[HK BOOKSTORE MANAGEMENT SYSTEM]

[CHONG HAO KEAT]

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[HK BOOKSTORE MANAGEMENT SYSTEM]

[CHONG HAO KEAT]

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

[TITLE OF FINAL YEAR PROJECT]

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : chonghakeat Date : 3/7/2021
(CHONG HAO KEAT)

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 (Software Development)] with Honours.

SUPERVISOR : yahyabinibrahim Date : 3/7/2021
(ENCIK YAHYA BIN IBRAHIM)

DEDICATION

I dedicate this project to God Almighty my creator, my strong pillar, my source of inspiration, wisdom, knowledge and understanding. He has been the source of my strength throughout this program and on His wings only have I soared. I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done.

ACKNOWLEDGEMENTS

First and foremost, I would like to express unlimited thanks to Universiti Teknikal Malaysia Melaka (UTeM) and Faculty of Information and Communication Technology for giving me an opportunity to apply the knowledge and skills that I have learnt in this project.

Furthermore, I would like to express my very great appreciation to my supervisor, En. Yahya Bin Ibrahim for his valuable and constructive suggestions during the planning and development of this project. Despite of being extraordinarily busy with his duties, took time out to share his knowledge and gave me plenty of advices during the project.

Last but not least, biggest thanks go to my family and friends for giving me moral and financial support during this period. Hence, I would also like to thank everyone who involves directly or indirectly in the project. I might not have done this project successfully without the help received from the people mentioned above.

ABSTRACT

The purpose of this project is to develop a computerized Bookstore Management System that can be used to create an integrated system which is having better efficiency in term of managing stocks and sales. The problem of bookstore facing now is difficulties in managing stock and sales at the same time. Furthermore, there needs a huge amount of ineffective paper work when all procedures are done using manpower. So, this system is to improve the management aspect by utilizing computerized system to coordinate each function and data. Moreover, to develop a system that is secured which is only can be accessed by authorized user. Thus, this system is designed to assist the user to having a better management such as parent can straight buy their desired books from bookstore while staff and admin can manage their sales for schools and suppliers. As a result, Bookstore Management System had successfully to improve the efficiency of all the works involved in the sales.

ABSTRAK

Tujuan projek ini adalah untuk membangunkan sebuah sistem komputerisasi *Bookstore Management System* yang dapat digunakan sebagai sistem bersepadu yang mempunyai kecekapan yang lebih baik dalam menguruskan stok dan penjualan. Masalah kedai buku yang dihadapi sekarang adalah kesukaran menguruskan stok dan penjualan pada masa yang sama. Tambahan pula, memerlukan sejumlah besar kertas kerja yang tidak berkesan apabila semua prosedur dilakukan menggunakan tenaga kerja. Jadi, sistem ini adalah untuk meningkatkan aspek pengurusan dengan menggunakan sistem berkomputer untuk menyelaraskan setiap fungsi dan data. Lebih-lebih lagi, untuk mengembangkan sistem yang terjamin yang hanya dapat diakses oleh pengguna yang sah. Oleh itu, sistem ini direka untuk membantu pengguna untuk mempunyai pengurusan yang lebih baik seperti ibu bapa boleh terus membeli buku yang mereka inginkan dari kedai buku sementara kakitangan dan pentadbir dapat menguruskan penjualan mereka untuk sekolah dan pembekal. Hasilnya, *Bookstore Management System* berjaya meningkatkan kecekapan semua kerja yang terlibat dalam penjualan.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
DBLC	**-**	**Database Life Cycle**
ERD	**-**	**Entity Relationship Diagram**
DBMS	**-**	**Database Management System**
SQL	**-**	**Structured Query Language**
DFD	**-**	**Data Flow Diagram**
DBLC	**-**	**Database Life Cycle**

CHAPTER 1: INTRODUCTION

1.1 Introduction

The keyword “centralize” means the system keeps all the system’ data in a database so that the record can be retrieved from anywhere. Next, “efficiency” means that the order placed can be sent immediately to prepare order to ensure that the customer can receive their order on time. Lastly, “visualize” means the system can generate report automatically based on the database to give a clearer visualization to the staff.

Every work can be done more easily and efficient using computer. Bookstore Management System is a system to store data for stocks and sales. In this system, staff can access in it to insert, update or delete bookstore’s data while reports and statistics only can be seen by admin. For parents, they can buy books for their children from the bookstore through this system. With this system, every process and procedure can be simplify and shortens the process time.

1.2 Problem Statement

Every day, staff have to count the number of stocks and manage sales by paperwork, this may waste more time compare using PC. Furthermore, it is complicated when they have two separated procedures. So, merging them into one will make staffs’ work more efficient and also can improve the data management quality of the bookstore. Integrated view and function of the system will be much better than separated system.

By creating this system, all the problems stated above may be solve easily.

1.3 Project Objectives

The objectives of this project are:

1. To design an integrated and attractive Bookstore Management System
2. To simplify the process of management of sales
3. To test the efficiency of the proposed system

1.4 Project Scopes

System Users:

- Bookstore Staff
- Bookstore Admin
- Parents

System Modules:

User Module

- Login
- User Verification and Authentication
- Manage the information of all user
- User Interface

Stock Module

- View, Insert and Update the data of books

Sales Module

- Making new transaction to purchase books
- View the record of books purchased

Supplies Module

- Making new transaction to get supply for books
- View the record of books supplied

Statistics Module

- View the statistics of transaction made by bookstore

1.5 Expected Output

Output 1 : Every transaction will be recorded and profits will be calculated.

Output 2 : Parents can buy books without going to school or bookstore just using this system.

Output 3 : Easier management for staff and admin.

1.6 Conclusion

This system is to make everyone's work more systematic and lighten their burden so that they can put more effort in servicing customers. This will benefits staffs and also customers. Staffs can just manage their bookstore in this system while customers can straight buy their desired books from bookstore.

CHAPTER 2: PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

In this chapter, the software and hardware requirements are defined in the early stage of development plans. It will also explain how the system are developed and the method used to develop this project. The software required includes Microsoft Visual Studio and Microsoft Management SQL Server Management Studio.

2.2 Project Methodology

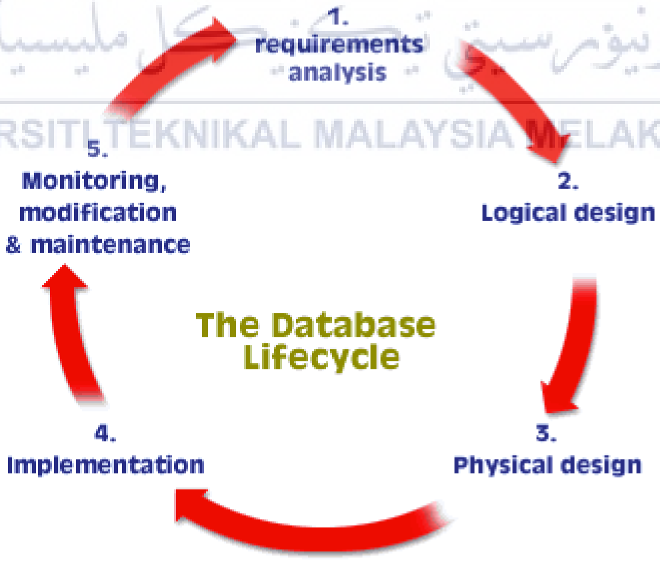


Figure 2.1 : The Database Lifecycle

2.2.1 Requirement Analysis

This is the first and most important stage in the Database Life Cycle. It is the most labor-intensive for the database designer. Here, I decided the topic of my project which is Bookstore Management System. Interviews and observations had been done to collect information and generate the main idea and requirement for this system.

2.2.2 Logical Design

Based on the conceptual data model and a set of mapping rules, every entity and relationship with attributes is converted into relations. Relationships that have attribute groups with data redundancies result in anomalies when adding, updating, or deleting data. Each relation attribute is determined by its data type and domain, including whether the data must be unique or not. The result is a specification for each relation.

2.2.3 Physical Design

Physical database design requires knowledge of the specific DBMS that will be used to implement the database. In the design and definition of physical databases, records organization, file organization, and use of indexes are determined. The goal is to design a data store that provides adequate performance and ensures proper database integrity, security, and recovery. . Thus, physical database design is carried out in coordination with the design of other aspects: programs, computer hardware, operating systems, and data communication networks.

2.2.4 Implementation

After all design had been done completely, data will load into the database. For this system, I used SQL Server as my database. All the data implemented must be match to the data table that created in database. A MVC C# based system is created as application to carry out all the queries and functionality.

2.2.5 Monitoring, Modification and Maintenance

When the database comes into operation, monitoring is carried out to see if performance requirements are being met, whether user expectations increase with demands for better performance. If not, modifications must be made to improve performance. Some regular maintenance activities required include:

- Backup
- Restoration
- Improves performance, adds entities and attributes
- Assignment of access permissions and their maintenance for new and old users
- Database access statistics to improve efficiency and usability of system audits and to monitor system performance.
- Periodic security audits based on system-generated statistics
- Summarize usage of the monthly, quarterly, or yearly system for internal billing or budgeting purposes.

2.3 Project Schedule and Milestones

Table 2.1: Project Milestone

Task	Duration	Start Date	End Date
Preparing Proposal	6	15-03-21	21-03-21
Approving Proposal	12	22-03-21	04-04-21
Project Planning	6	05-04-21	11-04-21
System Planning	7	12-04-21	18-04-21
System Analysing	6	19-04-21	24-04-21
System Design	5	25-04-21	30-04-21
Interface Design	5	01-05-21	05-05-21
System Implementation	11	06-05-21	16-05-21
Admin Module Coding	4	17-05-21	21-05-21
Information Module Coding	2	22-05-21	23-05-21
Attendance Module Coding	2	24-05-21	25-05-21
Result Module Coding	2	26-05-21	27-05-21
Discipline Module Coding	2	01-06-21	02-06-21
System Testing	3	03-06-21	05-06-21
Final Report Writing	6	15-6-21	21-06-21
Presentation and Demonstration	1		
Report Correction	5		
Submission	1		

2.4 Conclusion

The database life cycle consists of four phases, namely: requirements analysis, design, implementation, and maintenance. The requirement analysis specifies relevant data to the users entirely and accurately. The design phases consist of conceptual data modeling, logical design, and physical design. In the implementation phase, a database can be created using SQL or diagrams features existing in the DBMS or tools. The database must be maintained to ensure that evolving information requirements are met.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter will show the system Data Flow Diagram (DFD), Flow Chart and process description which represent and visualize the details of Bookstore Management System.

3.2 Problem Analysis

The existing systems is all in paper work there is no any automation software support system.

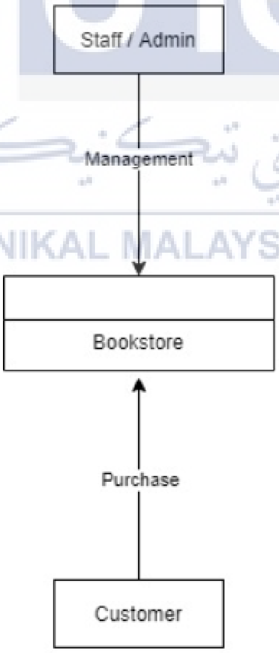


Figure 3.1 : Existing Flow Chart

3.3 The Proposed Solution

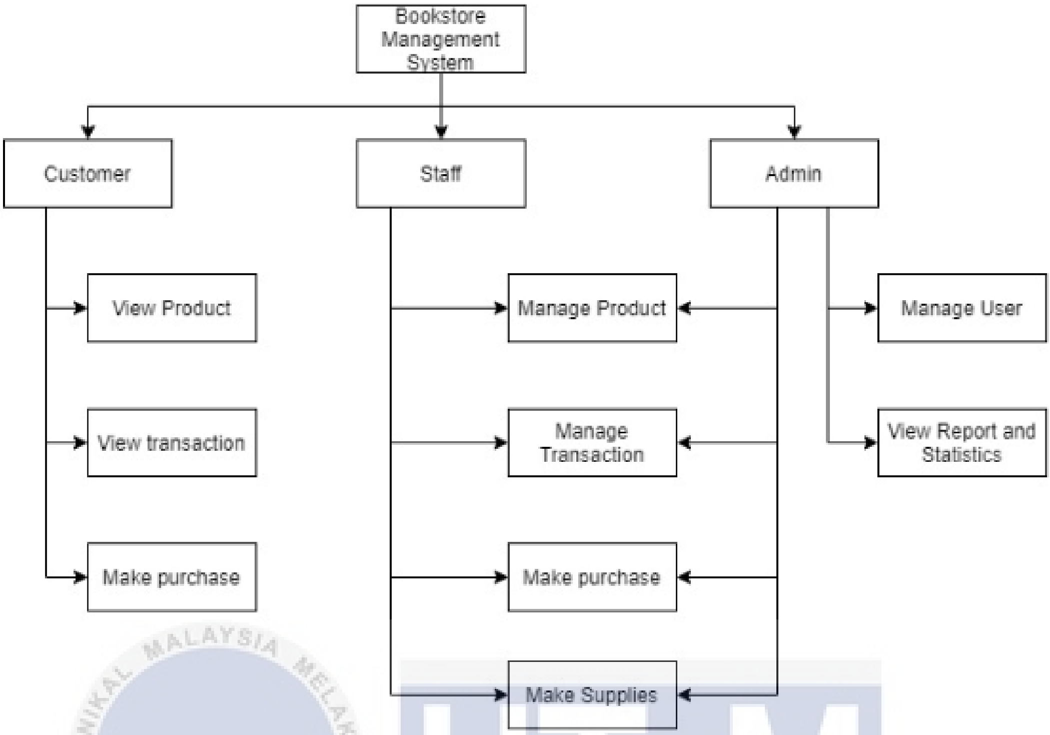


Figure 3.2 : Proposed Flow Chart

3.4 Functional Requirement

Table 3.1 : Functional Requirement

Process	Purpose	Definition
Login	To prevent not authorized user from using the system	Username and password are verified using username and password that registered by users
View Data	To view stock list, transaction, statistics and user.	Showing the content from the system that link from database
Manage product	Insert, Update or Delete product details	Managing spare parts details such as name, Quantity and price
Manage Transaction	View transaction	Staff will manage the transaction for statistics purpose
Generating report	To generate report for top sales.	All data stored can be generate into a report for easy and general view for specific customer

