

## SubEdPro: Subtitle Editor Professional

Nur Adilah Adha Ahmad Nawir<sup>1</sup>, Rosziati Ibrahim<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,*

*Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [rosziati@uthm.edu.my](mailto:rosziati@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.066>

### Article Info

Received: 27 July 2024

Accepted: 18 June 2025

Available online: 30 June 2025

### Keywords

Subtitle Editor, Typesetting

### Abstract

Subtitle Editor Professional is a new subtitle editor tool developed to tackle the problem of the existing subtitle editor, which is not user-friendly, has a lot of components, and has limitations. The limitation is that there is no function to change the animation speed, a limited animation direction, and no encoding function. The objectives of this project are to analyze and design the SubEdPro by using the UML specification, to develop the SubEdPro by using object-oriented programming, and to test the SubEdPro by using alpha and beta testing. SubEdPro was developed using the PHP, HTML, and JavaScript languages. The methodology used for SubEdPro development is a simple SDLC model adapted from the Waterfall model. The development of SubEdPro is expected to be able to solve the problem of the existing subtitle editor. Through this tool, users can have a better experience when typesetting the subtitles and reduce the time consumed, as timing the subtitles is already time-consuming.

## 1. Introduction

There are various editors for adding subtitles to videos that are available on the Internet, such as Typito [1], Jubler [2], and Aegisub [3]. Each editor has different functions. For the purpose of the project, Aegisub [3] is chosen as the subtitle editor for reference. Aegisub [3] is an advanced subtitle editor was chosen since it differs from video editors, as its primary function focuses on editing subtitles. The tool is popular among user to produce a self-subbed video of their favorite television shows in English or their native language. For example, anime, variety shows, and even a hospital-themed drama. Subtitles are essential to a viewer as they can help deaf viewers understand a show, and the subtitle usually contains the translation of a video into other languages, which can help viewers understand and learn a new language.

The flaw of the tool is that it is not user-friendly due to the complicated ways to typeset subtitles to the video. It has some limitations, such as changing the speed and the subtitle's direction, and does not include the encode function. Users also need to learn tutorials from the internet to use the tool, as subtitle typesetting involves using a tag similar to coding. The tag is being used to apply animation to the subtitle. In addition to using tags, each animation has different steps to apply the animation. The method is very complicated, and users need to remember it. A survey is conducted to the Aegisub [3] user to justify the problem of the Aegisub [3].

The project's main objectives are to analyze and design using UML specifications, develop using object-oriented programming, and test using alpha and beta testing. There are six modules, which are built-in video player, subtitle timing and editing, subtitle typesetting, karaoke, animation, and encoding.

## 2. Related Work

## 2.1 Study of Subtitle Editors

Subtitle is one of the audiovisual media translations to help people understand other languages in visual [4]. The subtitle is in the form of text that contains the translation of the original video either in English or other languages, which helps the targeted audience understand the content of the video better. To deliver the message, the subtitle must always be in sync with the original video's audio.

Subtitle editors have two basic main functions, which are timing the subtitle with audio and subtitle typesetting. The audio waveform viewer is a function used in the subtitle editor to help users better sync the subtitles with the audio. Subtitle typesetting is a function where users can change the subtitle appearances by changing the font type, size, and color. Users can follow the original video captions style for a more aesthetically pleasant video by having the subtitle typesetting in the subtitle editor.

### 2.1.1 Aegisub

Aegisub [3] is an advanced subtitle editor that is popular among users to produce self-subbed videos of their favorite television shows in English or their native language. For example, anime, variety shows, and even a hospital-themed drama.

The current Aegisub [3] has everything that users need, as the tool is free to use and has tools for typesetting. The typesetting includes changing the subtitle appearance, such as font, size, and color. The typesetting also includes adding animation and effect to the subtitle. For example, moving animation and karaoke effects. Besides, the main function of Aegisub [3] also includes timing the subtitle with the audio and caption in the video. Users can also have a preview of the subtitle when editing, as the software also includes a built-in video player to watch how the subtitle will look in the end product.

### 2.1.2 Typito

Typito [1] is not solely for subtitle editors as it is a video editor. The main focus of Typito [1] is not subtitling but video editing. Typito [1] is a web-based editor, and users must sign up and log in to use the subtitle function. Typito [1] provides three ways for subtitles, which are auto-generating subtitles, uploading subtitle files, and adding subtitles manually.

Next, Typito [1] provides different subtitle styles for user to select so that the subtitle is standardized with the same style. Users can add subtitles manually in the subtitle edit box Typito [1] provided. To time the subtitle, the user must adjust the duration by dragging the subtitle. Typito [1] provides different animation options that users can choose from, and the animations also can adjust their speeds and directions. Typito [1] also provides typesetting, from changing its font style to changing its effect. Lastly, Typito [1] also provides the export function to permanently encode the subtitle with the video. The export function is only available for users who bought the Typito [1] plan.

### 2.1.3 Jubler

Jubler [2] is a basic subtitle editor. Users can either upload subtitle files or add subtitles manually. It also provides a preview, but the user needs to install additional tools to play the video in Jubler [2], which is SMPlayer. It provides an audio waveform viewer to help during timing the subtitle. Jubler [2] has no advanced typesetting function and effect, as its default style is white, which cannot be changed. The position of the subtitle also cannot be changed, and animation cannot be added. Jubler [2] also has no encoding function to permanently encode the subtitle with video.

## 2.2 Audio Waveform Viewer

An audio waveform viewer is a visual representation of the sound waves of audio. [5]. It displays the audio signal as a waveform, with the amplitude represented by the graph's height. The audio waveform viewer is essential during the subtitle process. The waveform can help in timing and syncing the subtitle with the audio better and faster. The audio presence can be detected from the waveform presence in the viewer. The straight line in the viewer shows that no audio is present, so the user can ignore the part. Figure 1 shows the audio waveform viewer of Aegisub [3].

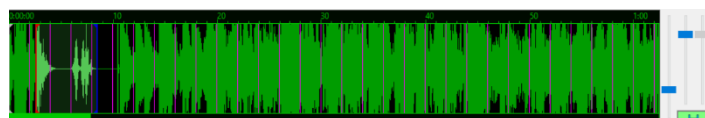


Fig. 1 Audio Waveform Viewer

### 2.3 Comparison with the Existing Tool

The existing subtitle editor, such as Typito, Jubler, and Aegisub, is compared with SubEdPro. The comparison includes the tool functions and user interface design. Table 1 compares Aegisub, Typito, Jubler, and SubEdPro. In summary, SubEdPro is the best tool among the existing subtitle editors listed, as it offers many features for basic and advanced subtitling tasks that are easy to use.

