11 AM	Result
changeEffect	24/8/2016 9:55:19 AM	Result
broadcast	24/8/2016 9:57:13 AM	Result
changeSize	24/8/2016 10:00:08 AM	Result
changePenSize	24/8/2016 10:02:13 AM	Result
changeY	24/8/2016 10:08:51 AM	Result
changeVolume	24/8/2016 10:11:45 AM	Result
divide	24/8/2016 10:12:49 AM	Result
changeVariable	24/8/2016 10:20:04 AM	Result
testingmove	24/8/2016 10:20:53 AM	Result
testingmove	24/8/2016 10:22:02 AM	Result
testingmove	24/8/2016 10:23:15 AM	Result
testingmove	24/8/2016 10:31:41 AM	Result
testingmove	24/8/2016 10:34:14 AM	Result
testingmove	24/8/2016 10:37:25 AM	Result
testingmove	24/8/2016 10:41:29 AM	Result
testingmove	24/8/2016 10:43:36 AM	Result
testingmove	24/8/2016 10:45:42 AM	Result
testingmove	24/8/2016 10:49:16 AM	Result
testingmove	24/8/2016 10:52:15 AM	Result
testingmove	24/8/2016 10:53:19 AM	Result
testingmove	24/8/2016 10:56:27 AM	Result

**Figure 5-13: User Profile page of Scratch School.**

Figure 5-13 shows the User Profile page of Scratch School. User can review their previous analysis results in this page.



**CHAPTER 6 TESTING****6.1 Navigation Bar**

<b>No.</b>	<b>Event</b>	<b>Expected Result</b>	<b>Status</b>
1	User click on the Scratch School logo	Navigate to Home Page	<b>Pass</b>
2	User click on Home	Navigate to Home Page	<b>Pass</b>
3	User click on About	Navigate to About Page	<b>Pass</b>
4	User click on Quiz	Navigate to Quiz Page	<b>Pass</b>
5	User click on Register	Navigate to Register Page	<b>Pass</b>
6	User click on Login	Navigate to Login Page	<b>Pass</b>

**Table 6-1: Navigation bar testing results.****6.2 Home Page**

<b>No.</b>	<b>Event</b>	<b>Expected Result</b>	<b>Status</b>
1	User click on Home	Navigate to Home Page	<b>Pass</b>
2	User click on About	Navigate to About Page	<b>Pass</b>
3	User click on Quiz	Navigate to Quiz Page	<b>Pass</b>
4	User click on Register	Navigate to Register Page	<b>Pass</b>
5	User click on Login	Navigate to Login Page	<b>Pass</b>
6	User click Choose File button	Pop up window to enable user to select file to upload	<b>Pass</b>
7	User click Upload File button after selected a sb2 file	File uploaded and navigate to Analysis Result Page	<b>Pass</b>
8	User click Upload File without file selected	Display error message to prompt user to select file	<b>Pass</b>
9	User click Upload File button after selected a file which is not a sb2 file	Display error message to prompt user to select sb2 file	<b>Pass</b>

**Table 6-2: Home Page testing results.**

### 6.3 About Page

No.	Event	Expected Result	Status
1	User click on Home	Navigate to Home Page	Pass
2	User click on About	Navigate to About Page	Pass
3	User click on Quiz	Navigate to Quiz Page	Pass
4	User click on Register	Navigate to Register Page	Pass
5	User click on Login	Navigate to Login Page	Pass

**Table 6-3: About Page testing results.**

### 6.4 Quiz Page

No.	Event	Expected Result	Status
1	User click on Home	Navigate to Home Page	Pass
2	User click on About	Navigate to About Page	Pass
3	User click on Quiz	Navigate to Quiz Page	Pass
4	User click on Register	Navigate to Register Page	Pass
5	User click on Login	Navigate to Login Page	Pass
6	User click on Create Question button	Navigate to Create Question Page	Pass
7	User click on Answer Question link button of one question	Navigate to Upload Answer Page	Pass

**Table 6-4: Quiz Page testing results.**

### 6.5 Create Question Page

No.	Event	Expected Result	Status
1	User click on Home	Navigate to Home Page	Pass
2	User click on About	Navigate to About Page	Pass
3	User click on Quiz	Navigate to Quiz Page	Pass
4	User click on Register	Navigate to Register Page	Pass
5	User click on Login	Navigate to Login Page	Pass
6	User click on Add button (with necessary inputs )	Question created display in panel	Pass
7	User click on Clear button	All the text boxes and selections are empty	Pass
	User click on Delete button	The created question has been	Pass

		deleted	
	User click on Publish button	Question saved and published in the Quiz Page	<b>Pass</b>
	User click on Add button (without fulfil inputs)	Display error message to prompt user to fill in the necessary inputs	<b>Pass</b>
	User click on Publish button (without question formed)	Display error message to prompt user to form question.	<b>Pass</b>

