

E-MANAGEMENT  
FOR SCHOOL RESOURCE CENTER

MUHAMMAD ASHRAF BIN KAMARUDIN

Bachelors Of Computer Science  
(Software Engineering)

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : MUHAMMAD ASHRAF BIN KAMARUDIN  
Date of Birth : SEPTEMBER, 5, 1995  
Title : E – MANAGEMENT FOR SCHOOL RESOURCE CENTER  
Academic Session : SEMESTER II 2018/2019

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

\_\_\_\_\_  
(Supervisor's Signature)

\_\_\_\_\_  
New IC/Passport Number  
Date:

\_\_\_\_\_  
Name of Supervisor  
Date:

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



## **SUPERVISOR'S DECLARATION**

I hereby declare that I have checked this thesis and in my opinion, this thesis/ is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering)

---

(Supervisor's Signature)

Full Name : AZMA BINTI ABDULLAH

Position :

Date : 30 MAY 2019



## **STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

---

(Student's Signature)

Full Name : MUHAMMAD ASHRAF BIN KAMARUDIN

ID Number : CB 16148

Date : 30 MAY 2019

E-MANAGEMENT  
FOR SCHOOL RESOURCE CENTER

MUHAMMAD ASHRAF BIN KAMARUDIN

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Computer Science(Software Engineering)

Faculty of Computer Systems and Software Engineering  
UNIVERSITI MALAYSIA PAHANG

DECEMBER 2018

## **ACKNOWLEDGEMENTS**

First of all, I would like to express my thanks and gratitude to my Supervisor, Madam Azma Binti Abdullah, who guides me throughout this project, she always have time in helping this project and always support me in all aspects while finishing this project. She always give me guidance on finishing this project report at any time.

Secondly, I would like to thanks to my family that gave me continuous support from behind, to all my friends for providing me unfailing supports and encouragement for me throughout my study years and throughout the process of developing this project as well as writing this thesis.

Lastly I would like to express my gratitude to University Malaysia Pahang that gave me opportunity to infinitely gain my knowledge there that cannot find at any other place.

## ABSTRAK

Tujuan kertas ini adalah untuk menukar atau menaik taraf sistem manual semasa ke dalam sistem berkomputer untuk mencapai pemprosesan maklumat yang lebih pantas dan cekap, penjanaan automatik pelbagai jenis laporan, pengurangan kos perkakasan dan perisian yang lebih besar dan mampu menghasilkan maklumat yang lebih tepat. Buku-buku di pusat sumber sekolah sering hilang kerana prosedur pengurusan yang betul tidak dilaksanakan. Guru sekolah juga menghadapi masalah untuk memeriksa buku apa yang terdapat di pusat sumber. Pelajar yang meminjam buku sering kali tidak memulangkan buku yang dipinjam tetapi tetap tidak disedari oleh pengurusan pusat sumber sekolah kerana proses mencari atau memeriksa buku yang ada dan yang hilang adalah tugas yang menyusahkan tanpa sistem pengurusan yang betul dan akan mengambil masa yang lama . Pelajar juga akan meninggalkan sekolah tanpa dihukum kerana rekodnya meminjam buku itu dengan mudah boleh hilang atau dimusnahkan tanpa jejak. Untuk menyelesaikan masalah yang dihadapi, kami mula-mula mengumpul keperluan sistem dan data berkaitan lain dari pusat sumber sekolah dan menganalisisnya dengan mewujudkan keperluan sistem permulaan. Kemudian kita mencari, menganalisis dan membandingkan sistem yang ada yang menangani masalah dengan cara mereka sendiri. Kami mencari kekuatan, kelemahan dan keperluan mereka yang kurang dan mula memreka bentuk sistem baru yang akan memenuhi semua keperluan. Kami menggunakan metodologi Pengembangan Aplikasi Rapid dari awal hingga akhir untuk memastikan projek itu dapat disampaikan tepat pada waktunya dalam anggaran projek dan berkualiti tinggi. Sistem akhir yang dihasilkan akan dapat menjejaki keberadaan mana-mana buku dalam beberapa saat dan dapat menghasilkan laporan yang tepat berdasarkan maklumat yang dicatat. Untuk membuat kesimpulan, sistem berkomputer mampu untuk menyelesaikan semua masalah yang sistem manual sekarang hadapi ditambahpula dengan penambahan beberapa penambahbaikan untuk memudahkan tugas mengurus pusat sumber sekolah.

## ABSTRACT

The purpose of this paper is to change or upgrade the current manual system into a computerized system to achieve faster and efficient in processing information, automatic generation of many types of reports , larger reductions in cost of hardware and software and have a more timely information produced. Books in school resource center are often lost as no proper managing procedure is implemented. School Teachers also have troubles finding out what book is in the resource center's possession. Student's who borrowed the books often not returned the borrowed books but remained unnoticed by the management of school resource center because checking and trying to find out what books is there and is missing is a tedious task without a proper management system and take a long time. The student also leave the school unpunished as the record of him borrowing the book can easily be lost or destroyed with no trace. To solve the problems at hand, system requirements and other related data from the school resource center is then gathered and analyzed to create a first defined system requirements. Then existing system that tackles the problems with their own ways is find, analyze and compare. Their strength, weakness and the requirements they are lacking is find and then begin to construct the new system design that fulfilled all requirements. The project use Rapid Application Development methodology from the beginning until the end to make sure the project can be delivered on time within the project's budget and is of high quality. The final system produced then be able to tracks the whereabouts of any books within seconds and able to produce accurate reports based on the information recorded. To conclude, the computerized system manage to solve all problems that the current manual system is facing with an addition of several improvement to ease the tedious task of managing the school resource center.



## TABLE OF CONTENT

|                                    |             |
|------------------------------------|-------------|
| <b>DECLARATION</b>                 |             |
| <b>TITLE PAGE</b>                  |             |
| <b>ACKNOWLEDGEMENTS</b>            | <b>ii</b>   |
| <b>ABSTRAK</b>                     | <b>iii</b>  |
| <b>ABSTRACT</b>                    | <b>iv</b>   |
| <b>TABLE OF CONTENT</b>            | <b>v</b>    |
| <b>LIST OF TABLES</b>              | <b>viii</b> |
| <b>LIST OF FIGURES</b>             | <b>ix</b>   |
| <b>LIST OF SYMBOLS</b>             | <b>x</b>    |
| <b>LIST OF ABBREVIATIONS</b>       | <b>xi</b>   |
| <b>CHAPTER 1 INTRODUCTION</b>      | <b>1</b>    |
| 1.1 INTRODUCTION                   | 1           |
| 1.2 PROBLEM STATEMENT              | 2           |
| 1.3 OBJECTIVE                      | 3           |
| 1.4 SCOPE                          | 3           |
| 1.5 REPORT ORGANIZATION            | 4           |
| <b>CHAPTER 2 LITERATURE REVIEW</b> | <b>5</b>    |
| 2.1 INTRODUCTION                   | 5           |
| 2.2 TECHNOLOGIES                   | 6           |
| 2.2.1 Web Browser                  | 6           |
| 2.2.2 APACHE HTTP Server           | 7           |
| 2.2.3 MySQL                        | 8           |

|   |   |           |
|---|---|-----------|
| 2.3   | COMPARISON EXISTING SYSTEM                | 9         |
| 2.3.1   | Tarantula PSS                             | 9         |
| 2.3.2   | E-Proses PSS                              | 10        |
| 2.3.3   | SENAYAN Library Automation System (SLiMS) | 11        |
| 2.4   | COMPARISON BETWEEN EXISTING SYSTEM        | 13        |
|   | System Comparison                         | 13        |
|   | Advantages Comparison                     | 14        |
|   | Disadvantages Comparison                  | 14        |
| 2.5   | CONCLUSION                                | 15        |
| <br><b>CHAPTER 3 METHODOLOGY</b>              |   | <b>16</b> |
| 3.1   | INTRODUCTION                              | 16        |
| 3.2   | METHODOLOGY                               | 17        |
| 3.2.1   | Requirement Planning Phase                | 17        |
| 3.2.2   | User Design                               | 20        |
| 3.2.3   | Construction                              | 22        |
| 3.2.4   | Cutover                                   | 23        |
| 3.3   | Hardware and Software                     | 24        |
| 3.3.1   | Hardware Requirement                      | 24        |
| 3.3.2   | Software Requirement                      | 25        |
| 3.4   | Gantt Chart                               | 25        |
| <br><b>CHAPTER 4</b>                          |   | <b>26</b> |
| IMPLEMENTATION, TESTING AND RESULT DISCUSSION |   | 26        |
| 4.1   | INTRODUCTION                              | 26        |
| 4.2   | IMPLEMENTATION                            | 26        |
| 4.2.1   | Development Environment                   | 26        |

|  |                               |           |
|--|-------------------------------|-----------|
| 4.2.2  | System Functionality          | 30        |
| 4.3  | Testing and Result discussion | 34        |
| 4.3.1  | Functional Testing            | 34        |
| <b>CHAPTER 5</b>   |                               | <b>36</b> |
| CONCLUSION AND FUTURE WORK                                 |                               | 36        |
| 5.1  | INTRODUCTION                  | 36        |
| 5.2  | RESEARCH CONSTRAINT           | 36        |
| 5.2.1  | Development Constraint        | 36        |
| 5.2.2  | System Constraint             | 37        |
| 5.3  | FUTURE WORK                   | 37        |
| <b>REFERENCES</b>  |                               | <b>38</b> |
| <b>APPENDIX A GANTT CHART</b>                              |                               | <b>39</b> |
| <b>APPENDIX B Software Requirement Specification (SRS)</b> |                               | <b>40</b> |
| <b>APPENDIX C Software Design Document (SDD)</b>           |                               | <b>41</b> |

## LIST OF TABLES

|  |    |
|--|----|
| Table 2.1 Existing System Comparison               | 13 |
| Table 2.2 Existing System Advantages Comparison    | 14 |
| Table 2.3 Existing System Disadvantages Comparison | 14 |
| Table 3.1 Hardware Requirement                     | 24 |
| Table 3.2 Software Requirement                     | 25 |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 2.1 Web Browsers                    | 6  |
| Figure 2.2 Apache Web Server               | 7  |
| Figure 2.3 MySQL                           | 8  |
| Figure 2.4 phpMyAdmin                      | 8  |
| Figure 2.5 Tarantula's Login Page          | 9  |
| Figure 2.6 E-Proses's Add Resource Page    | 10 |
| Figure 2.7 E-Proses Homepage               | 10 |
| Figure 2.8 SLiMS's Module Page             | 11 |
| Figure 2.9 SLiMS's Homepage                | 12 |
| Figure 3.1 Life Cycle Stages of RAD        | 17 |
| Figure 3.2 Context Diagram                 | 18 |
| Figure 3.3 Use Case Diagram                | 19 |
| Figure 3.4 Entity Relational Diagram       | 19 |
| Figure 4.1 Command for installing Composer | 27 |
| Figure 4.2 XAMPP Manager                   | 28 |
| Figure 4.3 Laravel Home                    | 29 |
| Figure 4.4 PHPMyAdmin                      | 29 |
| Figure 4.5 E-MSRC Homepage                 | 30 |
| Figure 4.6 E-MSRC Navbar                   | 30 |
| Figure 4.7 E-MSRC Borrow and Return page   | 31 |
| Figure 4.8 E-MSRC Student Check In         | 32 |
| Figure 4.9 Manage Book                     | 33 |
| Figure 4.10 Manage User                    | 34 |

## **LIST OF SYMBOLS**

Sdn Bhd

Sendirian Berhad

## **LIST OF ABBREVIATIONS**

|          |   |
|----------|---|
| SRC      | School Resource Centre                    |
| E - MSRC | E – Management for School Resource Centre |
| SaaS     | System as a Service                       |
| HTTP     | Hypertext Transfer Protocol               |
| RAD      | Rapid Application Development             |
| ERD      | Entity Relational Diagram                 |

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 INTRODUCTION**

The School Resource Centres (SRCs) in Malaysia provide information and ideas that are fundamental to function successfully in today's information and knowledge-based society. The SRCs equip students with information literacy skills and help them develop life-long learning habits, enabling them to be knowledgeable and responsible citizens. (ZAINUDIN, 2016). SRCs mission is to provide learning opportunities to everyone who visits and sometimes it also serves as a place for students to find research materials, do individual work or discussion on certain topics.

Currently, there are few existing system that helps SRCs to manage their resources efficiently. These systems are usually owned by a company that charge every school who uses the system monthly or yearly depending on the company. The school is given one admin account that can create or register new users. This is to give multiple teacher or staff in charge access to the system's functionality. Some of the functions in this system are user login for SRC admin, resource management and status. Unauthorized users cannot access any of this system functionality as they are all in a restricted area only for logged in users. Once the user is logged in, they can manage school's resources in here. They can add, edit and delete these resources from the system's databases. If any of the resources became unavailable, the teacher or staff in charge needs to update the resource's status as unavailable for future reference.



Despite having made managing the resources in SRC easier, these existing systems still have room for improvement. E – Management for School Resource Centre (EMRC) aims to improve the areas in which the existing system is lacking. EMRC should have a searching function available for all levels of the system user. Users just have to input the name of the item they wish to find out and the system provide details of the item including the item’s availability status. The system also provides a catalogue of all items in the system. Guest user, such as students only able to view the list and the item details while system’s admin have the ability to manage the catalogue. System’s admin may register students into the system either manually or with an excel sheet. If the student’s info is in the system, the student may borrow SRC’s items for personal use. All circulation transactions is recorded in the system. System admin is able to generate reports based on the recorded data.

## **1.2 PROBLEM STATEMENT**

Manual system used by school’s resource centre is fully paper-based. Every records is hand-written and recorded on a piece of paper. That piece of paper in which contain a valuable information about the resources is vulnerable to harm. It can get burned, wet, torn or lost.

Papers consume space. The school resource centre must prepare extra shelves to accommodate all of the papers. If by chance some of the data on the paper is needed, it require a huge amount of time to locate the exact paper that records the information they need. It requires more time if the school resource centre uses a poor filing system.

Paper based system also have weak security. If no one is currently watching or guarding the papers, intruders can easily gain access to the data. This would cause unwanted troubles and therefore not recommended.

### 1.3 OBJECTIVE

The objective of this system are as follow :

- i. To study the manual system and computerised the SRC Management system
- ii. To develop a prototype system of SRC Management system
- iii. To validate the proposed prototype system in web based.

### 1.4 SCOPE

The scope of this project has been discussed and agreed upon. These are the scope of the system that have been identified for the system development.

- i. User :
  - a. The users of this system is the students, and admin(teacher in charge) and Library's Prefect..
- ii. Client :
  - a. Sekolah Menengah Kebangsaan(SMK) Sri Nilam.
- iii. Function Available :
  - a. Staff Lead
    - i. Roles & Permission – Mange Roles and Permission for other users
    - ii. Login – Login to admin's dashboard
    - iii. Manage Catalogue – Add, Update and Delete item and item's details
    - iv. Student Registration – Able to register students to user list.
    - v. Report – Able to generate reports.
  - b. Staff
    - i. Login – Login to admin's dashboard
    - ii. Manage Catalogue – Add, Update and Delete item and item's details
    - iii. Student Registration – Able to register students to user list.
    - iv. Report – Able to generate reports.
    - v. Student Check- In – Check in for students
  - c. Library's Prefect

- i. Login – Login to dashboard
- ii. Manage Catalogue – Add, Update and Delete item and item's Details

## **1.5 REPORT ORGANIZATION**

This project is consist of five chapters that explain mostly everything that you needed to understand this project. Chapter 1 is where the project is first introduced which includes introduction, problem statement, objective, scope and report organization. This is where the problems are identified and scope and objective is specified that leads to the solution. The solution satisfy the speicified scope and objective.

Chapter 2 is literature review of existing systems and compare it to the system that is develop. This chapter includes introduction for this chapter, in-depth study of this project and techniques/method or technologies which is commonly applied by existing systems available on the market. The existing systems is analysed and compare with each other throughout this chapter.

Chapter 3 talks about methodology that is applied in the project development. T he methodology that is chosen is stated and explain throughout this chapter. The differ ent phases of development that is in the chosen methodology are explained. This chapt er also have introduction to this chapter, hardware, software and Gantt chart to be used in the project detail.

Chapter 4 talks about the implementation, testing and result discussion. This ch apter guides you on how the system had been implemented based on the finalize requir ements of the system. This chapter also talks about the test cases created for this syste m to ensure that the systems requirement is verified, validated and is free from bugs of any sort. The test results is then stated here.

Chapter 5 talks about the project conclusion. This chapter also includes the intr oduction for this chapter which concludes the completed projects, research constraints which explains the clarification on the constraints throughout the project and future work for this system

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION**

Literature review of several existing system that is similar to E-MSRC is included in this chapter. All information is gathered using facts and finding technique which is then analysed and applied in this literature review. Most of the technology applied by these existing systems is shown and explained. The comparison of these existing system that has been analysed is also included in this chapter. Then, several improvement for the system is proposed.

## 2.2 TECHNOLOGIES

Technologies used by these system is as follows :

### 2.2.1 Web Browser

A web browser, which is also commonly known as browser is a software application used to access information on the world wide web. Every web page, video and image is identified by a specific URL which enable browsers to retrieve them and display them on our screen. (Web Browser - Wikipedia, n.d.)



Figure 2.1 Web Browsers

Some example of a recent web browsers that is widely used is Google Chrome, Mozilla Firefox, Safari and Opera. Some web browsers may be better or faster than the others but they all served the same purpose for end-users. Now, these browsers also have variety of tools to help developers build their web systems. These tools includes console to log system's error, sources to check the file's source and network to check how long it takes for the page to load and what resource the page used. Developers can also changed the web-page code in the browser's element to see the changes right away before writing the code in the original file.

## 2.2.2 APACHE HTTP Server

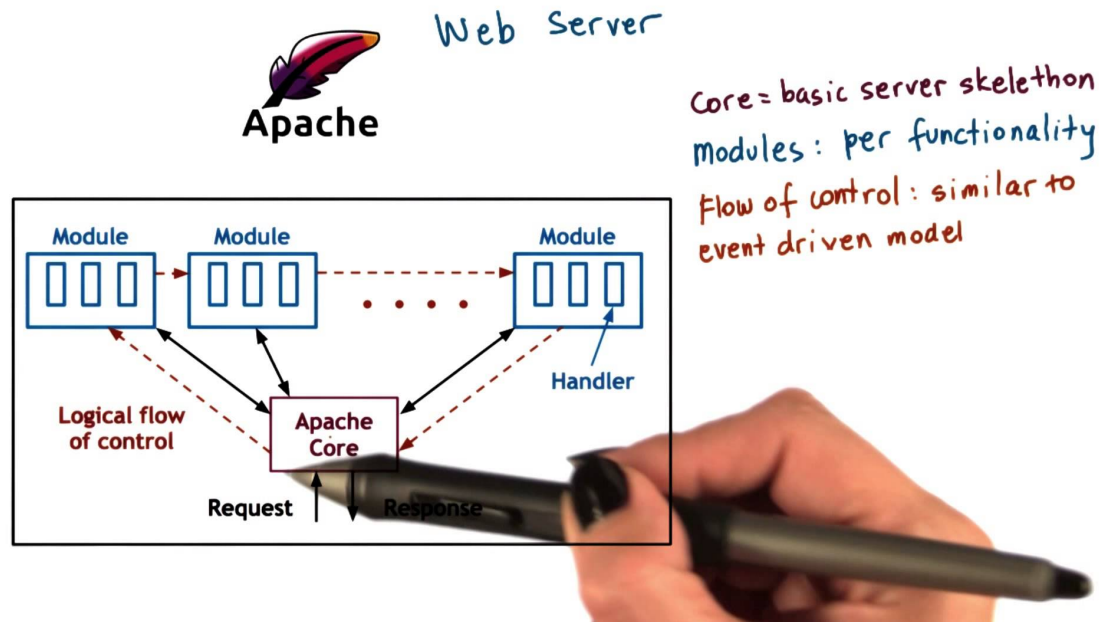


Figure 2.2 Apache Web Server

APACHE HTTP Server that is commonly known as APACHE is a free open source web server used widely in the world wide web. A web server, like its name suggest serves you. When you first requested for the web page, the web server checks for the web page you have requested and fetches it for your view. A web server also your host and it host your application file in the server. Basically, a web server is a software that receives your request for a web page. It runs a few security checks on your request and then serves you the page you requested . (What is Apache ? - What is web server ?, n.d.)

### 2.2.3 MySQL



Figure 2.3 MySQL

“MySQL is an open-source relational database management system (RDBMS). its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.” (MySQL - Wikipedia, n.d.)

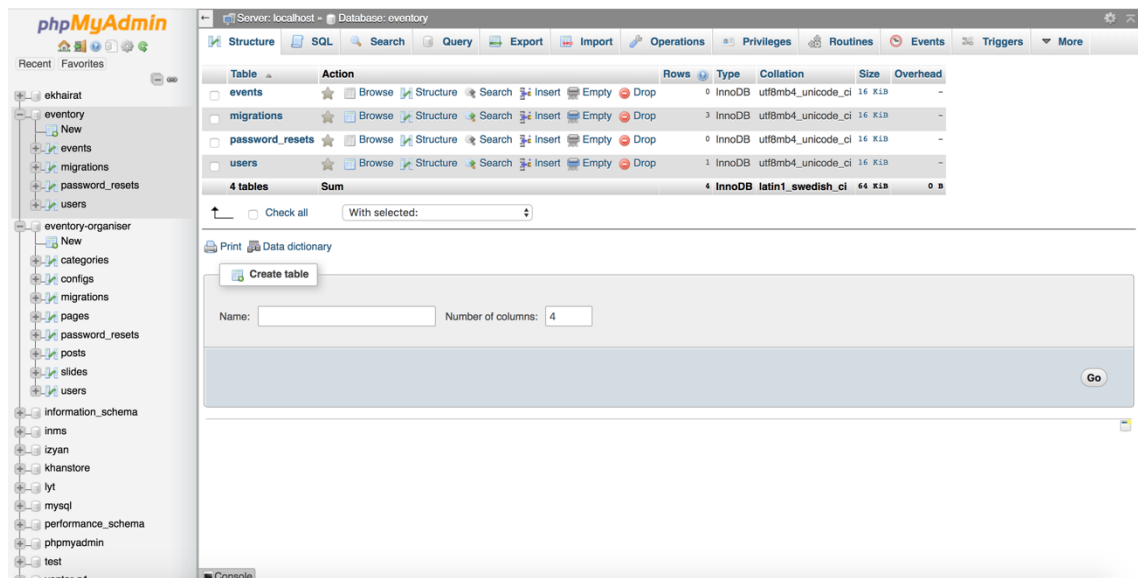


Figure 2.4 phpMyAdmin

Most database driven web-based system used MySQL as their database because MySQL offers easy way to store and manage data used by the system. MySQL uses SQL language, which is widely used by RDBMS. New and seasoned developers both learned and use SQL language to manage database operation. Most common database manager used by web developer is phpMyAdmin.

## 2.3 COMPARISON EXISTING SYSTEM

### 2.3.1 Tarantula PSS

Tarantula One Sdn Bhd develop a web-based system as system as a service(SaaS) for Malaysian's school's SRC.



Figure 2.5 Tarantula's Login Page

This system fulfilled most of the basic requirements of a management system. Teachers can login and manage SRC's resources in web browser without any prior installation. System user does not need advance computer skill to use this system. Tarantula One Sdn Bhd only charged once in a lifetime for new user who want access to this system.

The functionality of this system is, Teacher may add new entry to the system database. The entry may be book, equipment or anything that belong to the school's SRC. The teach may edit or remove the entry at any time after adding. Teacher may need to manually edit and update the book's condition and status in the system. No other use can use this system functionality but the teacher may add or allow new user to use it.



### 2.3.2 E-Proses PSS

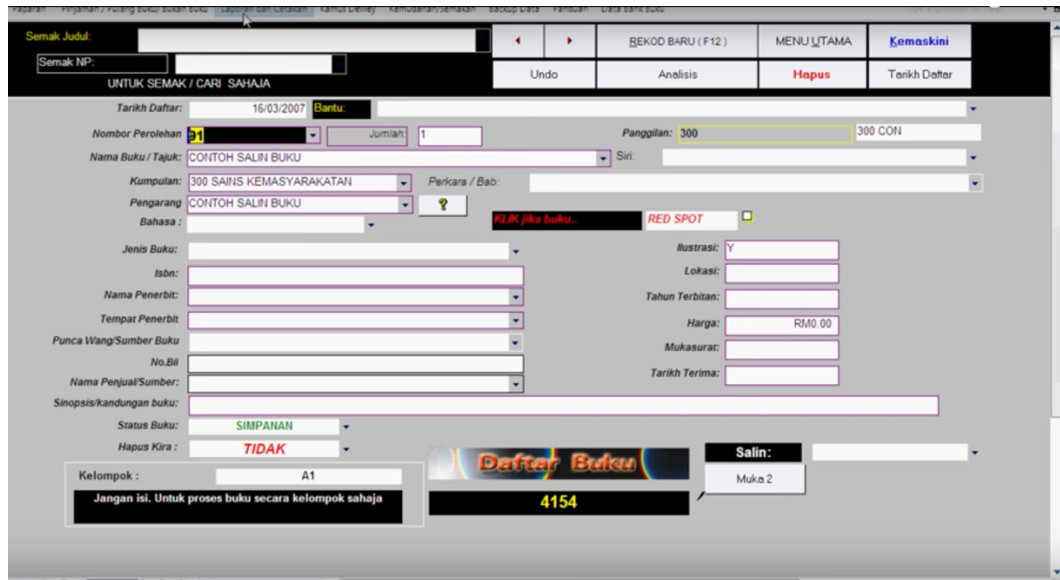


Figure 2.6 E-Proses's Add Resource Page

E-Proses also fulfilled most of the basic requirements of a management system and much more but this particular system lacks a login module. When you run this system, it bring you directly to the system's homepage.

