

Educational Resources Website for SMK Ulu Tiram

Nur Syazwani Suhaimi¹, Mohd Zanes Sahid^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,*

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

*Corresponding Author: zanes@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.128>

Article Info

Received: 18 June 2025

Accepted: 19 June 2025

Available online: 30 June 2025

Keywords

Learning management, learning website, educational resources, learning materials, storage issue, digital platform, students, school.

Abstract

In SMK Ulu Tiram, limited exposure to technology in daily secondary school settings necessitates reliance on physical textbooks. However, fluctuating student enrollment poses resource management challenges, with insufficient and deteriorating school-provided textbooks. A survey conducted via Google Form revealed issues in resource storage and management of educational resources. Manual production of modules by school staff results in printing issues, requiring the module to be reproduced. Therefore, they need to provide additional modules. This causes a lot of space, and they don't have a proper space to store the modules. Teachers face time constraints explaining workflows or solutions, resorting to post-work WhatsApp sessions. Since WhatsApp is typically used for personal purposes, students could find it difficult to retain everything related to their studies during the asynchronous discussion. There's a need for a mechanism to store and access information beyond the school sessions. To overcome these challenges, a proposed solution involves the development of an Educational Resources Website. This digital platform aims to replace manual processes, mitigating paper-related problems and serving as a centralized repository for free-access materials. By facilitating student interaction and collaboration, the platform seeks to enhance the educational experience at SMK Ulu Tiram, improving resource management for the school. The teachers will also have the ability to share learning materials to their students via online.

1. Introduction

The learning environment in secondary education differs greatly from the environment in higher education, due to the restriction of personal devices on school grounds. The restriction of such gadgets in this context emphasizes reliance on traditional learning resources such as textbooks, reference books, and exercise books. By serving as vital reference sources, these materials play an important role in forming students' educational experiences. It is the responsibility of the school to ensure that students have access to these vital learning resources. The management of educational resources at SMK Ulu Tiram operates in a separate way. SMK Ulu Tiram has to deal with two challenges regarding textbook supplies. There aren't enough textbooks for every student due to the growing enrolment, which makes it harder for the school to supply enough educational resources. At the same time, current textbooks are heavily used and show wear and tear, which presents a major challenge to their effectiveness. These textbooks' fading text, scribbles on the pages, and damaged covers make them less effective for learning.

The school has a management issue as a result of inadequate storage for physical educational resources, including modules and past year questions. Keeping these educational resources effectively organized and

maintained is difficult when there isn't enough storage space. These modules and exercise books are printed and booked manually, which not only requires a lot of work but also increases the amount of paper used and wastes the environment. The financial strain and environmental damage are further increased by reproduction efforts brought on by printing problems.

Developing an Educational Resources Website is one suggested remedy for the various issues SMK Ulu Tiram is facing. By acting as a comprehensive repository for reference materials, this web-based platform seeks to reduce the paper production, enhance the management flow for educational resources for this school including provide a great learning experience for students and teachers. Moreover, this website will facilitate student interaction through a comment section within the reference resource folder, improving learning experience. The teachers will also have the ability to share learning materials and monitor the sharing of information between the students to ensure accuracy and appropriateness.

This article consists of five parts. The first part explains the background of the project. The second section summarizes the literature review. The third part describes the project methodology and the findings from the system analysis and design. The implementation of the system is shown in the fourth part. The final section summarizes the overall project.

2. Related Work

In this chapter, the discussion about literature review related to the case study are describe in detail, which is an educational resources website and existing system that have been applied by the school. Section 2.1 discuss about the background of the case study for educational resources supply and storage management in SMK Ulu Tiram. Consequently, Section 2.2 discussed about the technology and approach that used in this project development. Finally, the comparison of the existing system is made in Section 2.3.

2.1 Educational Resources Management in SMK Ulu Tiram

An educational resource can produce in variety sources such as, textbooks, videos, learning materials and other. [1] discovered that the material that prepared by the teachers are believed to be more effective than textbooks. Textbooks, the primary resource, undergo a distribution process where students collect them based on course enrollment. Schools meticulously track textbook allocations to ensure returns in good condition or assess compensation for losses. Moreover, other resources like modules, crucial for students who take the major exams like PT3 and SPM which require proper storage before distribution, with each student potentially needing over seven modules. Manual production of modules by school staff often leads to printing issues, necessitating costly reproductions.

Furthermore, teachers provide handouts and past year questions as supplementary materials for their students, requesting bulk printing for multiple classes covering similar topics. The printing room becomes congested as materials accumulate, creating a challenge for resource distribution. The complexities of managing diverse educational materials highlight the need for an efficient system to streamline processes and reduce resource-related hurdles at SMK Ulu Tiram.

2.2 Web-Based Information System

A web-based information system is a system that uses web technologies to convey information and services to the user [2]. One of the most important breakthroughs in the world of information based on Internet technology is the website. Information systems consist of both information technology and human activity related to utilizing technology for better management and operations [3]. If there is a computer network used as an internet medium, this can function without any issues. Web-based information systems are useful for this project because information systems play a critical role in supporting organizations' development of sustainability. A web-based approach is used in the development of educational resources website for SMK Ulu Tiram since this system involves students, tutors and administrator who need to access the system at the same time. Since it is built on web-based technology, building this system is also considerably more economical than producing other apps. Able to access the system via an internet connection and a standard browser without having to install it on a computer makes it easier to manage and operate for the users.

2.3 Existing System

There are three existing systems studied and compared to get more information in developing educational resources website for SMK Ulu Tiram. The existing systems that will be compared are Schoology, Google Classroom and Author UTHM. Schoology was developed in 2007 by Jeremy Friedman, Ryan Hwang and Tim Trinidad. This application is multi-platform application as it can be access through Android and IOS operating system but also can be access on a browser. Google Classroom is a free learning platform developed by Google to help with creating, assigning, and grading assignments easier for educational institutions. Google Classroom's

main goal is to make file sharing between instructors and students easier. Lastly, AUTHOR system that owned by Tun Hussein Onn University's. AUTHOR system is a web-based learning management system (LMS) that allows academic staff to administer their courses and communicate with students online. Announcements, a chat system, file sharing, instructional links, forums, online tests, and assignments are a few of the main features of AUTHOR.

Table 1 shows the summarization of the comparison of 3 existing systems with proposed system. Three existing systems have been examined to obtain more useful modules and features for the proposed system development. The comparisons are between modules and features of the proposed system.