**Table 6-5: Create Question Page testing results.**

### 6.6 Register Page

No.	Event	Expected Result	Status
1	User click on Home	Navigate to Home Page	<b>Pass</b>
2	User click on About	Navigate to About Page	<b>Pass</b>
3	User click on Quiz	Navigate to Quiz Page	<b>Pass</b>
4	User click on Register	Navigate to Register Page	<b>Pass</b>
5	User click on Login	Navigate to Login Page	<b>Pass</b>
6	User click on Register button (with valid inputs)	User account created and navigate to User Profile Page	<b>Pass</b>
7	User click on Register button (with invalid inputs)	Display error message to prompt user to provide valid inputs	<b>Pass</b>

**Table 6-6: Register Page testing results.**

### 6.7 Login Page

No.	Event	Expected Result	Status
1	User click on Home	Navigate to Home Page	<b>Pass</b>
2	User click on About	Navigate to About Page	<b>Pass</b>
3	User click on Quiz	Navigate to Quiz Page	<b>Pass</b>
4	User click on Register	Navigate to Register Page	<b>Pass</b>
5	User click on Login	Navigate to Login Page	<b>Pass</b>
6	User click on Log in button (with valid inputs)	Navigate to User Profile Page	<b>Pass</b>
7	User click on Log in button (with invalid inputs)	Display error message to prompt user to provide valid	<b>Pass</b>

		inputs	
--	--	--------	--

**Table 6-7: Login Page testing results.****6.8 Analysis Result Page**

No.	Event	Expected Result	Status
1	User click on Home	Navigate to Home Page	<b>Pass</b>
2	User click on About	Navigate to About Page	<b>Pass</b>
3	User click on Quiz	Navigate to Quiz Page	<b>Pass</b>
4	User click on Register	Navigate to Register Page	<b>Pass</b>
5	User click on Login	Navigate to Login Page	<b>Pass</b>

**Table 6-8: Analysis Result Page testing results.****6.9 Complexity Level Scoring****6.9.1 Motion Score**

No.	Event	Expected Result	Status
1	Project with no Motion category	Motion Score = 0	<b>Pass</b>
2	Project with one Motion category	Motion Score = 1	<b>Pass</b>
3	Project with two Motion category	Motion Score = 2	<b>Pass</b>
4	Project with three Motion category	Motion Score = 3	<b>Pass</b>

**Table 6-9: Motion score testing results.**

**6.9.2 Looks Score**

No.	Event	Expected Result	Status
1	Project with no Looks category	Looks Score = 0	Pass
2	Project with one Looks category	Looks Score = 1	Pass
3	Project with two Looks category	Looks Score = 2	Pass
4	Project with three Looks category	Looks Score = 3	Pass

**Table 6-10: Looks score testing results.****6.9.3 Sound Score**

No.	Event	Expected Result	Status
1	Project with no Sound category	Sound Score = 0	Pass
2	Project with one Sound category	Sound Score = 1	Pass
3	Project with two Sound category	Sound Score = 2	Pass
4	Project with three Sound category	Sound Score = 3	Pass

**Table 6-11: Sound score testing results.****6.9.4 Pen Score**

No.	Event	Expected Result	Status
1	Project with no Pen category	Pen Score = 0	Pass
2	Project with one Pen category	Pen Score = 1	Pass
3	Project with two Pen category	Pen Score = 2	Pass
4	Project with three Pen category	Pen Score = 3	Pass

**Table 6-12: Pen score testing results.**

**6.9.5 Data Score**

No.	Event	Expected Result	Status
1	Project with no Data category	Data Score = 0	<b>Pass</b>
2	Project with one Data category	Data Score = 1	<b>Pass</b>
3	Project with two Data category	Data Score = 2	<b>Pass</b>

**Table 6-13: Data score testing results.****6.9.6 Event Score**

No.	Event	Expected Result	Status
1	Project with no Event category	Event Score = 0	<b>Pass</b>
2	Project with one Event category	Event Score = 1	<b>Pass</b>
3	Project with two Event category	Event Score = 2	<b>Pass</b>
4	Project with three Event category	Event Score = 3	<b>Pass</b>