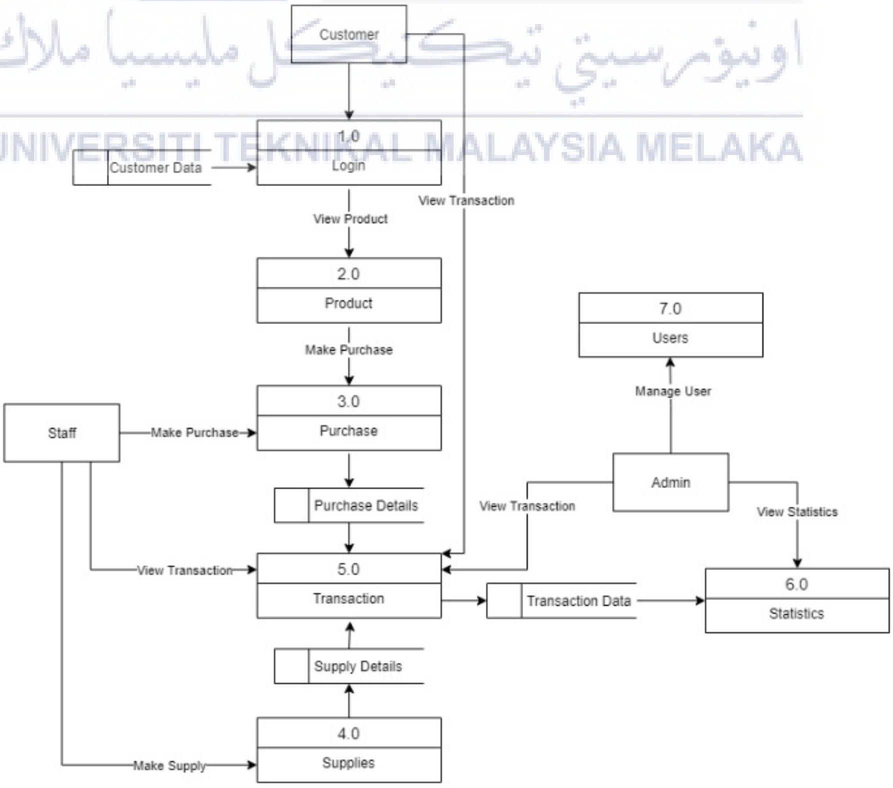


Figure 3.3 : New System DFD

3.5 Conclusion

This chapter is all about the problem analysis and how to overcome those problem to become a better system and shown above by using Data Flow Diagram (DFD), and flow chart.

CHAPTER 4: DESIGN

4.1 Introduction

According to the system development life cycle, system design is the phase after system analysis. This chapter is to explain how the project run, the database of the system and the interface design of the system. It consists of proposal system design, interface design, entity relationship diagram and data dictionary. In proposal system, it describe the module or programming specification. While in interface design, storyboard is illuminated to show the flow of the system. Lastly, entity relationship diagram and data dictionary are representing for the database of the system.

4.2 Database Design

4.2.1 Conceptual Design

ERD

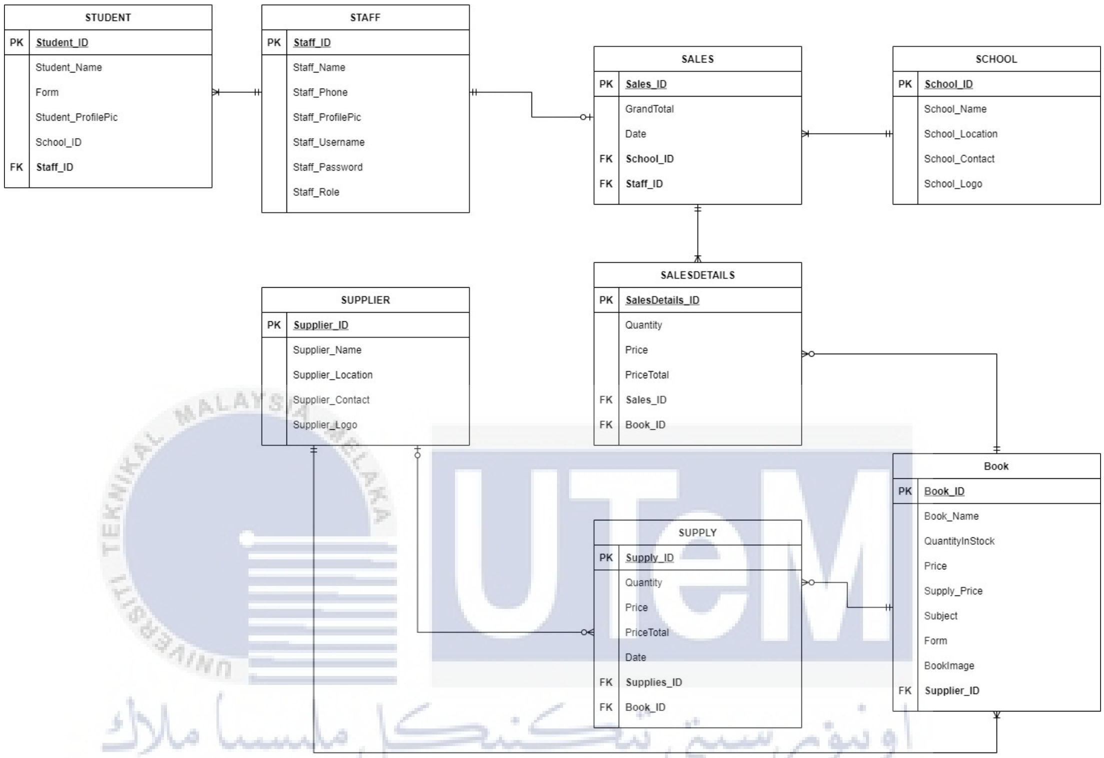


Figure 4.1 : ERD

Business Rule

- One user can have many children while each children only belongs to one user

- Each user can make one or many sales while each sales only can make by one specific user.

- Each supplier can make many supplies while each supply only can make by one supplier.

- Each book can be sales and supply many times and each sales and supply also can contains a lot of books.

4.2.2 Logical Design

Table 4.1 : Table Book

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	book_id	INT		PK	An identical ID for every book	YES
2	book_name	VARCHAR	MAX		Book's name	YES
3	quantity	INT			Book's quantity	YES
4	price	INT			Book's Price	YES
5	sprice	INT			Book's supply price	YES
6	subject	INT			Book's subject	YES
7	form	VARCHAR	MAX		Book's form	YES
8	book_image	VARBINARY	MAX		Book's image	YES
9	supplier_id	INT		FK	An identical ID for every supplier	YES

Table 4.2 : Table Sales

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	sales_id	INT		PK	An identical ID for every sales	YES
2	grandtotal	INT			Sales' total amount	YES
3	salesdate	DATE			Sales' date	YES
4	school_id	INT		FK	An identical ID for every school	YES
5	staff_id	INT		FK	An identical ID for every staff	YES

Table 4.3 : Table SalesDetails

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	salesdetails_id	INT		PK	An identical ID for every sales details	YES
2	quantity	INT			Quantity for single book	YES
3	price	INT			Price for single book	YES
4	pricetotal	INT			Total price for each type of book	YES
5	sales_id	INT		FK	An identical ID for every sales	YES
6	book_id	INT		FK	An identical ID for every school	YES

Table 4.4 : Table School

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	school_id	INT		PK	An identical ID for every school	YES
2	school_name	VARCHAR	MAX		School's name	YES
3	school_location	VARCHAR	MAX		School's location	YES
4	school_contact	INT			School's contact	YES
5	school_logo	VARBINARY	MAX		School's logo	YES

Table 4.5 : Table Supplier

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	supplier _id	INT		PK	An identical ID for every supplier	YES
2	supplier _name	VARCHAR	MAX		Supplier's name	YES
3	supplier _location	VARCHAR	MAX		Supplier's location	YES
4	supplier _contact	INT			Supplier's contact	YES
5	supplier _logo	VARBINARY	MAX		Supplier's logo	YES

Table 4.6 : Table Staff

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	staff_id	INT		PK	An identical ID for every staff	YES
2	staff_name	VARCHAR	MAX		Staff's name	YES
3	staff_phone	INT			Staff's phone	YES
4	staff_profilepic	VARBINARY	MAX		Staff's image	YES
5	staff_username	VARCHAR	MAX		Staff's username	YES
6	staff_password	VARCHAR	MAX		Staff's password	YES
7	staff_role	VARCHAR	MAX		Staff's role	YES

Table 4.7 : Table Student

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	student_id	INT		PK	An identical ID for every student	YES
2	student_name	VARCHAR	MAX		Student's name	YES
3	form	INT			Student's form	YES
4	student_profilepic	VARBINARY	MAX		Student's image	YES
5	staff_id	INT		FK	An identical ID for every staff	YES
6	school_id	INT		FK	An identical ID for every school	YES

Table 4.8 : Table Supply

No	Name	Data Type	Length	Constraint	Description	Mandatory
1	supply_id	INT		PK	An identical ID for every supply	YES
2	book_id	INT		FK	An identical ID for every book	YES
3	supplier_id	INT		FK	An identical ID for every supplier	YES
4	quantity	INT			Amount of specific book	YES
5	price	INT			Price for single book	YES
6	totalprice	INT			Total price for specific book	YES
7	date	DATE			Supply's date	YES

4.2.3 Physical Design

4.2.3.1 Selection of DBMS

For this system, using SQL server as the main database management system is because SQL Server is a database server by Microsoft.

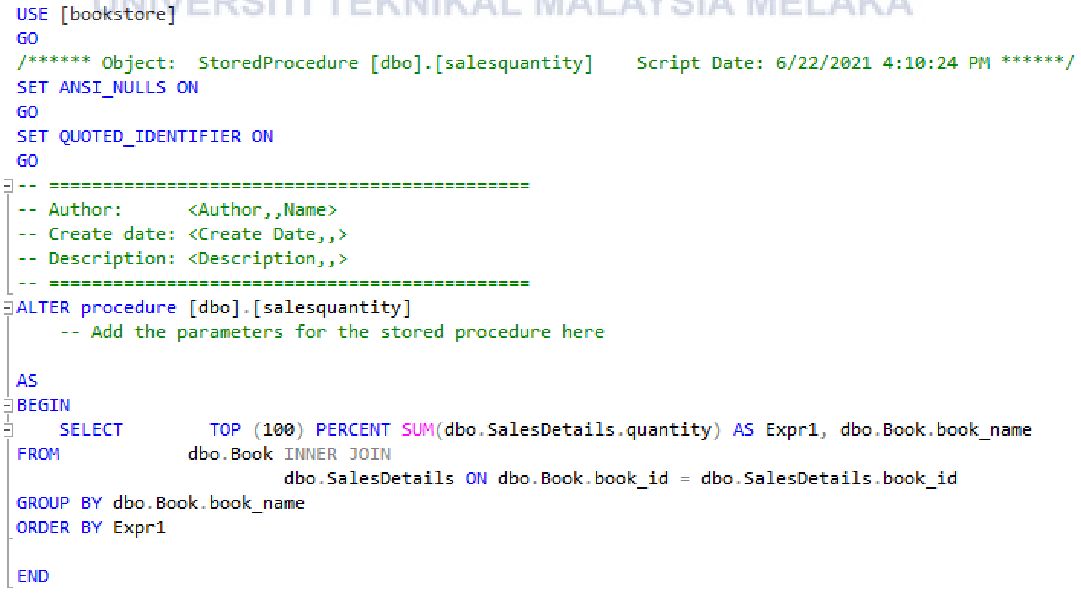
The Microsoft relational database management system is a software product which primarily stores and retrieves data requested by other applications. These applications may run on the same or a different computer.

Going more in-depth, in order to understand what a SQL Server is, you must first understand what SQL is.

SQL is a special-purpose programming language designed to handle data in a relational database management system. A database server is a computer program that provides database services to other programs or computers, as defined by the client-server model. Therefore, a SQL Server is a database server that implements the Structured Query Language (SQL).

4.2.3.2 Stored Procedure

There are two stored procedures that implement in this system.