Table 1 Comparison between 3 existing systems with proposed system

Features/System	Schoology	Google Classroom	Author UTHM	Educational Resources Website for SMK Ulu Tiram
Registration	√	√	√ (only admin)	√ (only admin)
Login	√	√	√	√
Classroom & Subject	√	√	√	√
Learning Materials	√	√	√	√
Resources Repository	√	√	√	√
Bookmark	X	X	X	√
Grades / mark	√	√	√	√
Comment Section	√	√	√	√
Students Performance Report	X	X	√	√

The last aspect to compare is the Students Performance Report. The comparison of the three current systems with the proposed system reveals that each of these systems serves a distinct purpose. Only the suggested system has this function, which assists the tutor in determining the degree of skill and knowledge of the students in the subject they are studying.

3. System Methodology

This chapter explain the use of agile model in this project and the activities that had been carried out. The development of Educational Website for SMK Ulu Tiram is using Agile model which consist of six phases which are requirement and analysis, design, development, testing, deployment and review. Each phase has their own deliverables. **Table 2** shows the activity and deliverables for each phase.

Table 2 Software Development Activities and Deliverables

Phase	Task	Deliverables
Requirement & Analysis	<ul style="list-style-type: none"> Distribute question form to the stakeholders. Analyse the requirements. 	<ul style="list-style-type: none"> Functional/ non-functional requirements DFD ERD
Design	<ul style="list-style-type: none"> Design system architecture. Design database schema. Design the user interface. 	<ul style="list-style-type: none"> System architecture diagram Database Schema UI/UX of the system (wireframe)

Table 2 (cont.)

Phase	Task	Deliverables
Development	<ul style="list-style-type: none"> Write the code to implement the designed module. Create database structure. 	<ul style="list-style-type: none"> Source Code Database
Testing	<ul style="list-style-type: none"> Conduct unit testing, system testing, integration testing, usability testing and user acceptance testing. 	<ul style="list-style-type: none"> Test Cases
Deployment	<ul style="list-style-type: none"> Install and configure the system on public. Monitor system performance. 	<ul style="list-style-type: none"> Deployed System
Review	<ul style="list-style-type: none"> Conduct Iteration Review. 	<ul style="list-style-type: none"> Stakeholders feedback. Updated system.

3.1 System Requirement Analysis

The required characteristics that must be defined for an item before any design work can begin are known as requirements. [4] mentioned in his book that the system requirement analysis is an organized method to determining an appropriate set of resources to meet a system's requirements as well as the specifications for those resources that serve as a solid basis for their design or selection. System requirements include functional and non-functional requirements, user requirements and system requirements. **Table 3** summarizes the functional modules of the proposed system.

Table 3 System Functional Module

No	Module	Function	User
1.	User Registration and Authentication Module	<ul style="list-style-type: none"> To register a new student or new teacher. To manage registered user log into the system. To redirect the valid users to its respective dashboard when successful login. 	Administrator, teachers, students
2.	Class and Subject Module	<ul style="list-style-type: none"> To create and manage subjects and classes. To assign students to class and subjects. To create and view learning materials and assignments. To download students' task answers. To update student marks/grade. To reply to student's questions in the comment section. To download and view learning materials and resources. To upload task answers. To view mark/grade. To generate a question through comment section and reply on others comment. 	Administrator, teachers, students
3.	Resources Repository Module	<ul style="list-style-type: none"> To upload and manage the repository for additional educational materials. To view and download the resources. To add the desired resources to the Bookmarks. 	Administrator, teachers, students
4.	Bookmarks Module	<ul style="list-style-type: none"> To display the list of bookmarks items that are set by students. To manage list of bookmarks. 	Students
5.	Reporting Module	<ul style="list-style-type: none"> To generate a student's performance report based on marks. To generate annual student performance report. 	Administrator, teachers

3.2 Functional and Non-Functional Requirement

The abilities that the solution needs to provide to its clients and end users are listed in the functional requirements. It outlines the functionality required for the solution to operate as intended [5]. **Table 4** shows the functional requirements of the proposed system.

Table 4 *Functional Requirements*

No	Module	Descriptions
1.	User Registration and Authentication Module	<ul style="list-style-type: none"> The system should allow administrator to register a new student or new teacher. Allow the registered users to login with the id and password. The system should be able to redirect the valid users to dashboard when successful login.
2.	Class and Subject Module	<ul style="list-style-type: none"> Allow administrator to create and manage subjects and classes. The system should enable admin to assign students to class and subjects. Allow teachers to create and manage learning materials and tasks. The system should allow teachers to download students' task answers. Allow teachers to update student marks/grade. Allow students to download and view learning materials and resources. Allow students to upload task answers. The system should allow students to view mark/grade. Allow students to generate a question through comment section and reply on others comment.
3.	Resources Repository Module	<ul style="list-style-type: none"> The system should be able to allow administrator and teacher to upload resources. Allow admin and teacher to manage their own resources in repository. Allow students to view and download the resources. The system should allow students to be able to add the desired resources to the Bookmarks.
4.	Bookmarks Module	<ul style="list-style-type: none"> The system should display the list of bookmarks items that are set by students to the students. The system should allow students to manage list of bookmarks.
5.	Reporting Module	<ul style="list-style-type: none"> The system should enable admin to generate a student's performance report based on marks. To generate annual student performance report.

Non-functional requirements are often more difficult to articulate and analyze [6]. Additionally, [7] in his book that non-functional requirements relate with "how well" the system should operate in several different domains. **Table 5** shows the non-functional requirements of the developed system.

Table 5 *Non - Functional Requirements*

No	Requirements	Descriptions
1.	Performance and Reliability	<ul style="list-style-type: none"> The system should always be usable. The system should not experience crash or hang other than operating system error.

Table 5 (cont.)

No	Requirements	Descriptions
2.	Operational and Availability	<ul style="list-style-type: none"> The loading time required for a website is no more than 1 minute. The system should always be available except for periodic maintenance. The maintenance period should be pre-scheduled and short.
3.	Security and Privacy	<ul style="list-style-type: none"> The system can be accessed only by authenticate users. Only administrator can add new user (student/teacher).
4.	Cultural and political	<ul style="list-style-type: none"> The system should be able to work on any web browser.
5.	Usability	<ul style="list-style-type: none"> The user interface of the system should make it easy for user to interact and used by user.

3.3 User Requirement Analysis

The user requirements statement defines user expectations from the system [8]. According to [9], they highlighted that when addressing a project's use cases, user requirements are often written down. **Table 6** shows the user requirements of the developed system.

