SK TANGGA BATU SPBT MANAGEMENT SYSTEM

NUR HANANI BINTI ABDUL WAHAB

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

SK TANGGA BATU SPBT MANAGEMENT SYSTEM

NUR HANANI BINTI ABDUL WAHAB

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Database Management)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

[SK TANGGA BATU SPBT MANAGEMENT SYSTEM]

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : _____ Date : 5 SEPT 2021

(NUR HANANI BINTI ABDUL WAHAB)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of [Computer Science (Database Management)] with Honours.

SUPERVISOR : _____ Date : 12/9/2021

(TS. DR. NORASHIKIN BINTI AHMAD)

DEDICATION

Thank you to everyone who has helped, encouraged, motivated, and cheered me up. And especially to my loving parents, lecturers, and friends for all their guidance, compassion, and care in helping me climb the success ladder.

My attractive and outstanding seniors also enlivened my childhood and added flavour to my existence.

ACKNOWLEDGEMENTS

I am very honored and fortunate with the inspiration and assistance from my supervisor, Dr Norashikin Binti Ahmad. Despite her busy schedule, she inspired and assisted me. Thank you for providing me with the opportunity to complete my Final Year Project.

Furthermore, I would like to express my gratitude to Ts. Abdul Razak Bin Hussain, my evaluator, for providing me with valuable recommendations and evaluating my work.

Nur Haslinda Binti Abdul Wahab and Nur Hidayah Binti Abdul Wahab deserve special thanks for consistently explaining the SPBT management system to me at school. Both are teachers with prior expertise managing SPBT in their respective schools.

It is with great pleasure that I express my heartfelt gratitude to my family for their unwavering support and encouragement throughout my studies.

Last but not least, many thanks to my lecturers, seniors, and all those who provided me with guidance that was both theoretically and practically beneficial to my studies.

ABSTRACT

The Textbook Loan Scheme Program (SPBT) was implemented in 1975. SPBT is a textbook loan program for eligible students in all government schools and government aid schools. The SPBT program was originally intended to alleviate the financial burden incurred by poor parents or guardians and ensure that pupils from undeserved families not to drop their schooling. At 2008, the Malaysian government has provided textbooks loans to all students who attend school at government schools regardless of their family's financial status. Hence, this situation makes the responsibility of the SPBT unit be more weighted to ensure that each book is processed and distributed to all students. Teachers, who incharge for SPBT are having troubles to organize the information of textbooks borrowed by students. Due to time limitation, teachers can only tick at the students' name list without record the series number of textbooks. The purpose of this system is to provide easy ways to handle the SPBT management process that can help the teachers to standardize the management and implementation of SPBT. This project is useful to save user's time in management of SPBT and prevent data loss. As for the conclusion, SPBT Management System will be a good solution in solving real life problems and makes the user life easier and simpler.

ABSTRAK

Program Skim Pinjaman Buku Teks (SPBT) dilaksanakan pada tahun 1975. SPBT adalah program pinjaman buku teks untuk pelajar yang layak di semua sekolah kerajaan dan sekolah bantuan kerajaan. Program SPBT pada asalnya bertujuan untuk meringankan beban kewangan yang ditanggung oleh ibu bapa atau penjaga yang lemah dan memastikan bahawa murid-murid dari keluarga yang tidak layak tidak berhenti sekolah. Pada tahun 2008, kerajaan Malaysia telah memberikan pinjaman buku teks kepada semua pelajar yang bersekolah di sekolah kerajaan tanpa mengira status kewangan keluarga mereka. Oleh itu, keadaan ini menjadikan tanggungjawab unit SPBT lebih ditimbang untuk memastikan setiap buku diproses dan diedarkan kepada semua pelajar. Guru-guru, yang mendapat kenaikan SPBT menghadapi masalah untuk mengatur maklumat buku teks yang dipinjam oleh pelajar. Oleh kerana had masa, guru hanya boleh mencatat senarai nama pelajar tanpa mencatatkan bilangan siri buku teks. Tujuan sistem ini adalah untuk menyediakan cara mudah untuk menangani proses pengurusan SPBT yang dapat membantu para guru untuk menyeragamkan pengurusan dan pelaksanaan SPBT. Projek ini berguna untuk menjimatkan masa pengguna dalam pengurusan SPBT dan mencegah kehilangan data. Sebagai kesimpulan, Sistem Pengurusan SPBT akan menjadi penyelesaian yang baik dalam menyelesaikan masalah kehidupan sebenar dan menjadikan kehidupan pengguna lebih mudah dan sederhana.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

CIA Triad	-	Confidentiality, Integrity and Availability
CSV file	-	Comma Separated Values File
DBLC	-	Database Life Cycle
DBMS	-	Database Management System
DDL	-	Data Definition Language
DFD	-	Data Flow Diagram
DML	-	Data Manipulation Language
ERD	-	Entity Relationship Diagram
FYP	-	Final Year Project
ID	-	Identification/Identity/Identifier
IDE	-	Integrated Development Environment
PDF	-	Portable Document Format
QR code	-	Quick Response code
RDBMS	-	Relational Database Management System
SDLC	-	System Development Life Cycle
SPBT	-	Skim Pinjaman Buku Teks
SQL	-	Structured Query Language
UAT	-	User Acceptance Testing

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

In 1975, the Textbook Loan Scheme Program (SPBT) was established. SPBT is a textbook loan scheme for students in government and government-aided institutions that qualify. The SPBT programme was created to relieve poor parents' or guardians' financial burdens and ensure that students from deserving homes did not drop out of school. The Malaysian government began lending textbooks to all kids who attend government schools in 2008, regardless of their family's socioeconomic situation. As a result, the SPBT unit's obligation to ensure that each book is processed and handed to all pupils at SK Tangga Batu is increased.

Based on my observation and interview from Puan Nur Haslinda Binti Abdul Wahab, the SPBT coordinator teacher at SK Tangga Batu, that school was still using a manual way for the process of borrowing textbooks from students. Teachers, who incharge for SPBT are having troubles to organize the information of textbooks borrowed by students. Due to pandemic Covid-19, primary school's students was not allowed attend to school. Therefore, the parents must attend to school to get their children's textbooks according to the time given by schools. Due to time limitation, teachers can only tick at the students' name list without record the series number of textbooks. To avoid this, we've developed a system that can handle the SPBT management procedure in a simple manner. This suggested approach would assist instructors in standardising the management and implementation of SPBT in SK Tangga Batu while maintaining a focus on excellence and parental involvement in the management and implementation of SPBT. This project will save users time and

prevent data loss when managing SPBT. Finally, the SPBT Management System will be a good solution for real-world problems and will make the user's life easier and simpler.

1.2 Problem Statement(s)

There are some problems faced by the teachers who in charge for SPBT who intended to manage the textbook. Currently, due to Covid-19, the teachers distribute textbooks to students through parents. They just tick at the students' list name without recording the series number of book taken due to time constraint. So, teachers do not have the information with them for the time being. Moreover, the teachers find it difficult to keep the data for hundreds of students and it's just very time consuming to keep and retrieve all the data manually. In certain cases, they might lost the papers. Lastly, sometimes the textbooks are damage or lost but teachers cannot identify which student should responsible for that textbooks. It is hard for teacher to check one by one manually in files.

1.3 Objective

The objectives of SK Tangga Batu SPBT Management System are:

i. To provide a platform where teacher SPBT can store, record and organize the information of textbooks more systematically.
ii. To develop a system for teacher which can help them to manage the information of students who borrowed the textbooks.
iii. To enable the teachers to track students who lost or damage the textbooks easily.

1.4 Scope

1.4.1 Scopes of the User

i. Admin
    - Able to log in as administrator
    - Able to manage teacher information

- Able to view all reports

ii. SPBT Teacher (The Coordinator and AJK of SPBT)

- Able to log in
- Able to manage students' information
- Able to manage the details of textbook
- Able to manage the details of perolehan textbook
- Able to manage fine record
- Able to manage the transaction of textbook
- Able to update profile and change password
- Able to view all the report
- Able to manage the publishing house information

iii. Class Teacher

- Able to log into management system
- Able to manage students' information
- Able to manage class for students

1.5 Project Significance

i. Motivation and inspiration for the project.

The aim to do this project is mainly to reduce the manual record burden to the teachers that in charge in SPBT. This system can also help to reduce the paperwork. Teachers can update the information of textbooks and manage all the processes involved in the school in the most effective ways. Hence, this system can help teachers to track students who lost or damage textbooks easily.

1.6 Expected Output

This system can store some information of teachers, students and textbooks. User can keep track the transaction of textbooks. When school is opening during New Year, class teacher can insert the information of students by using this system. So, SPBT teachers can access the information at any time. Moreover, the system will help teachers to track students who lost or damage the textbooks easier by looking at

students' information. The system supports reporting by summarizing the transaction records of textbooks.

1.7 Conclusion

In conclusion, this chapter provides background information on SPBT, a textbook lending programme available to qualifying students in all government and government-aided institutions. The chapter outlines the shortcomings of the current SPBT management process in schools, as well as the goals of the SPBT Management System, which is built specifically for everyday SPBT management operations. This chapter also reveals the project's scope, impact, and expected outcomes. The project planning and methods are discussed in the following chapter.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

The approach utilised to construct this SPBT Management System will be detailed in this chapter. As a result, the System Development Life Cycle (SDLC), an Interactive and Incremental Model, will be employed for each phase in the SPBT Management System. It will go over the specifics of each phase of the project development process based on the system requirements.

2.2 Project Methodology

The database life cycle (DBLC) phase of the project is described in this chapter. Database initial study, requirement gathering and analysis, database design, application design, implementation and loading, testing and evaluation, and operational maintenance are the phases of the database life cycle.

i. Database Initial Study

This phase establishes the project's mission and goals. The mission statement explains the primary goals of the database application, whereas the objectives define a specific activity that the database can help with. The scope and boundaries of the database project are then defined. We also gather information on how the desired system is supposed to work.

ii. Requirement Collection and Analysis

This phase entails gathering and evaluating data related to the organization's component that will be supported by the database system, as well as using that data to

identify users' needs that will be addressed by the system. An organization's informational requirements are analysed so that a database can be created to meet those requirements. Conversations with Nur Haslinda (teacher) are held on a regular basis in order to gain additional domain expertise in the education industry. (For further information on this person, see ‘ACKNOWLEDGEMENTS.')

iii. Database Design

Database design is divided into three stages. Conceptual database design, logical database design, and physical database design are the three types of database design. The goal of conceptual database design is to determine the database's conceptual representation, which includes the identification of important entities, relationships, and properties. The logical database design transforms the conceptual representation into the database's logical structure, which includes the design of the relationships. Physical database design determines how to physically implement the logical structure, which is tailored to a specific DBMS. The database management system chosen for this project is MySQL Workbench 8.0 CE.

iv. Application Design

Database and application design are carried out in tandem. The user interface and application programmes that use and process the database are designed in this phase. It consists of two main components: transaction design and user interface design. The transaction design identifies the operation performed by a user or application software that accesses or modifies database content. The creation of an interface layout that interacts with the user to facilitate human-computer interaction is known as user interface design.

v. Implementation and Loading

The physical manifestation of the database and application design takes place in this phase. To create a database in a database management system, use DDL and DML. The data must be loaded into the database tables after the database has been created. A programming language is used to create the application.

vi. Testing and Evaluation

The process of operating the database system with the goal of finding flaws is known as testing and evaluation. The database is put through its paces to ensure data integrity and security. This phase verifies that the database and application programmes are operating in compliance with the specifications. The system must be evaluated based on software quality dimension as the database and application programmes are built and tested. Nur Haslinda participates in User Acceptance Testing (UAT) to ensure that the programme meets the business needs. In Chapter 6, the intricacies of testing are detailed.

vii. Operational Maintenance

The database is considered operational once it has passed the evaluation stage. The database, its management, its users, and its application programmes are all part of a full information system at that point. The database should be maintained on a regular basis. Preventive and corrective maintenance (backup and recovery), adaptive maintenance, and the issuance of access permissions for old and new users are only a few of the essential periodic maintenance operations.

2.3 Project Schedule and Milestone

The project planning timeline may need to be updated as requirements become more clearly stated. Any changes will be discussed with the supervisor at project status conferences.

Table 2.1 shows FYP 1 and FYP 2 Milestones while project timeline or Gantt chart illustrating the milestones and the project tasks can be referred to Figure 2.1.

ID	Task Name	Start	Finish
1	Preplanning phase -Project identification	Mon 15/3/21	Sun 11/4/21
2	Project planning phase -Project methodology -Construct Gantt chart	Mon 12/4/21	Sun 25/4/21
3	Analysis phase -Problem analysis	Mon 26/4/21	Sun 2/5/21
4	Analysis phase -Requirement elicitation	Mon 3/5/21	Sun 9/5/21
5	Design phase -Database design -Input, output, interface design	Mon 17/5/21	Sun 23/5/21
6	Implementation phase -Coding	Mon 24/5/21	Sun 20/6/21
7	Testing phase -Test case and test data -Execute and record test result	Mon 19/7/21	Sun 15/8/21
8	Project closure -Completing deliverables	Mon 16/8/21	Fri 10/9/21

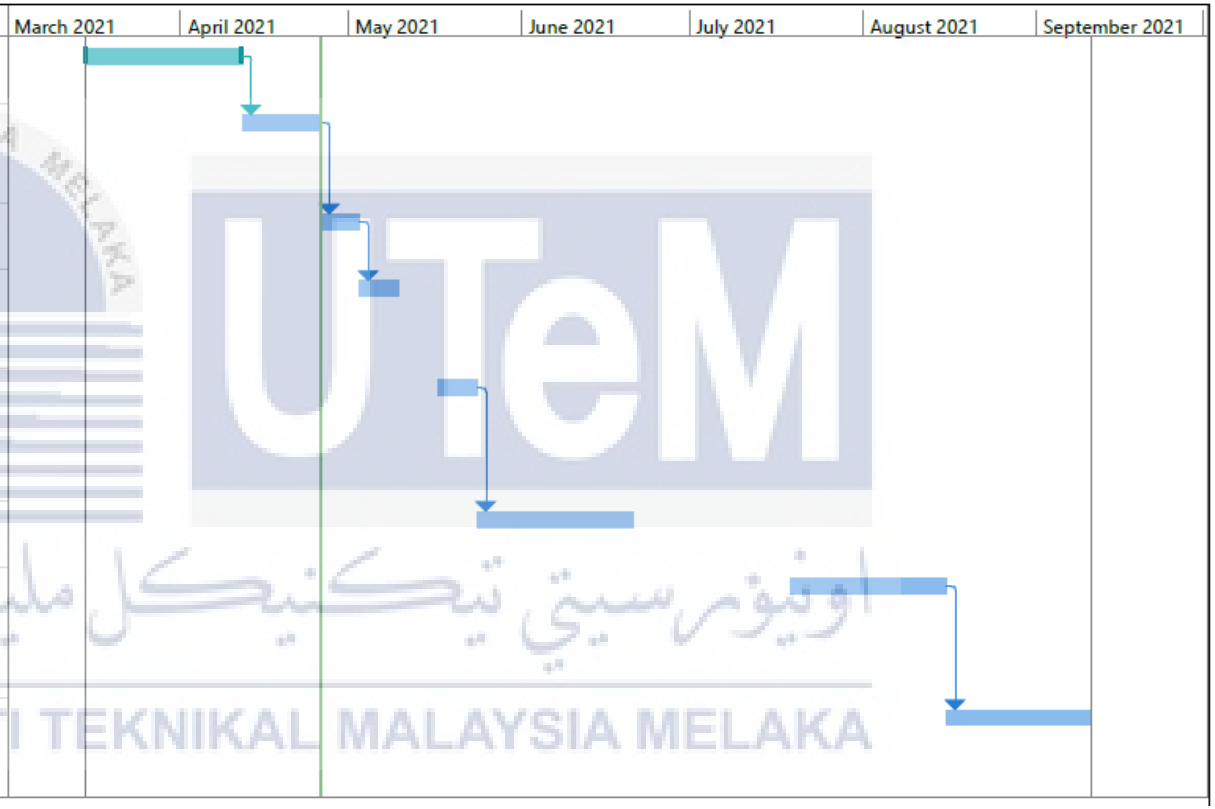


Figure 2.1: Project Timeline (Gantt Chart

Table 2.1: FYP 1 & FYP 2 Milestones

Final Year Project 1 (Semester 2 2020/2021)		
Target Week	**Start Date**	**Deliverable**
Week 1	15 Mac 2021	Proposal
Week 3	29 Mac 2021	Proposal Presentation (PP)
Week 4	5 April 2021	Chapter 1
Week 6	19 April 2021	Chapter 2 Progress Presentation 1
Week 8	3 May 2021	Chapter 3
Week 10	17 May 2021	Progress Presentation 2
Week 11	24 May 2021	Project Demonstration
Week 14	14 Jun 2021	FYP 1 Draft Report
Week 15	21 Jun 2021	Presentation
Week 16	28 Jun 2021	FYP 1 Logbooks
Final Year Project 2 (Semester 3 2020/2021)		
Target Week	**Start Date**	**Deliverable**
Week 1	19 July	Chapter 4
Week 2	26 July	Progress Presentation 1
Week 3	2 August	Chapter 5
Week 4	9 August	Progress Presentation 2
Week 5	16 August	Chapter 6
Week 6	23 August	FYP 2 Draft Report
Week 7	30 August	Presentation
Week 8	6 September	FYP 2 Logbooks
Week 9	13 September	Final FYP Report

2.4 Conclusion

In conclusion, this chapter concludes by defining project methodology and planning. Project methodology refers to the project's database life cycle (DBLC) phase. In project planning, project milestones are listed. Project planning also includes a project timeline or Gantt chart that illustrates the milestones and project tasks. The

project analysis phase, which examines the problem and requirements, is the subject of the next chapter.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis step comprises a thorough examination of the database development strategy that was devised ahead of time. In order to examine the system's performance, more than one developer (or a team of developers) evaluates the database development plan against elements including cost, timescale, development platform, programming languages, and planned development outcomes. This phase entails examining the current system as well as the system that will be constructed.

3.2 Problem Analysis

On 15 Mac 2021, a phone call is held with Nur Haslinda to better understand the normal business process flow in her school for SPBT management.The flowchart is illustrated at Figure 3.1 and Figure 3.2.

During the opening of schools, teachers who in charge for Skim Pinjaman Buku Teks (SPBT) will check the stock of textbooks and check the number of enrollment of students for that year. If the textbooks is not enough, the coordinator of SPBT need to add additional textbooks. Each of the additional textbooks will be given the unique series number and the textbooks information will be keep in files. The disposal of textbooks is only for textbook that can no longer be repaired or no longer used.