**Table 1** Existing Tool Comparison with the SubEdPro

Features / Tools	Aegisub	Typito	Jubler	SubEdPro
Exclusive as subtitles editor	√	X has editing video tools	√	√
Price	Free	Paid	Free	Free
User interface	Slightly outdated	Modern	Outdated	Modern
Built-in video preview	√	√	√ with additional installation	√
Audio waveform viewer	√	X	√	√
Changing font, color, position, and size	√	√	X	√
Styles Manager	√	X	X	√
Adding text effects	√ but the user needs to follow specific steps	√	X	√
Adding animation	√ but the user needs to follow specific steps	√	X	√
Changing animation speed and direction	X	√	X	√
Encode subtitle with video permanently	X	√ with an upgrade plan	X	√

Based on the comparison, Aegisub [3] is a free subtitle editor with a slightly outdated user interface. It has all the basic subtitling features, including a built-in video preview, audio waveform viewer, and the ability to change subtitles' font, color, position, and size. It can also add text effects and animation but requires specific steps.

Typito [1] is a paid subtitle editor with a modern user interface and a wide range of features, including built-in video preview, and the ability to change subtitles' font, color, position, and size. It can also add text effects and animation and permanently encode subtitles with video. Jubler [2] is a free subtitle editor with an outdated user interface. It has limited subtitling features, but it has a built-in video preview and audio waveform viewer.

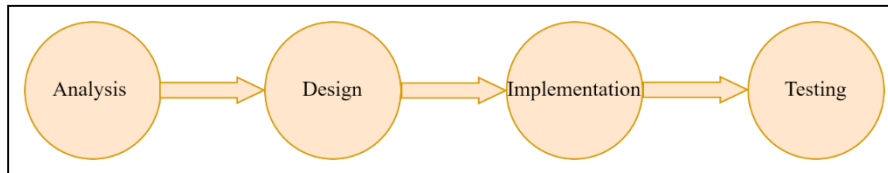
SubEdPro is a free and easy-to-use subtitle editor with a modern user interface. It has all the basic subtitling features, including a built-in video preview, audio waveform viewer, and the ability to change subtitles' font, color, position, and size. It can also add text effects and animation and permanently encode subtitles with video.

Overall, SubEdPro is the most versatile and powerful subtitle editor among the existing subtitle editors listed and offers many features for basic and advanced subtitle editing tasks that are easy for beginner and novice users.

### 3. Methodology

#### 3.1 Simple SDLC Model

A simple SDLC model adapted from the Waterfall model is for the development of this project [6]. The waterfall model is a commonly used model for project development. The model is a sequential approach to managing a project from the start to the end. There are four phases in the simple SDLC model, which include analysis, design, implementation, and testing, as shown in Figure 2.



**Fig. 2** Simple SDLC Model

Table 2 shows the task and output for each phase in the simple SDLC model.

**Table 2** Tool Development Activities and Their Task

Phase	Task	Output
Analysis	<ul style="list-style-type: none"> <li>Determine user requirement</li> <li>Study current tool and existing tool</li> </ul>	<ul style="list-style-type: none"> <li>Questionnaire</li> <li>Functional and non-functional requirements</li> <li>User requirement</li> </ul>
	<ul style="list-style-type: none"> <li>Draw a use case diagram, use case specification, requirement definition, and class diagram</li> </ul>	<ul style="list-style-type: none"> <li>Use case diagram</li> <li>Use case specification</li> <li>Class diagram</li> <li>Requirement traceability matrix</li> </ul>
Design	Design user interface	<ul style="list-style-type: none"> <li>Sign up interface</li> <li>Sign in interface</li> <li>Manage user profile interface</li> <li>Upload video interface</li> <li>Add subtitle interface</li> <li>Time subtitle interface</li> <li>Edit subtitle interface</li> <li>Save file interface</li> <li>Export file interface</li> </ul>
Implementation	Develop subtitle editor	Complete subtitle editor
Testing	Test tools by functional testing and user acceptance testing	Defect-free subtitle editor

### 3.2 Software and Hardware

There are few software and hardware used during the development of the project. Software and hardware are essential for developing a project, as the implementation process cannot be done without their help. Tables 3 and 4 show the software and hardware requirements during the project's development process.

**Table 3** Software Requirement

Software Requirement	Description
Programming Platform	<input type="checkbox"/> Microsoft Visual Studio <input type="checkbox"/> Microsoft Visual Studio Code
Documentation	Microsoft Words
Diagram	Draw.io
Gantt Chart	TeamGantt

**Table 4:** Hardware Requirement

Hardware Requirement	Description
Laptop	Laptop HP 15-fc0082AU

**Table 4:** *cont*

Processor	2.00 GHz AMD Ryzen 5 7530U
Graphic	AMD Radeon Graphics
Random Access Memory (RAM)	16 GB
Storage	512 GB
Peripheral Devices	<input type="checkbox"/> Mouse <input type="checkbox"/> Keyboard

## 4. Analysis and Design

### 4.1 Requirement Analysis

Requirement analysis is one of the structured methodologies that is used to identify attributes needed for the development of the project [7]. The requirement analysis for the development of this project includes the functional requirements, non-functional requirements, and user requirement analysis. A functional requirement is a requirement needed by the user to meet a successful development of a project [8], while A non-functional requirement is a requirement that is used to define the tool’s requirement other than its functionality [9]. Table 5 shows the functional and non-functional requirements of SubEdPro.

**Table 5** *Requirements and its Description of SubEdPro*

Functionality	Description
Register new account	The tool should save the new user information
Sign in	The tool should allow registered account to log in
Built-in video player	The tool should allow the user to view the video while editing
Subtitle timing and editing	The tool should allow the user to time the subtitle with video and edit the subtitle appearances
Save file	The tool should allow the user to save the file in the database
Encode	The tool should allow the user to encode the subtitle

User requirement is the task that the user needs to perform and they are able to perform it for each function that is in the tool [10]. Table 6 shows the user requirement of SubEdPro.

**Table 6** *User Requirements of SubEdPro*

No.	User Requirements
1.	Users can add any video with the format mp4.
2.	Users can choose any font type, color, and size.
3.	Users can choose any animation type.
4.	Users can change the speed of animations.
5.	Users can add effects to the subtitles.
6.	Users can change the subtitle animation direction.
7.	Users can change the subtitle direction.
8.	Users can time the subtitle with the audio and original caption.
9.	Users can encode the subtitles with video permanently.
10.	Users can save the subtitle file.
11.	Users can add subtitles.
12.	Users can view the subtitle preview.

## 4.2 Unified Modelling Language (UML)

UML documents an object-oriented project through the design diagram [11]. The diagram includes the use case diagram, sequence diagram, activity diagram, and class diagram.

### 4.2.1 Use Case Diagram

Use case diagram is one of UML diagram that is used to provide the visual representation of how a user interact with a system. The use case diagram of SubEdPro is shown in Figure 3. There are seven use cases, which are sign up, sign in, manage user profile, upload video, manage subtitle, save file, and export file.