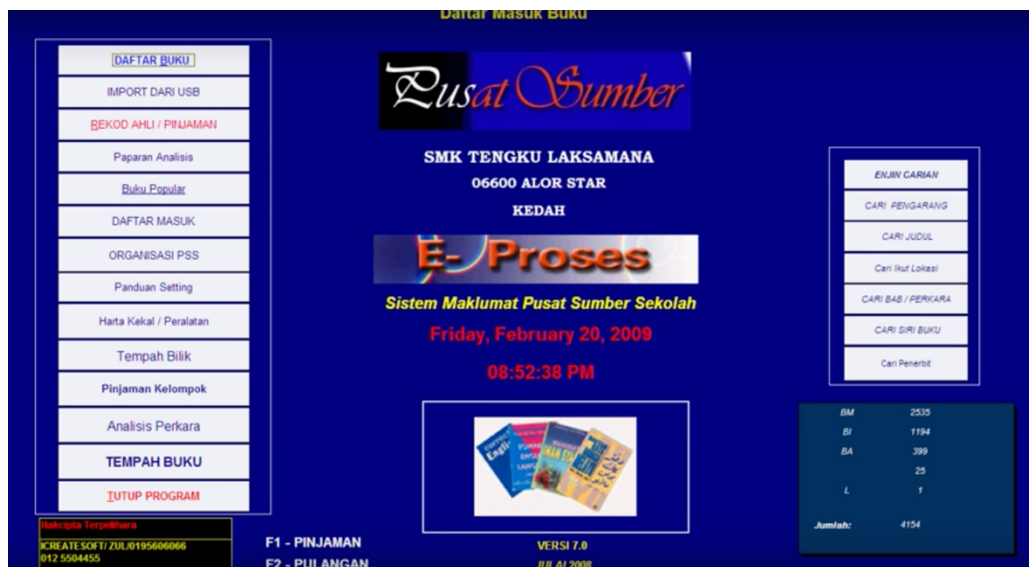


Figure 2.7 E-Proses Homepage

From this system's homepage, you can access all of the system's functionality. Some of this system's functionality is manage catalogue, generate reports and item's

circulation. To use this system, you first have to install it on your machine. This system can only be installed and run on a windows machine. A step by step guide is taught through phone on how to install and use the system. This system may be installed on multiple machine and have a centralized database between these machine. You cannot access this system from outside of the installed machines.

### 2.3.3 SENAYAN Library Automation System (SLiMS)

SLiMS is free to download and use provided you have some technical knowledge to install and run it. Just like the other 2 system, this system also fulfilled most of the basic functionality or requirements of a management system.

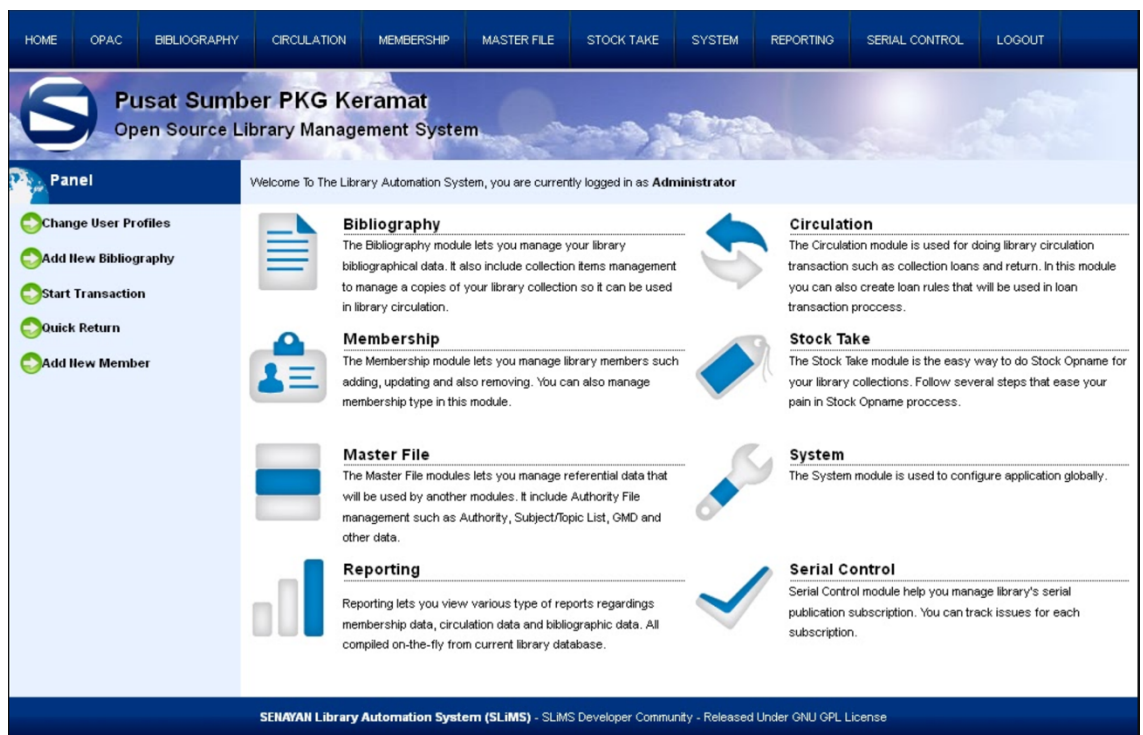


Figure 2.8 SLiMS's Module Page

This system do not require any other software or hardware updates for continuo us used. If you want to use a latest version of this system, you have to redownload it a nd do a fresh installation of the whole system. SLiMS is free to download on the intern et, the owner or developer of the system does not guide on you on the installation proc ess themselves. Instead, you can watch the installation tutorial on their blogpost on ho w to install.

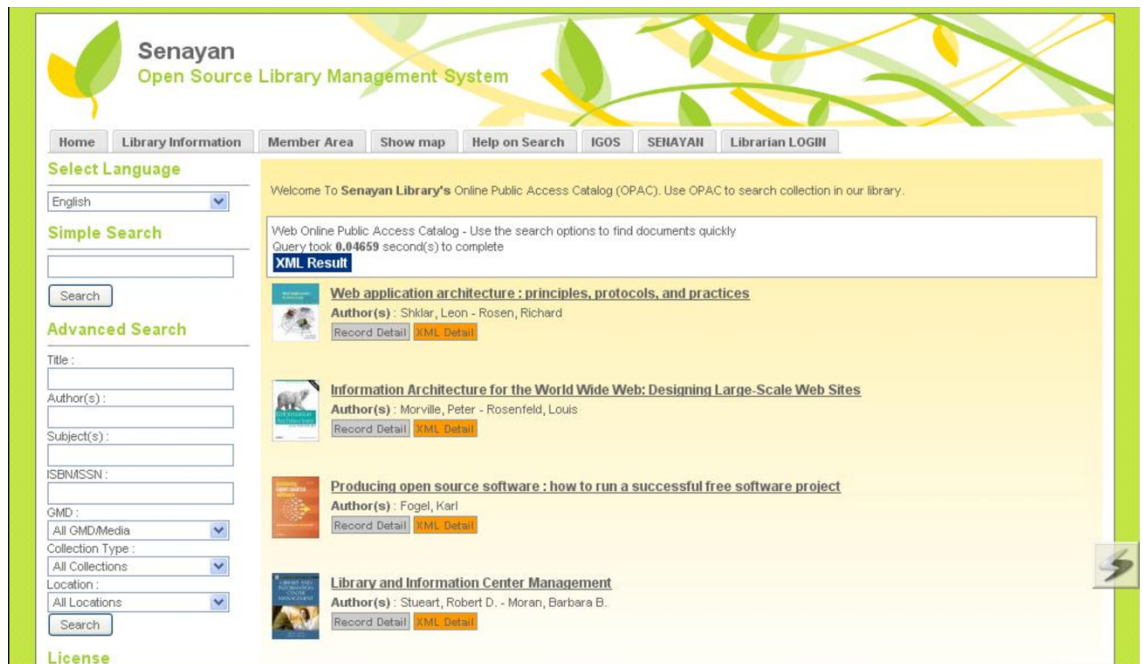


Figure 2.9 SLiMS's Homepage

Some of the systems functionality are manage catalogue(Bibliography), membership, Circulation, reporting and serial control. There are other extra ones like config where you can adjust the config of the system module globally. SLiM's possess 2 way of searching, one is a simple search where u key in the item's name you are looking for. The other one is an advanced search where you can search or sort the item with extra details to get better results.

All of the functions is only accessible by users with admin or teacher privileges . That includes the circulation module. This may be hard for students who wishes to borrow any book when no one is attending the system. Student's can't view resource's catalogue or status to know the book's availability.

## 2.4 COMPARISON BETWEEN EXISTING SYSTEM

### 2.4.1 System Comparison

Existing system that has been chosen to be analysed is then compare. The results is then recorded in the table 2.1

Table 2.1 Existing System Comparison

|                            | <b>Tarantula</b>  | <b>E-Proses</b>   | <b>SLiMS</b>  |
|----------------------------|---|---|---|
| Deployment                 | <ul style="list-style-type: none"><li>• Web-Based</li></ul>   | <ul style="list-style-type: none"><li>• Stand-Alone System</li></ul>  | <ul style="list-style-type: none"><li>• Web-Based</li></ul>   |
| Functions                  | <ul style="list-style-type: none"><li>• Login</li><li>• Add New User</li><li>• Manage Catalogue</li><li>• Searching</li><li>• Reporting</li></ul> | <ul style="list-style-type: none"><li>• Manage Catalogue</li><li>• Reporting</li><li>• Print Reports</li><li>• Item Circulation</li><li>• Searching</li></ul> | <ul style="list-style-type: none"><li>• Searching</li><li>• Manage Catalogue</li><li>• Reporting</li><li>• Circulation</li><li>• Membership</li></ul> |
| Supported Operating System | <ul style="list-style-type: none"><li>• Any OS with web browsers.</li></ul>   | <ul style="list-style-type: none"><li>• Windows</li></ul>   | <ul style="list-style-type: none"><li>• Any OS with web browsers</li></ul>  |

### 2.4.2 Advantages Comparison

Advantages of the existing system that has been chosen to be analysed is then compare. The results is then recorded in the table 2.2.

Table 2.2 Existing System Advantages Comparison

| System Name | Tarantula   | E-Proses  | SLiMS  |
|-------------|---|---|--|
| Advantages  | <ul style="list-style-type: none"> <li>• No Installation required</li> <li>• One-time payment for lifetime</li> <li>• Can be accessed from anywhere</li> <li>• No software or hardware upgrade needed</li> <li>• Multi-User</li> <li>• Cloud Based data management</li> </ul> | <ul style="list-style-type: none"> <li>• Easy Installation</li> <li>• One-time payment for lifetime</li> <li>• Import student's data into the system</li> <li>• Bar-Code Scanner</li> <li>• No software or hardware upgrade needed</li> <li>• Centralized Database</li> </ul> | <ul style="list-style-type: none"> <li>• Require some technical skill to install</li> <li>• Free to download and use</li> <li>• No software or hardware upgrade needed</li> <li>• No internet connection required</li> <li>• Centralized Database</li> </ul> |

### 2.4.3 Disadvantages Comparison

Disadvantages of the existing system that has been chosen to be analysed is then compare. The results is then recorded in the table 2.3.

Table 2.3 Existing System Disadvantages Comparison

| System Name   | Tarantula   | E-Proses  | System 3   |
|---------------|---|---|--|
| Disadvantages | <ul style="list-style-type: none"> <li>• Internet is required</li> <li>• Users have limited control and flexibility</li> <li>• No Circulation Module</li> <li>• Does not have any module that handle student's information</li> </ul> | <ul style="list-style-type: none"> <li>• Can only be used on Windows machine</li> <li>• Can only be accessed in the school's SRC</li> <li>• Vulnerability to attack</li> <li>•</li> </ul> | <ul style="list-style-type: none"> <li>• Can only be accessed on the installed machine</li> <li>• Can only be accessed in the school's SRC</li> <li>• Vulnerability to attack</li> </ul> |

## **2.5 CONCLUSION**

SRC Management system provides an easy way for school teacher's to manage and records the SRC's resources like books. Most of SRC management system in the market offers useable functions such as resource info, manage catalogue and resource searching to fulfil the basic or most important requirement of SRC management system should possess. Although these systems have its own advantages and disadvantages, none of them are able to fulfil an SRC individual needs, which is why this project is proposed. Several improvements is applied that is tailored to what the client needs for his project as compare to other existing system. This project focus more on both sides, the students and the teacher which most of the existing system is lacking for their own convenience.

## CHAPTER 3

### METHODOLOGY

#### 3.1 INTRODUCTION

This chapter discuss about the methodology chosen to be used for constructing the software development process in this project. “Methodology is the systematic, theoretical analysis of the methods applied to a field of study, or the theoretical analysis of the body of methods and principles associated with a branch of knowledge” ([https://en.wikipedia.org/wiki/Methodology#Methodology\\_as\\_a\\_buzzword](https://en.wikipedia.org/wiki/Methodology#Methodology_as_a_buzzword), 2018). Methodology are often considered as framework that is used to plan, control and structure the development process of any information system which is now more commonly known as software development methodology in software engineering.

As time passed, many methodology has evolved into a better version to better suit modern software development and each of these methodology has their own advantage and disadvantage. One of the methodology that has rapidly evolve in the recent years is called Rapid Application Development or famously known by people as RAD and is picked to be applied for this project. None of the methodology are expected to solve or suit every software development projects as the nature of these projects usually differs from one another and depends on the type of projects, technical organization and team considerations.

The reason why I chose RAD despite some of its disadvantages is because of its life cycle stages that allows changes to initial requirements or addition of new requirements during user design phase to further refine the systems that later be develop. This also ensure that it truly cater to user needs. Furthermore, RAD’s advantages far outweighs its disadvantages.

## 3.2 METHODOLOGY

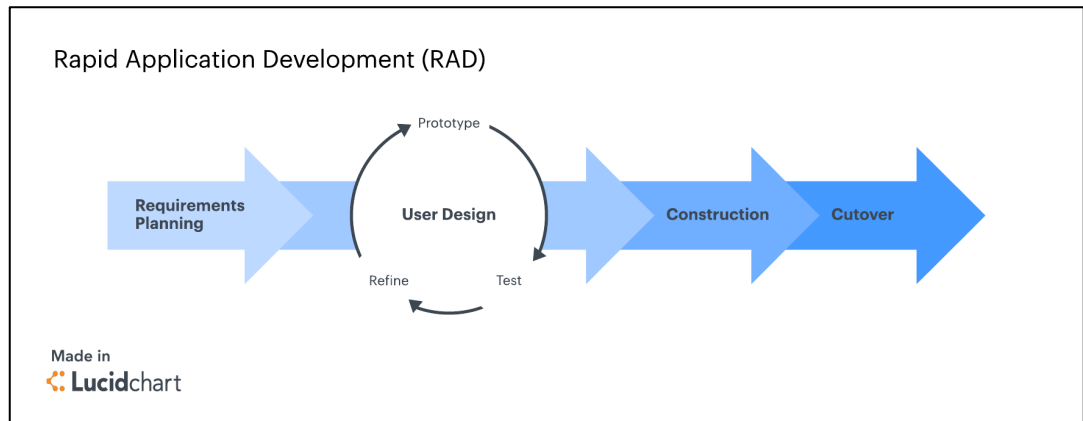


Figure 3.1 Life Cycle Stages of RAD

Figure 3.1 shows RAD (Rapid Application Development) life cycle. Requirement planning, user design construction and lastly cutover is the 4 stages that makes up full RAD life cycle. Each and every phases in the life cycle is unique with its own purpose and task and are very important and essential to project completion. (Rapid Application Development, 2018)

### 3.2.1 Requirement Planning Phase

System planning and System analysis phases of the SDLC (System Development Life Cycle) is combined to make up for the first stage of RAD, the Requirement Planning phase. This phase aims to aids in managing time, quality, cost, change, risk and issues that may surface during the duration of this project. This is all for the purpose of delivering the project in time and within the allocated budget allowed for the project.

Several meetings is conducted between requirements planning team and client in this phase to discuss on matters regarding the business needs, project scope, constraints and system requirements. This phase end when both party finds agreement on important key issues and have authorization to continue. (Requirement Planning Phase, 2013)

#### 3.2.1.1 Research Current Situation

The stage starts when the research team meets with the client at SMK SERI NILAM. To find the requirements the current system is lacking, the system is first analysed. Every information gathered from the system is then analysed once again to



create a list of initial requirements. The requirements is then compare the previous school resource management system for the purpose of looking any similarities, advantages and disadvantages between them.

### 3.2.1.2 Define Requirement

This is the stage where the stakeholders explain what is their expectations for the system. The purpose of the system, the systems functionality, what the system need to do and how many user or actor uses this system is fully explained by the stakeholders. The initial requirements listed in the 1<sup>st</sup> task is then be compared with the stakeholder’s expectation. Final system requirements is defined once the requirement satisfy the stakeholder’s expectations.

### 3.2.1.3 Finalize Requirement

This is the part where the team document the finalized requirement of the system. The stakeholders need to approve the documents. Context diagram and use case diagram is used to explain how the system works to all parties involved.

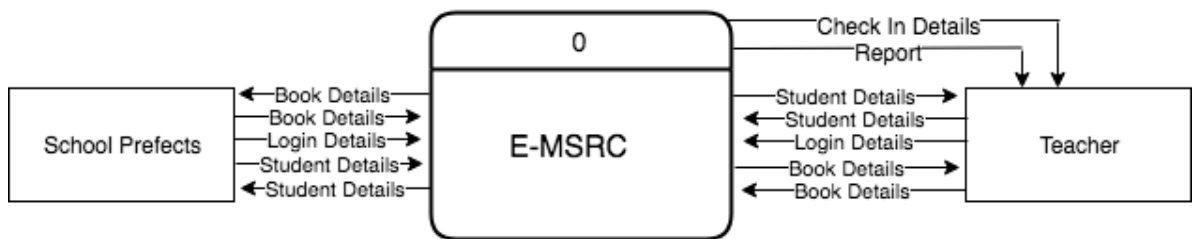


Figure 3.2 Context Diagram

Figure 3.2 shows the context diagram of the system. The purpose of this context diagram is to show the interaction between users and the system. The diagram makes it easy for stakeholders to understand the data flow of the system to ensure that it is the same as what they want. Refer SRS in Appendix B.

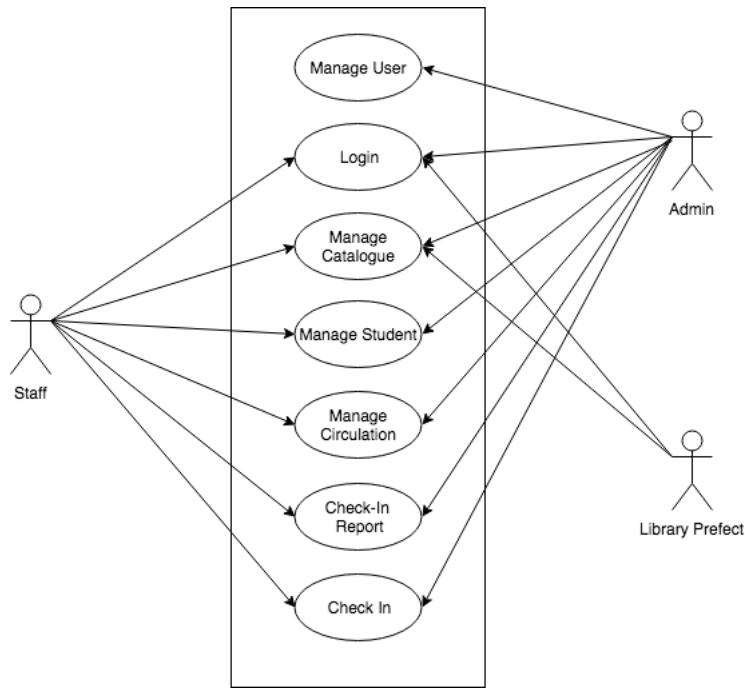


Figure 3.3 Use Case Diagram

Figure 3.3 shows the use case for the system. This use case is shown and explained to the stakeholders to ensure that it meets the requirements of the system. Refer SDD in Appendix C.

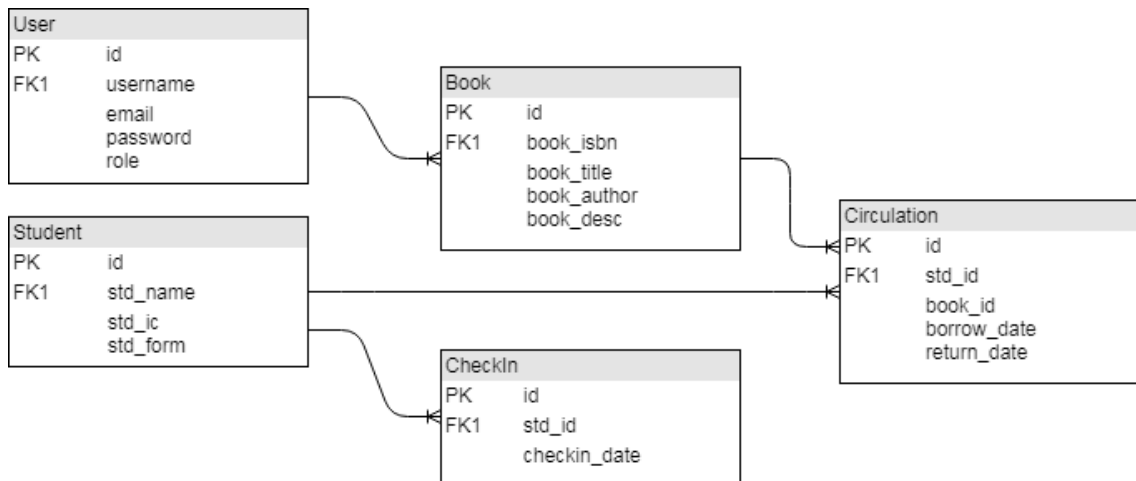


Figure 3.4 Entity Relational Diagram

Figure 3.4 shows the Entity Relational Diagram(ERD). This is the proposed database design of the system.

### **3.2.2 User Design**

This is the part where models and prototypes with full system processes, inputs and outputs that represents the real system is designed and develop by system analyst. RAD typically use a combination of Joint Application Development (JAD) techniques and CASE tools to translate user needs into working models. To let users understand, modify and eventually approve a working model of the system, the user design is processed continuously until it fulfil the user's needs.

#### **3.2.2.1 Produce Detailed System Area Model**

The details that is created in stages before this is transformed into system area model that better explain the system's design. The system scope is then refine based on the created models to make sure that the system's critical parts is delivered in time.

#### **3.2.2.2 Wireframes / Proposed System's User Interface**

This is the stage where the system is designed. Outlines of the interface is developed based on the area models. Once the interface is completed, a prototype is created by connecting the interfaces using a prototyping software. Required functions for the system, reusable design components, system structures and layouts of screens to be supported by the system are the deliverables that is produced in this phase.

#### **3.2.2.3 Refine System Design/Revisions**

The prototype and the system design is then reviewed one more time in this stage to be analysed. The result of the analysis is used to perform a detailed design of the system. The design is then verified to ensure it meets the user requirements. Any inconsistencies from missing modules or functions should be resolved and verified again. Technical and computer oriented detailed system design is generated from the general system design to the level of details that allows you to start development.

#### **3.2.2.4 Implementation Strategies**

This is where implementation strategies is created. A list is make that list out all of the task required to develop the system and convert it to operational use. The approach for implementation is selected after the system design is reviewed. An estimate is made

to estimate the effort and time required to finish each task and summarized into overall estimation.

#### **3.2.2.5 Finalize System Design**

To finalize system design, a meeting is conducted with the system stakeholders. Any changes to the design is suggested by the stakeholders on this stage if necessary. Any design issues is discussed and resolved if they have significant impact to the system development process. Any unforeseen problems or invalid assumptions in the plan is checked during the presentation of the implementation plan.

#### **3.2.2.6 Obtain Approval**

The result of the meeting is finalized and incorporated into the system design, implementation plan. Then the approval to proceed to the construction stage is needed to continue.

### **3.2.3 Construction**

The system is develop in cycles of development, testing and lastly refine requirements again until the development process ended..

Users or stakeholders of the system still continue to participate In this stage and can still suggest changes or improvement if the time allows for it as the system is being developed. Programming, coding, unit integration, application development and system testing are the task included in this stage.

