EVALUATING STUDENTS' LEARNING OUTCOME IN PROGRAMMING CONCEPTS USING SCRATCH 3.0

BY

ANG JIE QIAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: EVALUATING STUDENTS' LEARNING OUTCOMES IN PROGRAMMING CONCEPTS USING SCRATCH 3.0

Academic Session: _JUNE 2023_

I ANG JIE QIAN
(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,

SunTeikHeng

(Author's signature)

(Supervisor's signature)

Address:
2, Lintang Bukit Jambul 2,
Bayan Lepas, 11900,
Pulau Pinang.

Sun Teik Heng @ San Teik Heng
Supervisor's name

Date: 14-SEPT-2023

Date: 15-SEPT-2023

Universiti Tunku Abdul Rahman			
Form Title : **Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF _INFORMATION AND COMMUNICATION TECHNOLOGY_

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___14-SEPT-2023____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ___***Ang Jie Qian***__ (ID No: __***19ACB01812***__) has completed this final year project entitled "__Evaluating Students' Learning Outcome in Programming Concepts Using Scratch 3.0__" under the supervision of _Ts. Sun Teik Heng_ (Supervisor) from the Department of ___Information System___, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____
(*Ang Jie Qian*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**EVALUATING STUDENTS' LEARNING OUTCOMES IN PROGRAMMING CONCEPTS USING SCRATCH 3.0**" 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 : ___Ang Jie Qian___

Date : ___14-SEPT-2023______

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Mr Sun Teik Heng for granting me an opportunity to study more about web development. Besides, he also guided me throughout the project by resolving the problems I encountered during the project development phase and recommending any related information which assisted me in developing the web application.

Apart from that, I am also grateful to my family members and friends, who supported me and gave me the inspiration to brainstorm ideas for developing the web application.

ABSTRACT

Scratch is an educational website widely adopted in education sectors to introduce the programming world to children aged between 8 to 16 years old. However, unlike existing IDE that comes with error detections, Scratch only provides a platform where Scratchers, the registered users on the Scratch website, creatively design their scripts and share their works with others. It does not come with an error detection function, contributing to a higher time consumption for educators to mark the scripts of the projects individually without the help of any system or machine. Fortunately, some existing tools such as Hairball and Dr. Scratch were developed to assist them in evaluating Scratch projects, but these tools still hold some limitations which might affect the accuracy of the evaluation results and negatively impact users' experience and confidence level. Thus, this project focuses on developing a web application with ASP.NET core Razor Pages, which uses a novel evaluation method to evaluate Scratch projects by uploading the project onto the web application. This evaluation method considers the type of project to ensure fairness in assessing students' performance. By logging into the web application, users will be given different privileges while accessing the web application, and a database will be utilised to store and retrieve users' evaluation results based on their user privileges.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DOM	Document Object Model
JSON	JavaScript Object Notation
CT	Computational Thinking
VD	Variable Declaration (Code Smell)
UnC	Unused Code (Code Smell)
Red	Redundancy (Code Smell)
LT	Logical Thinking (CT Concept)
DR	Data Representation (CT Concept)
UI	User Interactivity (CT Concept)
P	Parallelism (CT Concept)
DA	Decomposition and Abstraction (CT Concept)
SYN	Synchronisation (CT Concept)
FC	Flow Control (CT Concept)

CHAPTER 1

Introduction

Scratch is an educational website aimed at nurturing mathematical and computational concepts among children between the ages of 8 and 16. [1] Through their previous research, it was identified that children's lack of interest in learning programming was mainly caused by the difficulty in learning the programming languages, little or no relation to their interests and inadequate expertise to guide them when things went wrong. Later, the rise of new programming languages such as Alice and Squeak Etoys that were developed for younger programmers inspired them to create Scratch. Since its first official release in May 2007, the team has been trying their best to enhance the functionalities in Scratch to improve user experience and create a comfortable environment for users to express their ideas through programming, leading to the introduction of Scratch 3.0 released on 2nd of January in the year 2019. The six common types of Scratch projects are Animation, Music, Games, Tutorials, Stories and Art.

1.1 Problem Statement and Motivation

Scratch is commonly used by educators to teach younger children about the world of programming. The drag-and-drop feature of command blocks in Scratch increases the interactivity between the children and programming, attracting their interest in producing creative projects. However, unlike existing IDE such as Jupyter Notebook, Android Studio and Microsoft Visual Studio which can detect errors and notify the programmers about them, Scratch purely provides an interface for the children to play around with. In other words, educators have to manually go through their projects individually to evaluate students' performance. It is a tedious task for an educator to evaluate all projects created by the students on their own. Dr. Scratch is developed to cope with this problem, but this web application does not evaluate the projects based on the type of projects. This potentially introduces unfairness in the evaluation system and eventually affects students' level of confidence as some projects do not necessarily utilise all blocks provided in Scratch. Moreover, Dr. Scratch only separates the users into two categories (registered users and unregistered users) and shows only the computed statistics based on the projects uploaded in the past. The users will not be able to retrieve the evaluation result of a specific project after leaving the evaluation dashboard. This is a critical issue as students are unable to check their weaknesses and improvements based on the evaluation results. Besides, the teachers are unable to access the students' evaluation results as Dr. Scratch does not implement role assignments to users that is capable of linking students' accounts to teachers' accounts.

Thus, this project aims to propose and develop a web application that uses a new method to evaluate students' learning outcomes in programming concepts using Scratch 3.0. This project will also include the usage of a database and a function enabling users to store, retrieve and compare their evaluation results even after leaving the dashboard. At the same time, the registered users will be further categorised into teacher, student and administrator roles.

1.2 Research Objectives

The project aims to enhance the accuracy and fairness in evaluating students' performance in using Scratch 3.0 by using a new algorithm. This new algorithm considers the type of project while evaluating students' projects. This is done by applying the weighting factor to each evaluation criterion based on the type of project chosen.

Besides, this project introduces different user privileges which are teachers, students and administrators. This increases the efficiency of the workflow as different user privileges play different roles. For instance, administrators manage the users of the web application by assigning roles to users, while teachers keep track of their student's performance by assessing the overall report generated by the web application.

Furthermore, users will be able to store, retrieve and compare the analysis results based on the users' privilege even after leaving the result page. Teachers would be able to grade their students based on the retrieved results, whereas the students can compare their latest performance with the past evaluation results to improve their skills in programming.

1.3 Project Scope and Direction

The final delivery of this project is a web application that can evaluate Scratch projects with improved accuracy by using a new algorithm and other functionalities that assist the teachers in conducting a programming class. The scopes of the project include:

- Web Application Development

  A web application will be developed for users to upload the Scratch projects for evaluation, store, display and compare the evaluation results. Teachers can create a new activity for project submission whereas the administrators can assign roles and classes to new users.

- Implementation of Database

  A database that can store and retrieve students' evaluation results, users' information and Scratch activities created by teachers. The Identity API will create a default database that stores the login information of the users including their usernames, passwords and email addresses.

- Implementation of User Privilege

  Three types of user privilege will be introduced which are teacher, student and administrator.

User Privilege	Actions
Teacher	• Upload students' project for evaluation. • View students' evaluation result. • View class performance. • Download the class performance report. • Download students' evaluation result. • Create new activity for project submission.
Student	• Upload project for evaluation. • View their evaluation results. • Download their evaluation results. • Compare their evaluation results.
Administrator	• Manage the roles of users. • Manage user details. • Assign students and teachers to different classes.

Table 1.1: User Privileges and Actions Summary

- Evaluation System

  A new evaluation criteria and calculation will be proposed to evaluate the Scratch projects in a more accurate and precise method. The implemented method and calculation will be further elaborated in Chapter 3.

1.4 Contributions

Firstly, the users are given roles to assess the different functionalities provided by the web application. The user privileges and actions are mentioned in Table 1.1.

Next, the web application benefits teachers and students by evaluating the Scratch projects based on a newly proposed evaluation method with improved accuracy and precision. The proposed method will evaluate the Scratch projects based on the type of projects and convert the points obtained into percentages after multiplying the points with weighting factors, which guarantees the fairness of the marks obtained by students. Besides, users can access different information based on their privileges. For example, students can view and download their past evaluation results while teachers can create a new activity and check the details of the activities created.

Most importantly, there are only a few existing systems that can evaluate Scratch projects created using Scratch 3.0, which means that this project also explores a new way to evaluate the projects created using Scratch 3.0. This is because Scratch 3.0 has introduced new blocks that were not in Scratch 1.4 and Scratch 2.0. Also, the way the exported project file of Scratch 3.0 stores the information of a project is different compared to the project created by using Scratch 2.0 due to the difference between the blocks provided and the naming of the key-value pairs in the JSON file.

1.5 Report Organization

There are seven chapters included in this report. Chapter 1 introduces this project which includes the problem statement and motivation, project scope and direction and contributions. In Chapter 2, the information relevant to evaluating Scratch projects and several existing systems to evaluate Scratch projects are studied. Then, Chapter 3 discusses the overall system design of the project and the proposed method to overcome the problem stated and achieve the project objectives. The fourth chapter discusses the overall system design of the project and the fifth chapter discusses the system implementation of the project, including the hardware and software setup to develop the web application and problems encountered during the implementation. In Chapter 6, a system test is done to detect the presence of defects and ensure that the web application is functioning properly as stated in the requirements. Lastly, Chapter 7 concludes the project and suggests potential improvements for the web application.

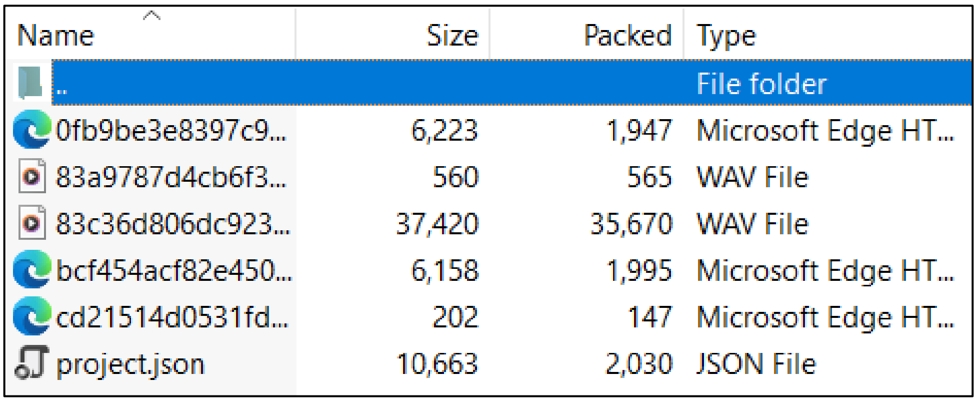
CHAPTER 2

Literature Review

2.1 Review of the Technologies

2.1.1 Scratch File Format

When users export their projects to their computers, it is saved as a ZIP archive with ".sb3" as the file extension. As shown in Figure 2.1, the exported project file consists of encoded information about the project, including the backdrops, costumes and sounds used. Most importantly, the file also contains a JSON file named "project.json" in general, which records all information about the blocks used by the users in the project.



Name	Size	Packed	Type
..			File folder
0fb9be3e8397c9...	6,223	1,947	Microsoft Edge HT...
83a9787d4cb6f3...	560	565	WAV File
83c36d806dc923...	37,420	35,670	WAV File
bcf454acf82e450...	6,158	1,995	Microsoft Edge HT...
cd21514d0531fd...	202	147	Microsoft Edge HT...
project.json	10,663	2,030	JSON File

Figure 2.1: Content of .sb3 File

The content of "project.json" is stored in the form of hierarchical tree structure. Figure 2.2 shows a partial of the tree structure of the JSON file. The "Unique_ID" is a randomly generated string of ID given to each block used in the project.

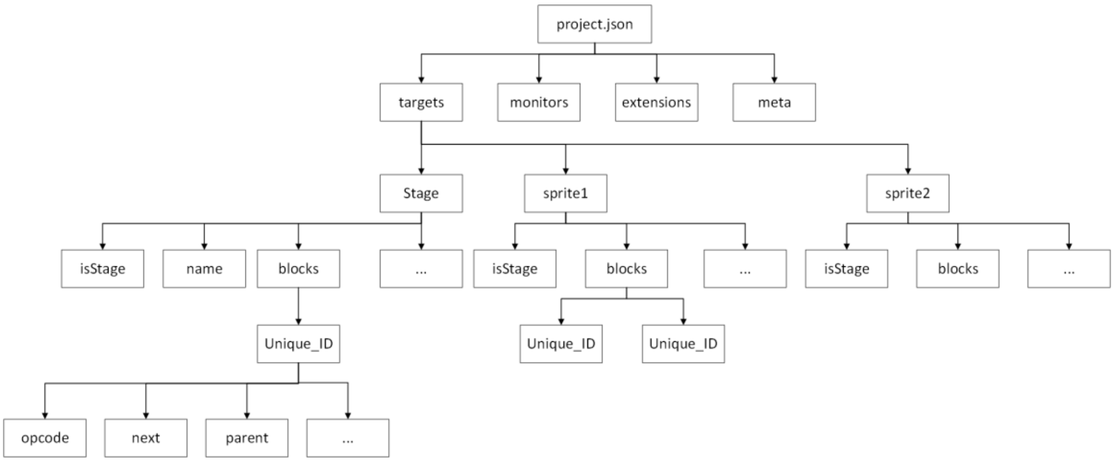


Figure 2.2: Partial Tree Structure of "project.json"

Table 2.1 shows the description of the properties in “project.json”. Most blocks are stored in “blocks” property in each target, represented by objects with the properties mentioned in Table 2.2. However, some of the blocks are represented as an array instead of an object, in which the first element is a number that represents the block applied (Table 2.3).

Properties	Description
targets	An array of targets that represents the stage or a sprite.
monitors	An array of monitors.
extensions	An array of extensions used. The extension will be removed from the project if none of its blocks are used in the project after adding an extension.
meta	Metadata about the project’s author and the Scratch version used.

Table 2.1: Partial Scratch File Format [2]

Properties	Description
isStage	True if the specific target is a Stage and false otherwise.
name	Name of the sprite. Always “Stage” for the stage.
blocks	An object that stores the information of the blocks used in the project. Below are a few properties that need to be put into consideration while evaluating the project: - 1. opcode: string name of the block 2. next: ID of the following block, null if the block is the last block. 3. parent: ID of the preceding block, null if the block is the first block. 4. topLevel: false if the block has a parent, and true otherwise.

Table 2.2: Partial Properties of Blocks as Object [2]

Block	1st element	2nd element	3rd element	4th element	5th element
Number	4	Value	-	-	-
Positive number	5	Value	-	-	-
Positive integer	6	Value	-	-	-
Integer	7	Value	-	-	-
Angle	8	Value	-	-	-
Colour	9	# followed by a hexadecimal numeral representing the colour	ID	-	-
String	10	Value	ID	x-coordinate, if it is top-level	y-coordinate, if it is top-level
Broadcast	11	Name	ID	x-coordinate, if it is top-level	y-coordinate, if it is top-level

Table 2.3: Information of Blocks Represented by Arrays [2]

2.1.2 Scratch Blocks

Below shows a brief overview of the categories of the blocks and types of blocks that will be used to evaluate a project.

2.1.2.1 Event Blocks

To trigger scripts to run without user click, it is necessary to put a Hat block as the first block of the script. Excluding the hat blocks included in extensions, most of the hat blocks are categorised under the Event category such as the “When green flag clicked” block and the “When () key pressed” block (Figure 2.3). These blocks will be used to evaluate several evaluation criteria such as Unused Code and Parallelism.

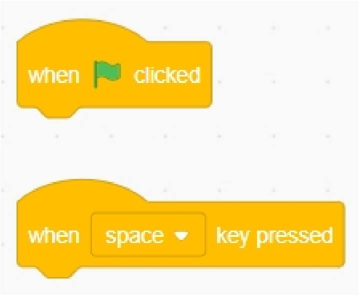


Figure 2.3: Hat Blocks (Event Blocks)

Apart from hat blocks, there are currently two stack blocks placed under the same category which are the “broadcast ()” block and the “broadcast () and wait” block (Figure 2.4). These blocks are mainly used to send a broadcast throughout the whole project, including the Stage and the Sprite that sends the broadcast. The “Broadcast ()” block enables the communication between sprites and scripts by sending broadcasts without any waits [3]. Meanwhile, the “Broadcast () and wait” block works similarly to the “broadcast ()” block, except the fact that it has to wait for all the scripts that are activated by the matching broadcast to end before executing the rest of the script [4].

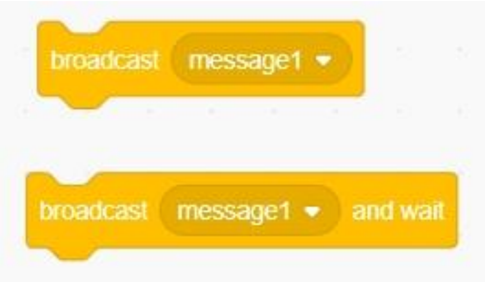


Figure 2.4: Broadcast Blocks (Event Blocks)

2.1.2.2 Control Blocks

The blocks under the Control category are used to control scripts under certain conditions [5]. One of the types of blocks available under this category is the conditional blocks (Figure 2.5). The usage of these blocks will affect the points obtained for Logical Thinking, one of the CT concepts that will be evaluated by the web application.

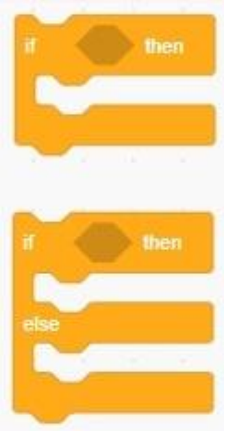
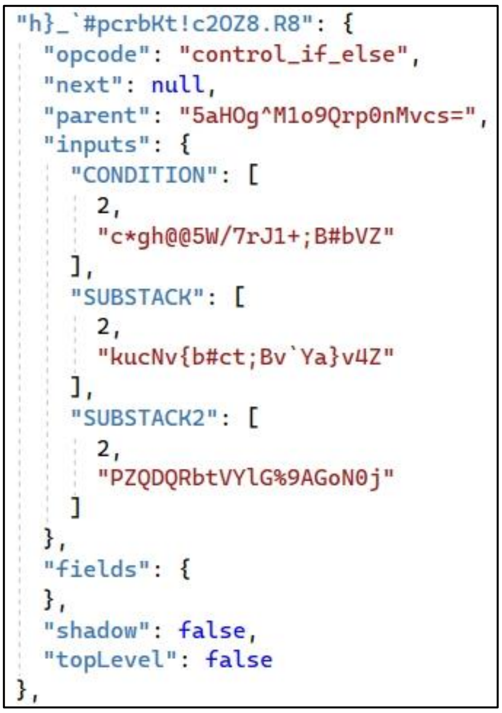


Figure 2.5: Conditional Blocks (Control Blocks)

Figure 2.6 shows the code snippet that stores the information related to the if-else block. As shown in the figure, the key “CONDITION” records the unique ID of the Boolean block used whereas the key “SUBSTACK” and “SUBSTACK2” record the first block placed immediately after true and false conditions. The number 2 in “CONDITION”, “SUBSTACK” and “SUBSTACK2” means that these inputs are visible in the workspace [2]. The values of these keys will be empty or indicated with the keyword “null” if the users did not place any Boolean block to construct the condition or stack block to continue the execution of the script if the condition is true.



```
"h}_`#pcrbKt!c2OZ8.R8": {
  "opcode": "control_if_else",
  "next": null,
  "parent": "5aHOg^M1o9Qrp0nMvcs=",
  "inputs": {
    "CONDITION": [
      2,
      "c*gh@@5W/7rJ1+;B#bVZ"
    ],
    "SUBSTACK": [
      2,
      "kucNv{b#ct;Bv`Ya}v4Z"
    ],
    "SUBSTACK2": [
      2,
      "PZQDQRbtVYlG%9AGoN0j"
    ]
  },
  "fields": {
  },
  "shadow": false,
  "topLevel": false
},
```

Figure 2.6: Code Snippet of If-Else block

Other than that, another type of block is iteration blocks which are the “forever”, “repeat ()” and “repeat until ()” blocks (Figure 2.7). These blocks will be used to evaluate the Flow Control criteria in the web application.

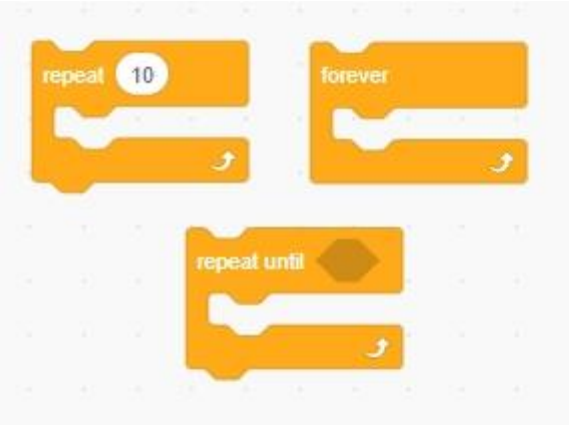
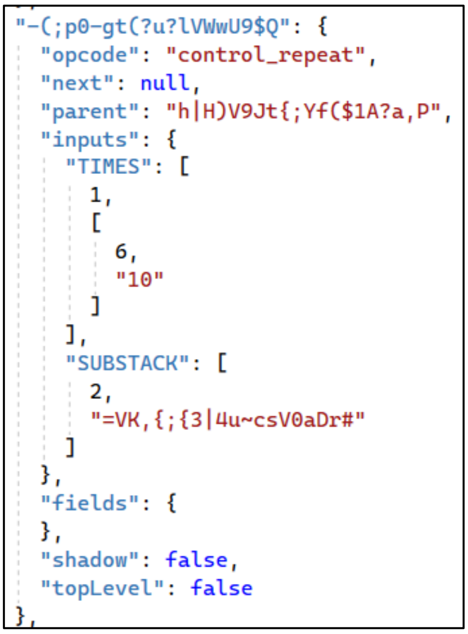


Figure 2.7: Iteration Blocks (Control Blocks)

Figure 2.8 shows the code snippet that stores the information related to the repeat block. As shown in the figure, the key “TIMES” records the desired amount of iteration to be done when the script runs whereas the key “SUBSTACK” records the first block located inside the block. The number 6 in “TIMES” means the block is a positive integer block. The value of“SUBSTACK” will remain empty or indicated with the keyword “null” if the users did not place any stack block to continue the execution of the script when the iteration starts.



```
"-(;p0-gt(?u?lVWwU9$Q": {
  "opcode": "control_repeat",
  "next": null,
  "parent": "h|H)V9Jt{;Yf($1A?a,P",
  "inputs": {
    "TIMES": [
      1,
      [
        6,
        "10"
      ]
    ],
    "SUBSTACK": [
      2,
      "=VK,{;{3|4u~csV0aDr#"
    ]
  },
  "fields": {
  },
  "shadow": false,
  "topLevel": false
},
```

Figure 2.8: Code Snippet of Repeat Block

Besides, there is another type of block which is cloning blocks (Figure 2.9). In Scratch, cloning allows a sprite to clone itself as the script runs. All clones share the same basic information with the parent sprite such as the costumes, scripts and variables, but the changes made during the run time are independent from each other. This type of block will be used to evaluate Decomposition and Abstraction criteria in the web application.

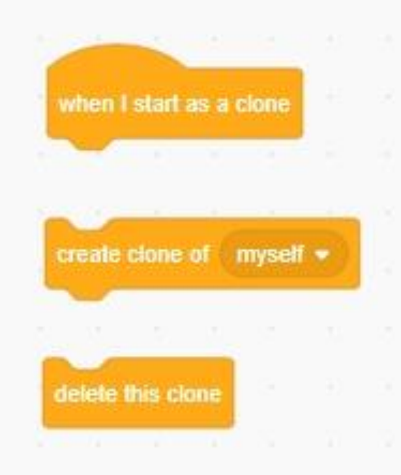


Figure 2.9: Cloning Blocks (Control Blocks)

2.1.2.3 Variable Blocks

The Variable category consists of two major subcategories which are the variable and the list. Users are free to create new variables and lists globally or locally to fit with the content of their projects. The global variable and list will be stored by the stage, whereas the local variable and list will be stored independently by each sprite or clone. However, the stage cannot have any local variables [6]. Table 2.4 shows the information of the array for variable and list block, in which the first element in the array represents the block applied in the JSON file.

Block	1st element	2nd element	3rd element	4th element	5th element
	12	Name	ID	x-coordinate, if it is top-level	y-coordinate, if it is top-level
	13	Name	ID	x-coordinate, if it is top-level	y-coordinate, if it is top-level

Table 2.4: Information of Variable and List Block Represented by Array [2]

2.1.2.4 My Blocks

Scratch also allows users to create a custom block under the category My Blocks. Due to its complexity, it takes more than two unique ID keys to store the information of the block created. Out of all blocks related to this category, the hat block that is visible in the workspace is named "procedures_definition" whereas the block that will be used inside the script to call upon the function is named "procedures_call" (Figure 2.10).

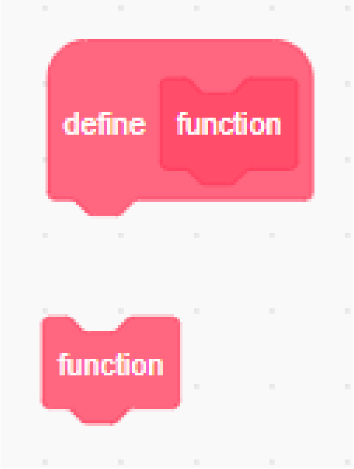


Figure 2.10: User-Made Blocks (My Blocks)

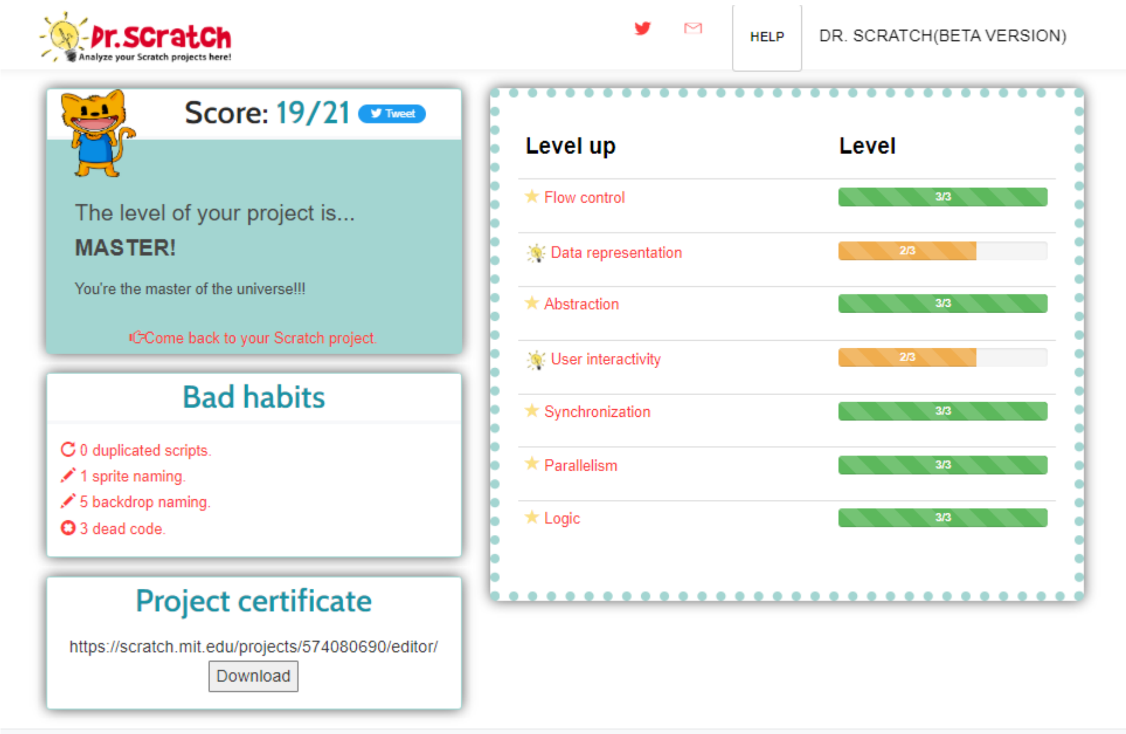
2.1.3 Summary of the Technologies Review

This section mainly talks about the information that is important in evaluating the Scratch projects. The Scratch file format, the JSON tree that represents information of the exported Scratch projects and Scratch blocks that are keys in assigning the points to the criteria are discussed.

2.2 Review of the Existing Systems/Applications

2.2.1 Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking

Dr. Scratch is an open web application developed to automatically analyse and evaluate the competence level of Scratchers' CT concept. By pasting the project URL or uploading the project file to Dr. Scratch, Scratchers will be able to have a clearer understanding of their competence level based on the CT score obtained. Being based on Hairball, a static code analyser that detects potential errors in the project, Dr. Scratch identifies certain bad habits in programming such as duplicated scripts, unused code and default naming [7]. Figure 2.11 shows the dashboard that is shown to Scratchers upon sending their projects for evaluation to Dr. Scratch. The analysed Scratch project is "Cuber's Journey | Part 1 | The volcano" created by a Scratcher named Joshisaurio. As seen in Figure 2.11, the 7 CT concepts that are being evaluated are Flow Control, Data Representation, Abstraction, User Interactivity, Synchronisation, Parallelism and Logic. Each of these concepts contributes at most 3 points to the total score of 21.


Dr.Scratch
Analyze your Scratch projects here!
HELP
DR. SCRATCH(BETA VERSION)
Tweet

Score: 19/21

The level of your project is...

MASTER!

You're the master of the universe!!!

Come back to your Scratch project.

Bad habits

- 0 duplicated scripts.
- 1 sprite naming.
- 5 backdrop naming.
- 3 dead code.

Project certificate

https://scratch.mit.edu/projects/574080690/editor/

Download

Level up	Level
★ Flow control	3/3
Data representation	2/3
★ Abstraction	3/3
User interactivity	2/3
★ Synchronization	3/3
★ Parallelism	3/3
★ Logic	3/3

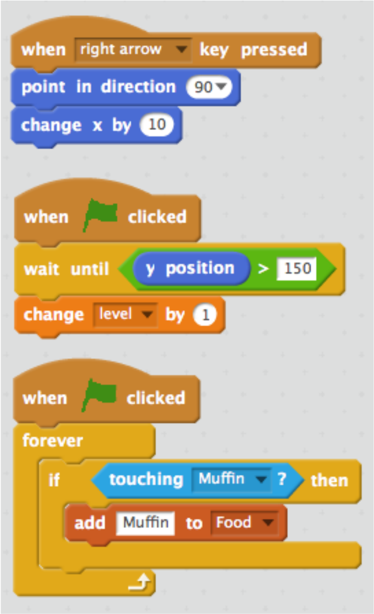
Figure 2.11: Scratch Analysis Result Dashboard

Three levels of CT development which are Basic, Developing and Master are introduced to adjust the feedback needed to be provided by Dr. Scratch to the Scratchers. Table 2.5 shows the rules used by Dr. Scratch in determining the points for the concepts. Figure 2.12 illustrates how Dr. Scratch determines the level of a project.

If Scratchers wish to improve their CT skills, they can do it by accessing the documentation prepared by Dr. Scratch. It provides a brief explanation of the CT concepts and justifies the allocation of marks for each CT concept.

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

Table 2.5: Competence Level for Each CT Concept [7]



```
when right arrow key pressed
point in direction 90
change x by 10

when clicked
wait until y position > 150
change level by 1

when clicked
forever
  if touching Muffin ? then
    add Muffin to Food
```

Figure 2.12: Competence Levels of Data Representation: basic (top), developing (center) and master (bottom) [7]

2.2.2 Towards Data-Driven Learning Paths to Develop Computational Thinking with Scratch

[8] aimed to establish a suitable syllabus for students in primary education by analysing projects under 5 main categories which are animation, games, art, music and stories. 50 projects are randomly chosen from each category and the mastery scores are calculated by using Dr. Scratch. Then, based on the results obtained, clusters are created to further classify the projects to maintain the homogeneity of projects within groups and the heterogeneity between projects of different clusters. In this paper, there are 3 clusters formed. Cluster 1 is mainly formed by animation, art and music projects, followed by Cluster 2 which is by stories and Cluster 3 by games.

Table 2.6 shows the scores of the prototypical projects of the clusters by using Dr. Scratch. From the table, it can be observed that both Cluster 1 and Cluster 2 did not provide any evidence of logic elements. Both share similar characteristics, but Cluster 2 showed a higher level of parallelism and synchronisation due to the increment in the usage of sprites and the occurrence of message passing. Cluster 3 got the highest master score among the clusters as games generally require a higher level of development for computational thinking concepts to make it more interactive, smoother and better ability in storing information throughout the gameplay.

	Cluster		
	1	2	3
Logic	0	0	3
Abstraction	1	1	3
User Interactivity	1	1	2
Data Representation	1	1	2
Parallelism	1	3	3
Synchronisation	1	2	3
Flow Control	2	2	2
Mastery	**7**	**10**	**18**

Table 2.6: Scores of the Prototypical Projects of the Clusters [8]

2.2.3 Framework to Analyse Scratch Project

Scratch School is a framework proposed by [9], which consists of functionalities similar to Dr. Scratch as it analyses the projects based on computational thinking. It also separated the project scores into 3 levels which are Basic, Developing and Proficient. However, Scratch School evaluates projects based on complexity level (Table 2.7) depending on the usage of Scratch blocks and 6 CT concepts (Table 2.8) which are logic, algorithms, decomposition, patterns, abstraction and synchronisation. The maximum marks that can be obtained through complexity level and CT concepts are 26 and 18 respectively. The project level will be determined based on the addition of marks for complexity level and CT concepts.

Elements	Code	Marks
Motion	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
Looks	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
Sound	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
Pen	clear, stamp, pen down, pen up	1
	change pen colour, set pen colour	1
	change pen shade, set pen shade, change pen size, set pen size	1
Data	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
Events	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
Control	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
Sensing	touching, touching colour, colour 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
Operator	+, -, *, /, pick random, mod, round or any mathematical calculation	1
	<, =, >, and, or, not	1
	join, letter of, length of	1
Total		**26**

Table 2.7: Evaluation Scheme for Level of Complexity [9]

Elements	**Marks**		
	3	**2**	**1**
Logic	and, or, not	if else	if
Algorithms	repeat until	repeat, forever	use 2 or more sprites
Decomposition	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
Patterns	set x, set y, set rotation style, set effect, clear graphic effect, set size, set instrument, set volume, set tempo, clear, set pen colour, 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
Abstraction	go to front, go to back	show, hide	switch costume, next costume, switch backdrop
Synchronisation	wait until, when backdrop switches, broadcast and wait	when I receive, broadcast, stop	wait
Total	**18**		

Table 2.8: Evaluation Scheme for Levels of CT Concepts [9]

Besides, the main difference between Scratch School and Dr. Scratch is that Scratch School enables users to create new questions to test the ability of students. It is also capable of storing users’ evaluation results for further reference.

2.2.4 DWES: A Dynamic Weighted Evaluation System for Scratch Based on Computational Thinking

Inspired by [7], [10] proposed a dynamic weighted evaluation system (DWES) which evaluates the Scratch projects based on CT concepts and project type. As shown in Table 2.9, the first modification they performed was the CT concepts. Eight CT concepts which are abstraction and problem decomposition, parallelism, logical thinking, synchronisation, flow control, user interactivity, data representation and code organisation are identified, with code organisation being the modification from [7]. Besides, instead of 3 points, a maximum of 5 points are allocated to these concepts, with each representing different competence levels which are Basic (1 point), Developing (2 points), Familiar (3 points), Mastered (4 points) and Proficient (5 points).

CT Concepts	Competence Levels				
	Basic (1 point)	**Developing (2 points)**	**Familiar (3 points)**	**Mastered (4 points)**	**Proficient (5 points)**
Abstraction and Problem Decomposition	More than one sprite and more than one script	Switch costumes or backdrops to	Make a new block	Use of clones	Use of recursion
Parallelism	Two scripts on green flag	Two scripts on key pressed or on the same sprite clicked	Two scripts on when backdrop switches to	Two scripts on loudness or video motion	Two scripts on when receive message
Logical Thinking	If	If else	Logic operations	Nest logical	Logical nest loop
Synchronisation	Wait, stop all	A script on keyboard or mouse operation, a script when backdrop or custom switches to	A script on touching colour, loudness, video, timer or object properties	Wait until	Broadcast, broadcast and wait, a script when receive message
Flow Control	Sequence of blocks	Repeat, forever	Repeat until	Loop condition contains logic operations	Nest if or if else in a loop, nest loop statements in a loop
User Interactivity	Say, think	Green flag	Keyboard, mouse	Webcam, input sound	Ask and wait, answer

Data Representation	Modifiers of object properties	Join	Variables	Lists	Queues, stacks
Code Organisation	Initialisations of object properties	Rename sprites, backdrops or costumes	Use of comment	No dead code	No useless broadcast

Table 2.9: New CT Evaluation Criteria Proposed [10]

Moreover, through their research, they found that project type is one of the factors that heavily affects the accuracy of CT competence level. Thus, they developed a dynamic weighting tool, where the CT score will be optimised based on the project type.

2.2.5 Summary and Limitation of the Existing Systems

Although a majority of students from the workshops conducted by J. Moreno-León et al. found Dr. Scratch attractive (74.1%) and easy to use (86.9%) [7], there are still some weaknesses that lie behind the website. First of all, Dr. Scratch does not evaluate Scratch projects based on project type. This is a critical issue as different types of projects focus on different CT concepts [8, 10]. For example, Music and Animation types of projects are weak in logical thinking whereas Games are balanced in all aspects [10]. This might contribute to a loss of confidence and interest in programming among children as they are unable to identify ways to improve themselves. Therefore, [10] modified the evaluation criteria of Dr. Scratch and introduced a new tool to classify the types of projects to increase the accuracy of the evaluation system. The evaluation criteria proposed by [10] are much better and more organised than Dr. Scratch, but it is not suitable to analyse Scratch projects as Scratch generally caters for children in the age range of 8 to 16 to develop their mathematical and computational thinking skills. Children might find it difficult to understand the evaluation criteria as most of them are not equipped with adequate programming knowledge.

On the other hand, [9] introduced another framework to evaluate Scratch projects by taking the complexity level into account. This increases the accuracy of Scratch project evaluation, but it does not focus on determining students' level of understanding of CT concepts. It also potentially introduces unfairness in the evaluation system as not all projects will utilise all the blocks provided by Scratch and are extremely vulnerable to version changes as the complexity level is dependent on the type of blocks.

Besides, Dr. Scratch does not permanently save the evaluation results. The statistics page in Dr. Scratch shows only the total CT scores earned in all analysed Scratch projects, average points for each CT concept and code smells of programming skills. Students are unable to check on the specific project analysed to get a better understanding of the mistakes made and improve themselves. In other words, they are unable to make comparisons between projects that are analysed.

CHAPTER 3

System Methodology/Approach

3.1 Methodology Used

Agile methodology is used to develop the web application in this project. It manages a project by separating it into different phases and requires constant collaboration with stakeholders while making improvements at every stage [11]. It has high adaptability to constantly changing user requirements and reduces the delivery time through rapid development and delivery. This methodology is considered the most suitable to run the project as the web application requires rapid development and delivery time. This methodology also gives more flexibility in changing requirements throughout the development lifecycle.

During the planning phase, research is done to investigate the existing software that can evaluate Scratch projects and the information contained inside the exported Scratch projects. Then, suitable methodology and requirements are decided during the analysis phase. Various diagrams such as an architectural diagram, block diagram, entity relationship diagram (ERD), use case diagram and activity diagram are sketched to illustrate the requirements and designs of the systems. From these diagrams, several main functions (evaluating projects and storing and retrieving evaluation results) are identified and a prototype is developed with incremental functions after the requirement gathering, analysis and design phase. Based on the feedback, the prototype is further developed with new improvements. The process of implementing the designs into the code, gathering feedback and improving the prototype is repeated until a satisfactory outcome is obtained. During the implementation, testing is constantly conducted to prevent critical defects that might delay the delivery of the web application. The development of the web application is considered completed when it covers all the functional and non-functional requirements and passes the system test.

3.2 System Design Diagram/Equation

3.2.1 System Architecture Diagram

Figure 3.1 is the architectural diagram used to map out the implementation of the components of the web application. As shown in the diagram, users will access the web application through their web browsers. The Microsoft Identity will assist in authorising the users and allow their access once their accounts are found in the database. If they are teachers or students, they will gain access to the Project Evaluation Module, Project Result Module and Project Report Module. Users with teacher roles can access an additional module called Project Activity Module to create and manage activities related to Scratch projects. If they are administrators, they will gain access to the User Role Management Module, Class Management Module and User Management Module. These modules are connected to the database counterpart, in which the Entity Framework and SQL Server work together to store the information of the system and enable the retrieval of the information.