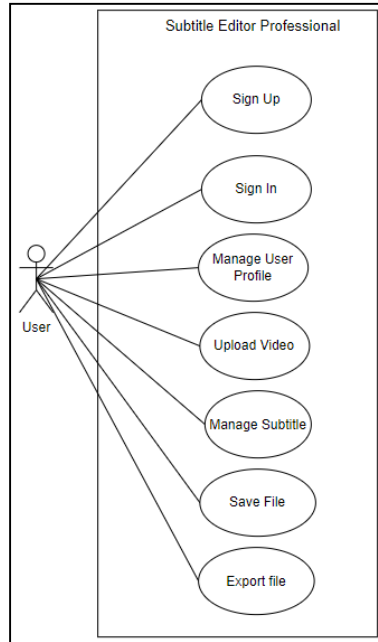


Fig. 3 Use Case Diagram

### 4.2.2 Sequence Diagram and Activity Diagram

A sequence diagram is used to visualize the interaction between objects, while an activity diagram is used to visualize the process flow within a program. Figure 4 [a] and [b] show the activity and sequence diagram of the upload video use case, respectively.

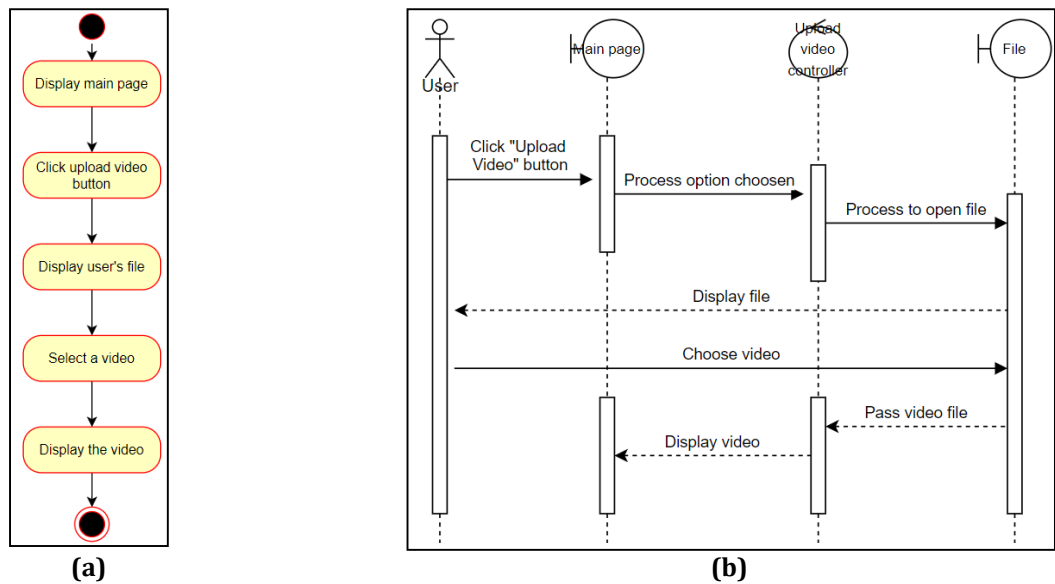


Fig. 4 Upload Video Use Case [a] Activity diagram; [b] Sequence diagram

Figure 5 [a] and [b] shows the activity and sequence diagram of the manage subtitle use case, respectively.

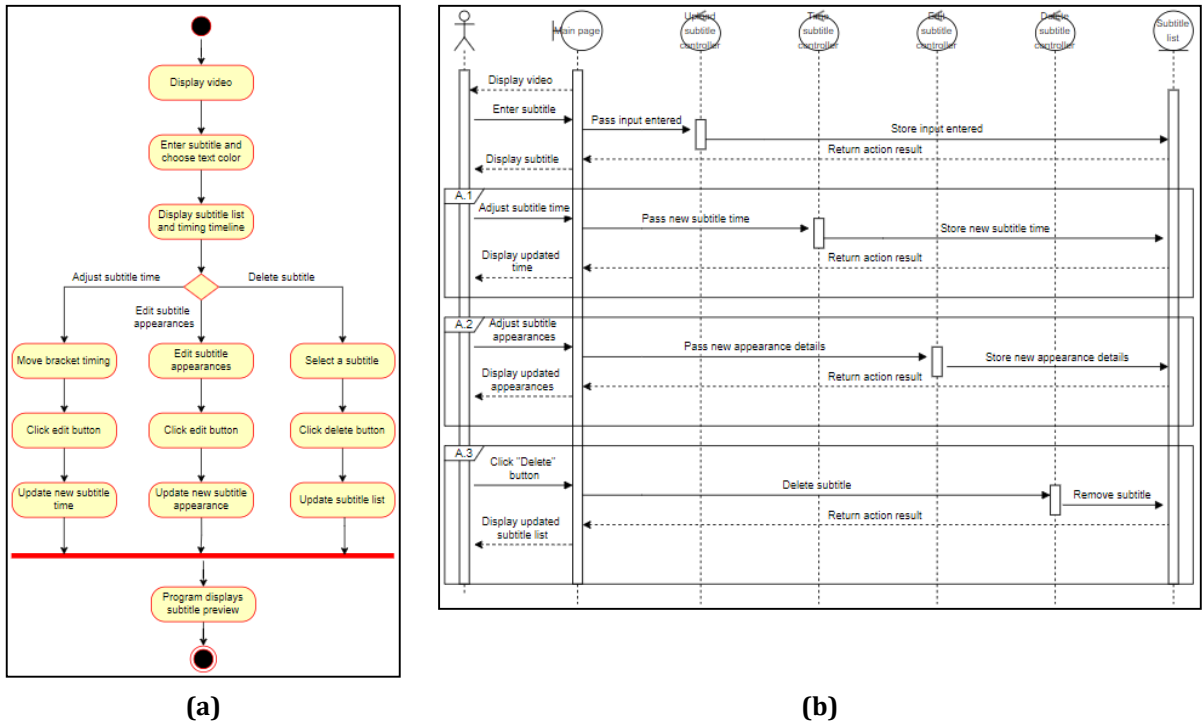


Fig. 5 Manage Subtitle Use Case [a] Activity diagram; [b] Sequence diagram

Figure 6 [a] and [b] shows the activity and sequence diagram of the save file use case, respectively.