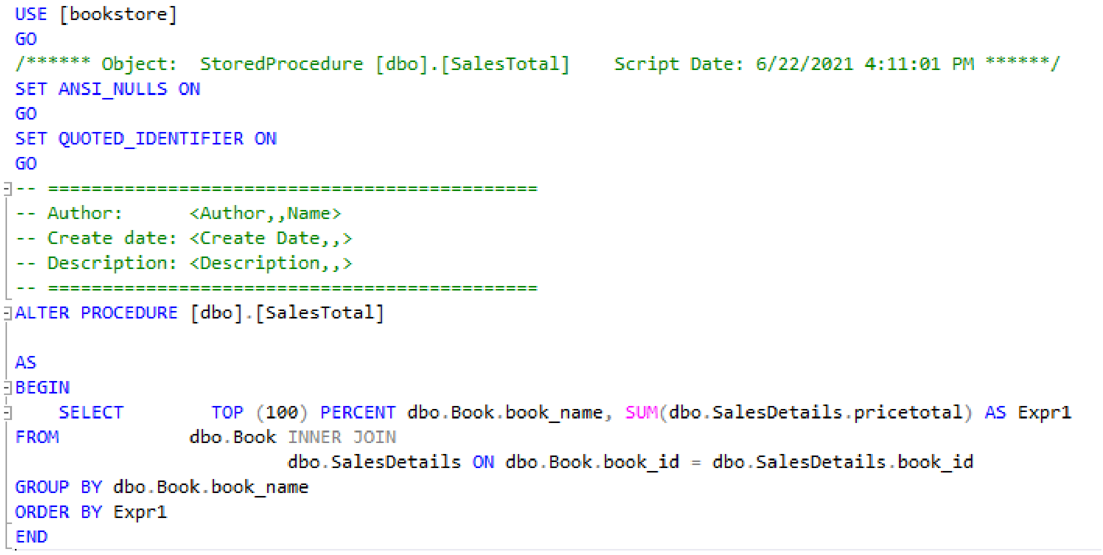


```
USE [bookstore]
GO
/****** Object:  StoredProcedure [dbo].[salesquantity]    Script Date: 6/22/2021 4:10:24 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
ALTER procedure [dbo].[salesquantity]
    -- Add the parameters for the stored procedure here

AS
BEGIN
    SELECT        TOP (100) PERCENT SUM(dbo.SalesDetails.quantity) AS Expr1, dbo.Book.book_name
FROM            dbo.Book INNER JOIN
                         dbo.SalesDetails ON dbo.Book.book_id = dbo.SalesDetails.book_id
GROUP BY dbo.Book.book_name
ORDER BY Expr1

END
```

Figure 4.1 : Stored Procedure 1



```
USE [bookstore]
GO
/****** Object:  StoredProcedure [dbo].[SalesTotal]    Script Date: 6/22/2021 4:11:01 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
ALTER PROCEDURE [dbo].[SalesTotal]

AS
BEGIN
    SELECT        TOP (100) PERCENT dbo.Book.book_name, SUM(dbo.SalesDetails.pricetotal) AS Expr1
FROM            dbo.Book INNER JOIN
                         dbo.SalesDetails ON dbo.Book.book_id = dbo.SalesDetails.book_id
GROUP BY dbo.Book.book_name
ORDER BY Expr1
END
```

Figure 4.2 : Stored Procedure 2

4.3 Graphical User Interface (GUI) Design

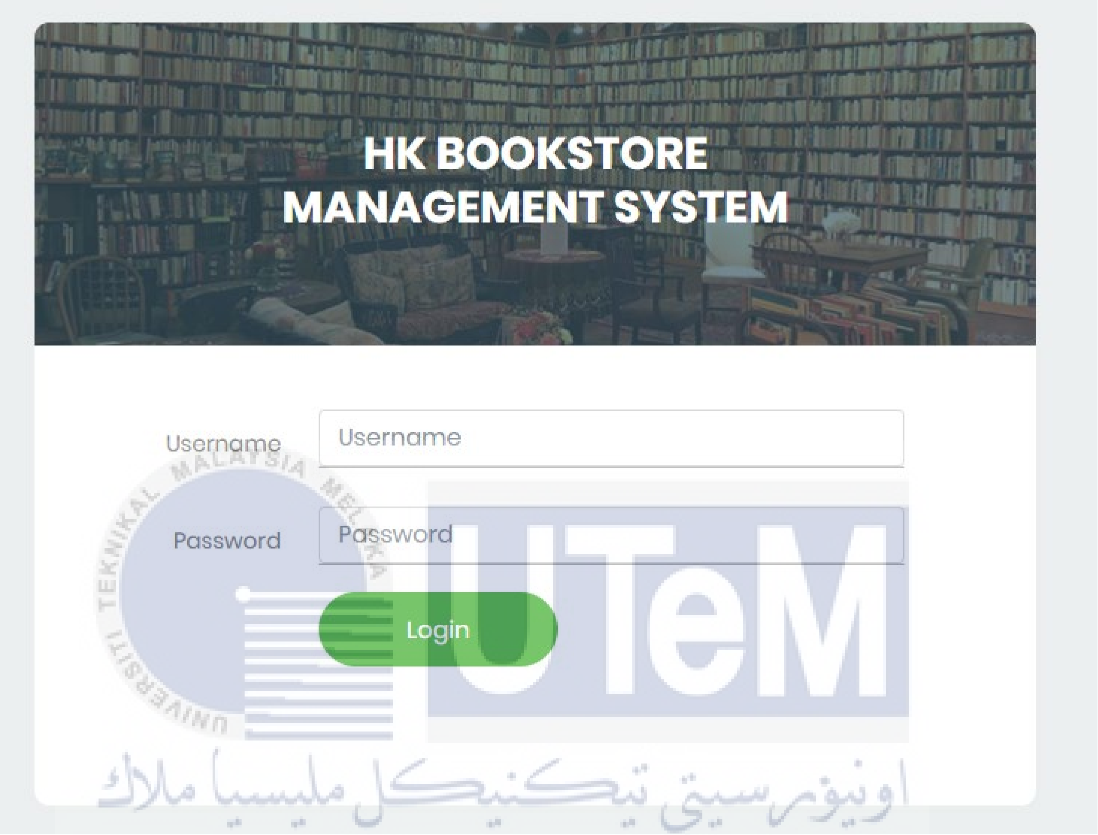
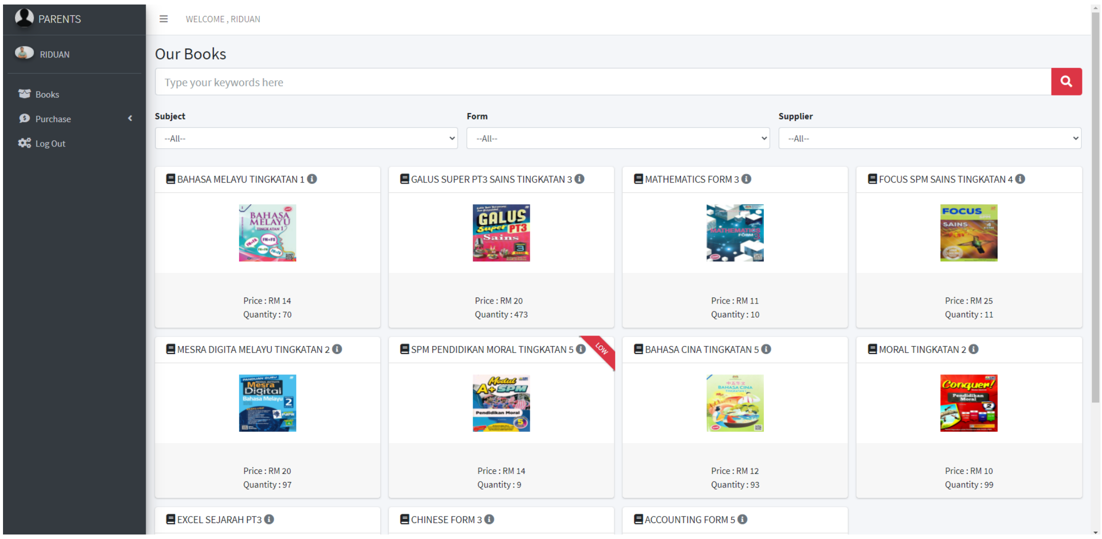


Figure 4.3 : Login



Our Books

Type your keywords here

Subject: --All--

Form: --All--

Supplier: --All--

BAHASA MELAYU TINGKATAN 1
Price : RM 14
Quantity : 70

GALUS SUPER PT3 SAINS TINGKATAN 3
Price : RM 20
Quantity : 473

MATHEMATICS FORM 3
Price : RM 11
Quantity : 10

FOCUS SPM SAINS TINGKATAN 4
Price : RM 25
Quantity : 11

MESRA DIGITA MELAYU TINGKATAN 2
Price : RM 20
Quantity : 97

SPM PENDIDIKAN MORAL TINGKATAN 5
LOW
Price : RM 14
Quantity : 9

BAHASA CINA TINGKATAN 5
Price : RM 12
Quantity : 93

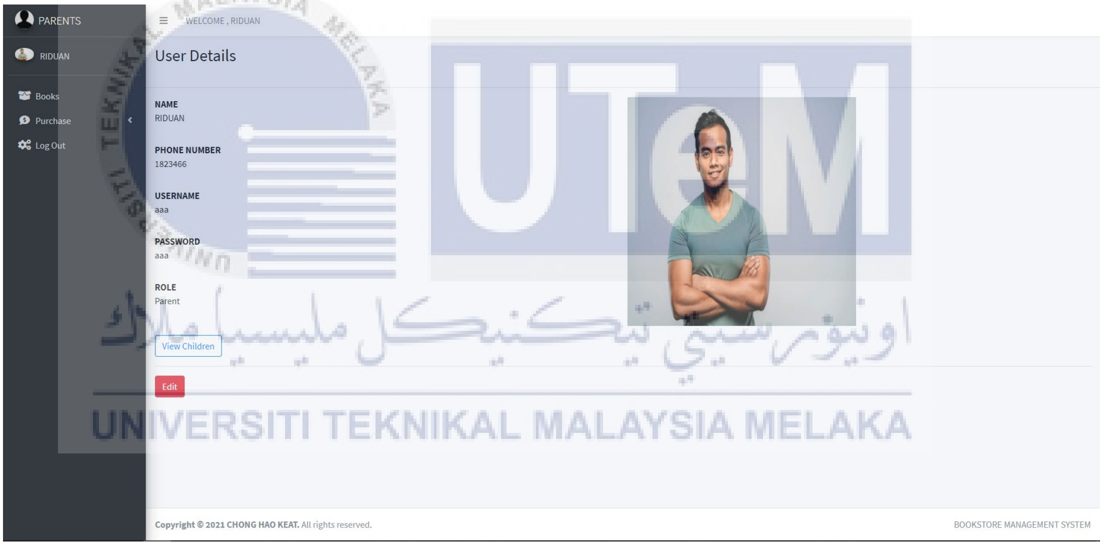
MORAL TINGKATAN 2
Price : RM 10
Quantity : 99

EXCEL SEJARAH PT3

CHINESE FORM 3

ACCOUNTING FORM 5

Figure 4.4 : Customer Home Page and View Stocks



User Details

NAME
RIDUAN

PHONE NUMBER
1823466

USERNAME
aaa

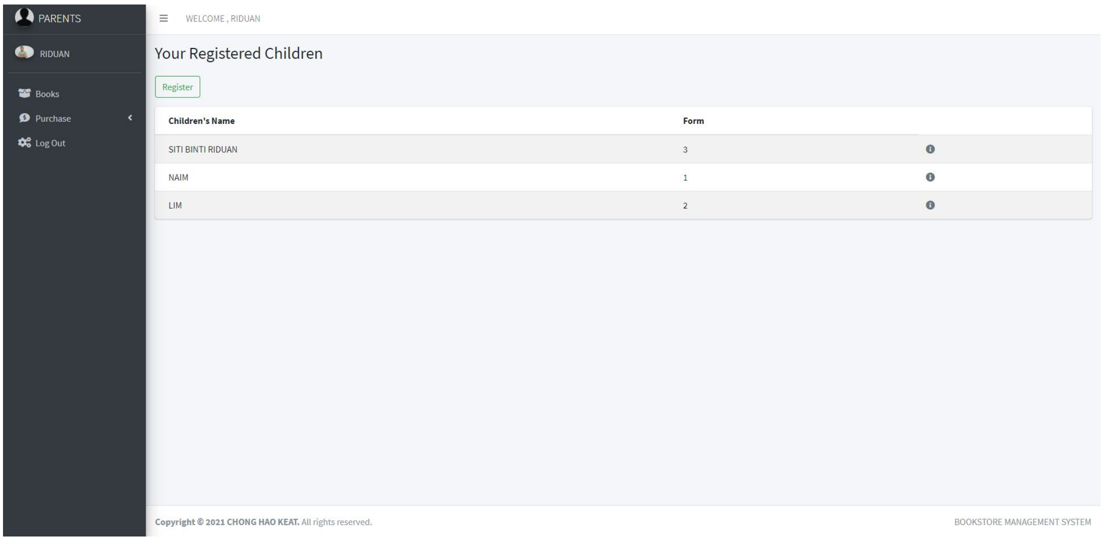
PASSWORD
aaa

ROLE
Parent

Copyright © 2021 CHONG HAO KEAT. All rights reserved.