#### **3.2.3.1 Setup Development Environment**

Before development began, development environment must first be setup. Since the development uses Laravel web framework, all the required software need to be installed and configured first. Setup and install XAMPP, GIT and Composer. Once finish, install Laravel.

#### **3.2.3.2 Develop System**

The system use all of the language required to build a web based application such as, Html, CSS and JavaScript. Since the system is developed on top of Laravel, a PHP web framework so PHP language is used. The system is develop using the finalized requirements.

#### **3.2.3.3 Generate Test Data and Document**

Several test on the Test methodology is conducted. The test is applied to verify that the system fulfil the client's need and is free or programmer's error. A full document on how the system is develop and run is recorded.

#### **3.2.3.4 Prepare for Implementation**

The activity in this task is the implementation of the system. First, In case of system failure, the initial plan for implementation activities and also the contingency plan is prepared. The implementation procedures is then produced and any problems that may arise related to the system deployment is resolved.

### **3.2.3.5 Verify System Requirement**

E-MSRC go through a series of test to ensure that the system that has been built cater to user needs.

### **3.2.4 Cutover**

Cutover stage is the last stage in the life cycle. The task in this phase include install production system, data conversion, user testing and user training. The entire process is different from traditional methods as they are compressed. The new way results in quick delivery time of the system and is placed in production faster.

#### **3.2.4.1 Install Production System**

The system production environment and the actual system is installed and configured on client's server. Necessary adjustment is made to the hardware and/or software to fit the production environment.

#### **3.2.4.2 Conduct Training**

A training session is conducted to teachers of SMK Seri Nilam to show and teach of how the system need to be operated.

#### **3.2.4.3 Perform Data Conversion**

Any data from the old system is converted into a format that is accessible by E-MSRC.

#### **3.2.4.4 Accept System Installation**

E-MSRC may be consider as successful if the system able to operates for a specific period of time within defined tolerances for performance, error rate and usability. User acceptance test is perform in this task. The acceptance of the system is based on the agreements among the users, stakeholders, and system documentation.

### 3.3 Hardware and Software

#### 3.3.1 Hardware Requirement

A list of hardware that is used throughout the development of this projects is listed in below in table 3.1

Table 3.1 Hardware Requirement

| Hardware                    | Specification  | Purpose  | Qty. |
|-----------------------------|--|--|------|
| Laptop<br>Personal Computer | 2.2 GHz Intel Core i7<br>16 GB 1600 MHz DDR3<br>RAM<br>macOS High Sierra | <ul style="list-style-type: none"><li>• To prepare system documentations</li><li>• To design and develop the system</li><li>• To test the system</li></ul> | 1    |
| External Hard Disk          | Seagate 2TB  | <ul style="list-style-type: none"><li>• Act as backup solution</li></ul>   | 1    |
| Printer                     | HP Desk Jet 2130 Series  | <ul style="list-style-type: none"><li>• To print</li></ul>   | 1    |

### 3.3.2 Software Requirement

A list of software that is used throughout the development of this projects is listed in below in table 3.2.

Table 3.2 Software Requirement

| Software                        | Purpose   |
|---------------------------------|---|
| Microsoft Office                | <ul style="list-style-type: none"><li>• To document the project</li><li>• To create Gantt chart</li><li>• To prepare presentation slide</li></ul> |
| IBM Rational Software Architect | To design UML diagram this project use  |
| Adobe XD                        | To design system's wireframe  |
| Sublime Text 3                  | As text editor to write codes for the system  |
| Sequel Pro                      | As Database Management System to manage system's database   |

### 3.4 Gantt Chart

This project will be developed within 2 period of time which is PSM 1 and PSM 2. The work or task in this projects is distributed using RAD and every task is given a specific time frame to be completed. Requirements planning and user design phase of RAD will be conducted in PSM 1 while construction phase and transition phase of RAD will be conducted in PSM 2. The project starts on 03/12/2018 and ends on 30/06/2019. Refer Appendix A for the details of the Gantt chart.



## CHAPTER 4

### IMPLEMENTATION, TESTING AND RESULT DISCUSSION

#### 4.1 INTRODUCTION

In this second last chapter, plans for implementation of E-Management for School Resource Center is documented. The implementation plan list every task that is required to develop the E-MSRC system. Once the implementations is completed, test data need to be generated for the system to be tested. Multiple test is applied to the E-MSRC system to verify the function and to find any error the system may have. By utilizing rapid application development methodologies, testing is produced and executed during the rapid construction phase.

#### 4.2 IMPLEMENTATION

This section explains all of the specific implementation requirements and procedures. Every process involved in the project development is recorded. The first part of the development process is setup development environment and Laravel installation.

##### 4.2.1 Development Environment

To start developing, you first have to setup your development environment. In order to start coding, first you need a text editor where you write your code in and a web server to run the code that you have written for various purposes including testing. My text editor of choice is Visual Studio Code because it have most of the functionality I need for coding and an integrated version control, git inside the text editor. Visual Studio Code is a Text Editor develop by Microsoft for all 3 operating system, Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets and code refactoring. You can also customize the text editor to fit your preferences.

For web server, I use XAMPP as it is easy to setup. All you have to do to setup xampp is download and install in your computers. What makes XAMPP a good choice is because the package have everything a web server need. XAMPP stands for Cross-

Platform (X), Apache (A), MariaDB (M), PHP (P), and Perl (P). It is a lightweight and simple Apache distribution that allows it to be extremely easy for this Project to create a virtual local web server. Most of the actual web server deployments in the web use the same components as XAMPP, so that allows transitioning this system from a local server to the live server easy. XAMPP comes with a several modules including OpenSSL, phpMyAdmin, MediaWiki, Joomla, Wordpress and more. Since phpMyAdmin is already bundled with XAMPP installation, phpMyAdmin is chosen as database management system for this project development.

After installing XAMPP, then proceed to install Laravel. Laravel requires Composer, a tool for dependency management in PHP to setup and install. Download and install Composer in the computer. After the installation is finished, use Composer to download Laravel installer.

```
composer global require "laravel/installer=~1.1"
```

Figure 4.1 Command for installing Composer

Once the installer is installed, by simply typing “Laravel new” command in the terminal create a fresh Laravel installation. Since the project used XAMPP, Laravel must be installed in htdocs folder inside XAMPP directory for it to work. The first thing that I do after installing Laravel is to run “composer install” to install all of Laravel dependencies. Then create a .env file in the Laravel directory and generate a key in the .env by “key:generate” command.

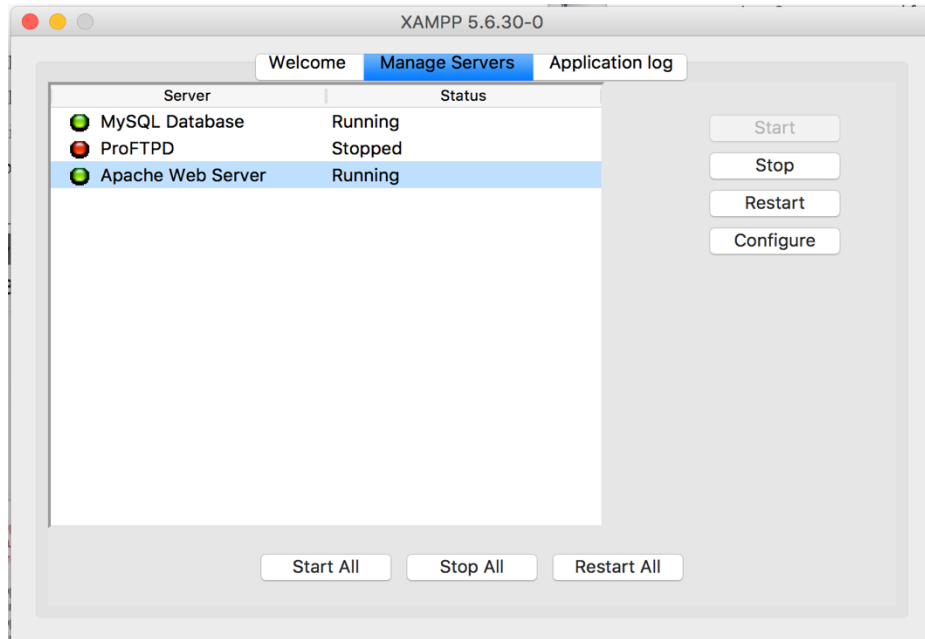


Figure 4.2 XAMPP Manager

To test whether your installation is working or not, first you need to start apache Web Server and MYSQL. But in this case MySQL is not yet needed. After you both MySQL Database and Apache Web Server is started then run “php artisan serve” command in command prompt or terminal in the system’s directory and navigate to <https://localhost:8000/>

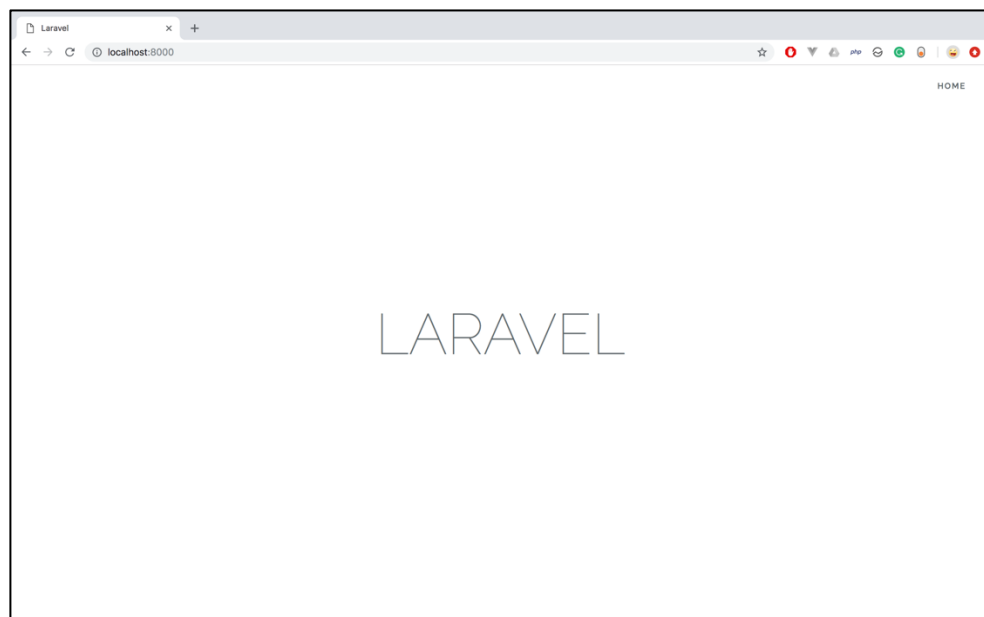


Figure 4.3 Laravel Home

If you see figure then it means your Laravel installation is successful. Once the installation is finished, then proceed to setup the database. Database management that is already provided in XAMPP is PHPMYAdmin. Figure below shows the database management system, PHPMYAdmin.

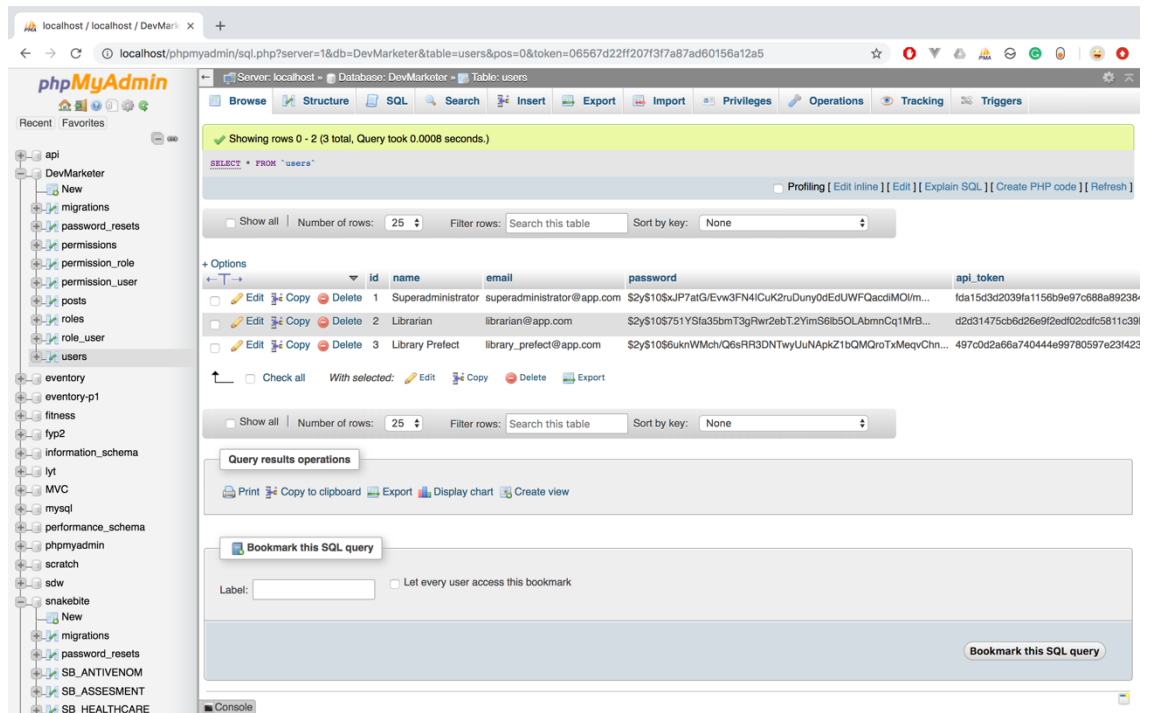


Figure 4.4 PHPMYAdmin

Database management system, PHPMYAdmin is used throughout the development to manage the system’s database. PHPMYAdmin allow create, update or delete any data in the database directly, usually to help with development and testing process.

## 4.2.2 System Functionality

This part describe about the functions that exist in the system, how the system interacts with the user. One of the important thing is the user interface as it must be user friendly to ensure that the user understands the system.

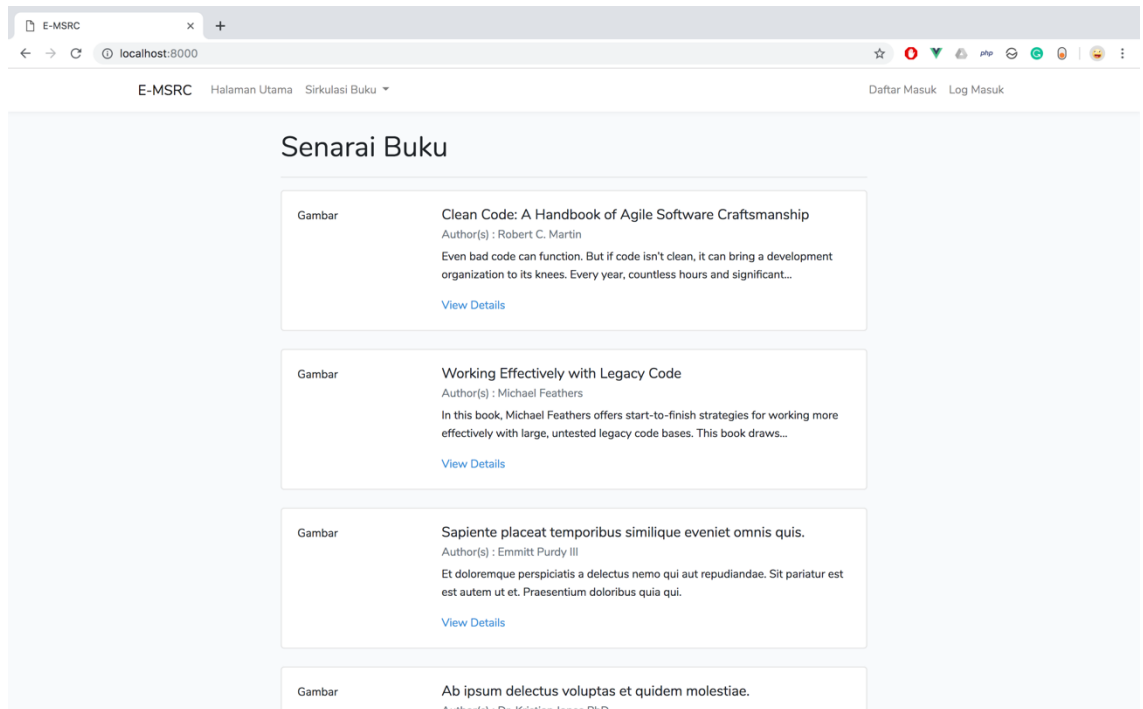


Figure 4.5 E-MSRC Homepage

Figure 4.5 shows a listing of book's available in the system. And on the top of navigation page are 3 other options which is Circulation, Check In and Login which is explained later in this section.

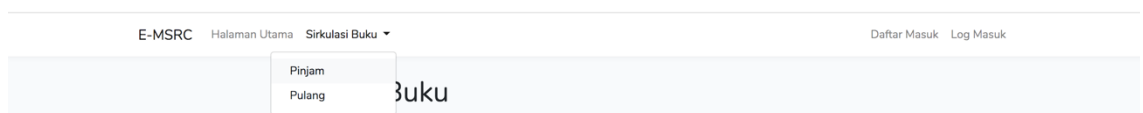


Figure 4.6 E-MSRC Navbar

Figure 4.6 above shows a book circulation dropdown. Choosing “pinjam” opens a book borrow page while choosing “pulang” opens a return borrowed book page. These

2 pages are restricted page only allowed to access once a staff/library's prefect is logged in.

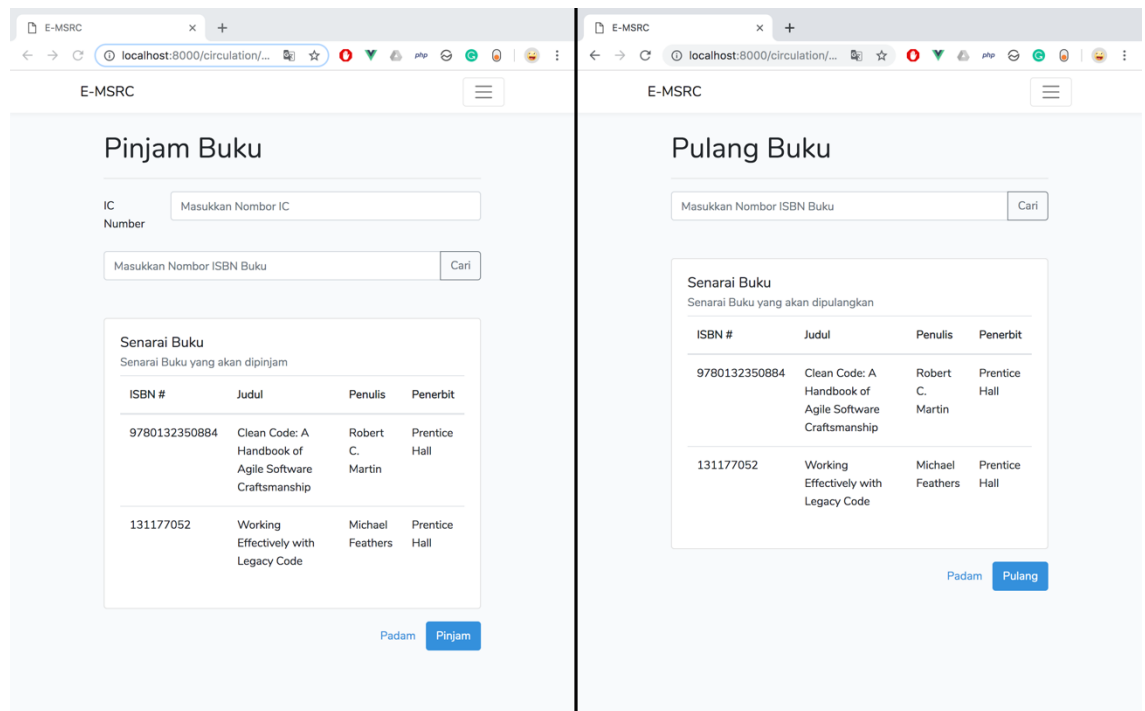


Figure 4.7 E-MSRC Borrow and Return page

Figure 4.7 above shows 2 interface, the left side is borrow book interface and the right side is return borrowed book interface. On the left side you can see, there's an input textbox with a search button. First you have to search the book that you want to borrow and it appear in the list. You do the same thing for return book too. The only difference from the two is, for borrow you have to input IC number before borrowing.

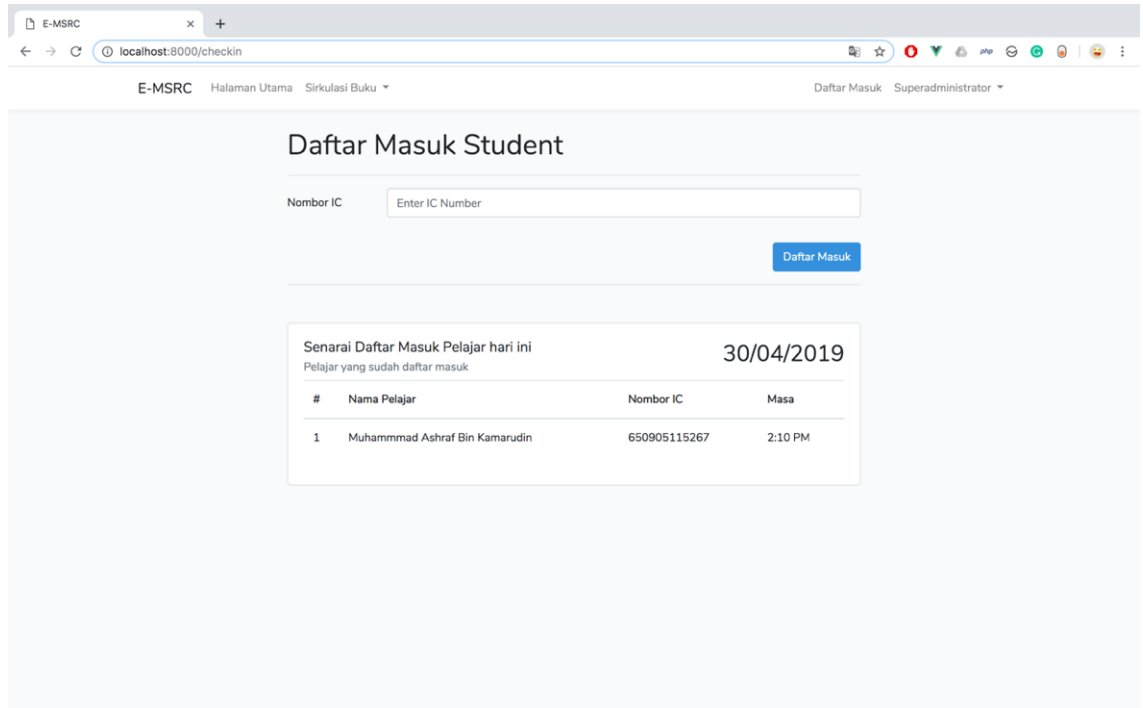


Figure 4.8 E-MSRC Student Check In

This interface shows recent Check-in students and also allow student to check in. Just input the student’s IC in the input box and click “Daftar Masuk” button and the check-in process is done provided that the info entered is correct or else an error message pops up.

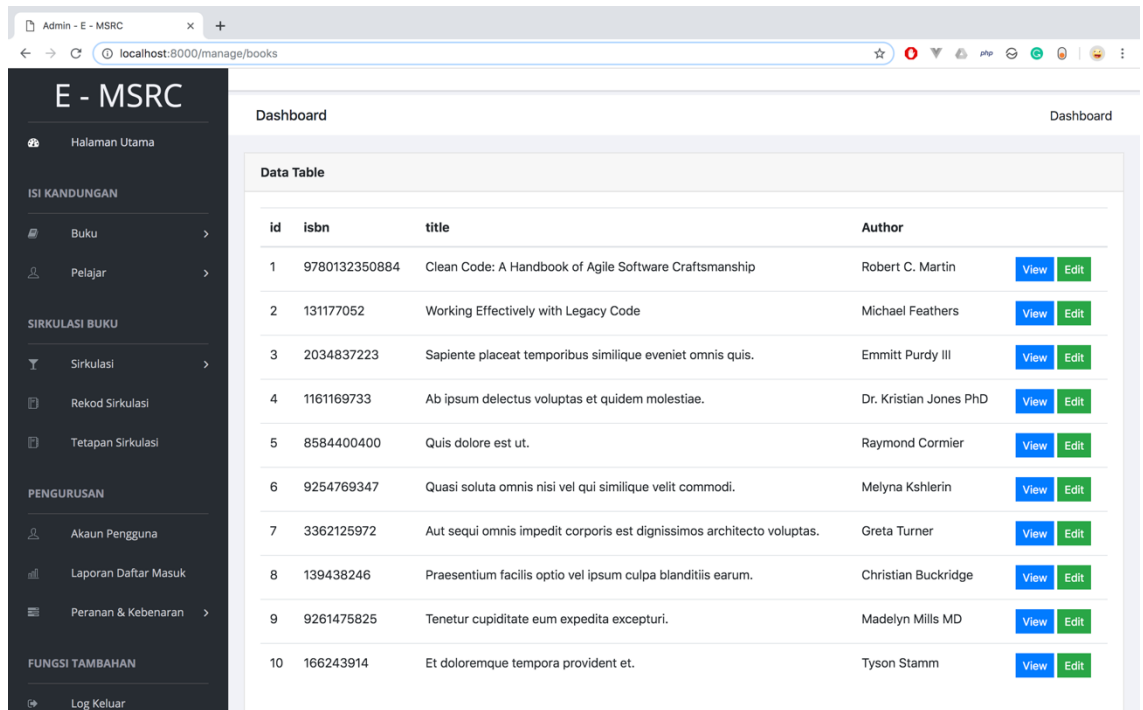


Figure 4.9 Manage Book

This is where manage book modules happen. You can create, Update and delete books available in the database here. A new form is shown once you choose to edit the book.