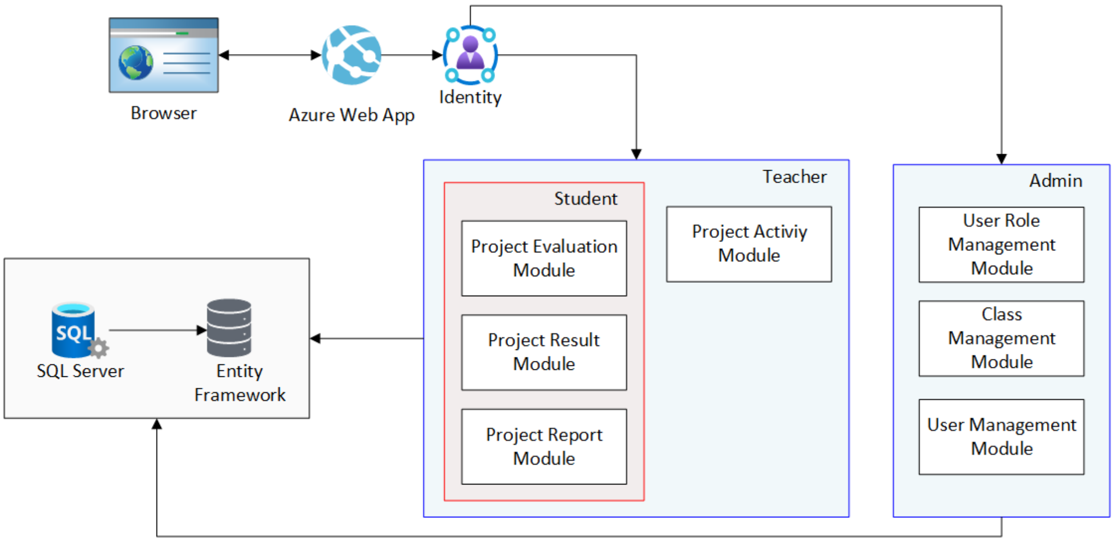


Figure 3.1: Architectural Diagram

3.2.2 Use Case Diagram

As shown in Figure 3.2, teachers and students share similar access to the system functionalities. They can upload a project, view a result and download a result. Students can compare results while teachers can manage activities by adding new activities for students to submit their projects, changing activity status (active and inactive) and checking information about the activities including viewing the submission status of the classes in charge and class performance. They can view the class performance based on the projects uploaded by students and download the report. Meanwhile, the administrators can manage the user role, class and details of users.

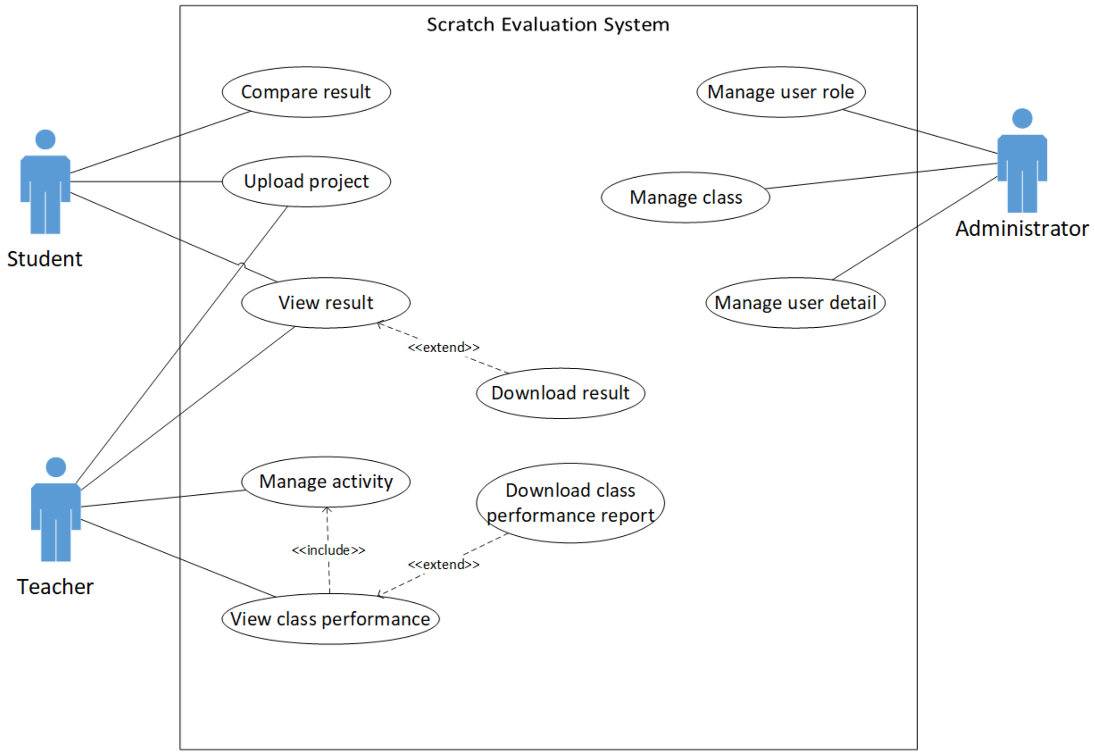


Figure 3.2: Use Case Diagram

3.2.3 Use Case Description

<table>
<tr><td><b>Use Case ID</b></td><td>UC001</td><td><b>Version</b></td><td>1.0</td></tr>
<tr><td><b>Use Case Name</b></td><td colspan="3">Compare Result</td></tr>
<tr><td><b>Purpose</b></td><td colspan="3">To allow students to compare in between two results.</td></tr>
<tr><td><b>Actor</b></td><td colspan="3">Student</td></tr>
<tr><td><b>Relationships</b></td><td colspan="3">N/A</td></tr>
<tr><td><b>Trigger</b></td><td colspan="3">Students click the “Compare Result” button on the Project Result Page, Home Page or from the navigation bar.</td></tr>
<tr><td><b>Precondition</b></td><td colspan="3">The student is assigned to a class.</td></tr>
<tr><td><b>Scenario Name</b></td><td><b>Step</b></td><td colspan="2"><b>Action</b></td></tr>
<tr><td rowspan="7"><b>Main Flow</b></td><td>1</td><td colspan="2">Students click the “Compare Result” button on the Project Result Page, Home Page or from the navigation bar.</td></tr>
<tr><td>2</td><td colspan="2">The system directs students to the Compare Result page.</td></tr>
<tr><td>3</td><td colspan="2">Students click the “Compare Result” button on the Compare Result page.</td></tr>
<tr><td>4</td><td colspan="2">The system retrieves a list of results from the database.</td></tr>
<tr><td>5</td><td colspan="2">Students select two results and click “Compare”.</td></tr>
<tr><td>6</td><td colspan="2">The system retrieves the results chosen.</td></tr>
<tr><td>7</td><td colspan="2">The system displays the information of the results chosen in a table.</td></tr>
<tr><td rowspan="2"><b>Alternate Flow – No Results Found</b></td><td>4.1</td><td colspan="2">The system is unable to find any results related to the students.</td></tr>
<tr><td>4.2</td><td colspan="2">The system displays an error message stating that no results can be found in the database.</td></tr>
<tr><td rowspan="2"><b>Alternate Flow – Only One or No Result Chosen</b></td><td>5.1.1</td><td colspan="2">Students select only one result or no result and click “Compare”.</td></tr>
<tr><td>5.1.2</td><td colspan="2">The system displays an error message stating that students must choose two results to do a comparison.</td></tr>
<tr><td rowspan="2"><b>Alternate Flow – More Than One Result Chosen</b></td><td>5.2.1</td><td colspan="2">Students select more than two results and click “Compare”.</td></tr>
<tr><td>5.2.2</td><td colspan="2">The system displays an error message stating that students can only select two results to do a comparison.</td></tr>
<tr><td><b>Rules</b></td><td colspan="3">N/A.</td></tr>
<tr><td><b>Author</b></td><td colspan="3">Ang Jie Qian</td></tr>
</table>

Table 3.1: Use Case Description for Compare Result (Student)

Use Case ID	UC002	**Version**	1.0
Use Case Name	Upload Project		
Purpose	To allow students to upload a Scratch project.		
Actor	Student		
Relationship	N/A		
Trigger	Students select "Upload Project" on the Home Page or from the navigation bar.		
Precondition	1. The student is assigned to a class. 2. One activity is created by teacher.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Students select "Upload Project" on the Home Page or from the navigation bar.	
	2	The system directs students to the Upload Project page.	
	3	Students fill in a Project Name.	
	4	Students select a Project Type.	
	5	Students select an Activity.	
	6	Students upload a Scratch project file.	
	7	Students click the "Upload Project" button.	
	8	The system retrieves all the input and evaluates the uploaded project.	
	9	The system stores the evaluation result in the database.	
	10	The system shows a message that the upload is successful.	
Alternate Flow – Empty Project Name	3.1	Students did not fill in a Project Name.	
	3.2	The system displays an error message and requests the user to fill in a Project Name.	
	3.3	Back to Main Flow Step 3.	
Alternate Flow – Duplicated Submission for Activities Created by Teachers	5.1	Students have done submissions to the activity before.	
	5.2	The system displays an error message stating that only one submission is allowed for one activity.	
Alternate Flow – Incorrect File Uploaded	6.1	Students did not upload files with a .sb3 extension.	
	6.2	The system displays an error message and requests the students to upload a valid file.	
	6.3	Back to Main Flow Step 6.	
Rules	N/A.		
Author	Ang Jie Qian		

Table 3.2: Use Case Description for Upload Project (Student)

Use Case ID	UC003	**Version**	1.0
Use Case Name	View Result		
Purpose	To allow students to view results.		
Actor	Student		
Relationship	Extend: Download Result		
Trigger	Students select "View Result" on the Home Page or from the navigation bar.		
Precondition	The student is assigned to a class.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Students select "View Result" on the Home Page or from the navigation bar.	
	2	The system directs students to the Result Master List page.	
	3	The system retrieves the result from the database.	
	4	The system shows the list of results.	
	5	Students click on the "View" icon to view details of one of the results.	
	6	The system retrieves the information of the selected results and displays the output.	
Alternate Flow – Download Result	6.1	Students click on the "Download Result" button.	
	6.2	The system generates the result in PDF format.	
	6.3	Students save the file on the computer.	
Alternate Flow – No Results Found	3.1	The system is unable to find any results related to the students.	
	3.2	The system displays an error message stating that no results can be found in the database.	
Alternate Flow – Delete Result	4.1	Students click on the delete button.	
	4.2	The system asks for confirmation to delete the result.	
	4.3	The result is deleted from the database if yes is chosen.	
Rules	N/A		
Author	Ang Jie Qian		

Table 3.3: Use Case Description for View Result (Student)

Use Case ID	UC004	**Version**	1.0
Use Case Name	Upload Project		
Purpose	To allow teachers to upload a Scratch project for students.		
Actor	Teacher		
Relationship	N/A		
Trigger	Teachers select "Upload Project" on the Home Page or from the navigation bar.		
Precondition	The teacher is assigned at least one class.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Teachers select "Upload Project" on the Home Page or from the navigation bar.	
	2	The system directs teachers to the Upload Project page.	
	3	The system retrieves the list of classes related to teachers and populates them into the dropdown list.	
	4	Teachers select a class and click "Select Class".	
	5	The system retrieves the list of students assigned to the selected class and populates them into the dropdown list.	
	6	Teachers select a student.	
	7	Teachers fill in a Project Name.	
	8	Teachers select a Project Type.	
	9	Teachers select an Activity.	
	10	Teachers upload a Scratch project file.	
	11	Teachers click the "Upload Project" button.	
	12	The system retrieves all the input and evaluates the uploaded project.	
	13	The system stores the evaluation result in the database.	
	14	The system shows a message that the upload is successful.	
Alternate Flow – Class Without Students	5.1	The system is unable to retrieve a list of students assigned to the selected class and populates them into the dropdown list.	
	5.2	The system displays an error message stating that no students are assigned to the selected class.	
Alternate Flow – Reselect Class	6.1	Teachers want to reselect the class and click on the "Reselect Class" button.	
	6.2	Back to Main Flow Step 2.	
Alternate Flow – Empty Project Name	7.1	Teachers did not fill in a Project Name.	
	7.2	The system displays an error message and requests the user to fill in a Project Name.	
	7.3	Back to Main Flow Step 7.	
Alternate Flow – Duplicated Submission for Activities Created by Teachers	9.1	Students have done submissions to the activity before.	
	9.2	The system displays an error message stating that only one submission is allowed for one activity.	
Alternate Flow – Incorrect File Uploaded	10.1	Teachers did not upload files with a .sb3 extension.	
	10.2	The system displays an error message and requests the teachers to upload a valid file.	
	10.3	Back to Main Flow Step 10.	
Rules	N/A		

Author	Ang Jie Qian

Table 3.4: Use Case Description for Upload Project (Teacher)

Use Case ID	UC005	**Version**	1.0
Use Case Name	View Result		
Purpose	To allow teachers to view results.		
Actor	Teacher		
Relationship	Extend: Download Result		
Trigger	Teachers select “View Result” on the Home Page or from the navigation bar.		
Precondition	The teacher is assigned at least one class.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Teachers select “View Result” on the Home Page or from the navigation bar.	
	2	The system directs teachers to the Result Master List page.	
	3	The system retrieves the list of classes related to teachers and populates them into the dropdown list.	
	4	Teachers select a class and click “Select Class”.	
	5	The system retrieves the list of students assigned to the selected class and populates them into the dropdown list.	
	6	Teachers select a student.	
	7	The system retrieves the results of the selected student from the database.	
	8	The system shows the list of results.	
	9	Teachers click on the “View” icon to view details of one of the results.	
	10	The system retrieves the information of the selected results and displays the output.	
Alternate Flow – Class Without Students	5.1	The system is unable to retrieve a list of students assigned to the selected class and populates them into the dropdown list.	
	5.2	The system displays an error message stating that no students are assigned to the selected class.	
Alternate Flow – Reselect Class	6.1	Teachers want to reselect the class and click on the “Reselect Class” button.	
	6.2	Back to Main Flow Step 2.	
Alternate Flow – No Results Found	7.1	The system is unable to find any results related to the students.	
	7.2	The system displays an error message stating that no results can be found in the database.	
Alternate Flow – Download Result	10.1	Students click on the “Download Result” button.	
	10.2	The system generates the result in PDF format.	
	10.3	Students save the file on the computer.	
Rules	N/A.		
Author	Ang Jie Qian		

Table 3.5: Use Case Description for View Result (Teacher)

Use Case ID	UC006	**Version**	1.0
Use Case Name	Manage Activity		
Purpose	To allow teachers to create and manage activities.		
Actor	Teacher		
Relationship	Include: View Class Performance Extend: Download Class Performance Report		
Trigger	Teachers select "Activity Manager" on the Home Page or from the navigation bar.		
Precondition	The teacher is assigned at least one class.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Teachers select "Activity Manager" on the Home Page or from the navigation bar.	
	2	The system directs teachers to the Activity Master List page.	
	3	The system retrieves the list of activities related to teachers and displays the output.	
	4	Teachers click on the information button next to the name of the activity.	
	5	The system retrieves the list of classes and the information on the results related to the activity and displays the output.	
Alternate Flow – Change Activity Status	3.1.1	Teachers click on the action buttons (Set Active and Set Inactive).	
	3.1.2	The system updates the status of the activity.	
	3.1.3	Back to Main Flow Step 2.	
Alternate Flow – No Activities Found	3.2.1	The system is unable to find any activities related to the teachers.	
	3.2.2	The system displays an error message stating that no activity can be found in the database.	
Alternate Flow – Create a New Activity	3.3.1	Teachers fill in the name and click the "Add New Activity" button.	
	3.3.2	The system reads the string and creates a new activity in the database.	
Alternate Flow – Delete Student's Submission	5.1.1	Teachers click on the delete button.	
	5.1.2	The system asks for confirmation to delete the result.	
	5.1.3	The result is deleted from the database if yes is chosen.	
Alternate Flow – Download Class Performance Report	5.2.1	Teachers click on the "Download Result" button.	
	5.2.2	The system generates the class performance report in PDF format.	
	5.2.3	Teachers save the file on the computer.	
Rules	N/A.		
Author	Ang Jie Qian		

Table 3.6: Use Case Description for Manage Activity (Teacher)

Use Case ID	UC007	**Version**	1.0
Use Case Name	Manage User Role		
Purpose	To allow administrators to manage the roles of users.		
Actor	Administrator		
Relationship	N/A		
Trigger	Administrator selects “Manage User Role” on the Home Page or “User Role Manager” from the navigation bar.		
Precondition	N/A.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Administrator selects “Manage User Role” on the Home Page or “User Role Manager” from the navigation bar.	
	2	The system directs teachers to the User Role Manager page.	
	3	The system retrieves the list of users and displays the output.	
	4	Administrators click on the “Manage Roles” button.	
	5	The system retrieves a list of roles from the database.	
	6	Administrators assign roles to users and click “Save”.	
	7	The system retrieves the input and stores the updated role in the database.	
Alternate Flow – Conflict with “No Role”	6.1.1	Administrators select “No Role” and any other roles.	
	6.1.2	The system displays an error message stating that “No Role” cannot be assigned with other roles.	
	6.1.3	Back to Main Flow Step 6 to select the valid option.	
Alternate Flow – Conflict with “Admin” and “Teacher” Roles	6.2.1	Administrators select “Teacher” or “Admin” roles with “Student” role.	
	6.2.2	The system displays an error message stating that “Student” cannot be assigned together with “Teacher” and “Admin”.	
	6.2.3	Back to Main Flow Step 6 to select the valid option.	
Alternate Flow – No Roles Selected	6.3.1	Administrators did not select any roles.	
	6.3.2	The system assigns “No Role” to the user.	
	6.3.3	Back to Main Flow Step 7.	
Rules	N/A		
Author	Ang Jie Qian		

Table 3.7: Use Case Description for Manage User Role (Administrator)

Use Case ID	UC008	**Version**	1.0
Use Case Name	Manage Class		
Purpose	To allow administrators to manage the classes.		
Actor	Administrator		
Relationship	N/A		
Trigger	Administrator selects “Manage Class List” on the Home Page or “Class Manager” from the navigation bar.		
Precondition	There must be at least one class.		
Scenario Name	**Step**	**Action**	
Main Flow	1	Administrator selects “Manage Class List” on the Home Page or “Class Manager” from the navigation bar.	
	2	The system directs administrators to the Class Manager page.	
	3	The system retrieves the list of classes and displays the output.	
	4	Administrators click on the “Select Teacher” button.	
	5	The system retrieves the list of active teachers from the database.	
	6	Administrators select a teacher and click “Save”.	
Alternate Flow – Create a New Class	3.1.1	Administrators fill in the name and click the “Add New Class” button.	
	3.1.2	The system reads the string and creates a new class in the database.	
Alternate Flow – Delete Class	3.2.1	Administrators click on the delete button.	
	3.2.2	The system asks for confirmation to delete the class.	
	3.2.3	The result is deleted from the database if yes is chosen.	
Alternate Flow – Edit Class Name	3.3.1	Administrators fill in the name and click the “Edit Name” button	
	3.3.2	The system reads the string and updates the name of the class in the database.	
Alternate Flow – No Active Teachers Available	5.1	The system is unable to find any active teachers in the database.	
	5.2	The system displays an error message stating that no active teacher is available.	
Alternate Flow – No Teacher Selected	6.1	Administrators did not select any teacher and click “Save”.	
	6.2	The system displays an error message stating that one teacher must be selected.	
	6.3	Back to Main Flow Step 5.	
Rules	N/A.		
Author	Ang Jie Qian		

Table 3.8: Use Case Description for Manage Class (Administrator)

Use Case ID	UC009	**Version**	1.0
Use Case Name	Manage User Detail		
Purpose	To allow administrators to manage user details.		
Actor	Administrator		
Relationship	N/A		
Trigger	Administrator selects “Manage Teacher/Student” on the Home Page or “Teacher/Student Manager” from the navigation bar.		
Precondition	N/A		
Scenario Name	**Step**	**Action**	
Main Flow	1	Administrator clicks on the “Manage Teacher/Student” on the Home Page or “Teacher/Student Manager” from the navigation bar and selects either “Teacher List” or “Student List”.	
	2	The system directs administrators to the Teacher List or Student List page.	
	3	The system retrieves the list of teachers or students and displays the output.	
Alternate Flow – Set Inactive Teacher/Student Back to Active	3.1.1	Administrators click on the “Details” button.	
	3.1.2	The system retrieves the information of the selected user and displays the output.	
	3.1.3	Administrators click on the “Set Active” button.	
	3.1.4	The system updates the status of the user in the database.	
Alternate Flow – Delete Teacher/Student	3.2.1	Administrators click on the delete button.	
	3.2.2	The system asks for confirmation to delete the teacher/student.	
	3.2.3	The teacher/student is deleted from the database if yes is chosen.	
Rules	The inputs must be within the valid range.		
Author	Ang Jie Qian		

Table 3.9: Use Case Description for Manage User Detail (Administrator)

3.2.4 Activity Diagram

Figure 3.3, Figure 3.4 and Figure 3.5 show the activity diagram for users with student roles. Meanwhile, Figure 3.6, Figure 3.7 and Figure 3.8 show the activity diagram for users with teacher roles. Figure 3.9, Figure 3.10 and Figure 3.11 show the activity diagram for users with administrator roles.

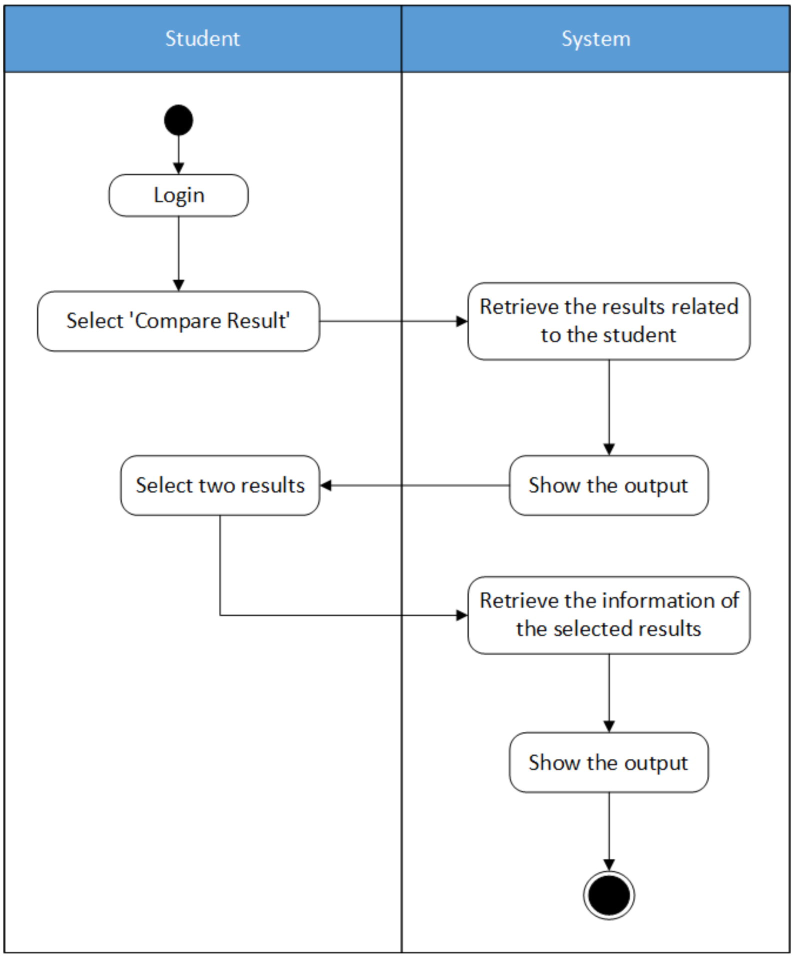

Student
System
Login
Select 'Compare Result'
Retrieve the results related to the student
Show the output
Select two results
Retrieve the information of the selected results
Show the output

Figure 3.3: Activity Diagram of Student Users (Compare Result)

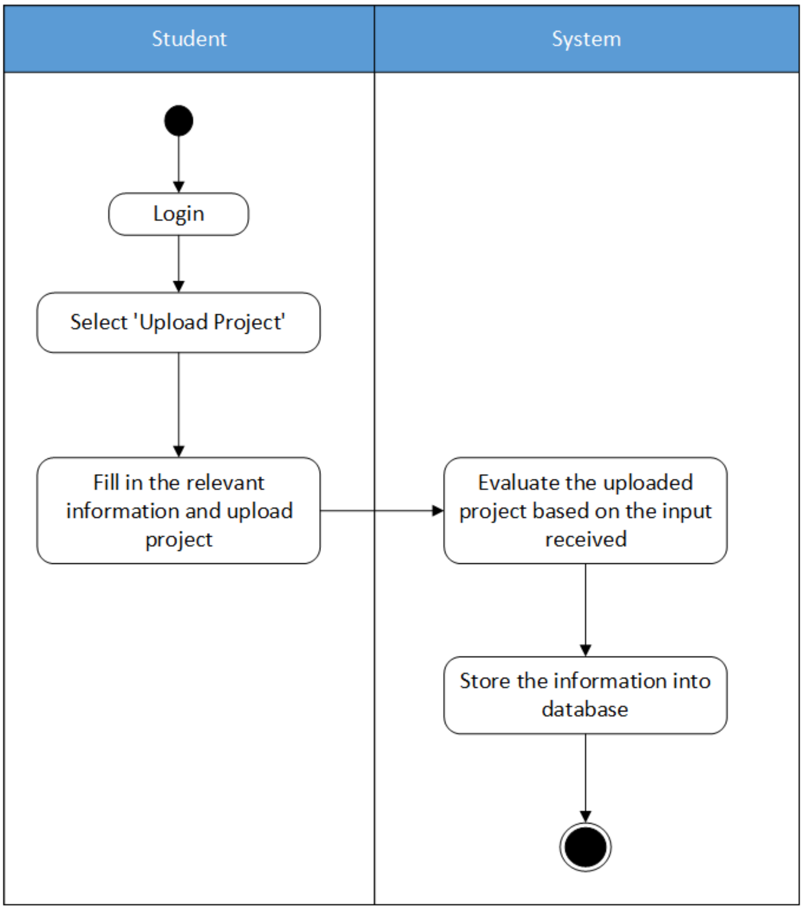

Student
System
Login
Select 'Upload Project'
Fill in the relevant information and upload project
Evaluate the uploaded project based on the input received
Store the information into database

Figure 3.4: Activity Diagram of Student Users (Upload Project)

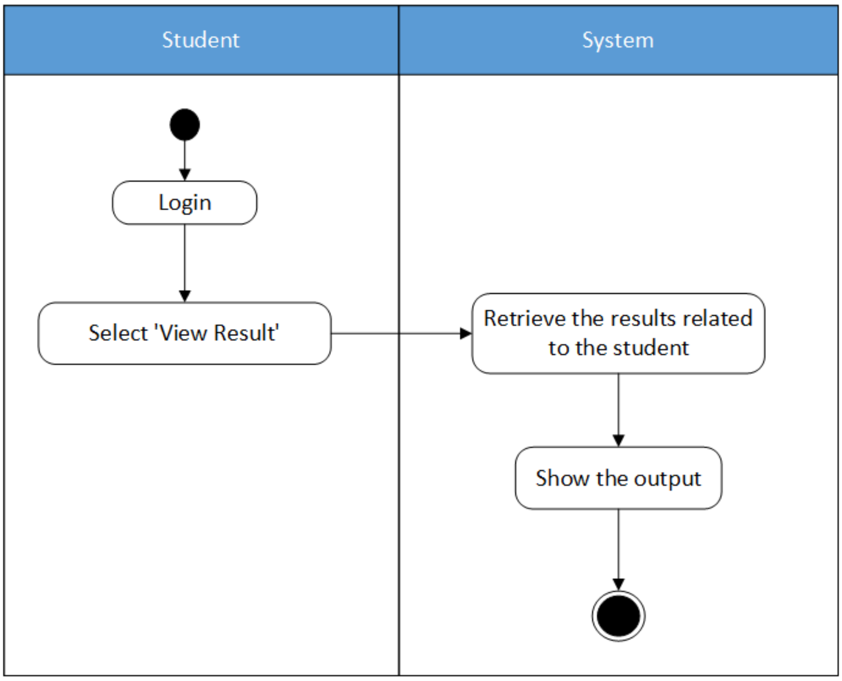


Figure 3.5: Activity Diagram of Student Users (View Result)

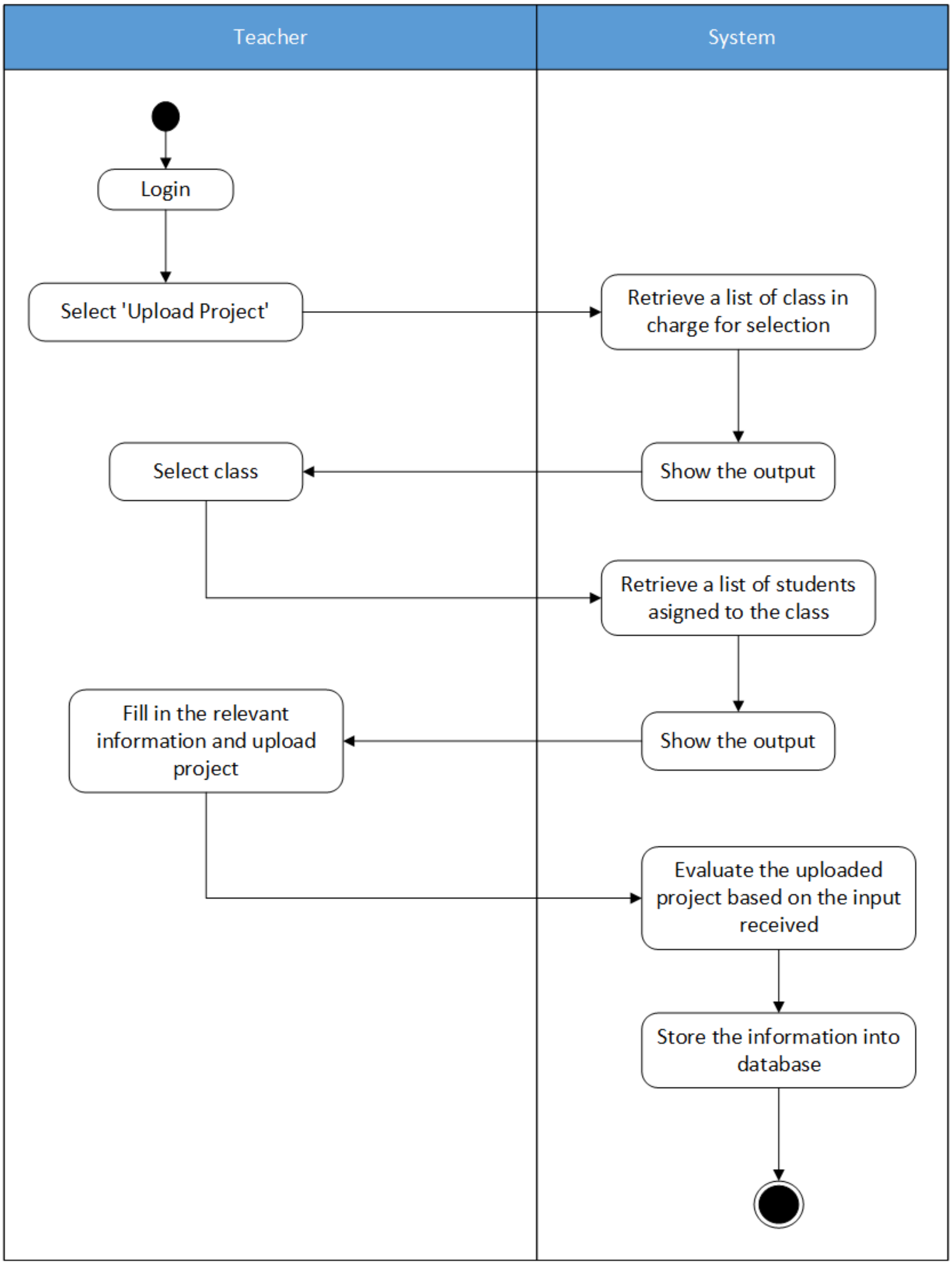

Teacher
System
Login
Select 'Upload Project'
Retrieve a list of class in charge for selection
Select class
Show the output
Retrieve a list of students asigned to the class
Fill in the relevant information and upload project
Show the output
Evaluate the uploaded project based on the input received
Store the information into database

Figure 3.6: Activity Diagram of Teacher Users (Upload Project)

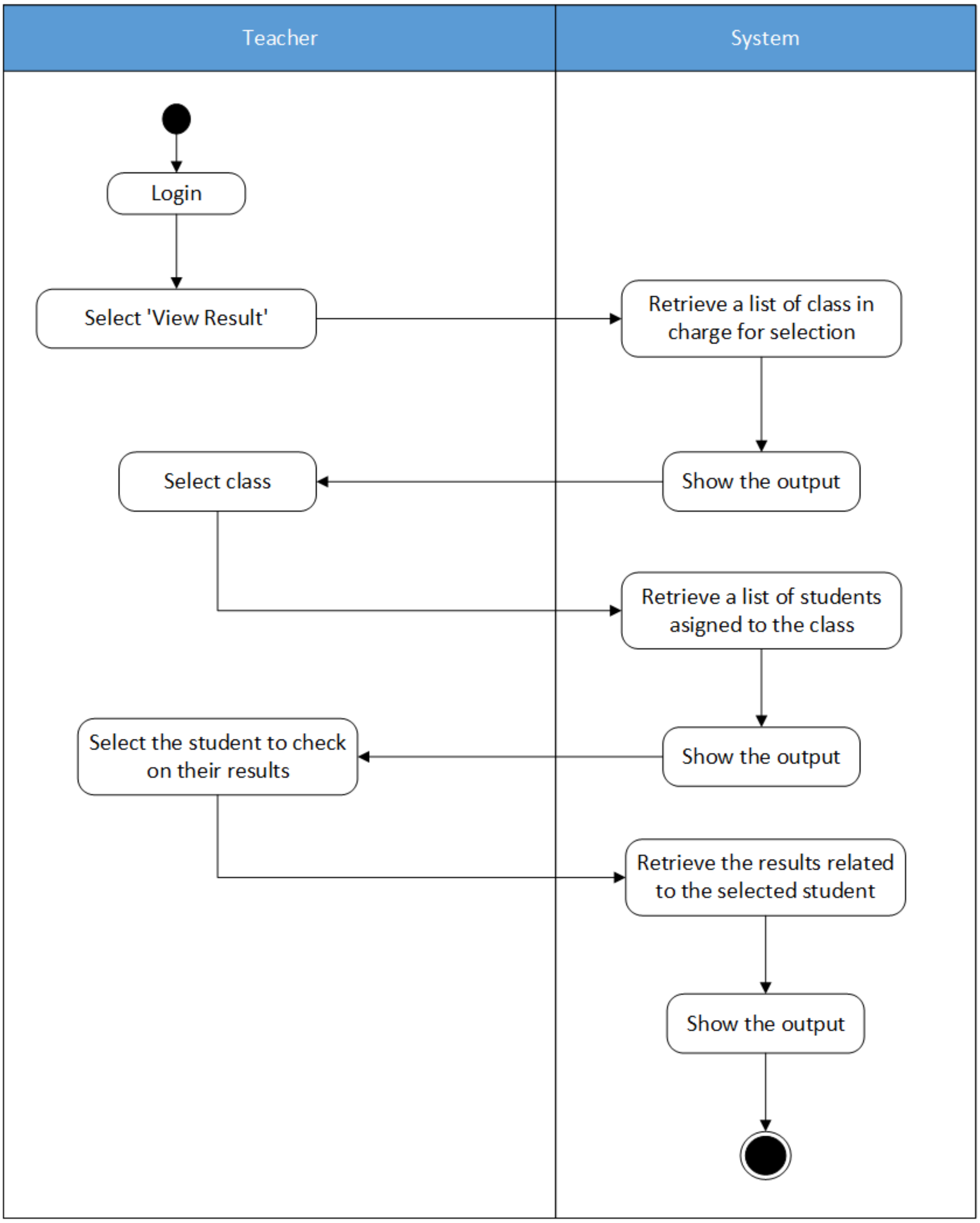

Teacher
System
Login
Select 'View Result'
Retrieve a list of class in charge for selection
Show the output
Select class
Retrieve a list of students asigned to the class
Show the output
Select the student to check on their results
Retrieve the results related to the selected student
Show the output

Figure 3.7: Activity Diagram of Teacher Users (View Result)

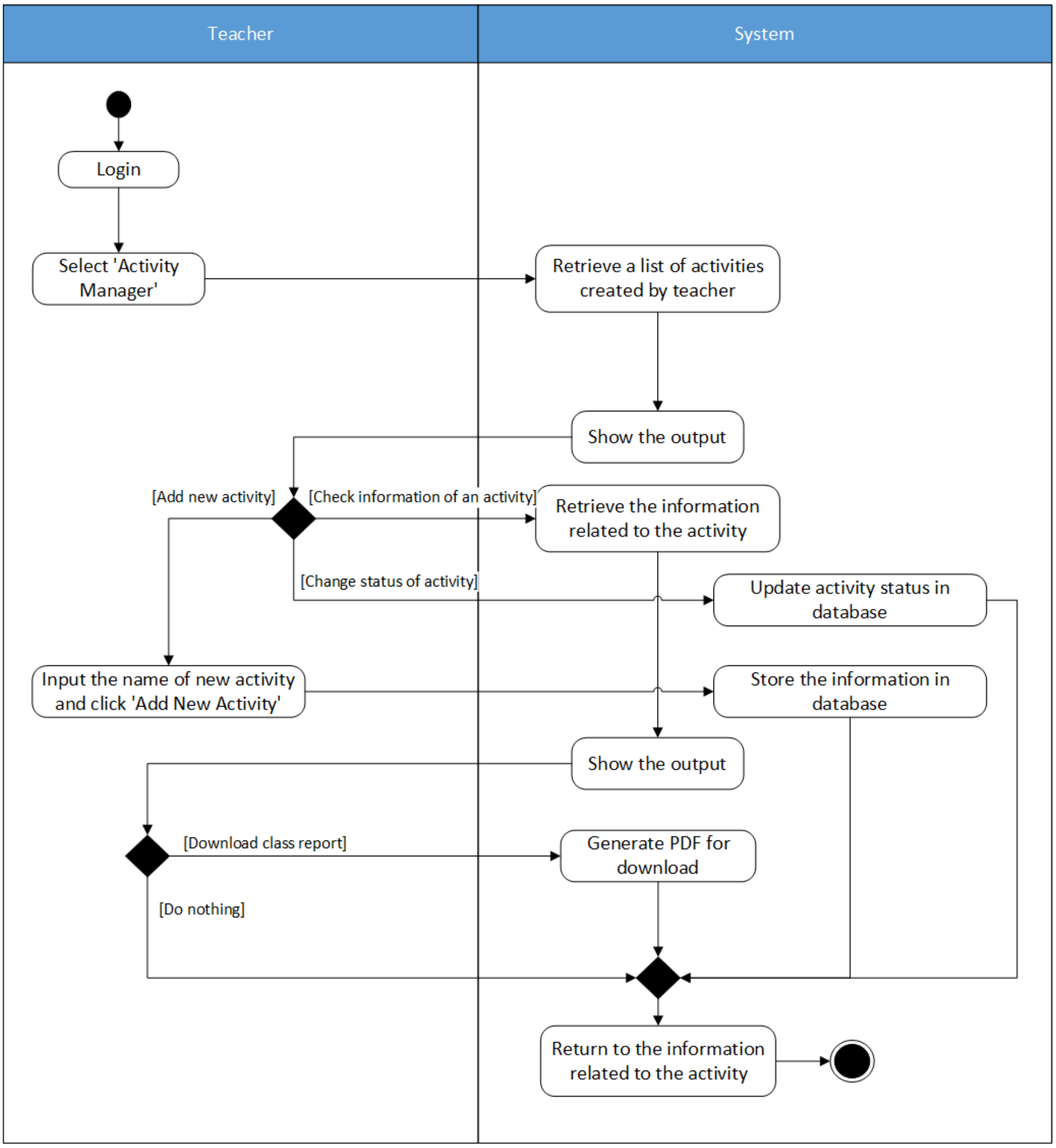

Teacher
System
Login
Select 'Activity Manager'
Retrieve a list of activities created by teacher
Show the output
[Add new activity]
[Check information of an activity]
Retrieve the information related to the activity
[Change status of activity]
Update activity status in database
Input the name of new activity and click 'Add New Activity'
Store the information in database
Show the output
[Download class report]
Generate PDF for download
[Do nothing]
Return to the information related to the activity

Figure 3.8: Activity Diagram of Teacher Users (Manage Activity)