The parents will get the textbooks and a form for them to fill the series number for each of the textbooks. They need to return the form along with the textbooks at end of school. During the returning of textbooks, the AJK of SBPT will check the condition

of textbooks. The students have to pay fine or make a replacement for the textbooks if any damage or loss occur.

Several steps in this business process are completed manually. Countless sheets of paper have been used to record data. Physical storage is required for thick materials grouped in folders. Physical papers provide a security and damage risk, resulting in data loss.

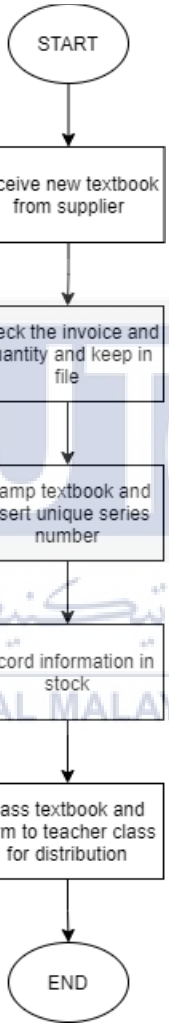


Figure 3.1: Textbook Business Flow

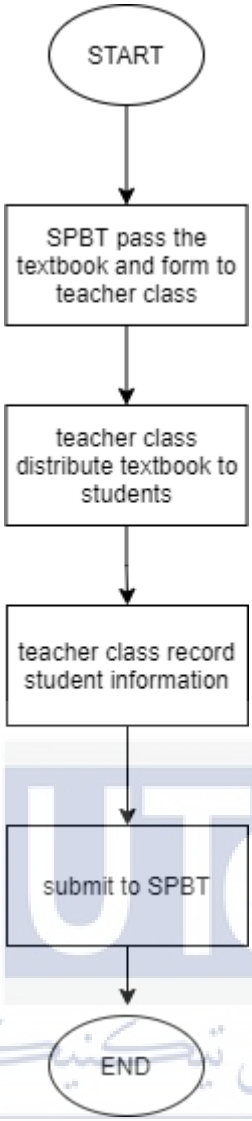


Figure 3.2: Borrow Textbook Business Flow

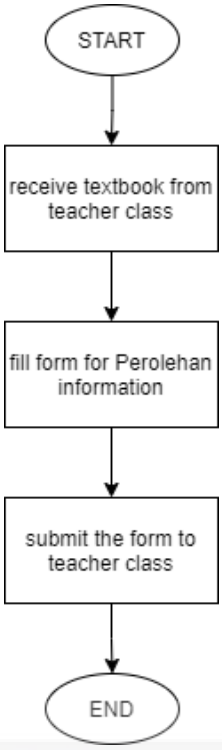


Figure 3.3: Student Receive Textbook Business Flow

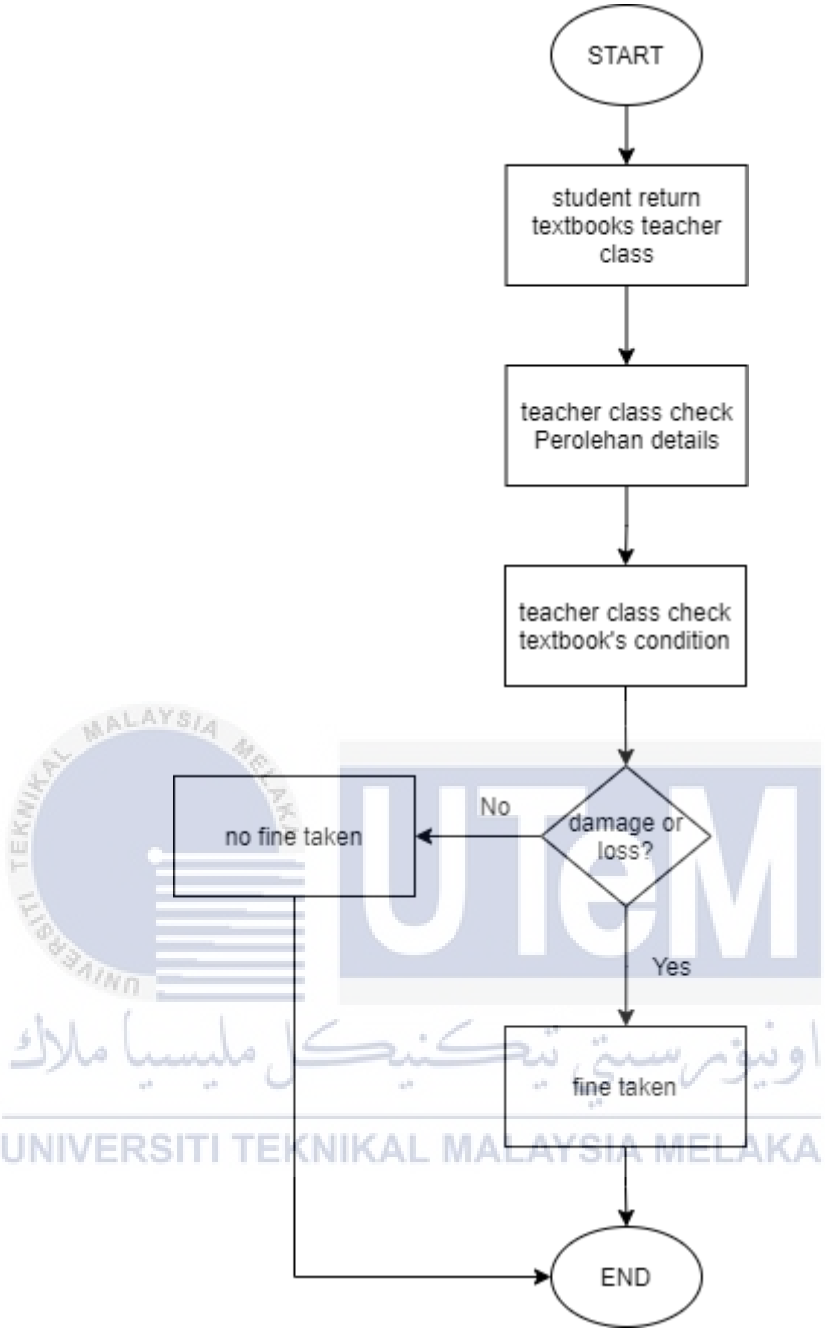


Figure 3.4: Return Textbook Business Flow

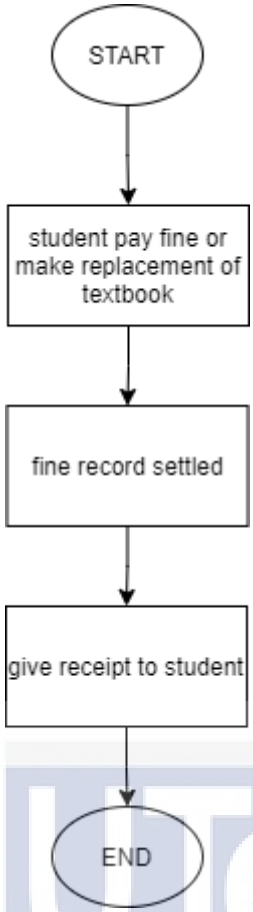


Figure 3.5: Fine Payment Business Flow

3.3 The Proposed Improvements / Solutions

A new system is proposed to improve some of the present system's features. The procedure has been digitised in the hopes of streamlining the business process.

The figures show how the system interact with external entities and the data flow.

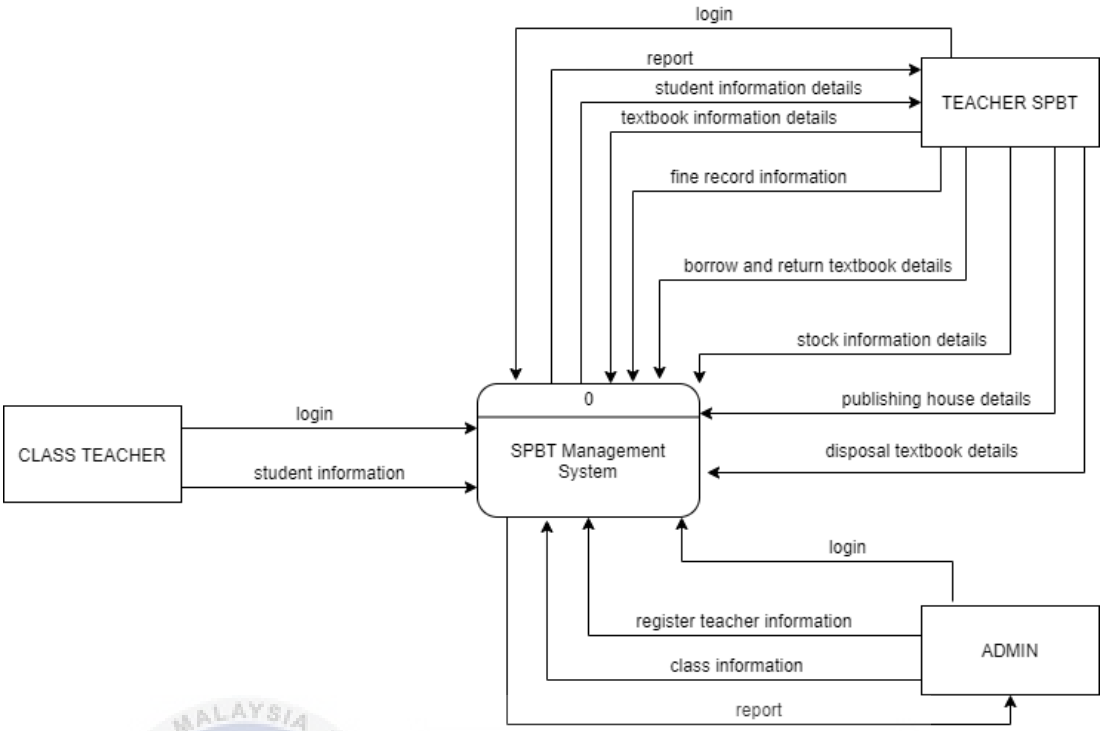


Figure 3.6: Context Diagram

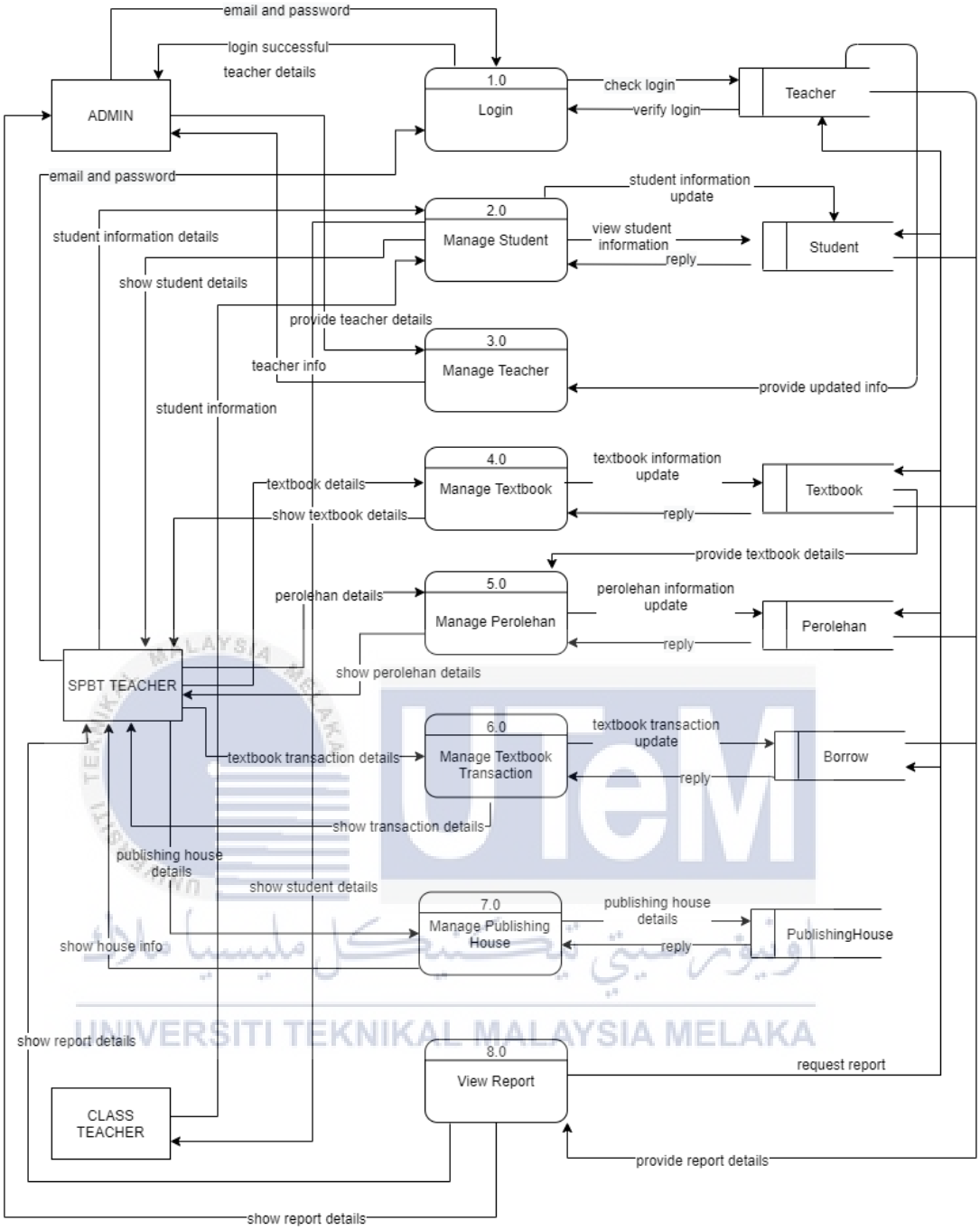


Figure 3.7: Data Flow Diagram Level 0

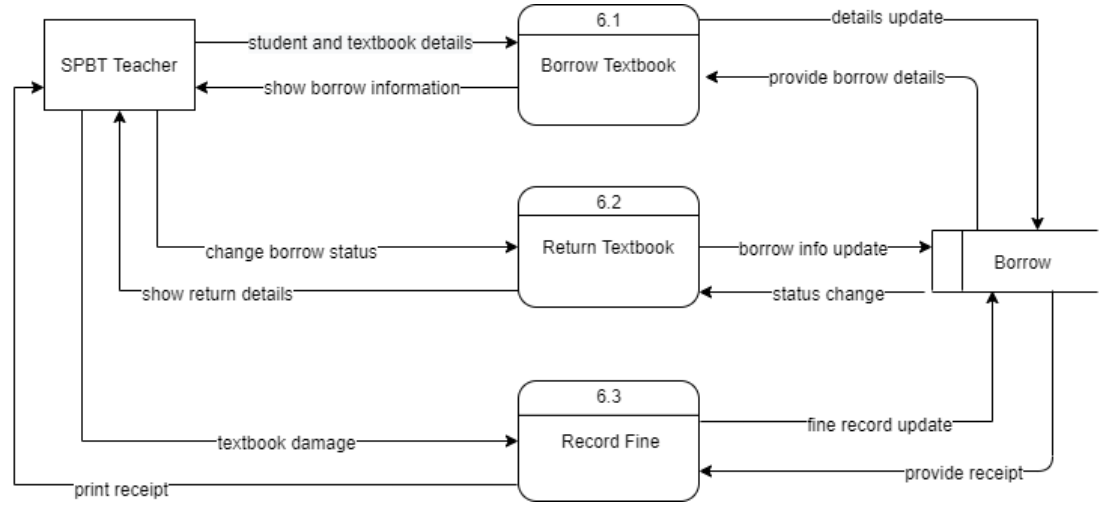


Figure 3.8: DFD Level 1 (Textbook Transaction)

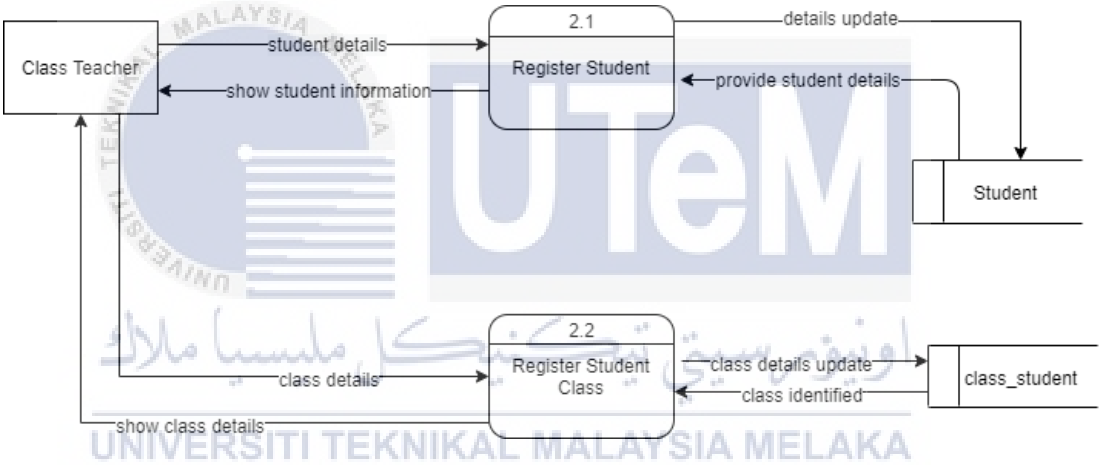


Figure 3.9: DFD Level 1 (Manage Student)

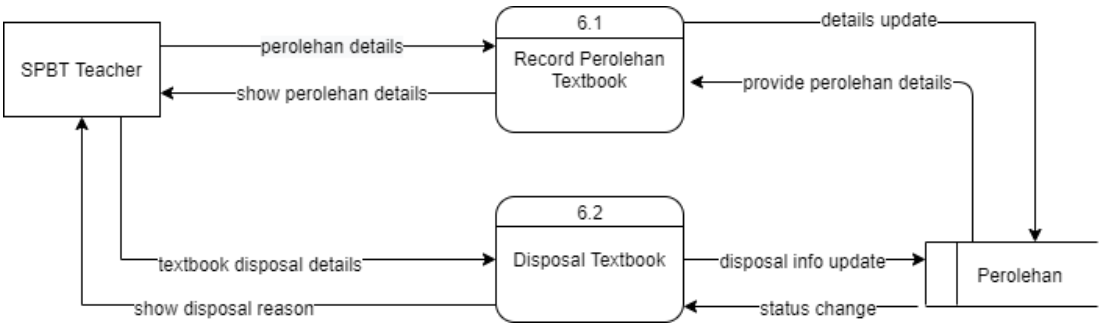


Figure 3.10: DFD Level 1 (Manage Perolehan)

3.4 Requirement Analysis of the To-Be System

3.4.1 Functional Requirement (Process Model)

Functional requirements are system capabilities that must be met in order for users' needs to be met. The system should be able to perform the tasks that have been asked.