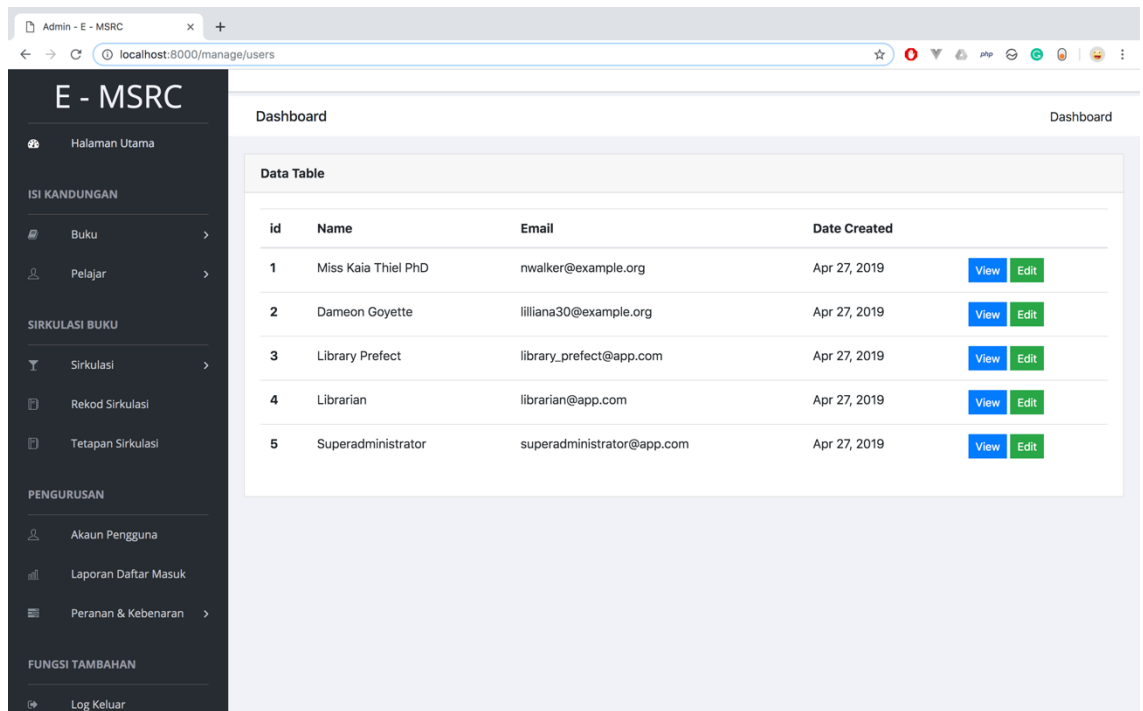


Figure 4.10 Manage User

This is where admin manage user. Admin have the option to change the role each user have to allow or remove access of a certain user to a certain functions or module included in the system.

### 4.3 Testing and Result discussion

E – Management for School Resource Centre is develop using PHP Web Framework, Laravel, JavaScript, Html and CSS programming language and uses MYSQL as database. Testing is crucial to the success of this project as it ensure the system is running as it should be. The system is tested during and after development is completed.

#### 4.3.1 Functional Testing

“Software Testing Methodology is defined as strategies and testing types used to certify that the Application Under Test meets client expectations”. (Software Testing Methodologies: Learn QA Models , n.d.).

Functional Testing is in the Software Testing methodology, and is consist of unit testing, integration testing, system testing and acceptance testing and is usually executed in this order. These testing is described below :

#### I. Unit Testing

Unit testing is a part of testing methodology where individual units or component of a system are tested. A unit is the smallest testable part of a software or system, usually require few or one input and a single output. A unit can be an individual program, a function or procedure. If unit testing failed or produce an error, developers fix the error and run the test again until it runs with no errors.

#### II. Integration Testing

Integration Testing is run after all Unit Test is passed. This is the part where units or components are combined and tested as a group. This testing is needed to expose faults that may occur during the interaction between integrated units or components. Once these test is passed, it proved that all parts able to interact with one another without any faults or conflicts.

#### III. System Testing

System Testing is run after Integration Testing. This is the part where a complete and integrated system is tested. This test is needed to evaluate the system's compliance with the functional and non-functional requirements and ensure that they have been met.

#### IV. User Acceptance Testing

User Acceptance Test is the last part of the testing process after System Testing. In this part, actual users of the system test the system to make sure it can handle the tasks in real-world scenarios and meet all the defined requirements of the client.

## **CHAPTER 5**

### **CONCLUSION AND FUTURE WORK**

#### **5.1 INTRODUCTION**

E-MSRC is a management system to help library's staff manage libraries resources and staff. Many of the library management systems available in the market cannot fully fulfilled the requirements that the client need so E-MSRC is developed. E-MSRC is developed to be web-based instead of a stand-alone or a mobile application as web-based system fulfilled the client's needs. There are several areas in which the system can be further improved.

#### **5.2 RESEARCH CONSTRAINT**

##### **5.2.1 Development Constraint**

Development Constraint is obstacle or challenges that is faced during development phase. These may be physical, mechanical, time and others during key development stage. All of the constraints during the development stage is listed below.

##### **I. Time Constraint**

Short time is needed to develop a basic functionality of a function or module but more time is required to fulfil the non-functional requirements of the system and to make it more robust so that the functions or module did not break unexpectedly during production stage. These functions need to be carefully tested and design to handle every types of users that may use the system.

##### **II. Setup Constraint**

Several software are required to be installed in the development laptop before development begin. Development environment need to have git and composer installed and setup before Laravel framework can be installed and used.

### **5.2.2 System Constraint**

System constraint is constraints that the system currently have which could not be avoided. This constraint restrict a degree of freedom for use in the system to ensure the system running perfectly with no errors.

#### **I. Platform**

The developed system is a web-based software so the system can only be run on web browsers. Any and every device with web browsers installed may access the system and all of the functionalities the system offers but the interface of the system may differ on each device screen size.

#### **II. Language**

The intended users of the system is malay school teachers and students thus the system uses Malay language and only user who understands Malay may use it. This is stated in the initial requirements hence it can't be changed.

### **5.3 FUTURE WORK**

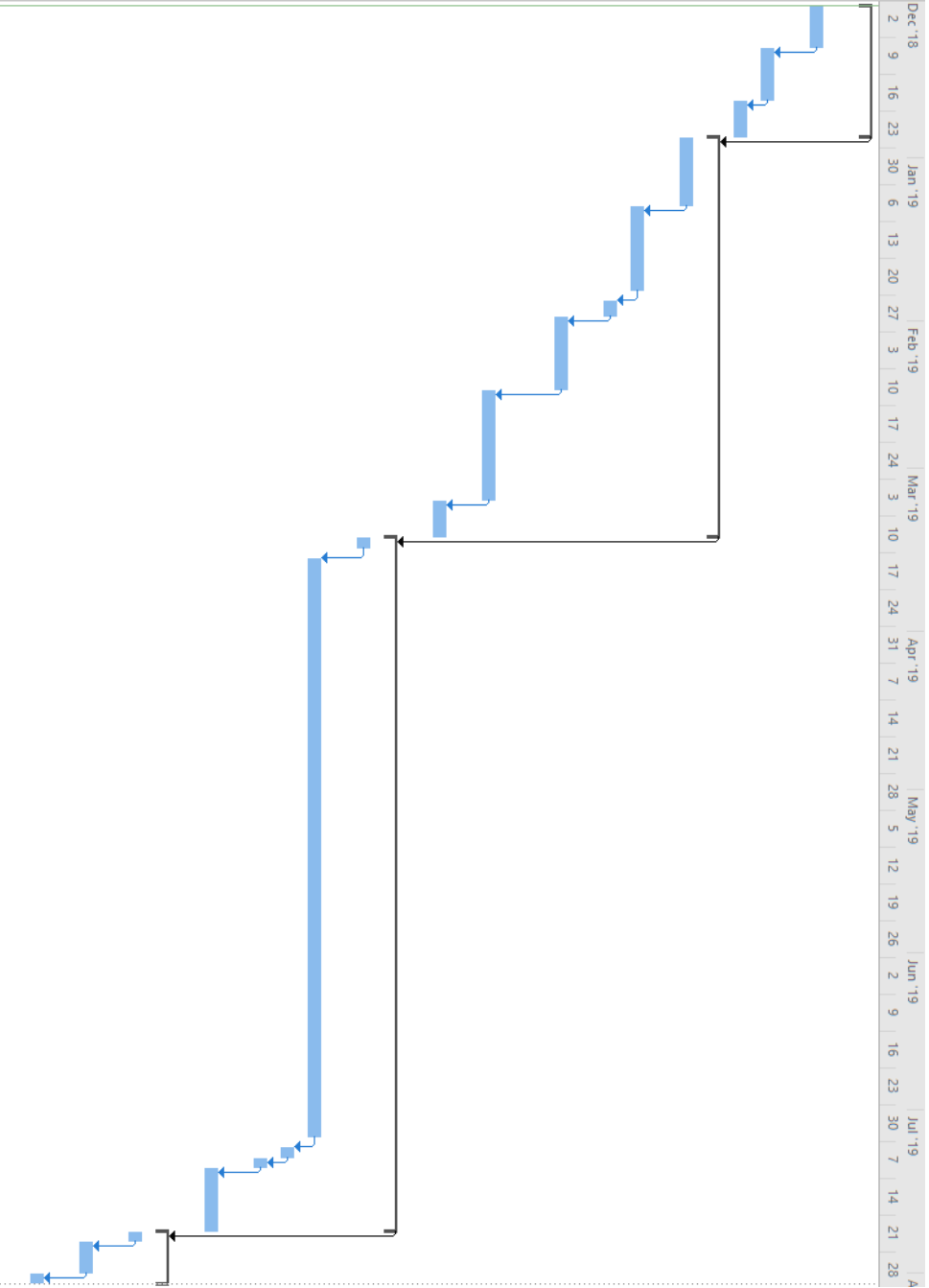
Further improvements can be made to the system on several aspects. One of the aspects that can be improved is, to have a new module that able to see and print receipt on books that have been returned. Reporting module can also be further improve by having users with the rights to view and generate report some options as on what kind of data they wanted to see in the generated report. Other than that, The system's messages can also be further improve by having all actions to display message on what has happened to let user know if an operation is a success or not.

## REFERENCES

- ZAINUDIN, F. (n.d.). *school resources centre in Malaysia*. Retrieved from sumberdigital.wordpress.com: <https://sumberdigital.wordpress.com/2016/02/15/school-resource-centres-in-malaysia/>
- What is Apache ? - What is web server ?* (n.d.). Retrieved from <https://www.wpbeginner.com/https://www.wpbeginner.com/glossary/apache/>
- Web Browser - Wikipedia*. (n.d.). Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Web\\_browser](https://en.wikipedia.org/wiki/Web_browser)
- MySQL - Wikipedia*. (n.d.). Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/MySQL>
- Requirement Planning Phase*. (4 June, 2013). Retrieved from <https://testingtypes.wordpress.com/https://testingtypes.wordpress.com/tag/requirements-planning-phase/>
- Rapid Application Development*. (12 August, 2018). Retrieved from [https://en.wikipedia.org/https://en.wikipedia.org/wiki/Rapid\\_application\\_development](https://en.wikipedia.org/https://en.wikipedia.org/wiki/Rapid_application_development)
- [https://en.wikipedia.org/wiki/Methodology#Methodology\\_as\\_a\\_buzzword](https://en.wikipedia.org/wiki/Methodology#Methodology_as_a_buzzword). (10 November, 2018). Retrieved from Wikipedia: <https://en.wikipedia.org>
- Dvorski, D. D. (2007). *INSTALLING, CONFIGURING, AND DEVELOPING WITH XAMPP. INSTALLING, CONFIGURING, AND DEVELOPING WITH XAMPP.*
- Martin, J. (1991). *Rapid Application Development*. Macmillan Publishing Company.
- Murch, R. (2012). *Rapid Application Development Lifecycle - The Complete Guide*.
- Software Testing Methodologies: Learn QA Models* . (n.d.). Retrieved from Guru99: <https://www.guru99.com/testing-methodology.html>

**APPENDIX A  
GANTT CHART**

| Task Mode | Task Name                        | Duration       | Start        | Finish       |
|-----------|----------------------------------|----------------|--------------|--------------|
| ▶         | ▶ <b>Requirement Planning</b>    | <b>19 days</b> | Mon 12/3/18  | Thu 12/27/18 |
| ▶         | Research Current Situation       | 6 days         | Mon 12/3/18  | Mon 12/10/18 |
| ▶         | Define Requirement Situation     | 8 days         | Tue 12/11/18 | Thu 12/20/18 |
| ▶         | Finalize Requirement             | 5 days         | Fri 12/21/18 | Thu 12/27/18 |
| ▶         | ▶ <b>User Design</b>             | <b>54 days</b> | Fri 12/28/18 | Wed 3/13/19  |
| ▶         | Produce Detailed System Design   | 9 days         | Fri 12/28/18 | Wed 1/9/19   |
| ▶         | Wireframe/Protot                 | 12 days        | Thu 1/10/19  | Fri 1/25/19  |
| ▶         | Refine System Design             | 3 days         | Mon 1/28/19  | Wed 1/30/19  |
| ▶         | Prepare Implementation Design    | 10 days        | Thu 1/31/19  | Wed 2/13/19  |
| ▶         | Finalize System Design           | 15 days        | Thu 2/14/19  | Wed 3/6/19   |
| ▶         | Obtain Approval for Construction | 5 days         | Thu 3/7/19   | Wed 3/13/19  |
| ▶         | ▶ <b>Construction</b>            | <b>94 days</b> | Thu 3/14/19  | Tue 7/23/19  |
| ▶         | Setup Development                | 2 days         | Thu 3/14/19  | Fri 3/15/19  |
| ▶         | Develop System                   | 80 days        | Mon 3/18/19  | Fri 7/5/19   |
| ▶         | Generate Test Data               | 2 days         | Mon 7/8/19   | Tue 7/9/19   |
| ▶         | Prepare for Implementation       | 2 days         | Wed 7/10/19  | Thu 7/11/19  |
| ▶         | Verify System Requirement        | 8 days         | Fri 7/12/19  | Tue 7/23/19  |
| ▶         | ▶ <b>Cutover</b>                 | <b>8 days</b>  | Wed 7/24/19  | Fri 8/2/19   |
| ▶         | Install Production System        | 2 days         | Wed 7/24/19  | Thu 7/25/19  |
| ▶         | Conduct User Training            | 4 days         | Fri 7/26/19  | Wed 7/31/19  |
| ▶         | Accept System Transition         | 2 days         | Thu 8/1/19   | Fri 8/2/19   |



**APPENDIX B**  
**SOFTWARE REQUIREMENT SPECIFICATION(SRS)**



2018

# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

E-MANAGEMENT FOR SCHOOL RESOURCE  
CENTER

MUHAMMAD ASHRAF BIN KAMARUDIN CB16148

To be submitted to the  
Faculty of Computer System and Software Engineering  
Bachelor of Computer Science (Software Engineering)



## DOCUMENT APPROVAL

|   | Name                                     | Date              |
|---|--|-------------------|
| <p><b>Authenticated by:</b></p> <hr/> <p>Project Leader</p> | <p>MUHAMMAD ASHRAF BIN<br/>KAMARUDIN</p> | <p>08/03/2018</p> |
| <p><b>Approved by:</b></p> <hr/> <p>Project Manager</p>     |  | <p>08/03/2018</p> |

Software : draw.io diagrams

Archiving Place : Google Drive  
(<https://drive.google.com/open?id=1hKmRfYOcuySXjE3YQdgH-Mk8ecPMmfcw>)

Copies Available : 1 copy

## Table of Contents

|  |                   |
|--|-------------------|
| <b>DOCUMENT APPROVAL</b> .....                   | <b><i>i</i></b>   |
| <b>LIST OF FIGURES</b> .....                     | <b><i>iii</i></b> |
| <b>LIST OF TABLES</b> .....                      | <b><i>iv</i></b>  |
| <b>LIST OF APPENDIXES</b> .....                  | <b><i>v</i></b>   |
| <b>1. INTRODUCTION</b> .....                     | <b>6</b>          |
| 1.1 <b>PURPOSE</b> .....                         | <b>6</b>          |
| 1.2 <b>SYSTEM IDENTIFICATION</b> .....           | <b>6</b>          |
| 1.3 <b>SYSTEM OVERVIEW</b> .....                 | <b>8</b>          |
| 1.4 <b>REFERENCES</b> .....                      | <b>7</b>          |
| 1.5 <b>DOCUMENT OVERVIEW</b> .....               | <b>10</b>         |
| <b>2. PRODUCT DESCRIPTION</b> .....              | <b>8</b>          |
| 2.1 <b>PRODUCT PERSPECTIVE</b> .....             | <b>8</b>          |
| 2.2 <b>SYSTEM INTERFACES</b> .....               | <b>9</b>          |
| 2.3 <b>PRODUCT FUNCTIONS</b> .....               | <b>9</b>          |
| 2.4 <b>USER CHARACTERISTICS</b> .....            | <b>10</b>         |
| 2.5 <b>CONSTRAINTS</b> .....                     | <b>11</b>         |
| 2.6 <b>ASSUMPTIONS AND DEPENDENCIES</b> .....    | <b>12</b>         |
| <b>3. SPECIFIC REQUIREMENTS</b> .....            | <b>13</b>         |
| 3.1 <b>SOFTWARE PRODUCT FEATURES</b> .....       | <b>13</b>         |
| 3.2 <b>EXTERNAL INTERFACE REQUIREMENTS</b> ..... | <b>25</b>         |
| 3.2.1 <b>User Interfaces</b> .....               | <b>25</b>         |
| 3.3 <b>REQUIREMENTS TRACEABILITY</b> .....       | <b>28</b>         |

---

## LIST OF FIGURES

|                                  |    |
|----------------------------------|----|
| FIGURE 1 CONTEXT DIAGRAM .....   | 8  |
| FIGURE 2 USE CASE .....          | 9  |
| FIGURE 3 SEARCH USE CASE .....   | 13 |
| FIGURE 4 LOGIN .....             | 14 |
| FIGURE 5 MANAGE CATALOGUE.....   | 15 |
| FIGURE 6 MANAGE USER .....       | 17 |
| FIGURE 7 CIRCULATION .....       | 20 |
| FIGURE 8 REPORT.....             | 21 |
| FIGURE 9 CHECK IN .....          | 23 |
| FIGURE 10 DIALOGUE DIAGRAM ..... | 25 |

---

**LIST OF TABLES**

|   |    |
|---|----|
| TABLE 1 USE CASE DESCRIPTION .....        | 10 |
| TABLE 2 USER CHARACTERISTICS.....         | 10 |
| TABLE 3 SEARCH DESCRIPTION .....          | 13 |
| TABLE 4 LOGIN DESCRIPTION .....           | 14 |
| TABLE 5 MANAGE CATALOGUE DESCRIPTION..... | 15 |
| TABLE 6 MANAGE USER DESCRIPTION .....     | 18 |
| TABLE 7 CIRCULATION DESCRIPTION.....      | 20 |
| TABLE 8 REPORT DESCRIPTION.....           | 21 |
| TABLE 9 CHECK IN DESCRIPTION.....         | 23 |

---

**LIST OF APPENDIXES**

|                                       |    |
|---------------------------------------|----|
| APPENDIX A - 1 HOMEPAGE/SEARCH .....  | 32 |
| APPENDIX S - 2 LOGIN .....            | 32 |
| APPENDIX S - 3 MANAGE CATALOGUE ..... | 33 |
| APPENDIX S - 4 MANAGE STUDENT .....   | 34 |
| APPENDIX S - 5 CIRCULATION .....      | 35 |
| APPENDIX S - 6 REPORT .....           | 35 |
| APPENDIX S - 7 CHECK IN .....         | 36 |
| <br>                                  |    |
| APPENDIX B - 1 HOMEPAGE.....          | 38 |
| APPENDIX B - 2 BOOK DETAILS .....     | 39 |
| APPENDIX B - 3 CHECK IN.....          | 39 |
| APPENDIX B - 4 BORROW BOOK .....      | 40 |
| APPENDIX B - 5 RETURN BOOK .....      | 40 |
| APPENDIX B - 6 DASHBOARD .....        | 41 |

---

## 1. INTRODUCTION

---

### 1.1 PURPOSE

The purpose of the Software Requirement Specification (SRS) for E-Management for School Resource Center is to gather together all possible system requirement from relevant stakeholders.

This SRS document shows an overview of the context of E-Management for School Resource Center (EMSRC). It includes the system's aims, user interface, constraints, hardware specification, related diagrams and model used in the system.

### 1.2 SYSTEM IDENTIFICATION

SRS-EMSRC-2018-V1

EMSRC : E – MANAGEMENT FOR SCHOOL RESOURCE CENTER

ESTABLISH : 2018

V1 : VERSION 1

DOCUMENT TYPE : SRS

### 1.3 REFERENCES

[1] Pohl, K. (1 st ed.). (2010). *Requirement Engineering: Fundamental, Principles and Techniques*. Germany: Springer.  
<http://www.requirements-book.com/>

[2] Software Planning & Requirement Workshop (2017). *SRS Template: Software Requirement Specification (SRS)*.



## 2. PRODUCT DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

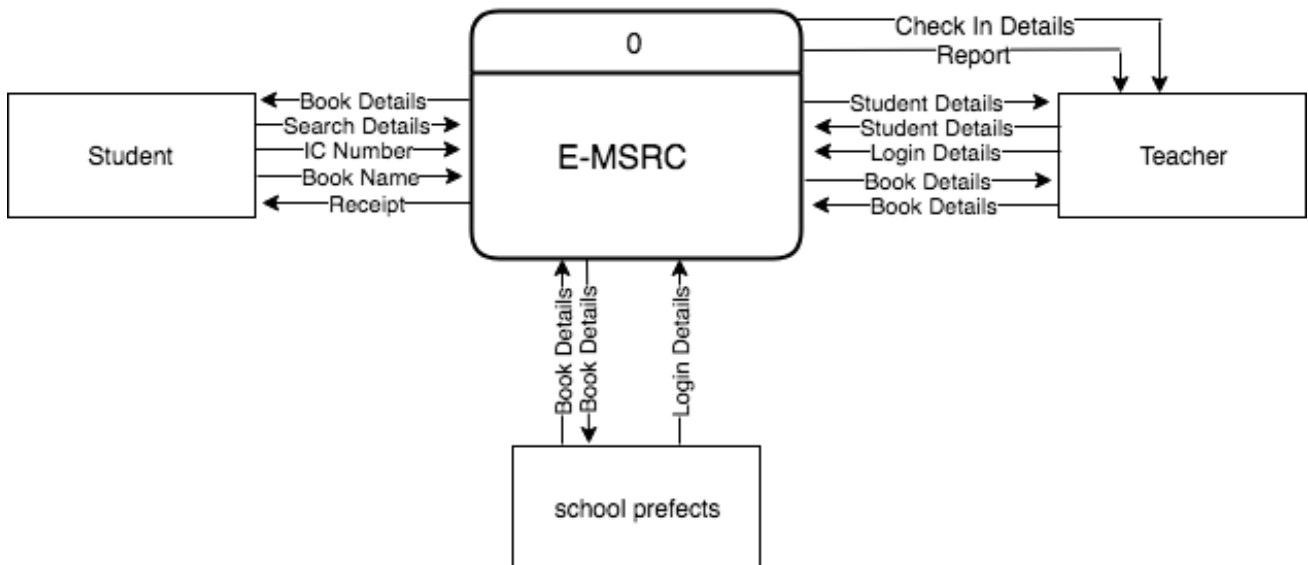


Figure 1 Context Diagram

A distributed E-Management for School Resource Center's system database stores the following information.

#### Student Details

It includes student information like Name, Form and IC Number. Students can also borrow, return books and check in the school resource center.

#### School Prefects Details

School prefects includes login details such as username or e-mail and password. School prefects also able to manage books recorded in the systems in place of teachers.

#### Teacher Details

Teacher able to access most of the data in the systems like student details and book details. Teacher also poses login details like username or e-mail and password. Teacher can generate reports of checked in students.