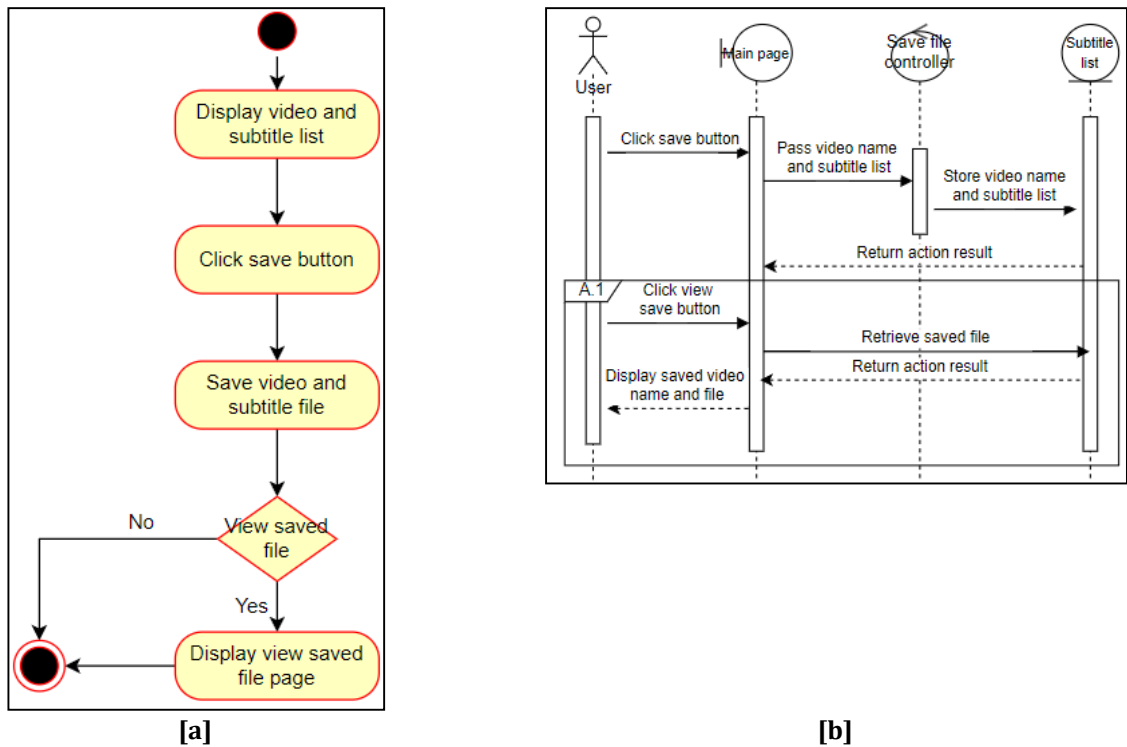


Fig. 6 Save File Use Case [a] Activity diagram; [b] Sequence diagram

Figure 7 [a] and [b] shows the activity and sequence diagram of the export file use case, respectively.

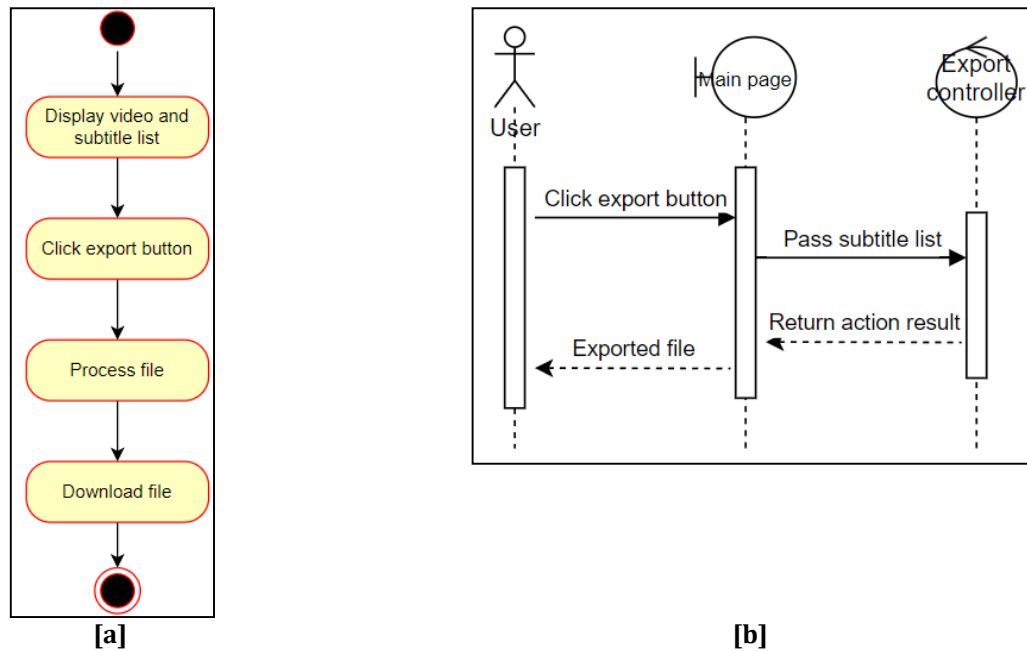


Fig. 7 Export File Use Case [a] Activity diagram; [b] Sequence diagram

### 4.2.3 Class Diagram

The class diagram is a diagram to visualize the relationship between a program’s attributes and functions [12]. Figure 8 shows the class diagram of SubEdPro. There are three classes in the class diagram which are user, video, and subtitle. The user’s class has a one-to-many relationship with the video class, while the subtitle class has a one-to-one relationship with the video class.

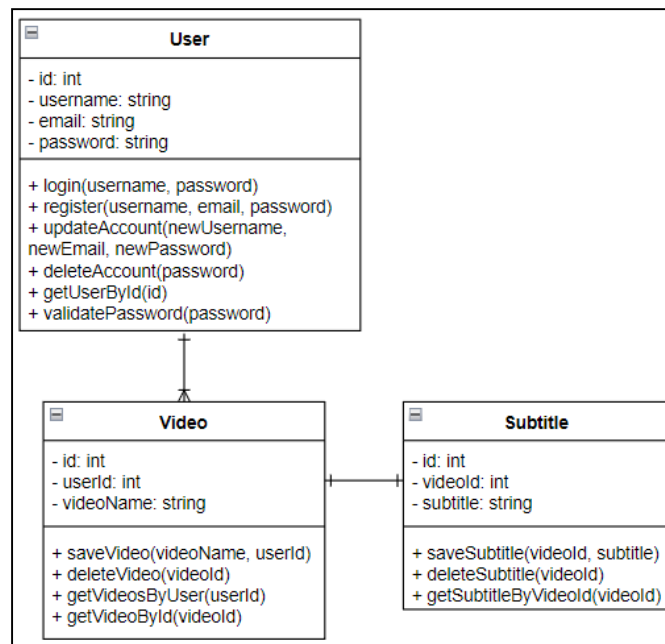


Fig. 8 Class Diagram

### 4.3 Implementation

The project’s implementation uses PHP, HTML, CSS, and JavaScript languages, and the Visual Studio Code is used to develop the program using the languages. XAMPP is used to execute SQL into the database. The user interface is designed to be user-friendly and visually appealing. The interface uses visual elements such as colors and icons to enhance usability and ensure a consistent look and feel.

### 4.3.1 Upload Video

Figure 9 shows the interface for uploading video, and Figure 10 shows the partial coding. Users can upload a video into the program when they click the “Choose File” button. The program will open the user’s computer file.