Table 3.1 shows functional requirement of the system.

Table 3.1: Functional Requirement

Requirement	Description
Login and Logout	The system shall enable Admin, Teacher SPBT and Teacher Class to logon to the system by inserting valid login credentials, which are email and password.
Manage Student	The system should enable Admin, Teacher SPBT and Teacher Class to manage student's information, including insert and update student information to the database.
Manage Textbook	The system enable the Teacher SPBT to manage textbook information, including insert and update.
Manage Fine Record	The system shall enable to manage fine record such as broken or loss.
Manage Textbook Transaction	The system shall enable to manage textbook transaction from students such as borrow and return.
Manage Publishing House	The system shall enable to manage publishing house for each textbook received.

3.4.2 Non-Functional Requirement

Non-functional requirements are those that aren't directly related to the system's functionalities. It frequently refers to system properties in order to ensure system quality.

Table 3.2 shows non-functional requirement of the system.

Table 3.2: Non-Functional Requirement

Requirement	Description
Accessibility	Colorblind people should be able to use the system.
Authentication	Authentication should be provided by the system.
Availability	The system should be accessible 24 hours a day, seven days a week.
Integrity	Unauthorized people should not be able to change data in the system.
Correctness	The probability of receiving a non-accurate outcome should not exceed 5%.
Reliability	The system's failure rate should not exceed two times each month.
Maintainability	The programmer should adhere to code conventions and naming conventions. Take, for example, className.
Usability	After some training and supervision, a new user should be able to operate the system.
Security	Unauthorized access to sensitive data should be prevented by the system.

3.4.3 Others Requirement

3.4.3.1 Software Requirement

The software used in this are listed below.

i. **Documentation Tools**

- Microsoft Office 365 ProPlus

Underneath Office 365, there is an Office version called Office 365 ProPlus. Access, Excel, OneNote, Outlook, Powerpoint, Publisher, Skype for Business, and Word are all included in this product. Word is a word processor programme that is used for documentation.

- Draw.IO

Draw.IO Diagrams is a cross-platform diagramming application to build flowcharts, workflows, org charts, Venn diagrams, wireframes, network diagrams, and any other chart or modular design.

ii. Development Tools

- Microsoft Visual Studio Code

Microsoft Visual Studio is a developer's integrated development environment (IDE) powered by Microsoft. It contains a code editor with AI-assisted IntelliSense that provides syntax highlighting and code completion.

- MySQL Workbench 8.0 CE

Microsoft SQL Server is a relational database management system that was developed by the company Microsoft. This software product's primary function is to entertain the attraction of other software applications for data storage and retrieval.

- XAMPP

XAMPP is an open-source software that contains Apache distributions for Apache server, MariaDB, PHP, and Perl. And it is basically a local host or a local server. This local server works on your own desktop or laptop computer. The use of XAMPP is to test the clients or your website before uploading it to the remote web server. This XAMPP server software gives you the suitable environment for testing MYSQL, PHP, Apache and Perl projects on the local computer.

3.4.3.2 Hardware Requirement

The hardware used in developing the system is laptop. Table 3.3 listed the specification of hardware used.

Table 3.3: Hardware (Laptop) Specification

Component	Specification
Processor	Intel Core i3-5005U, 2.0GHz
Memory	4GB
System Type	64-bit Operating System, x64-based processor
Graphic Card	NVIDIA GeForce 920M

3.5 Conclusion

In conclusion, this chapter is critical in the development of a system that will be more efficient, long-lasting, and functional for the end user. Furthermore, the system can be readily set up because the developer will be better familiar with the system's flow following the analysis step. The present SPBT management method is depicted in a flow chart. Data flow diagram graphically represents the flow of data in the system to be constructed. The graphic represents the understanding of each unit's inputs and outputs, as well as the evolution. Finally, the new system that will be constructed will be able to improve the functionality and efficacy of the current system. The next chapter delves into the database's conceptual, logical, and physical design.

CHAPTER 4: DESIGN

4.1 Introduction

The creation of a thorough database data model is known as database design. Conceptual design, logical design, and physical design are the three phases.

The conceptual database design phase reveals how diverse elements are linked to one another. It also specifies the qualities that each entity possesses. The design counts the delineations of all application field conceptions such as entities, attributes. Business rules and an entity-relationship diagram (ERD) are deliverables that are developed using data from users.

The task of logical database design is to organise data into a series of logical relationships known as entities and attributes. A block of data is represented by an entity. An entity's individuality is defined by its attribute, which is a component within the entity's boundaries. The result of logical database architecture is a data dictionary.

The purpose of physical database design is to put the database into action. Translate the data gathered during the logical design step into a description of the physical database, including tables and constraints, during this phase. A DBMS is chosen, and a Data Definition Language is created.

4.2 Introductory Preview to This Chapter

This system was created on a Windows environment and runs on it. It has a MySQL Workbench database attached to it.

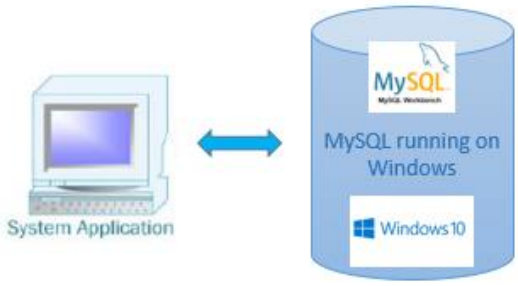


Figure 4.1: System Architecture

4.3 Database Design

4.3.1 Conceptual Design

4.3.1.1 Business Rules

- Teacher is the main user for the system. Teacher were categorized by role which are Admin, Class Teacher and Teacher SPBT.
- Class Teacher is the person who register student information. A class teacher can register one or many students. Student is identified by student ID, student name, birth certification, date of birth, gender, address, image, guardian name, guardian contact number and citizenship.
- Class is identified by class ID and class name. A class can has one or many class student member.
- A student is register only to one class in a year.
- Each class teacher can organize one or many student's information to get into their own specified class. The class student is identified by id, student ID, class ID, year and teacher ID.
- Textbook is the general information about all type of textbook. A teacher SPBT can register one or many textbooks. The textbook is identified by textbook ID, book title, category, book price, book code and book edition type.
- Each publishing house can have many textbook. Publishing house is identified by publishing house ID.

- Perolehan is another name for textbook copy. Teacher SPBT can register many perolehan textbook.
- Disposal is the record for disposal textbook. A disposal textbook can have many perolehan. Disposal is identified by disposal ID and reason.
- Borrow must have a borrower that can be teacher or student. Perolehan can be borrowed by many borrowers. Borrow is identified by borrow ID, student ID, teacher ID, date borrowed, status return, return date, fine description, fine payment, payment date and fine status.

4.3.1.2 Entity Relationship Diagram (ERD)

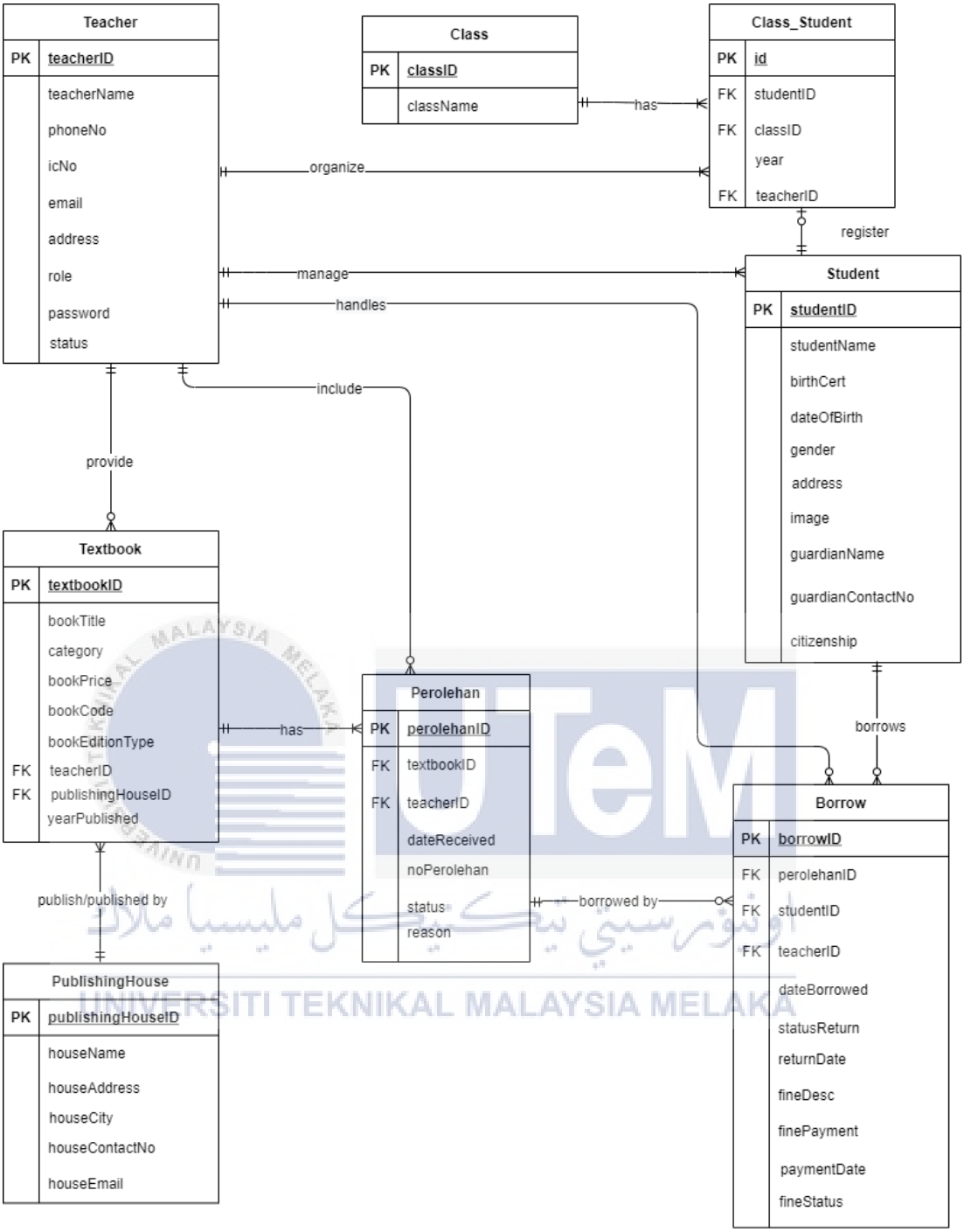


Figure 4.2: Entity Relationship Diagram (ERD)

4.3.2 Logical Design

4.3.2.1 Data Dictionary

Table 4.1: Data Dictionary of Entity TEACHER

TEACHER						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	teacherID	Teacher ID	INT	9999999999	Y	
	teacherName	Nama Guru	VARCHAR(50)	XXXXXXX	Y	
	phoneNo	Nombor Telefon	VARCHAR(15)	999-9999999	Y	
	icNo	Kad Pengenalan	VARCHAR(15)	999999-99-9999	Y	
	email	Email	VARCHAR(50)	x@x.x	Y	
	address	Alamat Rumah	VARCHAR(100)	XXXXXXX	Y	
	role	Jawatan	VARCHAR(50)	XXXXXXX	Y	
	password	Kata Laluan	VARCHAR(50)	Xxx9xxxX	Y	
	status	Status	VARCHAR(50)	XXXXXXX	Y	

Table 4.2: Data Dictionary of Entity STUDENT

STUDENT						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	studentID	Student ID	INT	9999999999	Y	
	studentName	Nama Pelajar	VARCHAR(100)	XXXXXXX	Y	
	birthCert	No Sijil Lahir	VARCHAR(50)	XX99999	Y	
	dateOfBirth	Tarikh Lahir	VARCHAR(50)	2020-08-31	Y	
	gender	Jantina	VARCHAR(50)	XXXXXXX	Y	
	address	Alamat	VARCHAR(100)	XXXXXXX	Y	
	image	Gambar	LONGBLOB	BLOB-9 B	Y	
	guardianName	Nama Penjaga	VARCHAR(50)	XXXXXXX	Y	
	guardianContactNo	No Tel Penjaga	VARCHAR(50)	999-9999999	Y	
	citizenship	Kewarganegaraan	VARCHAR(50)	XXXXXXX	Y	

Table 4.3: Data Dictionary of Entity CLASS_STUDENT

CLASS_STUDENT						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	id	ID	INT	9999999999	Y	
FK	studentID	Student ID	INT	9999999999	Y	STUDENT
FK	classID	Class ID	INT	9999999999	Y	CLASS
	year	Tahun	VARCHAR(10)	9999	Y	
FK	teacherID	Teacher ID	INT	9999999999	Y	TEACHER

Table 4.4: Data Dictionary of Entity CLASS

CLASS						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	classID	Class ID	INT	9999999999	Y	
	className	Nama Kelas	VARCHAR(100)	XXXXXXX	Y	

Table 4.5: Data Dictionary of Entity TEXTBOOK

TEXTBOOK						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	textbookID	Textbook ID	INT	99999999999	Y	
	bookTitle	Judul Buku	VARCHAR(100)	XXXXXXX	Y	
	category	Kategori	VARCHAR(50)	XXXXXXX	Y	
	bookPrice	Harga Buku	VARCHAR(100)	99999999.99	Y	
	bookCode	Kod Buku	VARCHAR(50)	XXXX9999	Y	
	bookEditionType	Edisi Buku	VARCHAR(100)	XXXXXXX	Y	
FK	teacherID	Teacher ID	INT	9999999999	Y	TEACHER
FK	publishingHouseID	Publishing House ID	INT	9999999999	Y	PUBLISHINGHOUSE
	yearPublished	Tahun Terbitan	VARCHAR(50)	2020-08-31	Y	

Table 4.6: Data Dictionary of Entity PEROLEHAN

PEROLEHAN						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	perolehanID	Perolehan ID	INT	9999999999	Y	
FK	textbookID	Textbook ID	INT	9999999999	Y	TEXTBOOK
FK	teacherID	Teacher ID	INT	9999999999	Y	TEACHER
	dateReceived	Tarikh Terima	DATE	2018-10-31	Y	
	noPerolehan	No Perolehan	VARCHAR(100)	9999/99/9999	Y	
	status	Status	VARCHAR(100)	XXXXXXX	Y	
	reason	Sebab Lupus	VARCHAR(50)	XXXXXXX	Y	

Table 4.7: Data Dictionary of Entity BORROW

BORROW						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	borrowID	Borrow ID	INT	9999999999	Y	
FK	perolehanID	Perolehan ID	INT	9999999999	Y	PEROLEHAN
FK	studentID	Student ID	INT	9999999999	Y	STUDENT
FK	teacherID	Teacher ID	INT	9999999999	Y	TEACHER
	dateBorrowed	Tarikh Pinjam	DATE	2018-08-31	Y	
	statusReturn	Status Pulang	VARCHAR(50)	XXXXXXX	Y	
	returnDate	Tarikh Pulang	VARCHAR(50)	2018-08-31	Y	
	fineDesc	Sebab Denda	VARCHAR(100)	XXXXXXX	Y	
	finePayment	Bayaran Denda	VARCHAR(50)	99999999.99	Y	
	paymentDate	Tarikh Dibayar	VARCHAR(50)	2018-08-31	Y	
	fineStatus	Status Denda	VARCHAR(50)	XXXXXXX	Y	

Table 4.8: Data Dictionary of Entity PUBLISHINGHOUSE

DISPOSAL						
Key	**Attribute**	**Content**	**Data Type and Size**	**Format**	**Required?**	**FK Reference Table**
PK	publishingHouseID	Publishing House ID	INT	9999999999	Y	
	houseName	Nama Penerbit	VARCHAR(100)	XXXXXXX	Y	
	houseAddress	Alamat Penerbit	VARCHAR(100)	XXXXXXX	Y	
	houseContactNo	No Telefon	VARCHAR(15)	999-9999999	Y	
	houseEmail	Email	VARCHAR(100)	x@x.x	Y	
	houseCity	Negeri	VARCHAR(50)	XXXXXXX	Y	

4.3.2.2 Validate Conceptual Design

The conceptual design is validated via user transactions, as seen in the examples below. In the diagram, the transaction pathway is labelled.

a. Teacher information of a given role.
b. Textbook information of a given teacher.
c. Perolehan information of a given textbook.
d. Textbook information of a given publishing house.
e. Image of a given student
f. Student included in a given class student
g. Class student of a given class
h. Borrow that done by a given student
i. Borrow information of a given perolehan.

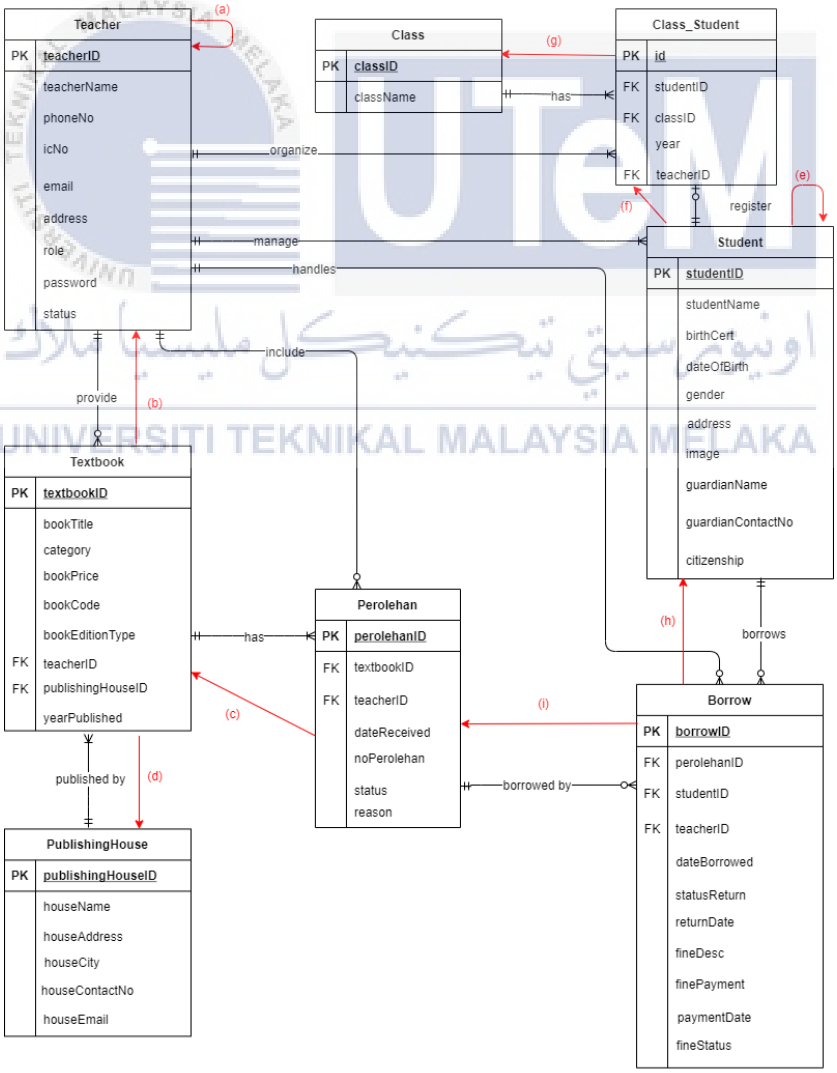


Figure 4.3: Query Transaction Pathway

4.3.2.3 Query Design

This section goes over some of the more complicated queries that are utilised in the proposed system.