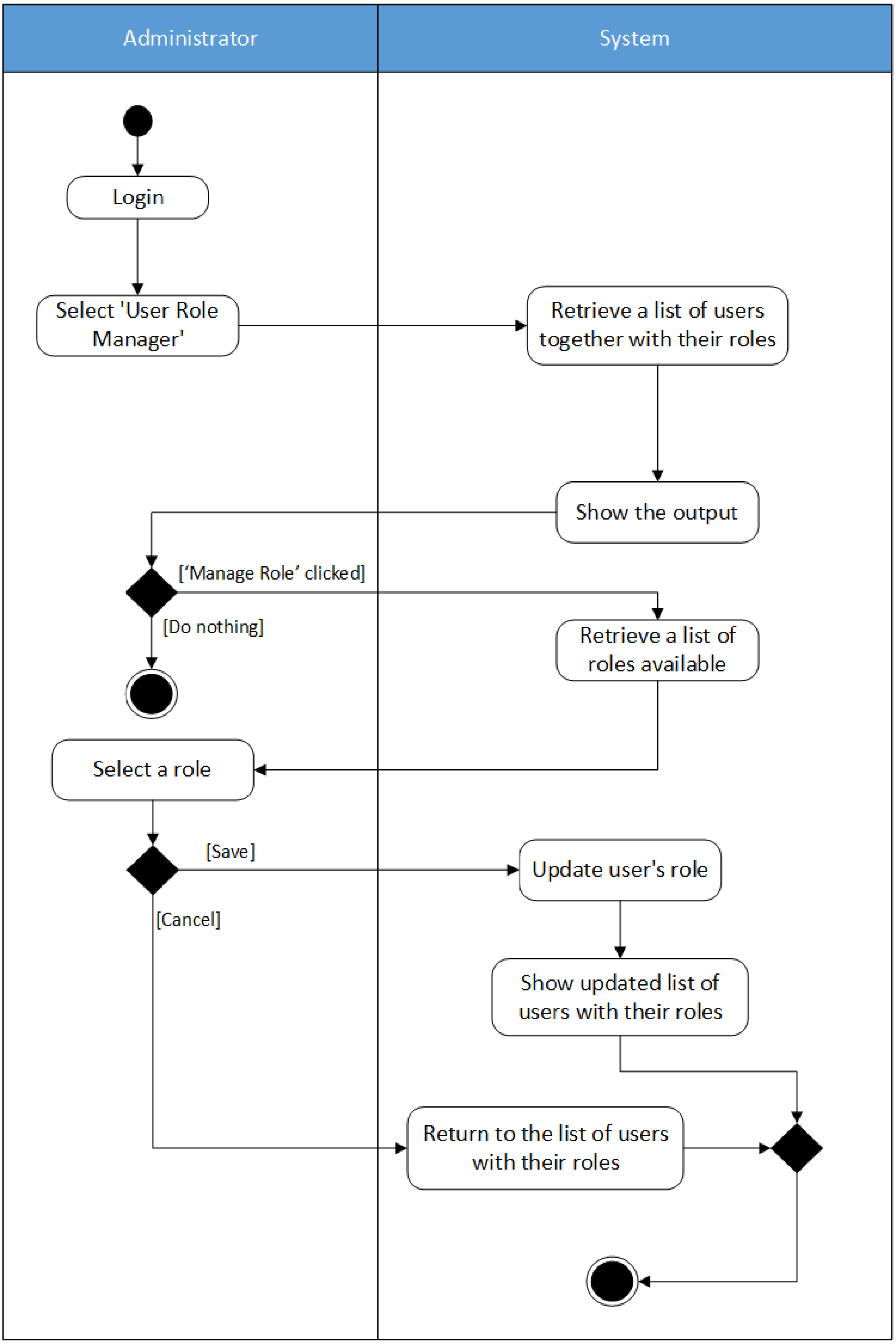

Administrator
System
Login
Select 'User Role Manager'
Retrieve a list of users together with their roles
Show the output
['Manage Role' clicked]
[Do nothing]
Retrieve a list of roles available
Select a role
[Save]
[Cancel]
Update user's role
Show updated list of users with their roles
Return to the list of users with their roles

Figure 3.9: Activity Diagram of Admin Users (Manage User Role)

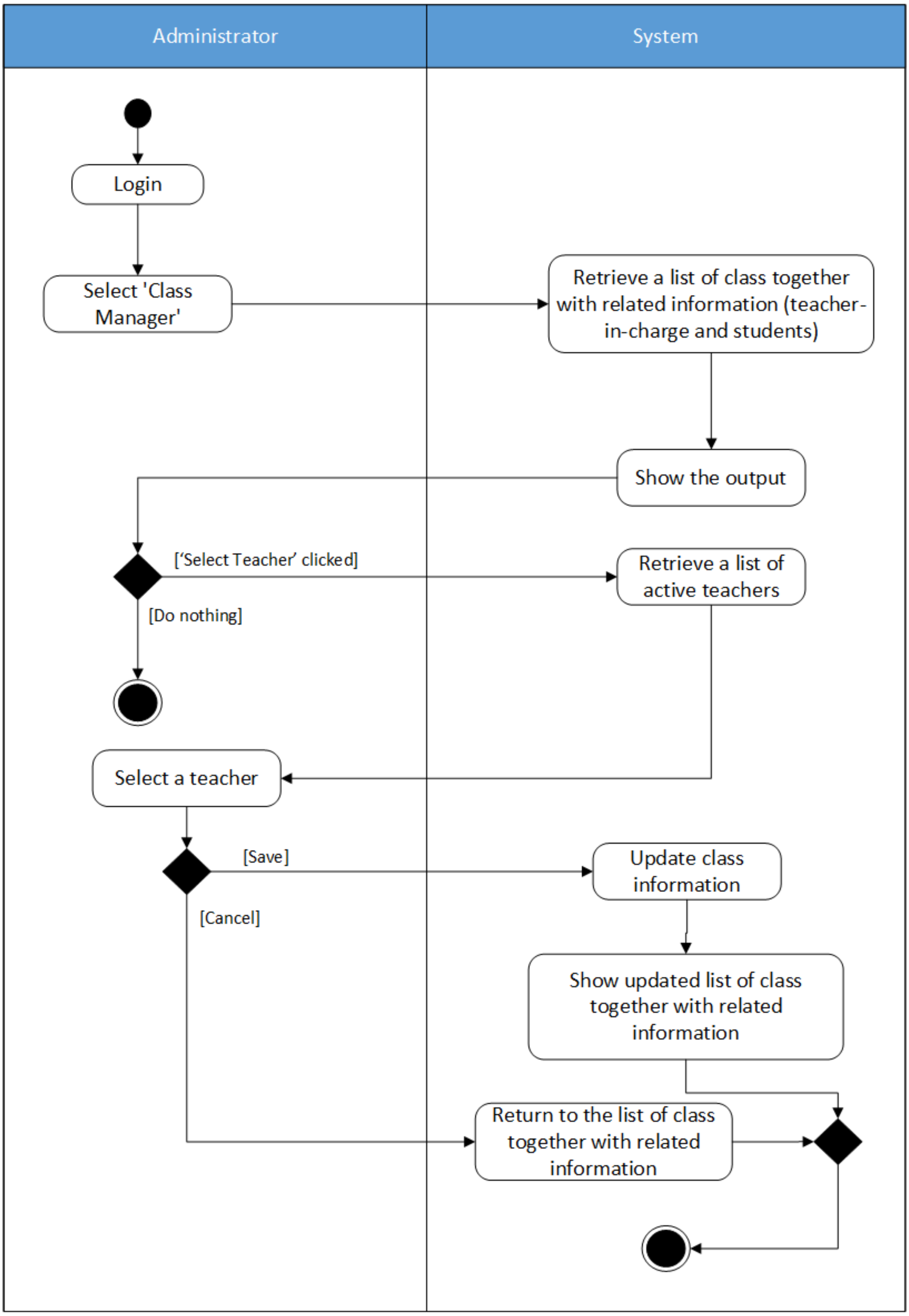

Administrator
System
Login
Select 'Class Manager'
Retrieve a list of class together with related information (teacher-in-charge and students)
Show the output
['Select Teacher' clicked]
[Do nothing]
Retrieve a list of active teachers
Select a teacher
[Save]
[Cancel]
Update class information
Show updated list of class together with related information
Return to the list of class together with related information

Figure 3.10: Activity Diagram of Admin Users (Manage Class)

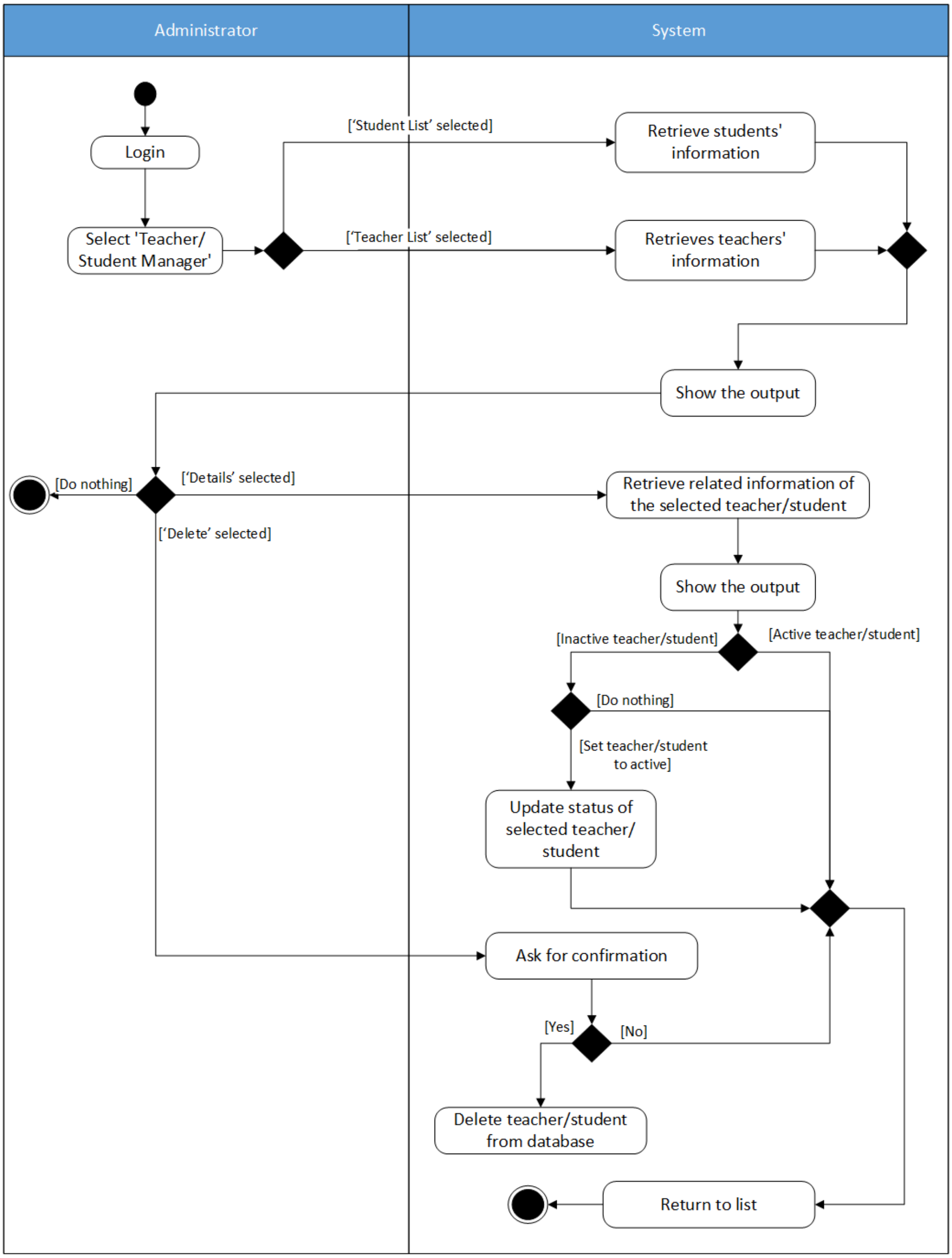

Administrator
System
Login
Select 'Teacher/ Student Manager'
['Student List' selected]
['Teacher List' selected]
Retrieve students' information
Retrieves teachers' information
Show the output
[Do nothing]
['Details' selected]
['Delete' selected]
Retrieve related information of the selected teacher/student
Show the output
[Inactive teacher/student]
[Active teacher/student]
[Do nothing]
[Set teacher/student to active]
Update status of selected teacher/ student
Ask for confirmation
[Yes]
[No]
Delete teacher/student from database
Return to list

Figure 3.11: Activity Diagram of Admin Users (Manage User Details)

3.3 Proposed Solution

This project proposes another type of evaluation criteria, which can increase the accuracy of the system performance by evaluating the projects based on CT concepts and project types. The CT concepts identified by the creators of Scratch, [12] are sequences, events, parallelism, conditionals, operators and data. Based on these concepts, this project benchmarked the evaluation criteria used by [7] and [10], and added a weighting factor which will be used to evaluate different types of projects in a more accurate and precise way. Table 3.10 shows the proposed evaluation table and the points allocated for each criterion. The CT concepts that are being evaluated are user interactivity, logical thinking, decomposition and abstraction, data representation, flow control, synchronisation and parallelism. Redundancy, variable declaration and unused code are used to determine the code smells.

	Competence Level			
Evaluation Criterion	0 point	1 point	2 points	3 points
User Interactivity	Empty workspace	1 Green Flag	Receives user's input (keyboard, mouse)	Usage of audio and video input
Logical thinking	• No conditional statement • Empty conditional statement • Empty workspace	If statement	If-else statement	Logical operator
Decomposition and Abstraction	• Only one sprite and one script • Empty workspace	Use more than 1 sprite and script	Usage of user-defined block	Usage of cloning
Data Representation	Empty workspace	No variable	Usage of variable	Usage of list
Flow control	• Empty loop • Empty condition • Empty workspace	Normal sequence without loop	Loop without condition	Loop with condition
Synchronisation	• Empty workspace	Wait, stop()	Usage of wait until, manual trigger of script and auto trigger of script (when backdrop switch to)	Usage of broadcast

Parallelism	- No 2 scripts that run together with same trigger - Empty workspace	2 scripts run when green flag is clicked	2 scripts run upon receiving user's input	2 scripts that will be triggered without user's input
Redundancy	Empty workspace	If more than one script is repeated	Only one repeated script	No repeated script
Variable declaration	-	If more than one sprite/backdrop is using default name	If only one sprite/backdrop is using default name	No default name used
Unused code	-	- If more than one script is not used - If the workspace has no usable code	If only one script is not used	No unused code

Table 3.10: Proposed Evaluation Table for Each Evaluation Criterion

Apart from that, these criteria will be further evaluated based on the project types chosen, ranging from 1 (least important), 2 (normal) to 3 (most important). A weighting factor of 2 is placed on redundancy, variable declaration and unused code because it is important to reduce the redundancy, usage of default names and unused code, but these habits will not greatly affect the flow of the script.

On the other hand, according to the Scratch site, the main types of projects consist of Animation, Music, Art, Story, Games and Tutorial projects. Art projects present artwork with a simple interaction or no interaction at all. If the project requires a higher level of user interaction and consists of a series of frames, this type of project is commonly labelled as an Animation project, a project that combines a series of motions of objects to create a video [13]. Story projects often fall into the Animation category as this type of project ranges from interactive stories to talk shows [14]. Music projects focus on music by involving imported music files or music generated with the Scratch program, whereas Games projects, being one of the most popular types of projects, feature user interaction in the gameplay. Meanwhile, Tutorial projects are usually designed to teach users the methods to perform a task. This type of project often includes templates to assist Scratchers in producing their related projects [15].

Table 3.11 shows the proposed weighting factor allocated for each type of project during the evaluation. In general, decomposition and abstraction, flow control, synchronisation and parallelism share the same level of importance, which is 3 among the Scratch projects. Meanwhile, Animation, Music and Art projects rarely place focus on logical thinking and data representation. These projects can work without receiving any input from a user, but user interactivity enables others to enjoy more from the projects. On the other hand, Story projects place more importance on data representation as more sprites and backdrops will be used to narrate a story. Lastly, the 7 CT concepts proposed in Table 3.10 weigh the same importance in Games and Tutorial projects to ensure that the flow is smooth when the scripts are running. All concepts are considered as the scripts to produce Games and Tutorial projects are more complicated than the other 4 types of projects.

Weighting Factor			
Project Types	Animation, Music, Art	Story	Games, Tutorials
Code Smell			
Variable Declare	2	2	2
Unused Code	2	2	2
Redundancy	2	2	2
Computational Thinking			
Logical Thinking	1	1	3
Data Representation	1	2	3
User Interactivity	2	2	3
Parallelism	3	3	3
Decomposition & Abstraction	3	3	3
Synchronisation	3	3	3
Flow Control	3	3	3

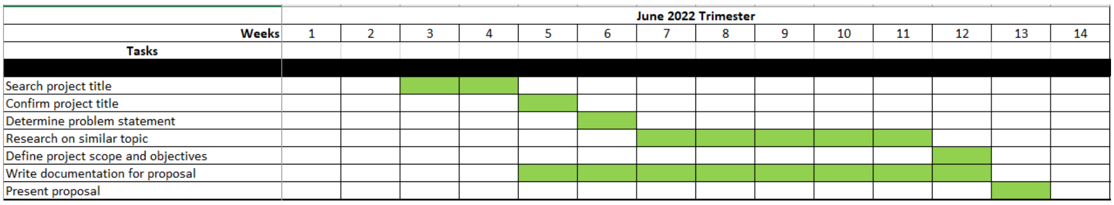
Table 3.11: Proposed Weighting Factor for Each Type of Project

This weighting factor is important to ensure the Scratch projects are evaluated reasonably by converting the project score from fraction to percentage. Based on the percentage obtained, the students will be classified into three types of project levels which are Basic (0% to 35%), Developing (36% to 70%) and Proficient (71% to 100%). The formula used to compute the project score is shown below:

$$Project\ Score = \frac{\sum Code\ Smell\ \times Weighting\ Factor + \sum CT\ Concept \times Weighting\ Factor}{Maximum\ points\ that\ can\ be\ earned\ based\ on\ type\ of\ project} \times 100\%$$

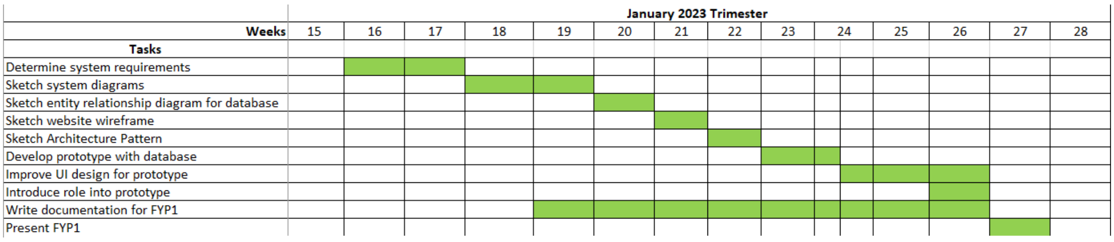
3.4 Project Timeline

Figure 3.12, Figure 3.13 and Figure 3.14 are the Gantt Charts of the project in different phases.



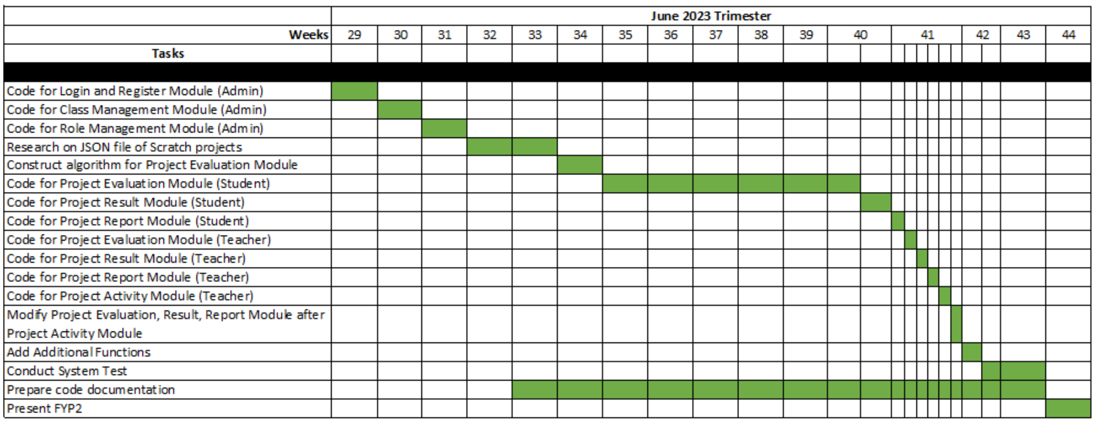
	June 2022 Trimester													
Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Tasks														
Search project title			🟩	🟩										
Confirm project title					🟩									
Determine problem statement						🟩								
Research on similar topic							🟩	🟩	🟩	🟩	🟩			
Define project scope and objectives												🟩		
Write documentation for proposal					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
Present proposal													🟩	

Figure 3.12: Gantt Chart Part 1



	January 2023 Trimester													
Weeks	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Tasks														
Determine system requirements		🟩	🟩											
Sketch system diagrams				🟩	🟩									
Sketch entity relationship diagram for database						🟩								
Sketch website wireframe							🟩							
Sketch Architecture Pattern								🟩						
Develop prototype with database									🟩	🟩				
Improve UI design for prototype										🟩	🟩	🟩		
Introduce role into prototype												🟩		
Write documentation for FYP1					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
Present FYP1													🟩	

Figure 3.13: Gantt Chart Part 2



	June 2023 Trimester															
Weeks	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Tasks																
Code for Login and Register Module (Admin)	🟩															
Code for Class Management Module (Admin)		🟩														
Code for Role Management Module (Admin)			🟩													
Research on JSON file of Scratch projects				🟩	🟩											
Construct algorithm for Project Evaluation Module						🟩										
Code for Project Evaluation Module (Student)							🟩	🟩	🟩	🟩	🟩	🟩				
Code for Project Result Module (Student)												🟩				
Code for Project Report Module (Student)													🟩			
Code for Project Evaluation Module (Teacher)													🟩			
Code for Project Result Module (Teacher)													🟩			
Code for Project Report Module (Teacher)													🟩			
Code for Project Activity Module (Teacher)													🟩			
Modify Project Evaluation, Result, Report Module after Project Activity Module													🟩			
Add Additional Functions														🟩		
Conduct System Test														🟩	🟩	
Prepare code documentation					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	
Present FYP2																🟩

Figure 3.14: Gantt Chart Part 3

CHAPTER 4

System Design

4.1 System Block Diagram

Figure 4.1 shows the block diagram of the web application. As shown in the figure, the users are categorised into three roles, which are Teacher, Student and Administrator. In the web application, both teachers and students can upload projects for evaluation, view and download the results. Teachers can manage activities by creating new activities for the students to submit their projects to and view the submission status of a particular activity and class performance based on the projects uploaded. Meanwhile, students can compare results to check for improvements. As for administrators, they are in charge of managing the roles of the users, assigning the users to their respective classes and updating user's details.

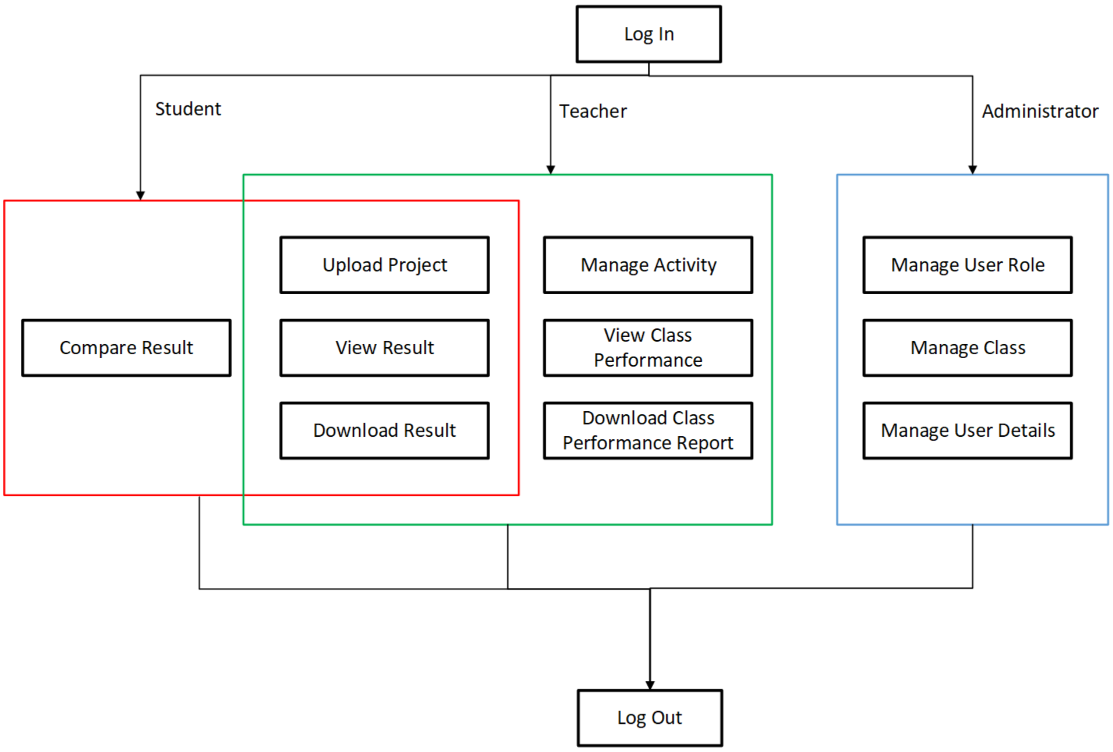


Figure 4.1: Block Diagram

4.2 System Components

4.2.1 Compare Result

In the web application, students can choose any two evaluation results stored in the database. An error message will be prompted if students own less than two evaluation results in the database.

4.2.2 Upload Project

In the web application, teachers and students can upload Scratch projects to evaluate the uploaded projects. The main difference between teachers' user interface and students' interface is teachers have to select a specific class and student before uploading the projects for their students, while the students can directly upload their projects for evaluation after filling in the required information. They can only upload files with the .sb3 extension. By default, the type of activity available in the web application is "Individual Project", which does not restrict the number of projects uploaded to the activity.

4.2.3 View Result

Similar to Upload Project component, teachers and students can view the evaluation results stored in the database. Teachers have to select a specific class and student to view students' results, while the students can directly view their results without any additional steps required.

4.2.4 Download Result

Similar to Upload Project component and View Result component, teachers and students can download the results in PDF form.

4.2.5 Manage Activity

Teachers can create a new activity for students to submit their projects and each student is only allowed to make one submission per activity. If they wish to reupload their projects, they must inform their teacher-in-charge to delete their previous submission before reuploading for evaluation.

To fix the start date and end date of submission, teachers can change the activity status to active and inactive depending on their needs. Students are unable to upload their projects to inactive activities until the teachers set them back to active.

Besides, teachers can view the information on the activities such as the submission status of students and the submitted date.

4.2.6 View Class Performance

Apart from managing activities, teachers can view the class performance as the average project score and project level of each class are shown in the list.

4.2.7 Download Class Performance Report

Teachers can download the class performance in PDF format.

4.2.8 Manage User Role

Administrators can assign roles to users (Admin, Teacher, Student, No Role). The user roles are set as “No Role” by default. A teacher can be an administrator at the same time, but a student cannot be an administrator at the same time. The users will be set back to “No Role” if no roles have been assigned to them.

4.2.9 Manage Class

Administrators can assign teachers and students to classes available. They can check students’ phone numbers and school names.

4.2.10 Manage User Details

Administrators can change the inactive status of students and teachers if necessary. They can also view the teachers' and students' details. If teachers or students no longer exist, administrators can delete them from the database.

4.3 Entity Relationship Diagram

The entity relationship diagram consists of two parts –the default database with entities created upon implementing the Microsoft Identity API (Figure 4.2) and the custom database created based on the requirements gathered (Figure 4.3). In Figure 4.2, most of the entities generated by the Identity API remained as default, with some additional attributes added (Name, AdminID, TeacherID, StudentID, UserDateRegistered and SchoolName) to connect the default Identity database to the custom database and to fit the requirements needed for the web application to function as a Scratch project evaluation application. As shown in Figure 4.3, there are six entities: Admin, Teacher, ActivityType, Result, Class and Student. Admin, Teacher and Student entities store the information of the users based on their roles. Meanwhile, the Result entity records the evaluation results, allowing the teachers and students to review the records anytime. As for the Class entity, it stores the information of the teacher in charge and is used to relate the teachers and the students. The ActivityType entity stores the information on the activities created by teachers, including the active or inactive status of the activity. These entities are related to each other through foreign keys. For instance, the Result entity relates to the Student entity via the student ID (StudentID).

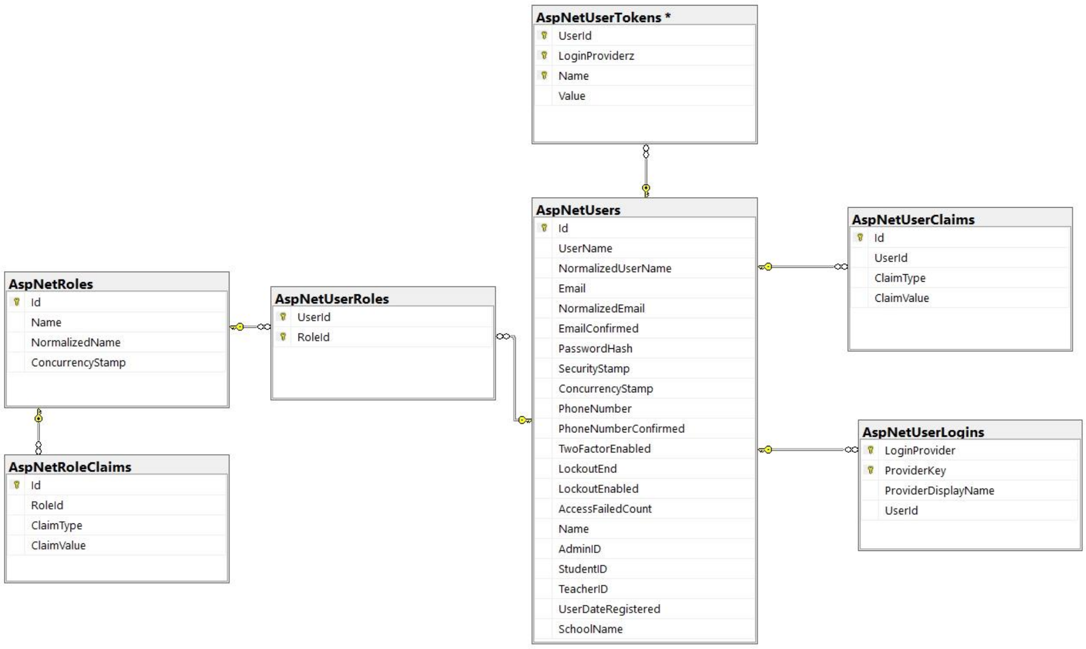


Figure 4.2: Entity Relationship Diagram (Identity Database)

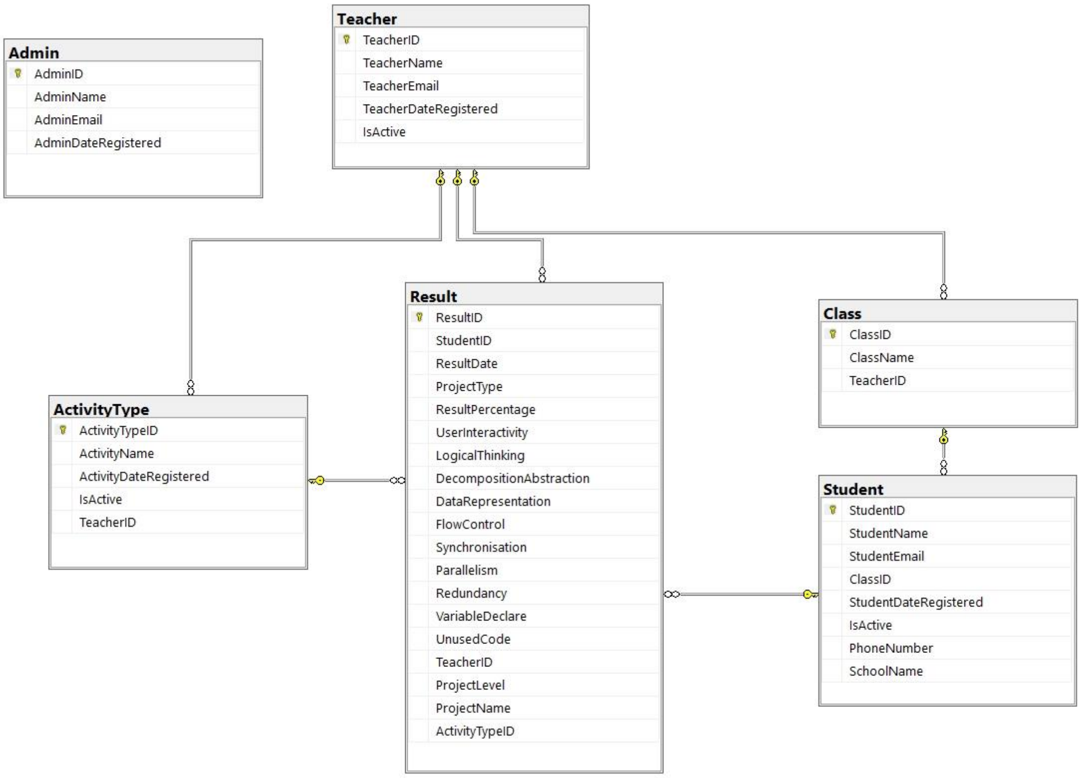


Figure 4.3: Entity Relationship Diagram (Custom Database)

4.4 Algorithm to Evaluate Scratch Projects

The Scratch projects are evaluated based on the blocks used in the projects. The evaluation starts by declaring all variables relevant to the evaluation process, such as arrays that are used to store the points earned in each criterion. The values of the arrays will change after passing through all the evaluation algorithms and saved into the database.

Since some categories will evaluate the projects based on the number of blocks used or the types of blocks used, multiple strings of arrays are declared and used throughout the evaluation process.

To validate the usage of certain blocks, Boolean-type variables are declared to validate the usage of the blocks. For example, for the broadcast function in Scratch projects, there must be a sprite that sends out a message and at least one sprite that receives the message. Missing either of them is considered invalid and points will not be rewarded to the relevant criteria.

Below are the main algorithms used to evaluate the Scratch projects.

4.4.1 Main Flow

```
GET the activity type, project name and project type from user
    IF duplicated submission for one activity && activity type != "Individual Project"
        Return an error message that duplicated submission is not allowed.
CREATE the directory if it does not exist in the web application
CHECK is any file uploaded in the page AND the file extension = ".sb3"
SAVE the uploaded file into the web application
READ the information from the .sb3 file
RETURN
```

4.4.2 Evaluate Variable Declaration (Code Smell)

The criterion “Variable Declaration” checks the number of default names used in the project. The way to evaluate this criterion is by summing up the number of default names used in the uploaded Scratch project.

```
DECLARE NumOfDefaultName = 0
WHILE i != array.Count // the elements in the array represents one stage or one sprite
    DESERIALISE the array to get the properties/keys of one stage or one sprite
    IF array belongs to Sprite && key == “name”
        CHECK does the value contains the string “Sprite”
            IF true
                NumOfDefaultName++
    IF array belongs to Stage && key == “costumes”
        CHECK does the value contains the string “backdrop”
            IF true
                NumOfDefaultName++


IF NumOfDefaultName > 1
    RETURN 1
ELSE IF NumOfDefaultName == 1
    RETURN 2
ELSE
    RETURN 3
```

4.4.3 Evaluate Unused Code (Code Smell)

The criterion “Unused Code” checks the number of unused codes in the project. In a Scratch project, unused codes mean the scripts that do not start with hat blocks. As the blocks under the category of Extension may contain hat blocks, the algorithm also checks whether does the name of the top blocks contains the word “event”.