BOOKSTORE MANAGEMENT SYSTEM

Figure 4.5 : Customer Profile Page



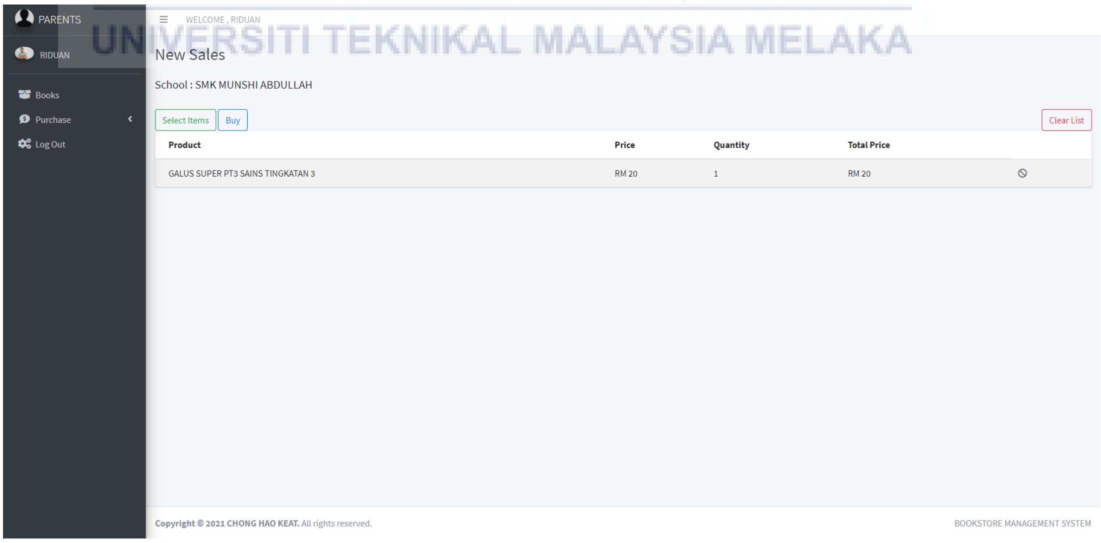
Your Registered Children

Children's Name	Form	
SITI BINTI RIDUAN	3	ℹ
NAIM	1	ℹ
LIM	2	ℹ

Copyright © 2021 CHONG HAO KEAT. All rights reserved.

BOOKSTORE MANAGEMENT SYSTEM

Figure 4.6 : Customer Children Management



New Sales

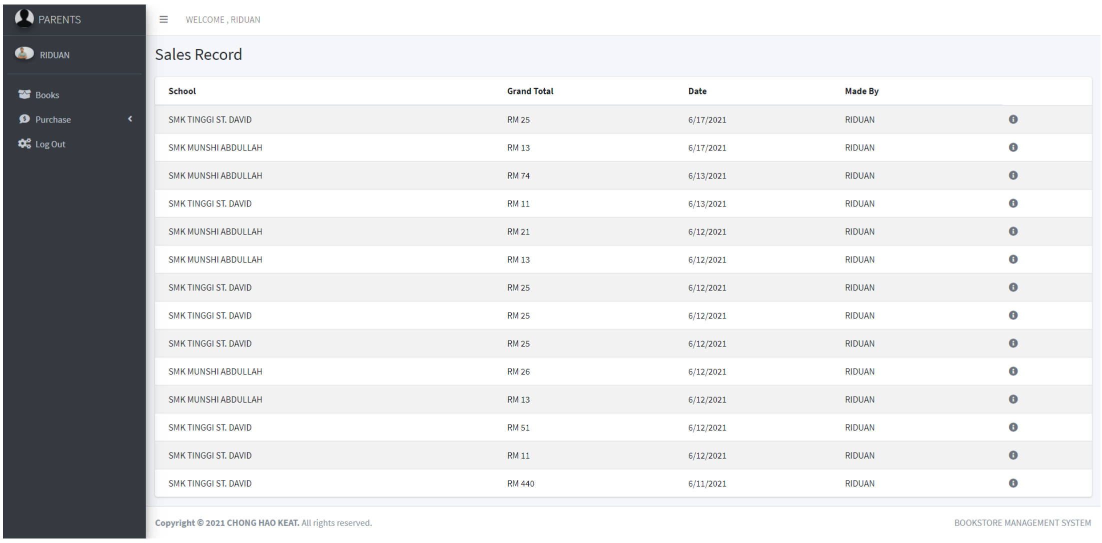
School : SMK MUNSHI ABDULLAH

Product	Price	Quantity	Total Price	
GALUS SUPER PT3 SAINS TINGKATAN 3	RM 20	1	RM 20	⊘

Copyright © 2021 CHONG HAO KEAT. All rights reserved.

BOOKSTORE MANAGEMENT SYSTEM

Figure 4.7 : Customer Purchase

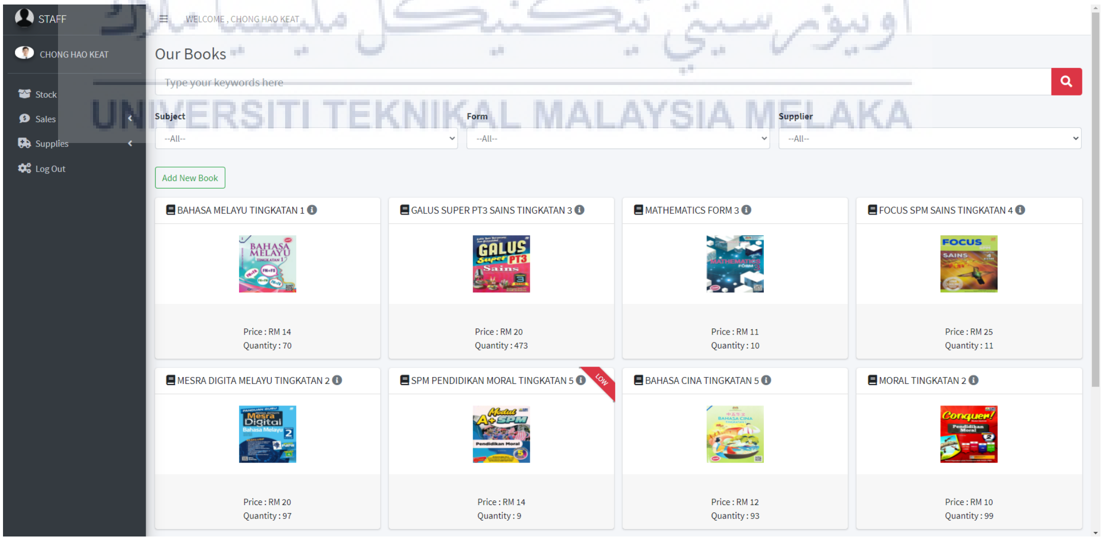


Sales Record

School	Grand Total	Date	Made By	
SMK TINGGI ST. DAVID	RM 25	6/17/2021	RIDUAN	ℹ
SMK MUNSHI ABDULLAH	RM 13	6/17/2021	RIDUAN	ℹ
SMK MUNSHI ABDULLAH	RM 74	6/13/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 11	6/13/2021	RIDUAN	ℹ
SMK MUNSHI ABDULLAH	RM 21	6/12/2021	RIDUAN	ℹ
SMK MUNSHI ABDULLAH	RM 13	6/12/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 25	6/12/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 25	6/12/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 25	6/12/2021	RIDUAN	ℹ
SMK MUNSHI ABDULLAH	RM 26	6/12/2021	RIDUAN	ℹ
SMK MUNSHI ABDULLAH	RM 13	6/12/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 51	6/12/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 11	6/12/2021	RIDUAN	ℹ
SMK TINGGI ST. DAVID	RM 440	6/11/2021	RIDUAN	ℹ

Copyright © 2021 CHONG HAO KEAT. All rights reserved. BOOKSTORE MANAGEMENT SYSTEM

Figure 4.8 : Customer Personal Purchase Record



Our Books

Subject: --All--
Form: --All--
Supplier: --All--

BAHASA MELAYU TINGKATAN 1 — Price : RM 14, Quantity : 70

GALUS SUPER PT3 SAINS TINGKATAN 3 — Price : RM 20, Quantity : 473

MATHEMATICS FORM 3 — Price : RM 11, Quantity : 10

FOCUS SPM SAINS TINGKATAN 4 — Price : RM 25, Quantity : 11

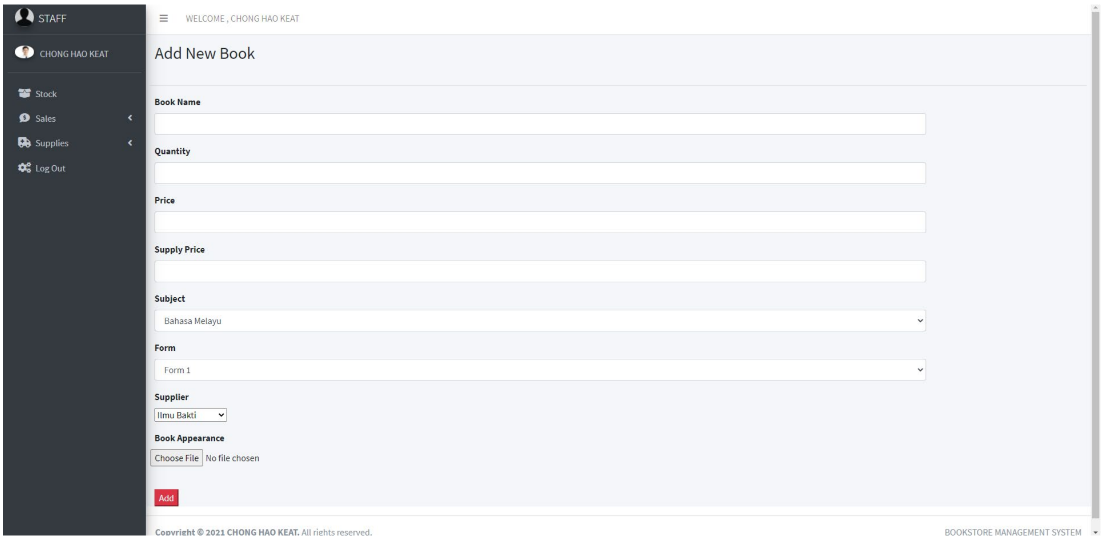
MESRA DIGITA MELAYU TINGKATAN 2 — Price : RM 20, Quantity : 97

SPM PENDIDIKAN MORAL TINGKATAN 5 — Price : RM 14, Quantity : 9

BAHASA CINA TINGKATAN 5 — Price : RM 12, Quantity : 93

MORAL TINGKATAN 2 — Price : RM 10, Quantity : 99

Figure 4.9 : Staff Home Page and Manage Stock



Add New Book

Book Name

Quantity

Price

Supply Price

Subject: Bahasa Melayu

Form: Form 1

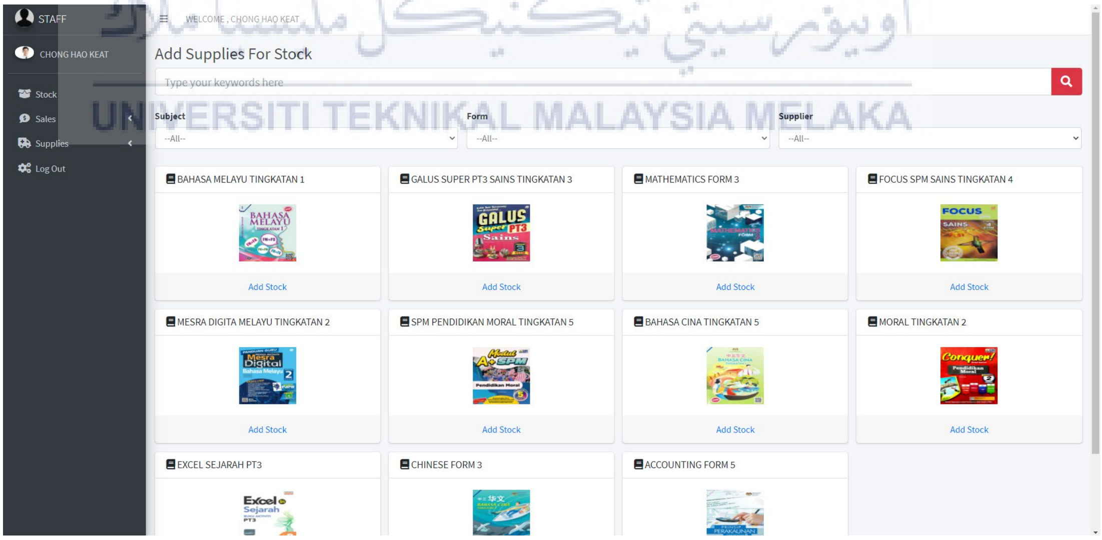
Supplier: Ilmu Bakti

Book Appearance: Choose File No file chosen

Copyright © 2021 CHONG HAO KEAT. All rights reserved.