Table 6 User Requirements

No	User	Requirements
1.	Administrator	<ul style="list-style-type: none"> Only admin should be able to generate students' performance report. Admin should be able to manage all user data. Admin should be able to upload and manage educational resources. Admin should be able to create classrooms for teachers and students.
2.	Teacher	<ul style="list-style-type: none"> Teachers should be able to upload learning material and assignments in the classroom. Teachers will be able to upload resources in the resource's repository. Teachers should be able to upload their student's marks.
3.	Student	<ul style="list-style-type: none"> Students should be able to access resources, learning materials and assignments. Students should be able to download resources, tasks, and learning materials. Students should be able to save the desired resources in the bookmark. Students should be able to upload their answers for their task or exercise. Students will be able to view their mark.

3.4 System Design

This system development used data flow diagram (DFD) to illustrate the flow of data and all the process that is executed in this system. The flow of the module process inside the system is shown in the data flow diagram.

3.4.1 Context Diagram

Context diagrams show the overview of interaction between the system and its user or known as the external entities with whom it is intended to engage. Context diagrams also present the flow of input and output to and from its user and system which are useful in understanding the situation in which the system will operate. **Figure 1** shows the context diagram of Educational Resources Website for SMK Ulu Tiram.

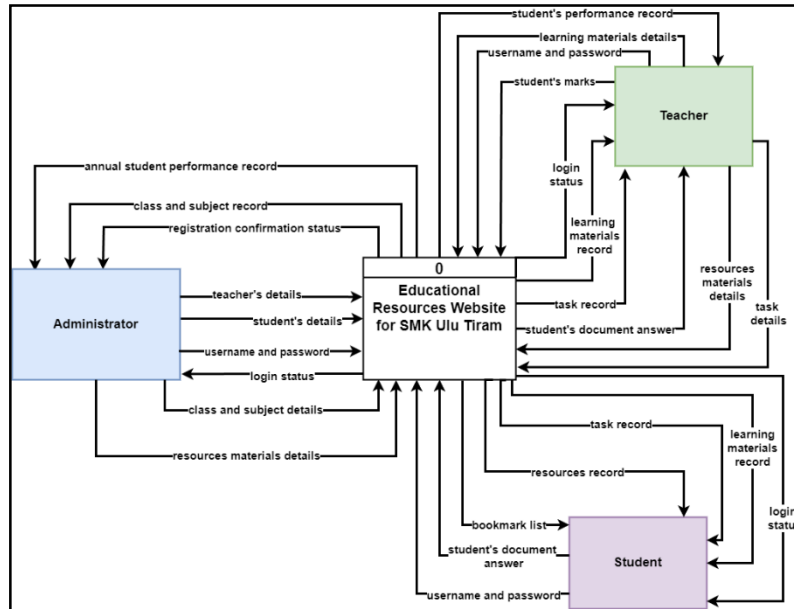


Fig. 1 Context Diagram of Proposed System

3.4.2 Data Flow Diagram

Data Flow Diagram (DFD) is a graphical representation of the flow of data or input from an entity through a process, which then generates output either to another entity or stored in data storage. DFD shows each input and output for each entity and process. The data flow diagram level 0 is the breakdown context diagram that shows in detail the process, data store location and the input or output for the data. Appendix A show the data flow diagram level 0 of the proposed system. It contains 9 processes in total which is registration, login, class creation, learning materials management, task management, grading task, educational resources management, student's performance report generation and bookmarks creation process.

Meanwhile, data flow diagram level 1 is known as the lower-level dataflow because this dataflow is a breakdown for the data flow diagram level 0. Each of the processes inside the data flow diagram level 0 will break down to the very detailed process in data flow diagram level 1. Appendix B shows the DFD Level 1 for each process.

3.5 Entity Relationship Diagram (ERD)

A model to determine entities to be recorded in a database and displaying the relationships between those entities is called the Entity Relational Model. The database schema specified by the entity relationship diagram graphically depicts a database's overall logical structure. The relationships between the entities discovered in the database are illustrated by the entity relationship diagram. Appendix C shows the entity relationship diagram (ERD) for Educational Resources Website for SMK Ulu Tiram.

3.6 User Interface Design

The goal of interface design is to illustrate the flow of user activities and make sure that the interface has features that are simple to use, understand, and accessible in order to support those actions. These interfaces are designed in Justinmind web application. **Figure 2** below shows some user interface design for the proposed system.

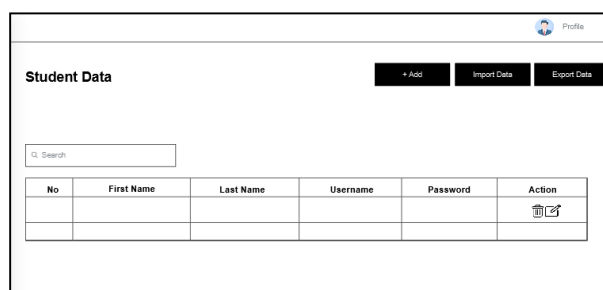


Fig. 2 User Registration by Admin

The registration of new students and new teachers is only done by the administrator. Based on **Figure 2**, this is the example interface design of student registration. Administrators can add student data manually or import the data from excel to register new students into the system. The interface design of registration for new teachers is also similar to this design. After the registration is made, the new user can access the website.

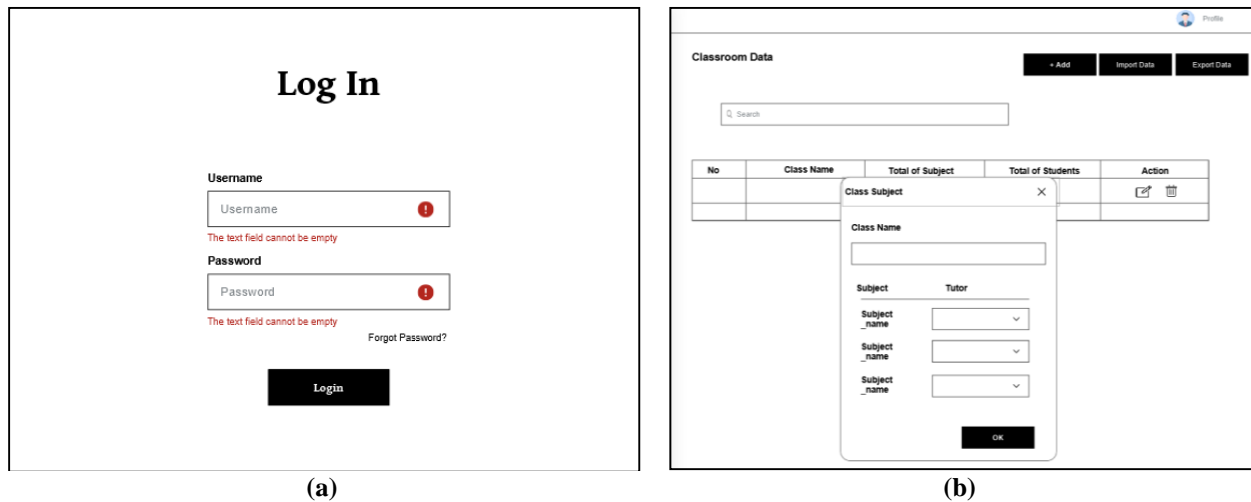


Fig. 3 (a) User Login Page; (b) Admin Classroom Creation Page

For the user to get access to the system, users need to log into the system by providing the correct username and password. The system will then validate the username and password. If both username and password are correct, the system will redirect user to the dashboard. If one of the usernames or passwords is incorrect or empty, the system will automatically display an error message as shown in **Figure 3** (a). After that, the system will request the User to re-enter the correct username and password. In order for students and teachers to access their classroom and subject, administrator needs to create the classroom and subject first including which students enrolled in it and the teachers that assigned to teach that subject as shown in **Figure 3** (b).