Table 4.9: Query Design

No	Transaction	SQL Statement
1.	List of students with information of class name and year. It show list of students by year.	SELECT s.studentID, s.studentName, s.gender, c.className, cs.year FROM class_student AS cs JOIN student AS s ON cs.studentID = s.studentID JOIN class AS c ON cs.classID = c.classID WHERE year = '2020' ORDER BY studentID ASC
2.	Show information of perolehan and textbook to fill form to borrow book.	SELECT p.perolehanID, p.noPerolehan, p.status, t.textbookID, t.bookCode, t.bookTitle FROM perolehan p JOIN textbook t ON p.textbookID = t.textbookID WHERE p.perolehanID=’”.$perolehanID.”’

3.	Display information of fine payment in receipt	SELECT b.borrowID, b.finePayment, b.fineStatus, b.paymentDate, s.birthCert, s.studentName, s.address, t.textbookID, t.bookTitle, t.category FROM borrow AS b JOIN student AS s ON b.studentID = s.studentID JOIN textbook AS t ON b.textbookID = t.textbookID WHERE b.borrowID =".$borrowID;
4.	Display list of student that have to pay fine by only select the fine payment that is not zero value.	SELECT b.borrowID, b.finePayment, b.fineStatus, b.paymentDate, s.birthCert, s.studentName FROM borrow b JOIN student s ON b.studentID = s.studentID WHERE b.finePayment != 0.00
5.	Display the list of disposal textbook also include some information of textbook	SELECT d.disposalID, d.reason, t.bookCode, t.bookTitle, t.bookEditionType, p.noPerolehan FROM disposal AS d JOIN textbook AS t ON d.textbookID = t.textbookID JOIN perolehan AS p ON d.perolehanID = p.perolehanID
6.	Display the student details including their class information and class teacher.	SELECT s.studentID, s.studentName, s.gender, s.image, c.className, cs.year, t.teacherName FROM class_student AS cs JOIN student AS s ON cs.studentID = s.studentID JOIN teacher AS t ON cs.teacherID = t.teacherID

		JOIN class AS c ON cs.classID = c.classID
7.	Display perolehan textbook with some information of textbook and teacher that receive and record the new perolehan textbook received.	SELECT p.perolehanID, p.dateReceived, p.noPerolehan, t.bookCode, t.category, t.bookTitle, tc.teacherName FROM perolehan AS p JOIN textbook AS t ON p.textbookID = t.textbookID JOIN teacher AS tc ON p.teacherID = tc.teacherID”
8.	Count all number of students that have completed fine payment or not completed yet.	SELECT fineStatus, count(*) as number FROM borrow GROUP BY fineStatus

4.3.3 Physical Design

4.3.3.1 DBMS Selection

MySQL Workbench 8.0 CE was used as the database management solution in this system. Microsoft's MYSQL Workbench is a relational database management system. This software solution assumes the primary job of data warehousing and extraction as required by third-party software applications.

4.3.3.2 Usage of Stored Procedures, Triggers and Other Database Objects

Besides The database objects developed and used in this system, in addition to tables, are procedure, and trigger. Chapter 5 delves more into the subject.

Program logic and SQL statements can both be encapsulated in a procedure. In this system, procedures are used to execute some insert and update statements. The operations that were completed utilising the method are given below.

Table 4.10: Usage of Stored Procedure

- Select count total student

When particular actions occur, the trigger is triggered.. Trigger before insert is used to uppercase the data inserted.

Table 4.11: Usage of Trigger

- Trigger before insert to uppercase teacher information - Trigger before insert to uppercase student information - Trigger before insert to uppercase textbook information - Trigger before insert to uppercase perolehan information - Trigger before insert to uppercase disposal information - Trigger before insert to uppercase borrow information

4.3.3.3 Security Mechanism

The value of password is not registered by admin during the registration. The password is auto generate with random password after the admin submit the registration. The password only can be edit by teacher after they login into the system at their profile.

```php
// create function for generate random password
function generate_password($len=8) {
    $chars="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
    $password=substr(str_shuffle($chars), 0, $len);
    return $password;
```

password

q9EuefG6

Figure 4.4: Auto Generate Password

4.3.3.4 Database Contingency

Database can be backup and restored in MySQL Workbench. The file type is in SQL file. User needs to import a SQL file only to restore the system's database.

Figure 4.5: Backup Database

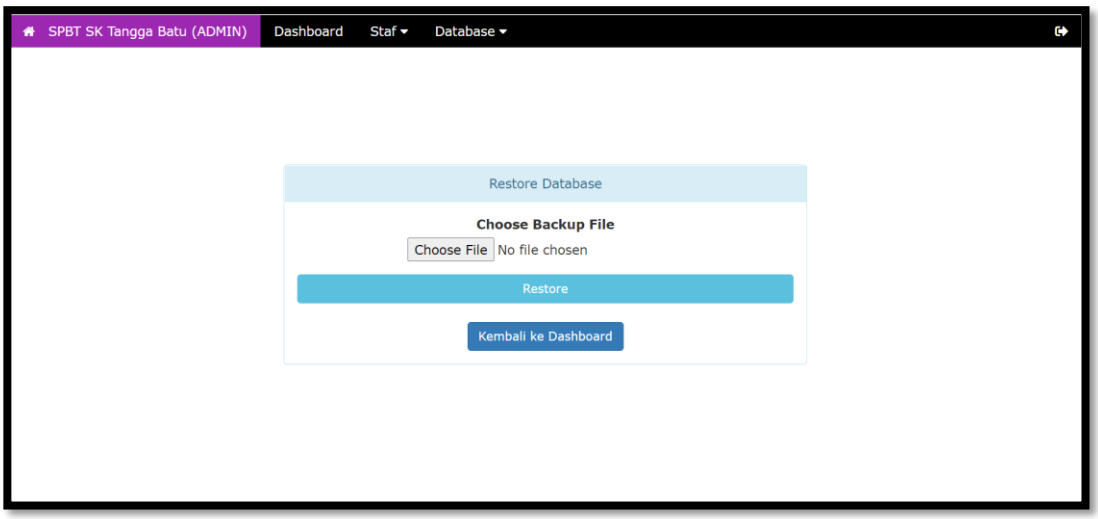


Figure 4.6: Restore Database

4.4 Graphical User Interface (GUI) Design

4.4.1 Navigation Structure

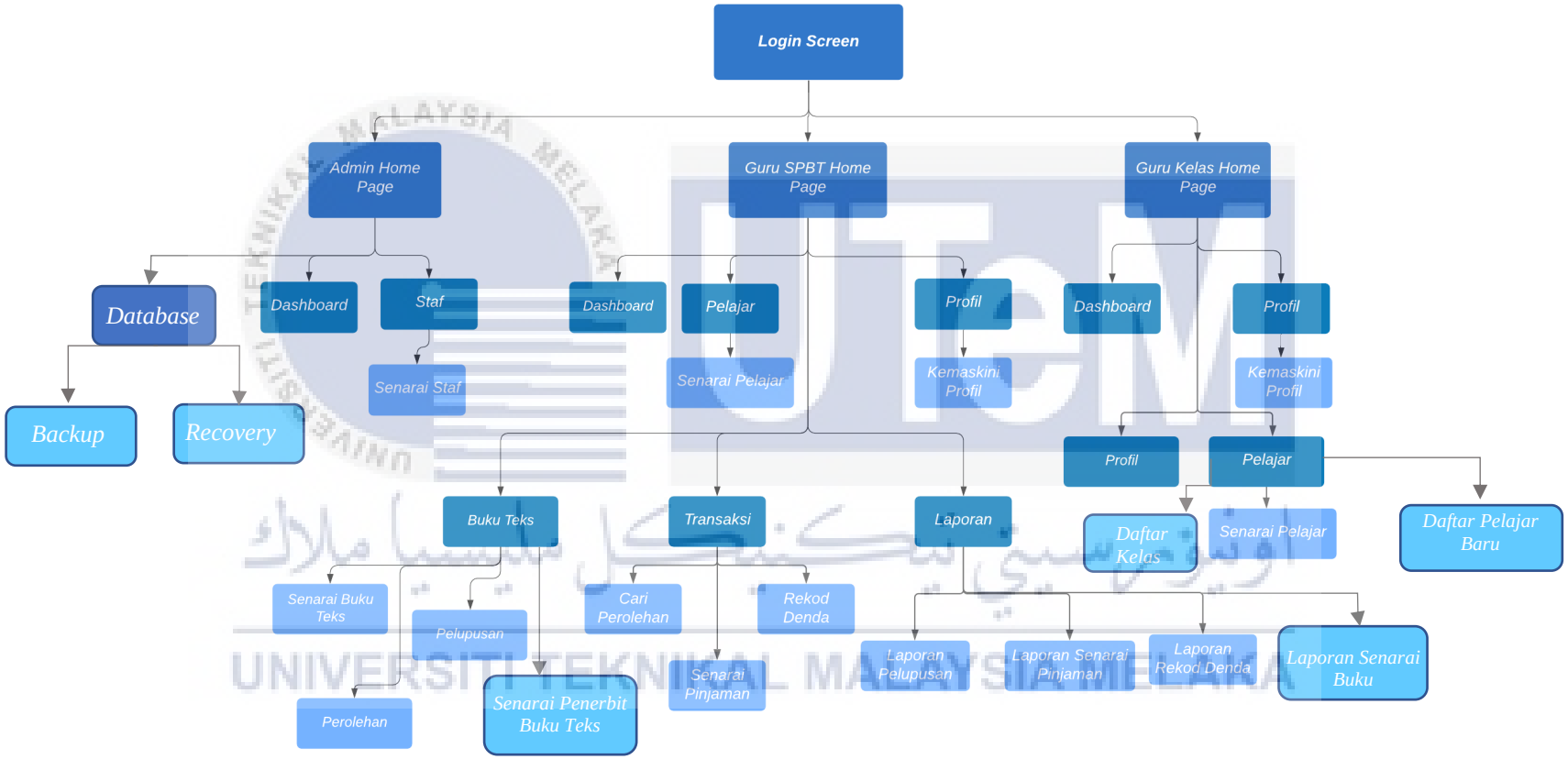


Figure 4.7: Navigation Structure

4.4.2 System Interface

Table 4.12: Login and Logout Interface

Interface	Description
Login and Logout	
 **Figure 4.8: Login Page**	A login page that allows a user to access the system by entering their login credentials correctly. There are three users which are Admin, SPBT Teacher, and Class Teacher. Each of the user will be directed to different Home Page.

Table 4.13: Admin Page

Interface	Description
Admin Page	
 **Figure 4.9: Admin Home Page**	For a successful login, Admin will be directed to Admin Home Page. The menu for admin are 'Dashboard' and 'Staf'.
	For menu 'Dashboard', this page shows the report for total number of teacher, total number of student, total number of textbook and pie

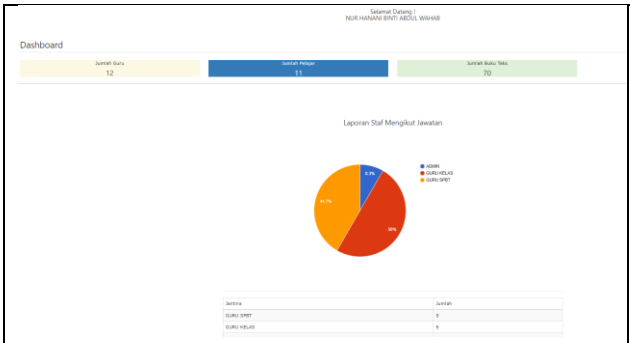


Figure 4.10: Admin Dashboard

chart for total number of staff according to their role.

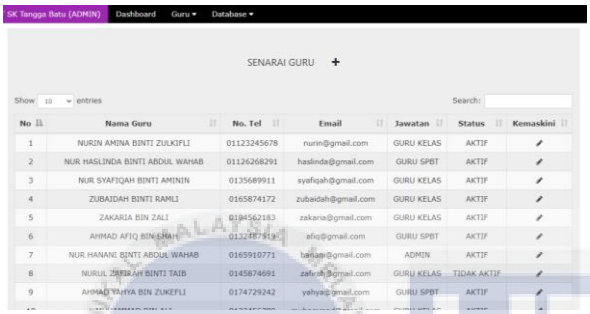


Figure 4.11: List of Teacher

For menu 'Staf', this page shows the list of staff and their details. Admin can register, edit and delete the teacher's information. The search textbook is function for any kind of keywords.

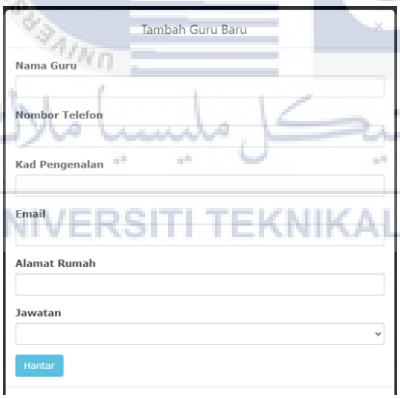


Figure 4.12: Register New Teacher

To register a new teacher, admin need to insert the name of teacher, phone number, identification number, email, address and role. The dropdown for role are 'Guru SPBT' and 'Guru Kelas'.

To edit the information of teachers, admin can edit all the details except for Id and Password. The password is automatically generated during the registration.

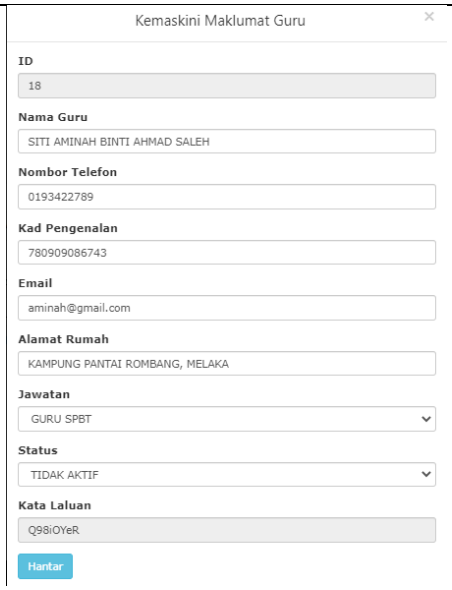


Figure 4.13: Edit Teacher's Information

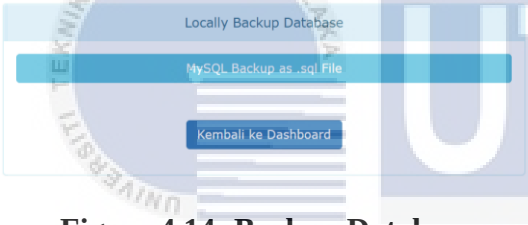


Figure 4.14: Backup Database

The backup page use to backup the database in MySQL.

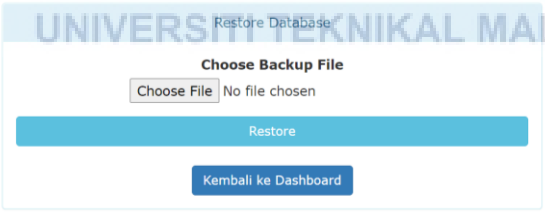


Figure 4.15: Restore Database

Restore Database is to restore the backup files and import to the MySQL.

Table 4.14: Class Teacher Page

Interface	Description
Class Teacher Page	
 **Figure 4.16: Class Teacher Home Page**	For a successful login, Class Teacher will be directed to Class Teacher Home Page. The menu for admin are 'Dashboard', 'Pelajar' 'Laporan' and 'Profil'.
 **Figure 4.17: Class Teacher Dashboard**	For menu 'Dashboard', it shows the report for total number of student and pie chart for total number of student according to their gender.
 **Figure 4.18: List of Students**	For menu 'Pelajar' and submenu 'Daftar Pelajar Baru', this page display the list of student and their details. Class Teacher can register, edit and delete the teacher's information. The search textbook is function for any kind of keywords.
	To register a new student, class teacher need to insert the birth certification, student name, address, date of birth, guardian name, guardian phone number, citizenship and upload an image.

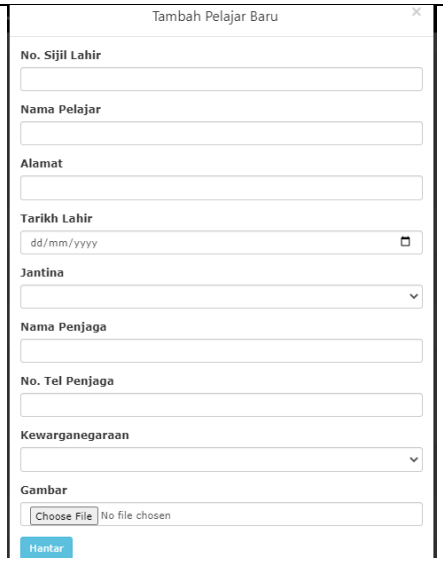


Figure 4.19: Register new student

Figure 4.20: Edit student information

To edit the information of student, class teacher can edit all the details. The information of student all in uppercase by using trigger before insert.

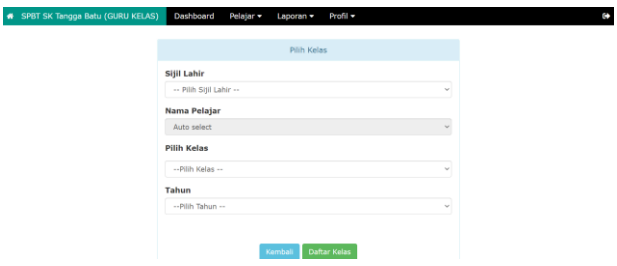
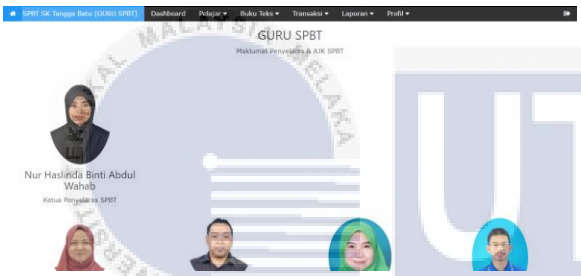


Figure 4.21: Register class for students

The submenu 'Daftar Kelas' is a page where class teacher can select class for their students.