**Table 6-14: Event score testing results.****6.9.7 Control Score**

No.	Event	Expected Result	Status
1	Project with no Control category	Control Score = 0	<b>Pass</b>
2	Project with one Control category	Control Score = 1	<b>Pass</b>
3	Project with two Control category	Control Score = 2	<b>Pass</b>
4	Project with three Control category	Control Score = 3	<b>Pass</b>

**Table 6-15: Control score testing results.**

**6.9.8 Sensing Score**

<b>No.</b>	<b>Event</b>	<b>Expected Result</b>	<b>Status</b>
1	Project with no Sensing category	Sensing Score = 0	<b>Pass</b>
2	Project with one Sensing category	Sensing Score = 1	<b>Pass</b>
3	Project with two Sensing category	Sensing Score = 2	<b>Pass</b>
4	Project with three Sensing category	Sensing Score = 3	<b>Pass</b>

**Table 6-16: Sensing score testing results.****6.9.9 Operators Score**

<b>No.</b>	<b>Event</b>	<b>Expected Result</b>	<b>Status</b>
1	Project with no Operators category	Operators Score = 0	<b>Pass</b>
2	Project with one Operators category	Operators Score = 1	<b>Pass</b>
3	Project with two Operators category	Operators Score = 2	<b>Pass</b>
4	Project with three Operators category	Operators Score = 3	<b>Pass</b>

**Table 6-17: Operators score testing results.**

## 6.10 CT Concepts Level Scoring

### 6.10.1 Logic Score

No.	Event	Expected Result	Status
1	Project with no Logic category	Logic Score = 0	Pass
2	Project with one-mark Logic category	Logic Score = 1	Pass
3	Project with two-mark Logic category	Logic Score = 2	Pass
4	Project with three-mark Logic category	Logic Score = 3	Pass

**Table 6-18: Logic score testing results.**

### 6.10.2 Algorithms Score

No.	Event	Expected Result	Status
1	Project with no Algorithms category	Algorithms Score = 0	Pass
2	Project with one-mark Algorithms category	Algorithms Score = 1	Pass
3	Project with two-mark Algorithms category	Algorithms Score = 2	Pass
4	Project with three-mark Algorithms category	Algorithms Score = 3	Pass

**Table 6-19: Algorithms score testing results.**



### 6.10.3 Decomposition Score

No.	Event	Expected Result	Status
1	Project with no Decomposition category	Decomposition Score = 0	Pass
2	Project with one-mark Decomposition category	Decomposition Score = 1	Pass
3	Project with two-mark Decomposition category	Decomposition Score = 2	Pass
4	Project with three-mark Decomposition category	Decomposition Score = 3	Pass

**Table 6-20: Decomposition score testing results.**

### 6.10.4 Patterns Score

No.	Event	Expected Result	Status
1	Project with no Patterns category	Patterns Score = 0	Pass
2	Project with one-mark Patterns category	Patterns Score = 1	Pass
3	Project with two-mark Patterns category	Patterns Score = 2	Pass
4	Project with three-mark Patterns category	Patterns Score = 3	Pass

**Table 6-21: Patterns score testing results.**

### 6.10.5 Abstraction Score

No.	Event	Expected Result	Status
1	Project with no Abstraction category	Abstraction Score = 0	Pass
2	Project with one-mark Abstraction category	Abstraction Score = 1	Pass
3	Project with two-mark Abstraction category	Abstraction Score = 2	Pass
4	Project with three-mark Abstraction category	Abstraction Score = 3	Pass

**Table 6-22: Abstraction score testing results.**

### 6.10.6 Synchronisation Score

No.	Event	Expected Result	Status
1	Project with no Synchronisation category	Synchronisation Score = 0	<b>Pass</b>
2	Project with one-mark Synchronisation category	Synchronisation Score = 1	<b>Pass</b>
3	Project with two-mark Synchronisation category	Synchronisation Score = 2	<b>Pass</b>
4	Project with three-mark Synchronisation category	Synchronisation Score = 3	<b>Pass</b>

**Table 6-23: Synchronisation score testing results.**

### 6.11 System Testing

The complete system will be tested by Bukit Timah Primary School's students. A pre-test is performed to identify the students' Scratch programming standards before using this proposed system. The students' projects are picked randomly. The projects are collected and evaluated by using Scratch School. The analysis results of these students are shown in Table 6-24.