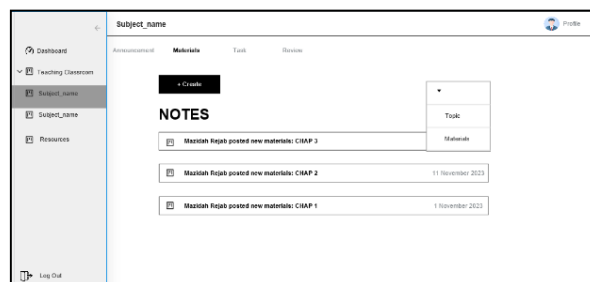


Fig. 4 Teacher Learning Materials Management Page

The most important content in this system is learning materials as those materials are used for both teacher and student. As shown in **Figure 4**, teachers are able to create and add learning materials for their students. At the top menu of teacher's page, there is announcement, materials, task, and review menu. The teachers not only can add learning materials, but they also can add tasks, grade them, and make announcements to give some information to their students. Meanwhile, the students' learning materials page is a little bit differ from the teacher's page. In students page, there is only a list of materials for their subject and does not have a create button like in teacher's learning materials page. The students are able to click which material they want to view, whether Chapter 1 notes or others which make it easier for students to search the material they intended to.

4. Result and Discussion

In this section, the result of implementation of Educational Resources Website for SMK Ulu Tiram are shown. This includes the test cases result and the result from user acceptance testing.

4.1 Implementation

As shown in **Figure 5**, the code is written to receive a user data input which will be passed from HTML page and store it into respective variable. Then, a system will verify if the ID for user has already existed, system will display

appropriate message error and the user data will not be saved into database and registration failed. However, if there is no missing field form or duplicate ID, the data will be saved in the database. **Figure 6** shows the user interface of sign in page. Generally, the user interface only consists of an input box which takes the email and password input, and Login button. The Login button will execute the code as illustrated in **Figure 5**.

```
public function rules()
{
    return [
        'email' => ['required', 'string', 'email'],
        'password' => ['required', 'string'],
    ];
}

/**
 * Attempt to authenticate the request's credentials.
 *
 * @return void
 * @throws \Illuminate\Validation\ValidationException
 */
public function authenticate()
{
    $this->ensureIsNotRateLimited();

    if (! $this->attempt($this->only('email', 'password'), $this->boolean('remember'))) {
        RateLimiter::hit($this->throttleKey());

        throw ValidationException::withMessages([
            'email' => trans('auth.failed'),
        ]);
    }
}
```

Fig. 5 Login Account Source Code

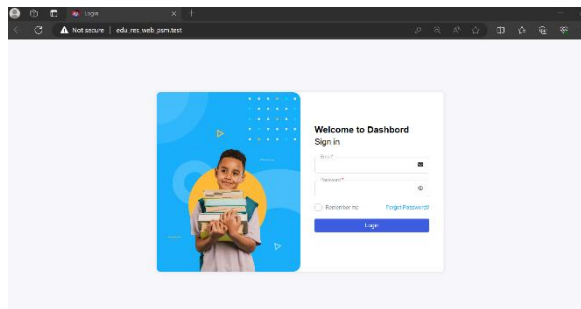


Fig. 6 Login User Interface

For Class and Subject module, **Figure 8** shows the user interface of Class and Subject module for teacher view. For this module, the user interface shows 4 sections of this module which is announcement, materials, task and review. In this module, the teacher is able to upload the materials and tasks including monitoring students learning comprehension. The code segment for creating new Class and assigned subjects into it is shown in **Figure 7**.

```
public function classSave(Request $request)
{
    $request->validate([
        'class_id' => 'required|string|unique:class,class_id',
        'class_name' => 'required|string',
        'class_desc' => 'required|string',
        'class_teacher' => 'required|integer',
        'sub_id' => 'required|array',
        'sub_id.*' => 'integer',
        'teacher_id' => 'required|array',
        'teacher_id.*' => 'integer',
    ]);

    try {
        DB::beginTransaction(); // Start the transaction

        // Create a new class
        $class = Classroom::create([
            'class_id' => $request->class_id,
            'class_name' => $request->class_name,
            'class_desc' => $request->class_desc,
        ]);

        // Save subjects for the class in the pivot table
        foreach ($request->sub_id as $index => $subId) {
            $class->subjects()->attach($subId, ['teacher_id' => $request->teacher_id[$index]]);
        }

        // Save classroom teacher for the class
        $class->teachers()->sync([$request->class_teacher => ['teacher_type' => 'Classroom Teacher']]);

        DB::commit(); // Commit the transaction

        Toastr::success('Class has been added successfully :)', 'Success');
        return redirect()->route('class/list/page');
    } catch (\Exception $e) {
        DB::rollback(); // Rollback the transaction in case of an exception
        Toastr::error('Failed to add a new class :(' . $e->getMessage(), 'Error');
        return redirect()->back();
    }
}
```