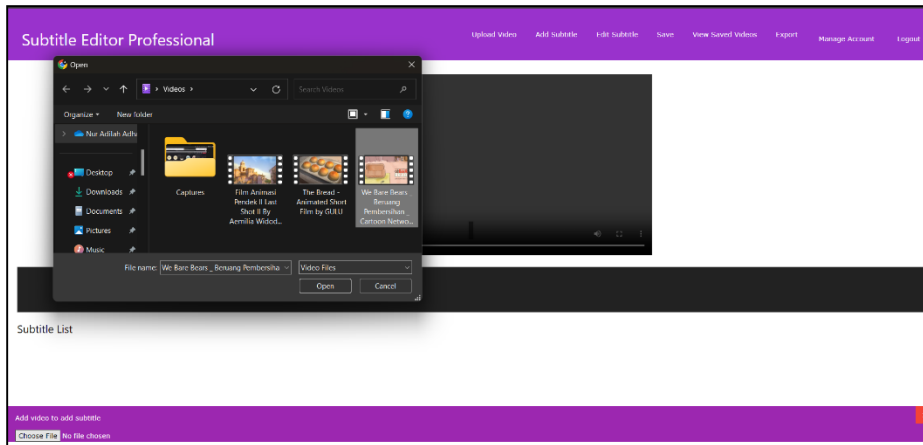


Fig. 9 Upload Video Interface

```
<button class="w3-button w3-none"
onclick="document.getElementById('id01').style.display='block'">Upload
Video</button>
<section id="videoSection">
<div id="id01" class="w3-panel w3-purple w3-display-container"
style="display:none">
  <span onclick="this.parentElement.style.display='none'" class="w3-
button w3-red w3-display-topright">x</span>
  <p>Add video to add subtitle</p>
  <section id="upload">
    <input type="file" id="videoUpload" accept="video/*">
  </section>
</div>
```

Fig. 10 Partial Coding of Upload Video Interface

### 4.3.2 Add Subtitle

Figure 11 shows the interface for adding subtitles, and Figure 12 shows the partial coding. Users can add subtitles to the text box and when the “Add Subtitle” is clicked, the subtitle will appear on the video.

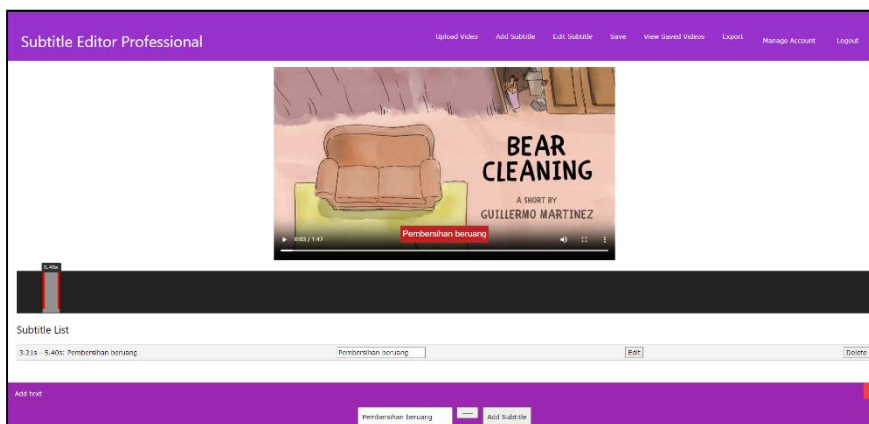


Fig. 11 Add Subtitle Interface

```

<button class="w3-button w3-none"
onclick="document.getElementById('id02').style.display='block'">Add
Subtitle</button>
<div id="id02" class="w3-panel w3-purple w3-display-container" style=
"display:none">
  <span onclick="this.parentElement.style.display='none'" class="w3-
button w3-red w3-display-topright"><</span>
  <p>Add text</p>
  <div id="subtitleControls">
    <input type="text" id="subtitleText" placeholder="Enter subtitle
text">
    <input type="color" id="textColor" value="#FFFFFF">
    <button id="addSubtitle">Add Subtitle</button>
  </div>
</div>

```

Fig. 12 Partial Coding of Add Subtitle Interface

### 4.3.3 Time Subtitle

Figure 13 shows the interface for timing subtitles, and Figure 14 shows the partial coding. Users can time the subtitle by dragging the timing bracket. After clicking the "Edit" button, the program will update the new subtitle time.

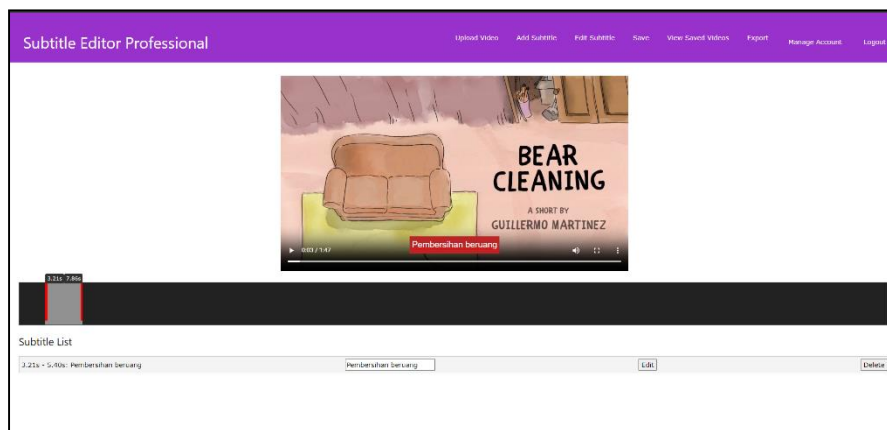


Fig. 13 Time Subtitle Interface

```

document.addEventListener('mousemove', function(e) {
  if (!dragging) return;
  const rect = timeline.getReboundingClientRect();
  const newTime = ((e.clientX - rect.left) / rect.width) *
videoPlayer.duration;
  if (dragging === 'start' && newTime < subtitle.endTime) {
    subtitle.startTime = newTime;
    block.style.left = `${(subtitle.startTime /
videoPlayer.duration) * 100}%`;
    block.style.width = `${((subtitle.endTime -
subtitle.startTime) / videoPlayer.duration) * 100}%`;
    startLabel.innerText = subtitle.startTime.toFixed(2) + 's';
  } else if (dragging === 'end' && newTime > subtitle.startTime) {
    subtitle.endTime = newTime;
    block.style.width = `${(subtitle.endTime -
subtitle.startTime) / videoPlayer.duration) * 100}%`;
    endLabel.innerText = subtitle.endTime.toFixed(2) + 's';
  } else if (dragging === 'move') {
    const blockWidth = (subtitle.endTime - subtitle.startTime) /
videoPlayer.duration;
    if (newTime - blockWidth / 2 >= 0 && newTime + blockWidth /
2 <= videoPlayer.duration) {
      subtitle.startTime = newTime - blockWidth / 2;
      subtitle.endTime = newTime + blockWidth / 2;
      block.style.left = `${(subtitle.startTime /
videoPlayer.duration) * 100}%`;
      startLabel.innerText = subtitle.startTime.toFixed(2) +
's';
      block.style.width = `${blockWidth * 100}%`;
      endLabel.innerText = subtitle.endTime.toFixed(2) + 's';
    }
  }
}

```

Fig. 14 Partial Coding of Time Subtitle Interface

### 4.3.4 Edit Subtitle

Figure 15 shows the interface for editing subtitles, and Figure 16 shows the partial coding. Users can change the subtitle appearances such as font, color, and size. The program will updates the subtitle appearances after the “Edit” button is clicked.