Student	Complexity Level	CT Concept Level	Overall Score
A	10	8	18
B	3	6	9
C	1	2	3
D	7	7	14
E	0	1	1
F	9	9	18
G	0	1	1
H	3	5	8
I	8	8	16
J	9	8	17
K	9	8	17
L	2	2	4
M	1	0	1

**Table 6-24: Analysis results of random picked students.**

The analysis results show that the students are still in the basic level of Scratch project because the overall score does not exceed 22 marks. There is only a pre-test is conducted due to the project submission deadline. Another testing will be conducted by same group of students after learned Scratch programming for one month. The analysis results of these students will be collected and compared with the results in pre-test.

## **CHAPTER 7 CONCLUSION**

In conclusion, the project deliverable is a web service analysis framework called Scratch School which is implemented to analyse Scratch projects. Scratch School can generate accurate and standardised analysis results. Students are able to access a reliable Scratch project analysis system any time and any where which enhance their learning on Scratch programming. The quality of the Scratch project is deepen by not rewarded fully empty command blocks any mark in Scratch School. Students can subscribe to the web service to store and review their analysis results. This help students in performing self-learning and performance comparison on Scratch projects. Students can improve their projects based on the recommendations in the analysis results and eliminate possible detected errors. Educators can use Scratch School to create Scratch questions and examine students based on their created questions. The specific requirements in the question will be checked so students need to have problem solving skill to pass the question. The analysis result is made up of complexity level and CT concept level of Scratch projects. The development of CT concepts among students affect students' decisions and methods used to solve a problem.

The scope of this framework including website development, database development, question creation, analysis system development, and generate recommendation while result display. The methodology uses in the development of the system is Agile methodology. The project will break to several parts such as web page layout design, file storing and extracting, and database design. The testing and development of the system will be carried out concurrently to ease the identification of errors and problems of the system. Test cases are developed to test the system functionality. This project framework is believed to provide a better Scratch analysis environment and contribute to the area of study.

### **7.1 Future Development**

This web service is design to cater with future development and improvement. One of the possible improvement is question suggestion. The system can suggest question to the users based on previous analysis results to help them overcome their weaknesses in Scratch programming. Moreover, question creation can be enhanced by allow users to create long or large question. The existing module

## CHAPTER 7 CONCLUSION

only enable users to create short questions and the inputs' data type is restricted. Therefore, long question can be implemented to evaluate more criteria needed by users.

## BIBLIOGRAPHY

- Barefoot Computing 2014, *Computational Thinking - Barefoot Computing Barefoot Computing*. Available from: <<http://barefootcas.org.uk/>>. [1 August 2016].
- Boe, B, Conrad, P, Dreschler, G, Franklin, D, Hill, C & Len, M 2013, *Hairball: Lint-inspired Static Analysis of Scratch Projects*. Available from: <<http://cs.ucsb.edu/~bboe/public/pubs/hairball-sigcse13.pdf>>. [26 November 2015].
- Colleen, L n.d., *Programming in Scratch*. Available from: <<https://www.edx.org/course/programming-scratch-harveymuddx-cs002x-0#!>>. [26 November 2015].
- Dr. Scratch 2014, *Dr. Scratch*. Available from: <<http://drscratch.programamos.es/>>. [26 November 2015].
- GitHub 2016, *GitHub - ucsb-cs-education/hairball: Hairball is a plugin-able framework useful for static analysis of Scratch projects*. Available from: <<https://github.com/ucsb-cs-education/hairball>>. [20 February 2016].
- Isong, J 2001, 'Developing an Automated Program Checker', *Proceedings of the twelfth annual CCSC south central conference*. Available from: <<http://www.cis.famu.edu/~ejones/papers/jisong-grading-ccscsc.pdf>>. [28 November 2015].
- Moffat, D & Wilson, A 2010, *Evaluating Scratch to introduce younger schoolchildren to programming*. Available from: <<http://scratched.gse.harvard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf>>. [30 January 2016].
- Moreno-León, J & Robles, G 2015, *Analyze your Scratch projects with Dr. Scratch and assess your Computational Thinking skills*. Available from: <<http://jemole.me/replication/2015scratch/InferCT.pdf>>. [28 November 2015].
- Moreno-León, J, Robles, G & Román-González, M 2015, *Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking*. Available from: <[http://www.um.es/ead/red/46/moreno\\_robles.pdf](http://www.um.es/ead/red/46/moreno_robles.pdf)>. [28 November 2015].
- Tutorialspoint 2016, *SDLC – Agile Model*. Available from: <[http://www.tutorialspoint.com/sdlc/sdlc\\_agile\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm)>. [4 March 2016].