Fig. 7 The code segment for creating new Class and assigned subjects

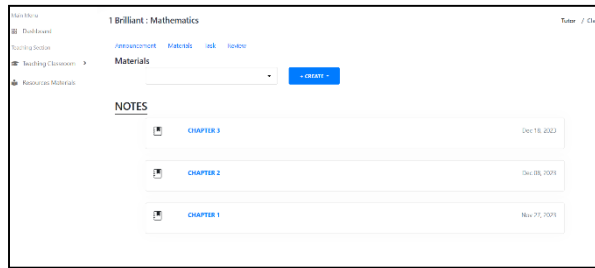


Fig. 8 User interface of Class and Subject module

For Resources Repository module, **Figure 10** shows the user interface of resources repository module. These interface shows, administrator enable to add the resource and able to manage the resources. However, if the resources are not uploaded by them, they are only able to view and download. The code segment for creating new past year question (PYQ) resources is shown in **Figure 9**.

```
public function uploadPq(Request $request)
{
    if (auth()-user()-role_name === 'Admin' || auth()-user()-role_name === 'Teachers') {
        // Validate the incoming request
        $validatedData = $request->validate([
            'file' => 'required|image|dox,docx,ppt,pptx,csv,txt,xls,xlsx,pdf|max:9120',
            'res_name' => 'required|string',
            'form' => 'required|string',
            'type' => 'required|string',
        ]);
        try {
            DB::beginTransaction(); // Start the transaction
            // Handle file upload
            $file = $request->file('file'); // Get the uploaded file
            $filename = time() . '-' . $file->getClientOriginalName(); // Generate a unique filename
            $filePath = $file->store('uploads', $filename, 'public'); // Store the file in the 'public/uploads' directory
            // Create a new resource material within the transaction
            $resource = new ResourceMaterial();
            $resource->res_name = $request->res_name;
            $resource->form = $request->form;
            $resource->file_name = $filePath; // Store the file path in the database
            $resource->res_type = $request->type; // Set the type
            $resource->uploaded_by = auth()-user()-id;
            $resource->save(); // Save the resource
            DB::commit(); // Commit the transaction
            // Redirect to the pqy_admin page after successful upload
            Toastr::success('Resource has been added successfully.!', 'Success');
            return redirect()-route('resources_a/pqy/page');
        } catch (Exception $e) {
            DB::rollback(); // Rollback the transaction in case of an exception
        }
    }
}
```

Fig. 9 Save PYQ Resource function for administrator

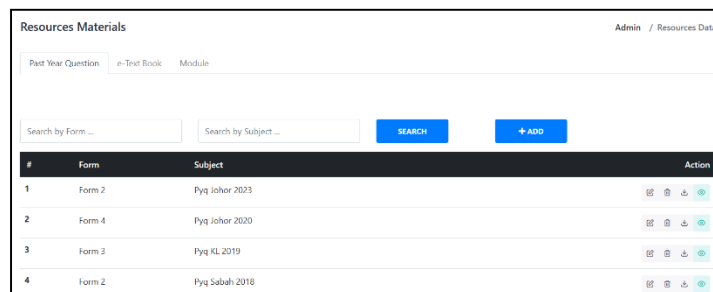


Fig. 10 User Interface of Resource Repository module for Admin view

For Bookmarks module, **Figure 12** shows the user interface of bookmark page. This interface shows students are enabled to add the resource that they intended into the bookmark and able to manage the bookmark. The code segment for storing the resources into the bookmark is shown in **Figure 11**.

```
// Save file resource into bookmark database
public function store(Request $request)
{
    log::info('Bookmark store method called');
    log::info('Resource ID: ' . $request->resource_id);
    // Begin a database transaction
    DB::beginTransaction();
    try {
        // Validate the request data
        $request->validate([
            'resource_id' => 'required|exists:resources_materials,res_id', // Ensure the resource exists
        ]);
        // Check if the resource is already bookmarked by the user
        $bookmarkExists = Bookmark::where('stud_id', Auth::id())
            ->where('resource_id', $request->resource_id)
            ->exists();
        if (!$bookmarkExists) {
            // If the resource is not already bookmarked, create a new bookmark entry
            $bookmark = create([
                'stud_id' => Auth::id(),
                'resource_id' => $request->resource_id,
            ]);
            // Commit the transaction
            DB::commit();
            return response()-json(['message' => 'Resource has been added successfully.!', 'status' => 'success']);
        } else {
            return response()-json(['message' => 'Resource already bookmarked!', 'status' => 'warning!']);
        }
    } catch (Exception $e) {
        // Rollback the transaction in case of an exception
        DB::rollback();
    }
}
```

Fig. 11 Store function in Bookmark Controller

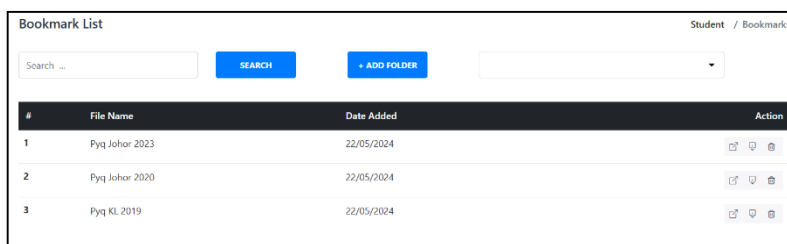


Fig. 12 Bookmark User Interface

For the Reporting module, **Figure 14** shows the user interface of reporting page from administrator view. This interface shows administrator are enabled to generate the annual students' performance report. While as shown in **Figure 16**, the class teacher is only able to generate the students' performance report for each student in their class. The code segment for generating the report is shown in **Figure 13 and 15**.

```
public function index()
{
    // Calculate class averages
    $classAverages = StudentDummy::select('student_dummies.class', 'subject_dummies.name as subject')
    ->selectRaw('AVG(carry_mark_dummies.carry_mark) as average_carry_mark')
    ->join('carry_mark_dummies', 'student_dummies.id', '=', 'carry_mark_dummies.student_id')
    ->join('subject_dummies', 'carry_mark_dummies.subject_id', '=', 'subject_dummies.id')
    ->groupBy('student_dummies.class', 'subject_dummies.name')
    ->get();
    ->groupBy('class');

    // Calculate form averages
    $formAverages = StudentDummy::selectRaw("SUBSTRING_INDEX(form, '-', 1) as form_number")
    ->selectRaw('subject_dummies.name as subject, AVG(carry_mark_dummies.carry_mark) as average_carry_mark')
    ->join('carry_mark_dummies', 'student_dummies.id', '=', 'carry_mark_dummies.student_id')
    ->join('subject_dummies', 'carry_mark_dummies.subject_id', '=', 'subject_dummies.id')
    ->groupBy('form_number', 'subject_dummies.name')
    ->get();
    ->groupBy('form_number');

    // Calculate subject averages
    $subjectAverages = SubjectDummy::select('subject_dummies.name')
    ->selectRaw('AVG(carry_mark_dummies.carry_mark) as average_carry_mark')
    ->join('carry_mark_dummies', 'subject_dummies.id', '=', 'carry_mark_dummies.subject_id')
    ->groupBy('subject_dummies.id')
    ->get();

    return view('report.index', compact('classAverages', 'formAverages', 'subjectAverages'));
}
```