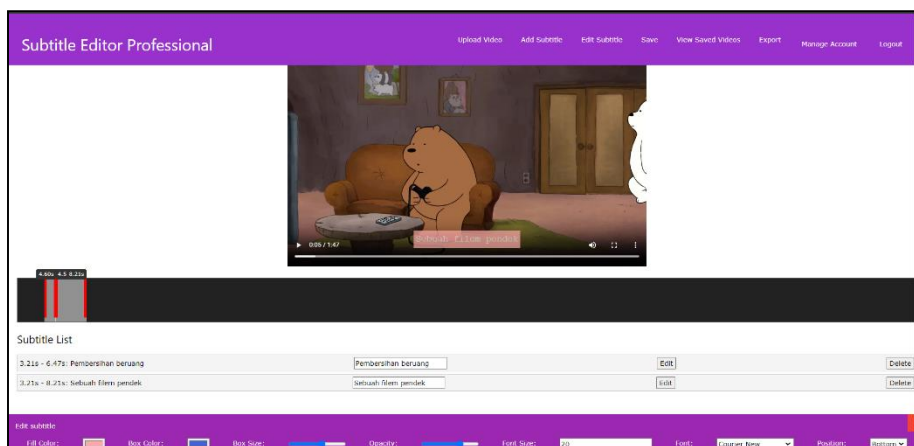


Fig. 15 Edit Subtitle Interface

```

<p>Edit subtitle</p>
<section id="editor">
  <div id="subtitleSettings" id="settings">
    <label for="fillColor">Fill Color:</label>
    <input type="color" id="fillColor" value="#bc2525">
    <label for="strokeColor">Box Color:</label>
    <input type="color" id="strokeColor" value="#000000">
    <label for="strokeWidth">Box Size:</label>
    <input type="range" id="strokeWidth" min="0" max="10"
value="1">
    <label for="opacity">Opacity:</label>
    <input type="range" id="opacity" min="0" max="1" step="0.1"
value="1">
    <label for="fontSize">Font Size:</label>
    <input type="number" id="fontSize" value="20">
    <label for="fontFamily">Font:</label>
    <select id="fontFamily"> </select>
    <label for="subtitlePosition">Position:</label>
    <select id="subtitlePosition">
      <option value="bottom">Bottom</option>
      <option value="top">Top</option>
    </select>
  </div>
</section>
</div>

```

Fig. 16 Partial Coding of Edit Subtitle Interface

### 4.3.5 Save File

Figure 17 shows the interface for saving files, and Figure 18 shows the partial coding. Users can save the subtitle file and video name into the program to be used later.

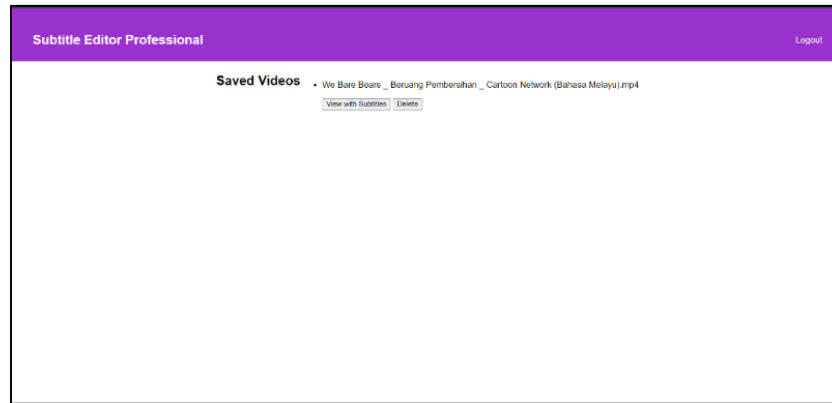


Fig. 17 Save File Interface

```

if (isset($_POST['subtitles']) && isset($_POST['videoName'])) {
    $subtitles = $_POST['subtitles'];
    $videoName = $_POST['videoName'];
    $stmt = $conn->prepare("INSERT INTO videos (user_id, video_name)
VALUES (?, ?)");
    if ($stmt === false) {
        die("Error preparing statement: " . $conn->error);
    }
    $stmt = $conn->prepare("INSERT INTO subtitles (video_id,
subtitle_json) VALUES (?, ?)");
    if ($stmt === false) {
        die("Error preparing statement: " . $conn->error);
    }
}

```

Fig. 18 Partial Coding of Save File Interface

### 4.3.6 Export File

Figure 19 shows the interface for exporting files, and Figure 20 shows the partial coding. Users can export the subtitle file after finished editing. The program will convert the subtitle list into a text file.

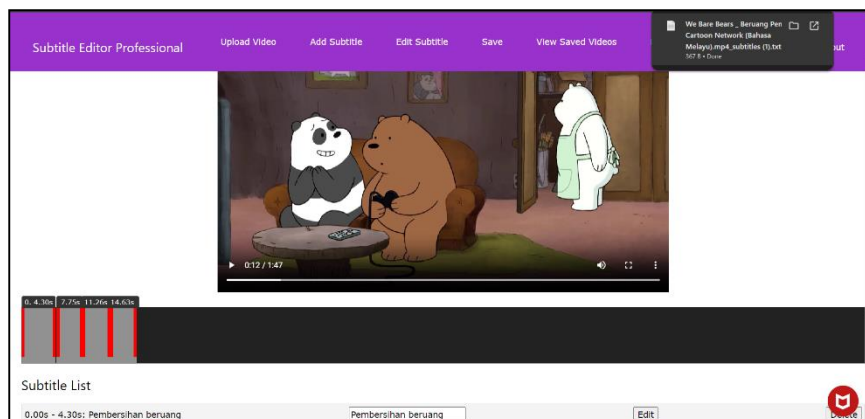


Fig. 19 Export File Interface

```

file_put_contents($subtitleFilePath, $subtitles);
header('Content-Description: File Transfer');
header('Content-Type: application/octet-stream');
header('Content-Disposition: attachment; filename="' .
basename($subtitleFilePath) . '.txt');
header('Content-Length: ' . filesize($subtitleFilePath));
flush();
readfile($subtitleFilePath);

```

Fig. 20 Partial Coding of Export File Interface

## 4.4 Testing

The testing phase was completed after the SubEdPro had been developed. The method used during the testing phase is alpha and beta. The testing process is very important to achieve the objectives and scopes. By doing the testing, errors in the program can be detected.