BOOKSTORE MANAGEMENT SYSTEM

Figure 4.10 : Staff Add New Books



Add Supplies For Stock

Type your keywords here

Subject: --All--

Form: --All--

Supplier: --All--

BAHASA MELAYU TINGKATAN 1 — Add Stock

GALUS SUPER PT3 SAINS TINGKATAN 3 — Add Stock

MATHEMATICS FORM 3 — Add Stock

FOCUS SPM SAINS TINGKATAN 4 — Add Stock

MESRA DIGITA MELAYU TINGKATAN 2 — Add Stock

SPM PENDIDIKAN MORAL TINGKATAN 5 — Add Stock

BAHASA CINA TINGKATAN 5 — Add Stock

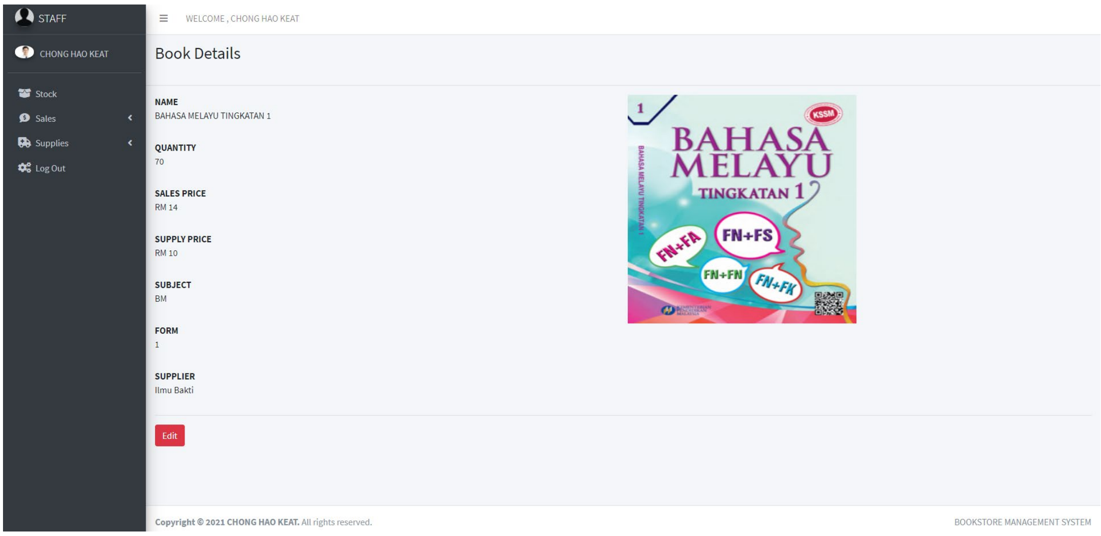
MORAL TINGKATAN 2 — Add Stock

EXCEL SEJARAH PT3

CHINESE FORM 3

ACCOUNTING FORM 5

Figure 4.11 : Staff Make Supplies


STAFF
CHONG HAO KEAT
Stock
Sales
Supplies
Log Out
WELCOME , CHONG HAO KEAT

Book Details

NAME
BAHASA MELAYU TINGKATAN 1

QUANTITY
70

SALES PRICE
RM 14

SUPPLY PRICE
RM 10

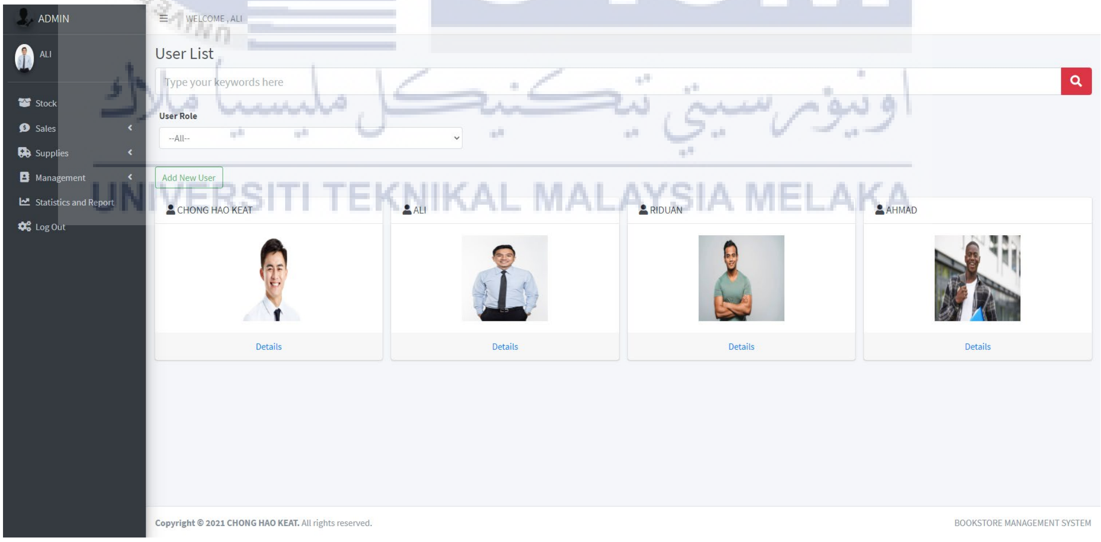
SUBJECT
BM

FORM
1

SUPPLIER
Ilmu Bakti

Edit
Copyright © 2021 CHONG HAO KEAT. All rights reserved.
BOOKSTORE MANAGEMENT SYSTEM

Figure 4.12 : Staff Edit Book Details



User List

Figure 4.13 : User Management

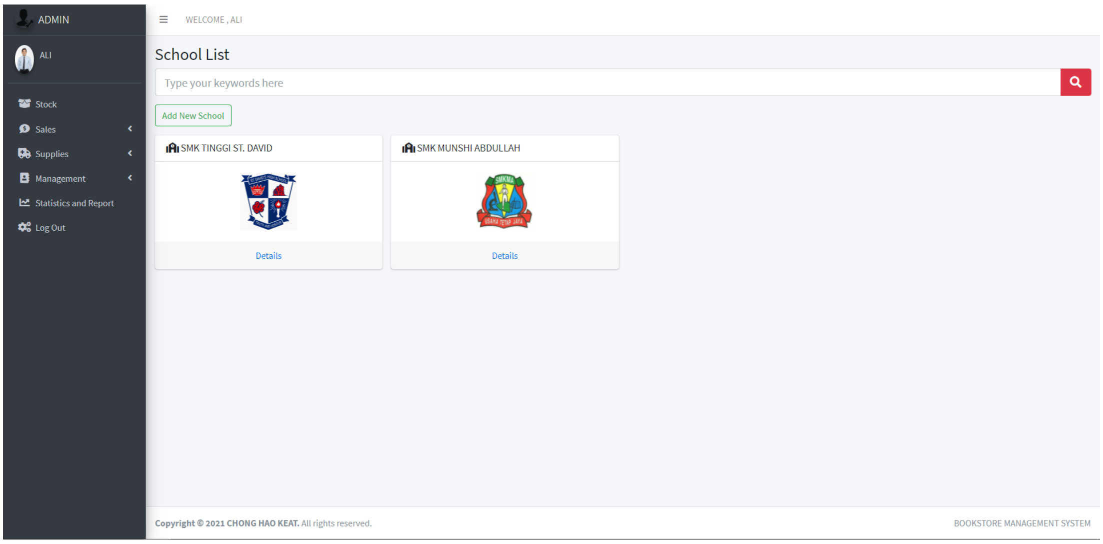


Figure 4.14 : School Management

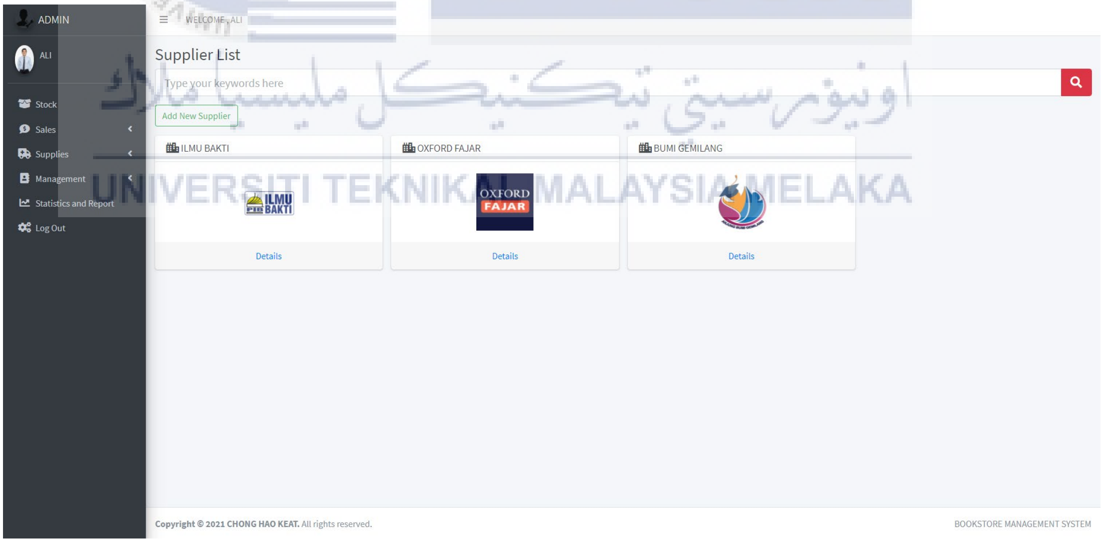
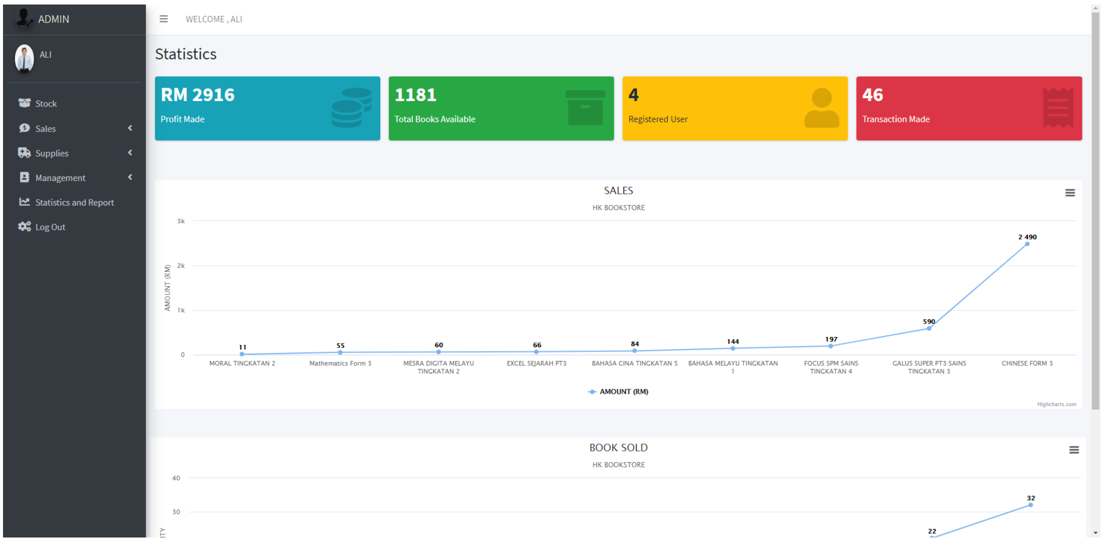


Figure 4.15 : School Management



Statistics

RM 2916 Profit Made

1181 Total Books Available

4 Registered User

46 Transaction Made

Figure 4.16 : Statistics

4.4 Conclusion

This chapter is all about the GUI and the functionality for user.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

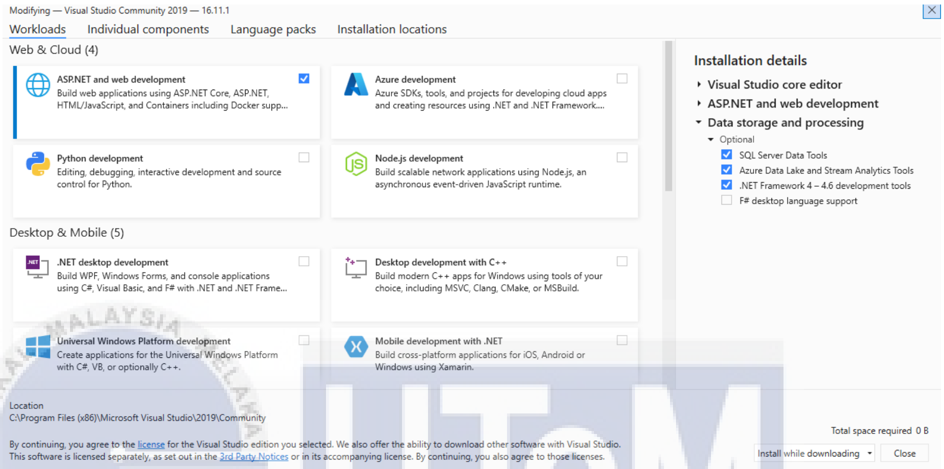
System implementation is a phase where the results of the results of the design will be converted to software code and logical procedure. In this phase, the system will be tested to ensure that development can be carried out successfully and achieve the needs of the users. The implementation phase includes the construction and experimental processes of the working system. It also is the stage where design decisions will be routed to software code and logical procedure. At this stage, the software will be tested to ensure the built-in software meets the requirements standards. In some processes, generally known as test-driven development, tests may be developed just before implementation and serve as a guide for the implementation's correctness.

This chapter will be talk about the implementation phase in this project. The software that I used for the development is Visual Studio 2019 and the database is Microsoft SQL Server Management Studio. In this chapter, software development environment setup and database implementation will be described.