 **Figure 4.22: List of students by year**	The submenu 'Senarai Pelajar' shows list of students by year. The information of students can be view or delete.

Table 4.15: SPBT Teacher Page

Interface	Description
SPBT Teacher Page	
 **Figure 4.23: SPBT Teacher Home Page**	For a successful login, SPBT Teacher will be directed to SPBT Teacher Home Page. The menu for SPBT Teacher are 'Dashboard', 'Pelajar', 'Buku Teks, 'Transaksi', 'Laporan' and 'Profil'.
 **Figure 4.24: SPBT Teacher Dashboard**	For menu 'Dashboard', it shows the report for total number of student, total number of textbook and total number of textbook copy (perolehan).
 **Figure 4.25: List of students**	For menu 'Pelajar', this page display the list of student and their details. SPBT Teacher can view the student's information. The search textbook can be used to search any keywords.

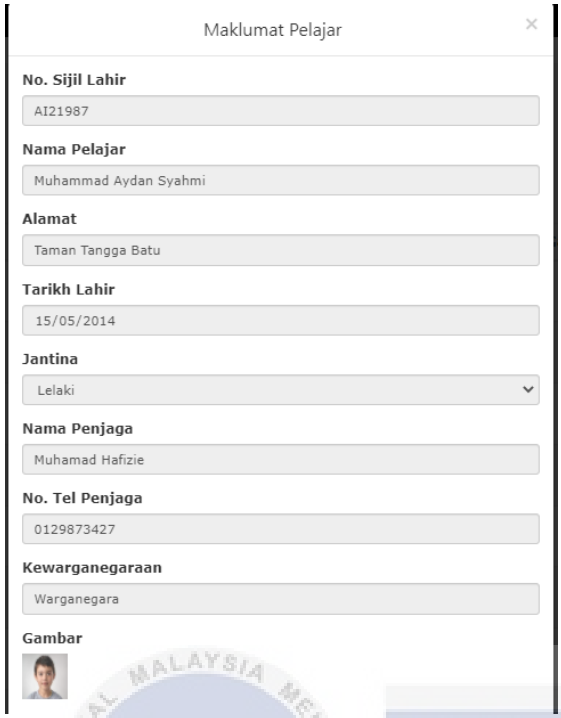
<table>
<tr>
<td> 

Figure 4.26: Information of students</td>
<td>This form display the information of students. User can view the details but cannot make any changes.</td>
</tr>
<tr>
<td> 

Figure 4.27: List of textbooks</td>
<td>The first sub menu for menu ‘Buku Teks’ is list of textbooks. All textbooks details are displayed such as category, textbook title, price of textbook, textbook code and type of book edition, publisher name and year published. User can register a new textbook details and can edit it by click at the icon.</td>
</tr>
<tr>
<td></td>
<td>This form is for user to register a new textbook into the system. The information needed are category, textbook title, price of textbook, textbook code and type of book edition.</td>
</tr>
</table>

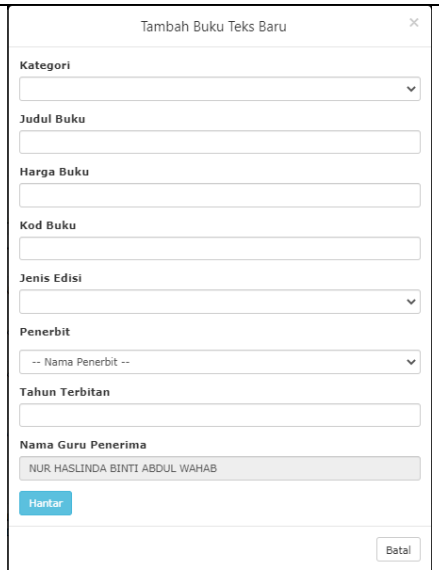


Figure 4.28: Register new textbook information

Figure 4.29: List of textbook copy

Next sub menu for menu 'Textbook' is Perolehan. This page display the list of perolehan and their details. User can add new perolehan and can delete any information of perolehan.

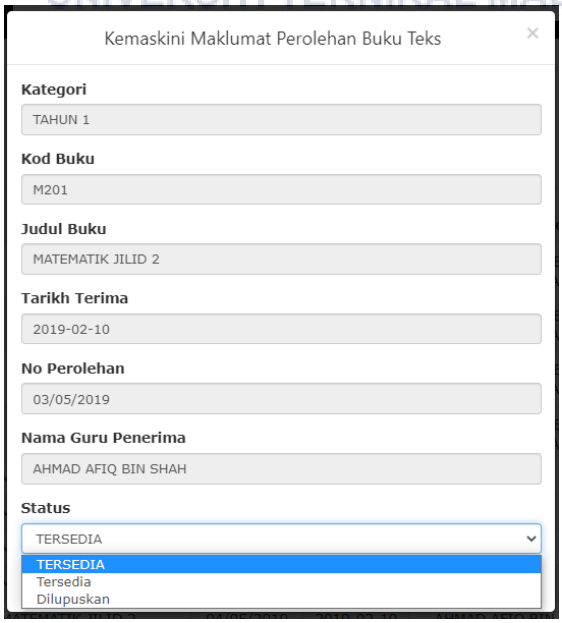
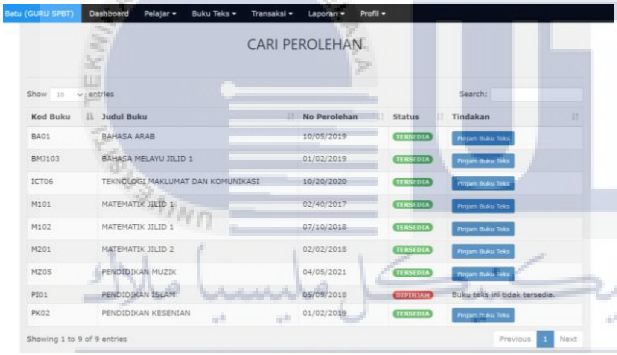


Figure 4.30: List of disposal textbook

The user can edit the status of perolehan textbook either 'Tersedia' or 'Dilupuskan'.

 **Figure 4.31: List of textbook disposal**	For submenu ‘Pelupusan’, this page shows list of textbooks that have to dispose with reason. User can edit the reason why the textbook is dispose in this page and also can delete the data.
 **Figure 4.32: List of publishing house**	The submenu ‘Senarai Penerbit Buku Teks’ shows list of publishing house and their details. User can add and edit the data for publishing house information.
 **Figure 4.33: Search perolehan**	The first function for menu ‘Transaksi’ is search Perolehan. This page is for SPBT Teacher to record which the textbook that borrowed by students referring to the perolehan series number. The green button is for available textbook, which mean that textbook can be borrow. While, the red button is for unavailable textbook which make the button for borrowing is disable.
	This form is for borrowing textbook. The series number, textbook code, textbook title are fetch from previous page. Teacher SPBT can choose the birth certification and the student name will automatically show. The date will be current date.

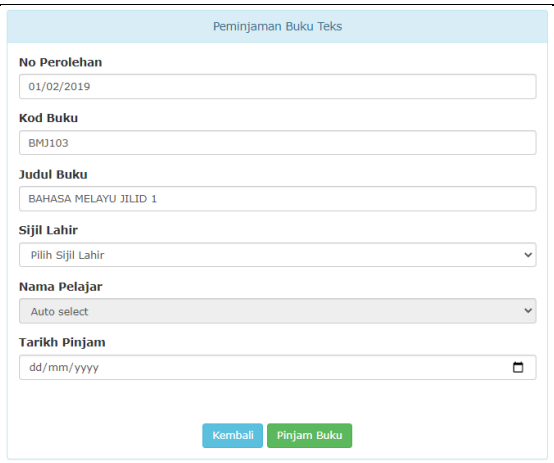


Figure 4.34: Borrow textbook

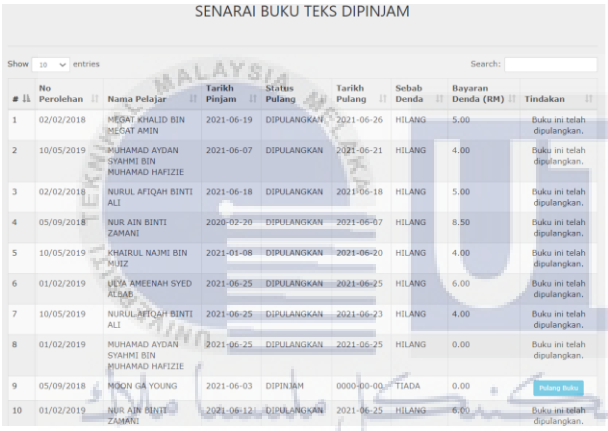


Figure 4.35: List student that borrow textbook

For next submenu in menu 'Transaksi' is the list of borrowed textbooks. In this page display the series number of textbook, student name, date of borrow textbook, status return, date of return textbook and button for return textbook. After returning textbook, the return button will be disable. If the student cause any damage to the textbook, the status fine will be 'Hilang' and student need to pay fine.

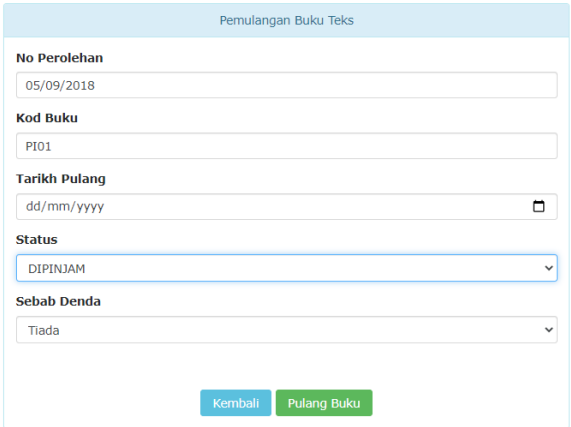


Figure 4.36: Return textbook

This form is for SPBT Teacher update the date of return, status return and choose reason of fine (if have).

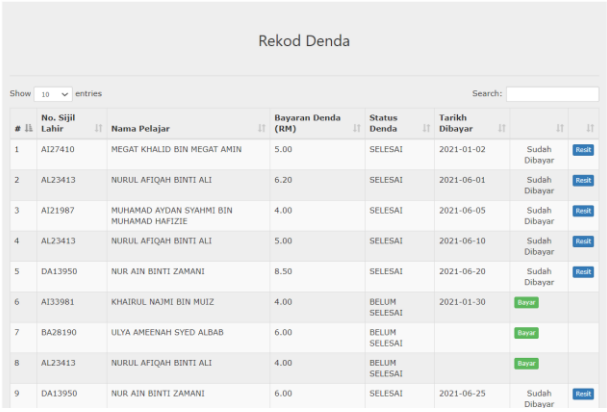


Figure 4.37: Fine Record

Next submenu for menu 'Transaksi' is the record of fine for student. All student that need to pay fine will listed in this page. SPBT Teacher will update the payment date and the status of fine after student pay their fine. Then, SPBT Teacher can print the receipt to give to that student.

Figure 4.38: Pay Fine Payment

This form is use for updating the date and status for paying fine.

Figure 4.39: Receipt Payment

This receipt have the information of student and description for their fine record.

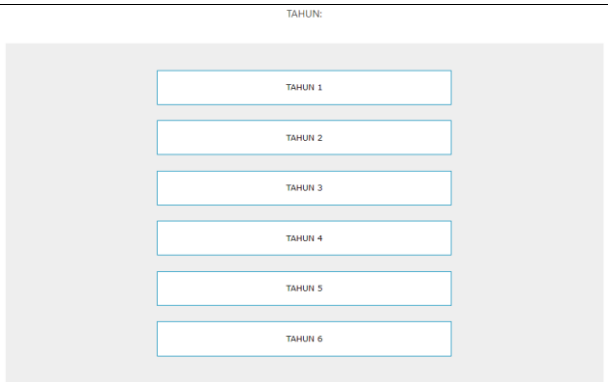


Figure 4.40: Choose Grade

The submenu 'Laporan Senarai Buku' is a report for the list of textbook in each grade which are Grade 1, Grade 2, Grade 3, Grade 4, Grade 5, and Grade 6.

Figure 4.41: List of textbook for Grade 1

The report is specifically show list of textbook for each grade. It show the number of perolehan, number book that missing, number of disposal and also calculate the balance textbook that still in good condition.

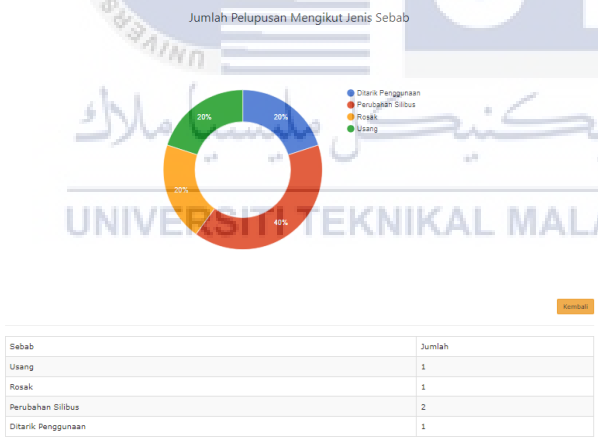
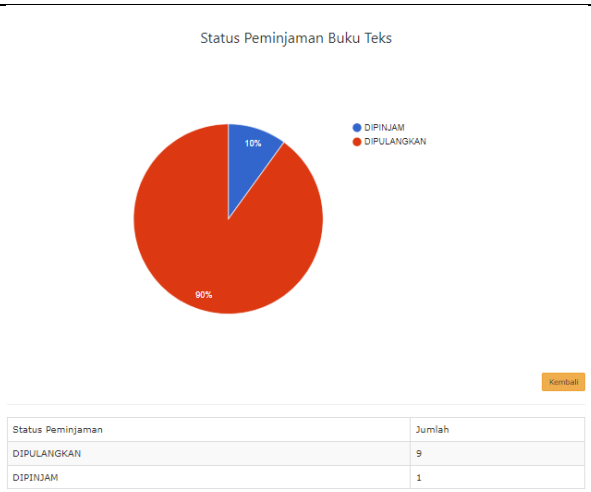


Figure 4.42: Report Disposal by Reason

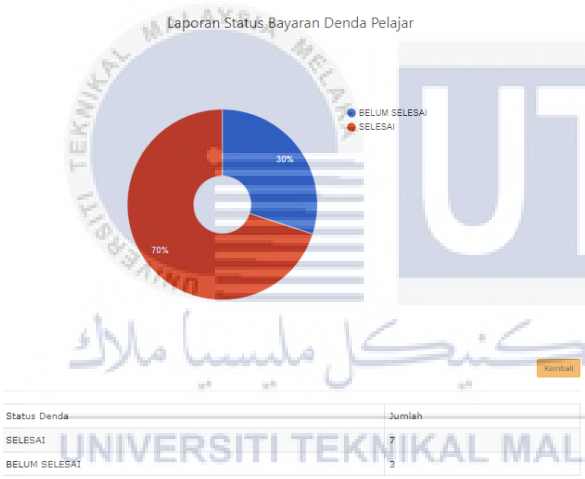
The first pie chart show the report for total of disposal textbook referring to their reason of dispose.

The second pie chart show the report for the status of textbook that borrowed by students.



Status Peminjaman	Jumlah
DIPULANGKAN	9
DIPINJAM	1

Figure 4.43: Report Status Return Textbook



Status Denda	Jumlah
SELESAI	7
BELUM SELESAI	3

Figure 4.44: Report Status Fine Payment

The third pie chart show the report for status of fine payment by students.

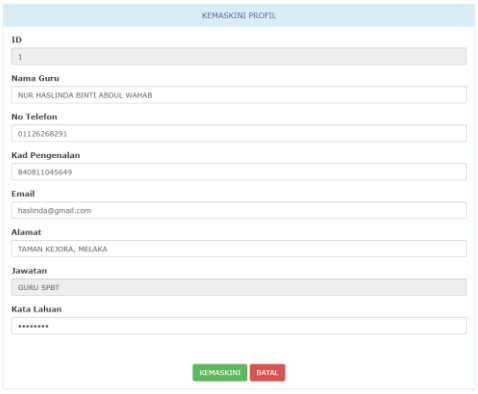


Figure 4.45: Edit Profile

The menu 'Profil' is for user to change their profile. User can edit their name, phone number, identification number, email, address and password. Id and role for 'Guru SPBT' cannot be edited as it set as readonly.

4.5 Conclusion

Database Designing a database entails more than just modelling data from sources. It also entails deciding on the database's conceptual approach, the functions it is expected to do, and how the database's tables and fields should operate at the most basic levels. The system's implementation is explained in the following chapter.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

The process of translating business requirements into a system is known as the implementation phase. The careful specification from the system design phase is handed over to the programmer, who then turns it into a working system. In order for the system and database environment to be ready for operation, they must be set up from the start. Only after the environment has been set up can the implementation phase begin. This includes the building of databases, database objects, data loading, and system development.

5.2 Software Development Environment Setup

This section shows how to set up the database, including database installation and database services start-up. This project uses Microsoft Visual Studio Code to connect to and operate with MySQL Workbench 8.0 CE. The database, as well as database objects such as tables, stored procedures, and triggers, are then built..

5.2.1 Database Installation

Functional requirements are system capabilities that must be met in order for users' needs to be met. The system should be able to perform the tasks that have been asked.

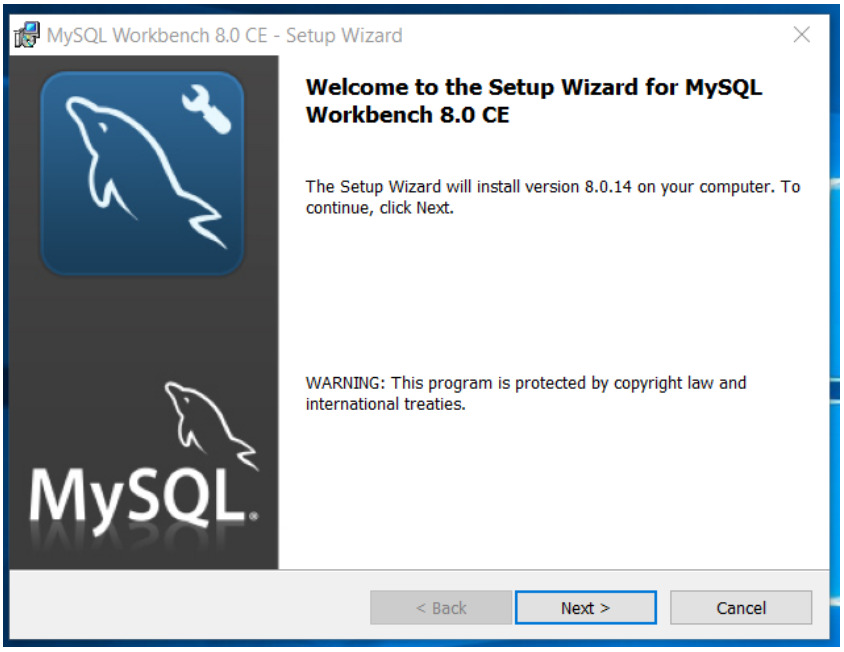


Figure 5.1: Installation of MySQL Workbench (Step 1)

Step 1: Download MySQL Workbench's installer from https://dev.mysql.com/downloads/mysql/. The start-up screen should something like figure above and click the next button to get started.