## 2.2 SYSTEM INTERFACES

## 2.3 PRODUCT FUNCTIONS

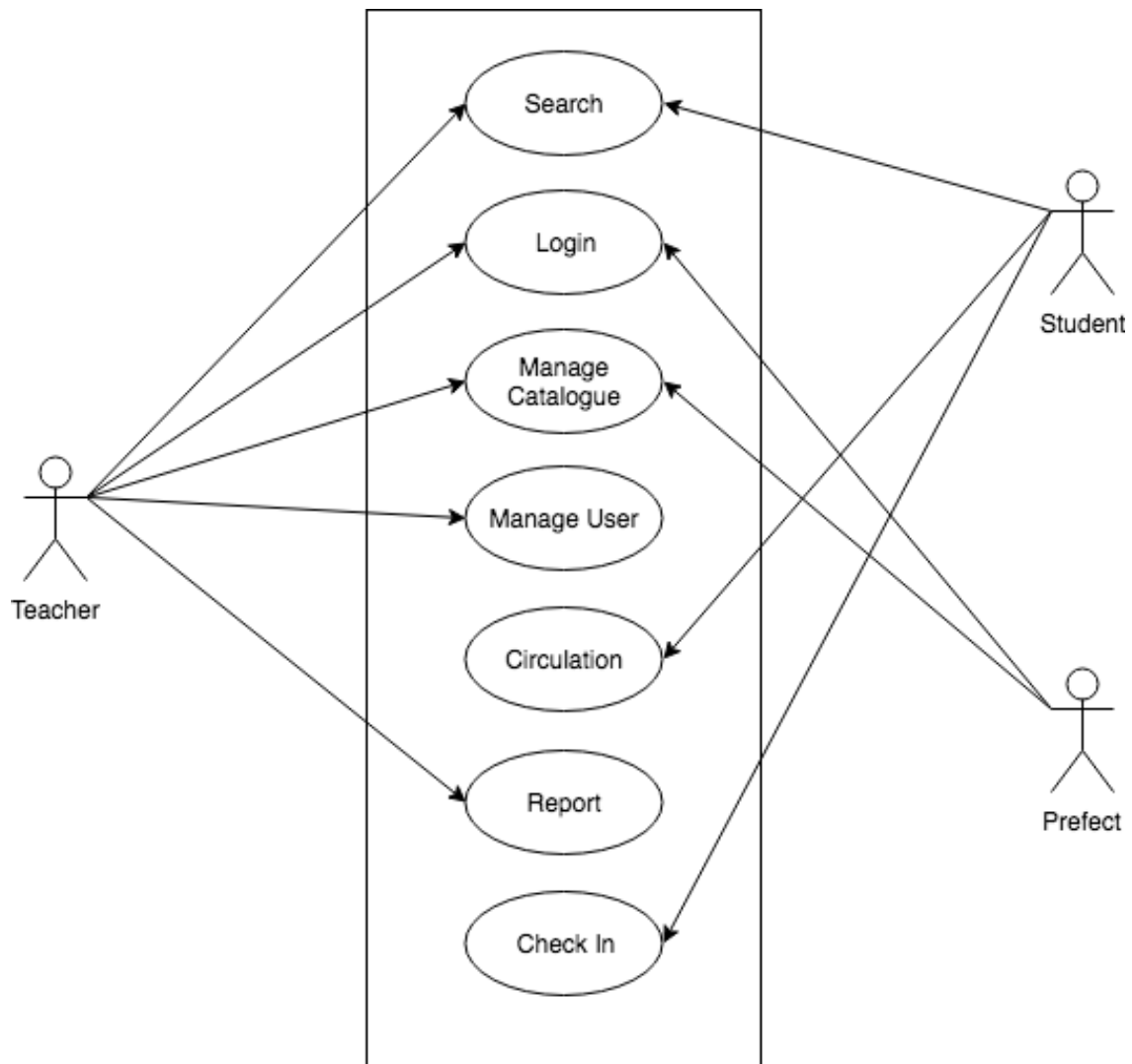


Figure 2 Use Case

Table 1 Use Case Description

| Use Case                | Description  |
|-------------------------|--|
| <b>Search</b>           | This use case allow all user to search information about books in the system.                            |
| <b>Login</b>            | This use case allow Teacher and School Prefects access restricted page                                   |
| <b>Manage Catalogue</b> | This use case allow Teacher and School Prefects to view, add, update and delete books.                   |
| <b>Manage Student</b>   | This use case allow teacher to register new batch of students into the system and manage student's data. |
| <b>Circulation</b>      | This use case allow students to borrow and return books that's available in the system                   |
| <b>Check In</b>         | This use case allow students to check in whenever they enter the school resource center                  |
| <b>Report</b>           | This use case allow Teacher to generate various reports from student's check in data.                    |

## 2.4 USER CHARACTERISTICS

Table 2 User Characteristics

| User                   | Characteristic   |
|------------------------|--|
| <b>Students</b>        | This use case allow all user to search information about books in the system.          |
| <b>School Prefects</b> | This use case allow Teacher and School Prefects access restricted page                 |
| <b>Students</b>        | This use case allow Teacher and School Prefects to view, add, update and delete books. |

## 2.5 CONSTRAINTS

- Privacy statement
  - o Notice/Awareness of the information practices, Choice/Consent to provide
- information, Access/Participation to/in your own data, Integrity/Security of the data collected and Enforcement/Redress through self-regulation.
- Hardware limitation
  - o The system must be use by using computer or other device that approximately
- to 8GB RAM to avoid unable to support. This system must use computer/ laptop to function, full set of device needed to make sure it can be use smoothly.
- The Internet connection is also a constraint for the system. System will fetch the database from internet, so this system must be connect to internet. It is crucial that there is an Internet connection for the system to function. Web portal will be constrained by the capacity of the database. Increase the time it takes to fetch data.
- Language must be in English which is international language and it must allow the user to understand easily.
- The system must be operated 24 hours period and all information must keep updated.
- The database must be backup at the end of the day in case of system failure, the data would not be loss and can be restored.
- Unauthorized user will not be able to access protected pages.

## 2.6 ASSUMPTIONS AND DEPENDENCIES

Below is the list of assumptions and dependencies for E – Management for School Resource Center:

1. The system is a web-based platform. It only can be accessed by using google chrome.
2. This EMSRC shall works only with the presence of internet connection with at least 100kbps download speed.

### 3. SPECIFIC REQUIREMENTS

#### 3.1 SOFTWARE PRODUCT FEATURES

##### 3.1.1 Search

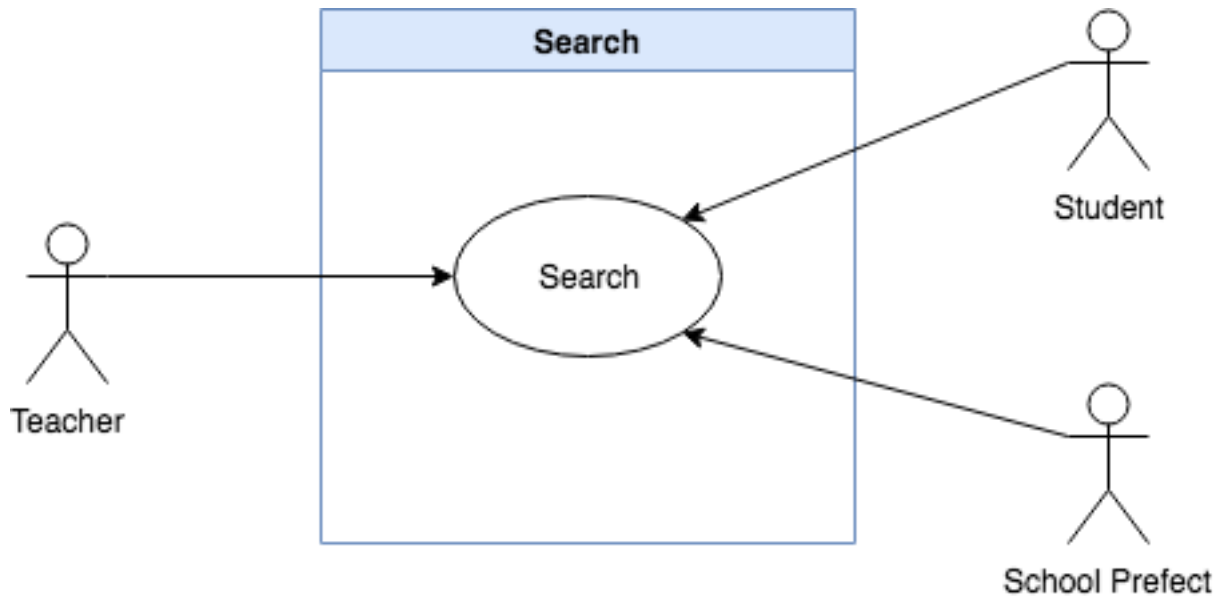


Figure 3 Search Use Case

Table 3 Search Description

|                          |  |
|--------------------------|--|
| <b>Use Case ID</b>       | UC01 – Search Use Case   |
| <b>Brief Description</b> | This use case is for Student. This use case allow Student to search for book details   |
| <b>Actor</b>             | Student  |
| <b>Pre-Conditions</b>    | Search page must be opened   |
| <b>Basic Flow</b>        | <ol style="list-style-type: none"> <li>1. This use case begins when Student opens a search page.</li> <li>2. Student key in book details.</li> <li>3. Then Student click &lt;&lt;Search&gt;&gt; button to search.</li> <li>4. A page with list of books will be displayed</li> </ol> |
| <b>Alternative Flow</b>  |  |
| <b>Exception Flow</b>    |  |
| <b>Post-Conditions</b>   | <ul style="list-style-type: none"> <li>• A list of books that has the search criteria will be displayed</li> </ul>   |

|                         |  |
|-------------------------|--|
| <b>Rules</b>            | <ul style="list-style-type: none"> <li>The system must have a database to manage data</li> </ul> |
| <b>Constraints</b>      | Students can only search book details.   |
| <b>Activity Diagram</b> | Refer Appendix<br>A-1 : Search   |

3.1.2 Login

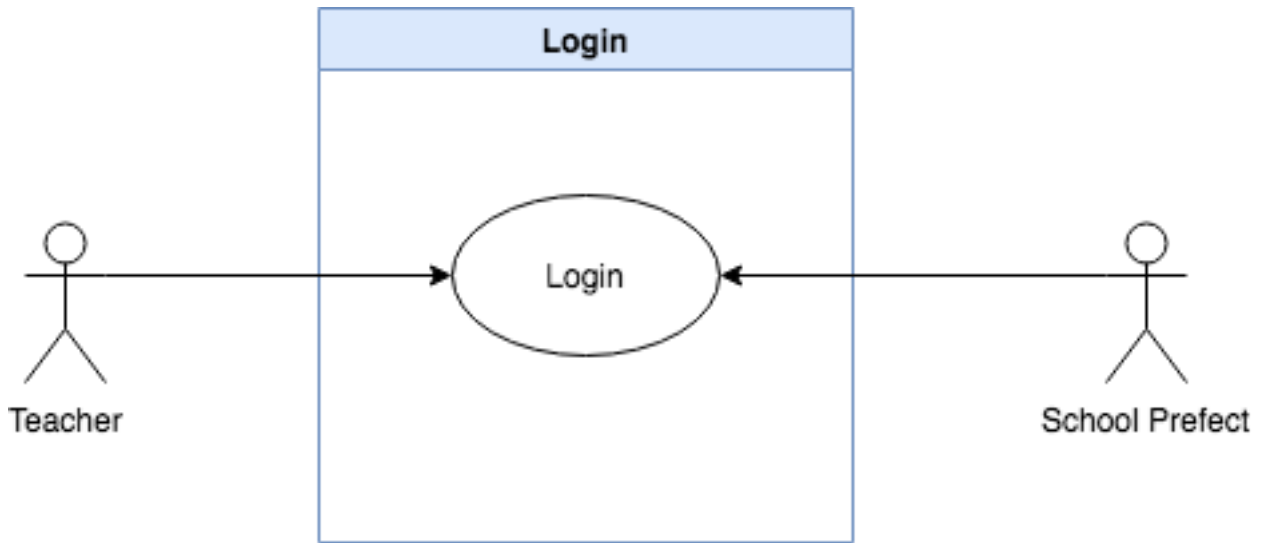


Figure 4 Login

Table 4 Login Description

|                          |   |
|--------------------------|---|
| <b>Use Case ID</b>       | UC02 – Login Use Case   |
| <b>Brief Description</b> | This use case is for Teacher and Prefect. This use case allow Teacher and Prefect to Log in   |
| <b>Actor</b>             | Teacher, Prefect  |
| <b>Pre-Conditions</b>    | User must have account created with the right role.   |
| <b>Basic Flow</b>        | <ol style="list-style-type: none"> <li>5. This use case begins when Teacher or Prefect opens a Log In page.</li> <li>6. Teacher or Prefect key in log in details.</li> <li>7. Then Teacher or Prefect click &lt;&lt;Log In&gt;&gt; button to log in.</li> <li>8. Teacher or Prefect is checked in.</li> </ol> |
| <b>Alternative Flow</b>  |   |
| <b>Exception Flow</b>    | <p>[E1:No User Found]</p> <ol style="list-style-type: none"> <li>1. Page will prompt an error that says no User data</li> </ol>   |

|                         |  |
|-------------------------|--|
|                         | found<br>[E2:Wrong Password]<br>1. The page will prompt that wrong password is used.                 |
| <b>Post-Conditions</b>  | <ul style="list-style-type: none"> <li>Teacher or Prefect is redirected to Logged In area</li> </ul> |
| <b>Rules</b>            | <ul style="list-style-type: none"> <li>The system must have a database to manage data</li> </ul>     |
| <b>Constraints</b>      | Only Teacher or Prefect with registered account can login  |
| <b>Activity Diagram</b> | Refer Appendix<br>A-2 : Login  |

### 3.1.3 Manage Catalogue

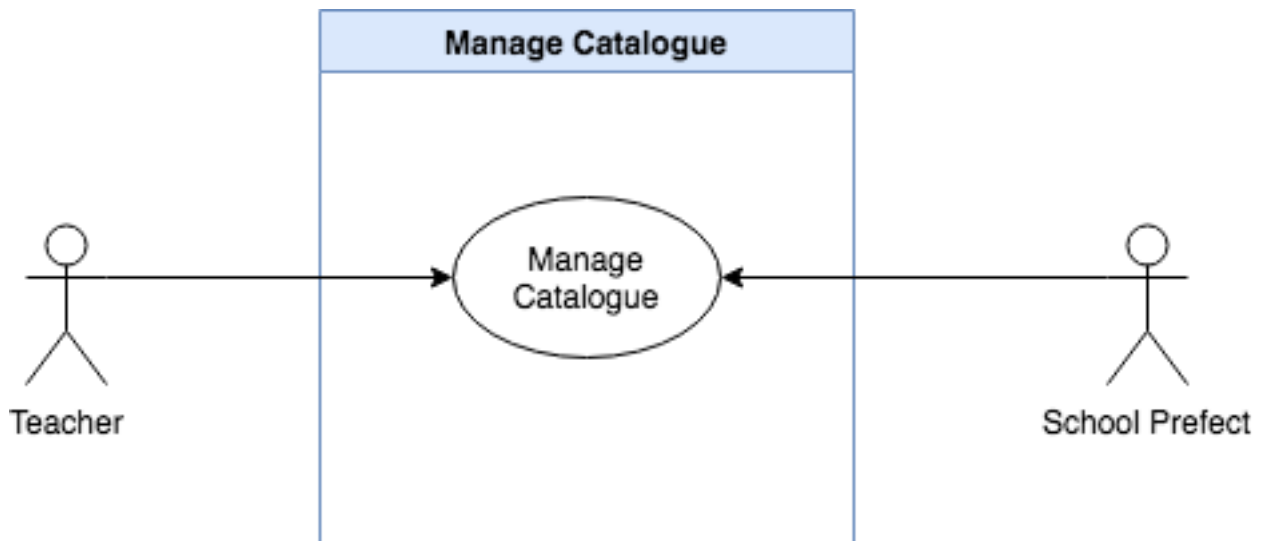


Figure 5 Manage Catalogue

Table 5 Manage Catalogue Description

|                          |  |
|--------------------------|--|
| <b>Use Case ID</b>       | UC03 – Manage Catalogue Use Case   |
| <b>Brief Description</b> | This use case is for Teacher, School Prefect and Student.<br>This use case allow Teacher and School Prefects to view, add, update and delete Book details. This use case only allow Students to view book details. |
| <b>Actor</b>             | Teacher, School Prefect, Student   |
| <b>Pre-Conditions</b>    | Must be logged in as staff or school prefect   |
| <b>Basic Flow</b>        | Login as Teacher or School Prefect   |



|                         |   |
|-------------------------|---|
|                         | <ol style="list-style-type: none"> <li>1. This use case begins when Teacher or School Prefect open Manage Catalogue page</li> <li>2. Teacher or School Prefect will see a list of Book with their details.</li> <li>3. Teacher or School Prefect can choose             <ol style="list-style-type: none"> <li>3.1. A1 : Add</li> <li>3.2. A2 : Update</li> <li>3.3. A3 : Delete</li> </ol> </li> <li>4. Teacher or School Prefect can navigate to other page or log out when done.</li> </ol> <p>Student</p> <ol style="list-style-type: none"> <li>1. This use case begins when Student open Catalogue page</li> <li>2. A list of books will be displayed</li> <li>3. Student click &lt;&lt;View Details&gt;&gt; link to go to another page that display information about the books in more details</li> </ol> |
| <b>Alternative Flow</b> | <p>[A1: Add Book]</p> <ol style="list-style-type: none"> <li>1. Add Book page is displayed with add book form.</li> <li>2. Teacher or School Prefect input Book Details into the Book form</li> <li>3. Teacher or School Prefect click &lt;&lt;Submit&gt;&gt; button to add.</li> </ol> <p>[A2: Update Book]</p> <ol style="list-style-type: none"> <li>1. Update Book page is displayed with the update form filled with existing information</li> <li>2. Teacher or School Prefect input new Book Details into the update Book form</li> <li>3. Teacher or School Prefect click &lt;&lt;Update&gt;&gt; button to update.</li> </ol> <p>[A3: Delete]</p> <ol style="list-style-type: none"> <li>1. Teacher or School Prefect click &lt;&lt;Delete&gt;&gt; button to delete.</li> </ol>                           |
| <b>Exception Flow</b>   | [E1 : Add Package]  |

|                         |  |
|-------------------------|--|
|                         | <ol style="list-style-type: none"> <li>2. Staff did not input package details in all fields</li> <li>3. Return to add package page</li> </ol>  |
| <b>Post-Conditions</b>  | <p>If login as Teacher or School Prefect and choose Add Book details</p> <ul style="list-style-type: none"> <li>• New Book details is added</li> </ul> <p>If login as Teacher or School Prefect and choose Update Package</p> <ul style="list-style-type: none"> <li>• Existing Book details is updated</li> </ul> <p>If login as Teacher or School Prefect and choose Delete Package</p> <ul style="list-style-type: none"> <li>• Book Details is deleted</li> </ul> <p>If Student opens Catalogue page</p> <ul style="list-style-type: none"> <li>• List of Book Details is displayed</li> </ul> |
| <b>Rules</b>            | <ul style="list-style-type: none"> <li>• The system must have a database to manage data</li> <li>• User must have permission to add, edit and delete data</li> </ul>   |
| <b>Constraints</b>      | Student cannot add, edit or delete book details  |
| <b>Activity Diagram</b> | Refer Appendix<br>A-3 : Manage Catalogue   |

#### 3.1.4 Manage User

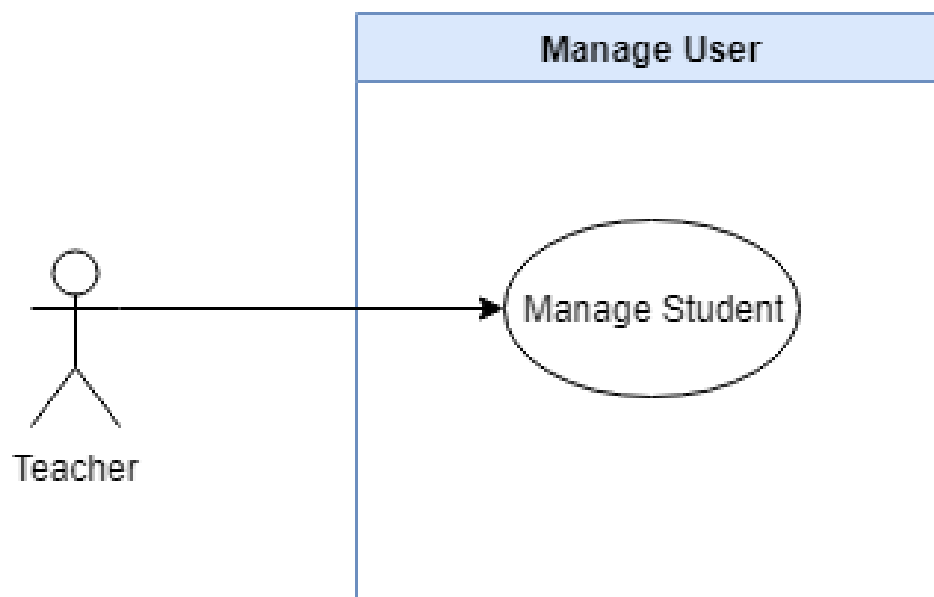


Figure 6 Manage User

Table 6 Manage User Description

|                          |  |
|--------------------------|--|
| <b>Use Case ID</b>       | UC04 – Manage Student Use Case   |
| <b>Brief Description</b> | This use case is for Teacher. This use case allow Teacher to manually register a student or import student data from excel spreadsheet and edit, update or delete the registered student.  |
| <b>Actor</b>             | Teacher  |
| <b>Pre-Conditions</b>    | Must be logged in as staff or school prefect   |
| <b>Basic Flow</b>        | <p>Login as Teacher</p> <ol style="list-style-type: none"> <li>1. This use case begins when Teacher login</li> <li>2. Teacher select Manage Student tab on the sidebar and a dropdown will appear.</li> <li>3. Teacher can choose <ol style="list-style-type: none"> <li>3.1. Add</li> <li>3.2. A1 : View Student List</li> </ol> </li> <li>4. Teacher choose Add</li> <li>5. A page with 2 option will appear.</li> <li>6. Teacher can choose <ol style="list-style-type: none"> <li>6.1. Export File</li> <li>6.2. A2 : Manual Add</li> </ol> </li> <li>7. Teacher Choose Export File</li> <li>8. A pop-up will appear that require Teacher to choose a file to be upload.</li> <li>9. Teacher upload excel sheet with student data</li> <li>10. Teacher click &lt;&lt;Import&gt;&gt; Button to import all data</li> </ol> |
| <b>Alternative Flow</b>  | <p>[A1: View Students]</p> <ol style="list-style-type: none"> <li>1. A page with list of student will appear.</li> <li>2. Teacher can choose <ol style="list-style-type: none"> <li>2.1. A2.1 : Update Student</li> <li>2.2. A2.2 : Delete Student</li> </ol> </li> </ol>  |

|                         |   |
|-------------------------|---|
|                         | <p>[A1.1: Update]</p> <ol style="list-style-type: none"> <li>1. A form with existing data of the student will be shown.</li> <li>2. Teacher update the data in the form</li> <li>3. Teacher click &lt;&lt;Update&gt;&gt; button to update student data.</li> </ol> <p>[A1.2: Delete]</p> <ol style="list-style-type: none"> <li>1. Teacher or School Prefect click &lt;&lt;Delete&gt;&gt; button to delete.</li> </ol> <p>[A2: Manual Add]</p> <ol style="list-style-type: none"> <li>1. An empty form of student details will be shown.</li> <li>2. Teacher fill in the form</li> <li>3. Teacher click &lt;&lt;Submit&gt;&gt; Button to add</li> </ol> |
| <b>Exception Flow</b>   |   |
| <b>Post-Conditions</b>  | <ul style="list-style-type: none"> <li>• New student data is added</li> <li>• Existing student data is updated</li> <li>• Existing student data is deleted</li> </ul>   |
| <b>Rules</b>            | <ul style="list-style-type: none"> <li>• The system must have a database to manage data</li> </ul>  |
| <b>Constraints</b>      | Only Teacher can add, edit, view and delete   |
| <b>Activity Diagram</b> | Refer Appendix<br>A-4 : Manage Student  |