5.2 Software Development Environment Setup

5.2.1 Installation of Visual Studio 2019

Select “ASP.NET and web development” and “Data storage and processing” to start the service for develop a system that using ASP.NET and link the database with SQL server.


Modifying — Visual Studio Community 2019 — 16.11.1
Workloads
Individual components
Language packs
Installation locations
Install while downloading
Close

Web & Cloud (4)

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

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

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

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

Desktop & Mobile (5)

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

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

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

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

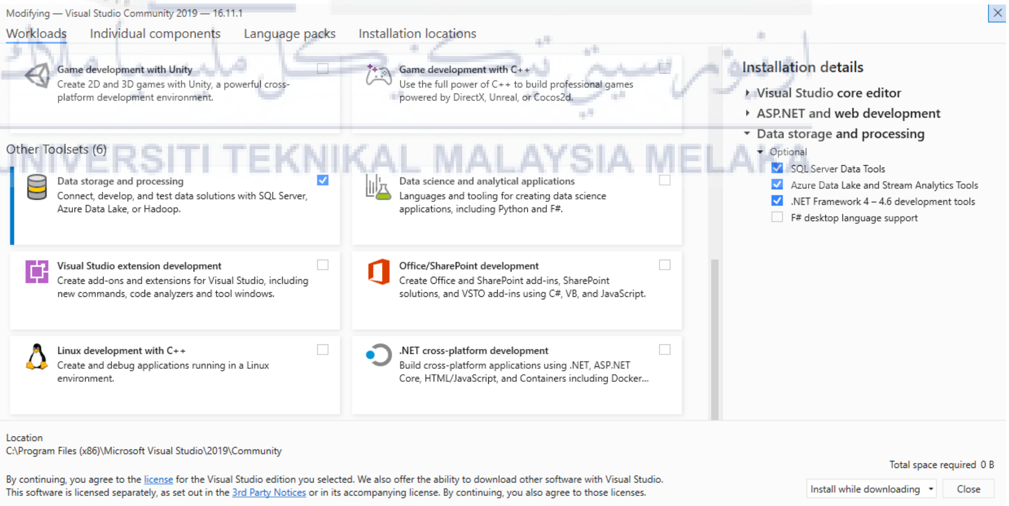
Installation details
- Visual Studio core editor
- ASP.NET and web development
- Data storage and processing
  - Optional
    - ☑ SQL Server Data Tools
    - ☑ Azure Data Lake and Stream Analytics Tools
    - ☑ .NET Framework 4 – 4.6 development tools
    - □ F# desktop language support

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

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

Total space required 0 B

Figure 5.1 : Installation of ASP.NET and Web Development


Modifying — Visual Studio Community 2019 — 16.11.1
Workloads
Individual components
Language packs
Installation locations
Install while downloading
Close

□ Game development with Unity
Create 2D and 3D games with Unity, a powerful cross-platform development environment.

□ Game development with C++
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.

Other Toolsets (6)

☑ Data storage and processing
Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.

□ Data science and analytical applications
Languages and tooling for creating data science applications, including Python and F#.

□ Visual Studio extension development
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.

□ Office/SharePoint development
Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.

□ Linux development with C++
Create and debug applications running in a Linux environment.

□ .NET cross-platform development
Build cross-platform applications using .NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker...

Installation details
- Visual Studio core editor
- ASP.NET and web development
- Data storage and processing
  - Optional
    - ☑ SQL Server Data Tools
    - ☑ Azure Data Lake and Stream Analytics Tools
    - ☑ .NET Framework 4 – 4.6 development tools
    - □ F# desktop language support

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

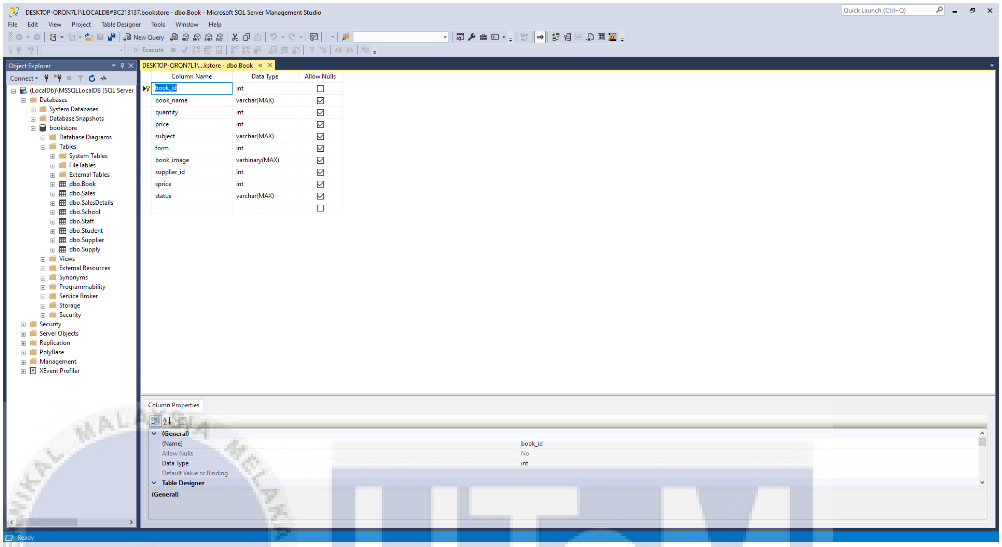
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those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

Total space required 0 B

Figure 5.2 : Installation of Data Storage and Processing

5.2.2 Creation of Database

First we need to create a database in SQL Server and add some tables and other functionality processes such as stored procedure and triggers to make it work smoothly.



Column Name	Data Type	Allow Nulls
book_id	int	□
book_name	varchar(MAX)	☑
quantity	int	☑
price	int	☑
subject	varchar(MAX)	☑
form	int	☑
book_image	varbinary(MAX)	☑
supplier_id	int	☑
sprice	int	☑
status	varchar(MAX)	☑
		□

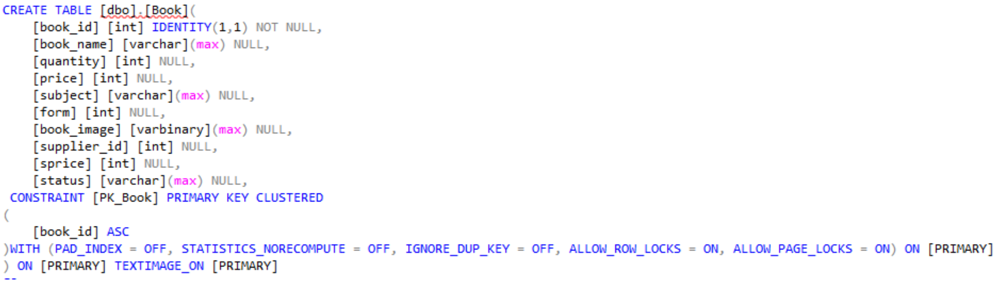
(General)	
(Name)	book_id
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
(General)	

Figure 5.3 : Database in SQL Server

5.3 Database Implementation

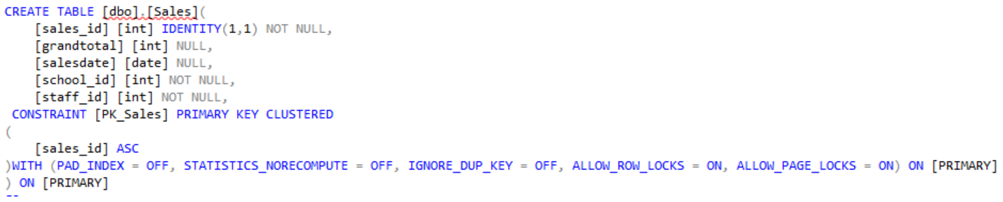
5.3.1 Database Tables

Here are some DDL statements that used to setup my database tables and stored procedures.



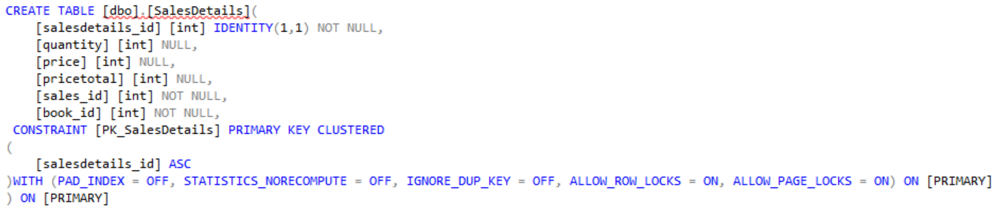
```
CREATE TABLE [dbo].[Book](
    [book_id] [int] IDENTITY(1,1) NOT NULL,
    [book_name] [varchar](max) NULL,
    [quantity] [int] NULL,
    [price] [int] NULL,
    [subject] [varchar](max) NULL,
    [form] [int] NULL,
    [book_image] [varbinary](max) NULL,
    [supplier_id] [int] NULL,
    [sprice] [int] NULL,
    [status] [varchar](max) NULL,
 CONSTRAINT [PK_Book] PRIMARY KEY CLUSTERED
(
    [book_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
--
```

Figure 5.4 : DDL Statement for table Book



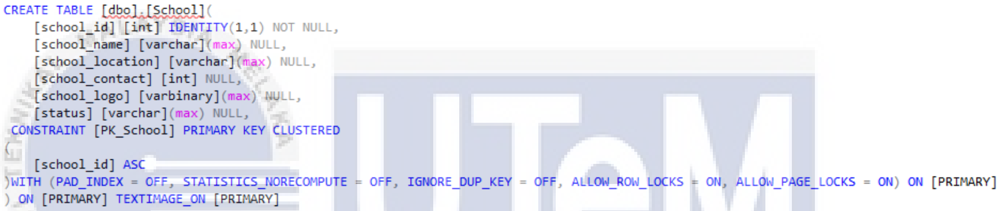
```
CREATE TABLE [dbo].[Sales](
    [sales_id] [int] IDENTITY(1,1) NOT NULL,
    [grandtotal] [int] NULL,
    [salesdate] [date] NULL,
    [school_id] [int] NOT NULL,
    [staff_id] [int] NOT NULL,
 CONSTRAINT [PK_Sales] PRIMARY KEY CLUSTERED
(
    [sales_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
--
```

Figure 5.5 : DDL Statement for table Sales



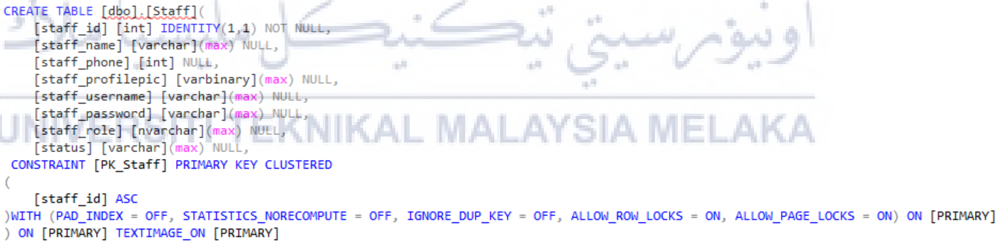
```
CREATE TABLE [dbo].[SalesDetails](
    [salesdetails_id] [int] IDENTITY(1,1) NOT NULL,
    [quantity] [int] NULL,
    [price] [int] NULL,
    [pricetotal] [int] NULL,
    [sales_id] [int] NOT NULL,
    [book_id] [int] NOT NULL,
 CONSTRAINT [PK_SalesDetails] PRIMARY KEY CLUSTERED
(
    [salesdetails_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Figure 5.6 : DDL Statement for table Sales Details



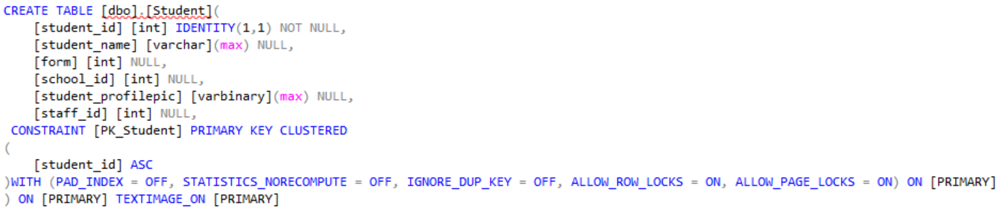
```
CREATE TABLE [dbo].[School](
    [school_id] [int] IDENTITY(1,1) NOT NULL,
    [school_name] [varchar](max) NULL,
    [school_location] [varchar](max) NULL,
    [school_contact] [int] NULL,
    [school_logo] [varbinary](max) NULL,
    [status] [varchar](max) NULL,
 CONSTRAINT [PK_School] PRIMARY KEY CLUSTERED
(
    [school_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Figure 5.7 : DDL Statement for table School