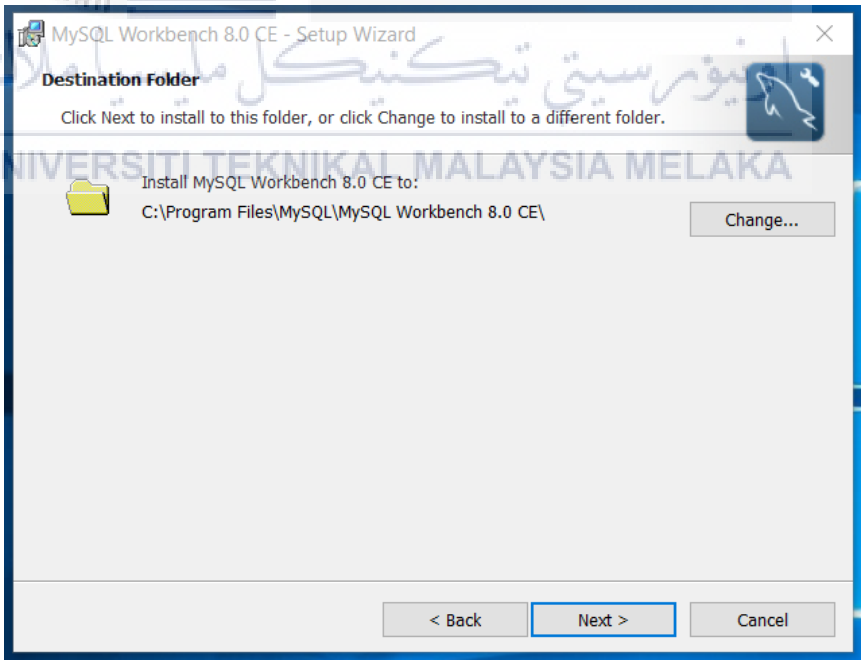


Figure 5.2: Installation of MySQL Workbench (Step 2)

Step 2: Choose file directory as to where we're going to install the program. The default directory is usually fine for most. Click the next button.

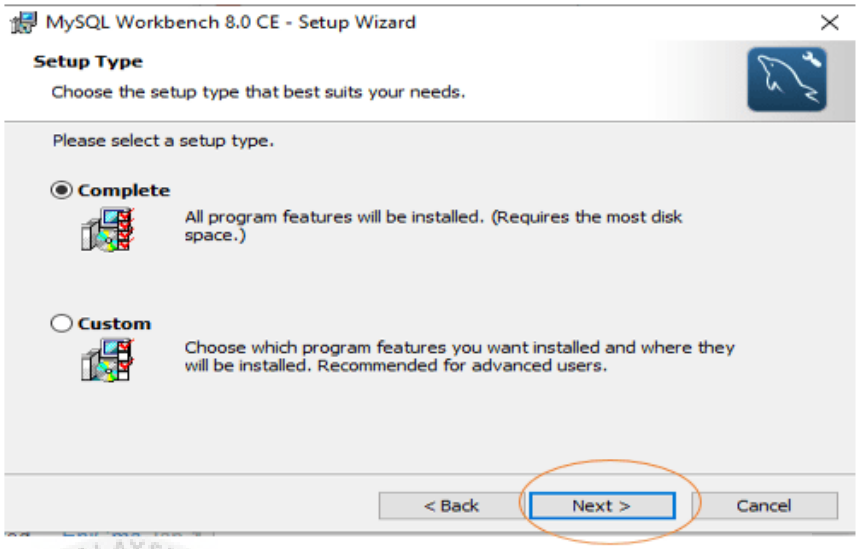


Figure 5.3: Installation of MySQL Workbench (Step 3)

Step 3: Choose setup type. It is safe to choose complete because the installer will install all the files needed. Choose Complete and click the next button.

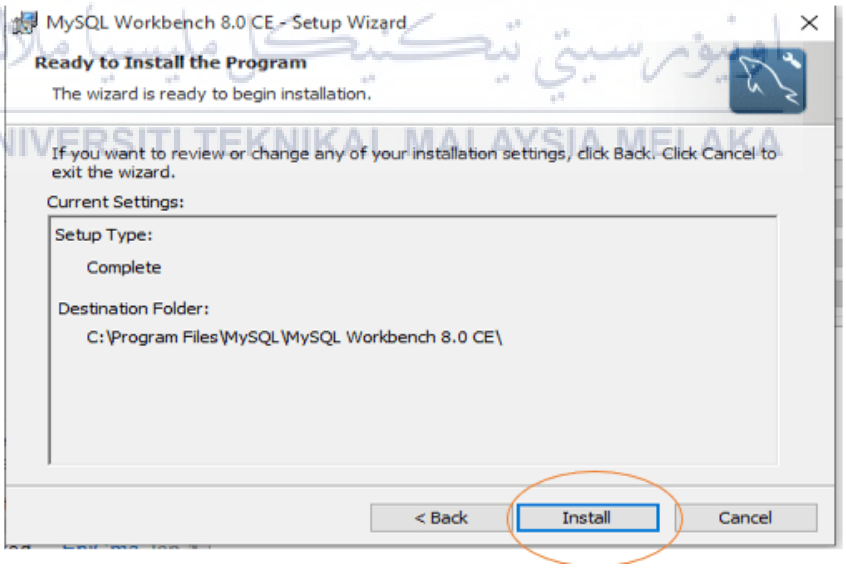


Figure 5.4: Installation of MySQL Workbench (Step 4)

Step 4: Click Install button.

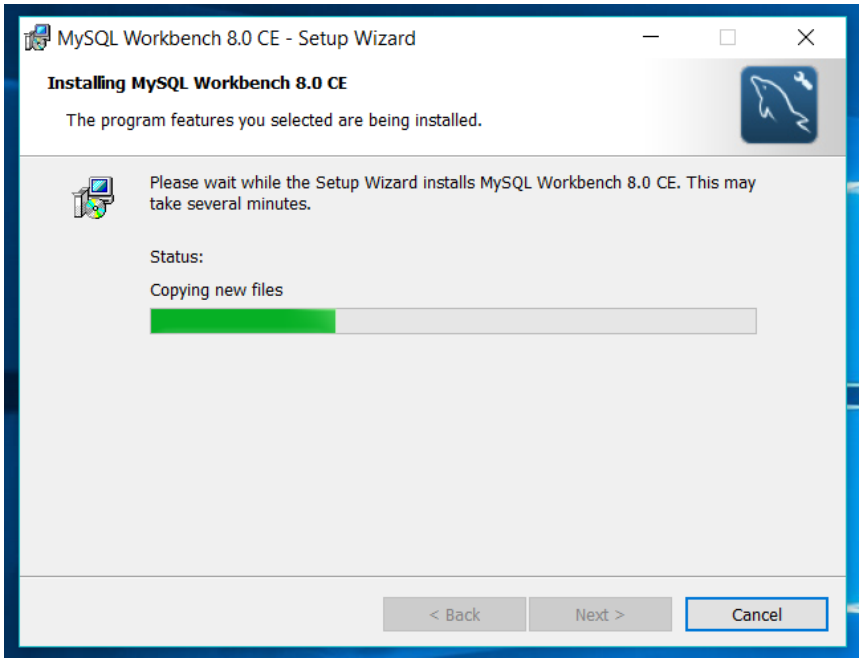


Figure 5.5: Installation of MySQL Workbench (Step 5)

Step 5: Installation began. The process might take several minutes.

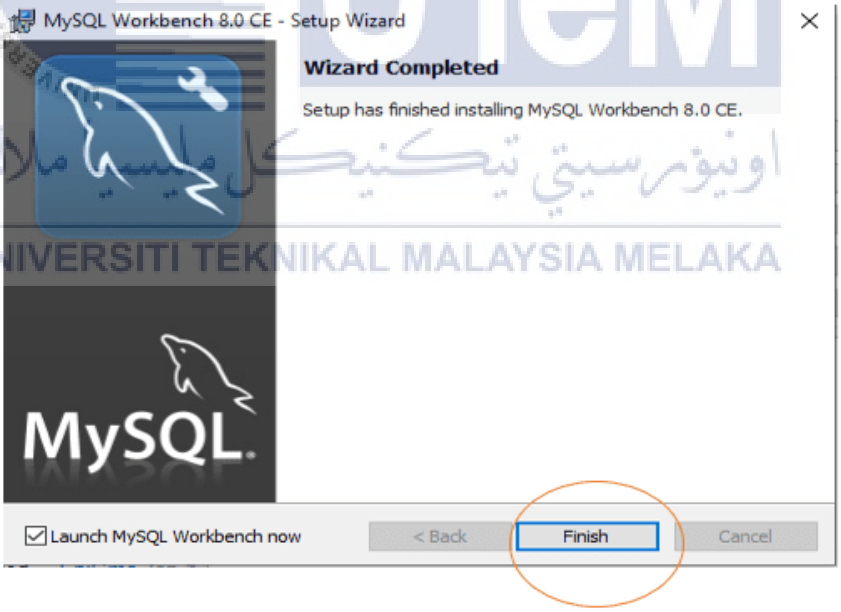


Figure 5.6: Installation of MySQL Workbench (Step 6)

Step 6: After finished installing, click on the checkbox at the bottom left corner to launch the program immediately after hitting the finish button.

5.2.2 Database and Database Object Creation

To begin, build a database. The database is given the name 'spbt.' Then, using the 4.3.2.1 Data Dictionary, build a table and populate it. Create a stored procedure, and a trigger after that. The details can be found in the next section.

5.3 Database Implementation

The statements for creating database objects, stored procedures, and triggers are presented in this section.

Below shows DDL statement to create one of the tables in SPBT Management System. The remaining tables' DDL statements can be found in Appendix.

1. DDL statement to create

```
CREATE TABLE `textbook` (
 `textbookID` int(11) NOT NULL AUTO_INCREMENT,
 `bookTitle` varchar(100) NOT NULL,
 `category` varchar(50) NOT NULL,
 `bookPrice` varchar(100) NOT NULL,
 `bookCode` varchar(50) NOT NULL,
 `bookEditionType` varchar(100) NOT NULL,
 `teacherID` int(11) NOT NULL,
 `publishingHouseID` int(11) NOT NULL,
 `yearPublished` varchar(50) NOT NULL,
 PRIMARY KEY (`textbookID`)
) ENGINE=InnoDB AUTO_INCREMENT=76 DEFAULT CHARSET=latin1
```


The purpose of the data loading procedure is to populate the database. DML statement is used to insert data into the tables.

Refer to 4.3.3.2 Usage of Stored Procedure and Triggers.

2. Syntax to create stored procedure to count total of students:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetTotalStudent`()
```

```
BEGIN
DECLARE totalStudent INT DEFAULT 0;

SELECT COUNT(*)
INTO totalStudent
FROM student;

SELECT totalStudent;
END
```

After that, create trigger. Below shows syntax to create trigger in this system.

3. Trigger before insert to uppercase for textbook

```
CREATE DEFINER=`root`@`localhost` TRIGGER
`textbook_BEFORE_INSERT` BEFORE INSERT ON `textbook` FOR EACH ROW
BEGIN
    declare recent integer;
    set recent := (select count(*) from textbook);
    set new.bookTitle := UPPER(new.bookTitle);
    set new.category := UPPER(new.category);
    set new.bookPrice := UPPER(new.bookPrice);
    set new.bookCode := UPPER(new.bookCode);
    set new.bookEditionType := UPPER(new.bookEditionType);