## 3.1.5 Circulation

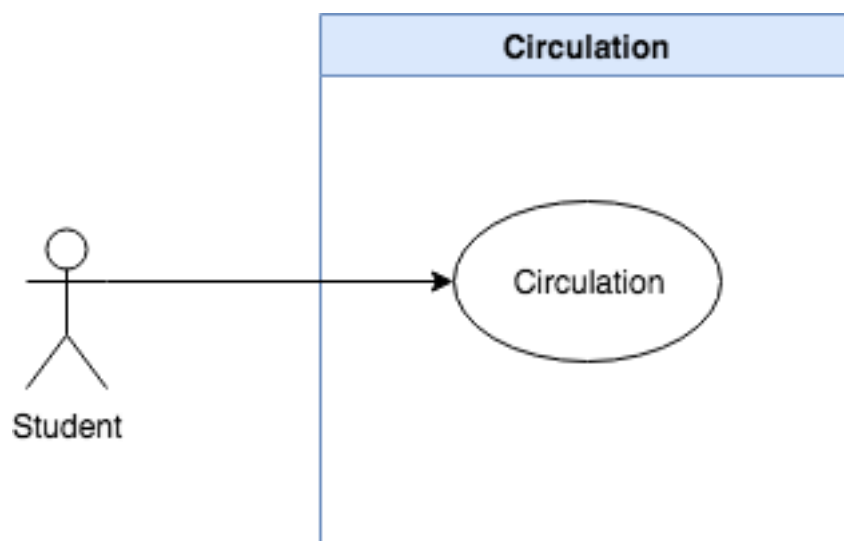


Figure 7 Circulation

Table 7 Circulation Description

|                          |   |
|--------------------------|---|
| <b>Use Case ID</b>       | UC05 – Circulation Use Case   |
| <b>Brief Description</b> | This use case is for Student. This use case allow Student borrow and return book .  |
| <b>Actor</b>             | Student   |
| <b>Pre-Conditions</b>    | Student record must exist in database   |
| <b>Basic Flow</b>        | <ol style="list-style-type: none"> <li>1. Student opens Borrow book page.</li> <li>2. Student enter book details to search and add to borrow list.</li> <li>3. Student click &lt;&lt;Borrow&gt;&gt; click to borrow the book.</li> <li>4. Student then navigate to Return book page.</li> <li>5. Student key in book's detail in to return book list</li> <li>6. Student click &lt;&lt;Return&gt;&gt; button to return book.</li> </ol> |
| <b>Alternative Flow</b>  |   |
| <b>Exception Flow</b>    |   |
| <b>Post-Conditions</b>   | <ul style="list-style-type: none"> <li>• Book is borrowed and returned</li> </ul>   |
| <b>Rules</b>             | <ul style="list-style-type: none"> <li>• The system must have a database to manage data</li> <li>• User must have permission to add, edit and delete data</li> </ul>  |
| <b>Constraints</b>       | Student return book he didn't borrow  |
| <b>Activity Diagram</b>  | Refer Appendix<br>A-5 : Circulation   |

## 3.1.6 Report

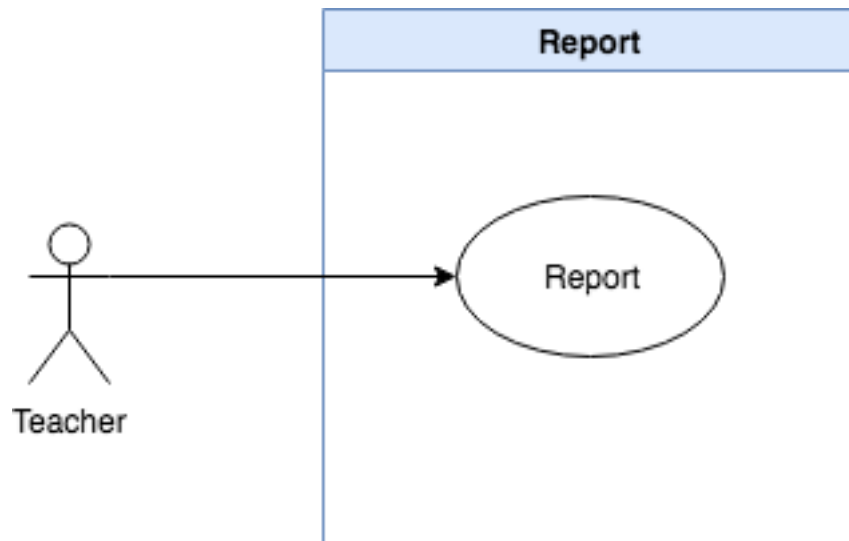


Figure 8 Report

Table 8 Report Description

|                          |   |
|--------------------------|---|
| <b>Use Case ID</b>       | UC06 – Report Use Case  |
| <b>Brief Description</b> | This use case is for Teacher. This use case allow Teacher to view or print check in report  |
| <b>Actor</b>             | Teacher   |
| <b>Pre-Conditions</b>    | Must be logged in as Teacher  |
| <b>Basic Flow</b>        | <p>Login as Teacher</p> <p>11. This use case begins when Teacher opens a Report page.</p> <p>12. Teacher can choose view check in Report by :</p> <p>12.1. A1 : Day</p> <p>12.2. A2 : Month</p> <p>12.3. A3 : Form</p> <p>12.4. A4 : Gender</p> |

|                         |   |
|-------------------------|---|
|                         | <p>13. Teacher can navigate to other page or log out when done.</p> <p>14. Student is checked in.</p>   |
| <b>Alternative Flow</b> | <p>[A1: Day]</p> <ol style="list-style-type: none"> <li>4. Opens a page with day form</li> <li>5. Teacher input date of report</li> <li>6. A list of data of check in on that said date is showed.</li> <li>7. Teacher can choose to print the data by clicking &lt;&lt;Print&gt;&gt; button</li> </ol> <p>[A2: Month]</p> <ol style="list-style-type: none"> <li>1. Opens a page with month form</li> <li>2. Teacher input month of report</li> <li>3. A list of data of check in on that said month is showed.</li> <li>4. Teacher can choose to print the data by clicking &lt;&lt;Print&gt;&gt; button</li> </ol> <p>[A3: Forms]</p> <ol style="list-style-type: none"> <li>1. Opens a page with forms form</li> <li>2. Teacher input student's forms in the form input field</li> <li>3. A list of data of check in on that said forms is showed.</li> <li>4. Teacher can choose to print the data by clicking &lt;&lt;Print&gt;&gt; button</li> </ol> <p>[A4: Gender]</p> <ol style="list-style-type: none"> <li>1. Opens a page with gender form</li> <li>2. Teacher input gender in the form input field</li> <li>3. A list of data of check in of the inputted gender will be showed.</li> <li>4. Teacher can choose to print the data by clicking &lt;&lt;Print&gt;&gt; button</li> </ol> |
| <b>Exception Flow</b>   |   |
| <b>Post-Conditions</b>  | <ul style="list-style-type: none"> <li>• A report of check in will be generated and displayed on the system</li> <li>• A report will be printed</li> </ul>  |

|                         |   |
|-------------------------|---|
| <b>Rules</b>            | <ul style="list-style-type: none"> <li>• The system must have a database to manage data</li> <li>• The system must have existing check in data</li> </ul> |
| <b>Constraints</b>      | Can only produce check in report.   |
| <b>Activity Diagram</b> | Refer Appendix<br>A-6 : Report  |

3.1.7 Check In

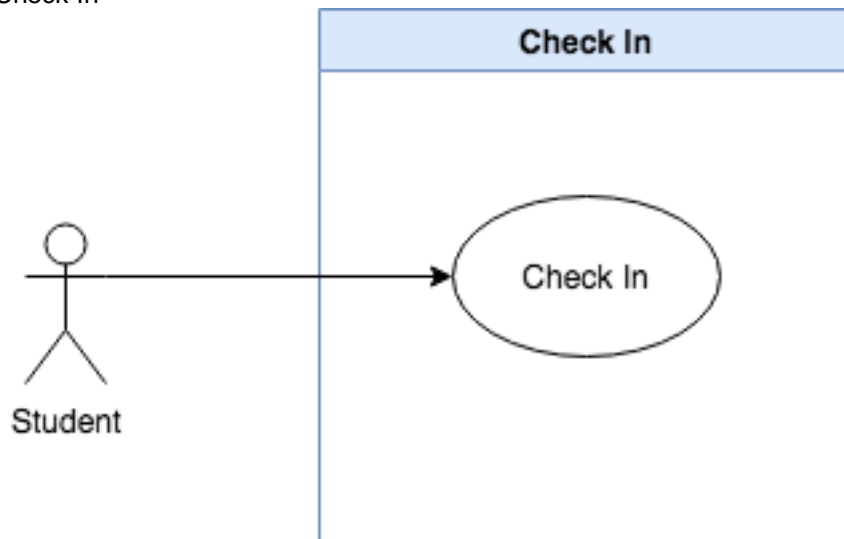


Figure 9 Check In

Table 9 Check In Description

|                          |   |
|--------------------------|---|
| <b>Use Case ID</b>       | UC07 – Check In Use Case  |
| <b>Brief Description</b> | This use case is for Student. This use case allow Student to check in   |
| <b>Actor</b>             | Student   |
| <b>Pre-Conditions</b>    | Check-In page must be opened  |
| <b>Basic Flow</b>        | <p>15. This use case begins when Student opens a Check In page.</p> <p>16. Student key in IC Number.</p> <p>17. Then Student click &lt;&lt;Check In&gt;&gt; button to check in.</p> <p>18. Student is checked in.</p> |
| <b>Alternative Flow</b>  |   |



|                         |  |
|-------------------------|--|
| <b>Exception Flow</b>   | [E1:No Result Found]<br>4. Page will prompt an error that says no student data found   |
| <b>Post-Conditions</b>  | <ul style="list-style-type: none"><li>• A success page will be displayed that tells the student the check In is successful</li></ul> |
| <b>Rules</b>            | <ul style="list-style-type: none"><li>• The system must have a database to manage data</li></ul>                                     |
| <b>Constraints</b>      | Students can only search book details.   |
| <b>Activity Diagram</b> | Refer Appendix<br>A-7 : Sequence Diagram   |

## 3.2 EXTERNAL INTERFACE REQUIREMENTS

### 3.2.1 USER INTERFACES

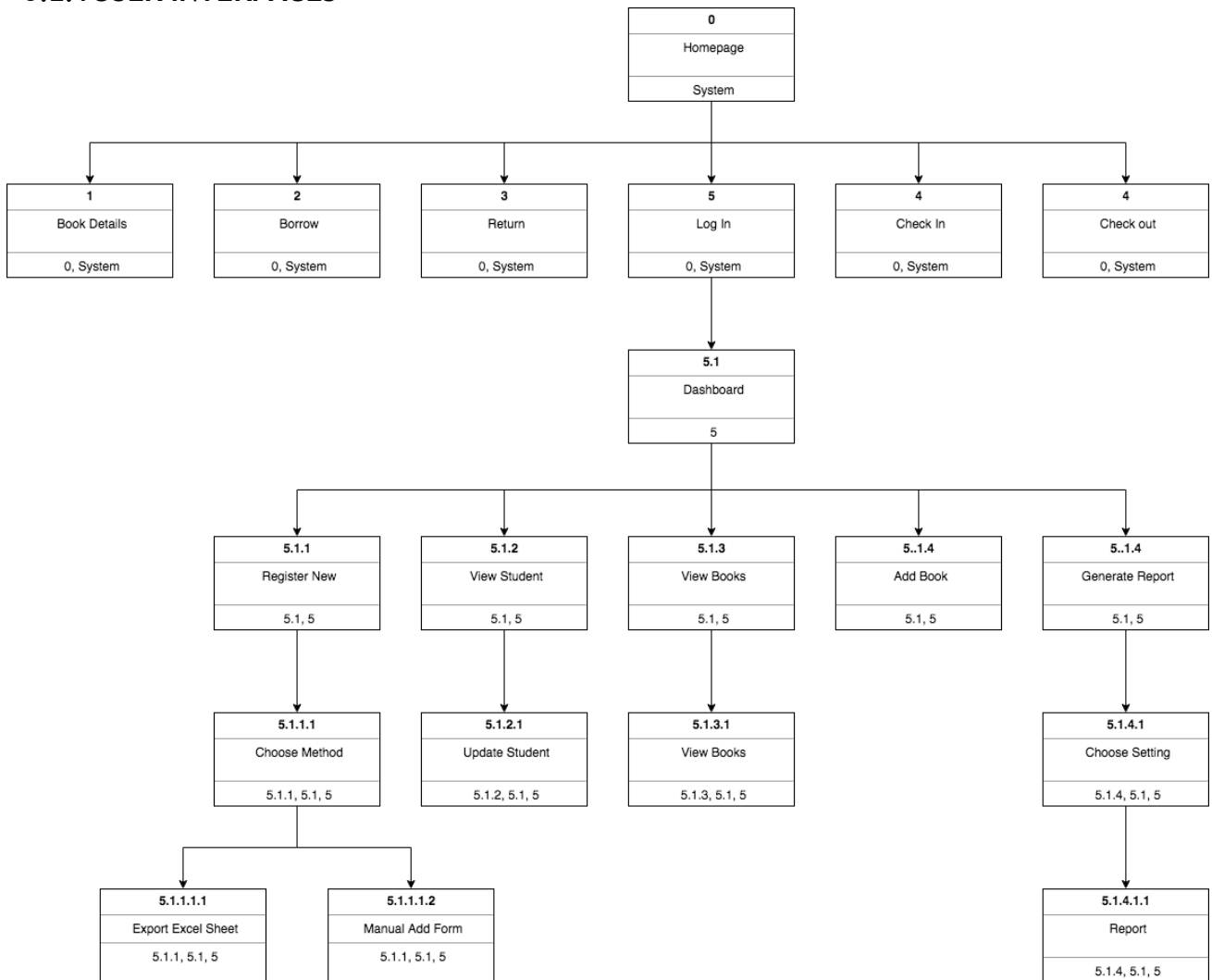


Figure 10 Dialogue Diagram

| User Interface Name   | Description  | User Interface Layout |
|-----------------------|--|-----------------------|
| Homepage              | A list of book sort by name is displayed. A search box is placed at the top of the list for user to search through the list. |                       |
| Book Details          | This page displays all details about the book.   |                       |
| Borrow                | This page is for students to borrow books  |                       |
| Return                | This page is for students to return borrowed book  |                       |
| Log In                | For Teachers and School Prefects to logged into restricted pages.  |                       |
| Check In              | For students to check in whenever they came into the school resource center.   |                       |
| Check Out             |  |                       |
| Dashboard             | The first page shown to logged in users.   |                       |
| Register New Students | A page to choose a method to register new students. Either by manually registering them one by one or with a excel file.     |                       |
| View Student          | View a list of registered student.   |                       |
| Update Student        | Update existing details of   |                       |

|                 |  |  |
|-----------------|--|--|
|                 | registered student.  |  |
| View books      | Teacher or School<br>Prefect can view list of<br>Books from database to<br>be updated or delete. |  |
| Update Book     | Teacher or School<br>Prefect can update<br>existing book's<br>information                        |  |
| Add Book        | Teacher or School<br>Prefect can add new<br>book to the database                                 |  |
| Generate Report | Teacher can choose<br>settings for the report.   |  |
| Report          | The generated report will<br>be displayed here.  |  |

### 3.2.2 Hardware Interface

- Not Applicable

### 3.2.3 Manage Payment

- Not Applicable

### 3.3 REQUIREMENTS TRACEABILITY

| Use Case ID | Requirement ID | Requirement Details ID   | Requirement Sources |
|-------------|----------------|--|---------------------|
| UC01        | RF 1           | Students can search any books.   | Stakeholder         |
| UC02        | RF 2           | Teacher or School Prefect can login into restricted area.                                      | Stakeholder         |
| UC03        | RF 3           | Teacher or School Prefect can register new books into the system.                              | Stakeholder         |
| UC03        | RF 4           | Teacher or School Prefect can edit Existing book details.                                      | Stakeholder         |
| UC03        | RF 5           | All level of users can View List of Books.   | Stakeholder         |
| UC03        | RF 6           | Teacher or School Prefect can delete Existing Books.   | Stakeholder         |
| UC04        | RF 7           | Teacher can manually register students or export student data into the system with excel file. | Stakeholder         |
| UC04        | RF 8           | Teacher can update existing student data.  | Stakeholder         |
| UC04        | RF 9           | Teacher can delete existing student data.  | Stakeholder         |

|      |       |   |             |
|------|-------|---|-------------|
| UC05 | RF 10 | Students can borrow books that is already registered in the system.                 | Stakeholder |
| UC05 | RF 11 | Students can return the books they borrow.  | Stakeholder |
| UC06 | RF 12 | Teachers can generate Reports of student's Check In data baes on multiple settings. | Stakeholder |
| UC07 | RF 13 | Students need to check in when they enter school resource center.                   | Stakeholder |

#### 4. ACRONYMS AND ABBREVIATION

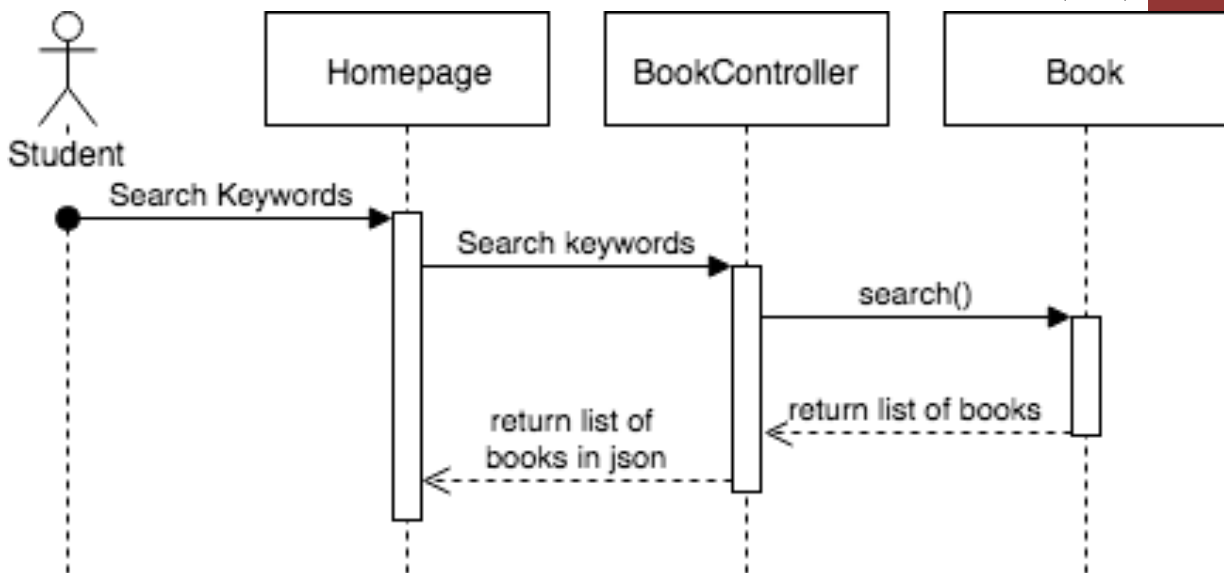
---

| Acronyms and Abbreviation | Description                               |
|---------------------------|---|
| EMSRC                     | E – Management for School Resource Center |
| UC                        | Use Case                                  |