Fig. 13 Annual report function controller code

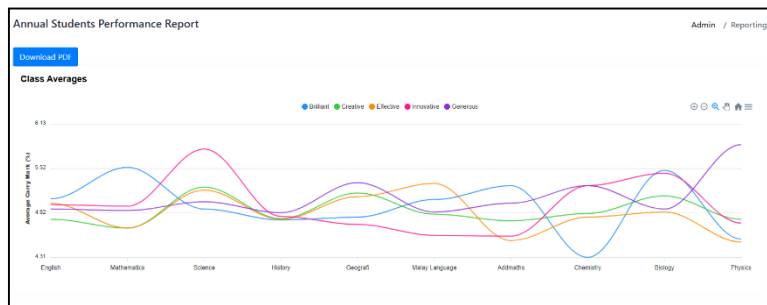


Fig. 14 Annual student performance report

```
public function generateStudentReport($studentId)
{
    // Retrieve the student with their marks
    $student = Student::with('marks')->find($studentId);

    if (!$student) {
        return response()->json(['message' => 'No student found with ID ' . $studentId], 404);
    }

    $report = "Performance Report for " . $student->name . " (ID: " . $student->id . "):\n";
    $report .= str_repeat("-", 50) . "\n";

    $totalMarks = 0;
    $subjectCount = $student->marks->count();

    foreach ($student->marks as $mark) {
        $report .= $mark->subject . ": " . $mark->marks . "\n";
        $totalMarks += $mark->marks;
    }

    $averageMarks = $subjectCount ? $totalMarks / $subjectCount : 0;
    $report .= str_repeat("-", 50) . "\n";
    $report .= "Total Marks: " . $totalMarks . "\n";
    $report .= "Average Marks: " . number_format($averageMarks, 2) . "\n";

    return response()->json(['report' => $report], 200);
}
```

Fig. 15 Student report function in controller

Student Performance Report

Name: Nur Amirah Syafiqah Bt Rosli **ID:** STU 102

Class: 1 Brilliant

Class Teacher: Lee Lee Jye

Subject	Mark
Mathematics	7.86
Science	8.88
English	5.76
History	6.70
Malay Language	7.30
Geografi	8.88

Date : 05/06/2024 10:10 am

Fig. 16 Student's performance report interface

4.2 System Testing

In this section, a test will be carried out to assess the functionality of each module. A User Acceptance Test (UAT) method is utilized to perform testing.

4.2.1 Functional Testing

On the developed modules, functional testing is carried out. The purpose of this testing is to make sure the developed system functions exactly as the user needs. Test cases provide instructions on how to run tests on a program, system, or application. A test case is a distinct set of steps or directives that a tester must adhere to verify the functionality of a particular feature of a product or application.

Table 7 Test Cases

No	Test Cases	Expected Outcome	Actual Outcome	Status
USER REGISTRATION AND AUTHENTICATIONMODULE				
1.	To check whether administrator can successfully register a new teacher and student account with valid details.	The administrator should be able to register account for teacher and student.	As expected	Pass
2.	Attempt to register a new account for student and teacher with an ID that already exists in the system.	The system should display appropriate error.	As expected	Pass
3.	Test registration with empty or invalid input for required fields (Id, password, email, etc).	The system should display appropriate error for following field form.	As expected	Pass
4.	To test whether teachers and students can log in successfully with their registered credentials.	Teachers and students should be able to login to their account.	As expected	Pass
5.	To test the authenticated users are granted appropriate access privileges based on their roles.	The authenticated users should be granted appropriate access privileges based on their roles (admin, teacher, student).	As expected	Pass
6.	Test login attempts with invalid usernames/email addresses and passwords.	The system should display appropriate error messages if the credentials to login are invalid.	As expected	Pass
CLASS AND SUBJECT MODULE				
1.	To check whether administrator can create new classes, assigned subject and teacher for each subject.	Administrator should be able to create new classes, assigned subject and teacher for each subject.	As expected	Pass
2.	To check whether the assigned class and subject appears in the teacher's and student account.	The class and subject that assigned to the teachers and students should be appear in their account.	As expected	Pass
3.	Test the ability of teachers to upload teaching materials (documents, presentations, videos, etc.) for specific classes.	Teachers should be able to upload teaching materials (documents, presentations, videos, etc.) for specific classes.	As expected	Pass

Table 7 (cont.)

No	Test Cases	Expected Outcome	Actual Outcome	Status
CLASS AND SUBJECT MODULE				
4.	Verify that teachers can view documents uploaded by students for assigned tasks or assignments.	Teachers should be able to view documents uploaded by students for assigned tasks or assignments.	As expected	Pass
5.	Test the grading functionality to ensure that teachers can assign grades or scores to student-submitted tasks or assignments.	The system should be able to display the grades that assigned by the teacher to the student submitted task or assignments.	As expected	Pass
6.	Verify that students can view and download documents uploaded by their teachers for each subject they are assigned in.	Students should be able to view and download documents uploaded by their subject teacher.	As expected	Pass
7.	To test that learning materials are accessible only to students enrolled in the respective class and subject.	Only the students that assigned in the respective class and subject should be able to access the learning materials.	As expected	Pass
8.	Test the ability of students to upload tasks or assignments given by their teachers.	Students should be able to upload tasks or assignments given by their teachers.	As expected	Pass
9.	To test that students can view their grades or scores for tasks or assignments graded by their teachers.	Students should be able to view their grades for tasks or assignments graded by their subject teachers.	As expected	Pass
RESOURCES REPOSITORY MODULE				
1.	Ensure that only administrator and teachers have permissions to upload, view, and delete documents.	The system should allow only the respective user have permissions to upload, view, and delete documents.	As expected	Pass
2.	Test document preview functionality to ensure that documents can be viewed directly within the application.	The documents should be viewed directly within the application.	As expected	Pass
3.	Test the ability of students to download documents from the repository.	Student should be able to download documents from the repository.	As expected	Pass
4.	Verify that downloaded documents are saved correctly and can be accessed offline.	The downloaded documents should save correctly and can be accessed offline.	As expected	Pass
5.	Verify that students can bookmark documents for easy access.	The students should be able to bookmark documents.	As expected	Pass
6.	Ensure that students do not have permissions to upload, edit, or delete documents in the repository.	The students should not have permissions to upload, edit, or delete documents in the repository.	As expected	Pass

Table 7 (cont.)