### 4.4.1 Test Cases

Table 7 shows the list of test cases for each module in the Subtitle Editor Professional. The test cases are used during the functional and user acceptance testing to guide the user. There are 7 test cases, with each test case having sub-test cases.

**Table 7** List of Test Cases

Test Case ID	Requirement ID	Description	Test Status
TC_100	REQ_100	Sign Up	
TC_100_01	REQ_101	The program shall display the signup page	PASS
TC_100_02	REQ_102	The program shall allow the user to enter the registration details	PASS
TC_100_03	REQ_103	The program shall display an error message if the registration details entered by the user are invalid or incomplete	PASS
TC_100_04	REQ_104	The program shall save the registration details entered by the user	PASS
TC_100_05	REQ_105	The program shall not allow registration request that uses the same email address	PASS
TC_200	REQ_200	Sign In	
TC_200_01	REQ_201	The program shall display the login page	PASS
TC_200_02	REQ_202	The program shall allow the user to enter login details	PASS
TC_200_03	REQ_203	The program shall display an error message if the login details are invalid	PASS
TC_200_04	REQ_204	The program shall verify user login details	PASS
TC_200_05	REQ_205	The program shall not allow the user with invalid details to log in to the program	PASS
TC_300	REQ_300	Manage User Profile	
TC_300_01	REQ_301	The program shall allow the user to change their account information	PASS
TC_300_02	REQ_302	The program shall allow the user to enter their information detail	PASS
TC_300_03	REQ_303	The program shall update the new information inside the database	PASS
TC_300_04	REQ_304	The program shall display a success message to notify the user that the changes have been made and saved	PASS
TC_300_05	REQ_305	The program shall display a confirmation message when the user clicks the delete account button	PASS
TC_300_06	REQ_306	The program shall redirect to the login page after the account has been deleted	PASS
TC_300_07	REQ_307	The program shall not allow the user with an invalid old password to change a new password	PASS

**Table 7 (cont)**

TC_300_08	REQ_308	The program shall not allow the user with the wrong password to delete the account	PASS
TC_400	REQ_400	Upload Video	
TC_400_01	REQ_401	The program shall display the main page	PASS
TC_400_02	REQ_402	The program shall allow the user to upload a video	PASS
TC_400_03	REQ_403	The program shall open the file from the user's computer	PASS
TC_400_04	REQ_404	The program shall allow the user to upload the video that has format .mp4	PASS
TC_400_05	REQ_405	The program shall display the video uploaded by the user	PASS
TC_500	REQ_500	Manage Subtitle	
TC_500_01	REQ_501	The program shall display the main page	PASS
TC_500_02	REQ_502	The program shall display the video uploaded by the user	PASS
TC_500_03	REQ_503	The program shall allow the user to add subtitle	PASS
TC_500_04	REQ_504	The program shall allow the user to choose the subtitle appearance	PASS
TC_500_05	REQ_505	The program shall allow the user to change the subtitle time	PASS
TC_500_06	REQ_506	The program shall display the subtitle timing function and subtitle list	PASS
TC_500_07	REQ_507	The program shall allow the user to delete a subtitle	PASS
TC_500_08	REQ_508	The program shall update the subtitle appearance and time	PASS
TC_500_09	REQ_509	The program shall display the subtitle preview	PASS
TC_500_10	REQ_510	The program shall allow the user to add animation	PASS
TC_500_11	REQ_511	The program shall allow the user to change animation speed	PASS
TC_600	REQ_600	Save File	
TC_600_01	REQ_601	The program shall allow the user to save the video name and subtitle file	PASS
TC_600_02	REQ_602	The program shall allow the user to view the saved file	PASS
TC_600_03	REQ_603	The program shall display the video name with its corresponding subtitle file	PASS
TC_600_04	REQ_604	The program shall allow the user to delete the saved file	PASS
TC_600_05	REQ_605	The program shall display the saved subtitle file for the user to re-edit	FAIL
TC_700	REQ_700	Export File	
TC_700_01	REQ_701	The program shall allow the user to export the subtitle file	PASS
TC_700_02	REQ_702	The program shall export the subtitle list into a text file	PASS
TC_700_03	REQ_703	The program shall allow the user to download the file	PASS

**Table 7 (cont)**

TC_700_04	REQ_704	The program shall allow the user to view the file	PASS
-----------	---------	---	------

#### 4.4.2 Overall Result of Functional Testing

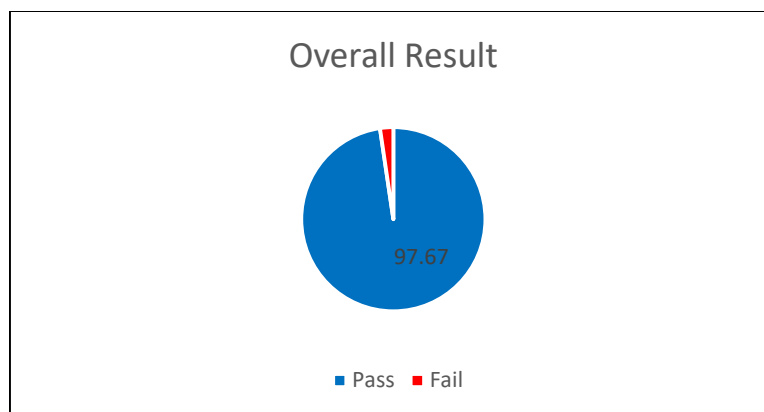
Functional testing is the method where the developer tests the program they developed independently [13]. Table 8 shows the overall result of functional testing. The result shows that there is a function that has not fulfilled the requirements needed.

**Table 8 Overall Test Result of Functional Testing**

Test Cases	Total Test Cases	Total Success	Total Failed
TC_100	5	5	-
TC_200	5	5	-
TC_300	8	8	-
TC_400	5	5	-
TC_500	11	11	-
TC_600	5	4	1
TC_700	4	4	-
Total	43	42	1

#### 4.4.3 Overall Result of User Acceptance Testing

User acceptance testing is a method where a user is given the task to use the program independently and answer the test form to ensure the program meets the user's needs [14]. There are a total of 10 respondents that have taken part in the testing. The test was conducted by giving the link of the website and form to the respondents. There are total of 12 questions asked in the form. Figure 21 shows the overall user acceptance testing result. It can be said that the program only fulfils 97.67% of the test cases.



**Fig. 21 Overall Test Case Result of User Acceptance Testing**

### 5. Conclusion

Overall, the project's objectives stated have been successfully achieved. The program provides a lot of advantages to the individual who wants to add subtitles to their video effectively. However, the program also has a few disadvantages that can be improved in the future to enhance the user experience. Finally, this program hopefully can be able to help the individual to add subtitles to their videos.