Firstly, the blocks with “topLevel: true” are gathered in a list. If the block is not an event block or a hat block, meaning that it is an unused script, the number of the variable “unusedCode” will be deducted. The scripts that are considered as unused scripts will be removed from the list for further evaluation purposes.

```
DECLARE unusedCode = 3
WHILE i != topBlockKeys.Count
    CHECK whether the opcode of the block matches the string in event_block array or contains “when” in the string
    IF false
        unusedCode – 1
        REMOVE the key from topBlockKeys
        REMOVE the related list from the nested list that stores the entire information of the project (one list represents one script)

REMOVE empty lists from the nested list

IF unusedCode <= 1
    RETURN 1
ELSE IF unusedCode == 2
    RETURN 2
ELSE
    RETURN 3
```

4.4.4 Evaluate Redundancy (Code Smell)

The criterion “Redundancy” checks the number of redundant codes in the project. In a Scratch project, redundant codes mean the scripts use identical blocks and structures.

```
DECLARE topBlockNum = topBlockKeys.Count
// topBlockKeys is a list of string that stores the key of the first block of a script in a
Scratch project
DECLARE list_A to store the opcode of the blocks placed immediately after the hat
blocks
DECLARE nestedList_A to store the list that represents a duplicated script
// one list represents one script

WHILE i != topBlockKeys.Count
    GET the information of the top block in Json Object
    RETRIEVE and ADD the opcode of the next block into list_A
IF list_A.Count != list_A.Distinct.Count
    GATHER the duplicated opcode of first block into a list (list_B)
    SEARCH for the list related to list_B and store the list in nestedList_A
    EXTRACT opcode into another list (nestedList_B)
    ELIMININATE the duplicated opcode (nestedList_C)

IF nestedList_A.Count – nestedList_C.Count >= 2
    RETURN 1
ELSE IF nestedList_A.Count – nestedList_C. Count == 1
    RETURN 2
ELSE
    RETURN 3
```

4.4.5 Evaluate Logical Thinking (CT Concept)

The criterion “Logical Thinking” checks the usage of conditional-related blocks in the project. In a Scratch project, this criterion is evaluated by detecting the presence of if blocks, if-else blocks and logical operators blocks. If these blocks are used incorrectly, meaning that they are either empty or invalid, this criterion will be marked as 0 regardless of the usage of other valid conditional-related blocks.

```
DECLARE logical_1 = 0, logical_2 = 0, logical_3 = 0
DECLARE list_A to store the node that represents conditional blocks (if and if-else)
DECLARE nestedList_A to store the list that that contains the node
// one list represents one script

WHILE i != blockOpcodeLists.Count
// blockOpcodeLists is a nested list that stores the lists that represents usable scripts
    ADD the list into nestedList_A if the list contains conditional blocks
IF nestedList_A.Count == 0
    RETURN 0

WHILE j != nestedList_A.Count
    ADD the conditional blocks into list_A

WHILE k != list_A.Count
    IF there is no conditional statement
        RETURN -1
    IF opcode == “control_if”
        IF it is an empty conditional statement
            RETURN -1
        logical_1++
    ELSE IF opcode == “control_if_else”
        IF it is an empty conditional statement || it is an incomplete conditional statement
            RETURN -1
        logical_2++
    CHECK does the project uses logical operators
    IF true
        DESERIALISE the block to get the information
        IF opcode == “operator_not”
            IF it is an invalid conditional statement (empty inputs)
                RETURN -1
            logical_3++
        ELSE IF opcode == “operator_and” || opcode == “operator_or”
            IF it is an invalid conditional statement (empty inputs)
                RETURN -1
            logical_3++

IF logical_3 != 0
    RETURN 3
ELSE IF logical_2 != 0
    RETURN 2
ELSE IF logical_1 != 0
    RETURN 1
ELSE
    RETURN 0
```

4.4.6 Evaluate Data Representation (CT Concept)

The criterion "Data Representation" checks the usage of variables and lists in the project. In a Scratch project, this criterion is evaluated by detecting the presence of blocks related to variables and lists, such as set () to () for variables and add () to () for lists. 0 will be given if the project uploaded is an empty project.

```
DECLARE variable = 0, list = 0

WHILE i != nestedList.Count
// nestedList is a nested list that is passed from function call into the function
definition
    CHECK is there any blocks related to variables or lists used
    IF variable used
        variable++
    ELSE IF list used
        list++

IF list != 0
    RETURN 3
ELSE IF variable != 0
    RETURN 2
ELSE
    RETURN 1
```

4.4.7 Evaluate User Interactivity (CT Concept)

The criterion “User Interactivity” checks the level of interactivity between the players and the Scratch projects. The allocation of points is divided into three groups: when a green flag is clicked, when the script receives the user’s input and usage of voice input or webcam.

```
DECLARE green_flag = 0, user_input = 0, voice_audio_input = 0

WHILE i != nestedList.Count
// nestedList is a nested list that is passed from function call into the function definition
    IF opcode == “event_whenflagclicked”
        green_flag++
    ELSE IF blocks that can capture user’s input (keyboard and mouse) is used
        user_input++

IF videoSensing is available in “extension” key
    voice_audio_input++

IF voice_audio_input != 0
    RETURN 3
ELSE IF user_input != 0
    RETURN 2
ELSE IF green_flag != 0
    RETURN 1
ELSE
    RETURN 0
```

4.4.8 Evaluate Parallelism (CT Concept)

The criterion “Parallelism” checks the possibility of several scripts triggering simultaneously. It is evaluated by checking on the hat blocks used in each script. If there is more than one script that starts with the same hat block, meaning that at least two scripts will run simultaneously, the project can be said to achieve parallelism. The allocation of points is divided into three groups: when a green flag is clicked, when the script receives the user’s input and scripts that trigger without the user’s input.

```
DECLARE parallelism = 0
DECLARE list_A to store the node that potentially is one of the blocks related to parallelism

WHILE i != blockOpcodeLists.Count
// blockOpcodeLists is a nested list that stores the lists that represents usable scripts)
    GET the hat block and STORE in list_A

DEFINE array to store the number of hat blocks used with size = 7
WHILE j != list_A.Count
    GET the opcode of the hatblocks
    IF the opcode matches with the 7 hat blocks
        ADD 1 into the array (1 index represents 1 of the hat blocks)

    IF there are at least 2 “event_whenflagclicked”
        parallelism = 1
    IF there is at least 2 “event_whenthisspriteclicked”, “event_whenstageclicked”
        parallelism = 2
    IF the rest of the event blocks is used less than once
        RETURN parallelism

WHILE j != list_A.Count
    GET the opcode of the hatblocks
    IF opcode == “event_whenkeypressed”
```

```
        GET the information and store in list_B
    ELSE IF opcode == “event_whenbackdropswitchesto”
        GET the information and store in list_C
    ELSE IF opcode == “event_whengreaterthan”
        GET the information and store in list_D
    ELSE IF opcode == “event_whenbroadcastreceived”
        GET the information and store in list_E

IF list_B.Count != 0
    CHECK is there duplicated inputs
        IF true
            parallelism = 2
IF list_C.Count != 0
    CHECK is there duplicated inputs
        IF true
            parallelism = 3
IF list_D.Count != 0
    CHECK is there duplicated inputs
        IF true
            parallelism = 3
IF list_E.Count != 0
    CHECK is there duplicated inputs
        IF true
            parallelism = 3
RETURN parallelism
```

4.4.9 Evaluate Decomposition and Abstraction (CT Concept)

The use of self-defined blocks, also known as procedures in Scratch, to perform a function more than once is called abstraction, whereas decomposition means defining objects, methods and functions [10]. The algorithm below only covers the part when there exists a procedure block in the project and the usage of cloning. The condition which will grant 1 point to this criterion is if more than one sprite and one script are used in the Scratch project. It is included in the Main Flow as the algorithm below focuses on evaluating script by script instead of the project as a whole.

```
DECLARE decom_abtract_2 = 0, decom_abtract_3 = 0

WHILE i != nestedList.Count
// nestedList is a nested list that is passed from function call into the function
definition
    CHECK is there any variables or lists used
    IF opcode == “procedures_call”
        decom_abtract_2++
    ELSE IF opcode == “control_create_clone_of”
        decom_abtract_3++

IF decom_abtract_3 != 0
    RETURN 3
ELSE IF decom_abtract_2 != 0
    RETURN 2
ELSE
    RETURN -1
```

4.4.10 Evaluate Synchronisation (CT Concept)

The criterion “Synchronisation” checks how the blocks used in the project coordinate different actions of multiple sprites.

```
DECLARE synchro_1 = 0, synchro_2 = 0, synchro_3 = 0

WHILE i != nestedList.Count
// nestedList is a nested list that is passed from function call into the function definition
    IF opcode == “control_wait” || opcode == “control_stop”
        synchro_1++
    ELSE IF opcode == “event_whenthisspriteclicked” || opcode == “event_whenkeypressed” || opcode == “event_whenstageclicked”
        synchro_2++
    ELSE IF opcode == “event_whenbackdropswitchesto”
        synchro_2++
    ELSE IF opcode == “control_wait_until”
        synchro_2++
    ELSE IF opcode == “event_whenbroadcastreceived”
        synchro_3++

IF synchro _3 != 0
    RETURN 3
ELSE IF synchro _2 != 0
    RETURN 2
ELSE IF synchro _1 != 0
    RETURN 1
ELSE
    RETURN 0
```

4.4.11 Evaluate Flow Control (CT Concept)

The criterion “Flow Control” checks the flow of the scripts. By default, 1 point is granted if there is at least one usable script in the project. However, if the project includes an empty loop, 0 will be given to the criterion even if there exists a usable script or valid loop structures.

```
DECLARE flow_2 = 0, flow_3 = 0
DECLARE list_A to store the node that represents control blocks
// repeat, forever, repeat_until
DECLARE nestedList_A to store the list that that contains the node
// one list represents one script

WHILE i != blockOpcodeLists.Count
// blockOpcodeLists is a nested list that stores the lists that represents usable
scripts)
    ADD the list into nestedList_A if the list contains control blocks

IF nestedList_A.Count == 0
    RETURN 1

WHILE j != nestedList_A.Count
    ADD the control blocks into list_A

WHILE k != list_A.Count
    IF opcode == “control_forever”
        IF it is empty loop statement
            RETURN -1
        flow_2++
    ELSE IF opcode == “control_repeat”
        IF there is no loop statement || it is an empty loop statement
            RETURN -1
        flow_3++
    ELSE IF opcode == “control_repeat_until”
```

```
        IF there is no conditional statement || it is an empty loop statement
            RETURN -1
        IF logical operator is used but it is invalid
            RETURN -1
        ELSE
            flow_3++

IF flow_3 != 0
    RETURN 3
ELSE IF flow_2 != 0
    RETURN 2
ELSE
    RETURN 1
```

CHAPTER 5

System Implementation

5.1 Hardware Setup

The hardware involved in this project is a computer. The development of the web application and Table 5.1 shows the specifications of the laptop used to develop the web application.

Description	Specifications
Model	HP 15S-DU3024TX Notebook
Processor	Intel i5-1135G7
Operating System	Windows 10 Home Single Language
Graphic	NVIDIA GeForce MX350
Memory	16GB RAM
Storage	512GB SSD

Table 5.1: Specifications of Laptop

5.2 Software Setup

The software that is involved in this project is shown below:

- Microsoft Visual Studio Community 2022

  Microsoft Visual Studio Community 2022 is used to develop the web application by using ASP.NET core. Workload-based installer helps to ensure the modules related to the web application development are installed.

  Its user-friendliness improves the experience of creating complex function modules and the ease of debugging the code. It is also capable of connecting the web application to a database to store the relevant information of the web application.

- Microsoft SQL Server 2019

  Microsoft SQL Server 2019 is required to build the database needed for the web application. Since Microsoft Visual Studio Community 2019 is used in creating the web application, Microsoft SQL Server 2019 is a better option than Oracle as both are developed by Microsoft, reducing the problems that arise due to compatibility issues.

- SQL Server Management Studio

  SQL Server Management Studio (SSMS) is an integrated environment to manage all types of SQL infrastructure such as SQL Server and Azure SQL Database. It is required in this project to manage the local database created in SQL Server. Besides, an entity relationship diagram can be generated using SSMS to ensure that the coded database is working as intended.

- Windows 10 Home Single Language

  Minimum operating system required to use Microsoft Visual Studio Community 2022, Microsoft SQL Server 2019 and SSMS.

5.3 Setting and Configuration

5.3.1 Workloads in Visual Studio Installer

Before the creation of the web application, there are a few workloads required to be downloaded in Visual Studio Installer: –

i. ASP.NET and web development

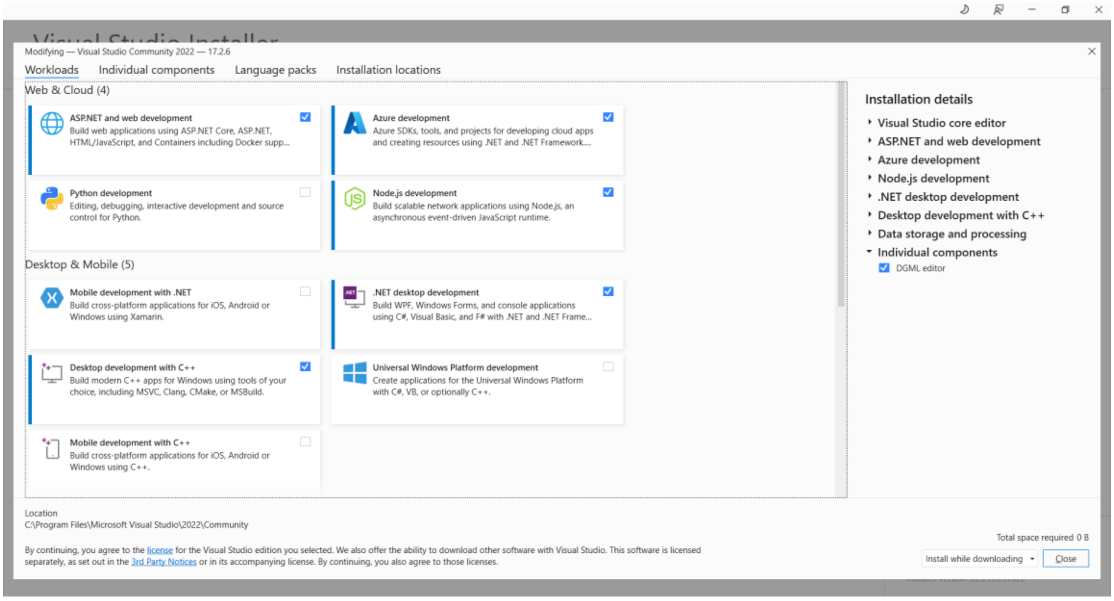
ii. Azure development

iii. Node.js development

iv. Desktop development with C++

v. .NET desktop development

vi. Data storage and processing



Modifying — Visual Studio Community 2022 — 17.2.6

Web & Cloud (4)

☑ ASP.NET and web development
Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...

☑ Azure development
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....

□ Python development
Editing, debugging, interactive development and source control for Python.

☑ Node.js development
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

□ Mobile development with .NET
Build cross-platform applications for iOS, Android or Windows using Xamarin.

☑ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

☑ Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

□ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

□ Mobile development with C++
Build cross-platform applications for iOS, Android or Windows using C++.

Installation details
- Visual Studio core editor
- ASP.NET and web development
- Azure development
- Node.js development
- .NET desktop development
- Desktop development with C++
- Data storage and processing
- Individual components
  - ☑ DGML editor

Location
C:\Program Files\Microsoft Visual Studio\2022\Community

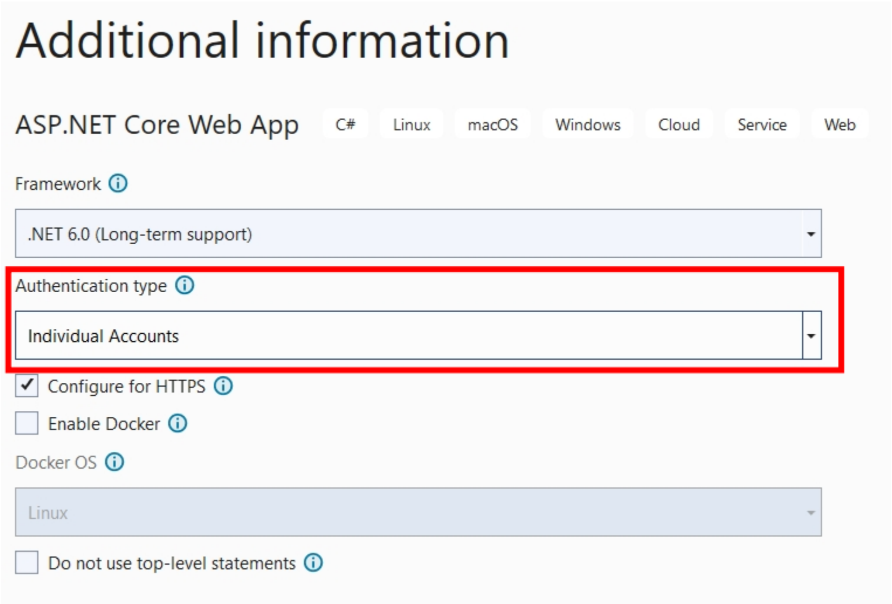
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 0 B

Figure 5.1: Interface of Visual Studio Installer

5.3.2 ASP.NET Core Identity

A type of API that supports user interface login functionality provided in Microsoft Visual Studio Community 2022. Users can create an account with their information stored in the Identity database (Figure 4.2) which is generated by default. All pages related to authorisation and authentication, such as Login, Register and Logout, are provided by default. Further modifications can be done on these pages by scaffolding the desired pages and modifying the scaffolded pages. Figure 5.2 shows the screenshot to implement the API in the web application.



Additional information

ASP.NET Core Web App

Framework: .NET 6.0 (Long-term support)

Authentication type: Individual Accounts

☑ Configure for HTTPS

□ Enable Docker

Docker OS: Linux

□ Do not use top-level statements

Figure 5.2: Setting Up Identity API

5.3.3 Bootstrap and Font Awesome

To develop a web application with creative design, icons from Bootstrap and Font Awesome are used. Bootstrap is given by default upon the creation of the web application, whereas the icons from Font Awesome can be obtained via the Content Delivery Network (CDN).

5.3.4 Entity Framework Core

Entity Framework Core is a lightweight, extensible and cross-platform of the popular Entity Framework data access technology. It enables the web application to work with the database and generates a model from an existing database. The model changes are made available without dropping and re-creating the database via Migrations – a function that compares the current model against the previous model to determine the differences and generates a migration source file to keep the database scheme in sync. This library is provided in the NuGet package in Microsoft Visual Studio Community 2022.

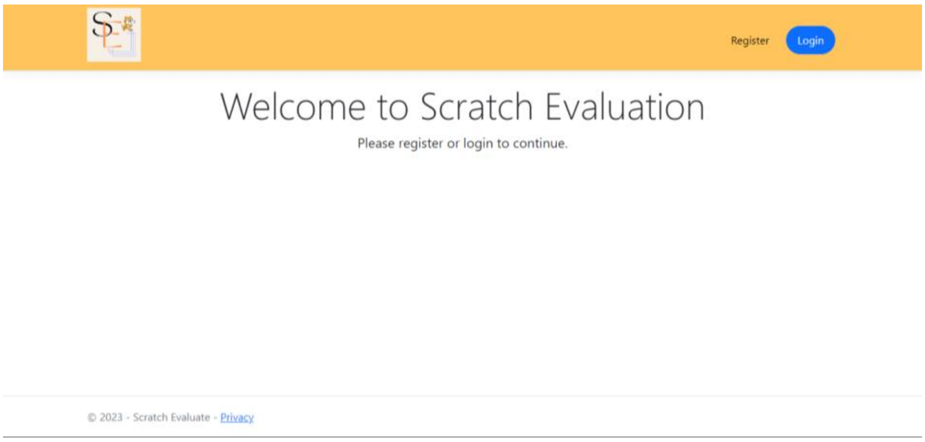
5.3.5 iText7

It is a third-party library given in the NuGet package in Microsoft Visual Studio Community 2022. It helps in manipulating PDF files in ASP.NET, a function not provided in ASP.NET Core by default.

5.4 System Operation (with Screenshot)

5.4.1 Register and Login Operations

Upon assessing the web application, users will be directed to the interface as shown in Figure 5.3. They can choose to register a new account or log in to their existing account by clicking on the buttons provided at the top right corner.



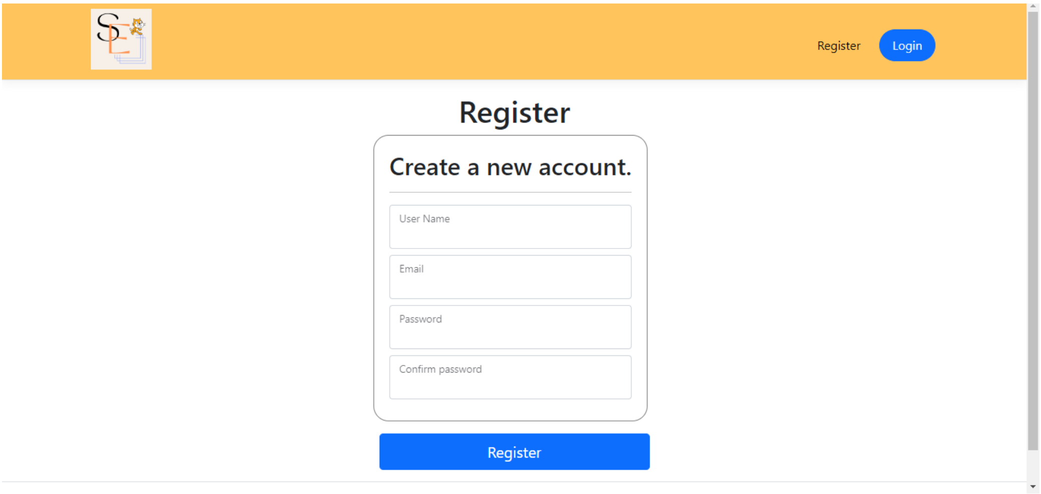
Welcome to Scratch Evaluation

Please register or login to continue.

© 2023 - Scratch Evaluate - Privacy

Figure 5.3: Home Page (Not Logged In)

Figure 5.4 shows the interface of Register and Figure 5.5 shows the interface of Login. Users with student role must fill in their phone numbers and school name in the Personal Information page shown in Figure 5.6. After logging into the web application, it will direct the users to the page as a reminder to fill in their inputs until their phone numbers and school names are not left empty.



Register

Create a new account.

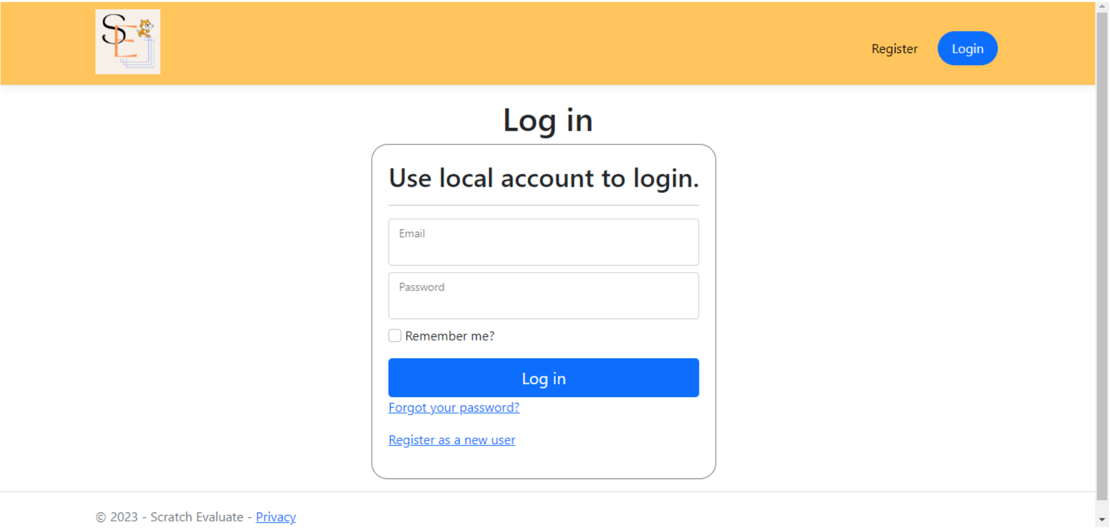
User Name

Email

Password

Confirm password

Figure 5.4: Register Interface



Log in

Use local account to login.

Email

Password

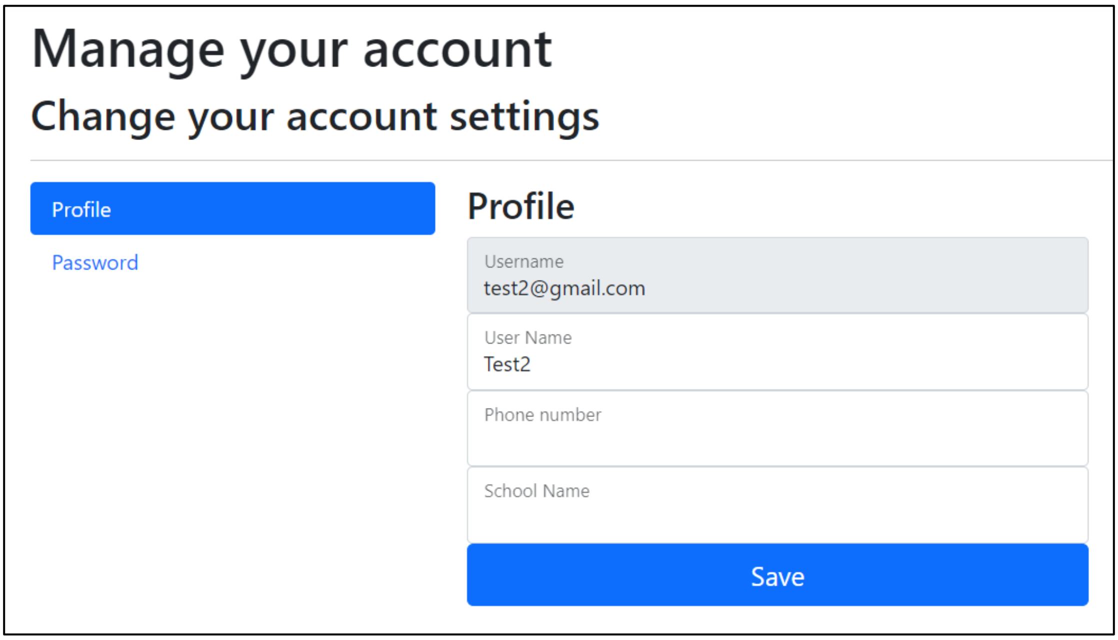
□ Remember me?

Forgot your password?

Register as a new user

© 2023 - Scratch Evaluate - Privacy

Figure 5.5: Login Interface



Manage your account

Change your account settings

Profile

Username
test2@gmail.com

User Name
Test2

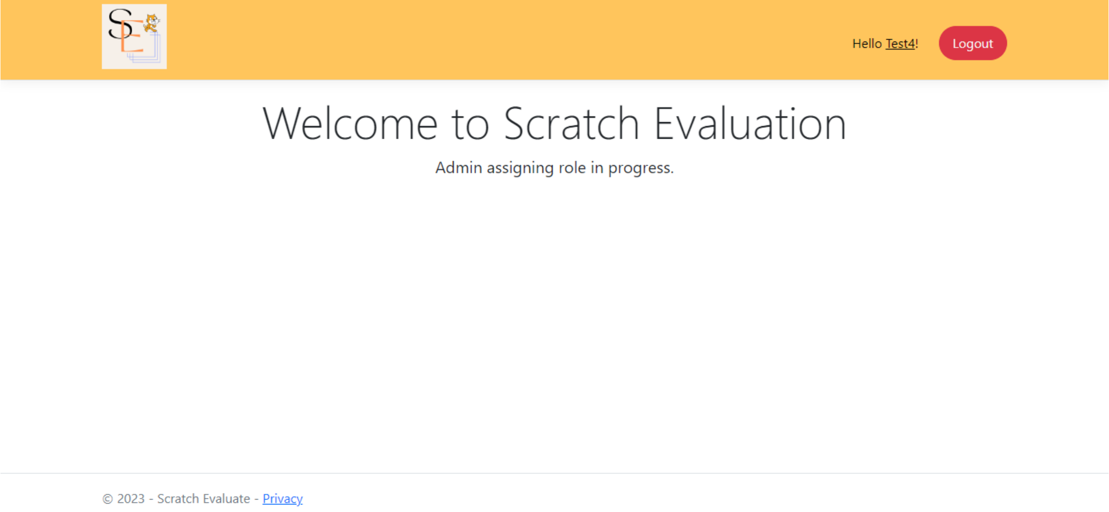
Phone number

School Name

Figure 5.6: Personal Information Interface

5.4.2 Home Page

Upon registering for a new account, the users must wait for the administrator to assign a role to them. Figure 5.7 shows the interface as they wait for role assignment. Figure 5.8, Figure 5.9, Figure 5.10 and Figure 5.11 shows the home page that is displayed to the users based on their roles.

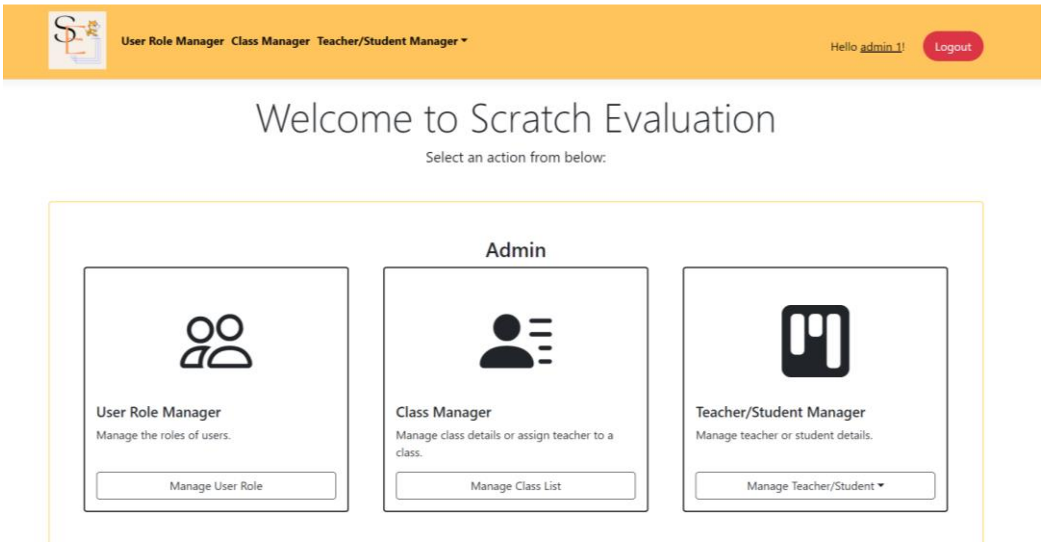


Welcome to Scratch Evaluation

Admin assigning role in progress.

© 2023 - Scratch Evaluate - Privacy

Figure 5.7: Home Page (User without Role)



Welcome to Scratch Evaluation

Select an action from below:

Admin

User Role Manager

Manage the roles of users.

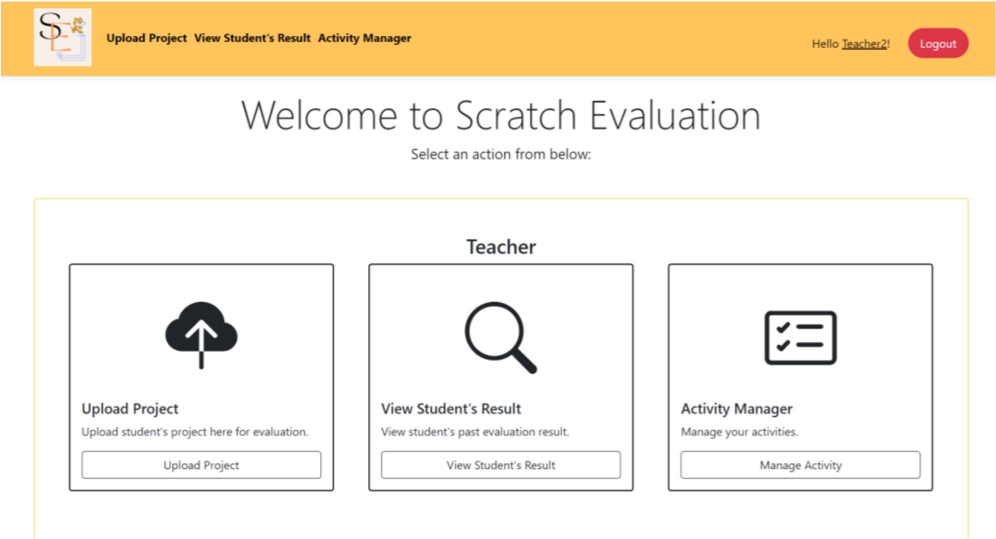
Class Manager

Manage class details or assign teacher to a class.

Teacher/Student Manager

Manage teacher or student details.

Figure 5.8: Home Page (Administrator)



Welcome to Scratch Evaluation

Select an action from below:

Teacher

Upload Project

Upload student's project here for evaluation.

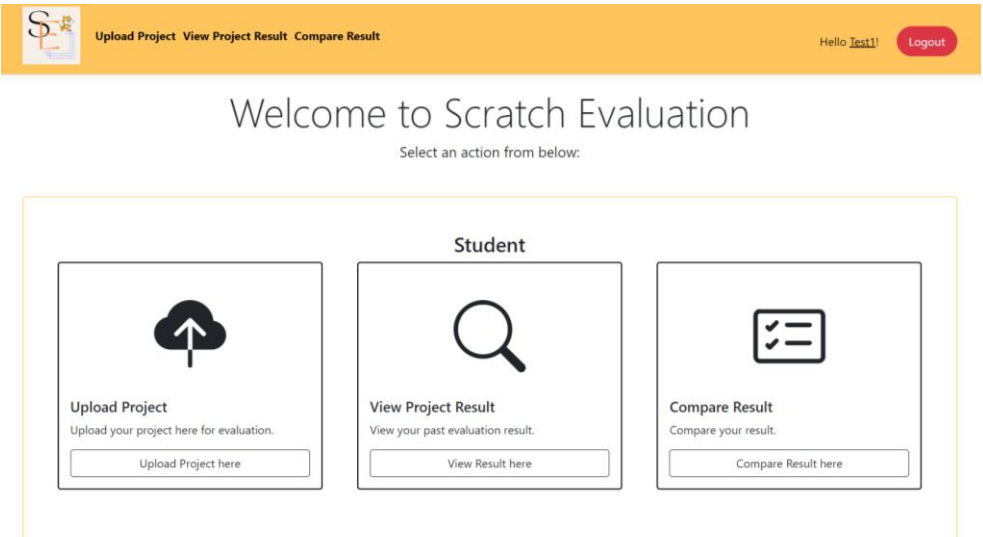
View Student's Result

View student's past evaluation result.

Activity Manager

Manage your activities.

Figure 5.9: Home Page (Teacher)



Welcome to Scratch Evaluation

Select an action from below:

Student

Upload Project

Upload your project here for evaluation.

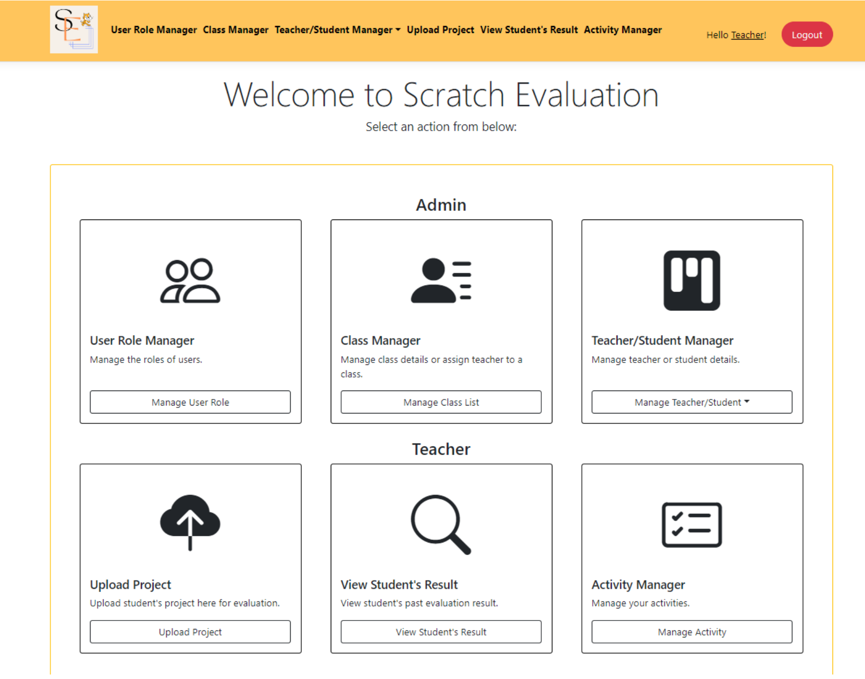
View Project Result

View your past evaluation result.

Compare Result

Compare your result.

Figure 5.10: Home Page (Student)



Welcome to Scratch Evaluation

Select an action from below:

Admin

User Role Manager

Manage the roles of users.

Class Manager

Manage class details or assign teacher to a class.

Teacher/Student Manager

Manage teacher or student details.

Teacher

Upload Project

Upload student's project here for evaluation.

View Student's Result

View student's past evaluation result.

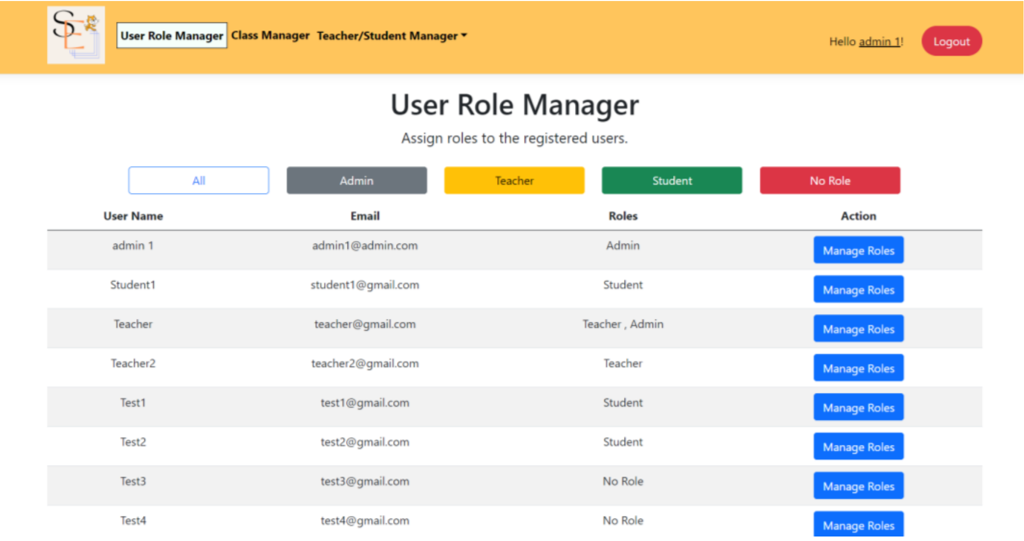
Activity Manager

Manage your activities.

Figure 5.11: Home Page (Teacher & Administrator)

5.4.3 User Role Management Module (Administrator)

A list of registered users is displayed in User Role Manager, together with their email and roles. Administrators can assign roles to the users by clicking on the “Manage Roles” button. They will be directed to the page shown in Figure 5.13 for role selection. Users can be a teacher and an administrator at the same time, but users can only be a student or users without a role. If the selection is left empty, users will be assigned to the “No Role” category.

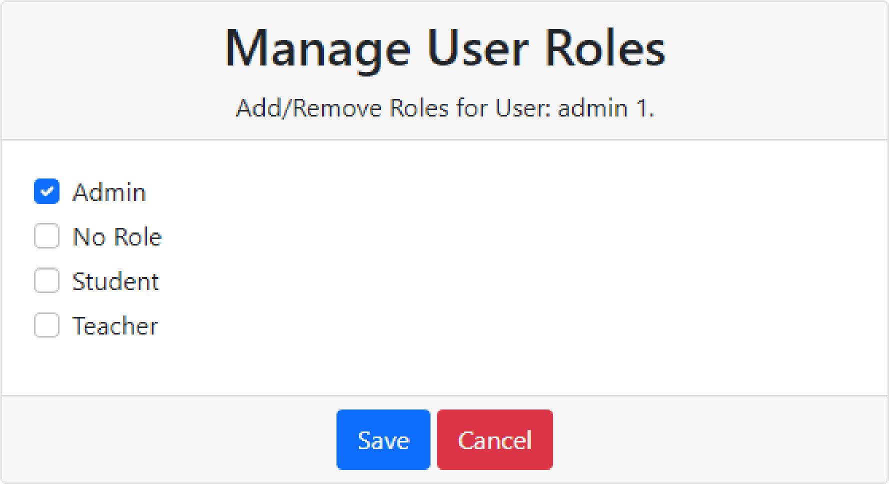


User Role Manager

Assign roles to the registered users.

User Name	Email	Roles	Action
admin 1	admin1@admin.com	Admin	Manage Roles
Student1	student1@gmail.com	Student	Manage Roles
Teacher	teacher@gmail.com	Teacher , Admin	Manage Roles
Teacher2	teacher2@gmail.com	Teacher	Manage Roles
Test1	test1@gmail.com	Student	Manage Roles
Test2	test2@gmail.com	Student	Manage Roles
Test3	test3@gmail.com	No Role	Manage Roles
Test4	test4@gmail.com	No Role	Manage Roles

Figure 5.12: User Role Manager Interface (Administrator)



Manage User Roles

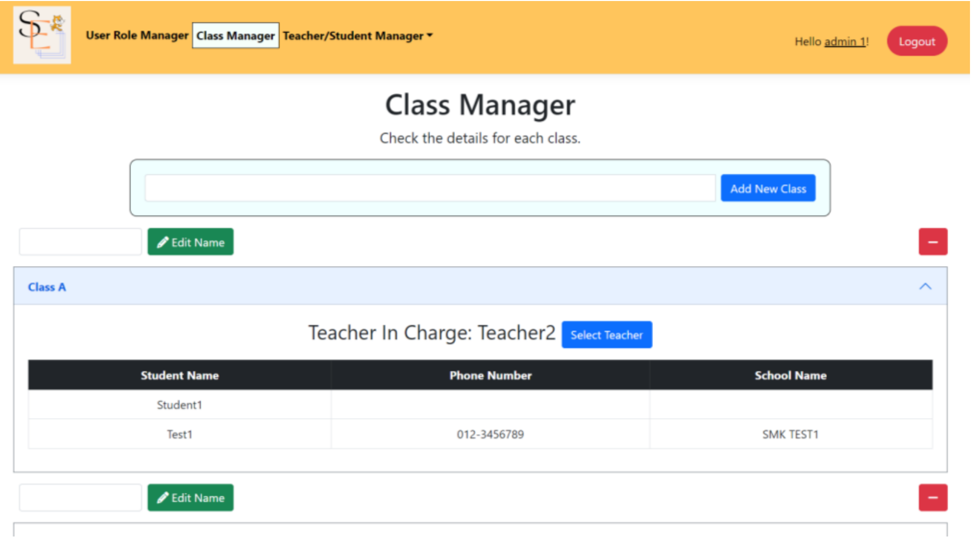
Add/Remove Roles for User: admin 1.

☑ Admin
☐ No Role
☐ Student
☐ Teacher

Figure 5.13: User Role Assignment Page (Administrator)

5.4.4 Class Management Module (Administrator)

A list of classes is displayed in Class Manager, together with the teacher in charge and a list of students (Figure 5.14). Administrators can create a new class by entering the name in the provided space and clicking "Add New Class". As for the existing classes, they can edit the name by entering the name in the provided space and clicking "Edit Name" or delete the class by clicking the delete button on the top right corner of the accordions. To assign or change the teacher in charge, click the "Select Teacher" button and they will be directed to the page shown in Figure 5.15. Only teachers with active status will be listed.



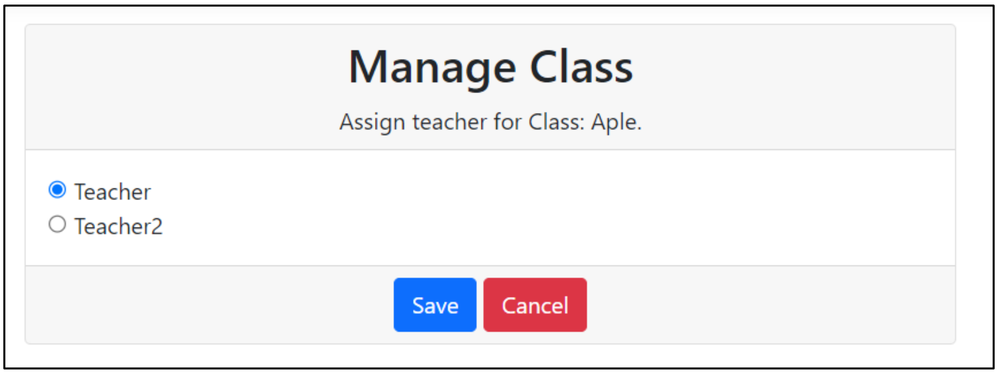
Class Manager

Check the details for each class.

Teacher In Charge: Teacher2

Student Name	Phone Number	School Name
Student1		
Test1	012-3456789	SMK TEST1

Figure 5.14: Class Manager (Administrator)



Manage Class

Assign teacher for Class: Aple.