No	Test Cases	Expected Outcome	Actual Outcome	Status
BOOKMARK MODULE				
1.	Test the ability of student to insert documents into their bookmarks directly from the repository interface.	Student should be able to insert documents into their bookmarks directly from the repository interface.	As expected	Pass
2.	Test folder creation with various names and ensure that folders are created successfully.	Student should be able to create folder.	As expected	Pass
3.	Test document preview functionality to ensure that documents can be viewed directly within the application.	Student should be able to view directly within the application.	As expected	Pass
4.	Verify that student can download documents from their bookmarks.	The student should be able to download documents from their bookmarks.	As expected	Pass
5.	Verify that students can remove documents from their bookmarks.	The student should be able to remove documents from their bookmarks.	As expected	Pass
REPORTING MODULE				
1.	Verify that administrator can generate a performance report for all students in the school.	Administrator should be able to generate a performance report for all students in the school.	As expected	Pass
2.	Test the printing functionality to ensure that administrator can print the student's annual performance report in a readable format.	Administrator should be able to print the student's annual performance report in a readable format.	As expected	Pass
3.	Verify that class teachers can generate performance reports for individual students in their class.	Class teachers should be able to generate performance reports for individual students in their class.	As expected	Pass
4.	Test the printing functionality to ensure that class teachers can print individual student performance reports.	The class teachers should be able to print individual student performance reports.	As expected	Pass

4.2.2 User Acceptance Testing

The expected user participates in the user acceptance testing via a form. On a scale of 1 to 5, which ranges from extremely dissatisfied to very satisfied, users can rate such aspects. The system is tested by 3 types of users, which is administrator, teacher and student. The total user satisfaction for every feature of the suggested system is displayed in **Figure 17**.

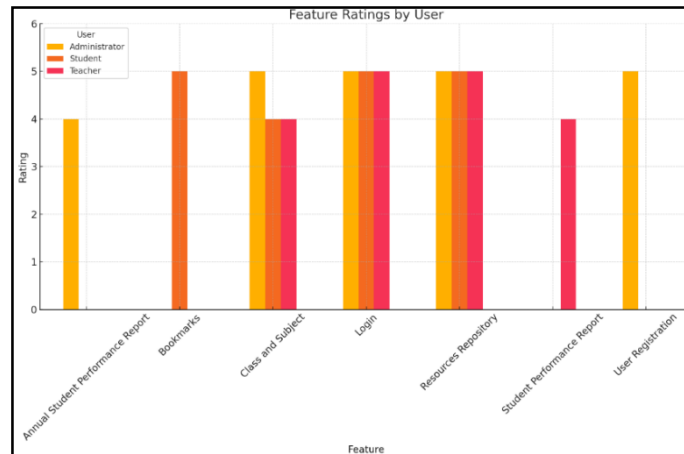


Fig. 17 Bar chart of user acceptance testing result

Based on **Figures 17**, this demonstrates how the functional requirements satisfy SMK Ulu Tiram's needs while also enhancing resource repository management. It shows 7 features that have satisfied and extremely satisfied the users.

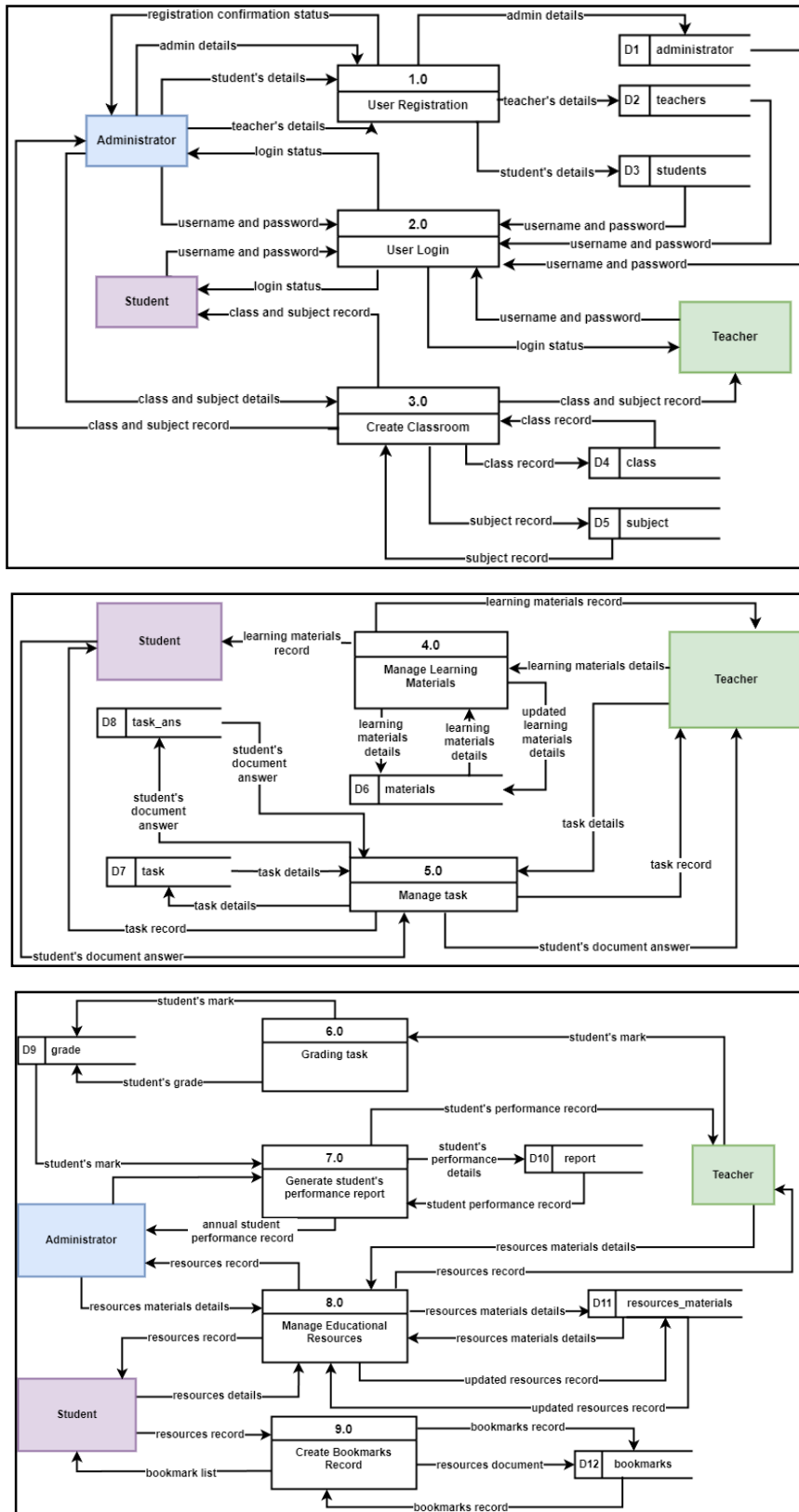
5. Conclusion

From overall discussion that has been made for this project, the main issue that faced by SMK Ulu Tiram regarding storage issue, the resources materials quality and management itself lead to the digital solution that were proposed which is developing an Educational Resources Website. The interaction between user and system are much clearer when illustrated by data flow diagram, entity relationship diagram and To-Be Model. Moreover, the data or known as attributes that placed in the ERD is to identify which data that required to be in system database. It makes much more sense when the user interfaces of Educational Resources Website for SMK Ulu Tiram are designed. From this design, the function or flow of user activities from the design page could give a big picture of what should be improved or modified in the design to align with the requirements and objective.