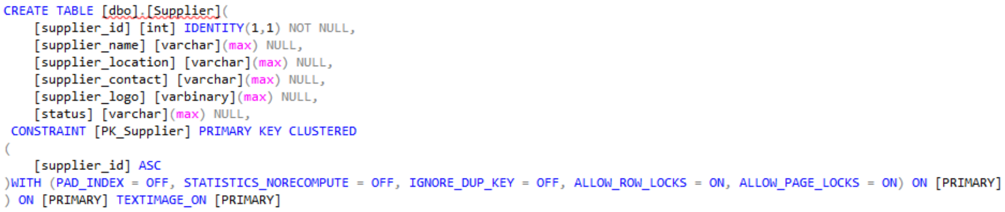
```
CREATE TABLE [dbo].[Staff](
    [staff_id] [int] IDENTITY(1,1) NOT NULL,
    [staff_name] [varchar](max) NULL,
    [staff_phone] [int] NULL,
    [staff_profilepic] [varbinary](max) NULL,
    [staff_username] [varchar](max) NULL,
    [staff_password] [varchar](max) NULL,
    [staff_role] [nvarchar](max) NULL,
    [status] [varchar](max) NULL,
 CONSTRAINT [PK_Staff] PRIMARY KEY CLUSTERED
(
    [staff_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Figure 5.8 : DDL Statement for table Staff



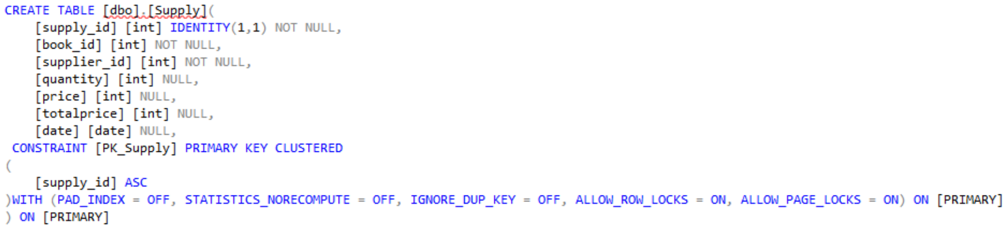
```
CREATE TABLE [dbo].[Student](
    [student_id] [int] IDENTITY(1,1) NOT NULL,
    [student_name] [varchar](max) NULL,
    [form] [int] NULL,
    [school_id] [int] NULL,
    [student_profilepic] [varbinary](max) NULL,
    [staff_id] [int] NULL,
 CONSTRAINT [PK_Student] PRIMARY KEY CLUSTERED
(
    [student_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Figure 5.9 : DDL Statement for table Student



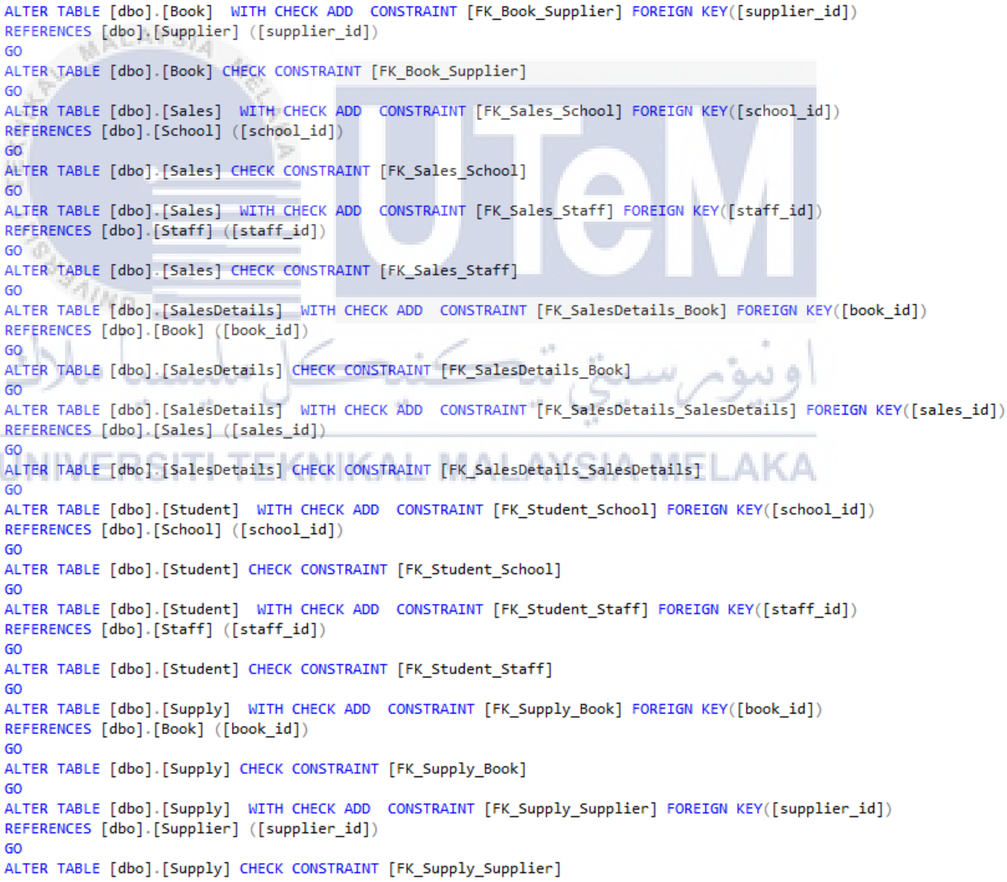
```
CREATE TABLE [dbo].[Supplier](
	[supplier_id] [int] IDENTITY(1,1) NOT NULL,
	[supplier_name] [varchar](max) NULL,
	[supplier_location] [varchar](max) NULL,
	[supplier_contact] [varchar](max) NULL,
	[supplier_logo] [varbinary](max) NULL,
	[status] [varchar](max) NULL,
 CONSTRAINT [PK_Supplier] PRIMARY KEY CLUSTERED
(
	[supplier_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

Figure 5.10 : DDL Statement for table Supplier



```
CREATE TABLE [dbo].[Supply](
	[supply_id] [int] IDENTITY(1,1) NOT NULL,
	[book_id] [int] NOT NULL,
	[supplier_id] [int] NOT NULL,
	[quantity] [int] NULL,
	[price] [int] NULL,
	[totalprice] [int] NULL,
	[date] [date] NULL,
 CONSTRAINT [PK_Supply] PRIMARY KEY CLUSTERED
(
	[supply_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Figure 5.11 : DDL Statement for table Supply

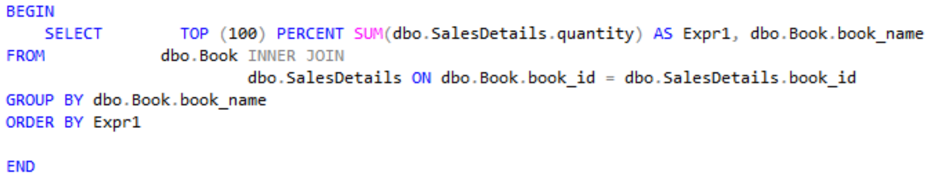


```
ALTER TABLE [dbo].[Book]  WITH CHECK ADD  CONSTRAINT [FK_Book_Supplier] FOREIGN KEY([supplier_id])
REFERENCES [dbo].[Supplier] ([supplier_id])
GO
ALTER TABLE [dbo].[Book] CHECK CONSTRAINT [FK_Book_Supplier]
GO
ALTER TABLE [dbo].[Sales]  WITH CHECK ADD  CONSTRAINT [FK_Sales_School] FOREIGN KEY([school_id])
REFERENCES [dbo].[School] ([school_id])
GO
ALTER TABLE [dbo].[Sales] CHECK CONSTRAINT [FK_Sales_School]
GO
ALTER TABLE [dbo].[Sales]  WITH CHECK ADD  CONSTRAINT [FK_Sales_Staff] FOREIGN KEY([staff_id])
REFERENCES [dbo].[Staff] ([staff_id])
GO
ALTER TABLE [dbo].[Sales] CHECK CONSTRAINT [FK_Sales_Staff]
GO
ALTER TABLE [dbo].[SalesDetails]  WITH CHECK ADD  CONSTRAINT [FK_SalesDetails_Book] FOREIGN KEY([book_id])
REFERENCES [dbo].[Book] ([book_id])
GO
ALTER TABLE [dbo].[SalesDetails] CHECK CONSTRAINT [FK_SalesDetails_Book]
GO
ALTER TABLE [dbo].[SalesDetails]  WITH CHECK ADD  CONSTRAINT [FK_SalesDetails_SalesDetails] FOREIGN KEY([sales_id])
REFERENCES [dbo].[Sales] ([sales_id])
GO
ALTER TABLE [dbo].[SalesDetails] CHECK CONSTRAINT [FK_SalesDetails_SalesDetails]
GO
ALTER TABLE [dbo].[Student]  WITH CHECK ADD  CONSTRAINT [FK_Student_School] FOREIGN KEY([school_id])
REFERENCES [dbo].[School] ([school_id])
GO
ALTER TABLE [dbo].[Student] CHECK CONSTRAINT [FK_Student_School]
GO
ALTER TABLE [dbo].[Student]  WITH CHECK ADD  CONSTRAINT [FK_Student_Staff] FOREIGN KEY([staff_id])
REFERENCES [dbo].[Staff] ([staff_id])
GO
ALTER TABLE [dbo].[Student] CHECK CONSTRAINT [FK_Student_Staff]
GO
ALTER TABLE [dbo].[Supply]  WITH CHECK ADD  CONSTRAINT [FK_Supply_Book] FOREIGN KEY([book_id])
REFERENCES [dbo].[Book] ([book_id])
GO
ALTER TABLE [dbo].[Supply] CHECK CONSTRAINT [FK_Supply_Book]
GO
ALTER TABLE [dbo].[Supply]  WITH CHECK ADD  CONSTRAINT [FK_Supply_Supplier] FOREIGN KEY([supplier_id])
REFERENCES [dbo].[Supplier] ([supplier_id])
GO
ALTER TABLE [dbo].[Supply] CHECK CONSTRAINT [FK_Supply_Supplier]
```

Figure 5.12 : Foreign keys for tables

5.3.2 Database Stored Procedure

This stored procedure display the amount of each books sold along with their names.

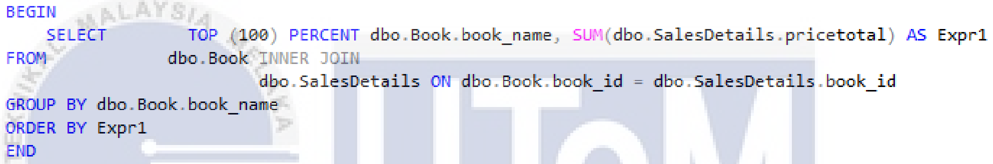


```
BEGIN
    SELECT        TOP (100) PERCENT SUM(dbo.SalesDetails.quantity) AS Expr1, dbo.Book.book_name
FROM            dbo.Book INNER JOIN
                         dbo.SalesDetails ON dbo.Book.book_id = dbo.SalesDetails.book_id
GROUP BY dbo.Book.book_name
ORDER BY Expr1

END
```

Figure 5.13 : Stored Procedure Book Sales 1

This stored procedure display the earnings of each books sold along with their names.



```
BEGIN
    SELECT        TOP (100) PERCENT dbo.Book.book_name, SUM(dbo.SalesDetails.pricetotal) AS Expr1
FROM            dbo.Book INNER JOIN
                         dbo.SalesDetails ON dbo.Book.book_id = dbo.SalesDetails.book_id
GROUP BY dbo.Book.book_name
ORDER BY Expr1
END
```

Figure 5.14 : Stored Procedure Book Sales 2

5.4 Conclusion

This chapter is all about the implementation of the system, including installation of software development software and database, setting up and applying the coding.

CHAPTER 6: TESTING

6.1 Introduction

In this phase, we test our outcome of implementation to make sure the product met the requirement of customer or user. Other than that, in this testing process, error or bug of the system can be detected and fix later. Solving error in this process is most important task, this is because error should not be existed in the completed project. It is straightly affecting the user experiences.

6.2 Test Plan

Our test plan for Bookstore Management System included design test and functional test. In this chapter, every test phase will be listed.

6.2.1 Test Organization

In software testing, test organization is an individual, company or an organization that are given the responsibilities to test the final product according to requirements and needs. For this project, software developer will be responsible to test all the functionality, errors and requirements.

Table 6.1 : Test Organization

Organization	Task and activities
Software Developer	• Test planning • Test cases • Run the test activities • Execute the system based on the test plan • Invest outcome of test result • Fix bug and handle errors occur • Make sure customer or user requirement met in the system.

6.2.2 Test Environment

Test Environment could be defined as the requirement of testing procedure. In this subchapter, the tools and resources use for testing procedure of mini mart management system will be declared. The requirement could be divided into two categories, they are software requirement and hardware requirement.

6.2.2.1 Software Requirement

Table 6.2 : Software Requirements

Software	Function
Visual Studio 2019	To implementation the code and model created by developer (Compiler)
Microsoft SQL Server Management Studio 2018	To store and retrieve data of mini mart system (Database)
Google Chrome	Web browser use to run ASP.net (MVC) (UI)

6.2.2.2 Hardware Requirement

Table 6.3 : Hardware Requirements

Minimum Requirement	Recommended Requirement
Windows 7	Windows 10
2 GHz Intel Dual Core Processor	2.3 GHz Intel Quad Core Processor
2 GB RAM	4 GB RAM
500GB Storage HHD	128GB Storage SSD
5 GB storage available	5 GB storage available

6.2.3 Test Schedule

Test schedule is the timeline of the tests. This included the name of test. duration of tests and date of tests started.