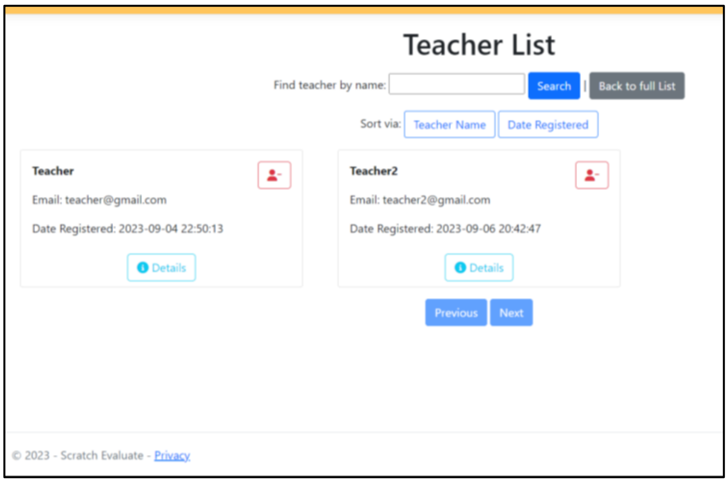
● Teacher
○ Teacher2

Figure 5.15: Class Assignment Page (Administrator)

5.4.5 User Management Module (Administrator)

A list of teachers and students is displayed in the Teacher List and Student List respectively as shown in Figure 5.16. Administrators can search for a user or sort the list based on name and date registered ascending or descending. They can delete a user by clicking on the delete button on the top right of the card, and the information of the users will be removed from the database and their roles will be set back to No Role.

To set an inactive teacher or student back to active status, administrators can click on the "Set Active" button that appears at the bottom of the page to change the user back to active state, and the roles will be automatically assigned back to the user (Figure 5.18).



Teacher List

Find teacher by name:

Sort via:

Teacher

Email: teacher@gmail.com

Date Registered: 2023-09-04 22:50:13

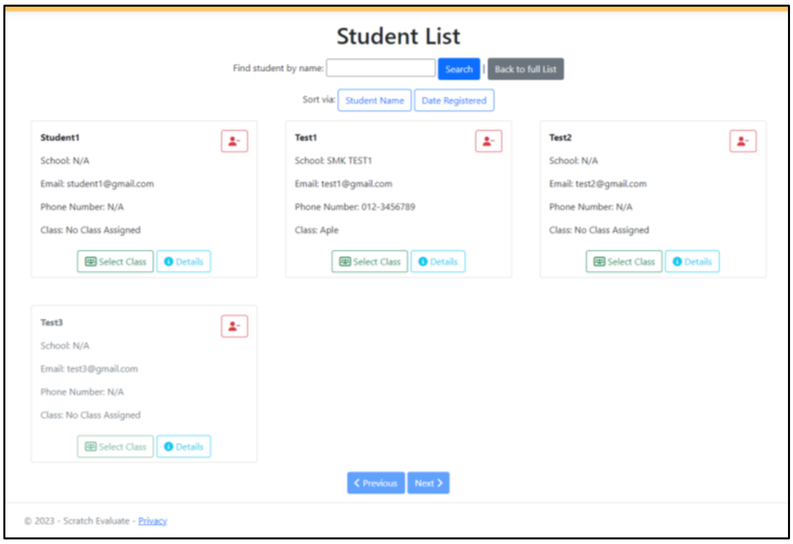
Teacher2

Email: teacher2@gmail.com

Date Registered: 2023-09-06 20:42:47

© 2023 - Scratch Evaluate - Privacy

Figure 5.16: Teacher/Student Manager - Teacher List (Administrator)



Student List

Find student by name:

Sort via:

Student1

School: N/A

Email: student1@gmail.com

Phone Number: N/A

Class: No Class Assigned

Test1

School: SMK TEST1

Email: test1@gmail.com

Phone Number: 012-3456789

Class: Aple

Test2

School: N/A

Email: test2@gmail.com

Phone Number: N/A

Class: No Class Assigned

Test3

School: N/A

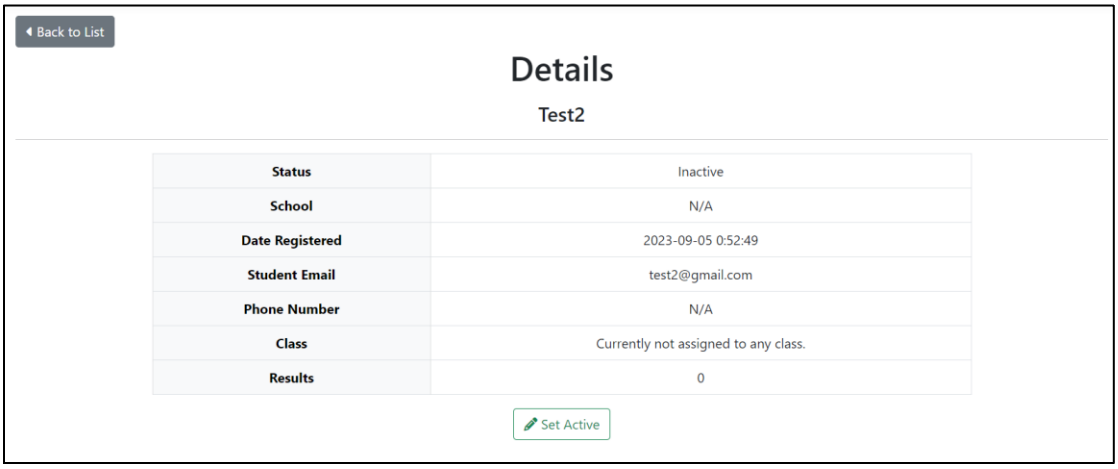
Email: test3@gmail.com

Phone Number: N/A

Class: No Class Assigned

© 2023 - Scratch Evaluate - Privacy

Figure 5.17: Teacher/Student Manager - Student List (Administrator)



◂ Back to List

Details

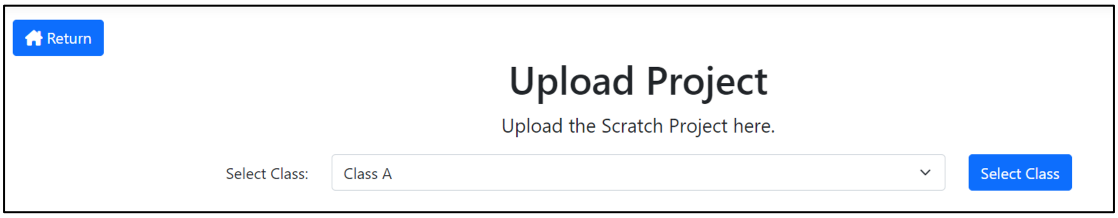
Test2

Status	Inactive
School	N/A
Date Registered	2023-09-05 0:52:49
Student Email	test2@gmail.com
Phone Number	N/A
Class	Currently not assigned to any class.
Results	0

Figure 5.18: Details Page for Inactive Teacher or Student (Administrator)

5.4.6 Project Evaluation Module (Teacher)

Teachers can upload the projects for the students. All fields must be completed before the submission, and only Individual Project and active activities will be populated into the dropdown box for selection. The web application will then confirm the upload by using a pop-up message box as activities created by teachers can only accept one submission from each student. Successful uploads will be notified via a notification as shown in Figure 5.21.

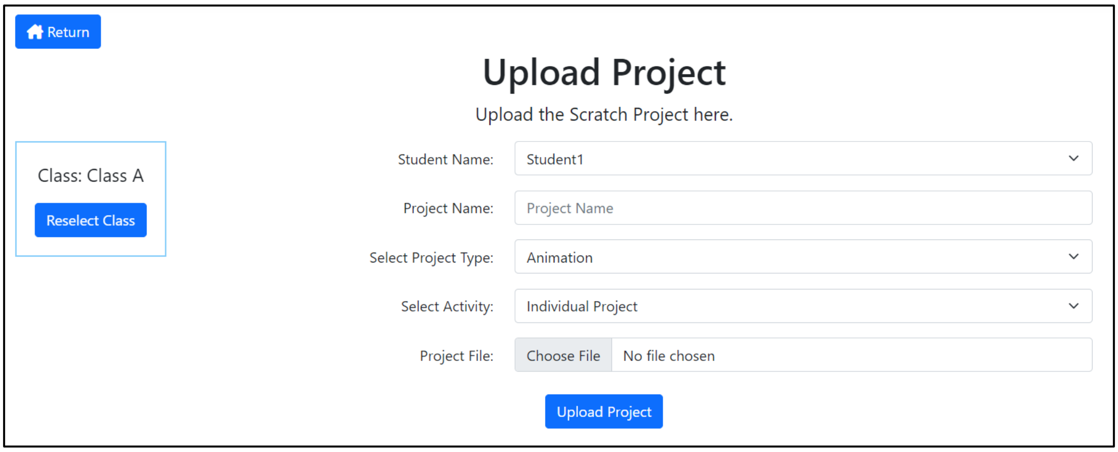


Upload Project

Upload the Scratch Project here.

Select Class: Class A

Figure 5.19: Upload Project - Select Class (Teacher)



Upload Project

Upload the Scratch Project here.

Class: Class A

Student Name: Student1

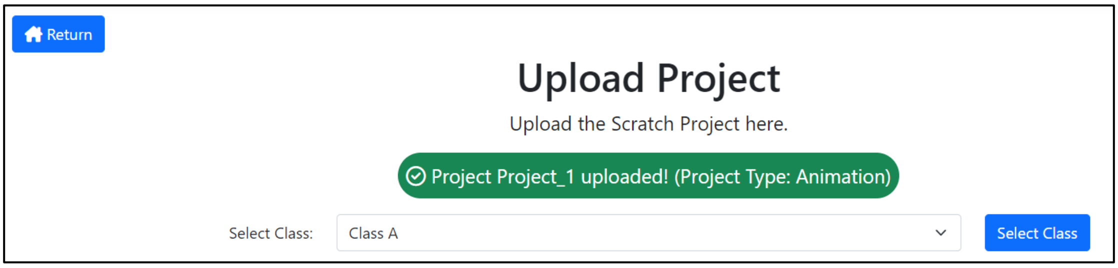
Project Name: Project Name

Select Project Type: Animation

Select Activity: Individual Project

Project File: Choose File No file chosen

Figure 5.20: Upload Project (Teacher)



Upload Project

Upload the Scratch Project here.

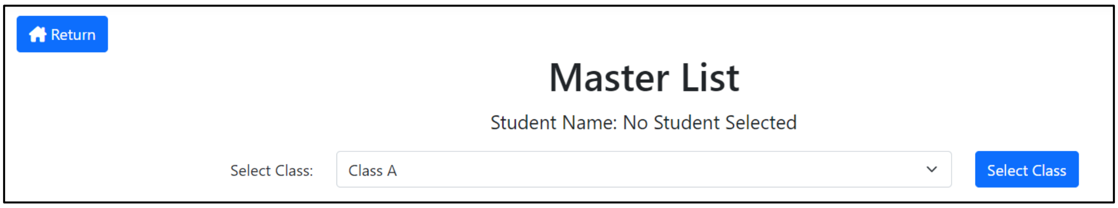
Project Project_1 uploaded! (Project Type: Animation)

Select Class: Class A

Figure 5.21: Successful Upload (Teacher)

5.4.7 Project Result Module and Project Report Module (Teacher)

Teachers can view students' results. They can select a student and check all results related to the selected student. They can view the details of an evaluation result by clicking on the view icon at the rightmost of the table (Figure 5.24). They will be directed to the page shown in Figure 5.25. The page displays basic information about the uploaded project and the score given. Teachers can download the student's results by clicking the "Download Result" button.

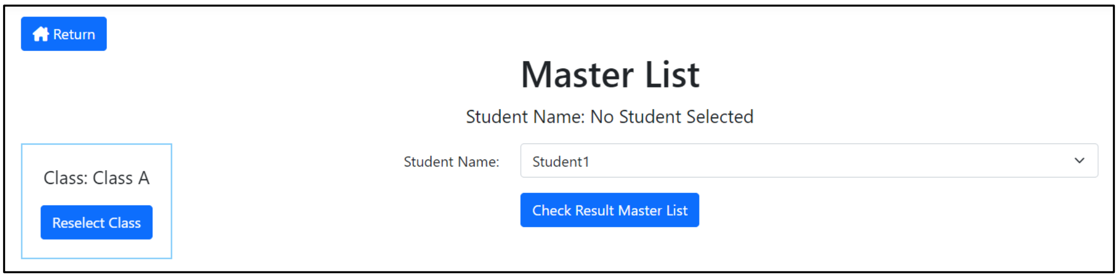


Master List

Student Name: No Student Selected

Select Class: Class A

Figure 5.22: View Student's Result - Select Class (Teacher)



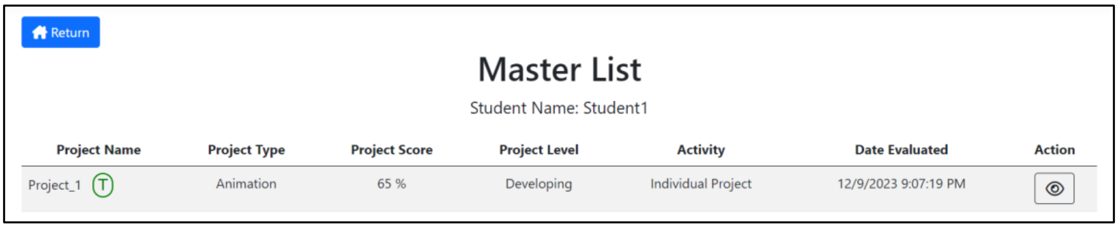
Master List

Student Name: No Student Selected

Class: Class A

Student Name: Student1

Figure 5.23: View Student's Result - Select Student (Teacher)

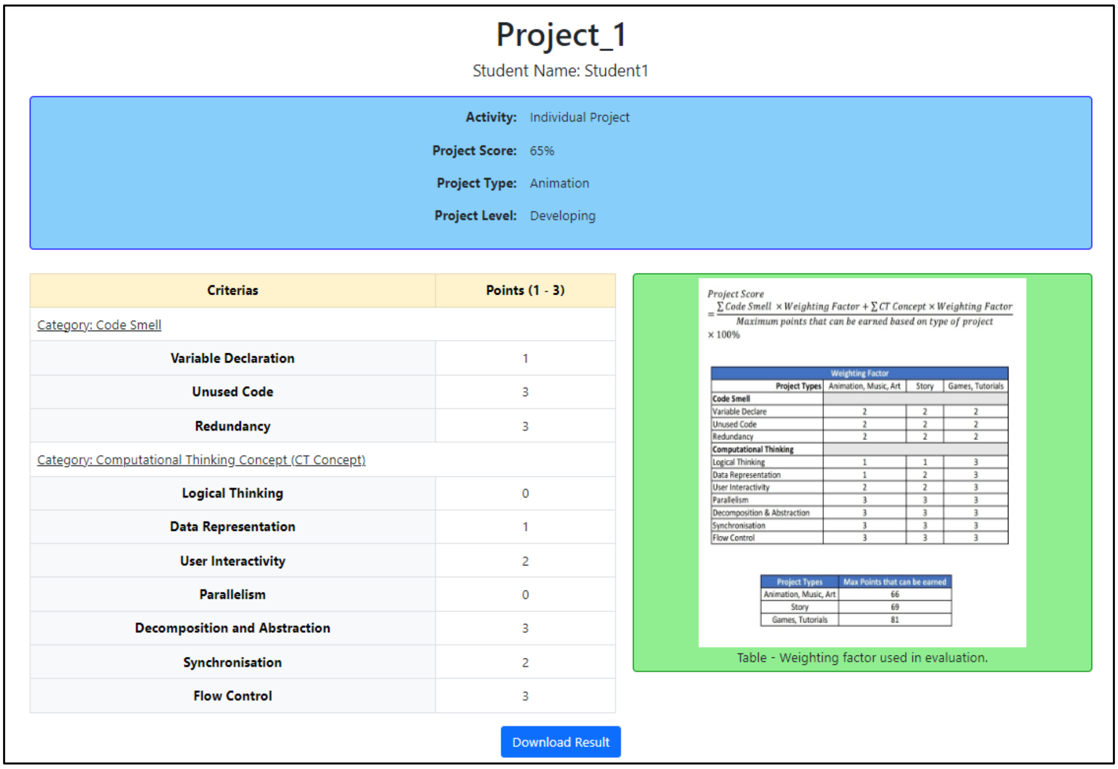


Master List

Student Name: Student1

Project Name	Project Type	Project Score	Project Level	Activity	Date Evaluated	Action
Project_1 Ⓣ	Animation	65 %	Developing	Individual Project	12/9/2023 9:07:19 PM	👁

Figure 5.24: View Student's Result - Master List (Teacher)



Project_1

Student Name: Student1

Activity: Individual Project

Project Score: 65%

Project Type: Animation

Project Level: Developing

Criterias	Points (1 - 3)
Category: Code Smell	
Variable Declaration	1
Unused Code	3
Redundancy	3
Category: Computational Thinking Concept (CT Concept)	
Logical Thinking	0
Data Representation	1
User Interactivity	2
Parallelism	0
Decomposition and Abstraction	3
Synchronisation	2
Flow Control	3

$$\textit{Project Score} = \frac{\sum \textit{Code Smell} \times \textit{Weighting Factor} + \sum \textit{CT Concept} \times \textit{Weighting Factor}}{\textit{Maximum points that can be earned based on type of project}} \times 100\%$$

Weighting Factor			
Project Types	Animation, Music, Art	Story	Games, Tutorials
Code Smell			
Variable Declare	2	2	2
Unused Code	2	2	2
Redundancy	2	2	2
Computational Thinking			
Logical Thinking	1	1	3
Data Representation	1	2	3
User Interactivity	2	2	3
Parallelism	3	3	3
Decomposition & Abstraction	3	3	3
Synchronisation	3	3	3
Flow Control	3	3	3

Project Types	Max Points that can be earned
Animation, Music, Art	66
Story	69
Games, Tutorials	81

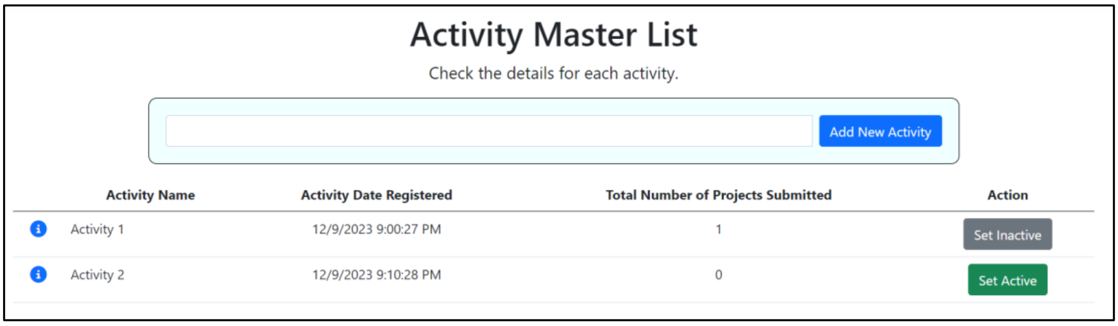
Table - Weighting factor used in evaluation.

Figure 5.25: View Student's Result - Individual Result (Teacher)

5.4.8 Project Activity Module (Teacher)

Teachers can create new activities for students to submit their projects. The newly created activities will be set as active by default and teachers can close the submission by changing the status of activity to inactive as shown in Figure 5.26. The page also gives an overview of the activities created by teachers.

Teachers can click on the information button at the left of the name of the activities to know the details of the activities created. On the Activity Details page (Figure 5.27), they can track the submission status of each class and the average project score and project level of a class. Only one submission is allowed for one activity and resubmission can only be done when teachers delete the previous upload in the activity page by clicking the delete button in the Activity Details page. A download button is provided for each class, meaning that teachers can download a class performance report that prints out the students' name and their project scores for the activity.

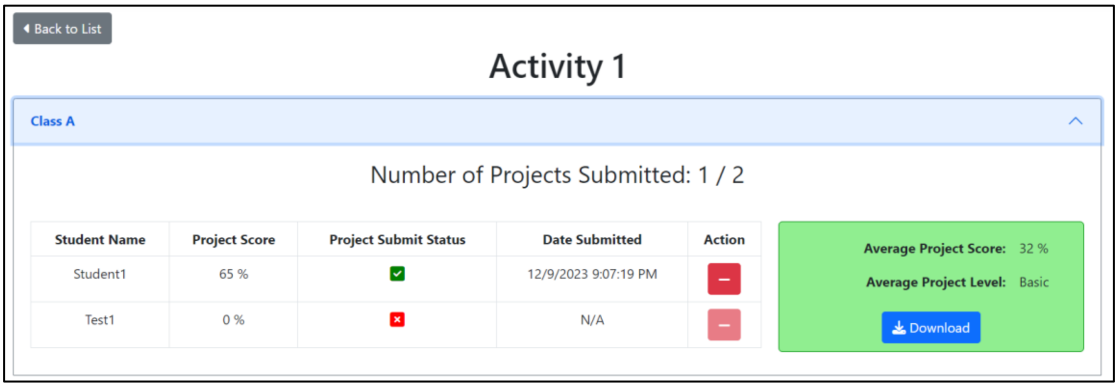


Activity Master List

Check the details for each activity.

	Activity Name	Activity Date Registered	Total Number of Projects Submitted	Action
ℹ	Activity 1	12/9/2023 9:00:27 PM	1	Set Inactive
ℹ	Activity 2	12/9/2023 9:10:28 PM	0	Set Active

Figure 5.26: Manage Activity - Activity Master List (Teacher)



Activity 1

Class A

Number of Projects Submitted: 1 / 2

Student Name	Project Score	Project Submit Status	Date Submitted	Action
Student1	65 %	☑	12/9/2023 9:07:19 PM	−
Test1	0 %	☒	N/A	−

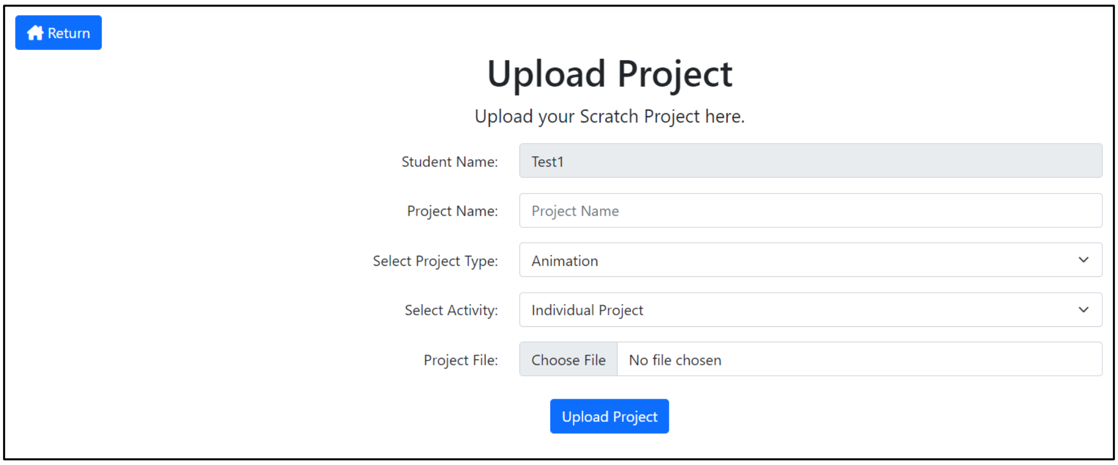
Average Project Score: 32 %

Average Project Level: Basic

Figure 5.27: Manage Activity - Activity Details (Teacher)

5.4.9 Project Evaluation Module (Student)

Students can upload Scratch projects for evaluation. All fields must be completed before the submission, and only Individual Project and active activities will be populated into the dropdown box for selection. The web application will then confirm the upload by using a pop-up message box as activities created by teachers can only accept one submission from each student. Successful uploads will be notified via a notification as shown in Figure 5.29.



Upload Project

Upload your Scratch Project here.

Student Name: Test1

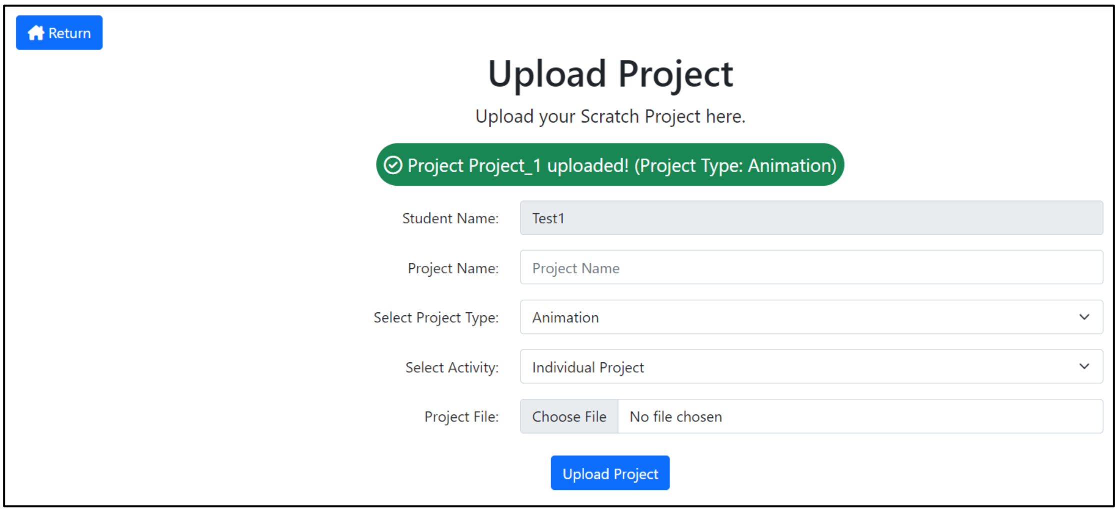
Project Name: Project Name

Select Project Type: Animation

Select Activity: Individual Project

Project File: Choose File No file chosen

Figure 5.28: Upload Project (Student)



Upload Project

Upload your Scratch Project here.

Project Project_1 uploaded! (Project Type: Animation)

Student Name: Test1

Project Name: Project Name

Select Project Type: Animation

Select Activity: Individual Project

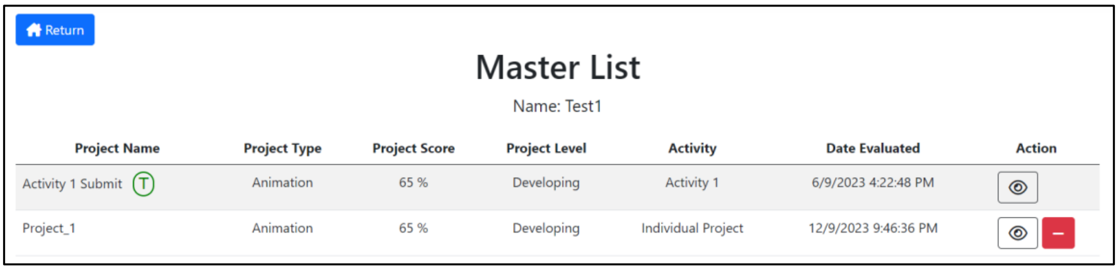
Project File: Choose File No file chosen

Figure 5.29: Successful Upload (Student)

5.4.10 Project Result Module and Project Report Module (Student)

Students can view their results. A list of results will be displayed and they can select one of the results to view the details by clicking on the view icon at the rightmost of the table (Figure 5.30). A green T will be placed beside the name of the projects, indicating that the results are uploaded by their teachers. Besides, they can delete the results if needed, but they can only delete the results under the activity type – Individual Project.

After clicking the view button, they will be directed to the page shown in Figure 5.31. The page displays basic information about the uploaded project and the score given. Students can download their results by clicking the "Download Result" button. For the projects uploaded for the activities created by their teachers, they need to approach the teachers so that their teachers will delete their previous submissions for resubmission purposes.

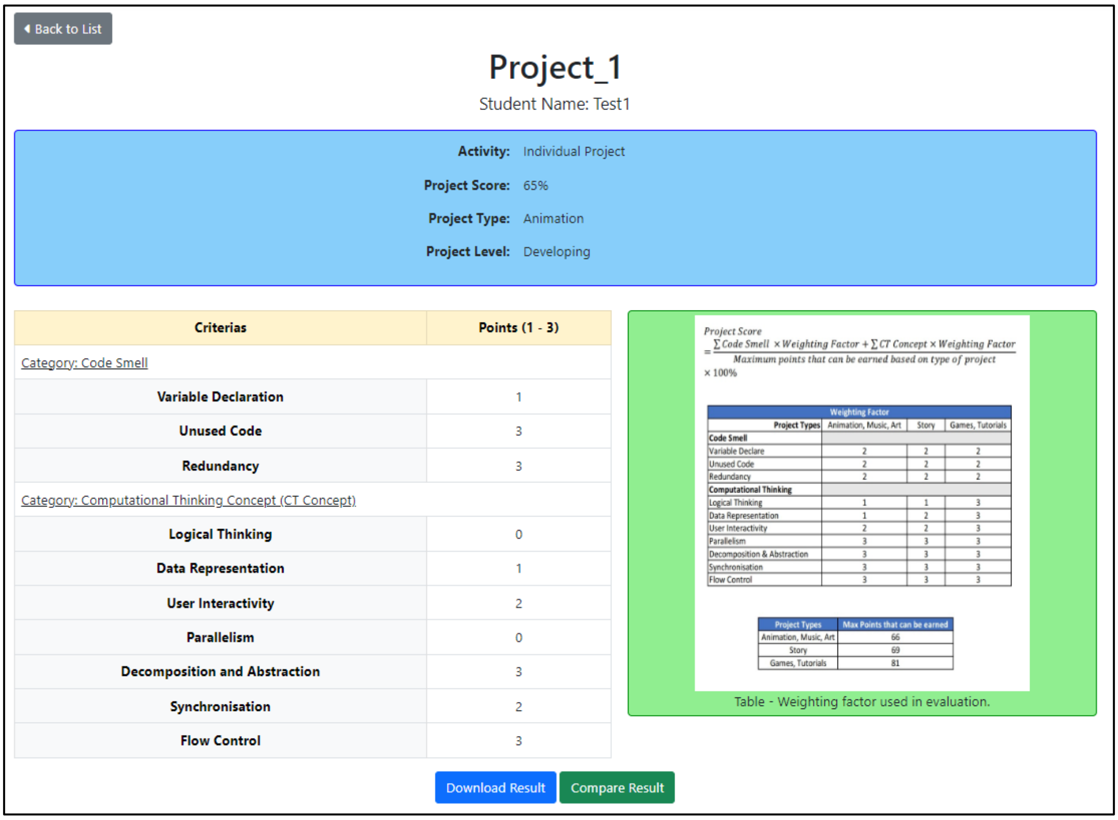


Master List

Name: Test1

Project Name	Project Type	Project Score	Project Level	Activity	Date Evaluated	Action
Activity 1 Submit Ⓣ	Animation	65 %	Developing	Activity 1	6/9/2023 4:22:48 PM	👁
Project_1	Animation	65 %	Developing	Individual Project	12/9/2023 9:46:36 PM	👁 –

Figure 5.30: View Result - Master List (Student)



Project_1

Student Name: Test1

Activity: Individual Project

Project Score: 65%

Project Type: Animation

Project Level: Developing

Criterias	Points (1 - 3)
Category: Code Smell	
Variable Declaration	1
Unused Code	3
Redundancy	3
Category: Computational Thinking Concept (CT Concept)	
Logical Thinking	0
Data Representation	1
User Interactivity	2
Parallelism	0
Decomposition and Abstraction	3
Synchronisation	2
Flow Control	3

$$\textit{Project Score} = \frac{\sum \textit{Code Smell} \times \textit{Weighting Factor} + \sum \textit{CT Concept} \times \textit{Weighting Factor}}{\textit{Maximum points that can be earned based on type of project}} \times 100\%$$

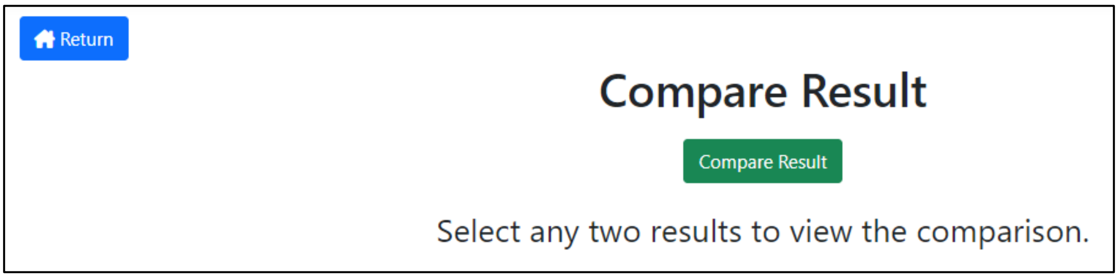
Weighting Factor			
Project Types	Animation, Music, Art	Story	Games, Tutorials
Code Smell			
Variable Declare	2	2	2
Unused Code	2	2	2
Redundancy	2	2	2
Computational Thinking			
Logical Thinking	1	1	3
Data Representation	1	2	3
User Interactivity	2	2	3
Parallelism	3	3	3
Decomposition & Abstraction	3	3	3
Synchronisation	3	3	3
Flow Control	3	3	3

Project Types	Max Points that can be earned
Animation, Music, Art	66
Story	69
Games, Tutorials	81

Table - Weighting factor used in evaluation.

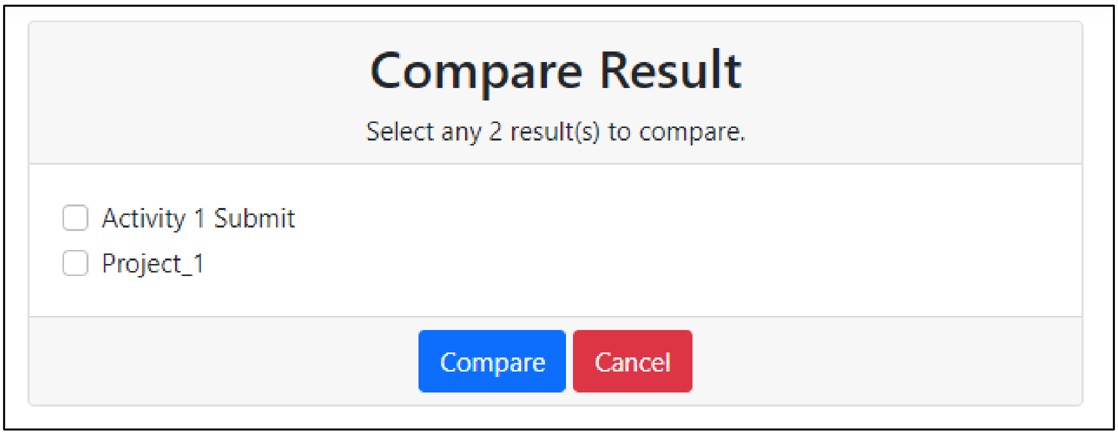
Figure 5.31: View Result - Individual Result (Student)

Students can compare between two results by clicking on the “Compare Result” button. They will be directed to a page shown in Figure 5.32. Students must select two results to compare. The comparison table is shown in Figure 5.34.



Select any two results to view the comparison.

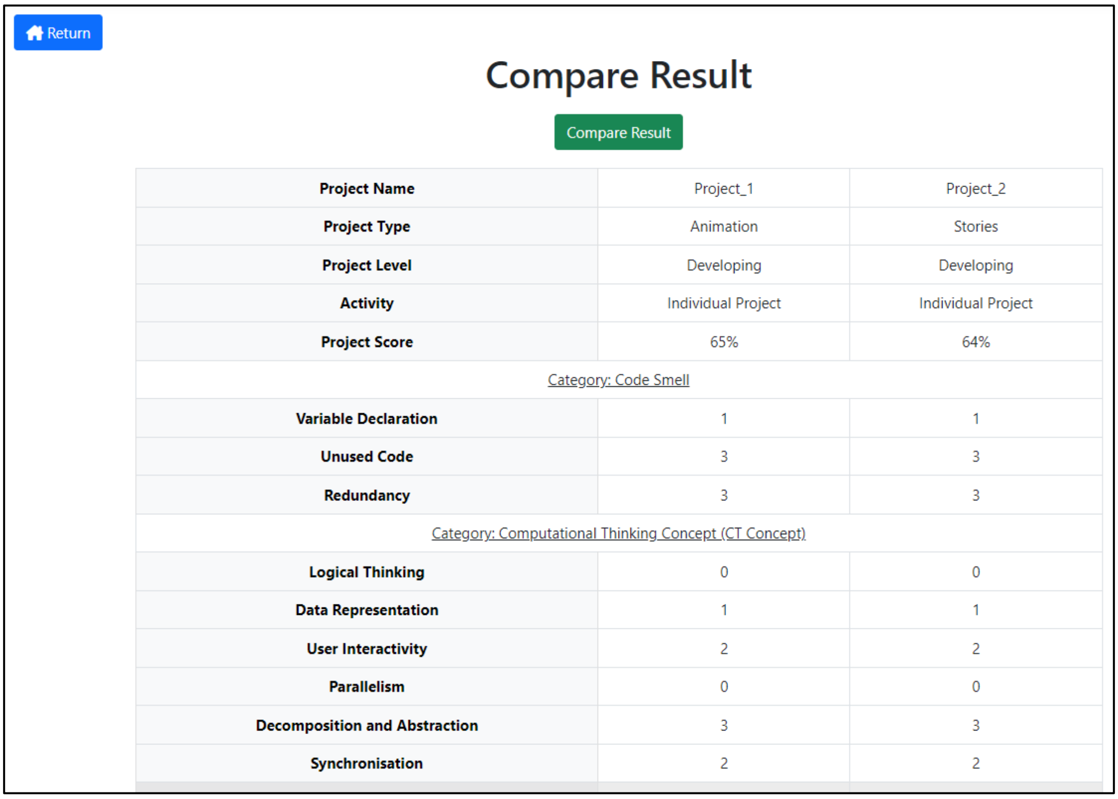
Figure 5.32: Compare Result (Student)



Select any 2 result(s) to compare.

□ Activity 1 Submit
□ Project_1

Figure 5.33: Compare Result - Select Two Results (Student)



Compare Result

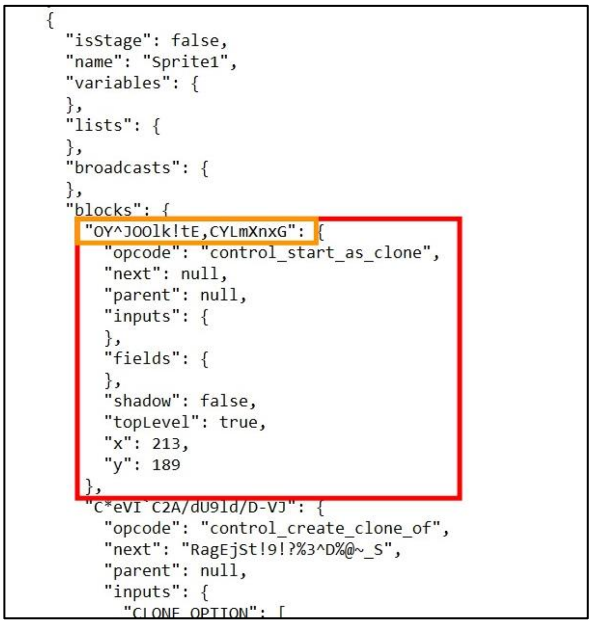
Project Name	Project_1	Project_2
Project Type	Animation	Stories
Project Level	Developing	Developing
Activity	Individual Project	Individual Project
Project Score	65%	64%
Category: Code Smell		
Variable Declaration	1	1
Unused Code	3	3
Redundancy	3	3
Category: Computational Thinking Concept (CT Concept)		
Logical Thinking	0	0
Data Representation	1	1
User Interactivity	2	2
Parallelism	0	0
Decomposition and Abstraction	3	3
Synchronisation	2	2

Figure 5.34: Compare Result - After Select Two Results (Student)

5.5 Implementation Issues and Challenges

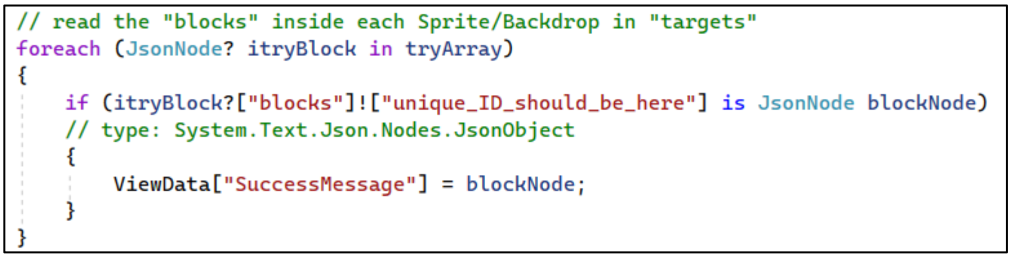
5.5.1 Problem in reading information resides in the “project.json” file.