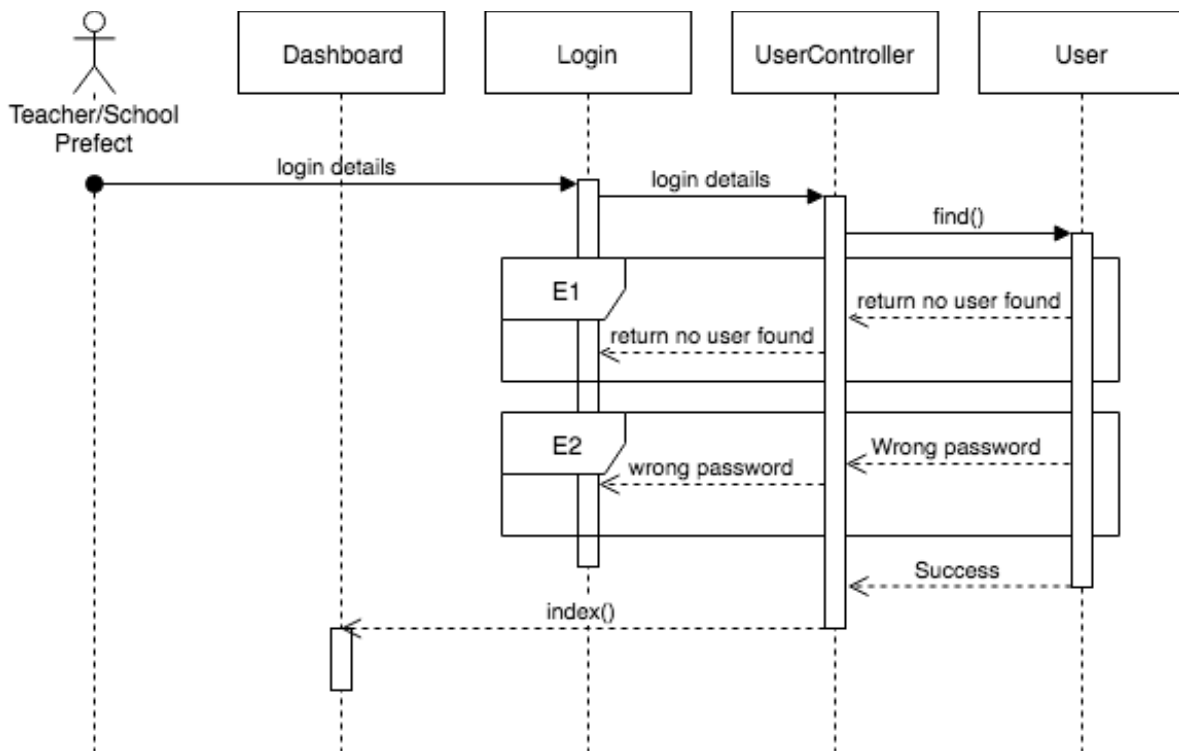
## APPENDIX A

### Sequence Diagram

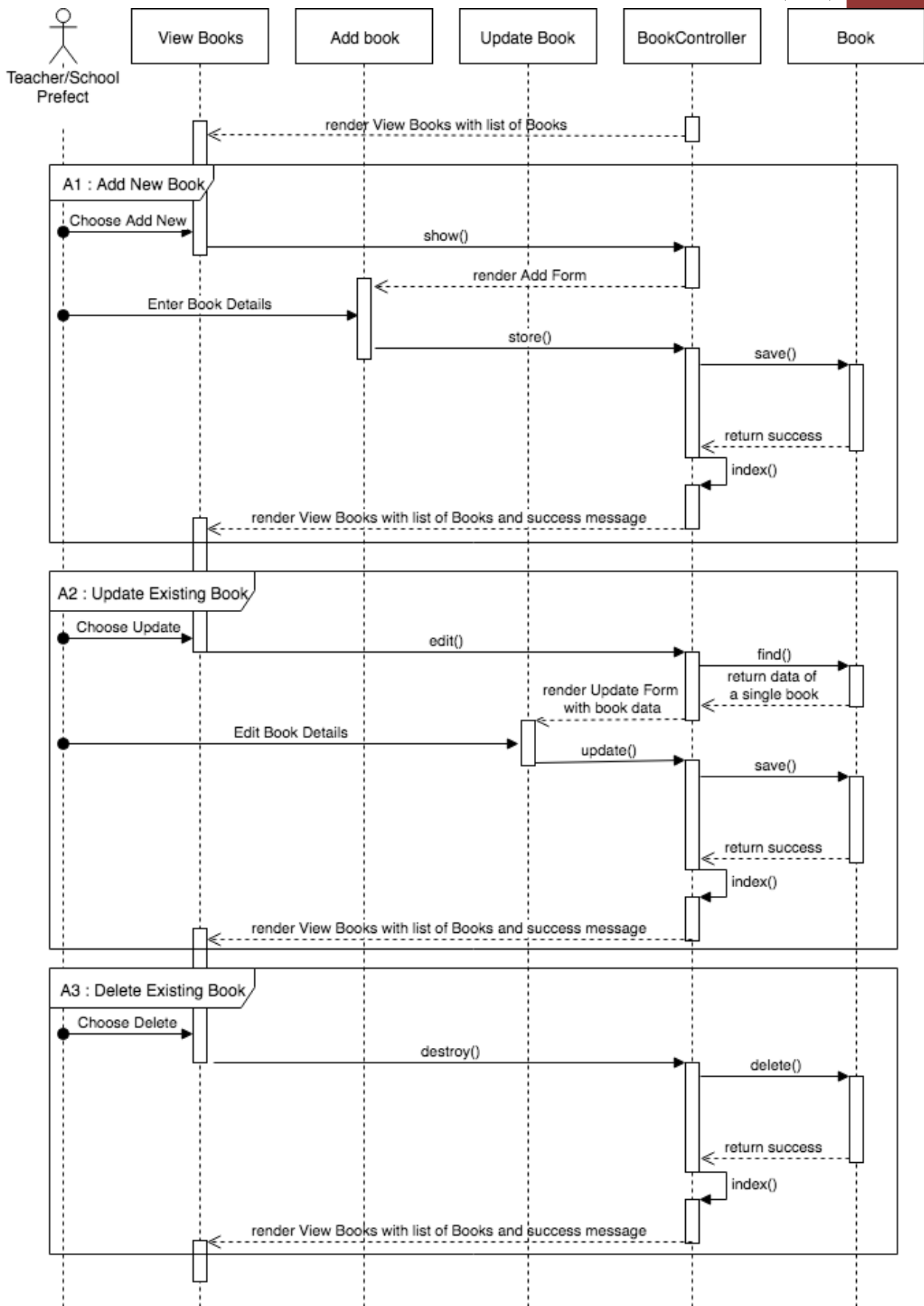


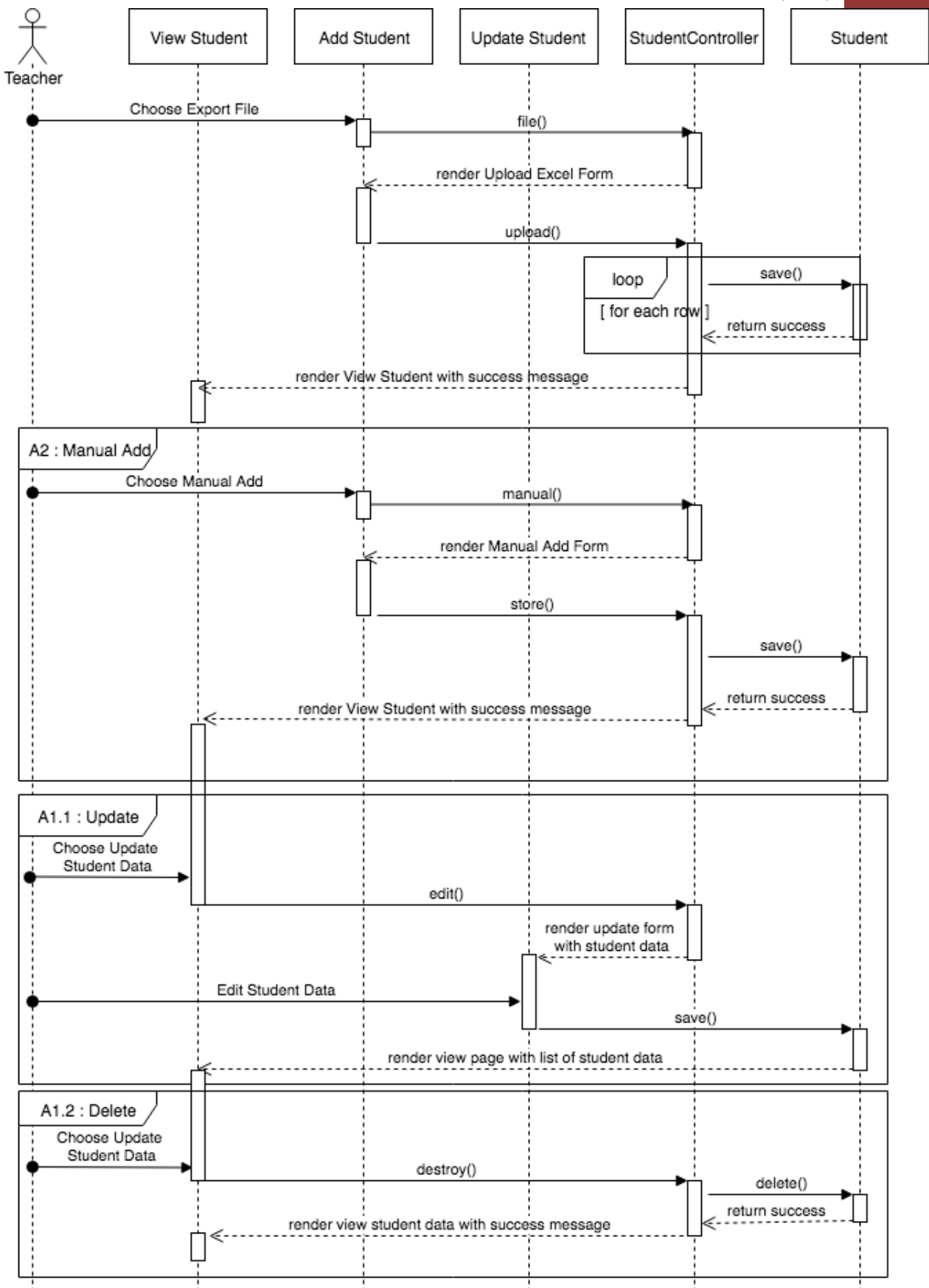


APPENDIX A - 1 Homepage/Search

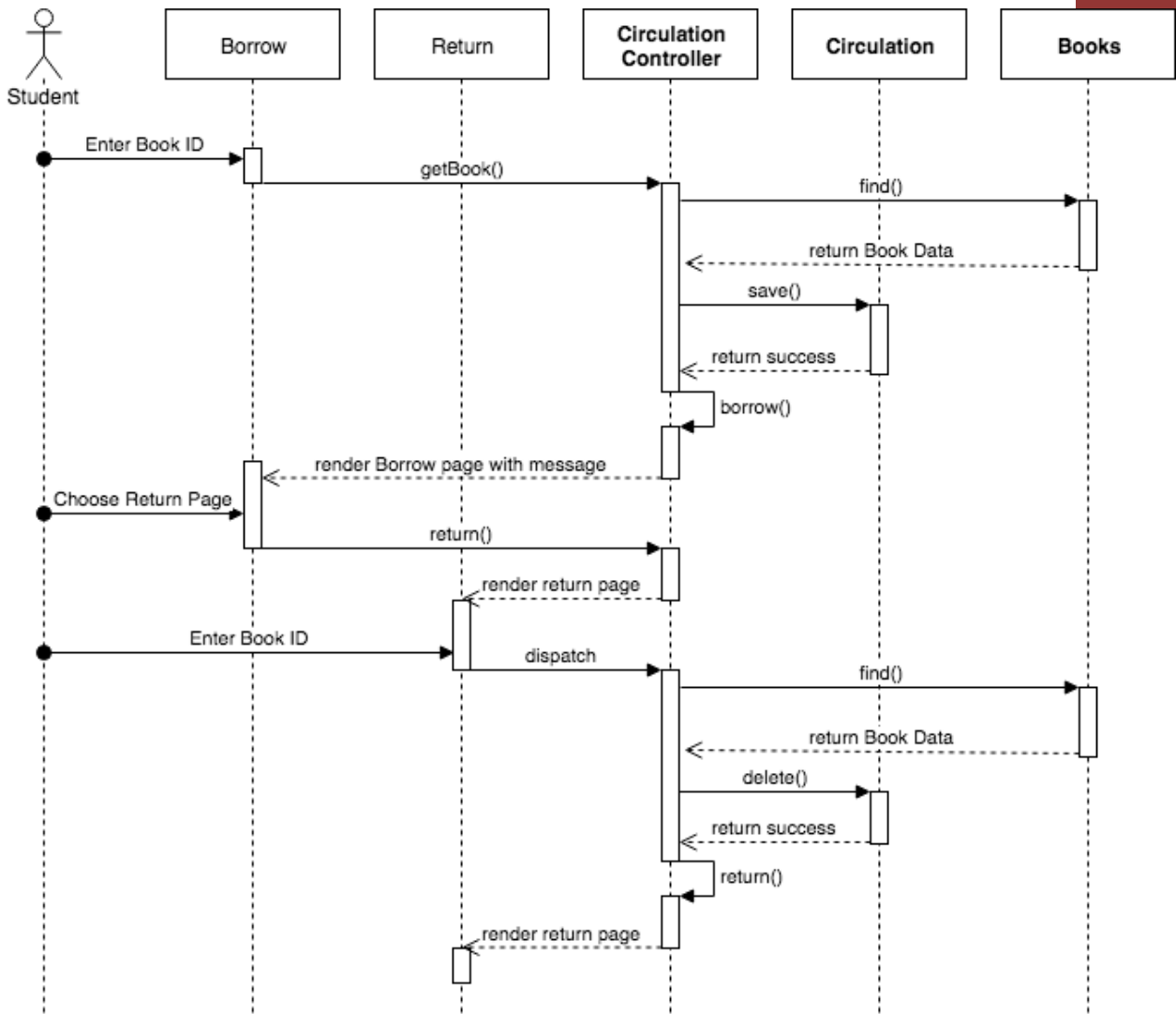


APPENDIX A - 2 Login

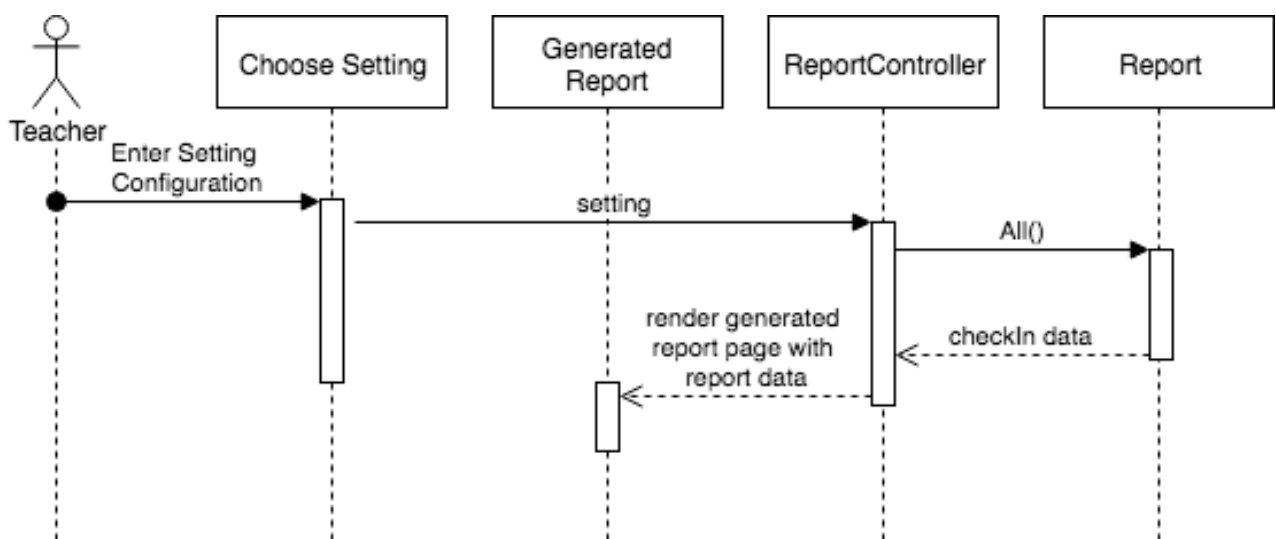




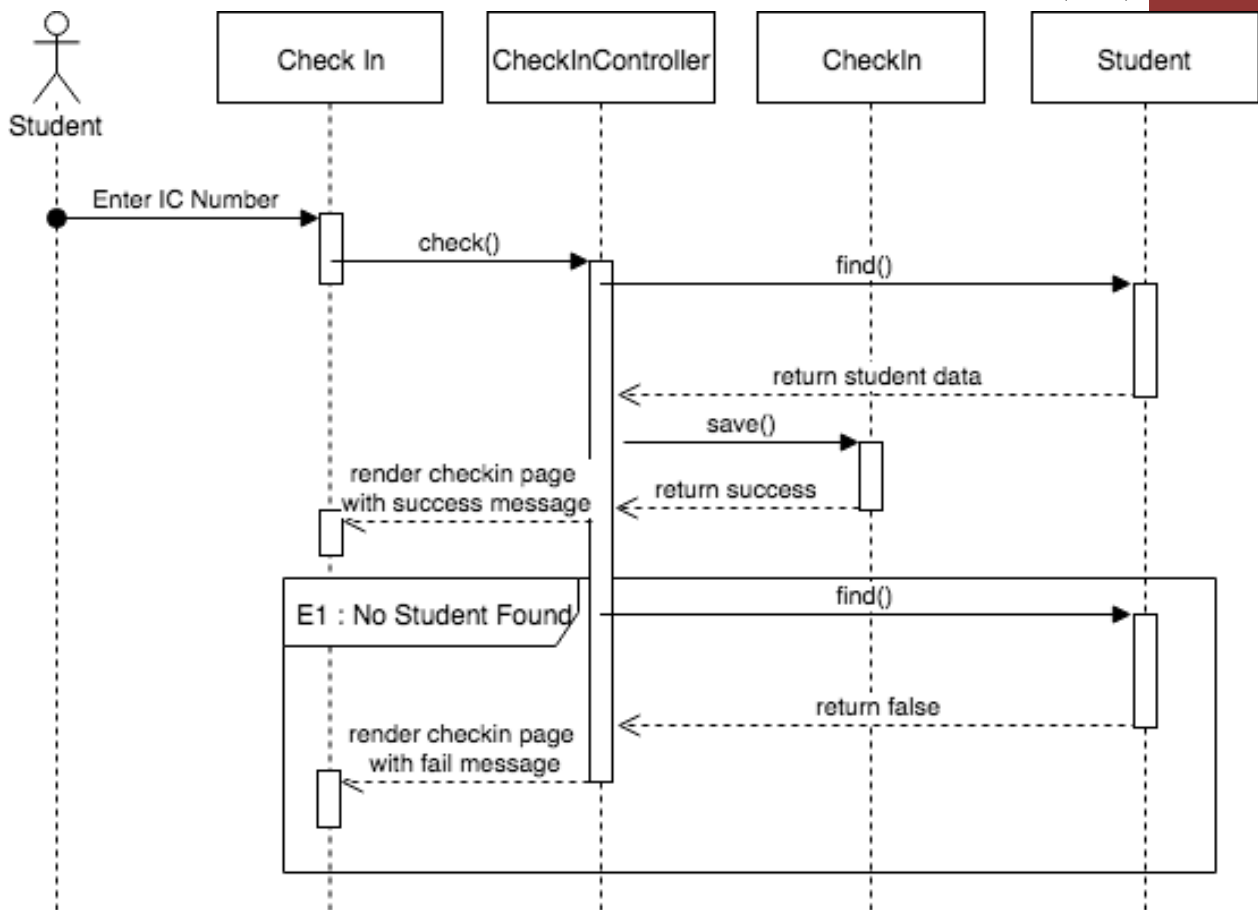
APPENDIX A - 4 Manage Student



APPENDIX A - 5 Circulation



APPENDIX A - 6 Report

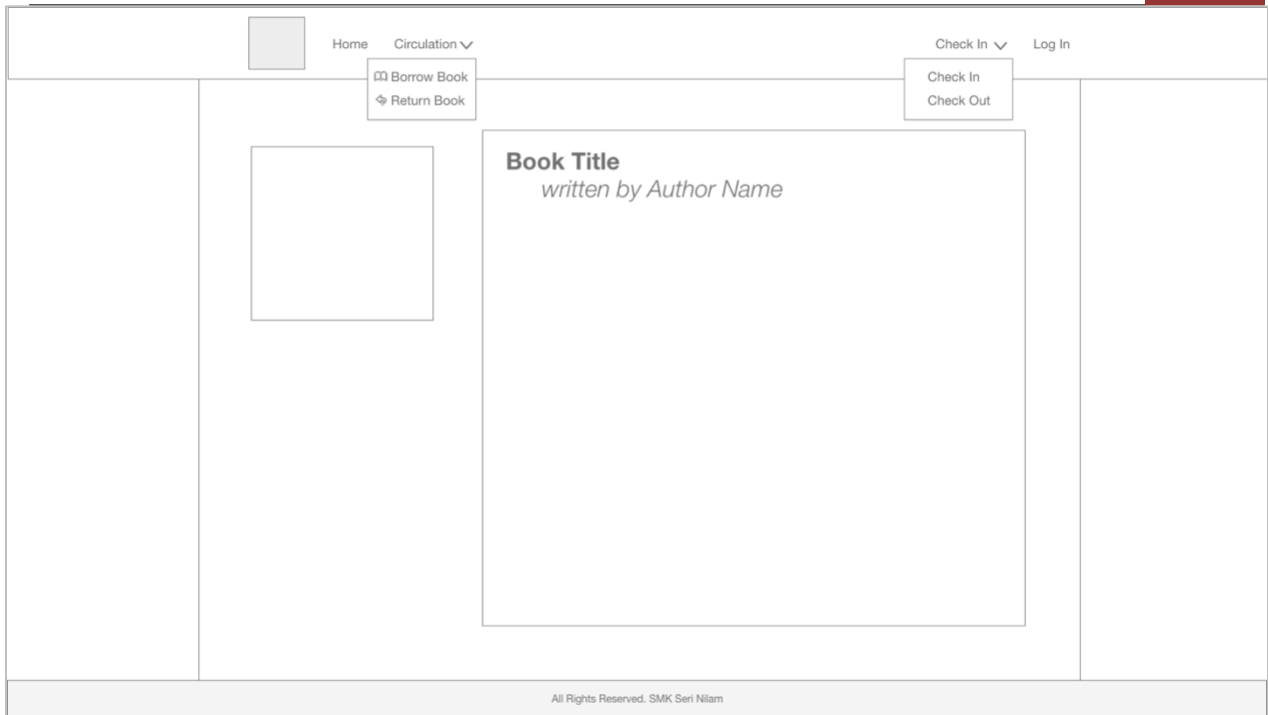


APPENDIX A - 7 Check In

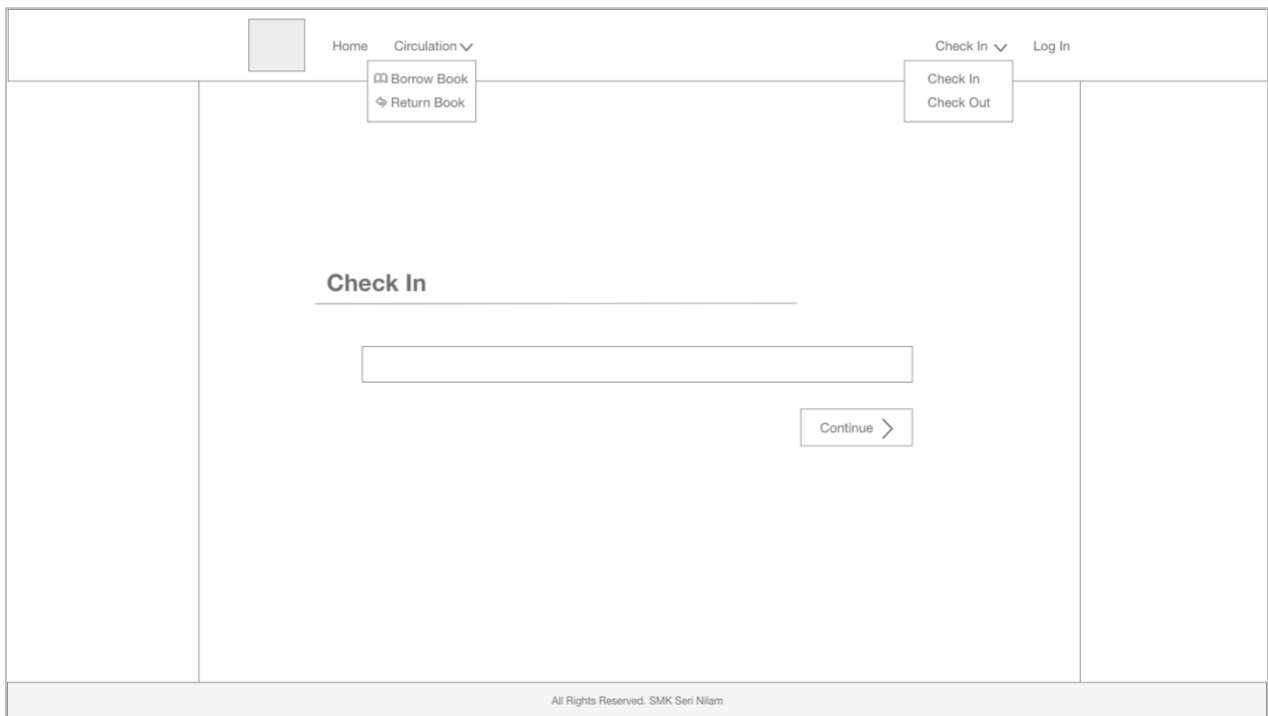
## APPENDIX B

### User Interface Design

|  |   |            |           |
|--|---|------------|-----------|
| Home   | Circulation ▾   | Check In ▾ | Log In    |
| Borrow Book  | Return Book   | Check In   | Check Out |
| <input type="text"/>                               |   |            |           |
| <b>Book List</b>                                   |   |            |           |
|  | <b>Book Title</b><br>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex<br>Author : Name of The Author<br>Year Published : Year <a href="#">View Details</a> |            |           |
|  | <b>Book Title</b><br>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex<br>Author : Name of The Author<br>Year Published : Year <a href="#">View Details</a> |            |           |
|  | <b>Book Title</b><br>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex<br>Author : Name of The Author<br>Year Published : Year <a href="#">View Details</a> |            |           |
|  | <b>Book Title</b><br>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex<br>Author : Name of The Author<br>Year Published : Year <a href="#">View Details</a> |            |           |
|  | <b>Book Title</b><br>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex<br>Author : Name of The Author<br>Year Published : Year <a href="#">View Details</a> |            |           |
|  | <b>Book Title</b><br>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex<br>Author : Name of The Author<br>Year Published : Year <a href="#">View Details</a> |            |           |
| < 1 2 3 >  |   |            |           |
| <small>All Rights Reserved. SMK Seri Nilam</small> |   |            |           |



APPENDIX B - 2 Book Details



APPENDIX B - 3 Check In



Home    Circulation ▾    Check In ▾    Log In

Borrow Book  
Return Book

Check In  
Check Out

### Borrow Book(s)

I/C Number

Search >

|  |  |  |  |
|--|--|--|--|
|  |  |  |  |
|  |  |  |  |

All Rights Reserved. SMK Seri Nilam

APPENDIX B - 4 Borrow Book

Home    Circulation ▾    Check In ▾    Log In

Borrow Book  
Return Book

Check In  
Check Out

### Log In

E-Mail Address

Password

Remember Me

Log In    [Forgot Password ?](#)

All Rights Reserved. SMK Seri Nilam

APPENDIX B - 5 Return Book



*APPENDIX B - 6 Dashboard*

**APPENDIX C**  
**SOFTWARE DEVELOPMENT DOCUMENT(SDD)**

2018

# SOFTWARE DESIGN DOCUMENT (SDD)

E – MANAGEMENT SYSTEM FOR SCHOOL  
RESOURCE CENTER

MUHAMMAD ASHRAF BIN KAMARUDIN CB16148

To be submitted to the  
Faculty of Computer System and Software Engineering  
Bachelor of Computer Science (Software Engineering)



## DOCUMENT APPROVAL

|   | Name                                     | Date              |
|---|--|-------------------|
| <p>Authenticated by:</p> <p><b>Ashraf</b></p> <hr/> <p>Project Leader</p> | <p>MUHAMMAD ASHRAF BIN<br/>KAMARUDIN</p> | <p>08/03/2018</p> |
| <p>Approved by:</p> <hr/> <p>Project Manager</p>                          |  | <p>08/03/2018</p> |

Software : draw.io diagrams

Archiving Place : GoogleDrive  
(<https://drive.google.com/open?id=1hKmRfYOcuySXjE3YQdgH-Mk8ecPMmfcw>)

Copies Available : 1 copy

## TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>DOCUMENT APPROVAL .....</b>            | <b>2</b>  |
| <b>TABLE OF CONTENTS .....</b>            | <b>3</b>  |
| <b>1. INTRODUCTION .....</b>              | <b>4</b>  |
| 1.1 PURPOSE.....                          | 4         |
| 1.2 IDENTIFICATION .....                  | 4         |
| <b>2. REFERENCED DOCUMENT.....</b>        | <b>5</b>  |
| <b>3. ARCHITECTURE DESCRIPTIONS .....</b> | <b>6</b>  |
| 3.1 Package Module.....                   | 6         |
| 3.1.1 Search .....                        | 6         |
| 3.1.2 Login .....                         | 6         |
| 3.1.3 Manage Catalogue .....              | 7         |
| 3.1.4 Manage Student .....                | 8         |
| 3.1.5 Circulation .....                   | 9         |
| 3.1.6 Report .....                        | 9         |
| 3.1.7 Check In.....                       | 10        |
| 3.2 Data Dictionary .....                 | 11        |
| <b>4. DETAIL DESIGN .....</b>             | <b>13</b> |
| 4.1 Login Controller.....                 | 13        |
| 4.2 Homepage Controller .....             | 13        |
| 4.3 Dashboard Controller .....            | 13        |
| 4.4 Check In Controller .....             | 15        |
| 4.5 Circulation Controller .....          | 15        |
| 4.6 Report Controller .....               | 16        |
| 4.7 Book .....                            | 16        |
| 4.8 Student .....                         | 17        |
| 4.9 Check In .....                        | 17        |
| 4.10 Circulation .....                    | 17        |
| 4.11 User.....                            | 18        |

---

## 1. INTRODUCTION

---

### 1.1 PURPOSE

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SDD document is to give a detailed description of the requirements for the “Driving School Management System”. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the clients for their approval and a reference for developing the system for the development team.

### 1.2 IDENTIFICATION

SDD-EMSRC-2018-V1

EMSRC : E – MANAGEMENT FOR SCHOOL RESOURCE CENTER

ESTABLISH : 2018

V1 : VERSION 1

DOCUMENT TYPE : SDD

---

## 2. REFERENCED DOCUMENT

---

[1] Tengku Diyana Binti Tengku Ibrahim. (2005 March). *DRIVING SCHOOL STUDENT MANAGEMENT SYSTEM.*

[http://umpir.ump.edu.my/3720/1/TENGGU\\_DIYANA\\_BINTI\\_TENGGU\\_IBRAHIM.PDF](http://umpir.ump.edu.my/3720/1/TENGGU_DIYANA_BINTI_TENGGU_IBRAHIM.PDF)

[2] LAIMOMCIN, (2017). *University Malaysia Pahang Software Requirement Workshop SRS Document.*

[3] Pohl, K. (1 st ed.). (2010). *Requirement Engineering: Fundamental, Principles and Techniques.* Germany: Springer.