The Subtitle Editor Professional is designed and developed to help individuals add subtitles to their videos to make them more understandable and visually appealing. There are some advantages of the program, which are allowing users to add any video, change subtitle appearance, add animations to the subtitle, remove any subtitle they do not want, save the file, export subtitle list to a text file, change subtitle timing, edit account information, delete their account, and providing a built-in video player.

The program's limitations are that it does not allow the user to save the video file inside the program, does not allow the user to change the subtitle position, and does not allow the user to have the encode function to encode the subtitle with video.

Given that the program has some limitations, some improvements can be made in the future to make the program better. The improvements include adding an encode function, and saving video files inside the program.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

## Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

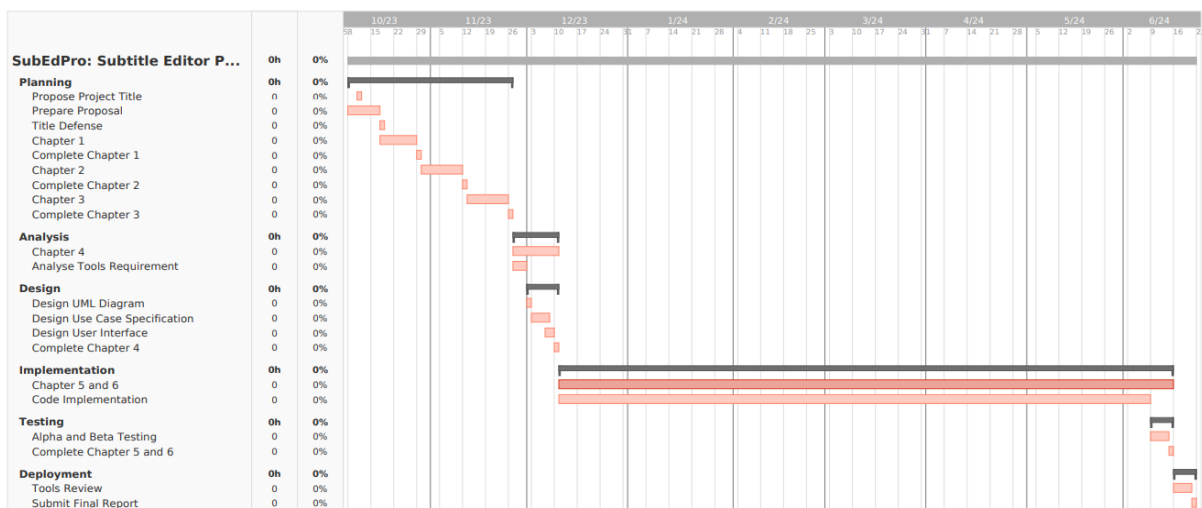
## Author Contribution

The authors confirm contribution to the paper as follows: **study conception and design:** Nur Adilah Adha Ahmad Nawir, Rosziati Ibrahim; **data collection:** Rosziati Ibrahim; **analysis and interpretation of results:** Nur Adilah Adha Ahmad Nawir, Rosziati Ibrahim; **draft manuscript preparation:** Rosziati Ibrahim. All authors reviewed the results and approved the final version of the manuscript.

## References

- [1] Typito, "Typito – Videos for Everyone," [Online]. Available: <https://typito.com/>. [Accessed: 20-oct-2023]
- [2] Jubler, "Jubler Subtitle Editor," [Online]. Available: <https://www.jubler.org/> [Accessed: 22-oct-2023]
- [3] Aegisub "Advanced Subtitle Editor," [Online]. Available: <https://aegisub.org/> [Accessed: 5-sept-2023]
- [4] A. Rohmatika and P. D. Arista, "The use of Aegisub Subtitling Editor in Subtitling class in IAIN Surakarta," Sep. 2019.
- [5] Adobe, "Displaying audio in the waveform editor," [Online]. Available: <https://helpx.adobe.com/audition/using/displaying-audio-waveform-editor.html> [Accessed: 25-oct-2023]
- [6] A. Dennis, B. H. Wixom, and R. M. Roth, *Systems analysis and design*. Hoboken, Nj: Wiley, 2021.
- [7] J. O. Grady, *System requirements analysis*. Amsterdam: Elsevier, 2014.
- [8] S. Alsaleh and H. Haron, "The Most Important Functional and Non-Functional Requirements of Knowledge
- [9] M. Glinz, "On Non-Functional Requirements," *15<sup>th</sup> IEEE International Requirements Engineering Conference (RE 2007)*, Oct. 2007, doi: <https://doi.org/10.1109/re.2007.45>
- [10] K. Wiegers and J. Beatty, *Software Requirements*. Sydney: Pearson Education, 2013.
- [11] B. Dathan and R. Sarnath, *Object-Oriented Analysis and Design*. India: Springer London, 2010.
- [12] Visual Paradigm., "What is Class Diagram?," <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/> [Accessed: 17-nov-2023]
- [13] Javapoint. "Functional testing," [Online]. Available: <https://www.javatpoint.com/functional-testing> [Accessed: 5-apr-2024]
- [14] Standford University Information Technology. (2020). *User Acceptance Testing (UAT)*. [Online]. Available: <https://uit.stanford.edu/pmo/UAT#:~:text=Details,Overview,intended%20audience%20or%20business%20representative.> [Accessed: 1-may-2024]

## Appendix A: Gantt Chart



### Appendix B: Questionnaire’s Question

\* Required

1. Do you require a long time to learn and use Aegisub? \* [🔒]

Yes

No

Maybe

2. Do you agree that Aegisub has a lot of components that is confusing for a first time user? \* [🔒]

Yes

No

Maybe

3. Do you agree that you need to learn through Youtube and Internet to use Aegisub? \* [🔒]

Yes

No

Maybe

4. Do you agree that when you want to add animation through Aegisub, you need to remember the specific tag and steps to add the animation? \* [🔒]

Yes

No

Maybe

5. Based on question 4, do you agree to have a feature that will solve the problem? \* [🔒]

Yes

No

Maybe

6. Do you agree that the animation speed in Aegisub cannot be adjusted? \* [🔒]

Yes

No

Maybe

7. Based on question 6, do you agree to have a feature that can adjust the speed of animation? \* [🔒]

Yes

No

Maybe

8. Do you agree that the animation direction in Aegisub is limited to one direction only? \* [🔒]

Yes

No

Maybe

9. Based on question 8, do you agree to have a feature that solve the problem? \* [🔒]

Yes

No

Maybe

10. Do you agree that Aegisub does not have the feature to encode the subtitle with the video permanently? \* [🔒]

Yes

No

Maybe

11. Based on question 10, do you agree to have a feature to encode the subtitle with the video permanently? \* [🔒]

Yes

No

Maybe

12. Is there anything else you want to share regarding your subtitle needs and preferences? \* [🔒]

### Appendix C: Questionnaire’s Result