To evaluate the uploaded Scratch projects, the web application needs to access the JSON file containing all information about the projects, such as the type of blocks used, the sequence of the blocks used and the backdrop used. However, there is a major issue to access the specific value as the file contains information that is dynamically generated upon extraction from Scratch. As shown in Figure 5.35, the string inside the orange box is the unique ID of a block. It is a string randomly generated by Scratch after users save their projects on their computers. One of the methods to read JSON content without the class to deserialize it is by converting it into JSON DOM and extracting the specific value nested inside the file. Even though this method can access values deep inside a JSON file with square bracket notation, it is not applicable as it still requires the web application to know the unique ID of the block. Figure 5.36 shows an example of the code snippet to access the JSON content with JSON DOM via the unique ID.



```
{
  "isStage": false,
  "name": "Sprite1",
  "variables": {
  },
  "lists": {
  },
  "broadcasts": {
  },
  "blocks": {
    "OY^JOOlk!tE,CYLmXnxG": {
      "opcode": "control_start_as_clone",
      "next": null,
      "parent": null,
      "inputs": {
      },
      "fields": {
      },
      "shadow": false,
      "topLevel": true,
      "x": 213,
      "y": 189
    },
    "C*eVI`C2A/dU9ld/D-VJ": {
      "opcode": "control_create_clone_of",
      "next": "RagEjSt!9!?%3^D%@~_S",
      "parent": null,
      "inputs": {
        "CLONE_OPTION": [
```

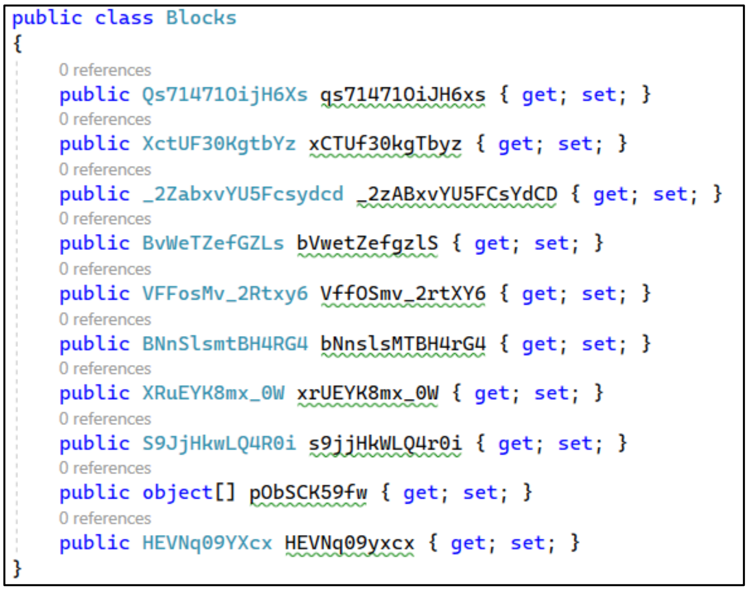
Figure 5.35: Snippet of the Content of “project.json”



```
// read the "blocks" inside each Sprite/Backdrop in "targets"
foreach (JsonNode? itryBlock in tryArray)
{
    if (itryBlock?["blocks"]!["unique_ID_should_be_here"] is JsonNode blockNode)
    // type: System.Text.Json.Nodes.JsonObject
    {
        ViewData["SuccessMessage"] = blockNode;
    }
}
```

Figure 5.36: Code Snippet of Accessing JSON File with JSON DOM

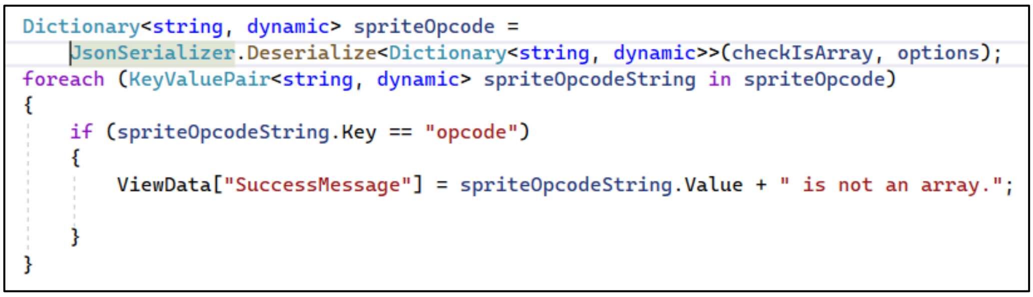
There is a second method, which is deserializing the JSON file as .NET objects. Nonetheless, this method requires a class to deserialize the JSON into. This method is initially not applicable as the class contains dynamic data as shown in Figure 5.37, a code snippet of a class generated by using “Paste Special → Paste JSON As Classes”.



```
public class Blocks
{
    0 references
    public Qs71471OijH6Xs qs71471OiJH6xs { get; set; }
    0 references
    public XctUF30KgtbYz xCTUf30kgTbyz { get; set; }
    0 references
    public _2ZabxvYU5Fcsydcd _2zABxvYU5FCsYdCD { get; set; }
    0 references
    public BvWeTZefGZLs bVwetZefgzlS { get; set; }
    0 references
    public VFFosMv_2Rtxy6 VffOSmv_2rtXY6 { get; set; }
    0 references
    public BNnSlsmtBH4RG4 bNnslsMTBH4rG4 { get; set; }
    0 references
    public XRuEYK8mx_0W xrUEYK8mx_0W { get; set; }
    0 references
    public S9JjHkwLQ4R0i s9jjHkWLQ4r0i { get; set; }
    0 references
    public object[] pObSCK59fw { get; set; }
    0 references
    public HEVNq09YXcx HEVNq09yxcx { get; set; }
}
```

Figure 5.37: Code Snippet of Blocks Class

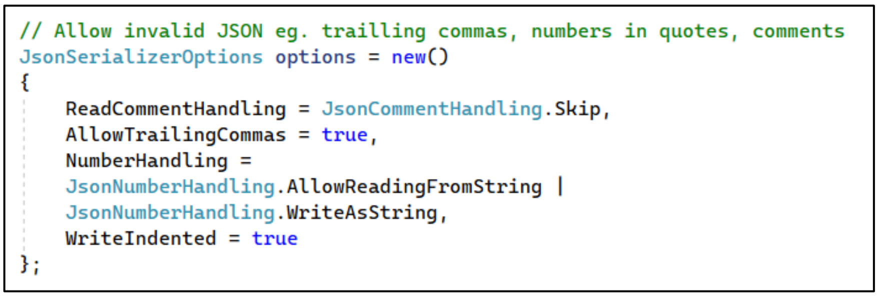
After conducting thorough research, the problem is finally solved by combining the usage of the Dictionary with the second method mentioned previously. The Dictionary is a generic collection, generally used to store key/value pairs. Unlike the second method, the introduction of Dictionary enables the web application to split the keys and values in the JSON content without the full acknowledgement of the format in the form of a class. Figure 5.38 shows an example of the usage of the combination that gives a successful result. As some of the values contain numbers in quotes, which is considered as invalid JSON format by default, options is defined to enable the web application to read the number from a string (Figure 5.39).



```
Dictionary<string, dynamic> spriteOpcode =
    JsonSerializer.Deserialize<Dictionary<string, dynamic>>(checkIsArray, options);
foreach (KeyValuePair<string, dynamic> spriteOpcodeString in spriteOpcode)
{
    if (spriteOpcodeString.Key == "opcode")
    {
        ViewData["SuccessMessage"] = spriteOpcodeString.Value + " is not an array.";

    }
}
```

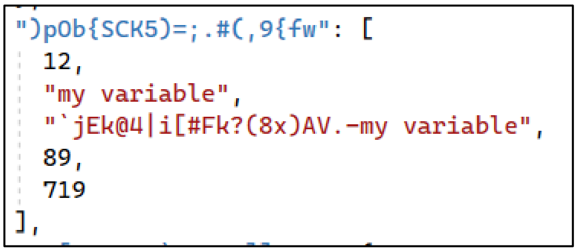
Figure 5.38: Code Snippet of Deserialize Function and Dictionary



```
// Allow invalid JSON eg. trailling commas, numbers in quotes, comments
JsonSerializerOptions options = new()
{
    ReadCommentHandling = JsonCommentHandling.Skip,
    AllowTrailingCommas = true,
    NumberHandling =
    JsonNumberHandling.AllowReadingFromString |
    JsonNumberHandling.WriteAsString,
    WriteIndented = true
};
```

Figure 5.39: Code Snippet of options

Lastly, there is a concern in which the blocks are represented in array form as shown in Figure 5.40. This type of block will cause the web application to throw JsonException as it will be converted as a JsonArray but not a JsonObject, in which the latter enables the usage of square bracket notation to access key/value pairs nested inside a key. This problem is overcome by making the web application run in two different ways, depending on the type of block (JsonArray and JsonObject).

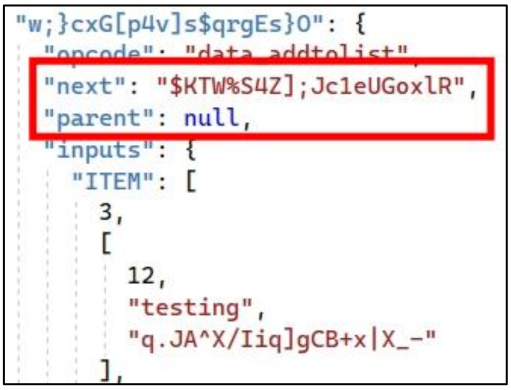


```
")pOb{SCK5)=;.#(,9{fw": [
  12,
  "my variable",
  "`jEk@4|i[#Fk?(8x)AV.-my variable",
  89,
  719
],
```

Figure 5.40: Code Snippet of Block Information in Array Form

5.5.2 Problem in arranging the blocks in the correct sequence.

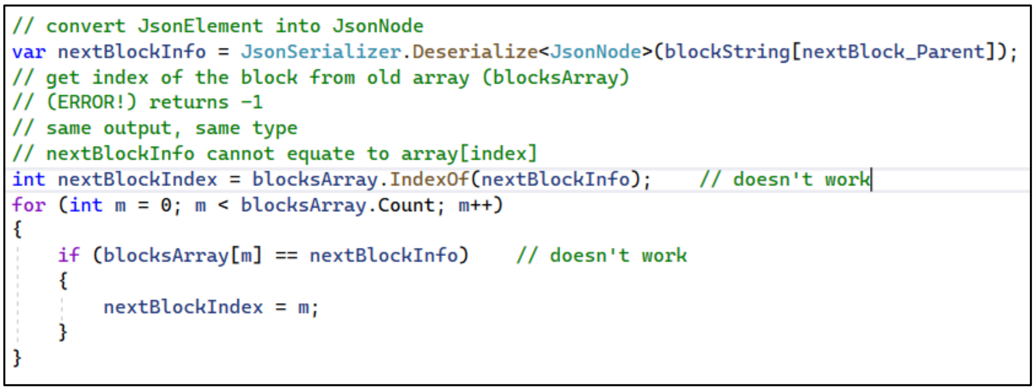
Since the sequence of the blocks in the "project.json" file does not guarantee the sequence of the blocks that the users have placed in the workspace, the web application has to rearrange the blocks into the correct sequence. Initially, JsonArray is used to store the information of the blocks, with one element representing one block. Two JsonArrays, named blocksArray and blocksArrayNew, are created to store unsorted elements and sorted elements respectively. Through research, it is known that the sequence of the blocks is related to the keys named "parent" and "next" (Figure 5.41). The "next" key represents the unique ID of the block immediately after the current block, so the web application has to get the information of the subsequent blocks via the unique ID to store the information of the blocks in an arranged manner.



```
"w;}cxG[p4v]s$qrgEs}O": {
  "opcode": "data_addtolist",
  "next": "$KTW%S4Z];JcleUGoxlR",
  "parent": null,
  "inputs": {
    "ITEM": [
      3,
      [
        12,
        "testing",
        "q.JA^X/Iiq]gCB+x|X_-"
      ],
```

Figure 5.41: Code Snippet of project.json (next and parent)

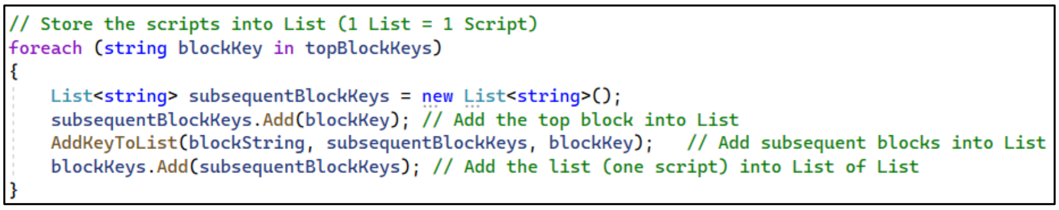
As the JsonArrays declared (blocksArray and blocksArrayNew) do not store the unique ID together with the information of the blocks, there is a need to obtain the information of the next block by comparing the value of the elements in blocksArray and retrieving the index of the specific element. However, this method failed as JsonArray.IndexOf does not retrieve the index by comparing the value of the elements, but the reference of the elements. As shown in Figure 5.42, two attempts of trying to get the index of the specific element to read the information stored inside the element do not work.



```
// convert JsonElement into JsonNode
var nextBlockInfo = JsonSerializer.Deserialize<JsonNode>(blockString[nextBlock_Parent]);
// get index of the block from old array (blocksArray)
// (ERROR!) returns -1
// same output, same type
// nextBlockInfo cannot equate to array[index]
int nextBlockIndex = blocksArray.IndexOf(nextBlockInfo);    // doesn't work
for (int m = 0; m < blocksArray.Count; m++)
{
    if (blocksArray[m] == nextBlockInfo)    // doesn't work
    {
        nextBlockIndex = m;
    }
}
```

Figure 5.42: Code Snippet of Failed Attempt (JsonArray)

The problem is solved by storing the keys, which are the unique ID of the blocks in a list. As shown in Figure 5.43, the sequence of the blocks is stored in the form of a list of lists, with one list representing one script in an arranged manner. With this method, the web application can directly access the information of a block by using the key. This method also reduces the complexity of the code and increases the reusability of the variables related to the blocks.



```
// Store the scripts into List (1 List = 1 Script)
foreach (string blockKey in topBlockKeys)
{
    List<string> subsequentBlockKeys = new List<string>();
    subsequentBlockKeys.Add(blockKey); // Add the top block into List
    AddKeyToList(blockString, subsequentBlockKeys, blockKey);   // Add subsequent blocks into List
    blockKeys.Add(subsequentBlockKeys); // Add the list (one script) into List of List
}
```

Figure 5.43: Code Snippet of Success Attempt (List of List)

5.6 Concluding Remark

The hardware and software setup to develop the software is discussed in this chapter, including the setting up of the required workloads and libraries to develop the web application. The system is completed with the functionalities mentioned in this chapter, with screenshots of the interfaces attached. Some implementation issues and challenges encountered and resolved during the web application development are also included in this chapter.

CHAPTER 6

System Evaluation and Discussion

6.1 Environment for System Testing

6.1.1 Hardware Setup

Table 6.1 shows the specifications of the laptop used to test the web application.

Description	Specifications
Model	HP 15S-DU3024TX Notebook
Processor	Intel i5-1135G7
Operating System	Windows 10 Home Single Language
Graphic	NVIDIA GeForce MX350
Memory	16GB RAM
Storage	512GB SSD

Table 6.1: Specifications of Laptop

6.1.2 Software Setup

No software is required to set up the test environment as the product of the project is a web application which is accessible in a browser.

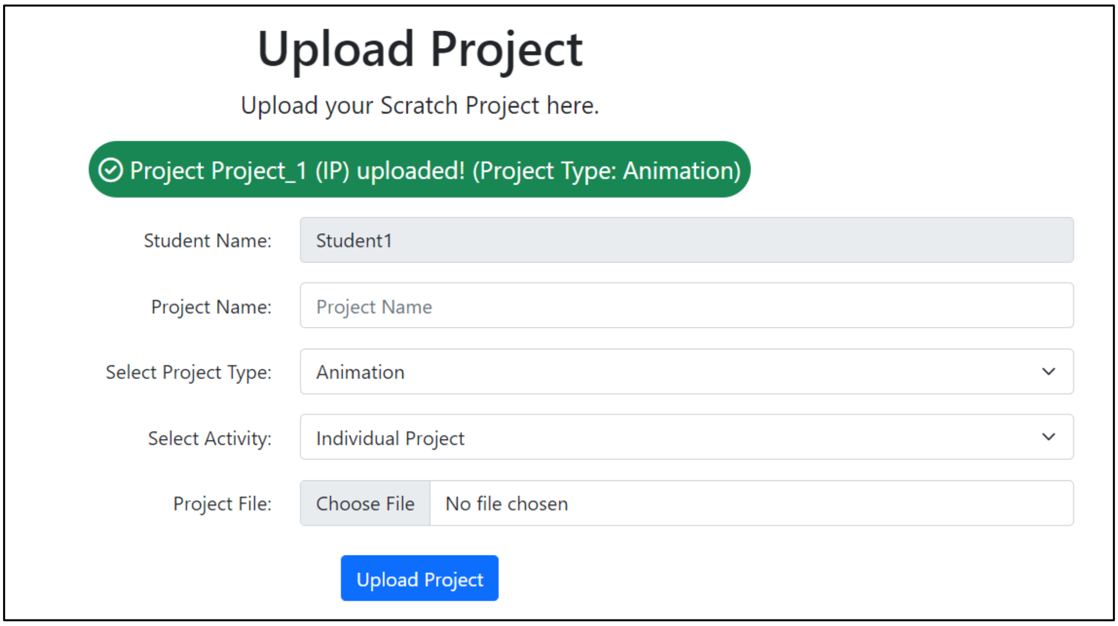
6.2 Test Cases and Result

The main functionalities of the web application are tested by using Use Case Testing (Table 6.2 – Table 6.10). As for the function that evaluates the uploaded Scratch projects, six projects are randomly selected from the Scratch website, with each of them representing different types of projects (Animation, Art, Games, Music, Tutorials and Stories) (Table 6.11 – Table 6.16). To ensure the weighting factor is implemented as expected, one project has been selected from the six and evaluated with another type of project (Table 6.17).

6.2.1 Use Case Testing

Test Case ID	TC_001
Test Item	UC002
Module Name	Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Upload Project) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC002.
Pre-Requisite (Special Conditions or Limitations)	1. Student is assigned to a class. 2. One activity created by teacher.
Test Steps	1. Main Flow: Follows the steps in UC002. 2. Alternate Flow – No Results Found: Follows the steps in UC002. 3. Alternate Flow – Only One or No Result Chosen: Follows the steps in UC002. 4. Alternate Flow – More Than One Result Chosen: Follows the steps in UC002.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Empty Project Name: Working as expected. 3. Alternate Flow – Duplicated Submission for Activities Created by Teachers: Working as expected. 4. Alternate Flow – Incorrect File Uploaded: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Empty Project Name: Working as expected. 3. Alternate Flow – Duplicated Submission for Activities Created by Teachers: Working as expected. 4. Alternate Flow – Incorrect File Uploaded: Working as expected.
Test Result	PASS

Table 6.2: Test Case for UC002



Upload Project

Upload your Scratch Project here.

Project Project_1 (IP) uploaded! (Project Type: Animation)

Student Name: Student1

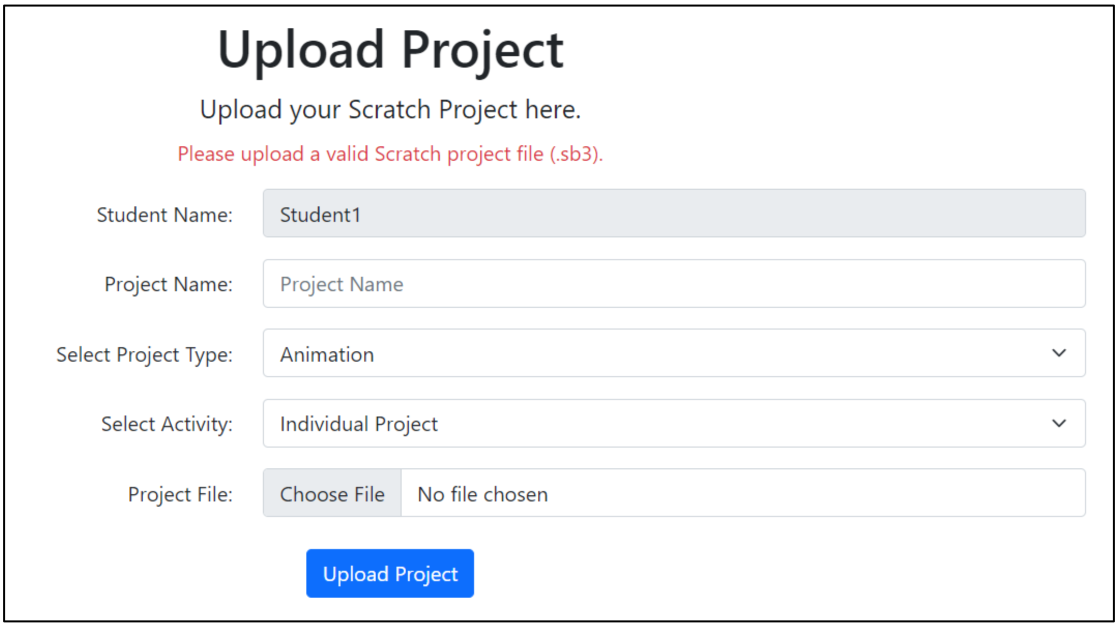
Project Name: Project Name

Select Project Type: Animation

Select Activity: Individual Project

Project File: No file chosen

Figure 6.1: Successful Upload



Upload Project

Upload your Scratch Project here.

Please upload a valid Scratch project file (.sb3).

Student Name: Student1

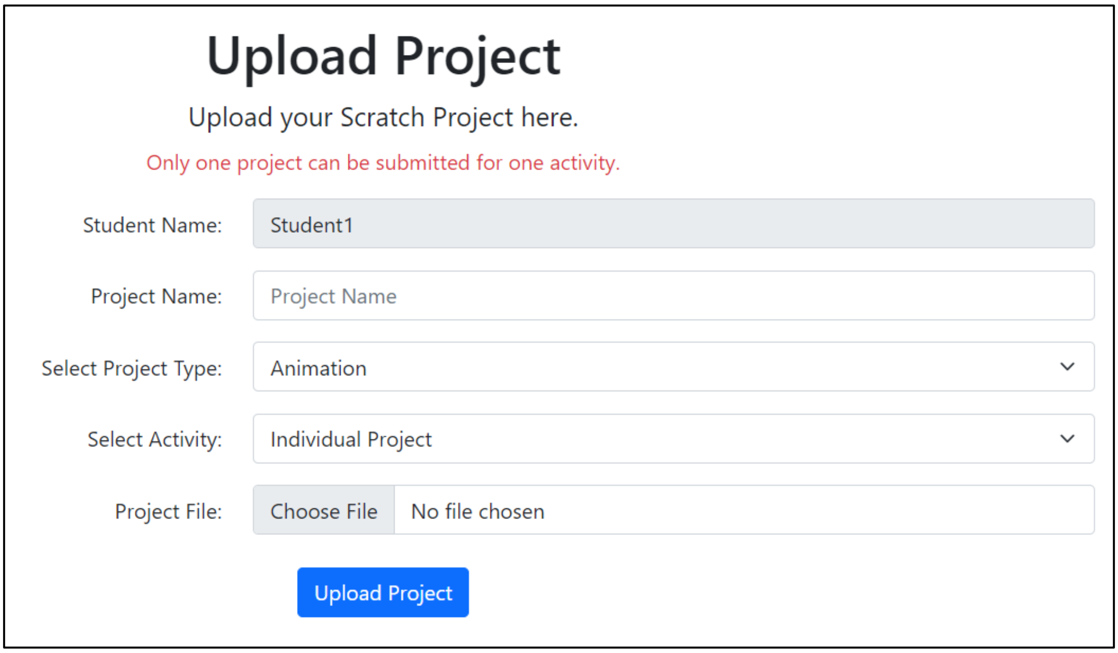
Project Name: Project Name

Select Project Type: Animation

Select Activity: Individual Project

Project File: No file chosen

Figure 6.2: Error Message for Alternate Flow – Incorrect File Uploaded



Upload Project

Upload your Scratch Project here.

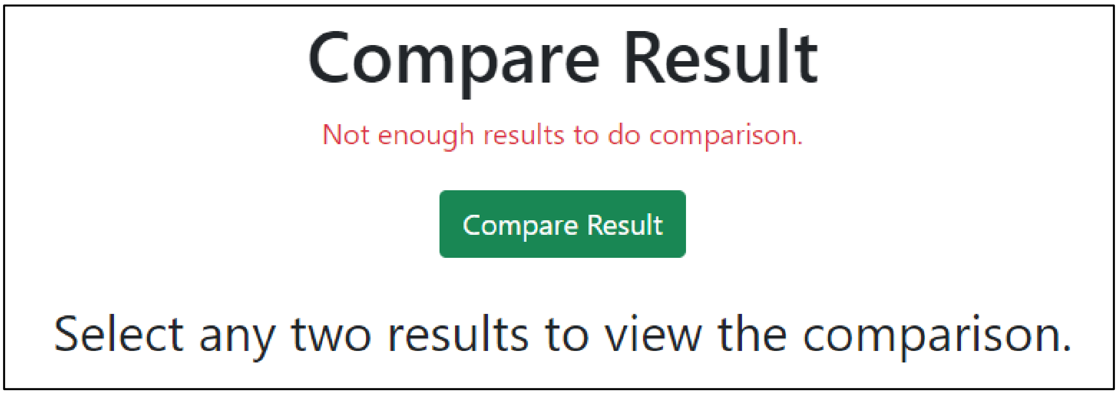
Only one project can be submitted for one activity.

Student Name:	Student1
Project Name:	Project Name
Select Project Type:	Animation
Select Activity:	Individual Project
Project File:	Choose File No file chosen

Figure 6.3: Error Message for Alternate Flow – Duplicated Submission for Activities Created by Teacher

Test Case ID	TC_002
Test Item	UC001
Module Name	Project Result Module (Student)
Reference Document	ViewModel: ProjectResultMasterListViewModel.cs Pages: CompareResultPage.cshtml, CompareResultPage.cshtml.cs, CompareResultMasterList.cshtml, CompareResultMasterList.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Compare Result) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC001.
Pre-Requisite (Special Conditions or Limitations)	Student is assigned to a class.
Test Steps	1. Main Flow: Follows the steps in UC001. 2. Alternate Flow – No Results Found: Follows the steps in UC001. 3. Alternate Flow – Only One or No Result Chosen: Follows the steps in UC001. 4. Alternate Flow – More Than One Result Chosen: Follows the steps in UC001.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – No Results Found: Working as expected. 3. Alternate Flow – Only One or No Result Chosen: Working as expected. 4. Alternate Flow – More Than One Result Chosen: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – No Results Found: Working as expected. 3. Alternate Flow – Only One or No Result Chosen: Working as expected. 4. Alternate Flow – More Than One Result Chosen: Working as expected.
Test Result	PASS

Table 6.3: Test Case for UC001

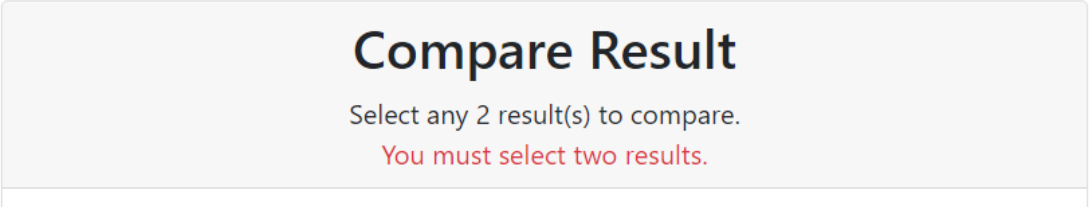


Compare Result

Not enough results to do comparison.

Select any two results to view the comparison.

Figure 6.4: Error Message for Alternate Flow – No Results Found

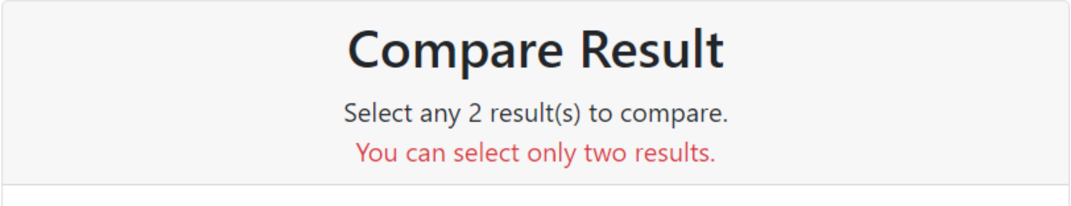


Compare Result

Select any 2 result(s) to compare.

You must select two results.

Figure 6.5: Error Message for Alternate Flow – Only One or No Result Chosen



Compare Result

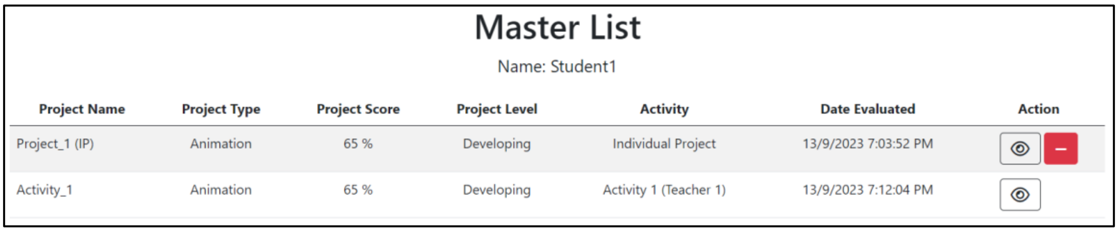
Select any 2 result(s) to compare.

You can select only two results.

Figure 6.6: Error Message for Alternate Flow – More Than One Result Chosen

Test Case ID	TC_003
Test Item	UC003
Module Name	Project Result Module (Student), Project Report Module (Student)
Reference Document	ViewModel: ProjectResultMasterListViewModel.cs Pages: ProjectResultMasterList.cshtml, ProjectResultMasterList.cshtml.cs, ProjectResultPage.cshtml, ProjectResultPage.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (View Result) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC003.
Pre-Requisite (Special Conditions or Limitations)	Student is assigned to a class.
Test Steps	1. Main Flow: Follows the steps in UC003. 2. Alternate Flow – Download Result: Follows the steps in UC003. 3. Alternate Flow – No Results Found: Follows the steps in UC003. 4. Alternate Flow – Delete Result: Follows the steps in UC003.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Download Result: Working as expected. 3. Alternate Flow – No Results Found: Working as expected. 4. Alternate Flow – Delete Result: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Download Result: Working as expected. 3. Alternate Flow – No Results Found: Working as expected. 4. Alternate Flow – Delete Result: Working as expected.
Test Result	PASS

Table 6.4: Test Case for UC003

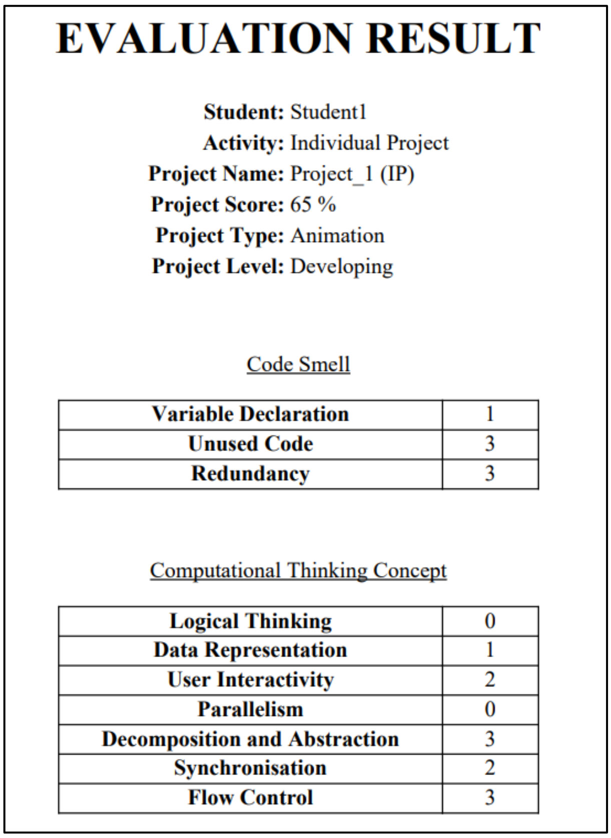


Master List

Name: Student1

Project Name	Project Type	Project Score	Project Level	Activity	Date Evaluated	Action
Project_1 (IP)	Animation	65 %	Developing	Individual Project	13/9/2023 7:03:52 PM	👁 −
Activity_1	Animation	65 %	Developing	Activity 1 (Teacher 1)	13/9/2023 7:12:04 PM	👁

Figure 6.7: Project Result Master List Page (Student)



EVALUATION RESULT

Student: Student1
Activity: Individual Project
Project Name: Project_1 (IP)
Project Score: 65 %
Project Type: Animation
Project Level: Developing

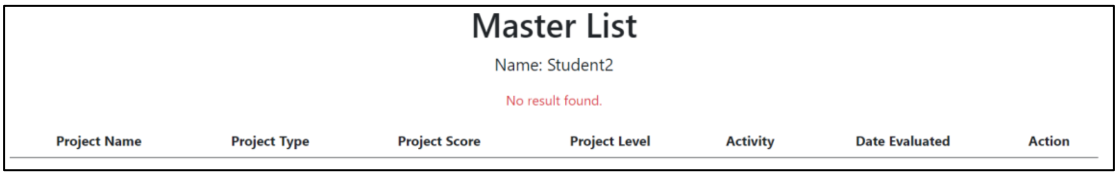
Code Smell

Variable Declaration	1
Unused Code	3
Redundancy	3

Computational Thinking Concept

Logical Thinking	0
Data Representation	1
User Interactivity	2
Parallelism	0
Decomposition and Abstraction	3
Synchronisation	2
Flow Control	3

Figure 6.8: Downloaded Report (Student)



Master List

Name: Student2

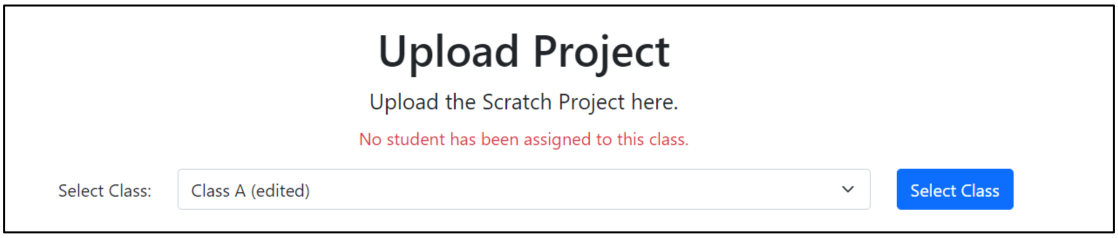
No result found.

Project Name	Project Type	Project Score	Project Level	Activity	Date Evaluated	Action

Figure 6.9: Error Message for Alternate Flow – No Results Found

Test Case ID	TC_004
Test Item	UC004
Module Name	Project Evaluation Module (Teacher)
Reference Document	Pages: UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Upload Project) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC004.
Pre-Requisite (Special Conditions or Limitations)	Teacher is assigned at least one class.
Test Steps	1. Main Flow: Follows the steps in UC004. 2. Alternate Flow – Class Without Students: Follows the steps in UC004. 3. Alternate Flow – Reselect Class: Follows the steps in UC004. 4. Alternate Flow – Empty Project Name: Follows the steps in UC004. 5. Alternate Flow – Duplicated Submission for Activities Created by Teachers: Follows the steps in UC004. 6. Alternate Flow – Incorrect File Uploaded: Follows the steps in UC004.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Class Without Students: Working as expected. 3. Alternate Flow – Reselect Class: Working as expected. 4. Alternate Flow – Empty Project Name: Working as expected. 5. Alternate Flow – Duplicated Submission for Activities Created by Teachers: Working as expected. 6. Alternate Flow – Incorrect File Uploaded: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Class Without Students: Working as expected. 3. Alternate Flow – Reselect Class: Working as expected. 4. Alternate Flow – Empty Project Name: Working as expected. 5. Alternate Flow – Duplicated Submission for Activities Created by Teachers: Working as expected. 6. Alternate Flow – Incorrect File Uploaded: Working as expected.
Test Result	PASS

Table 6.5:Test Case for UC004



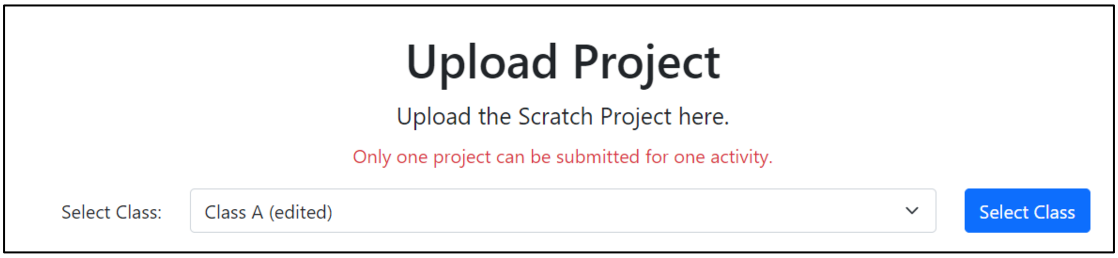
Upload Project

Upload the Scratch Project here.

No student has been assigned to this class.

Select Class: Class A (edited)

Figure 6.10: Error Message for Alternate Flow – Class Without Students



Upload Project

Upload the Scratch Project here.

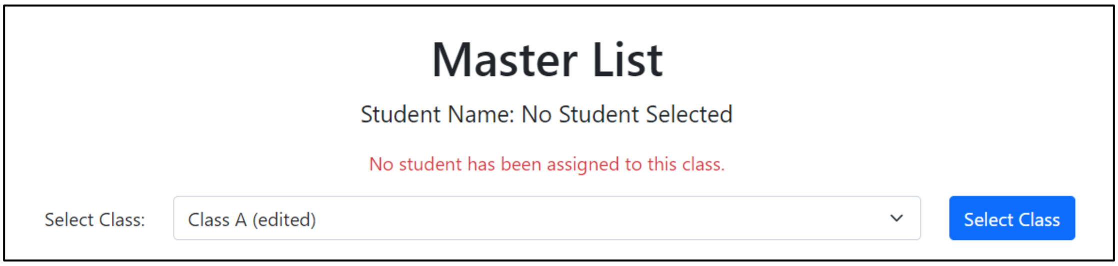
Only one project can be submitted for one activity.

Select Class: Class A (edited)

Figure 6.11: Error Message for Alternate Flow – Duplicated Submission for Activities Created by Teacher

Test Case ID	TC_005
Test Item	UC005
Module Name	Project Result Module (Teacher), Project Report Module (Teacher)
Reference Document	ViewModel: ProjectResultMasterListViewModel.cs Pages: ProjectResultMasterList_Teacher.cshtml, ProjectResultMasterList_Teacher.cshtml.cs, ProjectResultPage_Teacher.cshtml, ProjectResultPage_Teacher.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (View Result) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC005.
Pre-Requisite (Special Conditions or Limitations)	Teacher is assigned at least one class.
Test Steps	1. Main Flow: Follows the step in UC005. 2. Alternate Flow – Class Without Students: Follows the steps in UC005. 3. Alternate Flow – Reselect Class: Follows the steps in UC005. 4. Alternate Flow – No Results Found: Follows the steps in UC005. 5. Alternate Flow – Download Result: Follows the steps in UC005.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Class Without Students: Working as expected. 3. Alternate Flow – Reselect Class: Working as expected. 4. Alternate Flow – No Results Found: Working as expected. 5. Alternate Flow – Download Result: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Class Without Students: Working as expected. 3. Alternate Flow – Reselect Class: Working as expected. 4. Alternate Flow – No Results Found: Working as expected. 5. Alternate Flow – Download Result: Working as expected.
Test Result	PASS

Table 6.6: Test Case for UC005



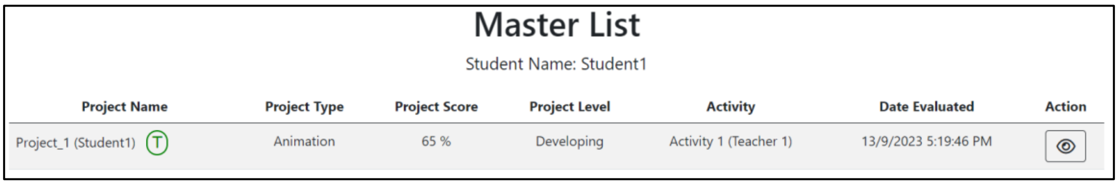
Master List

Student Name: No Student Selected

No student has been assigned to this class.