<http://www.requirements-book.com/>

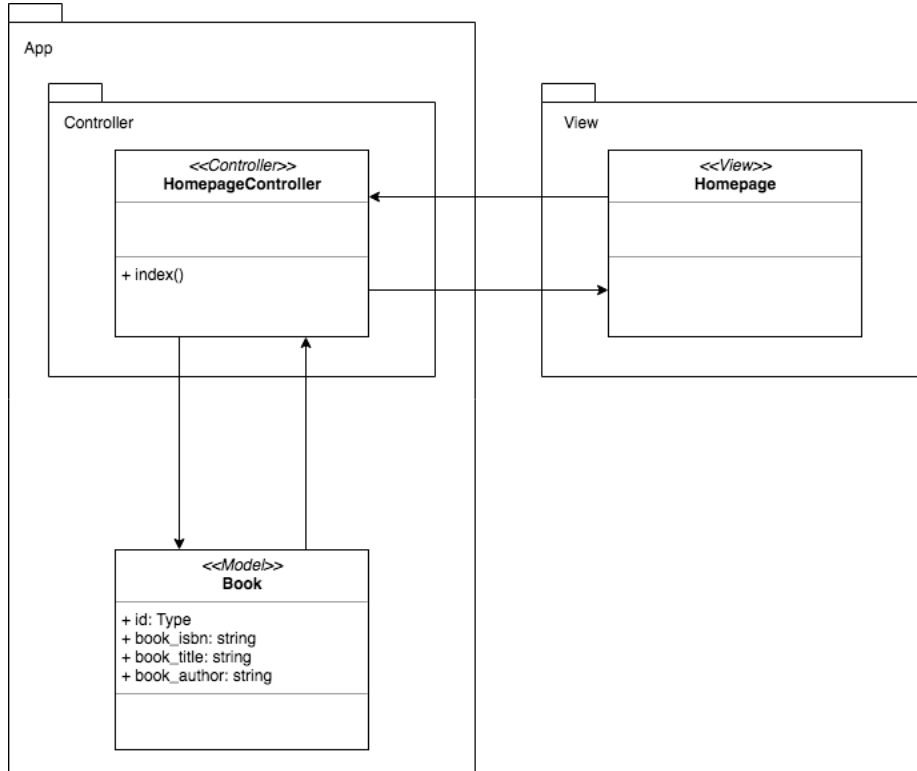
[4] Software Planning & Requirement Workshop (2017). *SRS Template: Software Requirement Specification (SRS).*



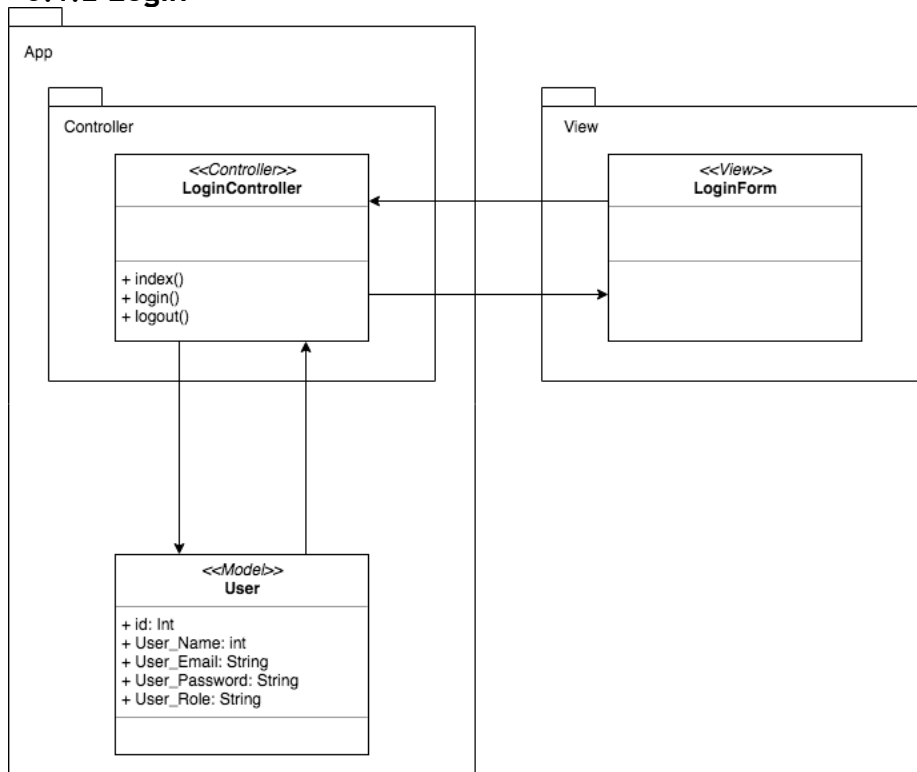
### 3. ARCHITECTURE DESCRIPTIONS

#### 3.1 Package Module

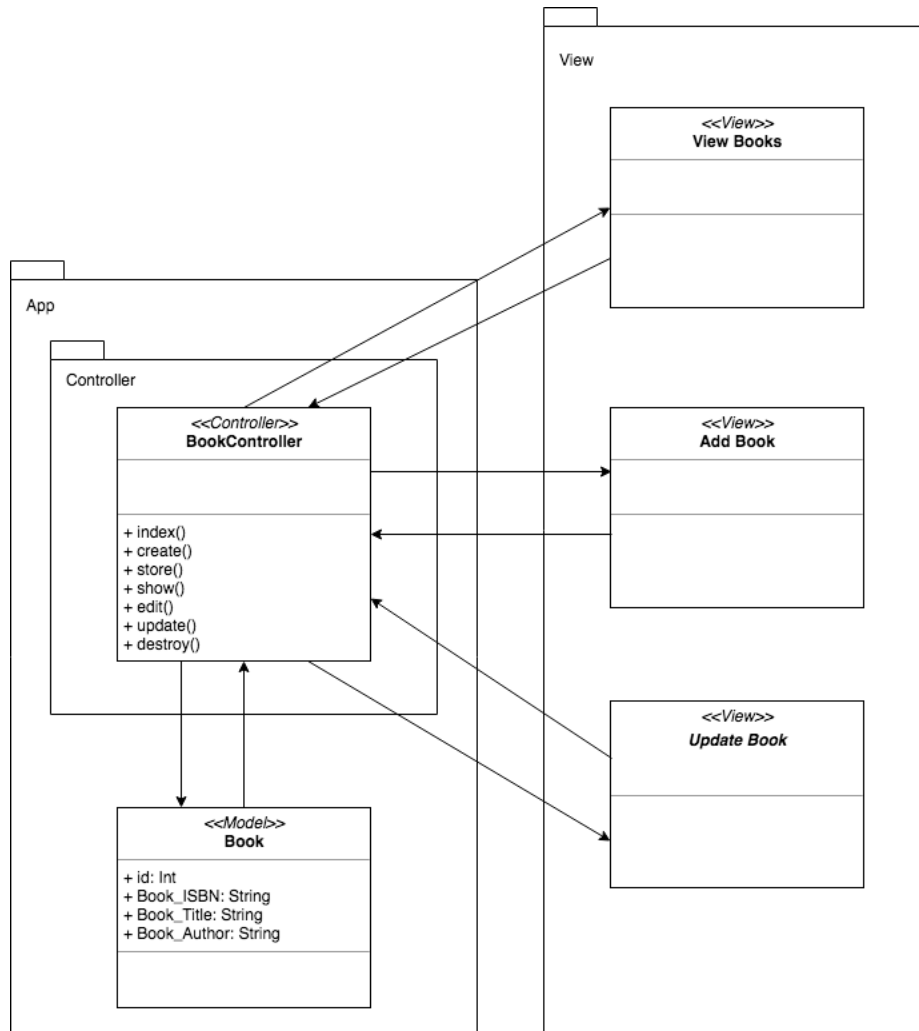
##### 3.1.1 Search



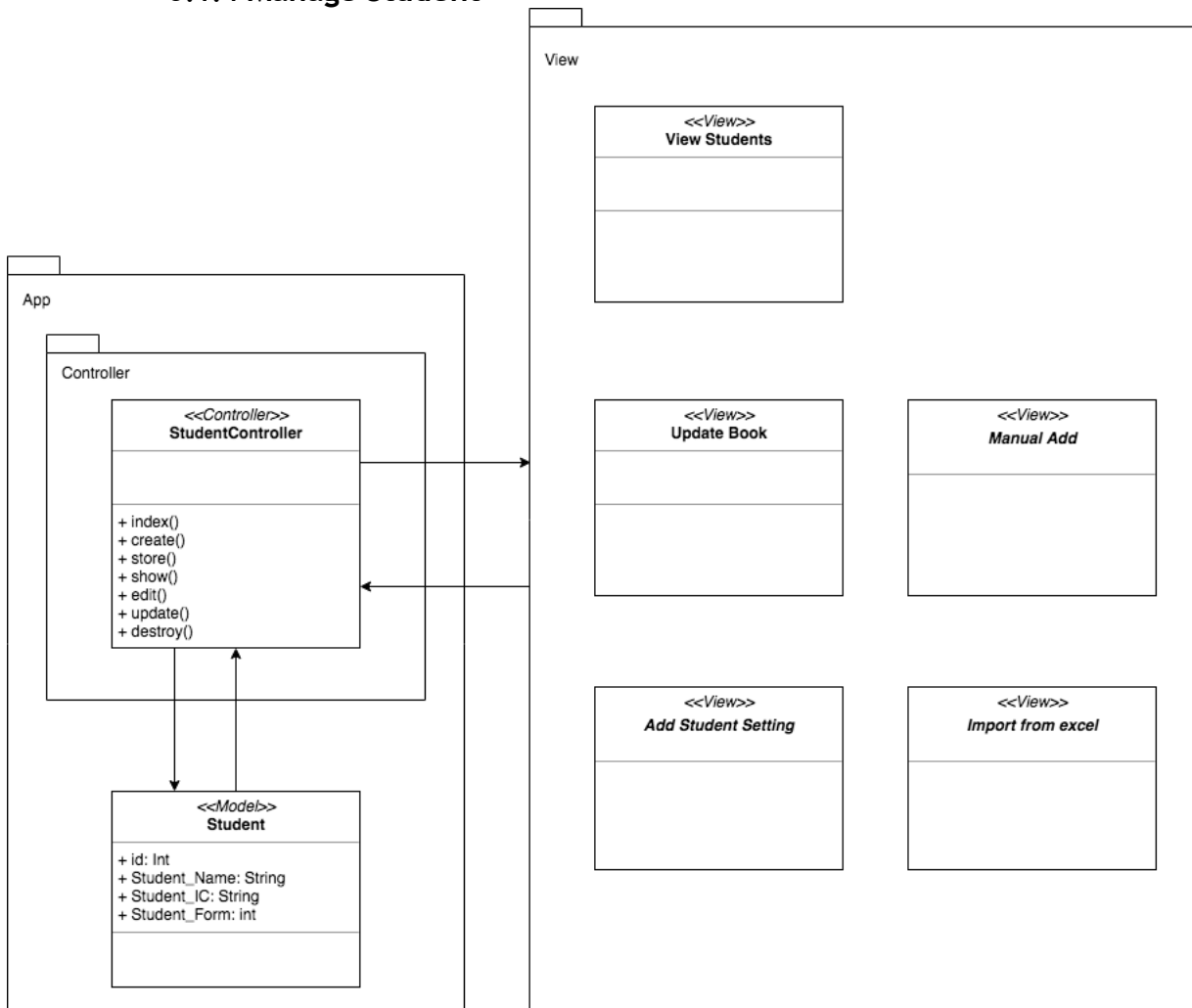
##### 3.1.2 Login



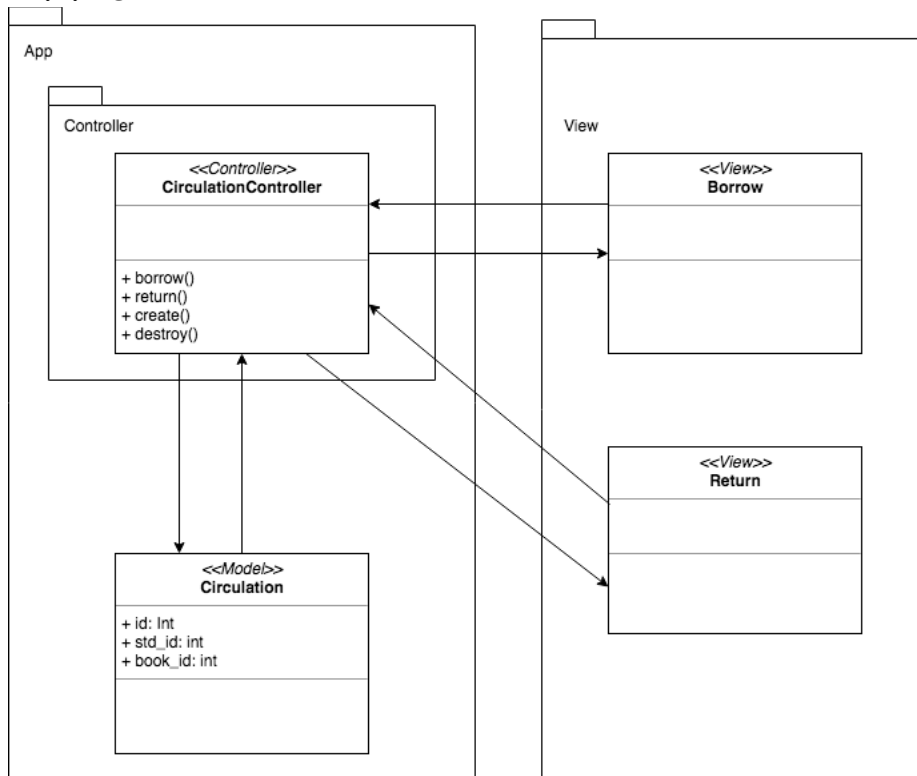
### 3.1.3 Manage Catalogue



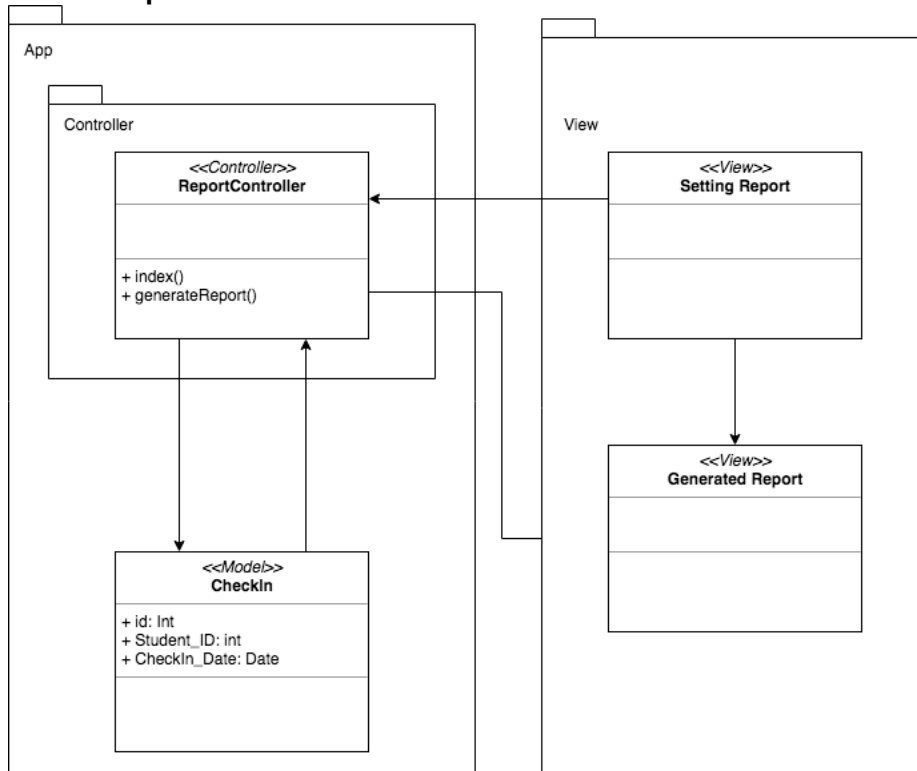
## 3.1.4 Manage Student



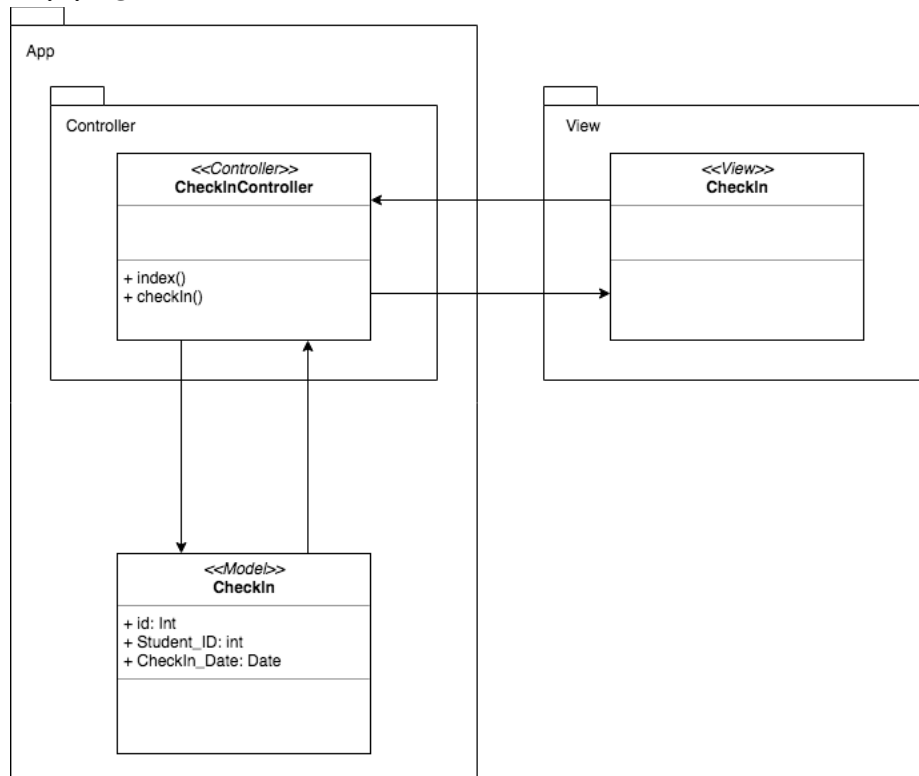
## 3.1.5 Circulation



## 3.1.6 Report



## 3.1.7 Check In



### 3.2 Data Dictionary

Table User

| Field    | Description   | Data Type    | Constraint |
|----------|---------------|--------------|------------|
| ID       | User ID       | int(11)      | PK         |
| Username | User Name     | Varchar(255) |            |
| email    | User Email    | Varchar(255) |            |
| Password | User Password | Varchar(255) |            |
| Roles    | User Roles    | Varchar(20)  |            |

Table Student

| Field    | Description  | Data Type    | Constraint |
|----------|--------------|--------------|------------|
| ID       | Student ID   | Int(11)      | PK         |
| Std_Name | Student Name | Varchar(255) |            |
| Std_IC   | Student IC   | Varchar(12)  | FK         |
| Std_Form | Student Form | Date         |            |

Table Book

| Field            | Description        | Data Type    | Constraint |
|------------------|--------------------|--------------|------------|
| ID               | Book ID            | Int(11)      | PK         |
| Book_ISBN        | Book ISBN          | Varchar(255) | FK         |
| Book_Title       | Book Title         | Varchar(255) |            |
| Book_Author      | Author of the book | Date         |            |
| Book_Description | Book Description   | Varchar(255) |            |

Table Circulation

| Field       | Description            | Data Type    | Constraint |
|-------------|------------------------|--------------|------------|
| ID          | Circulation ID         | Int(11)      | PK         |
| Book_ID     | ID of borrowed Book    | Varchar(255) | FK         |
| Std_ID      | IC of student          | Varchar(255) | FK         |
| Borrow_Date | Date Borrow            | Date         |            |
| Return_Date | Date need to be return | Date         |            |

Table Check In

| Field        | Description   | Data Type   | Constraint |
|--------------|---------------|-------------|------------|
| ID           | Check In ID   | Int(11)     | PK         |
| Std_ID       | Student ID    | Varchar(20) | FK         |
| CheckIn_Date | Check In Date | Date        |            |

## 4. DETAIL DESIGN

---

### 4.1 Login Controller

| LoginController |                          |  |
|-----------------|--------------------------|--|
| Class Type      | Controller               |  |
| Responsible     | Manage All Login Request |  |
| Attributes      |                          |  |
| Methods         | Index()                  | Render a log in form                       |
|                 | Login()                  | Login the user and redirect to dashboard   |
|                 | Logout()                 | Log out the user and redirect to homepage. |

### 4.2 Hhomepage Controller

| HhomepageController |                         |  |
|---------------------|-------------------------|--|
| Class Type          | Controller              |  |
| Responsible         | Manage All User Request |  |
| Attributes          |                         |  |
| Methods             | Index()                 | Render homepage with list of book details      |
|                     | Search()                | Search books based on keywords entered by user |

### 4.3 Dashboard Controller

| DashboardController |                         |
|---------------------|-------------------------|
| Class Type          | Controller              |
| Responsible         | Manage All User Request |



|            |         |                         |
|------------|---------|-------------------------|
| Attributes |         |                         |
| Methods    | index() | Render a dashboard page |

| BookController |                         |   |
|----------------|-------------------------|---|
| Class Type     | Controller              |   |
| Responsible    | Manage All Book Request |   |
| Attributes     |                         |   |
| Methods        | Index()                 | Get all books data from model and then render View Book Interface with the books record |
|                | Create()                | Render add book form  |
|                | Store()                 | Call save() method from model to save the book data into db                             |
|                | Show()                  | Render Book details page  |
|                | Edit()                  | Render Edit Book Page   |
|                | Update()                | Calls a save() method in model to update existing record                                |
|                | Destroy()               | Calls a delete() method from model to delete record                                     |

| StudentController |                            |   |
|-------------------|----------------------------|---|
| Class Type        | Controller                 |   |
| Responsible       | Manage All Student Request |   |
| Attributes        |                            |   |
| Methods           | Import()                   | Calls a save() method in model in a loop to save all record |
|                   | Index()                    | Get all student data from                                   |

|  |           |  |
|--|-----------|--|
|  |           | model and then render view student with list of student record |
|  | Create()  | Render manual add student form                                 |
|  | Store()   | Call save() method from model to save the student data into db |
|  | Show()    | Render student details page                                    |
|  | Edit()    | Render Edit Student Page                                       |
|  | Update()  | Calls a save() method in model to update existing record       |
|  | Destroy() | Calls a delete() method from model to delete record            |

#### 4.4 Check In Controller

| CheckInController |                             |   |
|-------------------|-----------------------------|---|
| Class Type        | Controller                  |   |
| Responsible       | Manage All Check In Request |   |
| Attributes        |                             |   |
| Methods           | llondex()                   | Render a check in form                    |
|                   | checkIn()                   | Call a save() method in model to check in |

#### 4.5 Circulation Controller

| CirculationController |                           |
|-----------------------|---------------------------|
| Class Type            | Controller                |
| Responsible           | Manage All Report Request |
| Attributes            |                           |

|         |           |   |
|---------|-----------|---|
| Methods | Borrow()  | Render a borrow form  |
|         | Return()  | Render a return form  |
|         | Create()  | Call a save() method in model to create new record in circulation table |
|         | Destroy() | Call a delete() method in model to remove circulation record            |

#### 4.6 Report Controller

| ReportController |                           |  |
|------------------|---------------------------|--|
| Class Type       | Controller                |  |
| Responsible      | Manage All Report Request |  |
| Attributes       |                           |  |
| Methods          | Index()                   | Render a setting form  |
|                  | generateReport()          | Get all check in record and generate report based on the setting, then render the report page. |

#### 4.7 Book

| Book        |                   |         |
|-------------|-------------------|---------|
| Class Type  | Entity            |         |
| Responsible | Manage Book Table |         |
| Attributes  | ID                | : Int   |
|             | Book_ISBN         | :String |
|             | Book_Title        | :String |
|             | Book_Author       | :String |
| Methods     |                   |         |

**4.8 Student**

| Student     |                      |         |
|-------------|----------------------|---------|
| Class Type  | Entity               |         |
| Responsible | Manage Student Table |         |
| Attributes  | ID                   | : Int   |
|             | Student_Name         | :String |
|             | Student_IC           | :String |
|             | Student_Form         | :int    |
| Methods     |                      |         |

**4.9 Check In**

| CheckIn     |                       |        |
|-------------|-----------------------|--------|
| Class Type  | Entity                |        |
| Responsible | Manage Check In Table |        |
| Attributes  | ID                    | : Int  |
|             | Student_ID            | : Int  |
|             | CheckIn_Date          | : Date |
| Methods     |                       |        |

**4.10 Circulation**

| Circulation |                          |       |
|-------------|--------------------------|-------|
| Class Type  | Entity                   |       |
| Responsible | Manage Circulation Table |       |
| Attributes  | ID                       | : Int |
|             | STD_ID                   | : Int |
|             | Book_ID                  | : Int |

|         |  |
|---------|--|
| Methods |  |
|---------|--|

#### 4.11 User

| User        |                   |          |
|-------------|-------------------|----------|
| Class Type  | Entity            |          |
| Responsible | Manage User Table |          |
| Attributes  | ID                | : Int    |
|             | User_Name         | : String |
|             | User_Email        | : String |
|             | User_Password     | :String  |
|             | User_Role         | :String  |
| Methods     |                   |          |

**APPENDIX A**