END
```


5.4 Conclusion

In conclusion, for the most part, this chapter explains the database implementation process. MySQL Workbench 8.0 CE with Microsoft Visual Studio Code was used as the RDBMS for this project. The MySQL Workbench installation process is reviewed and shown using diagrams. As a result, the database is built up for the system one step at a time. Creating a database, tables, and populating them, as well

as stored procedures and triggers. The testing phase of system development will be discussed in the following chapter.

CHAPTER 6: TESTING

6.1 Introduction

Following the installation of the system, testing is carried out. Various testing methodologies are used to identify and correct flaws during testing. Prior to testing, the test environment must be set up. Location of testing, hardware, firmware settings, preparations, and staff training are all examples of this. Then, in proportion to the test case produced by the tester, testing is carried out, and the test result is recorded.

6.2 Test Plan

Test Plan is the blueprint for the project's testing phase. Testing must be planned in advance to determine who, what, when, and how the testing will take place. ‘Who' is the tester, ‘what' is the feature to be tested, ‘when' is the desired testing period or duration, and ‘how' is the testing method or approach.

6.2.1 Test Organization

Project manager, testing lead, testing team, and development team are some of the roles that will involved in the management, planning, or execution of the test. The project manager goes over all of the deliverables, such as the Test Plan, Test Case, and Test Summary Report, and signs them off. The testing lead creates the test plan and supervises the testing. Testing teams are made up of testers with varying levels of freedom that create test cases and run tests. The development team consists of programmers who write code and apply remedies for flaws discovered by testers.

Developers test their own code in this project. At the same time, the developer appears on behalf of the tester. Test planning, test estimation, test design, test execution, and fix detection are all tasks performed by developers and testers. Nur Haslinda pretended to be a user acceptability tester with good domain knowledge (the education industry) but no testing experience.

.

6.2.2 Test Environment

The testing is done on a laptop at the developer's location. Table 3.3 of Chapter 3 lists the laptop's hardware specifications. The software required for testing is listed in Table 6.1.

Table 6.1: Software Specification

Component	Software
Operating System	Windows 10 Home Single Language
Database	MySQL Workbench 8.0 CE
Development Tools	Microsoft Visual Studio Code

6.2.3 Test Schedule

Unit testing, integration testing, system testing, and acceptance testing are all scheduled at various test levels. Unit testing examines the smallest testable item, such as a function, class, or procedure. Integration testing examines the interface and flow of information among a group of components. System testing examines the overall behaviour and functionality of the system. Acceptance testing determines if the system meets the needs of the users.

Table 6.2: Test Schedule

Test Level	Start Date	End Date	Duration
Unit Testing	14 Jun 2021	23 July 2021	6 weeks
Integration testing	21 Jun2021	6 August 2021	7 weeks
System testing	28 Jun 2021	20 August 2021	8 weeks
Acceptance testing	6 September 2021	10 September 2021	1 week

6.3 Test Strategy

The technique for doing testing, which includes both static and dynamic testing, is known as the test strategy. Dynamic testing requires the execution of the system, whereas static testing does not. The system is not executed during static testing. It performs a verification procedure to prevent flaws by analyzing coding and documentation, such as requirement specifications, design specifications, test cases, and other similar documents. The system is run during dynamic testing. By executing the system based on the test case, it does validation to find and remedy flaws. There are two types of testing: black box and white box. Black box testing generates input values without knowing the programme logic, whereas white box testing generates input values knowing the programme logic.

6.3.1 Classes of Tests

There are a variety of testing types that focus on specific objectives that were used to test this project:

Table 6.3: Testing Type and Description

Testing Type	Description
Functional testing	To test software functions, use the black box testing technique.
Structural testing	To test in detail according to the system architecture, use the white box testing technique.
Confirmation testing	Retesting is another term for it. After you've addressed the defects, run the test again to make sure they're gone.
Regression testing	After confirmation testing, test adjacent components to see if a new defect has arisen as a result of the adjustments.
Use case testing	Check to see if the user's path is operating as expected.
Security testing	To maintain the CIA trifecta of confidentiality, integrity, and availability in the system. Security domains such as authentication, authorization, privileges, and encryption should be highlighted.

6.4 Test Design

This section contains test cases and data for all of the modules in the SPBT Management System.

6.4.1 Test Description

6.4.1.1 Test Case for Administrator

Table 6.4: Test Case for Administrator

Module	Test Case ID	Description	Expected Result
Login	AD001_01	To authenticate user credential when logging into the system with correct email and password	System will direct the user into Admin Home Page
	AD001_02	To authenticate user credential when incorrect email or password is inserted	System will display an error message of 'Email or Password wrong, Please re-login.'
	AD001_03	To authenticate user when one field is left blank	System will ask user to fill out the blank field
Dashboard	AD002_01	To authenticate user can view the report of total number of teacher, student, textbook and pie chart for total staff by position	System will display the total number of teacher, student, textbook and pie chart for total staff by position
Teacher	AD003_01	To authenticate user can view list of teachers	System will display list of teachers
	AD003_02	To authenticate user can add new teacher by fill in the required field and click on 'Hantar' button	System will display message of 'Upload data success !'
	AD003_03	To authenticate when user left one field blank	System will ask user to fill out the blank field
	AD003_04	To authenticate user can update data by click on edit icon	System will display message of 'Update data success !'
Database	AD004_01	To authenticate user can backup database from MySQL as .sql file	System will backup the database locally

	AD004_02	To authenticate user can restore database by choosing backup file	System will restore the backup database chosen

6.4.1.2 Test Case for Class Teacher

Table 6.5: Test Case for Class Teacher

Module	Test Case ID	Description	Expected Result
Login	C001_01	To authenticate user credential when logging into the system with correct email and password	System will direct the user into Class Teacher Home Page
	C001_02	To authenticate user credential when incorrect email or password is inserted	System will display an error message of 'Email or Password wrong, Please re-login.'
	C001_03	To authenticate user when one field is left blank	System will ask user to fill out the blank field
Dashboard	C002_01	To authenticate user can view the report of total number of student	System will display the total number of student
Student	C003_01	To authenticate user can view list of students	System display list of students
	C003_02	To authenticate user can add new student by fill in the required field and click on 'Hantar' button	System will display message of 'Upload data success !'
	C003_03	To authenticate when user left one field blank	System will ask user to fill out the blank field
	C003_04	To authenticate user can update data by click on edit icon	System will display message of 'Update data success !'
	C003_05	To authenticate user can delete by click on delete icon	System will display message of 'Delete data success !'
	C003_06	To authenticate user can choose class and year for students	System will display 'Daftar kelas berjaya direkod!'

	C003_07	To authenticate user can view list of students in year 2020 and 2021	System will display list of students by year chosen
	C003_08	To authenticate user can delete by click on delete icon	System will display message of 'Delete data success !'
Reporting	C004_01	To authenticate user can view the report by pie chart for total number of student by gender	System will display the total number of student by gender
	C004_02	To authenticate user can view the report by pie chart for total number of student by citizenship	System will display the total number of student by citizenship
Profile	C005_01	To authenticate user can view personal details	System will display personal details
	C005_02	To authenticate user can update detail by click on 'Kemaskini' button	System will display message of 'Update data success !'

6.4.1.3 Test Case for SPBT Teacher

Table 6.6: Test Case for SPBT Teacher

Module	Test Case ID	Description	Expected Result
Login	T001_01	To authenticate user credential when logging into the system with correct email and password	System will direct the user into Teacher SPBT Home Page
	T001_02	To authenticate user credential when incorrect email or password is inserted	System will display an error message of 'Email or Password wrong, Please re-login.'
	T001_03	To authenticate user when one field is left blank	System will ask user to fill out the blank field
Dashboard	T002_01	To authenticate user can view the report of total number of student, textbook, perolehan, borrowed book, and available perolehan textbook	System will display the total number of student, textbook, perolehan, borrowed book, and available perolehan textbook
Student	T003_01	To authenticate user can choose year 2020 or 2021	System will direct user to chosen year page
	T003_02	To authenticate user can view list of students in year 2020 and 2021	System will display list of students by year chosen
Textbook	T004_01	To authenticate user can view list of textbooks	System display list of textbooks
	T004_02	To authenticate user can add new textbook by fill in the required field and click on 'Hantar' button	System will display message of 'Upload data success !'
	T004_03	To authenticate when user left one field blank	System will ask user to fill out the blank field
	T004_04	To authenticate user can update data by click on edit icon	System will display message of 'Update data success !'
	T004_05	To authenticate user can export the list of textbook into .CSV file by click at the 'Export' button	System will direct export file

Perolehan	T005_01	To authenticate user can view list of perolehan	System display list of perolehan
	T005_02	To authenticate user can add new perolehan by fill in the required field and click on 'Hantar' button	System will display message of 'Upload data success !'
	T005_03	To authenticate when user left one field blank	System will ask user to fill out the blank field
	T005_04	To authenticate user can update data by click on edit icon	System will display message of 'Update data success !'
	T005_05	To authenticate user can delete by click on delete icon	System will display message of 'Delete data success !'
Disposal	T006_01	To authenticate user can view list of disposal textbook	System display list of disposal textbook
	T006_02	To authenticate user can update reason for disposal by click on edit icon	System will display message of 'Update data success !'
	T006_03	To authenticate user can delete by click on delete icon	System will display message of 'Delete data success !'
Publishing House	T007_01	To authenticate user can view list of publishing house	System display list of publishing house
	T007_02	To authenticate user can add new publishing house by fill in the required field and click on 'Hantar' button	System will display message of 'Upload data success !'
	T007_03	To authenticate when user left one field blank	System will ask user to fill out the blank field
	T007_04	To authenticate user can update data by click on edit icon	System will display message of 'Update data success !'
Transaction Textbook	T008_01	To authenticate user can view list of perolehan textbook that are available/not available/dispose	System display list of perolehan textbook
	T008_02	To authenticate user can borrow textbook by click 'Pinjam Buku Teks' button	System will display message of 'Data pinjaman baru berjaya direkod!'

		and update birth certification and date of borrow	
	T008_03	To authenticate user can view list of borrowed textbook	System display list of borrowed textbook
	T008_04	To authenticate user can return book by click 'Pulang Buku' button and update return date, status and fine description	System will display message of 'Pemulangan Buku Berjaya !'
	T008_05	To authenticate user can view list of fine record	System display list of fine record
	T008_06	To authenticate user can pay fine by click 'Bayar' button and update payment date and fine status	System will display message of 'Update data success !'
	T008_07	To authenticate user can view the receipt by click 'Resit' button	System will display the receipt
	T008_08	To authenticate user can print the receipt by click 'Print' button	System will direct to print the receipt
Reporting	T009_01	To authenticate user can choose list of textbook by grade	System will direct user to grade chosen
	T009_02	To authenticate user can view list of textbook with total of perolehan, total of textbook missing, total of disposal textbook, and remaining of textbook left	System will display list of textbook with total of perolehan, total of textbook missing, total of disposal textbook, and remaining of textbook left
	T009_03	To authenticate user can view the report by pie chart for total of disposal textbook by reason	System will display the total of disposal textbook by reason
	T009_04	To authenticate user can view the report by pie chart for total borrowed textbook by status	System will display the total borrowed textbook by status
	T009_05	To authenticate user can view the report by pie chart	System will display the total of fine record by fine status

		for total of fine record by fine status	
Profile	T010_01	To authenticate user can view personal details	System will display personal details
	T010_02	To authenticate user can update detail by click on 'Kemaskini' button	System will display message of 'Update data success !'

6.4.2 Test Data

6.4.2.1 Test Data for Administrator

Table 6.7: Test Data for Administrator

Test Case ID	Test Data	Steps
AD001_01	Email: hanani@gmail.com Password: 123456	1. Fill in the required field with given data. 2. Click 'login' button
AD001_02	Email: hanani@gmail.com Password: 123abc (incorrect data)	
AD001_03	Email: (left blank) Password: 123456	
AD002_01	Dashboard (Reporting) Jumlah Guru: 16 Jumlah Pelajar: 35 Jumlah Buku Teks: 72 Guru SPBT: 5 Guru Kelas: 10 Admin: 1	1. Click 'Dashboard' Menu 2. Scroll down
AD003_01	New Teacher (correct data) Nama Guru: Nur Najah No Telefon: 0148977610 Kad Pengenalan: 910819045675 Email: najah@gmail.com Alamat Rumah: Taman Bertam, Melaka Jawatan: Guru Kelas Status: Aktif Password: Sn4BzlKW (random password)	1. Click 'Guru' Menu 2. Click 'Senarai Guru' 3. Fill in the required field with given data 4. Click 'Add/Update' icon button
AD003_02	New Teacher Nama Guru: Nur Najah No Telefon: 0148977610	

	Kad Pengenalan: 910819045675 Email: najah@gmail.com Alamat Rumah: Taman Bertam, Melaka Jawatan: Guru Kelas Status: (left blank)	
AD003_03	New Teacher Nama Guru: Nur Najah No Telefon: 0148977610 Kad Pengenalan: 910819045675 Email: najah@gmail.com Alamat Rumah: Taman Bertam, Melaka Jawatan: Guru Kelas Status: Tidak Aktif Password: najah123 UPDATE	
AD004_01	Backup Database Successfully backup SPBT database locally. File path is C:/xampp/htdocs/spbt/backups/spbt_backup_20210827081202.sql	1. Click 'Database' Menu 2. Click 'Backup Database' 3. Click 'MySQL Backup as .sql file' 4. Click 'Restore Database' 5. Choose backup file to restore
AD004_02	Restore Database (correct data) Choose File: spbt_backup_20210827081202.sql	
AD004_03	Restore Database Choose File: spbt_system.php (Invalid File Type)	

6.4.2.2 Test Data for Class Teacher

Table 6.8: Test Data for Class Teacher

Test Case ID	Test Data	Step
C001_01	Email: syafiqah@gmail.com Password: syafiqah	1. Fill in the required field with given data. 2. Click 'login' button
C001_02	Email: syafiqah@gmail.com Password: syaf123 (incorrect data)	
C001_03	Email: (left blank) Password: syafiqah	
C002_01	Jumlah Pelajar: 35	1. Click 'Dashboard' Menu

C003_01	<u>New Student (correct data)</u> No Sijil Lahir: AI27414 Nama Pelajar: Amani Rosa Binti Aman Alamat: Taman Mawar, Melaka Tarikh Lahir: 16/10/2012 Jantina: Perempuan Nama Penjaga: Aman Shah Bin Razak No. Tel Penjaga: 01126250982 Kewarganegaraan: Warganegara Gambar: student8.jpg	1. Click 'Pelajar Menu 2. Click 'Senarai Pelajar' 3. Fill in the required field with given data 4. Click 'Add/Update/Delete' icon button
C003_02	<u>New Student</u> No Sijil Lahir: AI27414 Nama Pelajar: Amani Rosa Binti Aman Alamat: Taman Mawar, Melaka Tarikh Lahir: 16/10/2012 Jantina: Perempuan Nama Penjaga: Aman Shah Bin Razak No. Tel Penjaga: 01126250982 Kewarganegaraan: Warganegara Gambar: (left blank)	
C003_03	<u>New Student</u> No Sijil Lahir: AI27414 Nama Pelajar: Amani Rosa Binti Aman Alamat: Taman Mawar, Melaka Tarikh Lahir: 16/10/2012 Jantina: Perempuan Nama Penjaga: Aman Shah Bin Razak No. Tel Penjaga: 01126250982 Kewarganegaraan: Warganegara Gambar: images (6).jpg <u>UPDATE</u>	
C003_04	<u>New Student</u> No Sijil Lahir: AI27414 Nama Pelajar: Amani Rosa Binti Aman Alamat: Taman Mawar, Melaka Tarikh Lahir: 16/10/2012 Jantina: Perempuan Nama Penjaga: Aman Shah Bin Razak No. Tel Penjaga: 01126250982 Kewarganegaraan: Warganegara Gambar: images (6).jpg <u>DELETE</u>	
C004_01	<u>Register class (correct data)</u>	1. Click 'Pelajar' Menu

	Sijil Lahir: AI23456 Nama Pelajar: Muhammad Hariz Bin Yusri Pilih Kelas: 3 Ibnu Qayyim Pilih Tahun: 2021	2. Click 'Daftar Kelas' Menu 3. Fill in the requires field with given data 3. Click ''Senarai Pelajar' and choose Year. 4. Click 'View/Delete' icon button
C004_02	Register class Sijil Lahir: (left blank) Nama Pelajar: Muhammad Hariz Bin Yusri Pilih Kelas: 3 Ibnu Qayyim Pilih Tahun: 2021	
C004_03	Student data in class No Sijil Lahir: AI23456 Nama Pelajar: Muhammad Hariz Bin Yusri Alamat: Taman Tangga Batu, Melaka Jantina: Lelaki Nama Penjaga: Yusri Bin Yasin No. Tel Penjaga: 0176543829 Kewarganegaraan: Warganegara Gambar: images.jpg DELETE	
C005_01	Report Student by Gender Perempuan: 16 Lelaki: 19	1. Click 'Laporan' Menu 2. Click 'Jumlah Pelajar Mengikut Jantina' 3. Click 'Jumlah Pelajar Mengikut Kewarganegaraan'
C005_02	Report Student by Citizenship Warganegara: 32 Bukan Warganegara: 3	
C006_01	Update profile details New Password: 12345	1. Click on 'Profil' 2. Click 'Kemaskini Profil' 3. Fill in the required field with given data 4. Click 'Kemaskini' button

6.4.2.3 Test Data for SPBT Teacher

Table 6.9: Test Data for SPBT Teacher

Test Case ID	Description	Step
T001_01	Email: haslinda@gmail.com Password: haslinda	1. Fill in the required field with given data. 2. Click 'login' button
T001_02	Email: haslinda @gmail.com Password: linda123 (incorrect data)	
T001_03	Email: (left blank) Password: haslinda	
T002_01	Dashboard (Reporting) Jumlah Pelajar: 35 Jumlah Jenis Buku Teks: 72 Jumlah Perolehan Buku Teks: 62 Jumlah Buku Teks Dipinjam: 2 Jumlah Buku Teks yang Tersedia: 58	1. Click 'Dashboard' Menu 2. Scroll down
T003_01	View Student No Sijil Lahir: SD16090 Nama Pelajar: Munirah Binti Ramli Alamat: Taman Tangga Batu, Melaka Jantina: Perempuan Nama Penjaga: Ramli Bin Adeeb No. Tel Penjaga: 01145678902 Kewarganegaraan: Warganegara Gambar: images.jpg	1. Click 'Pelajar' Menu 2. Click 'Senarai Pelajar' 3. Choose Year 4. Click at the icon to view data
T004_01	New Textbook (correct data) Kategori: Tahun 2 Judul Buku: Matematik Harga Buku: 9.50 Kod Buku: M02 Jenis Edisi: Biasa Penerbit: Sasbadi Sdn. Bhd. Tahun Terbitan: 2020 Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab	1. Click 'Buku Teks' Menu 2. Click 'Senarai Buku Teks' 3. Fill in the required field with given data 4. Click 'Add/Update' icon button
T004_02	New Textbook Kategori: Tahun 2 Judul Buku: Matematik Harga Buku: 9.50 Kod Buku: M02 Jenis Edisi: Biasa	

	Penerbit: Sasbadi Sdn. Bhd. Tahun Terbitan: (left blank) Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab	
T004_03	<u>New Textbook (correct data)</u> Kategori: Tahun 2 Judul Buku: Matematik Harga Buku: 9.50 Kod Buku: M02 Jenis Edisi: Biasa Penerbit: Dewan Pustaka Dan Bahasa Tahun Terbitan: 2020 Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab <u>UPDATE</u>	
T005_01	<u>Perolehan Details (correct data)</u> Kod Buku: BM01 Kategori: Tahun 1 Judul Buku: Bahasa Melayu No Perolehan: 01/04/2018 Tarikh Terima: 2018-02-05 Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab	1. Click 'Buku Teks' Menu 2. Click 'Perolehan' 3. Fill in the required field with given data 4. Click 'Add/Update/Delete' icon button
T005_02	<u>New Perolehan (correct data)</u> Kod Buku: BM06 Judul Buku: Bahasa Melayu Tarikh Terima: 2021-02-27 No Perolehan: 01/01/2021 Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab	
T005_03	<u>New Perolehan</u> Kod Buku: BM06 Judul Buku: Bahasa Melayu Tarikh Terima: 2021-02-27 No Perolehan: (left blank) Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab	
T005_04	<u>New Perolehan</u> Kod Buku: BM06 Judul Buku: Bahasa Melayu Tarikh Terima: 2021-02-27 No Perolehan: 01/01/2021	

	Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab Status: Tersedia/Dilupuskan UPDATE	
T005_05	New Perolehan Kod Buku: BM06 Judul Buku: Bahasa Melayu Tarikh Terima: 2021-02-27 No Perolehan: 01/01/2021 Nama Guru Penerima: Nur Haslinda Binti Abdul Wahab Status: Tersedia DELETE	
T006_01	Disposal Details (correct data) Kod Buku: ICT06 Judul Buku: Teknologi Maklumat dan Komunikasi No Perolehan: 01/01/2016 Jenis Edisi: Biasa Sebab Lupus: Usang	1. Click 'Buku Teks' Menu 2. Click 'Pelupusan' 3. Fill in the required field with given data 4. Click 'Update/Delete' icon button
T006_02	Disposal Details Kod Buku: ICT06 Judul Buku: Teknologi Maklumat dan Komunikasi No Perolehan: 01/01/2016 Jenis Edisi: Biasa Sebab Lupus: Rosak UPDATE	
T006_03	Disposal Details Kod Buku: ICT06 Judul Buku: Teknologi Maklumat dan Komunikasi No Perolehan: 01/01/2016 Jenis Edisi: Biasa Sebab Lupus: Rosak DELETE	
T007_01	New Publishing House (correct data) Nama Penerbit: Dewan Pustaka Dan Bahasa Alamat Penerbit: Pejabat, Menara Dbp, Jalan Dewan Bahasa, 50460 Kuala Lumpur Negeri: W.P. Kuala Lumpur	1. Click 'Buku Teks' Menu 2. Click 'Senarai Penerbit Buku Teks 3. Fill in the required field with given data

	No. Telefon: 03-21479000 Email: dpbkl@gmail.com	4. Click 'Add/Update' icon button
T007_02	New Publishing House Nama Penerbit: Dewan Pustaka Dan Bahasa Alamat Penerbit: Pejabat, Menara Dbp, Jalan Dewan Bahasa, 50460 Kuala Lumpur Negeri: W.P. Kuala Lumpur No. Telefon: 03-21479000 Email: (left blank)	
T007_03	New Publishing House (correct data) Nama Penerbit: Dewan Pustaka Dan Bahasa Alamat Penerbit: Pejabat, Menara Dbp, Jalan Dewan Bahasa, 50460 Kuala Lumpur Negeri: W.P. Kuala Lumpur No. Telefon: 03-21479000 Email: dpbkl65@gmail.com UPDATE	
T008_01	Search Perolehan Kod Buku: BA02 Judul Buku: Bahasa Arab No. Perolehan: 04/05/2019 Status: Tersedia	1. Click 'Transaksi' Menu 2. Click 'Cari Perolehan' 3. Click 'Pinjam Buku Teks' button to borrow book 4. Fill in the required field with given data 5. Click 'Pinjam Buku' button
T008_02	Borrow Textbook (correct data) No. Perolehan: 04/05/2019 Kod Buku: BA02 Judul Buku: Bahasa Arab Sijil Lahir: UL09228 Nama Pelajar: Nurul Jannah Binti Hakim Tarikh Pinjam: 27/08/2021	
T008_03	Borrow Textbook No. Perolehan: 04/05/2019 Kod Buku: BA02 Judul Buku: Bahasa Arab Sijil Lahir: (left blank)	

	Nama Pelajar: Nurul Jannah Binti Hakim Tarikh Pinjam: 27/08/2021	
T009_01	List of Borrowed Textbook No. Perolehan: 01/01/2020 Nama Pelajar: Jasri Bin Kamal Tarikh Pinjam: 2021-06-09 Status Pulang: Dipinjam Tarikh Pulang: 0000-00-00 Sebab Denda: Tiada Bayaran Denda (RM): 0.00 Tindakan: Pulang Buku (button)	1. Click 'Transaksi' Menu 2. Click 'Senarai Pinjaman' 3. Click 'Pulang Buku' button to return book 4. Fill in the required field with given data 5. Click 'Pulang Buku' button
T009_02	Return Textbook No. Perolehan: 01/01/2020 Kod Buku: PM03 Tarikh Pulang: 27/08/2021 Status: Dipulangkan Sebab Denda: Tiada/Hilang	
T009_03	Return Textbook No. Perolehan: 01/01/2020 Nama Pelajar: Jasri Bin Kamal Tarikh Pinjam: 2021-06-09 Status Pulang: Dipinjam Tarikh Pulang: 2021-08-27 Sebab Denda: Hilang Bayaran Denda (RM): 10.00 Tindakan: Buku ini telah dipulangkan.	
T010_01	Fine Record (Payment) No. Sijil Lahir: SA89101 Nama Pelajar: Jasri Bin Kamal Bayaran Denda (RM): 10.00 Status: Belum Selesai Tarikh Dibayar: NULL Bayar (button)	1. Click 'Transaksi' Menu 2. Click 'Rekod Denda 3. Click 'Bayar' button to pay fine 4. Fill in the required field with given data 5. Click 'Kemaskini' button 6. Click 'Resit' button to print receipt
T010_02	Fine Record (Change Status Payment) No. Sijil Lahir: SA89101 Nama Pelajar: Jasri Bin Kamal Tarikh Dibayar: 27/08/2021 Bayaran Denda (RM): 10.00 Status: Selesai	
T010_03	Fine Record (Status Payment) No. Sijil Lahir: SA89101 Nama Pelajar: Jasri Bin Kamal Tarikh Dibayar: 27/08/2021	

	Bayaran Denda (RM): 10.00 Status: Selesai Sudah Dibayar Resit (button)	
T011_01	Report List of Textbook Tahun 1 Subjek: Bahasa Melayu Bil. Perolehan: 4 Bil. Hilang: 0 Bil. Lupus: 1 Bil. Baki: 3	1. Click 'Laporan' Menu 2. Click 'Senarai Buku' and choose Grade 3. Click 'Laporan Pelupusan' 4. Click 'Laporan Senarai Pinjaman' 5. Click 'Laporan Rekod Denda'
T011_02	Disposal Report Usang: 1 Rosak: 2 Null: 59	
T011_03	Report List of Borrowed Textbook Dipulangkan: 17 Dipinjam: 3	
T011_04	Report Fine Record Selesai: 13 Belum Selesai: 7	
T012_01	Update profile details New Password: 12345	1. Click on 'Profil' 2. Click 'Kemaskini Profil' 3. Fill in the required field with given data 4. Click 'Kemaskini' button

6.5 Test Results and Analysis

The result of the test case defined in 6.4 Test Design is recorded in this section.

6.5.1 Test Result for Administrator

Table 6.10: Test Result for Administrator

Test Case ID	Actual Result	Result (Success/Fail)
AD001_01	System will direct the user into Admin Home Page	Success

AD001_02	System will display an error message of 'Email or Password wrong, Please re-login.'	Success
AD001_03	System will ask user to fill out the blank field	Success
AD002_01	System will display the total number of teacher, student, textbook and pie chart for total staff by position	Success
AD003_01	System will display list of teachers	Success
AD003_02	System will display message of 'Upload data success !'	Success
AD003_03	System will ask user to fill out the blank field	Success
AD003_04	System will display message of 'Update data success !'	Success
AD004_01	System will backup the database locally	Success
AD004_02	System will restore the backup database chosen	Success

6.5.2 Test Result for Class Teacher

Table 6.11: Test Result for Class Teacher

Test Case ID	Actual Result	Result (Success/Fail)
C001_01	System will direct the user into Class Teacher Home Page	Success
C001_02	System will display an error message of 'Email or Password wrong, Please re-login.'	Success
C001_03	System will ask user to fill out the blank field	Success
C002_01	System will display the total number of student	Success
C003_01	System display list of students	Success
C003_02	System will display message of 'Upload data success !'	Success
C003_03	System will ask user to fill out the blank field	Success
C003_04	System will display message of 'Update data success !'	Success

C003_05	System will display message of 'Delete data success !'	Success
C003_06	System will display 'Daftar kelas berjaya direkod!'	Success
C003_07	System will display list of students by year chosen	Success
C003_08	System will display message of 'Delete data success !'	Success
C004_01	System will display the total number of student by gender	Success
C004_02	System will display the total number of student by citizenship	Success
C005_01	System will display personal details	Success
C005_02	System will display message of 'Update data success !'	Success

6.5.3 Test Result for SPBT Teacher

Table 6.12: Test Result for SPBT Teacher

Test Case ID	Actual Result	Result (Success/Fail)
T001_01	System will direct the user into Teacher SPBT Home Page	Success
T001_02	System will display an error message of 'Email or Password wrong, Please re-login.'	Success
T001_03	System will ask user to fill out the blank field	Success
T002_01	System will display the total number of student, textbook, perolehan, borrowed book, and available perolehan textbook	Success
T003_01	System will direct user to chosen year page	Success
T003_02	System will display list of students by year chosen	Success
T004_01	System display list of textbooks	Success
T004_02	System will display message of 'Upload data success !'	Success
T004_03	System will ask user to fill out the blank field	Success

T004_04	System will display message of ‘Update data success !’	Success
T004_05	System will direct export file	Success
T005_01	System display list of perolehan	Success
T005_02	System will display message of ‘Upload data success !’	Success
T005_03	System will ask user to fill out the blank field	Success
T005_04	System will display message of ‘Update data success !’	Success
T005_05	System will display message of ‘Delete data success !’	Success
T006_01	System display list of disposal textbook	Success
T006_02	System will display message of ‘Update data success !’	Success
T006_03	System will display message of ‘Delete data success !’	Success
T007_01	System display list of publishing house	Success
T007_02	System will display message of ‘Upload data success !’	Success
T007_03	System will ask user to fill out the blank field	Success
T007_04	System will display message of ‘Update data success !’	Success
T008_01	System display list of perolehan textbook	Success
T008_02	System will display message of ‘Data pinjaman baru berjaya direkod!’	Success
T008_03	System display list of borrowed textbook	Success
T008_04	System will display message of ‘Pemulangan Buku Berjaya !’	Success
T008_05	System display list of fine record	Success
T008_06	System will display message of ‘Update data success !’	Success
T008_07	System will display the receipt	Success
T008_08	System will direct to print the receipt	Success
T009_01	System will direct user to grade chosen	Success
T009_02	System will display list of textbook with total of perolehan, total of textbook missing, total of disposal textbook, and remaining of textbook left	Success
T009_03	System will display the total of disposal textbook by reason	Success

T009_04	System will display the total borrowed textbook by status	Success
T009_05	System will display the total of fine record by fine status	Success
T010_01	System will display personal details	Success
T010_02	System will display message of 'Update data success !'	Success

6.6 Conclusion

In conclusion, this chapter focuses on the project's testing phase. Unit testing, integration testing, and system testing are the three layers of testing. The test methods used to test the project are static testing and dynamic testing. Black box testing and white box testing are two types of dynamic testing. Every single test case written for all modules is successfully executed on the first try. This project will come to a close in the next episode.

CHAPTER 7: CONCLUSION

7.1 Introduction

The SPBT Management System is expected to be the result of a final year project completed as part of a bachelor's degree requirement. It is a type of system that combines a variety of submodules and processes to ensure that the SPBT management, business operations, and data are all organized efficiently.

7.2 Observation on Weaknesses and Strengths

The proposed system has a fault in that its utility is limited. In a real SPBT Management System, there should be dozens of other submodules. Transaction of textbooks for teacher's edition, inventory control, invoicing management, and other submodules are examples of typical submodules that should be included in this system.

The system's basic functionality contributes to a more efficient SPBT management system. The system assists in the management of teacher information, student information, textbook information, and textbook transactions. The teachers can manage students' information and the transaction of textbook made by students such as borrow, return and fine record in one system.

7.3 Propositions for Improvement

Based on the weaknesses mentioned in the previous section, SPBT Management System can be improved in many ways. There are few suggestions for improvement of the system. One of the suggestion is a new module such as manage transaction of

textbook for teacher's edition can be introduced into the system. With the new module, the SPBT management can record for borrowing and return teacher's edition textbook made by each teachers in school. Aside from that, the system can be coupled with an inventory management system to maintain track of inventory levels. Inventory management software can prioritise and forecast inventory requirements. Furthermore, barcode or QR code technology can be used to achieve accuracy and effectiveness, and the inserted data will match the system.

7.4 Project Contribution

This project's contribution to textbook management is to enable easy access to data via a well-designed database architecture. Furthermore, data manipulation is simple. Because this technology automates every procedure, the time-consuming manual paperwork is considerably reduced. It increases instructor productivity by avoiding human errors and ensuring compliance. As a result, the teachers will be able to devote more time to other tasks that are more vital.

7.5 Conclusion

The SPBT Management System has been successfully built and implemented. Even for beginner users, the UI is user-friendly and straightforward to operate. It was able to complete the majority of the basic commercial processes in schools. In a nutshell, the goals outlined in 1.3 Objective have been met.

REFERENCES

Hamilton, T. (2021, August 27). *What Is Software Testing? Definition, Basics & Types In Software Engineering*. Guru99. https://www.guru99.com/software-testing-introduction-importance.html.

Fazerra, N. (2019, May). [web log]. https://noorfazerrakps3014.blogspot.com/p/topik-8-pengurusan-skim-pinjaman-buku_2.html.

Judul Dan Penerbit Buku Teks Terbitan Baharu. Judul Dan Penerbit Buku Teks Terbitan Baharu - Mampu. (2021, June 9). https://www.data.gov.my/data/ms_MY/dataset/judul-dan-penerbit-buku-teks-terbitan-baharu/resource/13c34e32-a399-4c2a-b5a2-11a717918a72?inner_span=True.

Pearson, L.T., 2015. *The Four Levels of Software Testing.* [online] Available at: <https://www.seguetech.com/the-four-levels-of-software-testing/> [Accessed 8 Aug. 2020].

Mysql create procedure by practical examples. MySQL Tutorial. (2021, April 26). https://www.mysqltutorial.org/getting-started-with-mysql-stored-procedures.aspx/.

Thomas Connolly. & Carolyn Begg. (2005). *Database System*. Fourth Edition. A Practical Approach to Design, Implementation, and Management.

Jamal, M., & Amalina, N. (1970, January 1). [PDF] *Implementation Of QR Code Technology For Textbook Management System* / Nur AMALINA muhammad Jamal: Semantic Scholar. Implementation of QR code technology for textbook management system. https://www.semanticscholar.org/paper/Implementation-of-QR-code-technology-for-textbook-%2F-Jamal-Amalina/dcd7b84186f4a790a05456a316798e6e6e0772f0b3.

Roger S. Pressman. (2010*). Software Engineering: A Practitioner's Approach*. Seventh Edition. The McGraw-Hill Companies.

William Stallings. (2009). *Operating Systems: Internal and Design Principles.* Sixth Edition.

Sachin, Sharma, R., Mohamed, A., Aryan, M. S., Saurabh, Jeffro, Kuchana, R., Bindu, Akanksha, Singh, Tanbir, Arash, Barath, Ramesh, M., Kannan, Mohit, Prasad, Shivaji, Mahesh, … Deepthi. (2021, August 26). *Database testing Complete GUIDE (Why, what, and how to test data).* Software Testing Help. https://www.softwaretestinghelp.com/database-testing-process/.

APPENDICES

A. DDL Statement to Create Table

- -DDL statement to create table BORROW