Select Class: Class A (edited)

Figure 6.12: Error Message for Alternate Flow – Class Without Students

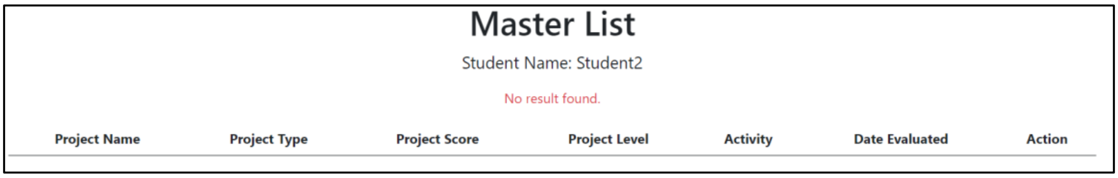


Master List

Student Name: Student1

Project Name	Project Type	Project Score	Project Level	Activity	Date Evaluated	Action
Project_1 (Student1) Ⓣ	Animation	65 %	Developing	Activity 1 (Teacher 1)	13/9/2023 5:19:46 PM	👁

Figure 6.13: Project Result Master List Page (Teacher)



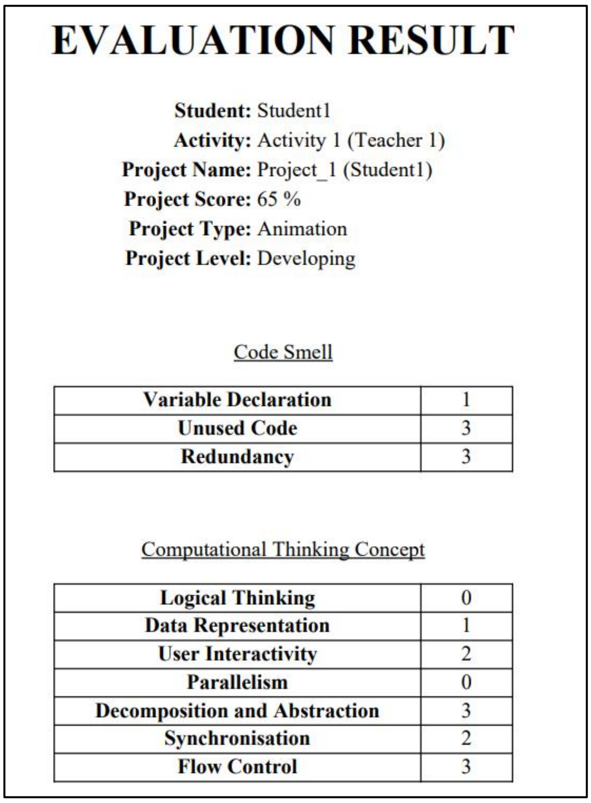
Master List

Student Name: Student2

No result found.

Project Name	Project Type	Project Score	Project Level	Activity	Date Evaluated	Action

Figure 6.14: Error Message for Alternate Flow – No Results Found



EVALUATION RESULT

Student: Student1
Activity: Activity 1 (Teacher 1)
Project Name: Project_1 (Student1)
Project Score: 65 %
Project Type: Animation
Project Level: Developing

Code Smell

Variable Declaration	1
Unused Code	3
Redundancy	3

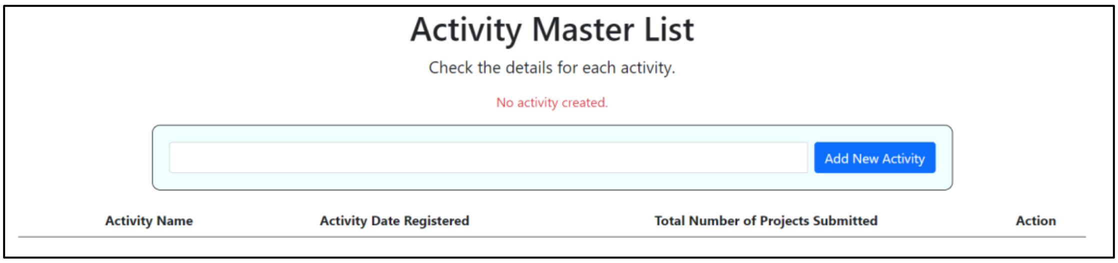
Computational Thinking Concept

Logical Thinking	0
Data Representation	1
User Interactivity	2
Parallelism	0
Decomposition and Abstraction	3
Synchronisation	2
Flow Control	3

Figure 6.15: Downloaded Report (Teacher)

Test Case ID	TC_006
Test Item	UC006
Module Name	Project Activity Module (Teacher)
Reference Document	ViewModel: ScratchActivityDetailViewModel.cs, ScratchActivityViewModel.cs Pages: ManageActivityType.cshtml, ManageActivityType.cshtml.cs, ActivityTypePage.cshtml, ActivityTypePage.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Manage Activity) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC006.
Pre-Requisite (Special Conditions or Limitations)	Teacher is assigned at least one class.
Test Steps	1. Main Flow: Follows the steps in UC005. 2. Alternate Flow – Change Activity Status: Follows the steps in UC006. 3. Alternate Flow – No Activities Found: Follows the steps in UC006. 4. Alternate Flow – Create New Activity: Follows the steps in UC006. 5. Alternate Flow – Delete Student's Submission: Follows the steps in UC006. 6. Alternate Flow – Download Class Performance Report: Follows the steps in UC006.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Change Activity Status: Working as expected. 3. Alternate Flow – No Activities Found: Working as expected. 4. Alternate Flow – Create New Activity: Working as expected. 5. Alternate Flow – Delete Student's Submission: Working as expected. 6. Alternate Flow – Download Class Performance Report: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Change Activity Status: Working as expected. 3. Alternate Flow – No Activities Found: Working as expected. 4. Alternate Flow – Create New Activity: Working as expected. 5. Alternate Flow – Delete Student's Submission: Working as expected. 6. Alternate Flow – Download Class Performance Report: Working as expected.
Test Result	PASS

Table 6.7: Test Case for UC006



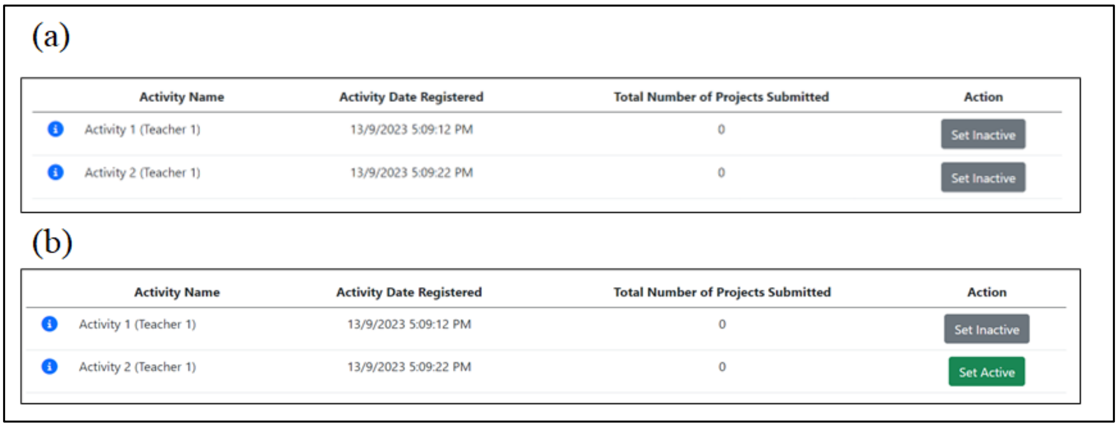
Activity Master List

Check the details for each activity.

No activity created.

Activity Name	Activity Date Registered	Total Number of Projects Submitted	Action

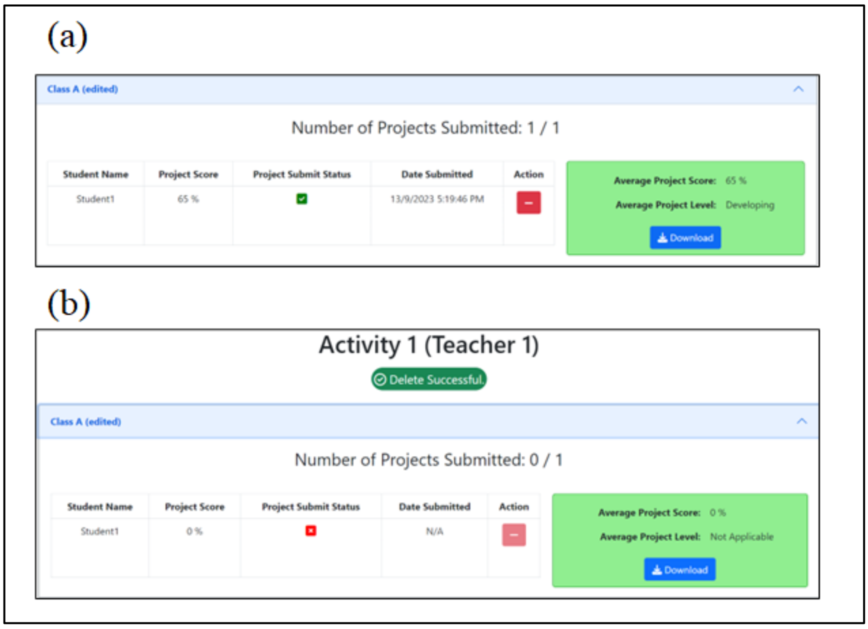
Figure 6.16: Error Message for Alternate Flow – No Activities Found



Activity Name	Activity Date Registered	Total Number of Projects Submitted	Action
Activity 1 (Teacher 1)	13/9/2023 5:09:12 PM	0	Set Inactive
Activity 2 (Teacher 1)	13/9/2023 5:09:22 PM	0	Set Inactive

Activity Name	Activity Date Registered	Total Number of Projects Submitted	Action
Activity 1 (Teacher 1)	13/9/2023 5:09:12 PM	0	Set Inactive
Activity 2 (Teacher 1)	13/9/2023 5:09:22 PM	0	Set Active

Figure 6.17: Alternate Flow – Change Activity Status: (a) Before, (b) After



Class A (edited)

Number of Projects Submitted: 1 / 1

Student Name	Project Score	Project Submit Status	Date Submitted	Action
Student1	65 %	☑	13/9/2023 5:19:46 PM	−

Average Project Score: 65 %

Average Project Level: Developing

Activity 1 (Teacher 1)

Delete Successful.

Class A (edited)

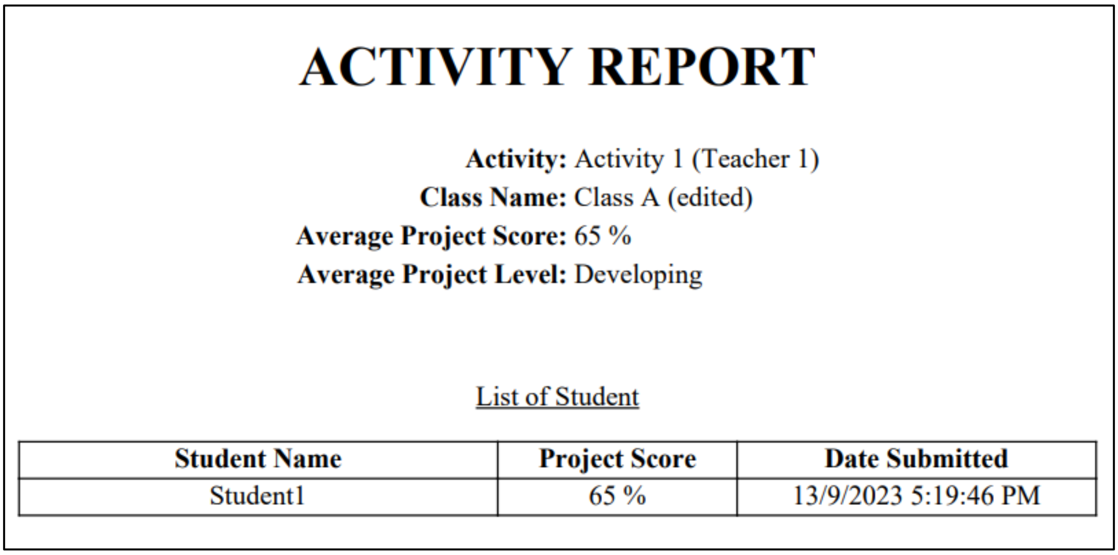
Number of Projects Submitted: 0 / 1

Student Name	Project Score	Project Submit Status	Date Submitted	Action
Student1	0 %	☒	N/A	−

Average Project Score: 0 %

Average Project Level: Not Applicable

Figure 6.18: Alternate Flow – Delete Student's Submission: (a) Before, (b) After



ACTIVITY REPORT

Activity: Activity 1 (Teacher 1)
Class Name: Class A (edited)
Average Project Score: 65 %
Average Project Level: Developing

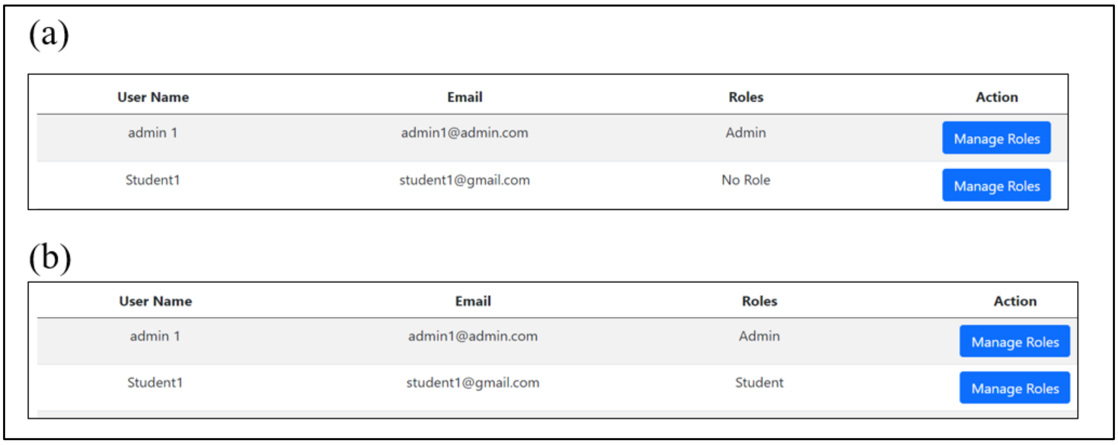
List of Student

Student Name	Project Score	Date Submitted
Student1	65 %	13/9/2023 5:19:46 PM

Figure 6.19: Downloaded Activity Report

Test Case ID	TC_007
Test Item	UC007
Module Name	User Role Management Module (Administrator)
Reference Document	ViewModel: ScratchRoleViewModel.cs, ScratchUsersViewModel.cs Pages: ManageUserRole.cshtml, ManageUserRole.cshtml.cs, UserRolePage.cshtml, UserRolePage.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Manage User Role) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC007.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Main Flow: Follows the steps in UC007. 2. Alternate Flow – Conflict with “No Role”: Follows the steps in UC007. 3. Alternate Flow – Conflict with “Admin” and “Teacher” Roles: Follows the steps in UC007. 4. Alternate Flow – No Roles Selected: Follows the steps in UC007.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Conflict with “No Role”: Working as expected. 3. Alternate Flow – Conflict with “Admin” and “Teacher” Roles: Working as expected. 4. Alternate Flow – No Roles Selected: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Conflict with “No Role”: Working as expected. 3. Alternate Flow – Conflict with “Admin” and “Teacher” Roles: Working as expected. 4. Alternate Flow – No Roles Selected: Working as expected.
Test Result	PASS

Table 6.8: Test Case for UC007



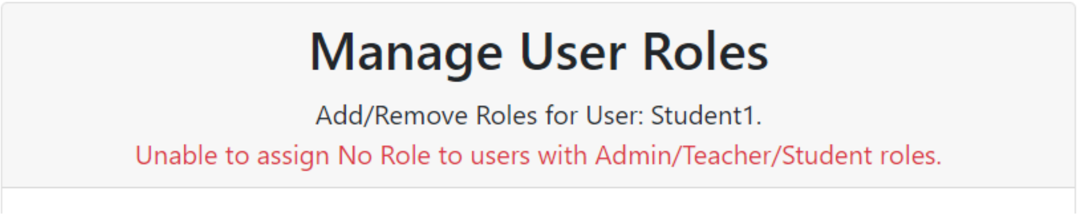
(a)

User Name	Email	Roles	Action
admin 1	admin1@admin.com	Admin	Manage Roles
Student1	student1@gmail.com	No Role	Manage Roles

(b)

User Name	Email	Roles	Action
admin 1	admin1@admin.com	Admin	Manage Roles
Student1	student1@gmail.com	Student	Manage Roles

Figure 6.20: Main Flow: (a) Before Assign A Role, (b) After Assign A Role

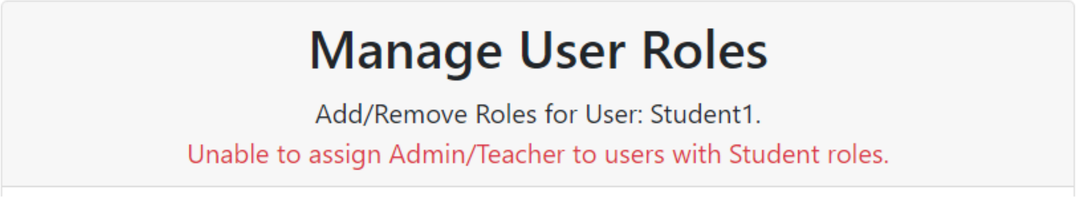


Manage User Roles

Add/Remove Roles for User: Student1.

Unable to assign No Role to users with Admin/Teacher/Student roles.

Figure 6.21: Error Message for Alternate Flow – Conflict with “No Role”



Manage User Roles

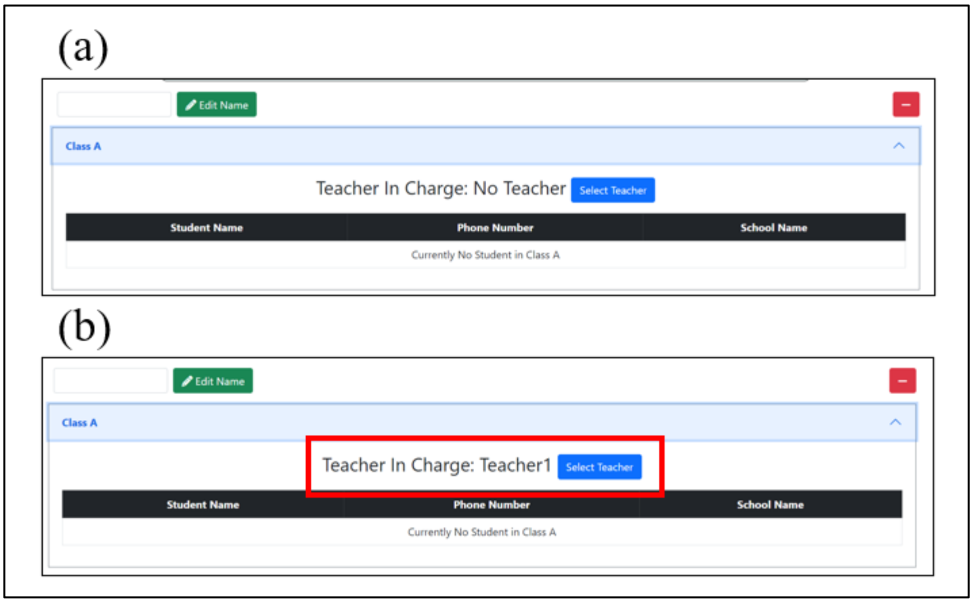
Add/Remove Roles for User: Student1.

Unable to assign Admin/Teacher to users with Student roles.

Figure 6.22: Error Message for Alternate Flow – Conflict with “Admin” and “Teacher” Roles

Test Case ID	TC_008
Test Item	UC008
Module Name	Class Management Module (Administrator)
Reference Document	ViewModel: ScratchClassViewModel.cs Pages: ManageClassList.cshtml, ManageClassList.cshtml.cs, ClassListPage.cshtml, ClassListPage.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Manage Class) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC008.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Main Flow: Follows the steps in UC008. 2. Alternate Flow – Create a New Class: Follows the steps in UC008. 3. Alternate Flow – Delete Class: Follows the steps in UC008. 4. Alternate Flow – No Active Teachers Available: Follows the steps in UC008. 5. Alternate Flow – No Teacher Selected: Follows the steps in UC008.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Create a New Class: Working as expected. 3. Alternate Flow – Delete Class: Working as expected. 4. Alternate Flow – No Active Teachers Available: Working as expected. 5. Alternate Flow – No Teacher Selected: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Create a New Class: Working as expected. 3. Alternate Flow – Delete Class: Working as expected. 4. Alternate Flow – No Active Teachers Available: Working as expected. 5. Alternate Flow – No Teacher Selected: Working as expected.
Test Result	PASS

Table 6.9: Test Case for UC008



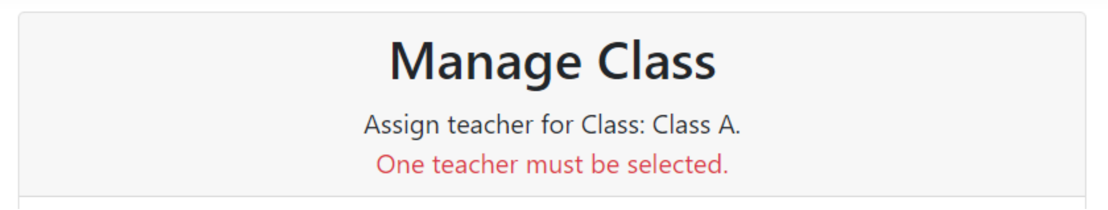
Teacher In Charge: No Teacher

Student Name	Phone Number	School Name
Currently No Student in Class A		

Teacher In Charge: Teacher1

Student Name	Phone Number	School Name
Currently No Student in Class A		

Figure 6.23: Main Flow: (a) Before Assign A Teacher, (b) After Assign A Teacher

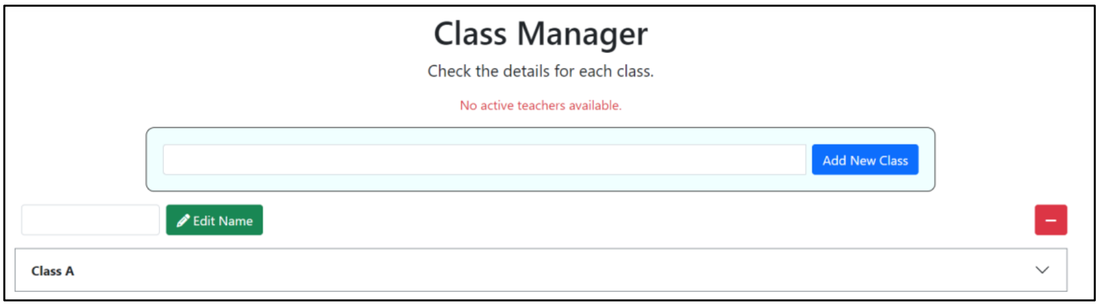


Manage Class

Assign teacher for Class: Class A.

One teacher must be selected.

Figure 6.24: Error Message for Alternate Flow – No Teacher Selected



Class Manager

Check the details for each class.

No active teachers available.

Figure 6.25: Error Message for Alternate Flow – No Active Teachers Available

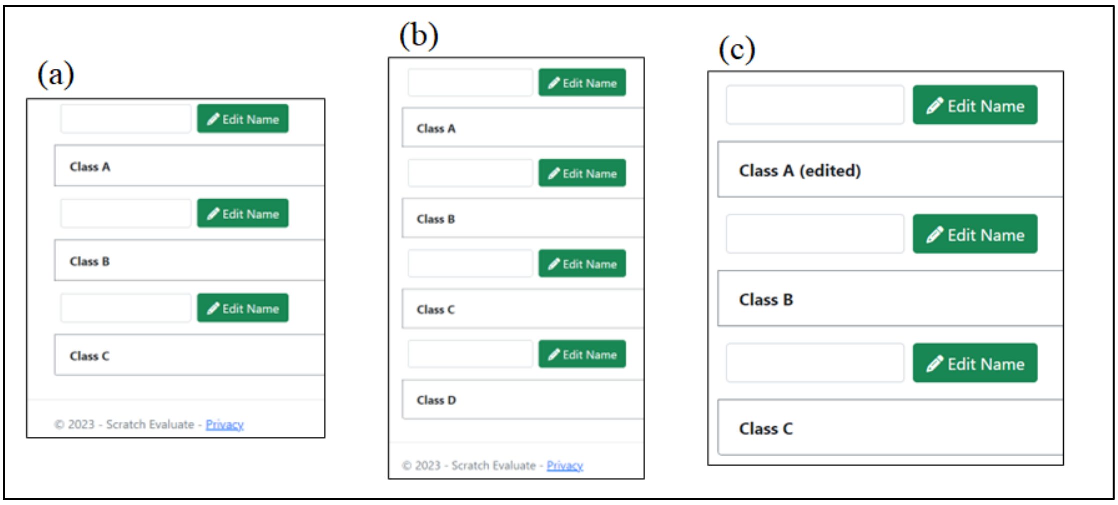
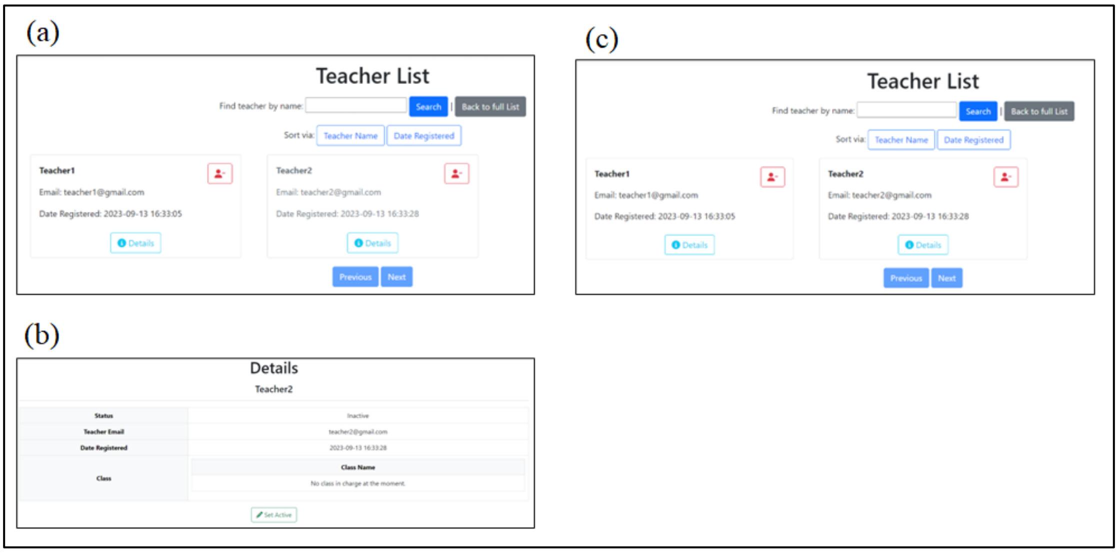


Figure 6.26: Manage Class: (a) Original List of Classes, (b) After Creating a New Class, (c) After Editing a Class Name

Test Case ID	TC_009
Test Item	UC009
Module Name	User Management Module (Administrator)
Reference Document	ViewModel: ScratchClassViewModel.cs Pages: (For Teachers and Students) – Index.cshtml, Index.cshtml.cs, Details.cshtml, Details.cshtml.cs, Delete.cshtml, Delete.cshtml.cs (For Students only) – StudentClassPage.cshtml, StudentClassPage.cshtml.cs
Created By	Ang Jie Qian
Test Objective	To verify that the use case (Manage User Detail) is working as expected.
Test Case Description	Test whether the functionality functions as the flows stated in UC009.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Main Flow: Follows the steps in UC009. 2. Alternate Flow – Set Inactive Teacher/Student Back to Active: Follows the steps in UC009. 3. Alternate Flow – Delete Teacher/Student: Follows the steps in UC009.
Expected Result	1. Main Flow: Working as expected. 2. Alternate Flow – Set Inactive Teacher/Student Back to Active: Working as expected. 3. Alternate Flow – Delete Teacher/Student: Working as expected.
Actual Result	1. Main Flow: Working as expected. 2. Alternate Flow – Set Inactive Teacher/Student Back to Active: Working as expected. 3. Alternate Flow – Delete Teacher/Student: Working as expected.
Test Result	PASS

Table 6.10: Test Case for UC009



Teacher List

Find teacher by name: Search | Back to full List

Sort via: Teacher Name Date Registered

Teacher1

Email: teacher1@gmail.com

Date Registered: 2023-09-13 16:33:05

Details

Teacher2

Email: teacher2@gmail.com

Date Registered: 2023-09-13 16:33:28

Details

Previous Next

Details

Teacher2

Status	Inactive
Teacher Email	teacher2@gmail.com
Date Registered	2023-09-13 16:33:28
Class	Class Name No class in charge at the moment.

Set Active

Teacher List

Find teacher by name: Search | Back to full List

Sort via: Teacher Name Date Registered

Teacher1

Email: teacher1@gmail.com

Date Registered: 2023-09-13 16:33:05

Details

Teacher2

Email: teacher2@gmail.com

Date Registered: 2023-09-13 16:33:28

Details

Previous Next

Figure 6.27: Alternate Flow – Set Inactive Teacher/Student Back to Active: (a) Teacher List with One Active Teacher and One Inactive Teacher, (b) Set Active Button, (c) Teacher List with Two Active Teachers

6.2.2 Randomly Selected Projects from Scratch

Test Case ID	TC_010
Test Item	[16] One Scratch Project from Scratch site (Animation)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the system is evaluating the uploaded projects as required.
Test Case Description	Test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select "Animation" as the Project Type. 3. Select "Individual Project" as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to "View Result" page and check the result.

Expected Result		**Actual Result**	
VD	1	**VD**	1
UnC	3	**UnC**	3
Red	3	**Red**	3
LT	0	**LT**	0
DT	2	**DT**	2
UI	1	**UI**	1
P	1	**P**	1
DA	1	**DA**	1
SYN	1	**SYN**	1
FC	3	**FC**	3
Total (Without Weighting Factor)	16	**Total (Without Weighting Factor)**	16
Project Score	55%	**Project Score**	55%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.11: Test Case for Animation Projects

Test Case ID	TC_011
Test Item	[17] One Scratch Project from Scratch site (Art)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the system is evaluating the uploaded projects as required.
Test Case Description	Test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select "Art" as the Project Type. 3. Select "Individual Project" as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to "View Result" page and check the result.

Expected Result		**Actual Result**	
VD	1	**VD**	1
UnC	2	**UnC**	2
Red	3	**Red**	3
LT	3	**LT**	3
DT	1	**DT**	1
UI	2	**UI**	2
P	1	**P**	1
DA	1	**DA**	1
SYN	2	**SYN**	2
FC	3	**FC**	3
Total (Without Weighting Factor)	19	**Total (Without Weighting Factor)**	19
Project Score	62%	**Project Score**	62%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.12: Test Case for Art Projects

Test Case ID	TC_012
Test Item	[18] One Scratch Project from Scratch site (Games)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the system is evaluating the uploaded projects as required.
Test Case Description	Test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select "Games" as the Project Type. 3. Select "Individual Project" as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to "View Result" page and check the result.

Expected Result		**Actual Result**	
VD	2	**VD**	2
UnC	3	**UnC**	3
Red	1	**Red**	1
LT	3	**LT**	3
DT	2	**DT**	2
UI	1	**UI**	1
P	1	**P**	1
DA	2	**DA**	2
SYN	3	**SYN**	3
FC	3	**FC**	3
Total (Without Weighting Factor)	21	**Total (Without Weighting Factor)**	21
Project Score	70%	**Project Score**	70%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.13: Test Case for Games Projects

Test Case ID	TC_013
Test Item	[19] One Scratch Project from Scratch site (Music)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the system is evaluating the uploaded projects as required.
Test Case Description	Test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select “Music” as the Project Type. 3. Select “Individual Project” as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to “View Result” page and check the result.

Expected Result		**Actual Result**	
VD	3	**VD**	3
UnC	1	**UnC**	1
Red	1	**Red**	1
LT	0	**LT**	0
DT	3	**DT**	3
UI	2	**UI**	2
P	1	**P**	1
DA	3	**DA**	3
SYN	3	**SYN**	3
FC	0	**FC**	0
Total (Without Weighting Factor)	17	**Total (Without Weighting Factor)**	17
Project Score	58%	**Project Score**	58%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.14: Test Case for Music Projects

Test Case ID	TC_014
Test Item	[20] One Scratch Project from Scratch site (Tutorials)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the system is evaluating the uploaded projects as required.
Test Case Description	Test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select "Tutorials" as the Project Type. 3. Select "Individual Project" as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to "View Result" page and check the result.

Expected Result		**Actual Result**	
VD	1	**VD**	1
UnC	3	**UnC**	3
Red	1	**Red**	1
LT	3	**LT**	3
DT	1	**DT**	1
UI	2	**UI**	2
P	1	**P**	1
DA	1	**DA**	1
SYN	3	**SYN**	3
FC	3	**FC**	3
Total (Without Weighting Factor)	19	**Total (Without Weighting Factor)**	19
Project Score	64%	**Project Score**	64%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.15: Test Case for Tutorials Projects

Test Case ID	TC_015
Test Item	[21] One Scratch Project from Scratch site (Stories)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the system is evaluating the uploaded projects as required.
Test Case Description	Test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select “Stories” as the Project Type. 3. Select “Individual Project” as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to “View Result” page and check the result.

Expected Result		**Actual Result**	
VD	1	**VD**	1
UnC	1	**UnC**	1
Red	2	**Red**	2
LT	3	**LT**	3
DT	2	**DT**	2
UI	1	**UI**	1
P	1	**P**	1
DA	3	**DA**	3
SYN	3	**SYN**	3
FC	3	**FC**	3
Total (Without Weighting Factor)	20	**Total (Without Weighting Factor)**	20
Project Score	68%	**Project Score**	68%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.16: Test Case for Stories Projects

Test Case ID	TC_016
Test Item	[21] One Scratch Project from Scratch site (Same as the test item for TC_016)
Module Name	Project Evaluation Module (Teacher), Project Evaluation Module (Student)
Reference Document	Pages: UploadProjectPage.cshtml, UploadProjectPage.cshtml.cs, UploadProjectPage_Teacher.cshtml, UploadProjectPage_Teacher.cshtml.cs, UploadProjectPage.cshtml.cs (Inherit)
Created By	Ang Jie Qian
Test Objective	To verify that the weighting factor is working as expected.
Test Case Description	Using the same test item used in TC_015, evaluate the project under the “Games” category and test whether the system produces the correct result.
Pre-Requisite (Special Conditions or Limitations)	N/A
Test Steps	1. Fill in all the particulars on the Upload Project page. 2. Select “Games” as the Project Type. 3. Select “Individual Project” as the Activity and upload. 4. Wait until the success message is prompted. 5. Goes to “View Result” page and check the result.

Expected Result		**Actual Result**	
VD	1	**VD**	1
UnC	1	**UnC**	1
Red	2	**Red**	2
LT	3	**LT**	3
DT	2	**DT**	2
UI	1	**UI**	1
P	1	**P**	1
DA	3	**DA**	3
SYN	3	**SYN**	3
FC	3	**FC**	3
Total (Without Weighting Factor)	20	**Total (Without Weighting Factor)**	20
Project Score	69%	**Project Score**	69%
Project Level	Developing	**Project Level**	Developing

Test Result	PASS

Table 6.17: Test Case to Check Weighting Factor

6.3 Project Challenges

6.3.1 Time Limitation

A brand-new web application to evaluate the Scratch project needs to be developed within a year, meaning that the development must be done rapidly from planning the software development to gathering requirements and implementing the software. Hence, it is crucial to identify and prioritise the development of main functionalities to ensure that a functional web application can be delivered within the limited timeframe.

In this project, the main functionality identified is to evaluate the uploaded Scratch projects and show the results to the users.

6.3.2 Changing Requirements

The requirements are constantly changing as the development progresses. To cope with this issue, Agile Methodology is implemented by developing the web application module by module. The prototype is developed with increased features until a satisfactory final deliverable is developed.

6.4 Objectives Evaluation

6.4.1 Enhance the accuracy and fairness in evaluating students' performance in using Scratch 3.0 by using a new algorithm

The new algorithm proposed in this project has successfully enhanced the accuracy and fairness in evaluating students' performance. This is because the evaluation no longer relies only on the points obtained from the criteria but also balances the acquisition of marks by introducing a weighting factor and percentage system. As shown in TC_011 (Table 6.12) and TC_014 (Table 6.15), both projects gain the same total points, but there is a slight difference between the final project scores due to the type of project. Besides, to ensure that the weighting factor and percentage system is working, TC_015 (Table 6.16) and TC_016 (Table 6.17) use the same test item to test the web application. Both produce different final project scores which are 68% and 69%.

6.4.2 Introduce different user privileges which are teachers, students and administrators

This increases the efficiency of the workflow as different user privileges play different roles and access different types of functionalities provided by the web application. The main functionalities are described in the use case descriptions (Table 3.1 – Table 3.9 and passed the system testing as shown in Table 6.2 (TC_001) until Table 6.9 (TC_009).

6.4.3 To allow storing and retrieving of past evaluation results

The web application stores the evaluated results in the database and allows teachers and students to retrieve the evaluation results even after leaving the web application. Teachers will be able to check the students' performance at any time, whereas the students can compare their latest performance with past evaluation results to improve their skills in programming.

6.5 Concluding Remark

In short, although this project encountered a few implementation issues and project challenges during the development of the web application, the web application has met the objectives and project scope of this project. Use case testing and selecting random projects from Scratch that represent different types of projects to test are the methods used to determine the usability of the web application. The system testing has proven that the web application achieved its intended purposes.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

In short, Scratch is a common tool for educators to teach younger children about programming. However, it is very tedious for educators to evaluate the projects manually. The existing systems do not provide sufficient support for educators as they do not allow the users to retrieve the results evaluated previously. Besides, the evaluation method potentially introduces unfairness in the grading system as they do not consider the types of projects as one of the factors contributing to the grades.

Thus, this project aims to develop a web application to allow educators to mark the students' projects through the system and retrieve the results when necessary. Students can check on their work themselves by uploading their projects into the systems and retrieving the results when required. Furthermore, the evaluation method used in this project differs from the existing system as it takes the types of projects into account and balances the final results through weighting factors.

The development of the web application is completed and achieved its intended purposes as elaborated in Chapter 6. This implies that the web application is ready to be used for evaluating Scratch projects more efficiently due to the implementation of different types of user privileges and a new evaluation algorithm. Teachers can now evaluate students' projects simply by uploading projects into the web application and waiting for it to produce the evaluation results. Teachers and students can retrieve and review past evaluation results even after leaving the web application as their information is stored in the database permanently.

7.2 Recommendation

One of the recommendations to improve the web application is to implement features that cater for a long list of outputs such as the lists of classes and the lists of users. Firstly, providing a search function enables the teachers and administrators to search for the desired class, activity and user quickly instead of scrolling the lists and searching with the naked eye. Next, a pagination function can help to separate a long list of retrieved results into several pages so that the outputs are not cramped within one page and affect the user experience. Besides, a function that enables the administrators to switch the view while checking the details of teachers and students depending on their preferences can be added. For example, an administrator chooses to view the list of teachers in a card view instead of in a list view.

Apart from introducing features to cater for the long list of outputs, the web application should provide an alternative for the administrators to assign a role to a user. For example, an administrator assigns a “Student” role to 20 users at once, instead of searching for 20 users and assigning the role to them individually.

Lastly, the teachers and students can decide whether they prefer to store the result of the uploaded project in the database or not to store it after uploading a project.

REFERENCES

[1] M. Resnick, J. Maloney, A. M. Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman and Y. B. Kafai, “Scratch: Programming for Everyone,” *Communications of the ACM,* pp. 60-67, 2009.

[2] “Scratch File Format,” [Online]. Available: https://en.scratch-wiki.info/wiki/Scratch_File_Format. [Accessed 18 August 2023].

[3] “Broadcast () (block),” [Online]. Available: https://en.scratch-wiki.info/wiki/Broadcast_()_(block). [Accessed 18 August 2023].

[4] “Broadcast () and Wait (block),” [Online]. Available: https://en.scratch-wiki.info/wiki/Broadcast_()_and_Wait_(block). [Accessed 18 August 2023].

[5] “Control Blocks,” [Online]. Available: https://en.scratch-wiki.info/wiki/Control_Blocks. [Accessed 18 August 2023].