Acknowledgement

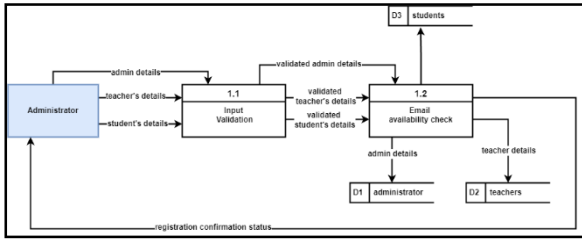
The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix A: DFD Level 0

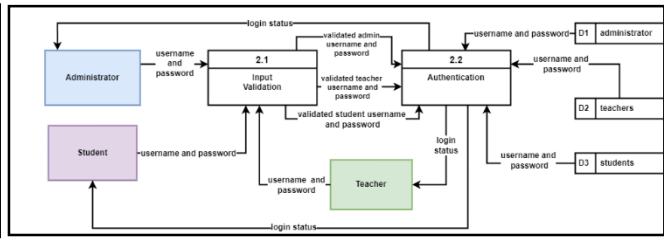


Data Flow Diagram Level 0

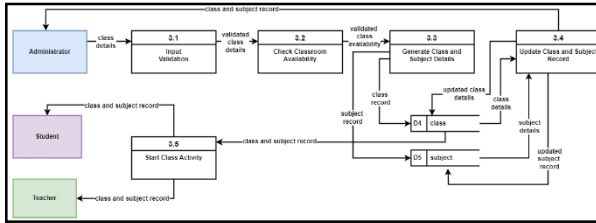
Appendix B: DFD Level 1



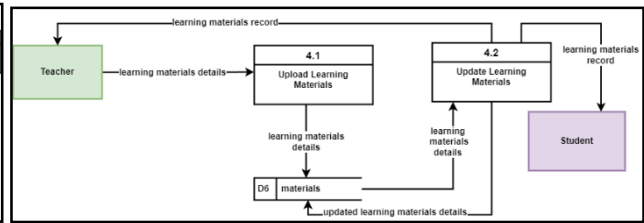
DFD Level 1 : Registration Process



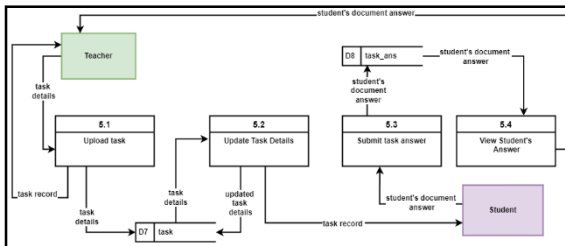
DFD Level 1 : Login Process



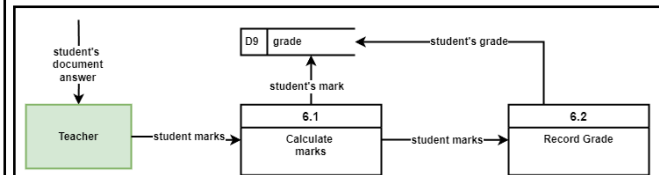
DFD Level 1 : Classroom Creation Process



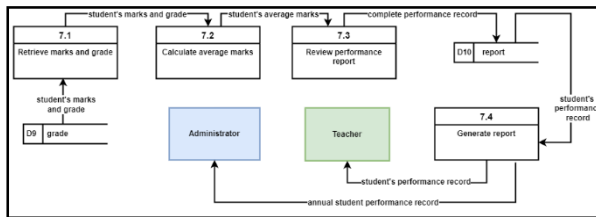
DFD Level 1 : Learning Materials Management Process



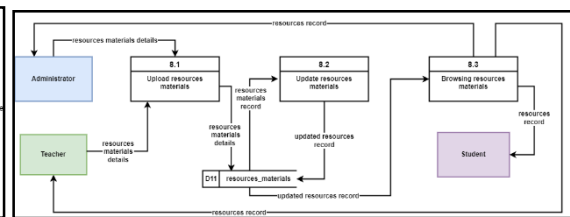
DFD Level 1 : Task Management Process



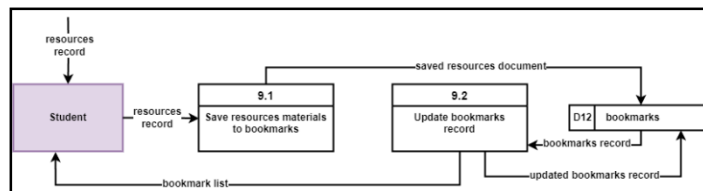
DFD Level 1 : Grading Task Process



DFD Level 1 : Report Generation Process



DFD Level 1 : Resources Management Process



DFD Level 1 : Bookmarks Management Process

References

- [1] P. Maclaren, "How is that done? Student views on resources used outside the engineering classroom," *Eur. J. Eng. Educ.*, vol. 43, no. 4, pp. 620–637, Jul. 2018.
- [2] Curino, C. A., Tanca, L., Moon, H. J., & Zaniolo, C. (2008). Schema evolution in wikipedia: toward a web information system benchmark. In *International Conference on Enterprise Information Systems (ICEIS)*.
- [3] Setyawati, E. (2021). Web-Based Management Information System for Services Development: A Literature Review. *International Journal of Current Science Research and Review*, 04(03). <https://doi.org/10.47191/ijcsrr/v4-i3-05>
- [4] Grady, J. O. (2010). *System requirements analysis*. Elsevier.
- [5] Cloud Data Centers and Cost Modeling. (2015). In *Elsevier eBooks*. <https://doi.org/10.1016/c2013-0-23202-5>
- [6] Bashar Nuseibeh, & Easterbrook, S. (2003). Requirements Engineering. *Elsevier eBooks*, 229–236. <https://doi.org/10.1016/b0-12-227410-5/00855-3>
- [7] Oshana, R. (2015). System Requirements. *Elsevier eBooks*, 159–188. <https://doi.org/10.1016/b978-0-12-405879-8.00006-4>
- [8] Blackledget, J. M. (2006). Principles of Software Engineering. *Elsevier eBooks*. <https://doi.org/10.1533/9780857099457.3.299>
- [9] Kraeling, M., & Lindsley Tania. (2019). *Software Development Process*, 33-87. <https://doi.org/10.1016/b978-0-12-809448-8.00002-3>