Table 6.4 : Test Schedule

No	Task name	Date Start	Duration
1	Test Login Module	9 Aug 2021	1 day
2	Test User Module	10 Aug 2021	6 days
3	Test Item Module	16 Aug 2021	2 days
4	Test Sales Module	18 Aug 2021	6 days
5	Test Graph Module	24 Aug 2021	4 days
6	User Requirement Checking	28 Aug 2021	2 days
7	Error and bug handling	30 Aug 2021	1 day

6.3 Test Strategy

There are many tests strategy can be choose as the way or approach to the test case for a system such as black box testing, white box testing, bottom-up and top down. It shows how the testing activities will be run and execute. For this project, only black box testing will be carried out.

Black box testing as known as functional test of the software. The implementation and structure are not known to the tester. This means every tester who run black box testing should not know the procedure or every single function of the mini mart management system. Black box tester will run the system as an end-user without guiding of the developer. It stresses on the input and investigate on the output the system produce. In this testing method, the developer can understand how well the system handle unwanted bug and errors that occurs during the test. This is because the black box testers are the people do not have any knowledge about the system or even programming. Black box testing usually carry out by software testers.

6.3.1 Classes of Tests

The classes of test mentioned below is used to test Bookstore Management System. It is including functionality test, security test and stress test. Functional test is the main purpose of the system. This test is to validate the system and its functional requirements and specifications. Other than this, this test is to ensure the needs of the user has been fulfilled. For security test, we will try to login with the wrong password or replace the password to the other user password and try to sneak into the system without authorize. For system stress test, we insert data as much as possible to let the system handle the data and ensure the interface and back-end are well handled.

6.4 Test Design

In our test phase, all the test procedure will be record down in details. Input and output will be determined and write down based on the test cases. Other than this, all the result of test will also be divided and note with pass or fail. Test design is helping the developer to determine or to select the good group of tests that carry from the system.

6.4.1 Test Description

In this project, test description is being documented based on the modules developed for the system. The following are some of the test description of the system testing which are the testing for Login Module, User Module, Sales Module, Item Module and Report Module.

Table 6.5 : Login Module Test

Test ID	LOGIN0001		
Module Name	Login Module		
Description	To allow user to access the system with authorize		
Test Designed By	System developer	**Test Executed By**	System developer
Pre-condition	User must create account with the help of administrator (admin)	**Post-condition**	User is successfully log into the system. The system will keep track the user details when session is not expired.
Test Case ID	**Test Case**	**Step to Execute**	**Expected Output**
LOGIN0001_01	Username and password not match	1. Input username and password that is not match in database 2. Click Login	Alert message: User is not found.
LOGIN0001_02	Input matched username and password	1. Input a matched username and password 2. Click Login	User with correct combination of username and password successfully log into system. Session has been started.

Table 6.6 : User Module Test

Test ID	USER0002		
Module Name	User Module		
Description	To allow admin(user) to add new account and set role for each specific user such as Admin, Staff or User.		
Test Designed By	System developer	**Test Executed By**	System developer
Pre-condition	User with admin role need to log into system	**Post-condition**	New account will be created with role based on the input of admin.
Test Case ID	**Test Case**	**Step to Execute**	**Expected Output**
USER 0002_01	Input empty detail	**1.** Go to the create user page **2.** Click the submit button without input any data	Alert message: Please fill in the columns
USER 0002_02	Input data without image/profile picture	**1.** Go to the create user page **2.** Input the details of user **3.** Click submit	Alert message: Please fill in the columns
USER 0002_03	Input data with image/ profile picture	**1.** Go to the create staff form **2.** Input the details of user **3.** Select the image for user profile picture **4.** Click submit	User was created and able to login as password input at the input data.

Table 6.7 : Item Module Test

Test ID	ITEM0003		
Module Name	Item Module		
Description	To allow user to add new item/product into system		
Test Designed By	System developer	**Test Executed By**	System developer
Pre-condition	Only Admin or Staff can create	**Post-condition**	Item will be created and able to sell at the price as input
Test Case ID	**Test Case**	**Step to Execute**	**Expected Output**
ITEM0003_01	Input empty detail for item	**1.** Go to create item page **2.** Click save without input any data	Alert message: Please fill in the columns
ITEM0003_02	Input data without selecting image for item	**1.** Go to create item page **2.** Input valid data for item **3.** Click save without selecting image	Alert message: Please fill in the columns
ITEM0003_03	Input negative value for quantity and price	**1.** Go to create item page **2.** Input valid data but negative value for quantity and price **3.** Click save	Alert message: Please fill in the columns
ITEM0003_04	Input valid data and select photo	**1.** Go to create item page **2.** Select image for item **3.** Input data for item **4.** Click save	Item was successfully created into the database.

Table 6.7 : Sales Module Test

Test ID	SALES0004		
Module Name	Sales Module		
Description	To allow user makes purchases		
Test Designed By	System developer	**Test Executed By**	System developer
Pre-condition	Any user can make purchase under own name	**Post-condition**	Price of item purchase must lower or equal than the quantity inside database.
Test Case ID	**Test Case**	**Step to Execute**	**Expected Output**
SALES0004_01	Select quantity that is more than the stock	**1.** Select any product for purchase **2.** Input quantity more than in stock **3.** Click add button	Alert message: Quantity for this item is not enough in the stock
SALES0004_02	Input valid quantity	**1.** Select product from the UI **2.** Click ‘Select item’ or more. **3.** Input a valid quantity. **4.** Click ‘Buy’ to complete purchase	Successfully purchase and purchase history had been recorded.

Table 6.8 : Report Module Test

Test ID	REPORT0005		
Module Name	Report Module		
Description	To allow admin to view the statistics/ graph of the bookstore.		
Test Designed By	System developer	**Test Executed By**	System developer
Pre-condition	Users need to log in into system as admin.	**Post-condition**	Admin will be allowed to view all the data of bookstore
Test Case ID	**Test Case**	**Step to Execute**	**Expected Output**
REPORT0005_01	Report and statistics viewing	**1.** Select the Stat at the navigation bar	Admin or user was able to view the graph and detail of bookstore (For example: total staff in mini mart, total sales, and number of items in store)

6.4.2 Test Data

Test data is list of data that is used as an input to execute the test cases of a software testing to uncover the faults or defects during testing, test data must be precise and comprehensive. Some of the data may be in form of positive testing in order to verify that the actual result meets the expected result. Other than that, negative testing may also present in the test data to identify the ability of the system to manage any unexpected input.

Table 6.9 : Test Data Login Module

Test ID	LOGIN0001
Module Name	Login Module
Test Case ID	LOGIN0001_01
Test Case	Username and password not match
Input Field	**Test Data**
Username	ali
Password	abu
Test Case ID	LOGIN001_02
Test Case	Input matched username and password
Input Field	**Test Data**
Username	ali
Password	ali

Table 6.10 : Test Data User Module

Test ID	USER0002
Module Name	User Module
Test Case ID	USER0002_01
Test Case	Input empty detail
Input Field	**Test Data**
Name	NULL
Phone	NULL
Profile Picture	NULL
Username	NULL
Password	NULL
Role	NULL
Test Case ID	USER0001_02
Test Case	Input data without image/profile picture
Input Field	**Test Data**
Name	Ali
Phone	012347698
Profile Picture	NULL
Username	ali
Password	ali
Role	Admin
Test Case ID	USER0001_03
Test Case	Input data with image/ profile picture
Input Field	**Test Data**
Name	Ali
Phone	012347698
Profile Picture	TRUE
Username	ali
Password	ali
Role	Admin

Table 6.11 : Test Data Item Module

Test ID	ITEM0003
Module Name	Item Module
Test Case ID	ITEM0003_01
Test Case	Input empty detail for item
Input Field	**Test Data**
Name	NULL
Quantity	NULL
Price	NULL
Sales Price	NULL
Supply Price	NULL
Subject	NULL
Form	NULL
Book Image	NULL
Supplier	NULL
Test Case ID	ITEM0003_02
Test Case	Input data without selecting image for item
Input Field	**Test Data**
Name	Bahasa Melayu Tingkatan 1
Quantity	12
Sales Price	15
Supply Price	10
Subject	Bahasa Melayu
Form	1
Book Image	NULL
Supplier	Sasbadi
Test Case ID	ITEM0003_03
Test Case	Input negative value for quantity and price
Input Field	**Test Data**
Name	Bahasa Melayu Tingkatan 1
Quantity	-12
Sales Price	15
Supply Price	-20

Subject	Bahasa Melayu
Form	1
Book Image	NULL
Supplier	Sasbadi
Test Case ID	ITEM0003_04
Test Case	Input valid data and select photo
Input Field	**Test Data**
Name	Bahasa Melayu Tingkatan 1
Quantity	-2
Sales Price	15
Supply Price	20
Subject	Bahasa Melayu
Form	1
Book Image	TRUE
Supplier	Sasbadi

Table 6.12 : Test Data Sales Module

Test ID	SALES0004
Module Name	Sales Module
Test Case ID	SALES0004_01
Test Case	Select quantity that is more than the stock
Input Field	**Test Data**
Quantity	2134235342
Test Case ID	SALES0004_02
Test Case	Input valid quantity
Input Field	**Test Data**
Quantity	5

6.5 Test Result and Analysis

The result of test cases may be different from the expected outcome of the developer. To ensure our test case carry out with useful result. The developer should observe and note down the outcome of the test result.

Table 6.13 : Test Result Login Module

Test ID	LOGIN0001	
Module Name	Login Module	
Test Case ID	Result of Test	Test Result
LOGIN0001_01	Alert message: User is not found.	PASS
LOGIN0001_02	User with correct combination of username and password successfully log into system. Session has been started.	PASS

Table 6.14 : Test Result User Module

Test ID	USER0002	
Module Name	User Module	
Test Case ID	Result of Test	Test Result
USER 0002_01	Alert message: Please fill in the columns	PASS
USER 0002_02	Alert message: Please fill in the columns	PASS
USER 0002_03	User was created and able to login as password input at the input data.	PASS

Table 6.15 : Test Result Item Module

Test ID	ITEM0003	
Module Name	Item Module	
Test Case ID	Result of Test	Test Result
ITEM0003_01	Alert message: Please fill in the columns	PASS
ITEM0003_02	Alert message: Please fill in the columns	PASS
ITEM0003_03	Alert message: Please fill in the columns	PASS
ITEM0003_04	Item was successfully created into the database.	PASS

Table 6.16 : Test Result Sales Module

Test ID	SALES0004	
Module Name	Sales Module	
Test Case ID	Result of Test	Test Result
SALES0004_01	Alert message: Quantity for this item is not enough in the stock	PASS
SALES0004_02	Successfully purchase and purchase history had been recorded.	PASS

Table 6.17 : Test Result Report Module

Test ID	REPORT0005	
Module Name	Report Module	
Test Case ID	Result of Test	Test Result
REPORT0005_01	Admin or user was able to view the graph and detail of bookstore (For example: total staff in mini mart, total sales, and number of items in store)	PASS

6.6 Conclusion

In this chapter, testing phase inside SDLC has been execute. All the design test, strategic test, test schedule and test activities were carries out by the developer only. Testing phase is very important to developer to stable the system they develop and make sure the requirement of end user has been fulfilled. Developer was ensured this system was able to assist or help the daily work of mini mart.

Test phase is a process that should not be skipped from the SDLC. This is because test phase is helping the developer to debug, fixing programming errors and avoid logical errors.

Besides that, there is still weakness, and it will be discussed in our next chapter of the report and the last chapter of the report for this project.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

The main purpose of this system, that is to simplify and automate the process of storing and managing a bookstore. During the system development, there are many challenges faced due to technical problems or insufficient of knowledge and skills. When the problems arise, it takes lot of time to solve it. Although some of the development phase are not done in time, but the valuable experience gained during dealing the problems are very useful to help me in the future stage. At the end, Bookstore Management System has been successfully developed in approximately six months.

7.2 Observations on Weaknesses and Strengths

7.2.1 Weaknesses

1. Unable to pay online

7.2.2 Strengths

1. Able to manage all data efficiently

2. Able to view graphs and statistics for sales

3. User friendly

7.3 Propositions for Improvement

The following points are the things that can be improve for this system in the future. The improvements are based on problems or weakness in the system mentioned above.

1. Allow user to pay online through the system.

2. Develop the system and make it suitable to be used in mobile browser. So that teacher can still manage the data even computer is not available.

7.4 Project Contribution

Bookstore Management System is a system that able to manage user and stocks in bookstore for easier management. The main purpose of this project is to replace the traditional ways that using paper work. After the creation of this project, I believed that it is able to increase the efficiency of their daily work.

7.5 Conclusion

In conclusion, Bookstore Management System is very useful and efficient for the staffs to manage their store's data. Moreover, the system that I developed had achieved the requirements for them which can help them save a lot of work compare to previous existing system.

REFERENCES

Jawatan Kuasa PSM & PD (2019). Bachelor Project Writing Guide, Universiti Teknikal Malaysia Melaka.

How to Run a Bookstore: 9 Strategies for a Successful Bookshop. accessed 1 March 2019, < https://koronapos.com/blog/how-to-run-a-bookstore/>.

Running A Bookstore: 5 Important Tips For Inventory Management. accessed 2017,<https://www.visualretailplus.com/inventory-management-system/running-a-bookstore-5-important-tips-for-inventory-management/>.