[6] “Variable,” [Online]. Available: https://en.scratch-wiki.info/wiki/Variable. [Accessed 18 August 2023].

[7] J. Moreno-León, M. Román-González and G. Robles, “Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking,” *RED-Revista de Educación a Distancia,* pp. 1-23, September 2015.

[8] J. Moreno-León, M. Román-González and G. Robles, “Towards Data-Driven Learning Paths to Develop Computational Thinking with Scratch,” *IEEE Transactions on Emerging Topics in Computing,* vol. 8, no. 1, pp. 193-205, 2017.

[9] B. H. Ngeow, “Framework to Analyse Scratch Project,” Universiti Tunku Abdul Rahman, 2016.

[10] X. Chai, Y. Sun, H. Luo and M. Guizani, “DWES: A Dynamic Weighted Evaluation System for Scratch based on Computational Thinking,” *IEEE Transactions on Emerging Topics in Computing,* vol. 10, no. 2, pp. 917-932, 2021.

[11] “Wrike,” [Online]. Available: https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/. [Accessed February 2023].

[12] M. Resnick and K. Brennan, “New Frameworks for Studying and Assessing the Development of Computational Thinking,” in *American Educational Research Association*, Vancouver, 2012.

[13] “Art Projects,” n. d. [Online]. Available: https://en.scratch-wiki.info/wiki/Art_Projects. [Accessed 29 August 2022].

[14] “Story Projects,” n. d. [Online]. Available: https://en.scratch-wiki.info/wiki/Story_Projects. [Accessed 29 August 2022].

[15] “Tutorial Projects,” n. d. [Online]. Available: https://en.scratch-wiki.info/wiki/Tutorial_Projects. [Accessed 29 August 2022].

[16] -EbikToons-, “Cha Cha Real,” 2023. [Online]. Available: https://scratch.mit.edu/projects/880183649.

[17] Jazzie-, “Hatsune Miku Vector Trace,” 2023. [Online]. Available: https://scratch.mit.edu/projects/793183941.

[18] -LJCToons-, “Anti-Gravity Platformer Part 2,” 2023. [Online]. Available: https://scratch.mit.edu/projects/880298789.

[19] GarboMuffin, “Minecraft Note Blocks,” 2023. [Online]. Available: https://scratch.mit.edu/projects/569447364/.

[20] Elkchippy, “clear pfp + thumbnail tutorial,” 2023. [Online]. Available: https://scratch.mit.edu/projects/886210451.

[21] Lovelythings, “Shapes,” 2023. [Online]. Available: https://scratch.mit.edu/projects/886087147/.

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 2**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Login and Register Module done.
- Class Management Module done.
- Role Management Module done.

2. WORK TO BE DONE

- Further develop "Project Evaluation" Module.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

Basic login & register feature and pages related to "Administrator" role completed.

Sun Teik Heng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 3**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Partial completion of View Evaluation Result Module.
- Find solution to solve the problem in reading the "project.json" file.

2. WORK TO BE DONE

- Further develop Project Evaluation Module.
- Continue to find the solution to solve the problem in reading the "project.json" file.

3. PROBLEMS ENCOUNTERED

Problem in reading information resides in the "project.json" file:

- "JsonSerializer.Deserialize" is not applicable as it requires the construction of a Class file which matches perfectly with the keys used in the JSON file, but some of the Classes contain dynamic information which will hinder the serialization process.
- Another method is to read JSON without the function by using JSON DOM, but one of the crucial string values (opcode) is nested within a key, which is a randomly generated string representing the specific block used by users in "project.json".

4. SELF EVALUATION OF THE PROGRESS

Currently focusing on Project Evaluation Module. Several approaches have been used to solve the problem mentioned but there is no solution to overcome the problem completely up to now.

Sun Teik Heng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 4**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Successfully find the solution to solve the problem in reading the "project.json" file.

2. WORK TO BE DONE

- Further develop Project Evaluation Module.
- Construct an algorithm to evaluate Scratch projects.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- The problem encountered is finally solved by using Deserialize function and dictionary to obtain the values contained inside the unique ID.
- Will start to progress with defining an algorithm that evaluate Scratch project.

Sun Teik Heng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 5**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Constructed an algorithm to evaluate Scratch projects.

2. WORK TO BE DONE

- Further develop Project Evaluation Module.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- The algorithm to evaluate Scratch projects is defined.
- Will start to progress with the code section that evaluates the Scratch project.

SunTeikHeng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 6**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Completion of "Evaluate Variable Declaration" function (Project Evaluation Module).

2. WORK TO BE DONE

- Further development on "Evaluate Unused Code" and "Evaluate Redundancy" functions (Project Evaluation Module).
- Develop the functions that evaluate 7 Computational Thinking Concepts (Project Evaluation Module).

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- Minor fixing is required for "Evaluate Variable Declaration" function (Project Evaluation Module).
- Out of four modules planned, User Account Management Module is completed, whereas Project Evaluation Module is currently developing in progress. The Project Result Module and Project Report Module are still pending to start.

SunTeikHeng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 7**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Completion of functions to evaluate code smell (Variable Declaration, Unused Code, Redundancy) under Project Evaluation Module.

2. WORK TO BE DONE

- Develop the functions that evaluate 7 Computational Thinking Concepts (Project Evaluation Module).
- Develop Project Result Module.
- Develop Project Report Module.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- Out of four modules planned, User Account Management Module is completed, whereas the Project Evaluation Module is 30% completed.
- The Project Result Module and Project Report Module are still pending to start.

SunTeikHeng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 9**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Completion of evaluation functions for Code Smell (Variable Declaration, Unused Code, Redundancy) and 7 Computational Thinking under Project Evaluation Module.

2. WORK TO BE DONE

- Develop weighting factor and project type and integrate them with the evaluation function (Project Evaluation Module).
- Develop Project Result Module.
- Develop Project Report Module.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- Out of four modules planned, User Account Management Module is completed, whereas the Project Evaluation Module is 80% completed.
- The Project Result Module and Project Report Module are still pending to start.

SunTeikHeng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 10**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Completion of Project Evaluation Module (Integrate the evaluation functions for Code Smell and 7 Computational Thinking with Weighting Factor and Project Type) for Student.
- Completion of Project Result Module for Student.

2. WORK TO BE DONE

- Develop Project Evaluation Module, Project Result Module and Project Report Module for Teacher.
- Introduce and implement new table and feature (ActivityType).
- Allow users to key in their phone numbers and school name.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- Most modules for Student and Admin are completed, the remaining pending to start modules are the modules for Teacher.

Sun Teik Heng

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y4T1	**Study week no.: 12**
Student Name & ID: Ang Jie Qian (19ACB01812)	
Supervisor: Ts. Sun Teik Heng	
Project Title: Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0	

1. WORK DONE

- Completion of Project Evaluation Module, Project Result Module and Project Report Module for Teacher.
- Implemented new table and feature (ActivityType).
- Add phone number and school name into database.

2. WORK TO BE DONE

- Perform system testing.
- Bug fixing if necessary.

3. PROBLEMS ENCOUNTERED

NA.

4. SELF EVALUATION OF THE PROGRESS

- The main functionalities of the web application are fully developed.
- Currently progressing with the system test and bug fixing.

Sun Teik Heng

Supervisor's signature

Student's signature

POSTER

Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0

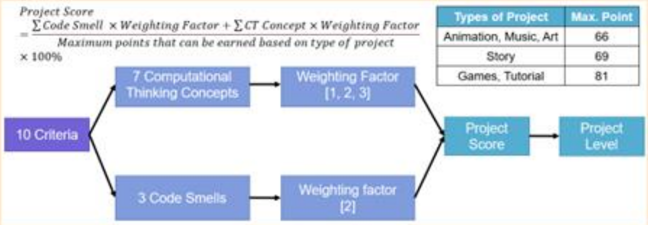
INTRODUCTION

- A web application that eases the teachers' burden in evaluating students' project created using Scratch 3.0.
- Users are only required to upload Scratch projects into the web application and it will do the evaluation automatically.

OBJECTIVES

- To implement new method to evaluate students' performance based on the type of projects.
- To introduce different types of user privileges.
- To allow storing and retrieving of past evaluation results.

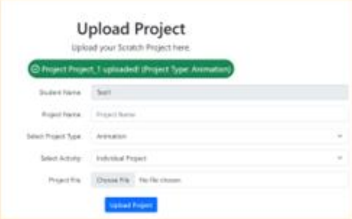
METHODOLOGY



$$\text{Project Score} = \frac{\sum \text{Code Smell} \times \text{Weighting Factor} + \sum \text{CT Concept} \times \text{Weighting Factor}}{\text{Maximum points that can be earned based on type of project}} \times 100\%$$

Types of Project	Max. Point
Animation, Music, Art	66
Story	69
Games, Tutorial	81

FINAL DELIVERABLE



Upload Project

Upload your Scratch Project here.

Project Project_1 uploaded! (Project Type: Animation)

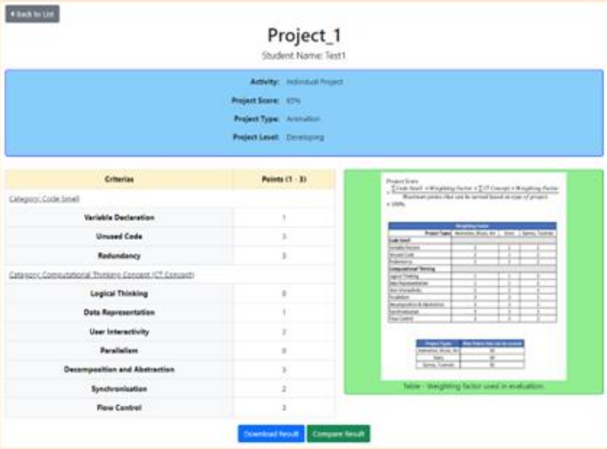
Student Name: Test1

Project Name: Project Name

Select Project Type: Animation

Select Activity: Individual Project

Project File: Choose File No file chosen



Project_1

Student Name: Test1

Activity: Individual Project

Project Score: 67%

Project Type: Animation

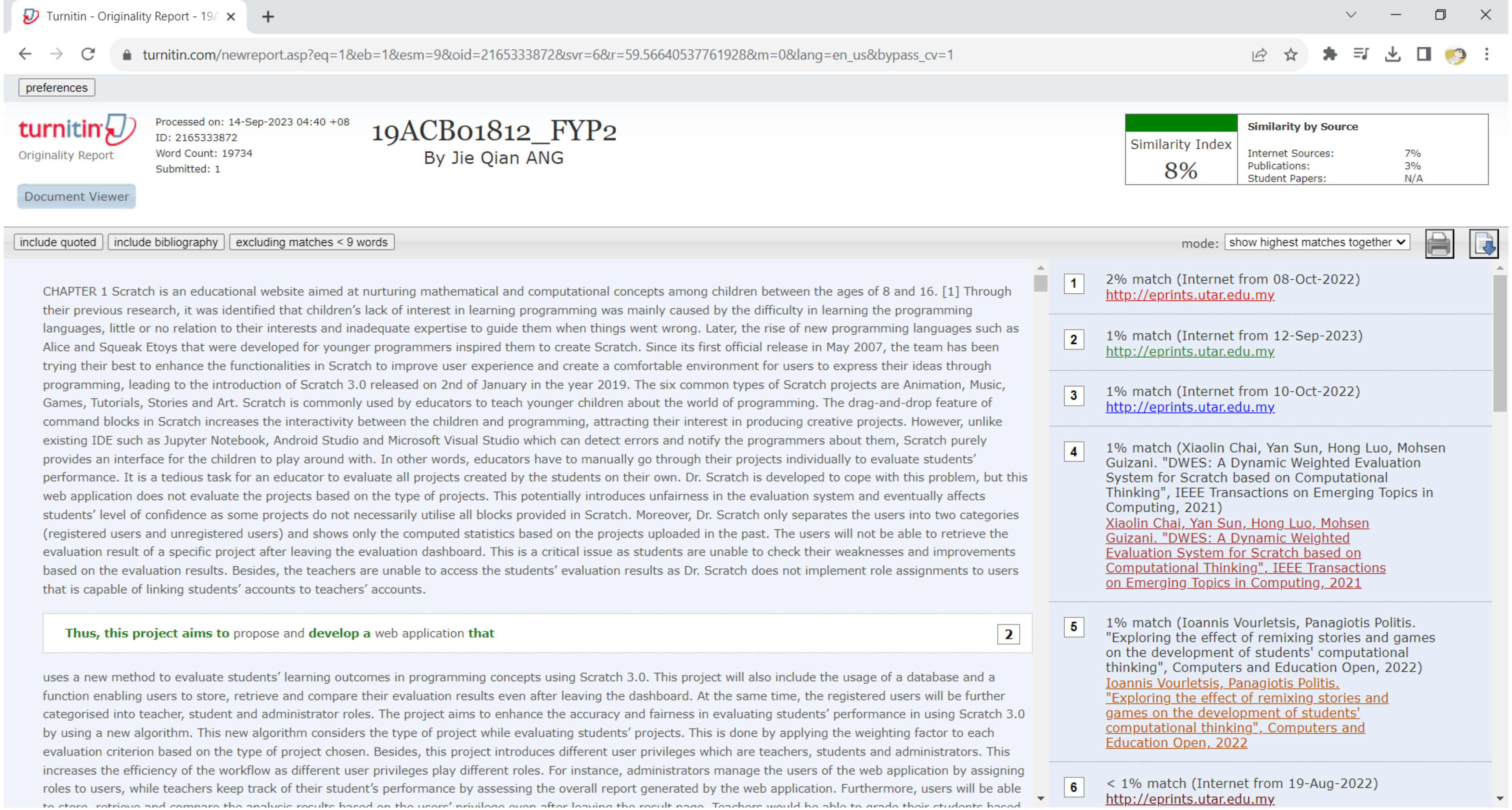
Project Level: Developing

Criterias	Points (1 - 3)
Category: Code Smell	
Variable Declaration	1
Unused Code	3
Redundancy	3
Category: Computational Thinking Concept (CT Concept)	
Logical Thinking	0
Data Representation	1
User Interactivity	2
Parallelism	0
Decomposition and Abstraction	3
Synchronization	2
Flow Control	3

By: Ang Jie Qian
Bachelor of Computer Science (HONS)

Supervisor:
Ts. Sun Teik Heng @ San Teik Heng

PLAGIARISM CHECK RESULT


Turnitin - Originality Report - 19/
turnitin.com/newreport.asp?eq=1&eb=1&esm=9&oid=2165333872&svr=6&r=59.56640537761928&m=0&lang=en_us&bypass_cv=1
preferences
turnitin
Originality Report
Document Viewer
include quoted
include bibliography
excluding matches < 9 words

Processed on: 14-Sep-2023 04:40 +08
ID: 2165333872
Word Count: 19734
Submitted: 1

19ACB01812_FYP2
By Jie Qian ANG

Similarity Index	Similarity by Source	
8%	Internet Sources:	7%
	Publications:	3%
	Student Papers:	N/A

mode: show highest matches together

CHAPTER 1 Scratch is an educational website aimed at nurturing mathematical and computational concepts among children between the ages of 8 and 16. [1] Through their previous research, it was identified that children's lack of interest in learning programming was mainly caused by the difficulty in learning the programming languages, little or no relation to their interests and inadequate expertise to guide them when things went wrong. Later, the rise of new programming languages such as Alice and Squeak Etoys that were developed for younger programmers inspired them to create Scratch. Since its first official release in May 2007, the team has been trying their best to enhance the functionalities in Scratch to improve user experience and create a comfortable environment for users to express their ideas through programming, leading to the introduction of Scratch 3.0 released on 2nd of January in the year 2019. The six common types of Scratch projects are Animation, Music, Games, Tutorials, Stories and Art. Scratch is commonly used by educators to teach younger children about the world of programming. The drag-and-drop feature of command blocks in Scratch increases the interactivity between the children and programming, attracting their interest in producing creative projects. However, unlike existing IDE such as Jupyter Notebook, Android Studio and Microsoft Visual Studio which can detect errors and notify the programmers about them, Scratch purely provides an interface for the children to play around with. In other words, educators have to manually go through their projects individually to evaluate students' performance. It is a tedious task for an educator to evaluate all projects created by the students on their own. Dr. Scratch is developed to cope with this problem, but this web application does not evaluate the projects based on the type of projects. This potentially introduces unfairness in the evaluation system and eventually affects students' level of confidence as some projects do not necessarily utilise all blocks provided in Scratch. Moreover, Dr. Scratch only separates the users into two categories (registered users and unregistered users) and shows only the computed statistics based on the projects uploaded in the past. The users will not be able to retrieve the evaluation result of a specific project after leaving the evaluation dashboard. This is a critical issue as students are unable to check their weaknesses and improvements based on the evaluation results. Besides, the teachers are unable to access the students' evaluation results as Dr. Scratch does not implement role assignments to users that is capable of linking students' accounts to teachers' accounts.

Thus, this project aims to propose and **develop a** web application **that** 2

uses a new method to evaluate students' learning outcomes in programming concepts using Scratch 3.0. This project will also include the usage of a database and a function enabling users to store, retrieve and compare their evaluation results even after leaving the dashboard. At the same time, the registered users will be further categorised into teacher, student and administrator roles. The project aims to enhance the accuracy and fairness in evaluating students' performance in using Scratch 3.0 by using a new algorithm. This new algorithm considers the type of project while evaluating students' projects. This is done by applying the weighting factor to each evaluation criterion based on the type of project chosen. Besides, this project introduces different user privileges which are teachers, students and administrators. This increases the efficiency of the workflow as different user privileges play different roles. For instance, administrators manage the users of the web application by assigning roles to users, while teachers keep track of their student's performance by assessing the overall report generated by the web application. Furthermore, users will be able to store, retrieve and compare the analysis results based on the users' privilege even after leaving the result page. Teachers would be able to grade their students based

1 2% match (Internet from 08-Oct-2022)
http://eprints.utar.edu.my

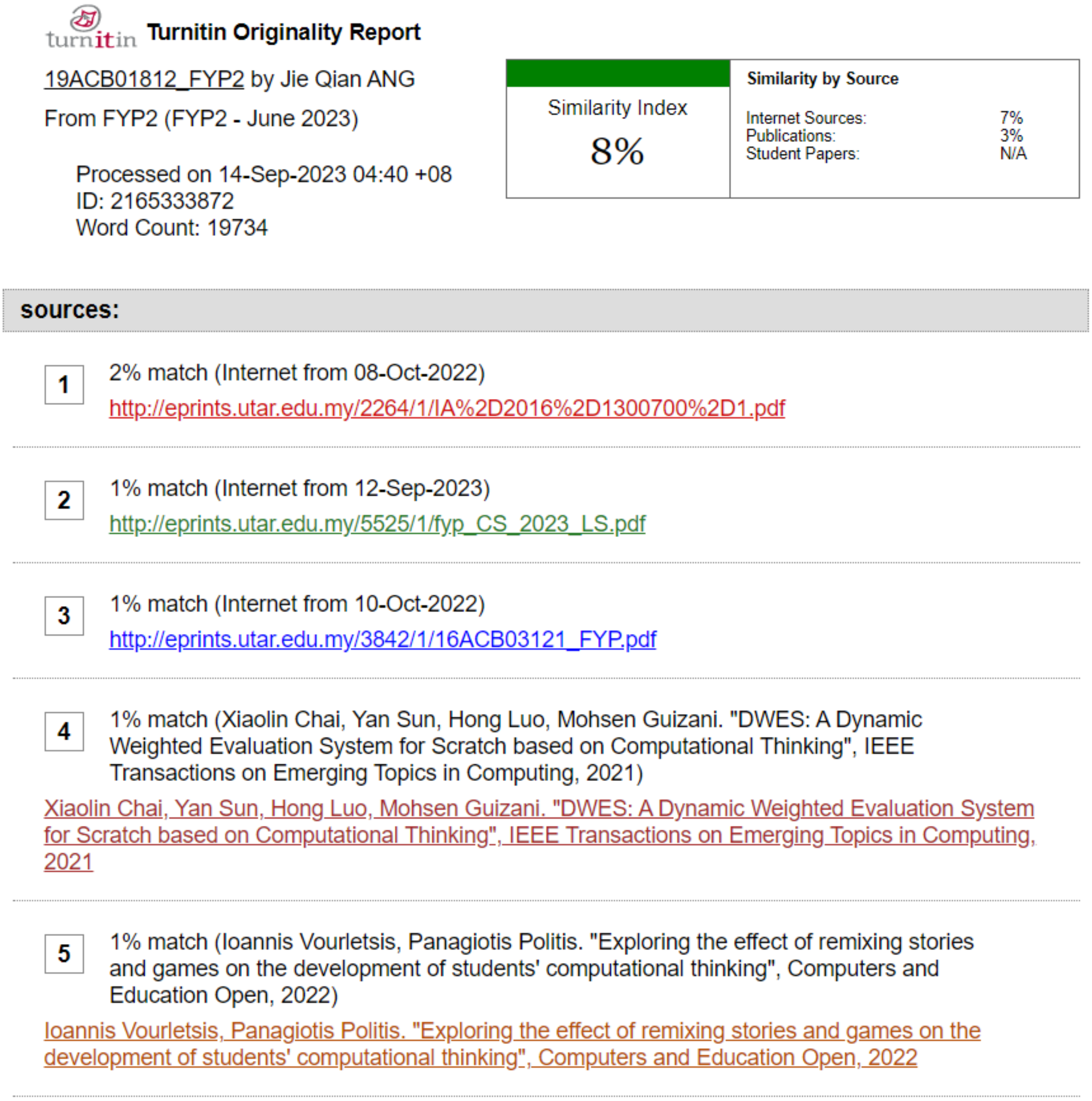
2 1% match (Internet from 12-Sep-2023)
http://eprints.utar.edu.my

3 1% match (Internet from 10-Oct-2022)
http://eprints.utar.edu.my

4 1% match (Xiaolin Chai, Yan Sun, Hong Luo, Mohsen Guizani. "DWES: A Dynamic Weighted Evaluation System for Scratch based on Computational Thinking", IEEE Transactions on Emerging Topics in Computing, 2021)
Xiaolin Chai, Yan Sun, Hong Luo, Mohsen Guizani. "DWES: A Dynamic Weighted Evaluation System for Scratch based on Computational Thinking", IEEE Transactions on Emerging Topics in Computing, 2021

5 1% match (Ioannis Vourletsis, Panagiotis Politis. "Exploring the effect of remixing stories and games on the development of students' computational thinking", Computers and Education Open, 2022)
Ioannis Vourletsis, Panagiotis Politis. "Exploring the effect of remixing stories and games on the development of students' computational thinking", Computers and Education Open, 2022

6 < 1% match (Internet from 19-Aug-2022)
http://eprints.utar.edu.my



Turnitin Originality Report

19ACB01812_FYP2 by Jie Qian ANG

From FYP2 (FYP2 - June 2023)

Processed on 14-Sep-2023 04:40 +08
ID: 2165333872
Word Count: 19734

Similarity Index	Similarity by Source	
8%	Internet Sources:	7%
	Publications:	3%
	Student Papers:	N/A

sources:

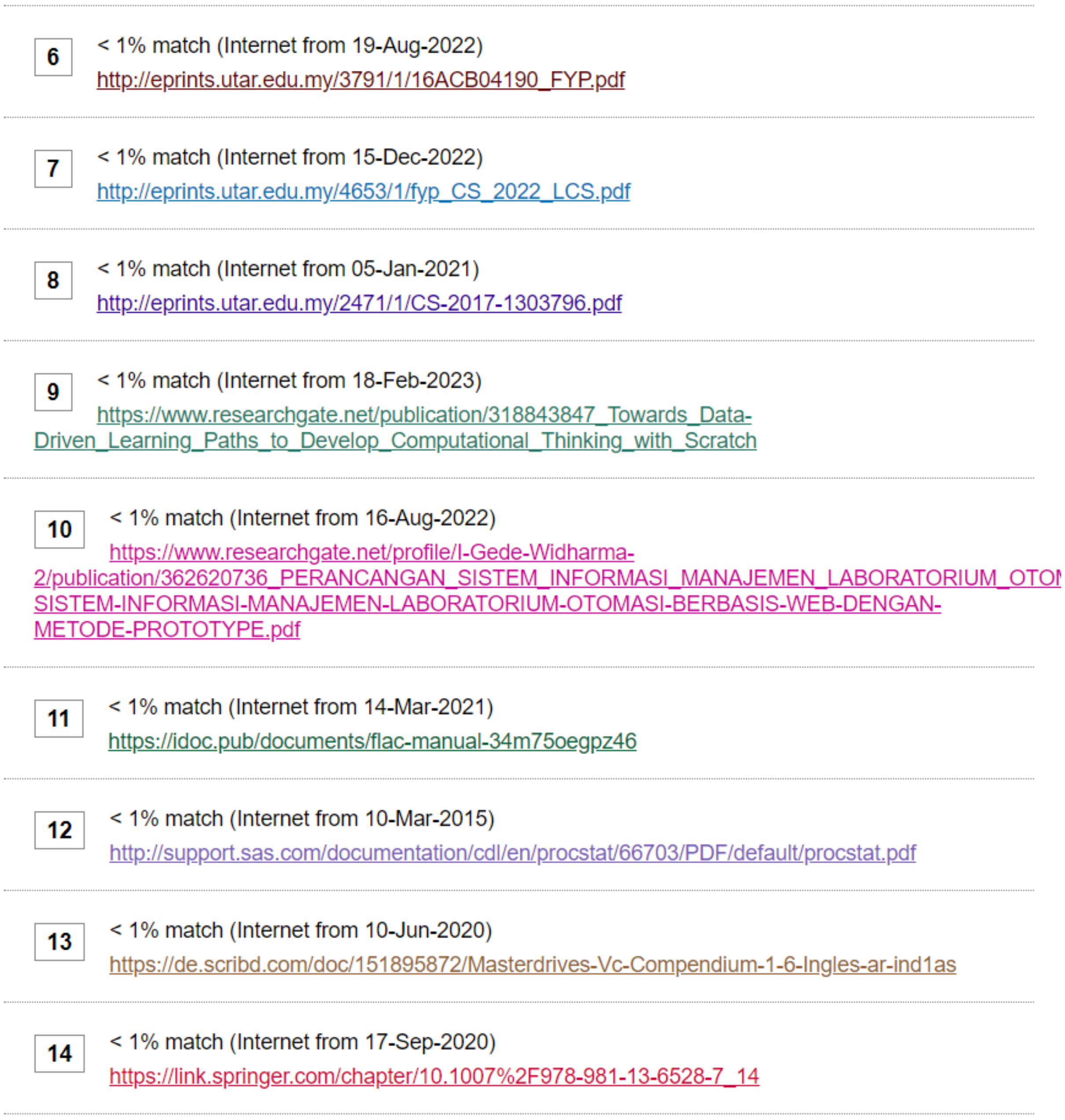
1 2% match (Internet from 08-Oct-2022)
http://eprints.utar.edu.my/2264/1/IA%2D2016%2D1300700%2D1.pdf

2 1% match (Internet from 12-Sep-2023)
http://eprints.utar.edu.my/5525/1/fyp_CS_2023_LS.pdf

3 1% match (Internet from 10-Oct-2022)
http://eprints.utar.edu.my/3842/1/16ACB03121_FYP.pdf

4 1% match (Xiaolin Chai, Yan Sun, Hong Luo, Mohsen Guizani. "DWES: A Dynamic Weighted Evaluation System for Scratch based on Computational Thinking", IEEE Transactions on Emerging Topics in Computing, 2021)
Xiaolin Chai, Yan Sun, Hong Luo, Mohsen Guizani. "DWES: A Dynamic Weighted Evaluation System for Scratch based on Computational Thinking", IEEE Transactions on Emerging Topics in Computing, 2021

5 1% match (Ioannis Vourletsis, Panagiotis Politis. "Exploring the effect of remixing stories and games on the development of students' computational thinking", Computers and Education Open, 2022)
Ioannis Vourletsis, Panagiotis Politis. "Exploring the effect of remixing stories and games on the development of students' computational thinking", Computers and Education Open, 2022



6 < 1% match (Internet from 19-Aug-2022)
http://eprints.utar.edu.my/3791/1/16ACB04190_FYP.pdf

7 < 1% match (Internet from 15-Dec-2022)
http://eprints.utar.edu.my/4653/1/fyp_CS_2022_LCS.pdf

8 < 1% match (Internet from 05-Jan-2021)
http://eprints.utar.edu.my/2471/1/CS-2017-1303796.pdf

9 < 1% match (Internet from 18-Feb-2023)
https://www.researchgate.net/publication/318843847_Towards_Data-Driven_Learning_Paths_to_Develop_Computational_Thinking_with_Scratch

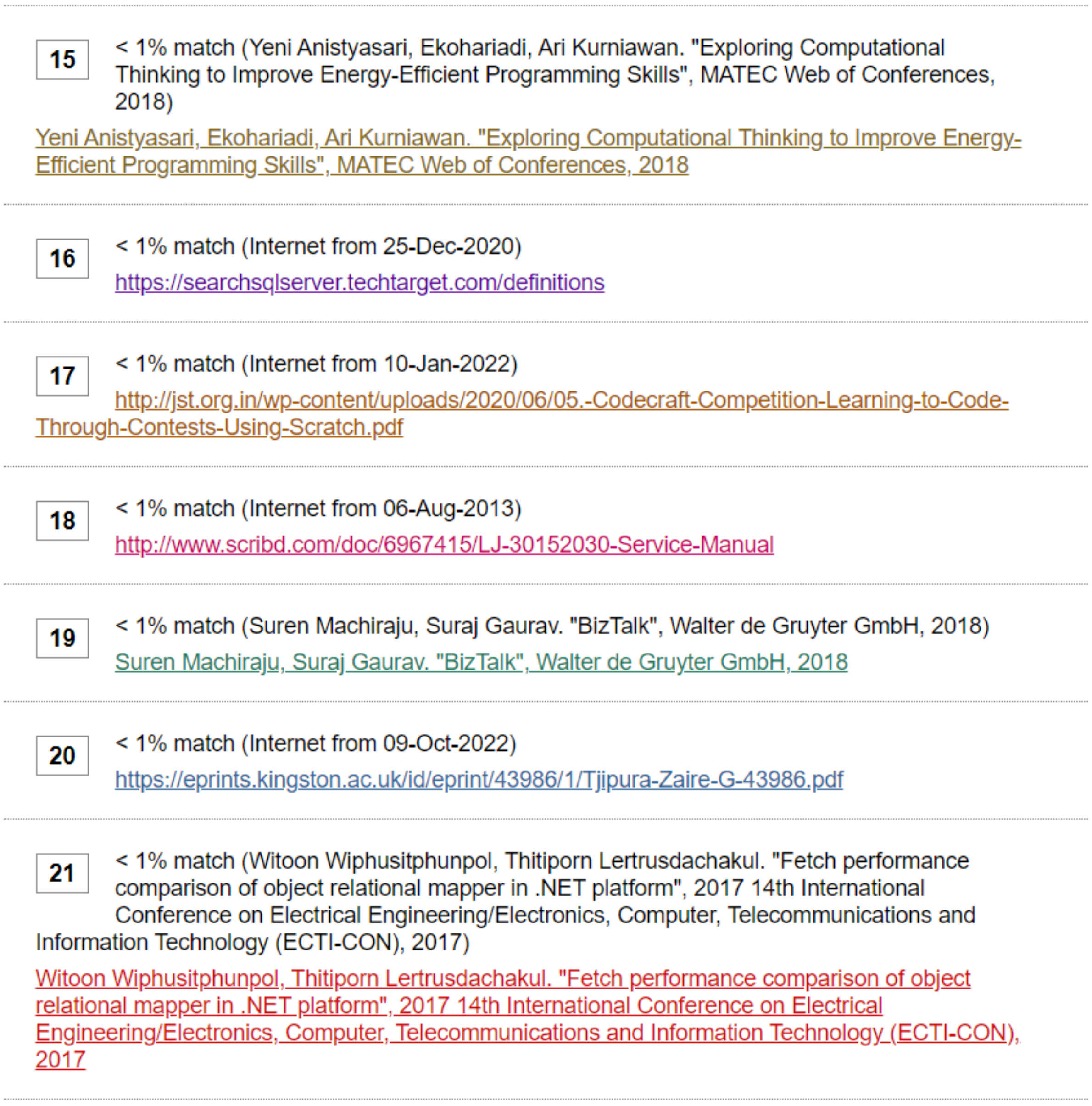
10 < 1% match (Internet from 16-Aug-2022)
https://www.researchgate.net/profile/I-Gede-Widharma-2/publication/362620736_PERANCANGAN_SISTEM_INFORMASI_MANAJEMEN_LABORATORIUM_OTOI
SISTEM-INFORMASI-MANAJEMEN-LABORATORIUM-OTOMASI-BERBASIS-WEB-DENGAN-METODE-PROTOTYPE.pdf

11 < 1% match (Internet from 14-Mar-2021)
https://idoc.pub/documents/flac-manual-34m75oegpz46

12 < 1% match (Internet from 10-Mar-2015)
http://support.sas.com/documentation/cdl/en/procstat/66703/PDF/default/procstat.pdf

13 < 1% match (Internet from 10-Jun-2020)
https://de.scribd.com/doc/151895872/Masterdrives-Vc-Compendium-1-6-Ingles-ar-ind1as

14 < 1% match (Internet from 17-Sep-2020)
https://link.springer.com/chapter/10.1007%2F978-981-13-6528-7_14



15 < 1% match (Yeni Anistyasari, Ekohariadi, Ari Kurniawan. "Exploring Computational Thinking to Improve Energy-Efficient Programming Skills", MATEC Web of Conferences, 2018)

Yeni Anistyasari, Ekohariadi, Ari Kurniawan. "Exploring Computational Thinking to Improve Energy-Efficient Programming Skills", MATEC Web of Conferences, 2018

16 < 1% match (Internet from 25-Dec-2020)

https://searchsqlserver.techtarget.com/definitions

17 < 1% match (Internet from 10-Jan-2022)

http://jst.org.in/wp-content/uploads/2020/06/05.-Codecraft-Competition-Learning-to-Code-Through-Contests-Using-Scratch.pdf

18 < 1% match (Internet from 06-Aug-2013)

http://www.scribd.com/doc/6967415/LJ-30152030-Service-Manual

19 < 1% match (Suren Machiraju, Suraj Gaurav. "BizTalk", Walter de Gruyter GmbH, 2018)

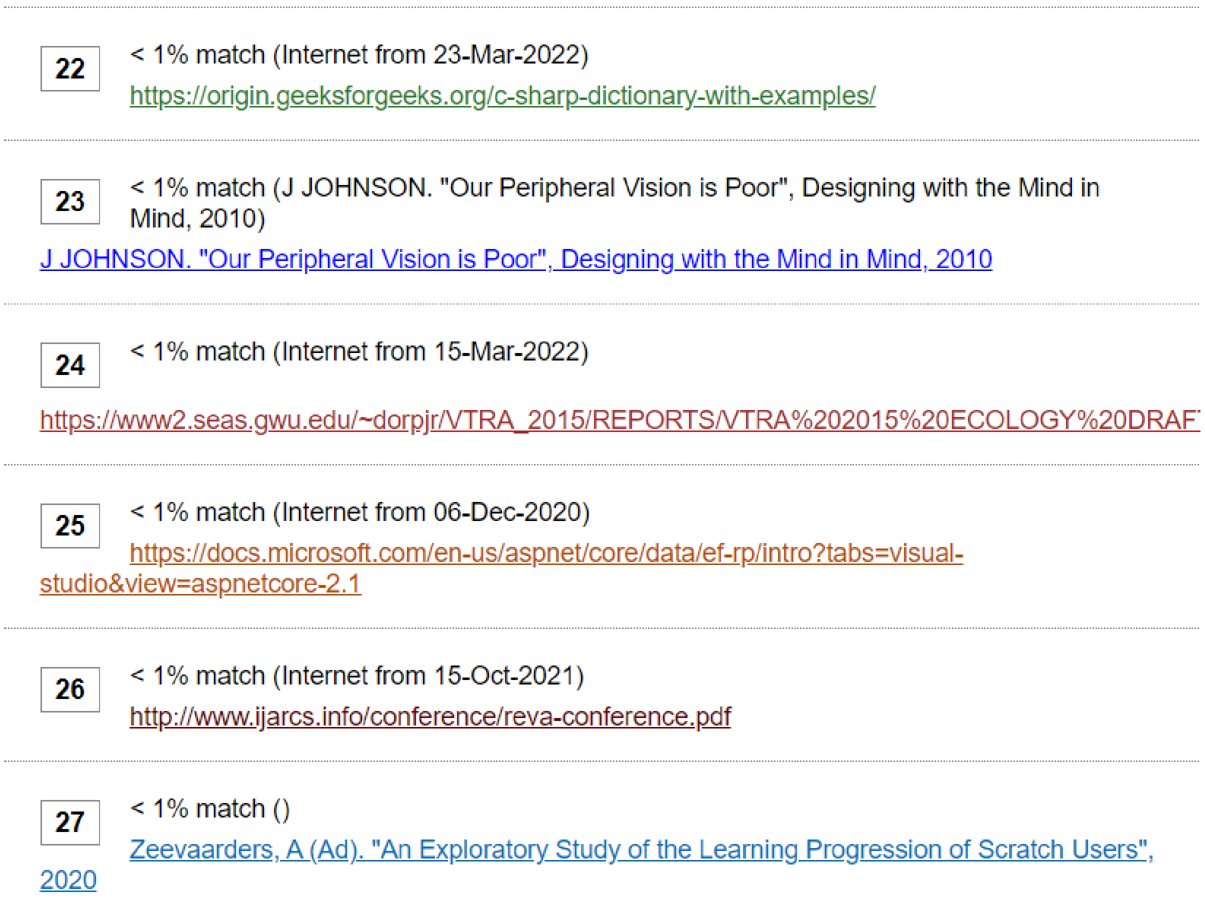
Suren Machiraju, Suraj Gaurav. "BizTalk", Walter de Gruyter GmbH, 2018

20 < 1% match (Internet from 09-Oct-2022)

https://eprints.kingston.ac.uk/id/eprint/43986/1/Tjipura-Zaire-G-43986.pdf

21 < 1% match (Witoon Wiphusitphunpol, Thitiporn Lertrusdachakul. "Fetch performance comparison of object relational mapper in .NET platform", 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2017)

Witoon Wiphusitphunpol, Thitiporn Lertrusdachakul. "Fetch performance comparison of object relational mapper in .NET platform", 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2017



22 < 1% match (Internet from 23-Mar-2022)
https://origin.geeksforgeeks.org/c-sharp-dictionary-with-examples/

23 < 1% match (J JOHNSON. "Our Peripheral Vision is Poor", Designing with the Mind in Mind, 2010)
J JOHNSON. "Our Peripheral Vision is Poor", Designing with the Mind in Mind, 2010

24 < 1% match (Internet from 15-Mar-2022)
https://www2.seas.gwu.edu/~dorpjr/VTRA_2015/REPORTS/VTRA%202015%20ECOLOGY%20DRAF

25 < 1% match (Internet from 06-Dec-2020)
https://docs.microsoft.com/en-us/aspnet/core/data/ef-rp/intro?tabs=visual-studio&view=aspnetcore-2.1

26 < 1% match (Internet from 15-Oct-2021)
http://www.ijarcs.info/conference/reva-conference.pdf

27 < 1% match ()
Zeevaarders, A (Ad). "An Exploratory Study of the Learning Progression of Scratch Users", 2020

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ang Jie Qian
ID Number(s)	19ACB01812
Programme / Course	Computer Science
Title of Final Year Project	Evaluating Students' Learning Outcomes in Programming Concepts Using Scratch 3.0

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 8 % **Similarity by source** Internet Sources: 7 % Publications: 3 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: **0**	

Parameters of originality required and limits approved by UTAR are as Follows:

(i) **Overall similarity index is 20% and below, and**
(ii) **Matching of individual sources listed must be less than 3% each, and**
(iii) **Matching texts in continuous block must not exceed 8 words**

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

SunTeikHeng
Signature of Supervisor

Name: Sun Teik Heng @ San Teik Heng

Date: 15/9/2023

Signature of Co-Supervisor

Name: ______

Date: ______

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB01812
Student Name	Ang Jie Qian
Supervisor Name	Ts. Sun Teik Heng

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
/	Title Page
/	Signed Report Status Declaration Form
/	Signed FYP Thesis Submission Form
/	Signed form of the Declaration of Originality
/	Acknowledgement
/	Abstract
/	Table of Contents
/	List of Figures (if applicable)
/	List of Tables (if applicable)
	List of Symbols (if applicable)
/	List of Abbreviations (if applicable)
/	Chapters / Content
/	Bibliography (or References)
/	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
/	Weekly Log
/	Poster
/	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
/	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____
(Signature of Student)
Date: 14-SEPT-2023