```
CREATE TABLE `borrow` (
  `borrowID` int(11) NOT NULL AUTO_INCREMENT,
  `textbookID` int(11) NOT NULL,
  `studentID` int(11) NOT NULL,
  `perolehanID` int(11) NOT NULL,
  `dateBorrowed` date NOT NULL,
  `statusReturn` varchar(50) NOT NULL,
  `returnDate` varchar(50) NOT NULL,
  `teacherID` int(11) NOT NULL,
  `fineDesc` varchar(100) NOT NULL,
  `finePayment` varchar(50) NOT NULL,
  `paymentDate` varchar(50) NOT NULL,
  `fineStatus` varchar(50) NOT NULL DEFAULT 'Belum Selesai',
  PRIMARY KEY (`borrowID`)
) ENGINE=InnoDB AUTO_INCREMENT=50 DEFAULT CHARSET=latin1
```

- -DDL statement to create table CLASS

```
CREATE TABLE `class` (
  `classID` int(11) NOT NULL AUTO_INCREMENT,
  `className` varchar(100) NOT NULL,
  PRIMARY KEY (`classID`)
) ENGINE=InnoDB AUTO_INCREMENT=20 DEFAULT CHARSET=latin1
```

- -DDL statement to create table CLASS_STUDENT

```
CREATE TABLE `class_student` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `studentID` int(11) NOT NULL,
  `classID` int(11) NOT NULL,
  `year` varchar(10) NOT NULL,
  `teacherID` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=utf8mb4
```

- -DDL statement to create table PEROLEHAN

```
CREATE TABLE `perolehan` (
  `perolehanID` int(11) NOT NULL AUTO_INCREMENT,
  `textbookID` int(11) NOT NULL,
  `teacherID` int(11) NOT NULL,
  `dateReceived` date NOT NULL,
  `noPerolehan` varchar(100) NOT NULL,
  `status` varchar(100) NOT NULL,
  `reason` varchar(50) NOT NULL,
  PRIMARY KEY (`perolehanID`)
) ENGINE=InnoDB AUTO_INCREMENT=70 DEFAULT CHARSET=utf8mb4
```


- -DDL statement to create table PUBLISHING HOUSE

```
CREATE TABLE `publishinghouse` (
  `publishingHouseID` int(11) NOT NULL AUTO_INCREMENT,
  `houseName` varchar(100) NOT NULL,
  `houseAddress` varchar(100) NOT NULL,
  `houseCity` varchar(50) NOT NULL,
  `houseContactNo` varchar(15) NOT NULL,
  `houseEmail` varchar(100) NOT NULL,
  PRIMARY KEY (`publishingHouseID`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1
```

- -DDL statement to create table STUDENT

```
CREATE TABLE `student` (
  `studentID` int(11) NOT NULL AUTO_INCREMENT,
  `studentName` varchar(100) NOT NULL,
  `birthCert` varchar(50) NOT NULL,
  `dateOfBirth` varchar(50) NOT NULL,
  `gender` varchar(50) NOT NULL,
  `address` varchar(100) NOT NULL,
  `image` longblob NOT NULL,
  `guardianName` varchar(50) NOT NULL,
  `guardianContactNo` varchar(50) NOT NULL,
  `citizenship` varchar(50) NOT NULL,
  PRIMARY KEY (`studentID`)
) ENGINE=InnoDB AUTO_INCREMENT=50 DEFAULT CHARSET=latin1
```

- -DDL statement to create table TEACHER

```
CREATE TABLE `teacher` (
  `teacherID` int(11) NOT NULL,
  `teacherName` varchar(50) NOT NULL,
  `phoneNo` varchar(15) NOT NULL,
  `icNo` varchar(15) NOT NULL,
  `email` varchar(50) NOT NULL,
  `address` varchar(100) NOT NULL,
  `role` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `status` varchar(50) NOT NULL,
  PRIMARY KEY (`teacherID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```


- -DDL statement to create table TEACHER

```
CREATE TABLE `textbook` (
  `textbookID` int(11) NOT NULL AUTO_INCREMENT,
  `bookTitle` varchar(100) NOT NULL,
  `category` varchar(50) NOT NULL,
  `bookPrice` varchar(100) NOT NULL,
  `bookCode` varchar(50) NOT NULL,
  `bookEditionType` varchar(100) NOT NULL,
  `teacherID` int(11) NOT NULL,
  `publishingHouseID` int(11) NOT NULL,
  `yearPublished` varchar(50) NOT NULL,
```

```
  PRIMARY KEY (`textbookID`)
) ENGINE=InnoDB AUTO_INCREMENT=76 DEFAULT CHARSET=latin1
```

B. Syntax to Create Stored Procedure

- -Syntax to create stored procedure to count total of students

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetTotalStudent`()
BEGIN
DECLARE totalStudent INT DEFAULT 0;

SELECT COUNT(*)
INTO totalStudent
FROM student;

SELECT totalStudent;
END
```

C. Syntax to Create Trigger

- -Syntax to create trigger BEFORE_INSERT in table borrow

```
CREATE  DEFINER=`root`@`localhost`  TRIGGER  `borrow_BEFORE_INSERT`  BEFORE
        INSERT ON `borrow` FOR EACH ROW BEGIN
  declare recent integer;
  set recent := (select count(*) from borrow);
  set new.statusReturn := UPPER(new.statusReturn);
  set new.fineDesc := UPPER(new.fineDesc);
  set new.fineStatus := UPPER(new.fineStatus);

END
```

- -Syntax to create trigger BEFORE_INSERT in table class

```
CREATE  DEFINER=`root`@`localhost`  TRIGGER  `class_BEFORE_INSERT`  BEFORE
        INSERT ON `class` FOR EACH ROW BEGIN
declare recent integer;
  set recent := (select count(*) from class);
  set new.className := UPPER(new.className);
END
```

- -Syntax to create trigger BEFORE_INSERT in table perolehan

```
CREATE DEFINER=`root`@`localhost` TRIGGER `perolehan_BEFORE_INSERT` BEFORE
        INSERT ON `perolehan` FOR EACH ROW BEGIN
declare recent integer;
  set recent := (select count(*) from perolehan);
  set new.status := UPPER(new.status);
END
```

- -Syntax to create trigger BEFORE_INSERT in table publishing house

```
CREATE DEFINER=`root`@`localhost` TRIGGER `publishingHouse` BEFORE INSERT ON
        `publishinghouse` FOR EACH ROW BEGIN
  declare recent integer;
  set recent := (select count(*) from publishingHouse);
  set new.houseName := UPPER(new.houseName);
  set new.houseAddress := UPPER(new.houseAddress);
  set new.houseCity := UPPER(new.houseCity);

END
```

- -Syntax to create trigger BEFORE_INSERT in student

```
CREATE DEFINER=`root`@`localhost` TRIGGER `student_BEFORE_INSERT` BEFORE
        INSERT ON `student` FOR EACH ROW BEGIN
declare recent integer;
  set recent := (select count(*) from student);
  set new.studentName := UPPER(new.studentName);
  set new.birthCert := UPPER(new.birthCert);
  set new.gender := UPPER(new.gender);
  set new.address := UPPER(new.address);
  set new.guardianName := UPPER(new.guardianName);
  set new.citizenship := UPPER(new.citizenship);

END
```

- -Syntax to create trigger BEFORE_INSERT in teacher

```
CREATE DEFINER=`root`@`localhost` TRIGGER `teacher_BEFORE_INSERT` BEFORE
        INSERT ON `teacher` FOR EACH ROW BEGIN
declare recent integer;
  set recent := (select count(*) from teacher);
  set new.teacherName := UPPER(new.teacherName);
  set new.address := UPPER(new.address);
```

```
    set new.role := UPPER(new.role);

END
```

- -Syntax to create trigger BEFORE_INSERT in textbook

```
CREATE  DEFINER=`root`@`localhost`  TRIGGER  `textbook_BEFORE_INSERT`  BEFORE
        INSERT ON `textbook`
 FOR EACH ROW BEGIN
   declare recent integer;
   set recent := (select count(*) from textbook);
   set new.bookTitle := UPPER(new.bookTitle);
   set new.category := UPPER(new.category);
   set new.bookPrice := UPPER(new.bookPrice);
   set new.bookCode := UPPER(new.bookCode);
   set new.bookEditionType := UPPER(new.bookEditionType);

END
```